Outlier Privacy

Edward Lui
Cornell University
luied@cs.cornell.edu

Rafael Pass*
Cornell University
rafael@cs.cornell.edu

January 20, 2015

Abstract

We introduce a generalization of differential privacy called *tailored differential privacy*, where an individual's privacy parameter is "tailored" for the individual based on the individual's data and the data set. In this paper, we focus on a natural instance of tailored differential privacy, which we call *outlier privacy*: an individual's privacy parameter is determined by how much of an "*outlier*" the individual is. We provide a new definition of an outlier and use it to introduce our notion of outlier privacy. Roughly speaking, $\epsilon(\cdot)$-*outlier privacy* requires that each individual in the data set is guaranteed "$\epsilon(k)$-differential privacy protection", where k is a number quantifying the "outlierness" of the individual. We demonstrate how to release accurate histograms that satisfy $\epsilon(\cdot)$-outlier privacy for various natural choices of $\epsilon(\cdot)$. Additionally, we show that $\epsilon(\cdot)$-outlier privacy with our weakest choice of $\epsilon(\cdot)$—which offers no explicit privacy protection for "non-outliers"—already implies a "distributional" notion of differential privacy w.r.t. a large and natural class of distributions.

*Pass is supported in part by an Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, NSF Award CCF-1214844, NSF Award CNS-1217821, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2-0211.

1 Introduction

Enormous amounts of data are collected by hospitals, social networking systems, government agencies, and other organizations. There are huge social benefits in analyzing this data, but we must protect the *privacy* of the individuals in the data. The current standard definition of privacy for data analysis is *differential privacy* [DMNS06, Dwo06], which requires that the output distribution of the data analysis algorithm changes very little when a single individual's data is added or removed from the data set. Accurate differentially private algorithms for a wide variety of tasks have been developed, allowing for useful and private data analysis (e.g., see [Dwo08, Dwo09]).

Currently, the standard notion of differential privacy guarantees the *same* level of privacy protection for all individuals. More precisely, in ϵ-differential privacy, every individual has the same "ϵ-differential privacy protection", which guarantees that the algorithm's output distribution changes by at most ϵ when adding or removing the individual's data from the data set. While this is a strong privacy guarantee if ϵ is very small (we elaborate more on this below), it clearly also does result in a non-trivial privacy loss for moderate values of ϵ. Additionally, it has also been established that to achieve non-trivial utility, ϵ cannot be too small—in particular, $\epsilon \gg 1/n$ where n is the number of individuals in the data set. Furthermore, to answer a counting query with ϵ-differential privacy and with error at most α, we must have $\epsilon \geq \Omega(1/\alpha)$.

An alternative idea is to provide *different levels of privacy* protection to different individuals—intuitively, some individuals require more privacy than others, and the algorithm should accommodate this. This general idea, which first appeared in the work of Ghosh and Roth [GR11], has been partly investigated in a mechanism design setting (e.g., see [GR11, FL12, LR12, RS12, NVX14]), where individuals are requested to not only submit their data, but also their "privacy valuation". The mechanism then tries to accommodate each individual's privacy valuation, while at the same time releasing data that is useful. Unfortunately, however, in the most realistic setting—where an individual's privacy valuation may be correlated with her data and thus also needs to be protected—the literature is plagued by strong impossibility results.

Tailored Differential Privacy: Protecting Outliers. In this paper, we consider a different approach to deal with the issue that different individuals may have different privacy needs. Instead of having the individuals specify their own privacy valuation/parameter, an individual's privacy parameter will be determined based on the individual's data and the data set. In other words, an individual's privacy parameter will be *tailored* for the individual based on the data set—we refer to such a notion as *tailored differential privacy*. In this paper, we focus on a natural instance of tailored differential privacy: an individual's privacy parameter will be determined by how much of an "*outlier*" the individual is (w.r.t. the data set). Roughly speaking, "outliers"—intuitively, individuals that are "far away", or "vastly different" from most other individuals—will be granted higher privacy protection than individuals that "mix" with lots of other individuals. One reason for providing higher privacy protection to outliers is that we may want to limit the amount of information leaked about a group of outliers. Let us present an example to illustrate what we mean.

Example 1 (Salaries of a Company's Employees)**.** Consider the standard ϵ-differentially private algorithm for releasing a histogram, which simply adds (Laplace) $Lap(1/\epsilon)$ noise[1] to each bin independently. Suppose such an algorithm is used to release a histogram of the salaries of a large company's employees, where the range of possible salaries is partitioned into intervals, which correspond to the bins of the histogram. Assume there exists a (small, but non-trivial) group

[1] $Lap(\lambda)$ is the Laplace distribution with mean 0 and scale λ, whose associated pdf is $f_\lambda(x) = \frac{1}{2\lambda}\exp(-\frac{|x|}{\lambda})$.

of, say, 100 managers, and all these managers have similar salaries that belong to the same bin; assume further that the other employees in the company have much lower salaries. Since the group of managers is relatively small, we consider them to be outliers and would like to prevent their (approximate) salary from being revealed. But, if ϵ is not small enough, by choosing the highest-salary bin with a noisy count of at least 50, the bin containing the managers can be predicted with "high" probability (roughly $1 - \exp(-50\epsilon)$).

Leaking the salary information of a small group of managers may perhaps not be considered a serious "breach" of their privacy. However, the same argument still holds if we further partition each salary bin into two sub-bins corresponding to HIV positive and HIV negative individuals. If the fraction of HIV positive managers is significantly higher than what is usual, this fact will released by the ϵ-differentially private algorithm (assuming ϵ is not too small).

In contrast, if we could provide sufficiently higher privacy protection (i.e., a sufficiently smaller privacy parameter) to each of the managers, then the amount of information leaked about the group of managers would be significantly less, and thus the managers' salary, or information about their HIV status, will not be (significantly) revealed.

In the above example, the managers are considered "outliers"—the group of outliers is "small" and other individuals in the data set are "far" from them; thus we consider it a violation of their privacy that sensitive information about them is leaked. In contrast, if the group of managers was "huge", we would no longer consider them outliers, and releasing aggregate information about a huge group of people should not be considered a violation of privacy. Indeed, note that in the above example, the sensitive information that is leaked is not about a *single* individual, it is about the *group* of managers; this clarifies why traditional differential privacy (which is only meant to mask a single individual's information) does not suffice to protect this information.

The notion of (k, ϵ)-*group differential privacy* (which in particular is implied by ϵ/k-differential privacy), on the other hand, could be used to protect information about the group of managers (if we let $k = 100$). But using such a strong notion of privacy would require adding noise proportional to $100/\epsilon$ to all the bins in the above example, and would render the released data useless. On the other hand, if we *tailor* the level of privacy required by an individual to whether the individual is an outlier or not (which, looking forward, will be enabled by our notion of *outlier privacy*), we could make sure to guarantee $(\epsilon/100)$-differential privacy for *only* the managers (and thus any information about the group of managers is protected), and only ϵ-differential privacy for everyone else.

Let us now turn to formalizing our notion of *outlier privacy*. Towards doing this, we first need to provide a mathematical definition of what if means for an individual to be an outlier.

A New Mathematical Definition of "Outliers". As mentioned above, intuitively, outliers are data points or records that are "far away" or "vastly different" from the rest of the data. There are many existing methods of identifying outliers (see [CBK09] for a survey); for example, for a set of data points, an outlier can be defined as a data point that is not within a certain distance of any other data point. However, such methods are often problematic for high-dimensional data (which is quite common), since the data points tend to be sparsely spaced and thus every data point may be an outlier (e.g., see [NS08]). As far as we know, all of the existing methods for identifying an outlier only look at the data itself and do not explicitly consider the *algorithm* that will be run on the data. In contrast, similar to the notion of differential privacy, we provide a definition of an outlier that depends on the algorithm that operates on the data set. (Additionally, existing methods of identifying outliers are also designed for some specific type of data (e.g., data points in $\mathbb{R}^d$); in contrast, we seek a method that works for any type of data.)

We aim to capture the intuition that a data record t in a data set is an outlier if, "from the perspective of the algorithm", the data record is not "equivalent" to sufficiently many data records in the data set. More formally, we say that a data record t is *equivalent* to another data record t' w.r.t. an algorithm A if A can never distinguish t and t'—that is, for every data set D containing t, the output distribution of the algorithm A does not change if we replace t by t' in D. (For instance, for computing a histogram, two individuals t and t' are equivalent if they correspond to the same bin in the histogram.) We now call a data record t a *k-outlier* w.r.t. the data set D and the algorithm A if t is equivalent (w.r.t. A) to at most k records in the data set. The parameter k quantifies to what extent the data record is an outlier.

Defining Outlier Privacy. We now turn to (informally) defining our notion of outlier privacy. Roughly speaking, *$\epsilon(\cdot)$-outlier privacy* requires that for every data set D, every $k > 0$, and every k-outlier t in the data set D, t is guaranteed "$\epsilon(k)$-differential privacy protection"—that is, if we remove t from the data set, the output distribution of the algorithm changes by at most $\epsilon(k)$, where the metric used is the same as that in differential privacy.

To address the privacy issues illustrated in Example 1, let us first consider $\epsilon(\cdot)$-outlier privacy for a specific "threshold" function $\epsilon(\cdot)$, which is specified by two parameters k and ϵ; we refer to the resulting notion as *(k, ϵ)-simple outlier privacy*. Roughly speaking, (k, ϵ)-simple outlier privacy requires ϵ/k-differential privacy for k-outliers, but does not have any privacy requirements for the other individuals. By requiring ϵ/k-differential privacy for k-outliers, (k, ϵ)-simple outlier privacy provides "(k, ϵ)-group differential privacy protection" for each *group* of k-outliers where the group size is at most k—that is, if we *simultaneously* remove k or fewer k-outliers from the data set, the output distribution of the algorithm changes by at most ϵ. (This fact follows from the observation that we can remove the k-outliers in the group one at a time, each time causing the output distribution to change by at most ϵ/k; since the group size is bounded by k, the total change in the output distribution is at most ϵ.)

Note that $(100, \epsilon)$-simple outlier privacy suffices to protect the privacy of the managers in Example 1. However, it does not protect the privacy of any of the other individuals. A minimal privacy guarantee would be to require that the managers' privacy is guaranteed (as a group) and everyone else gets the "individual" differential privacy guarantee; that is, we seek an algorithm that satisfies both $(100, \epsilon)$-simple outlier privacy, and ϵ-differential privacy. Again, this can be viewed as an instance of $\epsilon(\cdot)$-outlier privacy for a slightly different threshold function $\epsilon(\cdot)$. More precisely, our notion of *(k, ϵ)-simple outlier differential privacy* requires ϵ/k-differential privacy for k-outliers and ϵ-differential privacy for the other individuals.

(k, ϵ)-simple outlier differential privacy provides just *two* separate levels of privacy protection. We may also consider a more general instance of $\epsilon(\cdot)$-outlier privacy, which we refer to as *staircase outlier privacy*. In staircase outlier privacy, there are ℓ thresholds $k_1 > \ldots > k_\ell$, and $\ell + 1$ privacy parameters $\epsilon_0 > \ldots > \epsilon_\ell$, and we require that for every $1 \leq i \leq \ell$, every k_i-outlier is protected by ϵ_i-differential privacy; also, it is required that all the individuals are protected by ϵ_0-differential privacy by default.

1.1 Our Results

Our central results consist of demonstrating efficient algorithms for releasing accurate histograms that satisfy $\epsilon(\cdot)$-outlier privacy for various natural choices of $\epsilon(\cdot)$—in particular, we consider, simple outlier privacy, simple outlier differential privacy, staircase outlier privacy, and finally $\epsilon(\cdot)$-outlier privacy for a relatively general choice of $\epsilon(\cdot)$, and provide various (different) algorithms for releasing histograms that achieve these notions. Additionally, we show that the weakest notion of just

simple outlier privacy (recall that this notion only protects outliers, and requires no privacy protection for the other individuals)—which we demonstrate can be achieved using particularly simple algorithms—actually already implies a "distributional" notion of differential privacy, and thus also a distributional notion of simple outlier differential privacy. Roughly speaking, the distributional notion of differential privacy only requires the differential privacy property to hold if the data set is drawn from some class of distributions. The class of distributions can represent a set of possible distributions that contains the supposed "true distribution", or the class can represent a set of possible beliefs an adversary may have about the data set. In our result, we consider a large and natural class of distributions obtained by sampling from any population. Our class of distributions includes quite general distributions/beliefs based on biased and imperfect sampling from a population, in a setting where the adversary may even know whether certain individuals were sampled or not.

Algorithms for Simple, Simple Differentially Private, and Staircase Outlier Privacy. Let us start by giving an example of a (k, ϵ)-simple outlier private algorithm for releasing a histogram (recall that (k, ϵ)-simple outlier privacy requires ϵ/k-differential privacy for all k-outliers, and no privacy for everyone else). Consider an algorithm that computes a histogram but suppresses the counts for all bins that have a count $\leq k$. A data record t is a k-outlier if and only if its bin has a count $\leq k$, so by suppressing the counts of those bins to 0, we ensure that output of the algorithm does not change if t is removed from the database. Simple outlier privacy may seem like a weak privacy guarantee—after all, the privacy of non-outliers is not explicitly protected. However, we will show that simple outlier privacy in fact implies a certain distributional notion of differential privacy, which might provide sufficient privacy protection in many settings. Thus, simple outlier privacy already implies a distributional notion of simple outlier differential privacy.

Let us now turn to directly designing simple outlier differentially private algorithms. We are able to design a histogram algorithm that achieves (k, ϵ)-simple outlier differential privacy. Roughly speaking, the algorithm first adds sufficient noise to each bin to achieve ϵ-differential privacy; then, the algorithm goes through each bin of the histogram, and if the bin has a noisy count that is less than k, the algorithm adds sufficient noise to the bin to achieve ϵ/k-differential privacy. The algorithm then outputs the resulting noisy histogram.

Finally, by generalizing the above approach, we can design a histogram algorithm that achieves staircase outlier privacy. Roughly speaking, the algorithm first adds sufficient noise to each bin to achieve ϵ_0-differential privacy; then, the algorithm goes through each of the "levels (i.e., steps) of the staircase" starting from the top, and if a bin currently has a noisy count that is at most the threshold for the current level i, the algorithm adds sufficient noise to the bin to achieve ϵ_i-differential privacy. The algorithm then outputs the resulting noisy histogram.

Outlier Private Algorithms for General $\epsilon(\cdot)$. We also provide histogram algorithms that satisfy $\epsilon(\cdot)$-outlier privacy for a relatively general $\epsilon(\cdot)$. Let us provide some intuition for how the outlier private histogram algorithms work. The standard ϵ-differentially private algorithm for releasing a histogram simply adds (Laplace) $Lap(1/\epsilon)$ noise to each bin count independently. By adding $Lap(1/\epsilon)$ noise to each bin, when a data record t is removed from the data set, the output distribution over noisy histograms only changes by at most ϵ (w.r.t. the metric used in differential privacy). To achieve $\epsilon(\cdot)$-outlier privacy, the output distribution can only change by at most $\epsilon(k)$, where k is the count of t's bin (t is the data record that is removed). Thus, one may try adding $Lap(1/\epsilon(k))$ noise to each bin, where k is the count of the bin. However, this does not work, since the amount of noise added depends on the count k in a way that is too sensitive. In particular, when

we remove t from the data set and the count of t's bin decreases from k to $k-1$, the magnitude of the noise changes from $1/\epsilon(k)$ to $1/\epsilon(k-1)$, which changes the output distribution by more than $\epsilon(k)$.

One way to fix this problem is to add noise to the $\epsilon(\cdot)$ function, so that the $1/\epsilon(k)$ and the $1/\epsilon(k-1)$ become noisy and would be "ϵ'-close" for some $\epsilon' > 0$. To allow for a variety of solutions, we will consider using any algorithm $\mathcal{A}$ that approximates $\epsilon(\cdot)$ in a "differentially private" way—that is, $\mathcal{A}(k) \approx \mathcal{A}(k-1)$ for every $k > 0$. Then, we will add $\approx Lap(1/\mathcal{A}(k_b))$ noise to each bin b, where k_b is the count for bin b. This works as long as the noise magnitude $1/\mathcal{A}(k_b)$ is large enough; the noise magnitude $1/\epsilon(k_b)$ is large enough, but since $\mathcal{A}(k_b)$ only approximates $\epsilon(k_b)$, $\mathcal{A}(k_b)$ might be too large. Thus, we will also require that $\mathcal{A}(k_b)$ is at most $\epsilon(k_b)$ with very high probability.

Comparison to Related Work. There are some similarities between simple outlier privacy and the notion of crowd-blending privacy in [GHLP12]. Crowd-blending privacy uses a notion of "ϵ-blend", where $\epsilon > 0$, whereas in our definition of an outlier, we use a notion of equivalence w.r.t. the algorithm, which corresponds to ϵ-blend with $\epsilon = 0$. Also, in (k, ϵ)-simple outlier privacy, when removing a k-outlier, the output distribution is only allowed to change by at most ϵ/k, whereas in (k, ϵ)-crowd-blending privacy, the output distribution is allowed to change by at most ϵ. Our result that simple outlier privacy implies distributional differential privacy is somewhat similar to the result in [GHLP12] that states that if one combines a crowd-blending private algorithm with a natural pre-sampling step, the combined algorithm is zero-knowledge private (which implies differential privacy; see [GLP11]) if we view the population as the input data set to the combined algorithm. In contrast, our result achieves a distributional notion of differential privacy on the data set as opposed to the population, which is a different model and definition.

Our result that simple outlier privacy implies distributional differential privacy also has some similarities to a result in [BGKS13], where it is shown that a histogram algorithm that suppresses small counts achieves a notion of distributional differential privacy (slightly weaker than ours, since their definition permits choosing a simulator, but in our definition, the simulator has to be the algorithm itself), but for a class of distributions incomparable to the class we consider (the classes are somewhat similar, but neither is a subset of the other). However, our class of distributions includes distributions/beliefs based on biased and imperfect sampling (from a population) in a setting where the adversary may even know whether certain individuals were sampled or not; the class of distributions considered in [BGKS13] does not consider such an adversarial setting. Also, we consider the class of simple outlier private algorithms, which includes but is more general than just histogram algorithms that suppress small counts.

Some Remarks on Outlier Privacy. Our notion of $\epsilon(\cdot)$-outlier privacy usually does not satisfy composition; that is, if an algorithm A is $\epsilon_A(\cdot)$-outlier private and an algorithm B is $\epsilon_B(\cdot)$-outlier private, the composition of A and B is usually not $(\epsilon_A + \epsilon_B)(\cdot)$-outlier private. This is due to the fact that a k-outlier w.r.t. the composition of A and B might not be a k-outlier w.r.t. A or B.

In our definition of $\epsilon(\cdot)$-outlier privacy, a k-outlier t is guaranteed "$\epsilon(k)$-differential privacy protection"—that is, if we *remove* t from the data set, the output distribution of the algorithm only changes by at most $\epsilon(k)$. Note, however, that this does not mean that if we replace t with *any* other individual t', the output distribution of the algorithm only changes by at most $\epsilon(k)$. In particular, if we replace t with a "non-outlier" t', then the output distribution may change more significantly. More precisely, the only thing we can say about the change in the output distribution is that it is bounded by $\epsilon(k) + \epsilon(k')$ if t is an k-outlier and t' is an k'-outlier—this follows since removing t changes the output distribution by at most $\epsilon(k)$, and adding t' changes the output

distribution by at most $\epsilon(k')$.

Possible Future Directions and Additional Applications. Our results in this paper have focused mostly on histograms. To some extent, this is because our notion of an outlier is very liberal, due to the fact that our notion of equivalence between individuals is very strict (and thus it is "easier" to be classified as an outlier). One can consider generalizing our definition of a k-outlier to a (k, ϵ')-outlier, where the definition is the same except that (k, ϵ')-outlier uses ϵ'-blending (as in [GHLP12]) to define equivalence between individuals. If we are using a notion of outlier privacy that guarantees at least ϵ_0-differential privacy for every individual, then every individual would $2\epsilon_0$-blend with every other individual (by "transitivity"), so we should choose the blending parameter ϵ' to be smaller than $2\epsilon_0$. Using the definition of a (k, ϵ')-outlier in our various notions of outlier privacy, one can perhaps construct useful algorithms that satisfy these new notions of outlier privacy. For example, the algorithm in [GHLP12] for releasing synthetic data points would satisfy our generalized notion of (k, ϵ, ϵ')-simple outlier privacy where the notion of a (k, ϵ')-outlier is used. We leave the exploration of these generalized notions of outlier privacy for future work.

In the area of robust statistics, one of the main goals is to design statistical methods and estimators that are not significantly affected by outliers. A simple approach would be to first remove the outliers from the data set, and then apply non-robust statistical methods to the remaining data set. In order to use this approach, one needs a method of identifying outliers. Our mathematical definition of an outlier, or a variant of it, can be used to remove outliers before running non-robust statistical methods or algorithms on the data. Also, our notions of outlier privacy can be adapted to define a notion of "outlier robustness" for statistical computations. We leave the exploration of such ideas for future work.

2 Outlier Privacy

A *data set* is a finite *multiset* of *data records*, where a data record is simply an element of some fixed set X, which we refer to as the *data universe*. Let $\mathcal{D}$ be the set of all data sets. Given a data set D and data records t and t', let $D_{-t} = D \setminus \{t\}$ and $(D, t') = D \uplus \{t'\}$. Given $\epsilon, \delta \geq 0$ and two random variables (or distributions) Z and Z', we shall write $Z \approx_{\epsilon,\delta} Z'$ to mean that for every $Y \subseteq \mathrm{Supp}(Z) \cup \mathrm{Supp}(Z')$, we have

$$\Pr[Z \in Y] \leq e^{\epsilon} \Pr[Z' \in Y] + \delta$$

and

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

We shall also write $Z \approx_{\epsilon} Z'$ to mean $Z \approx_{\epsilon,0} Z'$. Differential privacy ([DMNS06, Dwo06]) can now be defined in the following manner:

Definition 1 ((ϵ, δ)-differential privacy [DMNS06, Dwo06])**.** An algorithm $\mathcal{M}$ is said to be (ϵ, δ)-*differentially private* if for every pair of data sets D and D' differing in only one data record, we have $\mathcal{M}(D) \approx_{\epsilon,\delta} \mathcal{M}(D')$.

Intuitively, differential privacy protects the privacy of each individual by requiring the output distribution of the algorithm to not change much when an individual's data is added or removed from the data set. Achieving differential privacy often involves adding noise drawn from some distribution, usually the Laplace distribution. We will use $Lap(\lambda)$ to denote the Laplace distribution

with mean 0 and scale λ, whose associated pdf is $f_\lambda(x) = \frac{1}{2\lambda}\exp(-\frac{|x|}{\lambda})$. For convenience, we will sometimes abuse notation and use $Lap(\lambda)$ to denote a random variable that has the Laplace distribution $Lap(\lambda)$.

We now define our notion of *tailored differential privacy* as described in the introduction. Roughly speaking, $(\epsilon(\cdot), \delta(\cdot))$-tailored differential privacy requires that each individual t in the data set D is protected by $(\epsilon(t, D), \delta(t, D))$-differential privacy, where $\epsilon(\cdot)$ and $\delta(\cdot)$ are *functions* that, on input a data record t and a data set D, outputs privacy parameters $\epsilon(t, D)$ and $\delta(t, D)$ for t. Recall that X is the set of possible data records, and $\mathcal{D}$ is the set of all data sets.

Definition 2 (tailored differential privacy)**.** Let $\epsilon(\cdot), \delta(\cdot) : X \times \mathcal{D} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$. An algorithm $\mathcal{M}$ is said to be $(\epsilon(\cdot), \delta(\cdot))$-*tailored differentially private* if for every data set D and every data record $t \in D$, we have $\mathcal{M}(D) \approx_{\epsilon(t,D),\delta(t,D)} \mathcal{M}(D \setminus \{t\})$.

In this paper, we focus on a specific instance of tailored differential privacy, which we call *outlier privacy*. Outlier privacy tailors an individual's privacy parameter to the "outlierness" of the individual. Let us first describe our definition of an *outlier*. In the definitions below, let $\mathcal{M}$ be any algorithm that takes a data set as input. Roughly speaking, we say that a pair of data records $t, t' \in X$ are *equivalent w.r.t.* $\mathcal{M}$ (or $\mathcal{M}$-*equivalent*), denoted $t \equiv_{\mathcal{M}} t'$, if the algorithm $\mathcal{M}$ can never distinguish the two data records, regardless of the input data set.

Definition 3 (equivalent w.r.t. $\mathcal{M}$, or $\mathcal{M}$-equivalent)**.** Given a pair of data records $t, t' \in X$, we say that t *is equivalent to* t' *w.r.t.* $\mathcal{M}$, or t *is* $\mathcal{M}$-*equivalent to* t', denoted $t \equiv_{\mathcal{M}} t'$, if for every data set D' containing t, we have $\mathcal{M}(D') = \mathcal{M}(D'_{-t}, t')$ (in distribution).

Using the definition of a pair of data records being equivalent w.r.t. an algorithm $\mathcal{M}$, we now define the notion of a k-*outlier*. Roughly speaking, a k-outlier is a data record that is $\mathcal{M}$-equivalent to at most k data records in the data set (including itself).

Definition 4 (k-outlier)**.** Given a data set D, a data record $t \in D$ is said to be a k-*outlier in* D *w.r.t.* $\mathcal{M}$ if there are at most k data records in D that are equivalent to t w.r.t. $\mathcal{M}$.

As the parameter k increases, the property of being a k-outlier becomes weaker (i.e., easier to satisfy), and the set of k-outliers becomes larger. Using the definition of a k-outlier, we now define our new notion of privacy called $(\epsilon(\cdot), \delta(\cdot))$-*outlier privacy*. Roughly speaking, $(\epsilon(\cdot), \delta(\cdot))$-outlier privacy requires that for every $k > 0$ and every k-outlier t in the data set, t is protected by $(\epsilon(k), \delta(k))$-differential privacy—that is, if we remove t from the data set, the output distribution of the algorithm changes by at most $(\epsilon(k), \delta(k))$, where the metric used is the same as that in (ϵ, δ)-differential privacy.

Definition 5 ($(\epsilon(\cdot), \delta(\cdot))$-outlier privacy)**.** Let $\epsilon(\cdot), \delta(\cdot) : \mathbb{N} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$. An algorithm $\mathcal{M}$ is said to be $(\epsilon(\cdot), \delta(\cdot))$-*outlier private* if for every data set D, every $k > 0$, and every k-outlier t in D, we have $\mathcal{M}(D) \approx_{\epsilon(k),\delta(k)} \mathcal{M}(D \setminus \{t\})$.

We will often write $\epsilon(\cdot)$-*outlier private* to mean $(\epsilon(\cdot), \delta(\cdot))$-outlier private with $\delta(k) = 0$ for every k. $(\epsilon(\cdot), \delta(\cdot))$-outlier privacy generalizes differential privacy by allowing one to specify different levels of privacy protection for different individuals based on how much of an outlier the individuals are. Intuitively, one may want to provide greater privacy protection to outliers, since their privacy may be more at risk. By setting $\epsilon(\cdot)$ and $\delta(\cdot)$ to be constants ϵ and δ respectively, one recovers the definition of (ϵ, δ)-differential privacy.

2.1 Simple Outlier Privacy

Let us first consider $\epsilon(\cdot)$-outlier privacy with a specific $\epsilon(\cdot)$ function, together which we call (k, ϵ)-*simple outlier privacy*. Roughly speaking, (k, ϵ)-simple outlier privacy requires ϵ/k-differential privacy for k-outliers, but does not have any privacy requirements for the other individuals.

Definition 6 ((k, ϵ)-simple outlier privacy)**.** Let $k, \epsilon > 0$. An algorithm $\mathcal{M}$ is said to be (k, ϵ)-*simple outlier private* if for every data set D and every k-outlier t in D, we have $\mathcal{M}(D) \approx_{\epsilon/k} \mathcal{M}(D \setminus \{t\})$.

(k, ϵ)-simple outlier privacy is equivalent to $\epsilon(\cdot)$-outlier privacy with the function $\epsilon(\cdot)$ defined by $\epsilon(k') = \epsilon/k$ if $k' \leq k$, and $\epsilon(k') = \infty$ otherwise. By requiring ϵ/k-differential privacy for k-outliers, (k, ϵ)-simple outlier privacy provides "(k, ϵ)-group differential privacy protection" for each *group* of k-outliers where the group size is at most k—that is, if we *simultaneously* remove k or fewer k-outliers from the data set, the output distribution of the algorithm changes by at most ϵ. (This fact follows from the observation that we can remove the k-outliers in the group one at a time, each time causing the output distribution to change by at most ϵ/k; since the group size is bounded by k, the total change in the output distribution is at most ϵ.) This privacy protection for groups of k-outliers can be particularly useful when one needs to protect the privacy of a group of outliers. In some cases, in order to protect the privacy of a single outlier, one needs to protect the privacy of an entire group of outliers simultaneously. In such cases, ordinary differential privacy may not be sufficient, like in Example 1 in the introduction. For completeness, let us now formalize what we mean when we say that (k, ϵ)-simple outlier privacy provides "(k, ϵ)-group differential privacy protection" for each group of k-outliers where the group size is at most k.

Proposition 7. *Let $\mathcal{M}$ be any algorithm that is (k, ϵ)-simple outlier private. Then, for every data set D and every $A \subseteq D$ of size at most k and consisting of only k-outliers in D, we have $\mathcal{M}(D) \approx_\epsilon \mathcal{M}(D \setminus A)$.*

Proof. Let D be any data set, and let $A \subseteq D$ be of size at most k and consisting of only k-outliers in D. Let $A = \{t_1, \ldots, t_r\}$, where $r \leq k$. Now, for $i = 0, \ldots, r$, let $D^{(i)} = D \setminus \{t_1, \ldots, t_i\}$. We note that $D^{(0)} = D$ and $D^{(r)} = D \setminus A$. Since $\mathcal{M}$ is (k, ϵ)-simple outlier private and A only consists of k-outliers in D, and since k-outliers in D remain as k-outliers after removing data records from D, we have $\mathcal{M}(D^{(i)}) \approx_{\epsilon/k} \mathcal{M}(D^{(i+1)})$ for every $0 \leq i \leq r - 1$. Thus, we have $\mathcal{M}(D) \approx_\epsilon \mathcal{M}(D \setminus A)$, as required. □

Let us now give some examples of simple outlier private algorithms. Our first example is an algorithm that computes a histogram but suppresses the small counts to 0. Intuitively, data records in the same bin are equivalent w.r.t. $\mathcal{M}$, while a pair of data records belonging to separate bins are not equivalent w.r.t. $\mathcal{M}$. Thus, a data record is a k-outlier if and only if its bin has a count $\leq k$, so to achieve $(k, 0)$-simple outlier privacy, the algorithm "suppresses" the counts $\leq k$ to 0.

Example 2 (Simple Outlier Private Histogram with Suppression of Small Counts)**.** Let $k > 0$. Let $\mathcal{M}$ be an algorithm that, on input a data set D, computes a histogram from D, and then for every bin count that is $\leq k$, $\mathcal{M}$ "suppresses" (i.e., changes) the bin count to 0. $\mathcal{M}$ then outputs the modified histogram.

Theorem 8. *The above algorithm $\mathcal{M}$ is $(k, 0)$-simple outlier private.*

Proof. Let D be any data set, and let t be any k-outlier in D. We note that t is $\mathcal{M}$-equivalent to precisely those records that belong in the same bin as t. Since t is a k-outlier, there are at most k records in t's bin. Thus, $\mathcal{M}$ will suppress t's bin count to 0. We observe that removing t from the data set (and thus from t's bin) will still result in $\mathcal{M}$ suppressing t's bin count to 0. Thus, $\mathcal{M}$ is $(k, 0)$-simple outlier private. □

Instead of suppressing small counts to 0, one can add noise to the small counts to achieve (k, ϵ)-simple outlier privacy.

Example 3 (Simple Outlier Private Histogram with Noise Added to Small Counts)**.** Let $k > 0$. Let $\mathcal{M}$ be an algorithm that, on input a data set D, computes a histogram from D, and then for each bin count that is $\leq k$, $\mathcal{M}$ adds $Lap(k/\epsilon)$ noise to the bin count independently. $\mathcal{M}$ then outputs the modified histogram.

Theorem 9. *The above algorithm $\mathcal{M}$ is (k, ϵ)-simple outlier private.*

Proof. Let D be any data set, and let t be any k-outlier in D. We note that t is $\mathcal{M}$-equivalent to precisely those records that belong in the same bin as t. Since t is a k-outlier, there are at most k records in t's bin. Thus, $\mathcal{M}$ will add $Lap(k/\epsilon)$ noise to t's bin count. We observe that removing t from the data set (and thus from t's bin) will still result in $\mathcal{M}$ adding $Lap(k/\epsilon)$ noise to t's bin count; using the pdf of $Lap(k/\epsilon)$ and performing some standard calculations for proving differential privacy (e.g., see [DMNS06]), one can easily show that the noisy count of t's bin after removing t is ϵ/k-close (i.e., $\approx_{\epsilon/k}$) to the noisy count of t's bin before removing t. Thus, $\mathcal{M}$ is (k, ϵ)-simple outlier private. □

The simple outlier private algorithms above also satisfy a distributional notion of differential privacy for a large and natural class of distributions, since simple outlier privacy implies such a distributional notion of differential privacy, which we show in Section 3.

Relationship of Simple Outlier Privacy to Other Privacy Definitions. Since (k, ϵ)-simple outlier privacy requires ϵ/k-differential privacy for k-outliers (and no privacy guarantee for the other individuals), we see that ϵ/k-differential privacy implies (k, ϵ)-simple outlier privacy.

Proposition 10. *Let $k, \epsilon > 0$. If an algorithm $\mathcal{M}$ is ϵ/k-differentially private, then it is (k, ϵ)-simple outlier private.*

Proof. This follows immediately from the definition of ϵ/k-differential privacy and (k, ϵ)-simple outlier privacy. □

Although (k, ϵ)-simple outlier privacy can be obtained by achieving ϵ/k-differential privacy, achieving ϵ/k-differential privacy normally requires substantially more "noise" to be added. As demonstrated in the above examples, one can achieve better accuracy/utility with (k, ϵ)-simple outlier privacy because only the k-outliers require ϵ/k-differential privacy.

In [GHLP12], a notion of a pair of data records "ϵ-blending with each other" is used (in their notion of crowd-blending privacy), where it is required that the algorithm cannot distinguish the two records by more than ϵ. More precisely, a data record t *ϵ-blends with* t' w.r.t. $\mathcal{M}$ if for every data set D' containing t, we have $\mathcal{M}(D') \approx_\epsilon \mathcal{M}(D'_{-t}, t')$. In this paper, in our definition of equivalence w.r.t. $\mathcal{M}$ and in our definition of a k-outlier, we require the "blending" to be perfect (i.e., $\epsilon = 0$), since for an $(\epsilon/2)$-differentially private algorithm, every record ϵ-blends with every other record, and thus there would be no outliers. Furthermore, by setting $\epsilon = 0$, the "blends with" relation is an equivalence relation on the set of all possible data records. For an algorithm releasing histograms, the equivalence classes are precisely the bins of the histogram. In other words, a pair of data records blend with one another if and only if they belong to the same bin. There are also some similarities between simple outlier privacy and the notion of crowd-blending privacy in [GHLP12], which we now recall.

Definition 11 (Crowd-blending privacy [GHLP12])**.** An algorithm $\mathcal{M}$ is (k, ϵ)-*crowd-blending private* if for every data set D and every data record $t \in D$, at least one of the following conditions hold:

- There are at least k data records in D that ϵ-blend with t.
- $\mathcal{M}(D) \approx_\epsilon \mathcal{M}(D \setminus \{t\})$

The first condition in crowd-blending privacy is roughly saying that t is not a $(k-1)$-outlier, except that in the definition of $(k-1)$-outlier, the weaker notion of ϵ-blending is used instead of 0-blend. In the second condition, when t is removed from D, the output distribution of $\mathcal{M}$ changes by at most ϵ, but in (k, ϵ)-simple outlier privacy, the output distribution of $\mathcal{M}$ is only allowed to change by at most ϵ/k (for reasons we have explained above). We now formally show that simple outlier privacy implies crowd-blending privacy.

Proposition 12. *If an algorithm $\mathcal{M}$ is (k, ϵ)-simple outlier private, then it is $(k+1, \epsilon/k)$-crowd-blending private.*

Proof. Suppose an algorithm $\mathcal{M}$ is (k, ϵ)-simple outlier private. We will show that $\mathcal{M}$ is also $(k+1, \epsilon/k)$-crowd-blending private. Let D be any data set, let $t \in D$, and let A be the multiset of all data records t' in D such that $t' \equiv_{\mathcal{M}} t$. If A is of size at least $k+1$, then the first property in $(k+1, \epsilon)$-crowd-blending privacy holds. Otherwise, t is a k-outlier in D, so by the definition of (k, ϵ)-simple outlier privacy, we have $\mathcal{M}(D) \approx_{\epsilon/k} \mathcal{M}(D \setminus \{t\})$, which is the second property in $(k+1, \epsilon/k)$-crowd-blending privacy. □

2.2 Simultaneously Achieving Simple Outlier Privacy and Differential Privacy

Although (k, ϵ)-simple outlier privacy protects the privacy of k-outliers, there is no privacy guarantee for the other individuals. Thus, we now consider a stronger notion of outlier privacy that provides ϵ/k-differential privacy for k-outliers and ϵ-differential privacy for everyone else. In other words, the stronger notion of outlier privacy provides both (k, ϵ)-simple outlier privacy and ϵ-differential privacy. We call this notion of outlier privacy *simple outlier differential privacy*. We first generalize (k, ϵ)-simple outlier privacy to (k, ϵ, δ)-simple outlier privacy so that we can define (k, ϵ, δ)-simple outlier differential privacy.

Definition 13 ((k, ϵ, δ)-simple outlier privacy)**.** Let $k, \epsilon > 0$. An algorithm $\mathcal{M}$ is said to be (k, ϵ, δ)-*simple outlier private* if for every data set D and every k-outlier t in D, we have $\mathcal{M}(D) \approx_{\epsilon/k, \delta} \mathcal{M}(D \setminus \{t\})$.

We now define (k, ϵ, δ)-simple outlier differential privacy.

Definition 14 ((k, ϵ, δ)-simple outlier differential privacy)**.** Let $k, \epsilon > 0$. An algorithm $\mathcal{M}$ is said to be (k, ϵ, δ)-*simple outlier differentially private* if $\mathcal{M}$ is (k, ϵ, δ)-simple outlier private and (ϵ, δ)-differentially private.

We will write (k, ϵ)-*simple outlier differentially private* to mean (k, ϵ, δ)-simple outlier differentially private with $\delta = 0$. In the definition of (k, ϵ, δ)-simple outlier differential privacy, the same parameters ϵ and δ are used for both the simple outlier privacy requirement and the differential privacy requirement; however, one can easily consider a more general definition where separate parameters are used for the two requirements. (k, ϵ)-simple outlier differential privacy is equivalent to $\epsilon(\cdot)$-outlier privacy with the function $\epsilon(\cdot)$ defined by $\epsilon(k') = \epsilon/k$ if $k' \leq k$, and $\epsilon(k') = \epsilon$ otherwise. We now describe an algorithm for releasing histograms that achieves simple outlier differential privacy.

Example 4 (Simple Outlier Differentially Private Histogram with Suppression of Small Counts)**.** Let $k, \alpha, \epsilon > 0$. Let $\mathcal{M}$ be an algorithm that, on input a data set D, computes a histogram from D, and then adds $Lap(1/\epsilon)$ noise to each bin count independently. Then, for every new (noisy) bin count that is $\leq k + \alpha/\epsilon$, $\mathcal{M}$ "suppresses" the bin count to 0. $\mathcal{M}$ then outputs the modified histogram.

Theorem 15. *The above algorithm $\mathcal{M}$ is $(k, \epsilon, e^{-\alpha}/2)$-simple outlier differentially private.*

Proof. We first show that $\mathcal{M}$ is ϵ-differentially private. We note that $\mathcal{M}$ first computes a noisy histogram using the standard ϵ-differentially private algorithm for releasing a noisy histogram. After that, $\mathcal{M}$ does not look at the input data set anymore, so the output of $\mathcal{M}$ is simply a post-processing of the output of an ϵ-differentially private algorithm. Thus, $\mathcal{M}$ itself is ϵ-differentially private.

We now show that $\mathcal{M}$ is $(k, 0, e^{-\alpha}/2)$-simple outlier private. Let D be any data set, and let t be any k-outlier in D. We need to show that $\mathcal{M}(D) \approx_{0, e^{-\alpha}/2} \mathcal{M}(D \setminus \{t\})$. It suffices to show that regardless of whether the data set is D or $D \setminus \{t\}$, we have that with probability at least $1 - e^{-\alpha}/2$, $\mathcal{M}$ will suppress t's bin count to 0. This event occurs precisely when the new (noisy) count for t's bin is $\leq k + \alpha/\epsilon$. Since t is a k-outlier, there are at most k records in t's bin (before any noise is added), so the probability of this event is at least the probability that $Lap(1/\epsilon) \leq \alpha/\epsilon$. One can easily verify that this latter event occurs with probability at least $1 - e^{-\alpha}/2$, as required. □

In the above example, instead of suppressing the noisy bin count to 0, the algorithm $\mathcal{M}$ can add $Lap(k/\epsilon)$ noise to the noisy bin count. Let us now describe such an algorithm more formally.

Example 5 (Simple Outlier Differentially Private Histogram with Noise Added to Small Counts)**.** Let $k, \alpha, \epsilon > 0$. Let $\mathcal{M}$ be an algorithm that, on input a data set D, computes a histogram from D, and then adds $Lap(1/\epsilon)$ noise to each bin count independently. Then, for every new (noisy) bin count that is $\leq k + \alpha/\epsilon$, $\mathcal{M}$ adds $Lap(k/\epsilon)$ noise to the noisy bin count. $\mathcal{M}$ then outputs the modified histogram.

Theorem 16. *The above algorithm $\mathcal{M}$ is $(k, \epsilon, e^{-\alpha})$-simple outlier differentially private.*

Proof. We first show that $\mathcal{M}$ is ϵ-differentially private. We note that $\mathcal{M}$ first computes a noisy histogram using the standard ϵ-differentially private algorithm for releasing a noisy histogram. After that, $\mathcal{M}$ does not look at the input data set anymore, so the output of $\mathcal{M}$ is simply a post-processing of the output of an ϵ-differentially private algorithm. Thus, $\mathcal{M}$ itself is ϵ-differentially private.

We now show that $\mathcal{M}$ is $(k, \epsilon, e^{-\alpha})$-simple outlier private. Let D be any data set, and let t be any k-outlier in D. We need to show that $\mathcal{M}(D) \approx_{\epsilon/k, e^{-\alpha}} \mathcal{M}(D \setminus \{t\})$. We first show that regardless of whether the data set is D or $D \setminus \{t\}$, we have that with probability at least $1 - e^{-\alpha}/2$, the first noisy count for t's bin is $\leq k + \alpha/\epsilon$ (this is the condition that determines whether $Lap(k/\epsilon)$ noise will be further added to the noisy bin count). Since t is a k-outlier, there are at most k records in t's bin (before any noise is added), so the probability of this event is at least the probability that $Lap(1/\epsilon) \leq \alpha/\epsilon$. One can easily verify that this latter event occurs with probability at least $1 - e^{-\alpha}/2$, as required.

Now, let $\mathcal{M}'$ be the same as $\mathcal{M}$ except that for t's bin, instead of checking the condition that the first noisy count for t's bin is $\leq k + \alpha/\epsilon$, $\mathcal{M}'$ simply pretends that the condition is true. Then, we have $\mathcal{M}(D) \approx_{0, e^{-\alpha}/2} \mathcal{M}'(D)$ and $\mathcal{M}(D \setminus \{t\}) \approx_{0, e^{-\alpha}/2} \mathcal{M}'(D \setminus \{t\})$. Thus, to show that $\mathcal{M}(D) \approx_{\epsilon/k, e^{-\alpha}} \mathcal{M}(D \setminus \{t\})$, it suffices to show that $\mathcal{M}'(D) \approx_{\epsilon/k} \mathcal{M}'(D \setminus \{t\})$. Since $\mathcal{M}'$ adds $Lap(k/\epsilon)$ noise to t's bin count, it is easy to show using standard calculations that $\mathcal{M}'(D) \approx_{\epsilon/k} \mathcal{M}'(D \setminus \{t\})$, as required. □

Revisiting the "Salaries of a Company's Employees" Example. The above simple outlier differentially private histogram algorithms can be used to protect the privacy of the managers and the other employees in the example described in the introduction. As mentioned previously, one can also protect the privacy of the managers by using a group differentially private algorithm for releasing a histogram. For comparison, let us now describe the standard group differentially private algorithm for releasing a histogram.

Example 6 (The Standard Group Differentially Private Histogram)**.** Let $k, \epsilon > 0$. Let $\mathcal{M}$ be an algorithm that, on input a data set D, computes a histogram from D, and then adds $Lap(k/\epsilon)$ noise to each bin count independently. $\mathcal{M}$ then outputs the modified histogram.

It is known that the algorithm $\mathcal{M}$ is (k, ϵ)-group differentially private (e.g., see [DMNS06]).

As we can see, the standard group differentially private histogram algorithm adds $Lap(k/\epsilon)$ noise to *all* the bins, including the bins with many individuals in them. Our simple outlier differentially private algorithms suppress or add $\approx Lap(k/\epsilon)$ noise (depending on which variant we are using) to only the bins that contain outliers, and for the other bins, our algorithms only add $Lap(1/\epsilon)$ noise, which is substantially less than $Lap(k/\epsilon)$ noise. Thus, in the "Salaries of a Company's Employees" example, our algorithms have much better accuracy.

2.3 Staircase Outlier Privacy

In simple outlier differential privacy, there are only *two* separate levels of privacy protection: ϵ/k-differential privacy for k-outliers, and ϵ-differential privacy for everyone else. We can generalize this notion of outlier privacy to have more than two levels of privacy protection. We call this generalized notion *staircase outlier privacy*. In staircase outlier privacy, there are ℓ thresholds $k_1 > \ldots > k_\ell$, and $\ell + 1$ privacy parameters $\epsilon_0 > \ldots > \epsilon_\ell$, and we require that for every $1 \leq i \leq \ell$, every k_i-outlier is protected by (ϵ_i, δ)-differential privacy; also, it is required that all the individuals are protected by (ϵ_0, δ)-differential privacy by default.

Definition 17 (Staircase Outlier Privacy)**.** Let $\ell > 0$, let $k_1 > \ldots > k_\ell > 0$, let $\infty \geq \epsilon_0 > \epsilon_1 > \ldots > \epsilon_\ell \geq 0$, and let $\delta \geq 0$. An algorithm $\mathcal{M}$ is said to be $((k_1, \ldots, k_\ell), (\epsilon_0, \ldots, \epsilon_\ell), \delta)$-*staircase outlier private* if $\mathcal{M}$ is (ϵ_0, δ)-differentially private, and for every data set D, every $1 \leq i \leq \ell$, and every k_i-outlier t in D, we have $\mathcal{M}(D) \approx_{\epsilon_i, \delta} \mathcal{M}(D \setminus \{t\})$.

We will write $((k_1, \ldots, k_\ell), (\epsilon_0, \ldots, \epsilon_\ell))$-*staircase outlier private* to mean $((k_1, \ldots, k_\ell), (\epsilon_0, \ldots, \epsilon_\ell), \delta)$-staircase outlier private with $\delta = 0$. In the above definition, a single δ parameter is used, but one can easily generalize the above definition to allow for $\ell + 1$ different levels of δ: $\delta_0 > \delta_1 > \ldots > \delta_\ell$. Staircase outlier privacy generalizes simple outlier privacy and simple outlier differential privacy: (k, ϵ)-simple outlier privacy is equivalent to $(k, (\infty, \epsilon/k))$-staircase outlier privacy, and (k, ϵ, δ)-simple outlier differential privacy is equivalent to $(k, (\epsilon, \epsilon/k), \delta)$-staircase outlier privacy. $((k_1, \ldots, k_\ell), (\epsilon_0, \ldots, \epsilon_\ell), \delta)$-staircase outlier privacy is equivalent to $(\epsilon(\cdot), \delta)$-outlier privacy with a "staircase" $\epsilon(\cdot) : \mathbb{N} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ function, where $\epsilon(k) = \epsilon_0$ if $k > k_1$, $\epsilon(k) = \epsilon_1$ if $k_2 < k \leq k_1$, $\epsilon(k) = \epsilon_2$ if $k_3 < k \leq k_2$, and so forth. More formally, $\epsilon(\cdot)$ is defined by $\epsilon(k) = \epsilon_j$, where j is the smallest integer such that $k \leq k_j$, and $j = 0$ if no such integer exists.

For convenience and simplicity, we will define $x/0 = \infty$ and $x/\infty = 0$ for any real $x > 0$. Also, "adding $Lap(\infty)$ noise" to some value means suppressing (i.e., changing) the value to 0, and "adding $Lap(0)$ noise" to some value means adding no noise at all to the value, i.e., the value is left unmodified. Let us now describe a histogram algorithm that achieves staircase outlier privacy. Roughly speaking, the algorithm first adds noise to each bin to achieve ϵ_0-differential privacy; then, the algorithm goes through each of the "levels of the staircase" starting from the top, and if a bin

currently has a noisy count that is at most the threshold for that level, the algorithm adds sufficient noise to the bin to achieve ϵ_i-differential privacy. The algorithm then outputs the resulting noisy histogram.

Example 7 (Staircase Outlier Private Algorithm for Releasing a Histogram)**.** Let $\ell > 0$, let $k_1 > \ldots > k_\ell > 0$, and let $\infty \geq \epsilon_0 > \epsilon_1 > \ldots > \epsilon_\ell \geq 0$. Let $\alpha > 0$, and let $\mathcal{M}$ be an algorithm that, on input a data set D, computes a histogram from D, and then adds $Lap(1/\epsilon_0)$ noise to each bin count independently. Then, for $i = 1, \ldots, \ell$, $\mathcal{M}$ does the following: For every current noisy bin count that is $\leq k_i + (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i-1})$, $\mathcal{M}$ adds $Lap(1/\epsilon_i)$ noise to the current noisy bin count. $\mathcal{M}$ then outputs the modified histogram.

Theorem 18. *The above algorithm $\mathcal{M}$ is $((k_1, \ldots, k_\ell), (\epsilon_0, \ldots, \epsilon_\ell), \ell e^{-\alpha})$-staircase outlier private.*

Proof. We first show that $\mathcal{M}$ is ϵ_0-differentially private. We note that $\mathcal{M}$ first computes a noisy histogram using the standard ϵ_0-differentially private algorithm for releasing a noisy histogram. After that, $\mathcal{M}$ does not look at the input data set anymore, so the output of $\mathcal{M}$ is simply a post-processing of the output of an ϵ_0-differentially private algorithm. Thus, $\mathcal{M}$ itself is ϵ_0-differentially private.

We now show that for every data set D, every $1 \leq i \leq \ell$, and every k_i-outlier t in D, we have $\mathcal{M}(D) \approx_{\epsilon_i, \ell e^{-\alpha}} \mathcal{M}(D \setminus \{t\})$. Let D be any data set, let $1 \leq i \leq \ell$, and let t be any k_i-outlier in D. We need to show that $\mathcal{M}(D) \approx_{\epsilon_i, \ell e^{-\alpha}} \mathcal{M}(D \setminus \{t\})$. We first show that regardless of whether the data set is D or $D \setminus \{t\}$, we have that with probability at least $1 - \ell e^{-\alpha}/2$, it holds that at every iteration $i' \leq i$ in the algorithm $\mathcal{M}$, the condition that the current noisy count for t's bin is $\leq k_{i'} + (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1})$ is true. We note that this holds if for $i' = 0, \ldots, i-1$, the noise $Lap(1/\epsilon_{i'})$ added by $\mathcal{M}$ is $\leq \alpha/\epsilon_{i'}$ (note that the original true count of t's bin is $\leq k_{i'}$, since t is a k_i-outlier and $k_i \leq k_{i'}$). One can easily verify that each of these latter events occurs with probability at least $1 - e^{-\alpha}/2$. Thus, by the union bound, with probability at least $1 - \ell e^{-\alpha}/2$, it holds that at every iteration $i' \leq i$ in the algorithm $\mathcal{M}$, the condition that the noisy count for t's bin is $\leq k_{i'} + (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1})$ is true.

Let $\mathcal{M}'$ be the same as $\mathcal{M}$ except that for every iteration $i' \leq i$, instead of checking the condition that the current noisy bin count for t's bin is $\leq k_{i'} + (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1})$, $\mathcal{M}'$ simply pretends that the condition is true. Then, we have $\mathcal{M}(D) \approx_{0, \ell e^{-\alpha}/2} \mathcal{M}'(D)$ and $\mathcal{M}(D \setminus \{t\}) \approx_{0, \ell e^{-\alpha}/2} \mathcal{M}'(D \setminus \{t\})$. Thus, to show that $\mathcal{M}(D) \approx_{\epsilon_i, \ell e^{-\alpha}} \mathcal{M}(D \setminus \{t\})$, it suffices to show that $\mathcal{M}'(D) \approx_{\epsilon_i} \mathcal{M}'(D \setminus \{t\})$. Since $\mathcal{M}'$ adds $Lap(1/\epsilon_i)$ noise to t's bin during iteration i, and since all the computation afterwards can be viewed as post-processing, it is easy to show using standard calculations that $\mathcal{M}'(D) \approx_{\epsilon_i} \mathcal{M}'(D \setminus \{t\})$, as required. □

In the above example, the algorithm $\mathcal{M}$ can be modified to output bits for each bin b indicating at which iterations i noise was added to bin b. The privacy guarantee (Theorem 18) and its proof would still be exactly the same, but by outputting such information, a data analyst would know exactly what noise distributions were added to the true count of each bin.

Analyzing the Accuracy/Utility of the Above Algorithm $\mathcal{M}$. Let us now investigate the utility/accuracy of the above algorithm $\mathcal{M}$. We note that $\mathcal{M}$ processes each bin separately and independently, so we can simply analyze the accuracy of a single bin b. Suppose the count of a bin b is exactly k. Let j be the smallest integer such that $k \leq k_j$, and $j = 0$ if no such integer exists. From the proof of Theorem 18, it is not hard to see that with probability at least $1 - \ell e^{-\alpha}$, it holds that at every iteration $i = 1, \ldots, j$, the algorithm $\mathcal{M}$ adds $Lap(1/\epsilon_i)$ noise to bin b. This means that with probability at least $1 - \ell e^{-\alpha}$, $\mathcal{M}$ will add at least $\sum_{i=0}^{j} Lap(1/\epsilon_i)$ noise to bin b.

Let us now try to derive a probabilistic upper bound on the noise added to bin b. Let us investigate whether noise will be added to bin b on a particular iteration i'. We note that for iteration $i = 1, \ldots, i' - 1$, $\mathcal{M}$ adds either $Lap(1/\epsilon_i)$ noise or no noise to bin b, and with probability at least $1 - e^{-\alpha}$, this noise will not decrease the current noisy count by more than α/ϵ_i. Thus, by the union bound, with probability at least $1 - \ell e^{-\alpha}$, the noisy count at iteration i' will be at least $k - (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1})$, and if this number is $> k_{i'} + (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1})$, $\mathcal{M}$ will not add any noise to bin b at iteration i'. Let I be the set of $i' \in \{1, \ldots, \ell\}$ such that this inequality does not hold, i.e., $k - (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1}) \leq k_{i'} + (\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1})$, which is equivalent to $k \leq k_{i'} + 2(\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i'-1})$. Then, with probability at least $1 - \ell e^{-\alpha}$, the noise distributions added to bin b is a subset of $\{i \in I : Lap(1/\epsilon_i)\} \cup \{Lap(1/\epsilon_0)\}$ (recall that $Lap(1/\epsilon_0)$ noise is added to bin b at the beginning by default).

Suppose $j < \ell$. If the k_i's are "well-spaced" and the ϵ_i's are not "too small", then we can show that with probability at least $1 - \ell e^{-\alpha}$, $\mathcal{M}$ will add at most $\sum_{i=0}^{j+1} Lap(1/\epsilon_i)$ noise to bin b. More formally, suppose that for every $1 \leq i \leq \ell - 1$, we have $k_i > k_{i+1} + 2(\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_i)$. Then, by the definition of j above, we have $k > k_i$ for $i = j + 1, \ldots, \ell$, so $k > k_{i+1} + 2(\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_i)$ for $i = j + 1, \ldots, \ell - 1$, which is equivalent to $k > k_i + 2(\alpha/\epsilon_0 + \cdots + \alpha/\epsilon_{i-1})$ for $i = j + 2, \ldots, \ell$. This means that for every $j + 2 \leq i \leq \ell$, we have $i \notin I$, so with probability at least $1 - \ell e^{-\alpha}$, $\mathcal{M}$ will add at most $\sum_{i=0}^{j+1} Lap(1/\epsilon_i)$ noise to bin b, as required. We note that $\sum_{i=0}^{j+1} Lap(1/\epsilon_i)$ noise can be substantially lower than the $Lap(1/\epsilon_\ell)$ noise added by the standard ϵ_ℓ-differentially private algorithm for releasing a histogram.

2.4 Examples of Outlier Private Histogram Algorithms for General $\epsilon(\cdot), \delta(\cdot)$

In this section, we provide some examples of outlier private histogram algorithms for general $\epsilon(\cdot)$ and $\delta(\cdot)$ functions. Let us first provide some intuition for how the outlier private histogram algorithms work. The standard ϵ-differentially private algorithm for releasing a histogram simply adds $Lap(1/\epsilon)$ noise to each bin count independently. By adding $Lap(1/\epsilon)$ noise to each bin, when a data record t is removed from the data set, the output distribution over noisy histograms only changes by at most ϵ (w.r.t. the metric used in differential privacy). To achieve $\epsilon(\cdot)$-outlier privacy, the output distribution over noisy histograms can only change by at most $\epsilon(k)$, where k is the count of t's bin (t is the data record that is removed). Thus, one may try adding $Lap(1/\epsilon(k))$ noise to each bin, where k is the count of the bin. However, this does not work, since the amount of noise added depends on the count k in a way that is too sensitive. In particular, when we remove t from the data set and the count of t's bin decreases from k to $k - 1$, the magnitude of the noise changes from $1/\epsilon(k)$ to $1/\epsilon(k - 1)$, which changes the output distribution over noisy histograms by more than $\epsilon(k)$.

One way to fix this problem is to add noise to the $\epsilon(\cdot)$ function, so that the $1/\epsilon(k)$ and the $1/\epsilon(k-1)$ become noisy and would be "ϵ'-close" for some $\epsilon' > 0$. To allow for a variety of solutions, we will consider using any algorithm $\mathcal{A}$ that approximates $\epsilon(\cdot)$ in a "differentially private" way—that is, $\mathcal{A}(k) \approx \mathcal{A}(k - 1)$ for every $k > 0$. Then, we will add $\approx Lap(1/\mathcal{A}(k_b))$ noise to each bin b, where k_b is the count for bin b. This works as long as the noise magnitude $1/\mathcal{A}(k_b)$ is large enough; the noise magnitude $1/\epsilon(k_b)$ is large enough, but since $\mathcal{A}(k_b)$ only approximates $\epsilon(k_b)$, $\mathcal{A}(k_b)$ might be too large. Thus, we will also require that $\mathcal{A}(k)$ is at most $\epsilon(k)$ with very high probability. Below, instead of adding Laplace noise to each bin, we consider a general algorithm $\mathcal{B}$ that outputs a noisy count, and satisfies $\mathcal{B}(k, \epsilon') \approx_{\epsilon'} \mathcal{B}(k - 1, \epsilon')$ for every $k > 0$ and $\epsilon' \geq 0$, which is the property we need; adding Laplace noise satisfies this property. For generality, we also add a $\delta(\cdot)$ parameter and consider $(\epsilon(\cdot), \delta(\cdot))$-outlier privacy. Let us now describe the required properties for $\mathcal{A}$.

Definition 19 (Differentially private lower bound for $(\epsilon(\cdot), \delta(\cdot))$)**.** Let $\epsilon(\cdot), \delta(\cdot) : \mathbb{N} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$

be functions. An algorithm $\mathcal{A}$ is said to be an $(\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}, \delta'_{\mathcal{A}})$-*differentially private lower bound for* $(\epsilon(\cdot), \delta(\cdot))$ if $\mathcal{A}$ takes an integer $k \geq 0$ as input and satisfies the following properties:

- $\mathcal{A}(k) \approx_{\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}} \mathcal{A}(k-1)$ for every integer $k > 0$.
- For every $k \in \mathbb{N}$, with probability at least $1 - \delta'_{\mathcal{A}}$, $\mathcal{A}(k)$ outputs an $(\epsilon_{total}, \delta_{total})$ satisfying $\epsilon_{\mathcal{A}} \leq \epsilon_{total} \leq \epsilon(k)$ and $\delta_{\mathcal{A}} + \delta'_{\mathcal{A}} \leq \delta_{total} \leq \delta(k)$.

We now describe our outlier private histogram algorithm for general $\epsilon(\cdot)$ and $\delta(\cdot)$ functions.

Example 8 (Outlier Private Histogram Algorithm for General $\epsilon(\cdot), \delta(\cdot)$)**.** Let $\epsilon(\cdot), \delta(\cdot) : \mathbb{N} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ be monotone functions. Let $\mathcal{A}$ be any $(\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}, \delta'_{\mathcal{A}})$-differentially private lower bound for $(\epsilon(\cdot), \delta(\cdot))$, and suppose that $\epsilon(\cdot)$ and $\delta(\cdot)$ are bounded from below by $\epsilon_{\mathcal{A}}$ and $\delta_{\mathcal{A}} + \delta'_{\mathcal{A}}$ respectively, i.e., $\epsilon(k) \geq \epsilon_{\mathcal{A}}$ and $\delta(k) \geq \delta_{\mathcal{A}} + \delta'_{\mathcal{A}}$ for every $k \in \mathbb{N}$. Let $\mathcal{B}$ be any algorithm that satisfies $\mathcal{B}(k, \epsilon', \delta') \approx_{\epsilon', \delta'} \mathcal{B}(k-1, \epsilon', \delta')$ for every integer $k > 0$, every $\epsilon', \delta' \geq 0$.

Let $\mathcal{M}$ be an algorithm that, on input a data set D, computes a histogram from D, and then does the following for each bin b independently: Let k_b be the count for bin b. $\mathcal{M}$ runs $\mathcal{A}(k_b)$ to get its output $(\epsilon_{total}, \delta_{total})$, and then runs $\mathcal{B}(k_b, \epsilon_{total} - \epsilon_{\mathcal{A}}, \delta_{total} - \delta_{\mathcal{A}} - \delta'_{\mathcal{A}})$ and uses its output to replace the count k_b for bin b. After going through all the bins, $\mathcal{M}$ outputs the modified histogram (and the output $(\epsilon_{total}, \delta_{total})$ of $\mathcal{A}(k_b)$ for each bin b, if this is desired).

Theorem 20 (Outlier Private Histogram Algorithm for General $\epsilon(\cdot), \delta(\cdot)$)**.** *The above algorithm* $\mathcal{M}$ *is* $(\epsilon(\cdot), \delta(\cdot))$*-outlier private.*

Proof. Let D be any data set, let $k > 0$, and let t be any k-outlier in D. We need to show that $\mathcal{M}(D) \approx_{\epsilon(k), \delta(k)} \mathcal{M}(D \setminus \{t\})$. We note that t is equivalent to (w.r.t. $\mathcal{M}$) with precisely those records that belong to the same bin as t, so k is an upper bound on the count for t's bin. Since $\epsilon(\cdot)$ and $\delta(\cdot)$ are monotone, we can assume without loss of generality that k is equal to the count for t's bin. Now, consider removing t from the data set D; the count for t's bin decreases by 1, but the counts of the other bins remain the same. Since $\mathcal{M}$ processes each bin separately and independently, it suffices to show that

$$\mathcal{B}(k, \epsilon_{total,k} - \epsilon_{\mathcal{A}}, \delta_{total,k} - \delta_{\mathcal{A}} - \delta'_{\mathcal{A}}) \approx_{\epsilon(k), \delta(k)} \mathcal{B}(k-1, \epsilon_{total,k-1} - \epsilon_{\mathcal{A}}, \delta_{total,k-1} - \delta_{\mathcal{A}} - \delta'_{\mathcal{A}}), \quad (1)$$

where $(\epsilon_{total,k}, \delta_{total,k}) \sim \mathcal{A}(k)$ and $(\epsilon_{total,k-1}, \delta_{total,k-1}) \sim \mathcal{A}(k-1)$. By definition of $\mathcal{A}$, we have $\mathcal{A}(k) \approx_{\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}} \mathcal{A}(k-1)$, so $(\epsilon_{total,k}, \delta_{total,k}) \approx_{\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}} (\epsilon_{total,k-1}, \delta_{total,k-1})$, so

$$\mathcal{B}(k, \epsilon_{total,k} - \epsilon_{\mathcal{A}}, \delta_{total,k} - \delta_{\mathcal{A}} - \delta'_{\mathcal{A}}) \approx_{\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}} \mathcal{B}(k, \epsilon_{total,k-1} - \epsilon_{\mathcal{A}}, \delta_{total,k-1} - \delta_{\mathcal{A}} - \delta'_{\mathcal{A}}). \quad (2)$$

By definition of $\mathcal{B}$, we have $\mathcal{B}(k, \epsilon', \delta') \approx_{\epsilon', \delta'} \mathcal{B}(k-1, \epsilon', \delta')$ for every $\epsilon', \delta' \geq 0$, and by definition of $\mathcal{A}$, with probability at least $1 - \delta'_{\mathcal{A}}$, $\mathcal{A}(k-1)$ outputs an $(\epsilon_{total,k-1}, \delta_{total,k-1})$ satisfying $\epsilon_{\mathcal{A}} \leq \epsilon_{total,k-1} \leq \epsilon(k-1)$ and $\delta_{\mathcal{A}} + \delta'_{\mathcal{A}} \leq \delta_{total,k-1} \leq \delta(k-1)$, so

$$\begin{aligned} &\mathcal{B}(k, \epsilon_{total,k-1} - \epsilon_{\mathcal{A}}, \delta_{total,k-1} - \delta_{\mathcal{A}} - \delta'_{\mathcal{A}}) \\ \approx_{\epsilon(k-1) - \epsilon_{\mathcal{A}}, \delta(k-1) - \delta_{\mathcal{A}}} &\mathcal{B}(k-1, \epsilon_{total,k-1} - \epsilon_{\mathcal{A}}, \delta_{total,k-1} - \delta_{\mathcal{A}} - \delta'_{\mathcal{A}}). \end{aligned} \quad (3)$$

Now, combining (2) and (3) and noting that $\epsilon(k-1) \leq \epsilon(k)$ and $\delta(k-1) \leq \delta(k)$ (since $\epsilon(\cdot)$ and $\delta(\cdot)$ are monotone), we get (1), as required. □

A typical choice for the algorithm $\mathcal{B}$ in the above example is the algorithm that adds Laplace noise: The algorithm $\mathcal{B}$, on input $k \geq 0$ and $\epsilon', \delta' \geq 0$, adds $Lap(1/\epsilon')$ noise to k and then outputs the modified (noisy) k. Let us now give some examples of the algorithm $\mathcal{A}$:

- Adding noise to k and then computing $\epsilon(\cdot)$ on the noisy k: Let $\epsilon_{\mathcal{A}}, \alpha > 0$, and suppose that $\epsilon(\cdot)$ and $\delta(\cdot)$ are bounded from below by $\epsilon_{\mathcal{A}}$ and $e^{-\alpha}/2$, respectively. Let $\mathcal{A}$ be an algorithm that, on input $k \geq 0$, samples $\lambda \sim Lap(1/\epsilon_{\mathcal{A}})$, lets $k' = \max\{\lfloor k + \lambda - \alpha/\epsilon_{\mathcal{A}} \rfloor, 0\}$, and then outputs $(\epsilon(k'), e^{-\alpha}/2)$. Then, $\mathcal{A}$ is an $(\epsilon_{\mathcal{A}}, 0, e^{-\alpha}/2)$-differentially private lower bound for $(\epsilon(\cdot), \delta(\cdot))$.

- Adding noise to $\epsilon(k)$ calibrated to global sensitivity of $\epsilon(\cdot)$: Let $\epsilon_{\mathcal{A}}, \alpha > 0$, and suppose that $\epsilon(\cdot)$ and $\delta(\cdot)$ are bounded from below by $\epsilon_{\mathcal{A}}$ and $e^{-\alpha}/2$, respectively. Let $\Delta(\epsilon) = \sup_{k' \in \mathbb{Z}_{>0}} |\epsilon(k') - \epsilon(k'-1)|$, and suppose that $\Delta(\epsilon) < \infty$. Let $\mathcal{A}$ be an algorithm that, on input $k \geq 0$, samples $\lambda \sim Lap(\Delta(\epsilon)/\epsilon_{\mathcal{A}})$, and then outputs $(\max\{\epsilon(k) + \lambda - \alpha\Delta(\epsilon)/\epsilon_{\mathcal{A}}, \epsilon_{\mathcal{A}}\}, e^{-\alpha}/2)$. Then, $\mathcal{A}$ is an $(\epsilon_{\mathcal{A}}, 0, e^{-\alpha}/2)$-differentially private lower bound for $(\epsilon(\cdot), \delta(\cdot))$.

- Adding noise to $\epsilon(k)$ calibrated to smooth sensitivity of $\epsilon(\cdot)$: Let $\epsilon_{\mathcal{A}}, \alpha > 0$, and suppose that $\epsilon(\cdot)$ and $\delta(\cdot)$ are bounded from below by $\epsilon_{\mathcal{A}}$ and $\delta_{\mathcal{A}} + e^{-\alpha}/2$, respectively. Let $\delta_{\mathcal{A}} \in (0,1)$, and let $0 \leq \beta \leq \frac{\epsilon_{\mathcal{A}}}{2\ln(2/\delta_{\mathcal{A}})}$. Let $S^*_{\epsilon,\beta}(k) = \sup_{k' \in \mathbb{Z}_{>0}}(|\epsilon(k) - \epsilon(k')| \cdot e^{-\beta|k-k'|})$, and suppose that $S^*_{\epsilon,\beta}(k) < \infty$ for every k. Let $\mathcal{A}$ be an algorithm that, on input $k \geq 0$, samples $\lambda \sim Lap(2S^*(k)/\epsilon_{\mathcal{A}})$, and then outputs $(\max\{\epsilon(k) + \lambda - 2\alpha S^*_{\epsilon,\beta}(k)/\epsilon_{\mathcal{A}}, \epsilon_{\mathcal{A}}\}, \delta_{\mathcal{A}} + e^{-\alpha}/2)$. Then, $\mathcal{A}$ is an $(\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}, e^{-\alpha}/2)$-differentially private lower bound for $(\epsilon(\cdot), \delta(\cdot))$ (see [NRS07]).

- Adding noise to the "noise magnitude function" $1/\epsilon(\cdot)$, calibrated to global sensitivity of $1/\epsilon(\cdot)$: Let $\epsilon_{\mathcal{A}}, \alpha > 0$, and suppose that $\epsilon(\cdot)$ and $\delta(\cdot)$ are bounded from below by $\epsilon_{\mathcal{A}}$ and $e^{-\alpha}/2$, respectively. Let $\Delta(1/\epsilon) = \sup_{k' \in \mathbb{Z}_{>0}} |1/\epsilon(k') - 1/\epsilon(k'-1)|$, and suppose that $\Delta(1/\epsilon) < \infty$. Let $\mathcal{A}$ be an algorithm that, on input $k \geq 0$, samples $\lambda \sim Lap(\Delta(1/\epsilon)/\epsilon_{\mathcal{A}})$, and then outputs $\left(\max\left\{\frac{1}{\max\{1/\epsilon(k)+\lambda-\alpha\Delta(1/\epsilon)/\epsilon_{\mathcal{A}},0\}}, \epsilon_{\mathcal{A}}\right\}, e^{-\alpha}/2\right)$. Then, $\mathcal{A}$ is an $(\epsilon_{\mathcal{A}}, 0, e^{-\alpha}/2)$-differentially private lower bound for $(\epsilon(\cdot), \delta(\cdot))$.

- Adding noise to the "noise magnitude function" $1/\epsilon(\cdot)$, calibrated to smooth sensitivity of $1/\epsilon(\cdot)$: Let $\epsilon_{\mathcal{A}}, \alpha > 0$, and suppose that $\epsilon(\cdot)$ and $\delta(\cdot)$ are bounded from below by $\epsilon_{\mathcal{A}}$ and $\delta_{\mathcal{A}} + e^{-\alpha}/2$, respectively. Let $\delta_{\mathcal{A}} \in (0,1)$, and let $0 \leq \beta \leq \frac{\epsilon_{\mathcal{A}}}{2\ln(2/\delta_{\mathcal{A}})}$. Let $S^*_{1/\epsilon,\beta}(k) = \sup_{k' \in \mathbb{Z}_{>0}}(|1/\epsilon(k) - 1/\epsilon(k')| \cdot e^{-\beta|k-k'|})$, and suppose that $S^*_{1/\epsilon,\beta}(k) < \infty$ for every k. Let $\mathcal{A}$ be an algorithm that, on input $k \geq 0$, samples $\lambda \sim Lap(2S^*(k)/\epsilon_{\mathcal{A}})$, and then outputs $\left(\max\left\{\frac{1}{\max\{1/\epsilon(k)+\lambda-2\alpha S^*(k)/\epsilon_{\mathcal{A}},0\}}, \epsilon_{\mathcal{A}}\right\}, \delta_{\mathcal{A}} + e^{-\alpha}/2\right)$. Then, $\mathcal{A}$ is an $(\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}}, e^{-\alpha}/2)$-differentially private lower bound for $(\epsilon(\cdot), \delta(\cdot))$ (see [NRS07]).

In the above example, the algorithm $\mathcal{M}$ can also release the output $(\epsilon_{total}, \delta_{total})$ of $\mathcal{A}(k_b)$ for each bin b. By releasing this extra information, a data analyst would know exactly what noise distribution was added to the true count of each bin.

Analyzing the Accuracy/Utility of the Above Algorithm $\mathcal{M}$. Let us now investigate the utility/accuracy of the above algorithm $\mathcal{M}$. We note that $\mathcal{M}$ processes each bin separately and independently, so we can simply analyze the accuracy of a single bin b. Suppose the count of a bin b is exactly k. For simplicity, we will assume that $\mathcal{B}$ is the algorithm described above that adds Laplace noise. Let us now consider the various algorithms for $\mathcal{A}$ described above. All of the algorithms involve adding Laplace noise to some value that is used in determining the ϵ_{total} outputted by $\mathcal{A}$. By using the cdf of the Laplace distribution, one can obtain a probabilistic upper bound on the amount of noise added, which gives a probabilistic lower bound on ϵ_{total}. Since the algorithm $\mathcal{B}$ adds $Lap(\frac{1}{\epsilon_{total}-\epsilon_A})$ to bin b, we can obtain a probabilistic upper bound on the amount of noise added to bin b. If we apply this analysis to each of the above algorithms for $\mathcal{A}$, we get the following results:

- Adding noise to k and then computing $\epsilon(\cdot)$ on the noisy k: With probability at least $1-e^{-\alpha}$, the amount of noise added to bin b is at most $Lap(1/\epsilon')$, where $\epsilon' = \epsilon(\max\{\lfloor k - 2\alpha/\epsilon_{\mathcal{A}} \rfloor, 0\}) - \epsilon_{\mathcal{A}}$.
- Adding noise to $\epsilon(k)$ calibrated to global sensitivity of $\epsilon(\cdot)$: With probability at least $1-e^{-\alpha}$, the amount of noise added to bin b is at most $Lap(1/\epsilon')$, where $\epsilon' = \max\{\epsilon(k) - 2\alpha\Delta(\epsilon)/\epsilon_{\mathcal{A}} - \epsilon_{\mathcal{A}}, 0\}$.
- Adding noise to $\epsilon(k)$ calibrated to smooth sensitivity of $\epsilon(\cdot)$: With probability at least $1-e^{-\alpha}$, the amount of noise added to bin b is at most $Lap(1/\epsilon')$, where $\epsilon' = \max\{\epsilon(k) - 4\alpha S^*_{\epsilon,\beta}(k)/\epsilon_{\mathcal{A}} - \epsilon_{\mathcal{A}}, 0\}$.
- Adding noise to the "noise magnitude function" $1/\epsilon(\cdot)$, calibrated to global sensitivity of $1/\epsilon(\cdot)$: With probability at least $1-e^{-\alpha}$, the amount of noise added to bin b is at most $Lap(1/\epsilon')$, where $\epsilon' = \max\left\{\frac{1}{\max\{1/\epsilon(k) - 2\alpha\Delta(1/\epsilon)/\epsilon_{\mathcal{A}}, 0\}} - \epsilon_{\mathcal{A}}, 0\right\}$.
- Adding noise to the "noise magnitude function" $1/\epsilon(\cdot)$, calibrated to smooth sensitivity of $1/\epsilon(\cdot)$: With probability at least $1-e^{-\alpha}$, the amount of noise added to bin b is at most $Lap(1/\epsilon')$, where $\epsilon' = \max\left\{\frac{1}{\max\{1/\epsilon(k) - 4\alpha S^*_{\epsilon,\beta}(k)/\epsilon_{\mathcal{A}}, 0\}} - \epsilon_{\mathcal{A}}, 0\right\}$.

We note that the amount of noise added in the above algorithms can be substantially lower than the $Lap(1/\epsilon(1))$ noise added by the standard $\epsilon(1)$-differentially private algorithm for releasing a histogram.

2.5 Comparing the Staircase Algorithm and the Algorithms for General $\epsilon(\cdot), \delta(\cdot)$

Suppose we want to release a histogram while satisfying $(\epsilon(\cdot), \delta)$-outlier privacy for some monotone function $\epsilon(\cdot)$ and some small $\delta > 0$. If $\epsilon(\cdot)$ only takes on a small number of possible values, then $\epsilon(\cdot)$ is a "staircase" (i.e., piecewise constant) function, so we may want to use the staircase outlier private algorithm for releasing a histogram. If $\epsilon(\cdot)$ takes on infinitely many possible values, then the staircase algorithm cannot even be used. If $\epsilon(\cdot)$ takes on a large but finite number of possible values, the staircase algorithm can still be used, but the amount of noise added to each bin may be too large. This is because the staircase algorithm goes through all the "levels of the staircase" starting from the top, each time adding noise if the current noisy count is less than the top boundary of the level. For bins with a low true count, a lot of noise is added.

For $\epsilon(\cdot)$ functions that take on infinitely many or a large number of possible values, one would want to use our outlier private algorithm for a general $\epsilon(\cdot)$. For example, consider the function $\epsilon(k) = k\epsilon_0$ for some small constant $\epsilon_0 > 0$. Such a function has global sensitivity $\Delta(\epsilon(\cdot)) := \sup_{k' \in \mathbb{Z}_{>0}} |\epsilon(k') - \epsilon(k'-1)| = \epsilon_0$, which is small. Thus, we can use our general outlier private histogram algorithm and choose $\mathcal{A}$ to be the algorithm described above that adds noise to $\epsilon(k)$ calibrated to the global sensitivity of $\epsilon(\cdot)$. If $\epsilon(\cdot)$ has high global sensitivity but low "local sensitivity" for most input values, then one can choose $\mathcal{A}$ to be the algorithm described above that adds noise to $\epsilon(k)$ calibrated to the smooth sensitivity (see [NRS07]) of $\epsilon(\cdot)$. Recall that we allow $\epsilon(\cdot)$ to take on the value ∞ (usually for sufficiently high inputs k), meaning that there is no privacy requirement. If $\epsilon(\cdot)$ does take on the value ∞, then both the global sensitivity and the smooth sensitivity of $\epsilon(\cdot)$ would be ∞, which is not allowed. In such cases, we may want to choose $\mathcal{A}$ to be one of the algorithms described above that add noise to the "noise magnitude function" $1/\epsilon(\cdot)$ instead of $\epsilon(\cdot)$. (Recall that we define $1/\infty$ to be equal to 0.) Alternatively, we can choose $\mathcal{A}$ to be the algorithm that adds noise to k and then computes $\epsilon(\cdot)$ on the noisy k.

We note that for our outlier private algorithm for general $\epsilon(\cdot)$, the function $\epsilon(\cdot)$ needs to be bounded from below by some constant $\epsilon_{\mathcal{A}} > 0$. This is because running the algorithm $\mathcal{A}$ results in "$\epsilon_{\mathcal{A}}$-privacy loss". Our staircase algorithm does not have this restriction; the staircase algorithm works even if the lowest level has an ϵ requirement of 0, in which case the staircase algorithm suppresses counts in the lowest level to 0 with very high probability.

3 Simultaneously Achieving Simple Outlier Privacy and Distributional Differential Privacy

In this section, we show that simple outlier privacy implies a certain notion of *distributional differential privacy*, very similar to the one in [BGKS13]. Let us first state the definition of distributional differential privacy w.r.t. a set of distributions over data sets. Let Φ be any set of distributions over data sets.

Definition 21 (Distributional differential privacy w.r.t. Φ)**.** An algorithm $\mathcal{M}$ is said to be (ϵ, δ)-*differentially private w.r.t.* Φ if for every distribution $\phi \in \Phi$ and every $t \in \bigcup Supp(\phi)$, if we let $\mathcal{D} \sim \phi$, then

$$\mathcal{M}(\mathcal{D})|_{t \in \mathcal{D}} \approx_{\epsilon,\delta} \mathcal{M}(\mathcal{D} \setminus \{t\})|_{t \in \mathcal{D}}.$$

The definition in [BGKS13] is slightly weaker than ours, since their definition permits choosing a simulator that is used instead of $\mathcal{M}$ on the right hand side of the $\approx_{\epsilon,\delta}$, but in our definition, the simulator has to be the algorithm $\mathcal{M}$ itself. The set of distributions Φ can represent a set of possible distributions that contains the supposed "true distribution", or Φ can represent a set of possible beliefs an adversary may have about the data set (see [BGKS13] for more information). We will consider a very large and natural class of distributions that even includes relatively "adversarial" beliefs. Let us now describe our class of distributions.

We begin with some necessary terminology and notation. A *population* is a collection of individuals each holding a data record. For simplicity and convenience, we will not distinguish between an individual and the data record the individual holds; thus, an individual is simply a data record, and a population is simply a multiset of data records. Given a population $\mathcal{P}$ and a function $\pi : \mathcal{P} \to [0, 1]$, let $Sam(\mathcal{P}, \pi)$ be the distribution over data sets obtained by sampling each individual t in the population $\mathcal{P}$ with probability $\pi(t)$ independently. We note that for $Sam(\mathcal{P}, \pi)$, two individuals in $\mathcal{P}$ with the same data record will have the same probability of being sampled. However, we can easily modify the data universe X to include personal/unique identifiers so that we can represent an individual by a unique data record in X.

Let $RS(p, p', \ell)$ be the convex hull of the set of all distributions $Sam(\mathcal{P}, \pi)$, where $\mathcal{P}$ is any population, and $\pi : \mathcal{P} \to [0, 1]$ is any function such that $|\{t \in \mathcal{P} : \pi(t) \notin [p, p'] \cup \{0\}\}| \leq \ell$, i.e., for every individual t in $\mathcal{P}$ except for at most ℓ individuals, π assigns to t some probability in $[p, p'] \cup \{0\}$. Such distributions $Sam(\mathcal{P}, \pi)$ represent sampling from the population $\mathcal{P}$ in a very natural way, where most/all individuals are sampled with probability in between p and p' (inclusive) or with probability 0. We allow at most ℓ individuals to be sampled with probability outside this range, to model the fact that an adversary may know whether certain individuals were sampled or not. The set $RS(p, p', \ell)$ includes all such natural ways of sampling from a population, and also captures a large class of possible beliefs an adversary may have about the data set. (In fact, $RS(p, p', \ell)$ is the convex hull of such a large set of distributions.)

Let us now state our theorem that says that simple outlier privacy implies distributional differential privacy w.r.t. $RS(p, p', \ell)$.

Theorem 22. *Let $\mathcal{M}$ be any (k,ϵ)-simple outlier private algorithm with $k \geq 2$, let $0 < p \leq p' < 1$, and let $0 \leq \ell < k-1$. Then, for every $0 < \epsilon_{Sam} \leq \ln 2$, $\mathcal{M}$ is also $(k, \epsilon_{DP}, \delta_{DP})$-distributional differentially private w.r.t. $RS(p,p',\ell)$, where*

$$\epsilon_{DP} = \max\left\{\frac{\epsilon}{k}, \ln\left(\frac{p'}{p}\frac{1-p}{1-p'}\right) + \epsilon_{Sam}\right\} \text{ and}$$
$$\delta_{DP} = \max\left\{\frac{1}{p}, \frac{1}{1-p'}\right\} e^{-\Omega((k-\ell)\cdot(1-p')^2\cdot{\epsilon_{Sam}}^2)}.$$

Remark. In Theorem 22, it suffices for $\mathcal{M}$ to be (k,ϵ,ϵ')-*simple outlier private*, which is the same as (k,ϵ)-simple outlier private except that the notion of equivalence is replaced by the notion of ϵ'-blends. The proof would be almost exactly the same, but the ϵ_{DP} parameter we achieve would be $\epsilon_{DP} = \max\left\{\frac{\epsilon}{k}, \ln\left(\frac{p'}{p}\frac{1-p}{1-p'}\right) + \epsilon_{Sam} + \epsilon'\right\}$ instead (the δ_{DP} parameter remains the same). The reason we start off with a (k,ϵ)-simple outlier private algorithm is that, as motivated in the introduction, we want an algorithm that satisfies both (k,ϵ)-simple outlier privacy and some notion of (distributional) differential privacy.

Before we prove Theorem 22, let us make some remarks. Our result (Theorem 22) is somewhat similar to the result in [GHLP12] that states that if one combines a crowd-blending private algorithm with a natural pre-sampling step, the combined algorithm is zero-knowledge private (which implies differential privacy) if we view the population as the input data set to the combined algorithm. In contrast, our result achieves a distributional notion of differential privacy on the data set as opposed to the population, which is a different model and definition. For example, one difference is that in distributional differential privacy, the individual t whose privacy we need to protect is guaranteed to be sampled, but in the model of [GHLP12], the individual t in the population might not even be sampled at all, in which case t's privacy is already protected. This leads to differences in the privacy parameters we can achieve.

Our result also has some similarities to a result in [BGKS13], where it is shown that a histogram algorithm that suppresses small counts achieves a notion of distributional differential privacy (described above), but for a class of distributions incomparable to the class we consider (the classes are somewhat similar, but neither is a subset of the other). However, our class of distributions includes distributions/beliefs based on biased and imperfect sampling in a setting where the adversary may even know whether certain individuals were sampled or not; the class of distributions considered in [BGKS13] does not consider such an adversarial setting. Also, we consider the class of simple outlier private algorithms, which includes but is more general than just histogram algorithms that suppress small counts.

Let us now prove Theorem 22. We begin by stating a lemma about the smoothness of the Poisson binomial distribution[2] near its expectation, which has appeared in [GHLP12], and will be used later in the proof of Lemma 24.

Lemma 23 (Smoothness of the Poisson binomial distribution near its expectation)**.** *Let $\mathcal{P}$ be any population, $0 < p \leq p' < 1$, $\pi : \mathcal{P} \to [0,1]$ be any function, and $\epsilon_{Sam} > 0$. Let A be any non-empty (multi)subset of $\mathcal{P}$ such that $\pi(a) \in [p,p']$ for every $a \in A$. Let $\widetilde{D} = Sam(\mathcal{P},\pi)$, $\widetilde{m} = |\widetilde{D} \cap A|$, $n = |A|$, and $\bar{p} = \frac{1}{n}\sum_{a\in A}\pi(a)$. Then, for every integer $m \in \{0,\ldots,n-1\}$, we have the following:*

- *If $m+1 \leq (n+1)\bar{p}\cdot\frac{e^{\epsilon_{Sam}}}{\bar{p}e^{\epsilon_{Sam}}+(1-\bar{p})}$, then $\Pr[\widetilde{m}=m] \leq \frac{p'}{p}\frac{1-p}{1-p'}e^{\epsilon_{Sam}}\Pr[\widetilde{m}=m+1]$.*

[2]The Poisson binomial distribution is the distribution of the sum of independent Bernoulli random variables, where the success probabilities in the Bernoulli random variables are not necessarily the same.

- *If* $m+1 \geq (n+1)\bar{p} \cdot \frac{1}{\bar{p}+(1-\bar{p})e^{\epsilon_{Sam}}}$, *then* $\Pr[\widetilde{m}=m] \geq \frac{p}{p'}\frac{1-p'}{1-p}e^{-\epsilon_{Sam}} \Pr[\widetilde{m}=m+1]$.

The proof of Lemma 23 can be found in the full version of [GHLP12]. We now prove a lemma that roughly says that if an individual is $\mathcal{M}$-equivalent to many people in the population, then the individual's privacy is protected.

Lemma 24. *Let* $\mathcal{M}$ *be any algorithm,* $\mathcal{P}$ *be any population,* $0 < p \leq p' < 1$, *and* $\pi : \mathcal{P} \to [0,1]$ *be any function. Let* $t \in \mathcal{P}$, *and let* $A \subseteq \mathcal{P} \setminus \{t\}$ *such that* $A \neq \emptyset$ *and for every* $t' \in A$, $t' \equiv_{\mathcal{M}} t$ *and* $\pi(t') \in [p,p']$. *Let* $n = |A|$ *and* $\bar{p} = \frac{1}{n}\sum_{t' \in A} \pi(t')$. *Then, for every* $0 < \epsilon_{Sam} \leq \ln 2$, *we have*

$$\mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi) \uplus \{t\}) \approx_{\epsilon_{total}, \delta_{total}} \mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi)),$$

where $\epsilon_{total} = \ln\left(\frac{p'}{p}\frac{1-p}{1-p'}\right) + \epsilon_{Sam}$ *and* $\delta_{total} = \max\left\{\frac{1}{\bar{p}}, \frac{1}{1-\bar{p}}\right\} \cdot e^{-\Omega\left((n+1)\bar{p}\cdot(1-\bar{p})^2 \cdot {\epsilon_{Sam}}^2\right)}$.

Proof. Let $0 < \epsilon_{Sam} \leq \ln 2$, $\widetilde{D} = Sam(\mathcal{P} \setminus \{t\}, \pi)$, $\widetilde{m} = |\widetilde{D} \cap A|$, and $Y \subseteq Range(\mathcal{M})$. We first show that for every $m \in \{0, \ldots, n-1\}$, we have

$$\mathcal{M}(\widetilde{D} \uplus \{t\})|_{\widetilde{m}=m} = \mathcal{M}(\widetilde{D})|_{\widetilde{m}=m+1}. \tag{1}$$

It is known that there exists a "draw-by-draw" selection procedure for drawing samples from A (one at a time) such that right after drawing the j^{th} sample, the samples chosen so far has the same distribution as $Sam(A,\pi)|_{|Sam(A,\pi)|=j}$ (e.g., see Section 3 in [CDL94]). More formally, there exists a vector of random variables $(X_1, \ldots, X_n)$ jointly distributed over A^n such that for every $j \in [n]$, $\{X_1, \ldots, X_j\}$ has the same distribution as $Sam(A,\pi)|_{|Sam(A,\pi)|=j}$. Now, fix $m \in \{0, \ldots, n-1\}$. Then, we have $(\widetilde{D} \uplus \{t\})|_{\widetilde{m}=m} = Sam(\mathcal{P} \setminus (A \uplus \{t\}), \pi) \uplus \{X_1, \ldots, X_m\} \uplus \{t\}$ and $\widetilde{D}|_{\widetilde{m}=m+1} = Sam(\mathcal{P} \setminus (A \uplus \{t\}), \pi) \uplus \{X_1, \ldots, X_m\} \uplus \{X_{m+1}\}$. The condition (1) then follows from the fact that $t \equiv_{\mathcal{M}} t'$ for every individual $t' \in A$, and $Supp(X_{m+1}) \subseteq A$. Thus, we have shown (1).

Let $\alpha = \frac{e^{\epsilon_{Sam}}}{\bar{p}e^{\epsilon_{Sam}}+(1-\bar{p})}$ and $\beta = \frac{1}{\bar{p}+(1-\bar{p})e^{\epsilon_{Sam}}}$. Let $\epsilon_{total} = \ln(\frac{p'}{p}\frac{1-p}{1-p'}) + \epsilon_{Sam}$, and let $\delta_{total} = \max\{\Pr[\widetilde{m}+1 > (n+1)\bar{p}\cdot\alpha], \Pr[\widetilde{m} < (n+1)\bar{p}\cdot\beta]\}$. By Lemma 23 and (1) (and the fact that $m = n$ does not satisfy $m+1 \leq (n+1)\bar{p}\cdot\alpha$), we have

$$\begin{aligned}
&\Pr[\mathcal{M}(\widetilde{D} \uplus \{t\}) \in Y] \\
\leq &\sum_{\substack{m \in \{0,\ldots,n\} \\ m+1 \leq (n+1)\bar{p}\cdot\alpha}} \Pr[\widetilde{m}=m] \cdot \Pr[\mathcal{M}(\widetilde{D} \uplus \{t\}) \in Y \mid \widetilde{m}=m] + \Pr[\widetilde{m}+1 > (n+1)\bar{p}\cdot\alpha] \\
\leq &\sum_{\substack{m \in \{0,\ldots,n\} \\ m+1 \leq (n+1)\bar{p}\cdot\alpha}} \frac{p'}{p}\frac{1-p}{1-p'}e^{\epsilon_{Sam}} \Pr[\widetilde{m}=m+1] \cdot \Pr[\mathcal{M}(\widetilde{D}) \in Y \mid \widetilde{m}=m+1] + \delta_{total} \\
\leq &\, e^{\epsilon_{total}} \Pr[\mathcal{M}(\widetilde{D}) \in Y] + \delta_{total}
\end{aligned} \tag{3}$$

and

$$\begin{aligned}
&\Pr[\mathcal{M}(\widetilde{D} \uplus \{t\}) \in Y] \\
&\geq \sum_{\substack{m \in \{0,\ldots,n-1\} \\ m+1 \geq (n+1)\bar{p}\cdot\beta}} \Pr[\widetilde{m} = m] \cdot \Pr[\mathcal{M}(\widetilde{D} \uplus \{t\}) \in Y \mid \widetilde{m} = m] \\
&\geq \sum_{\substack{m \in \{0,\ldots,n-1\} \\ m+1 \geq (n+1)\bar{p}\cdot\beta}} \frac{p}{p'}\frac{1-p'}{1-p} e^{-\epsilon_{Sam}} \Pr[\widetilde{m} = m+1] \cdot \Pr[\mathcal{M}(\widetilde{D}) \in Y \mid \widetilde{m} = m+1] \\
&\geq \frac{p}{p'}\frac{1-p'}{1-p} e^{-\epsilon_{Sam}} \cdot (\Pr[\mathcal{M}(\widetilde{D}) \in Y] - \Pr[\widetilde{m} < (n+1)\bar{p}\cdot\beta]) \\
&\geq e^{-\epsilon_{total}} \cdot \Pr[\mathcal{M}(\widetilde{D}) \in Y] - \delta_{total}. \qquad (4)
\end{aligned}$$

Thus, we have $\mathcal{M}(\widetilde{D} \uplus \{t\}) \approx_{\epsilon_{total},\delta_{total}} \mathcal{M}(\widetilde{D})$. Now, we observe that

$$\begin{aligned}
&\delta_{total} \\
&= \max\{\Pr[\widetilde{m}+1 > (n+1)\bar{p}\cdot\alpha], \Pr[\widetilde{m} < (n+1)\bar{p}\cdot\beta]\} \\
&\leq \max\left\{\frac{1}{\bar{p}}\Pr[\widetilde{m} + Bin(1,\bar{p}) > (n+1)\bar{p}\cdot\alpha], \frac{1}{1-\bar{p}}\Pr[\widetilde{m} + Bin(1,\bar{p}) < (n+1)\bar{p}\cdot\beta]\right\} \\
&\leq \max\left\{\frac{1}{\bar{p}}\exp\left(-\Omega\left((n+1)\bar{p}\cdot(\alpha-1)^2\right)\right), \frac{1}{1-\bar{p}}\exp\left(-\Omega\left((n+1)\bar{p}\cdot(1-\beta)^2\right)\right)\right\} \\
&\leq \max\left\{\frac{1}{\bar{p}}, \frac{1}{1-\bar{p}}\right\} \cdot \exp\left(-\Omega\left((n+1)\bar{p}\cdot(1-\bar{p})^2{\epsilon_{Sam}}^2\right)\right),
\end{aligned}$$

where $Bin(1,\bar{p})$ is a binomial random variable with 1 trial and success probability $\bar{p}$, and the second last inequality follows from multiplicative Chernoff bounds (and the fact that $\alpha \leq 2$, since $\epsilon_{Sam} \leq \ln 2$). □

We now prove a lemma that roughly says that even if an individual is $\mathcal{M}$-equivalent to only a few people in the population, the individual's privacy is still protected.

Lemma 25. *Let $\mathcal{M}$ be any (k,ϵ)-simple outlier private algorithm with $k \geq 2$, let $\mathcal{P}$ be any population, and let $\pi : \mathcal{P} \to [0,1]$ be any function. Let $t \in \mathcal{P}$, and let $A \subseteq \mathcal{P} \setminus \{t\}$ such that $t' \equiv_{\mathcal{M}} t$ for every $t' \in A$. Let $n = |A|$, $s = |\{t' \in \mathcal{P} \setminus \{t\} : t' \equiv_{\mathcal{M}} t \text{ and } t' \notin A\}|$, and $\bar{p} = \frac{1}{n}\sum_{t' \in A}\pi(t')$. Then, if $s < k-1$, $\bar{p} > 0$, and $n\bar{p} \leq \frac{k-s-1}{2}$, then we have*

$$\mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi) \uplus \{t\}) \approx_{\epsilon/k,\delta_{total}} \mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi)),$$

where $\delta_{total} = e^{-\Omega(k-s)}$.

Proof. Suppose $s < k-1$, $\bar{p} > 0$, and $n\bar{p} \leq \frac{k-s-1}{2}$. Let $\widetilde{D} = Sam(\mathcal{P} \setminus \{t\}, \pi)$ and $\widetilde{m} = |\widetilde{D} \cap A|$. We note that if $\widetilde{m} < k-s-1$, then t is $\mathcal{M}$-equivalent to fewer than k people in $\widetilde{D} \uplus \{t\}$, and since $\mathcal{M}$ is (k,ϵ)-simple outlier private, we have

$$\mathcal{M}(\widetilde{D} \uplus \{t\})|_{\widetilde{m}<k-s-1} \approx_{\epsilon} \mathcal{M}(\widetilde{D})|_{\widetilde{m}<k-s-1}$$

Let $\delta' = \Pr[\widetilde{m} \geq k-s-1]$. Then, we have

$$\mathcal{M}(\widetilde{D} \uplus \{t\}) \approx_{\epsilon,\delta'} \mathcal{M}(\widetilde{D}). \qquad (1)$$

Let $\tau = \frac{k-s-1}{2\bar{p}}$. Then, we have $n \leq \tau$. The lemma now follows from (1) and the inequality

$$\begin{aligned}
\delta' =& \Pr[\widetilde{m} \geq 2\tau\bar{p}] \\
\leq& \Pr[\widetilde{m} + Bin(\lfloor\tau\rfloor - n, \bar{p}) + Bin(1, (\tau - \lfloor\tau\rfloor)\bar{p}) \geq 2\tau\bar{p}] \\
\leq& e^{-\Omega(\tau\bar{p})} \\
\leq& e^{-\Omega(k-s)},
\end{aligned}$$

where $Bin(j, q)$ denotes a binomial random variable with j trials and success probability q, and the second inequality follows from a multiplicative Chernoff bound (note that the expectation of $\widetilde{m} + B(\lfloor\tau\rfloor - n, \bar{p}) + B(1, (\tau - \lfloor\tau\rfloor)\bar{p})$ is $\tau\bar{p}$). □

We will now use the above lemmas to prove Theorem 22.

of Theorem 22. Recall that $RS(p, p', \ell)$ is the convex hull of a set of distributions, which we denote by Φ'. From the definition of distributional differential privacy w.r.t. $RS(p, p', \ell)$, it is easy to see that it suffices to show differential privacy w.r.t. Φ' instead. Let $\phi = Sam(\mathcal{P}, \pi) \in \Phi'$, where $\mathcal{P}$ is the population associated with ϕ, and $\pi : \mathcal{P} \to [0, 1]$ is the sampling probability function associated with ϕ. It is easy to see that without loss of generality, we can assume that $\pi(t') > 0$ for every $t' \in \mathcal{P}$. Let t be any individual in $\mathcal{P}$, and let $\mathcal{D} \sim Sam(\mathcal{P}, \pi)$. We need to show that

$$\mathcal{M}(\mathcal{D})|_{t\in\mathcal{D}} \approx_{\epsilon_{DP},\delta_{DP}} \mathcal{M}(\mathcal{D} \setminus \{t\})|_{t\in\mathcal{D}}.$$

We note that $\mathcal{M}(\mathcal{D})|_{t\in\mathcal{D}} = \mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi) \uplus \{t\})$ and $\mathcal{M}(\mathcal{D} \setminus \{t\})|_{t\in\mathcal{D}} = \mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi))$. Thus, it suffices to show

$$\mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi) \uplus \{t\}) \approx_{\epsilon_{DP},\delta_{DP}} \mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi)). \tag{1}$$

To this end, let $A = \{t' \in \mathcal{P} \setminus \{t\} : t' \equiv_{\mathcal{M}} t \text{ and } \pi(t') \in [p, p']\}$, $n = |A|$, $\bar{p} = \frac{1}{n}\sum_{t'\in A} \pi(t')$, and $s = |\{t' \in \mathcal{P} \setminus \{t\} : t' \equiv_{\mathcal{M}} t \text{ and } t' \notin A\}|$. We note that $s \leq l$, which we use later in some of the inequalities below. Let $\tau = \frac{k-s-1}{2\bar{p}}$. We will consider two cases: $n > \tau$ and $n \leq \tau$.

Suppose $n > \tau$. By Lemma 24, we have

$$\mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi) \uplus \{t\}) \approx_{\epsilon_{DP},\delta_1} \mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi)),$$

where

$$\begin{aligned}
\delta_1 &= \max\left\{\frac{1}{\bar{p}}, \frac{1}{1-\bar{p}}\right\} \cdot e^{-\Omega\left((n+1)\bar{p}\cdot(1-\bar{p})^2\cdot{\epsilon_{Sam}}^2\right)} \\
&\leq \max\left\{\frac{1}{p}, \frac{1}{1-p'}\right\} \cdot e^{-\Omega\left((k-s-1)\cdot(1-p')^2\cdot{\epsilon_{Sam}}^2\right)} \\
&\leq \delta_{DP}.
\end{aligned}$$

Now, suppose $n \leq \tau$. By Lemma 25, we have

$$\mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi) \uplus \{t\}) \approx_{\epsilon_2,\delta_2} \mathcal{M}(Sam(\mathcal{P} \setminus \{t\}, \pi)),$$

where $\epsilon_2 = \epsilon/k \leq \epsilon_{DP}$ and $\delta_2 = e^{-\Omega(k-s)}$, so $\delta_2 \leq \delta_{DP}$.

Thus, we have shown (1), as required. □

References

[BGKS13] Raef Bassily, Adam Groce, Jonathan Katz, and Adam Smith, *Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy.*, FOCS, 2013, pp. 439–448.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar, *Anomaly detection: A survey*, ACM Comput. Surv. **41** (2009), no. 3, 15:1–15:58.

[CDL94] Xiang-Hui Chen, Arthur P. Dempster, and Jun S. Liu, *Weighted finite population sampling to maximize entropy*, Biometrika **81** (1994), no. 3, pp. 457–469.

[DMNS06] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith, *Calibrating noise to sensitivity in private data analysis*, Proc. of the 3rd Theory of Cryptography Conference, 2006, pp. 265–284.

[Dwo06] Cynthia Dwork, *Differential privacy*, ICALP, 2006, pp. 1–12.

[Dwo08] Cynthia Dwork, *Differential privacy: A survey of results*, Theory and Applications of Models of Computation, Lecture Notes in Computer Science, vol. 4978, Springer Berlin / Heidelberg, 2008, pp. 1–19.

[Dwo09] C. Dwork, *The differential privacy frontier*, Proc. of the 6th Theory of Cryptography Conference (TCC), 2009.

[FL12] Lisa K. Fleischer and Yu-Han Lyu, *Approximately optimal auctions for selling privacy when costs are correlated with data*, Proceedings of the 13th ACM Conference on Electronic Commerce, EC '12, ACM, 2012, pp. 568–585.

[GHLP12] Johannes Gehrke, Michael Hay, Edward Lui, and Rafael Pass, *Crowd-blending privacy*, Advances in Cryptology CRYPTO 2012, Lecture Notes in Computer Science, vol. 7417, Springer Berlin Heidelberg, 2012, pp. 479–496.

[GLP11] Johannes Gehrke, Edward Lui, and Rafael Pass, *Towards privacy for social networks: a zero-knowledge based definition of privacy*, Proceedings of the 8th conference on Theory of cryptography, TCC'11, 2011, pp. 432–449.

[GR11] Arpita Ghosh and Aaron Roth, *Selling privacy at auction*, Proceedings of the 12th ACM Conference on Electronic Commerce, EC '11, ACM, 2011, pp. 199–208.

[LR12] Katrina Ligett and Aaron Roth, *Take it or leave it: Running a survey when privacy comes at a cost*, Proceedings of the 8th International Conference on Internet and Network Economics, WINE'12, Springer-Verlag, 2012, pp. 378–391.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith, *Smooth sensitivity and sampling in private data analysis*, STOC 2007, 2007, pp. 75–84.

[NS08] Arvind Narayanan and Vitaly Shmatikov, *Robust de-anonymization of large sparse datasets*, Proceedings of the 2008 IEEE Symposium on Security and Privacy, SP '08, IEEE Computer Society, 2008, pp. 111–125.

[NVX14] Kobbi Nissim, Salil Vadhan, and David Xiao, *Redrawing the boundaries on purchasing data from privacy-sensitive individuals*, Proceedings of the 5th Conference on Innovations in Theoretical Computer Science, ITCS '14, ACM, 2014, pp. 411–422.

[RS12] Aaron Roth and Grant Schoenebeck, *Conducting truthful surveys, cheaply*, Proceedings of the 13th ACM Conference on Electronic Commerce, EC '12, ACM, 2012, pp. 826–843.