NEW MODELS FOR DATA PRIVACY AND VOTING

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Edward Wing Dek Lui

August 2015

NEW MODELS FOR DATA PRIVACY AND VOTING

Edward Wing Dek Lui, Ph.D.

Cornell University 2015

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 data privacy is *differential privacy* [22, 19].

In this thesis, we introduce new definitions of data privacy that can be better than differential privacy in certain ways. We first argue that differential privacy might not be strong enough in social network settings. We then introduce a zero-knowledge based definition of privacy called *zero-knowledge privacy*, which is strictly stronger than differential privacy and is particularly attractive when modeling privacy in social networks.

Both differential privacy and zero-knowledge privacy provide strong privacy guarantees. However, for certain tasks, mechanisms satisfying these privacy definitions have to add a lot of "noise", thus lowering the utility of the released data. Thus, we introduce a new definition of privacy called *crowd-blending privacy* that strictly relaxes the notion of differential privacy. We demonstrate crowd-blending private mechanisms for histograms and for releasing synthetic data points, achieving strictly better utility than what is possible using differentially private mechanisms.

Differential privacy guarantees the *same* level of privacy protection for all individuals. However, we demonstrate that some individuals may need more privacy than others. Thus, we introduce a generalization of differential privacy called *tai-*

lored differential privacy, where an individual's privacy parameter is "tailored" for the individual based on the individual's data and the data set. 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.

In this thesis, we also study the problem of *strategy-proof voting*, which is plagued by impossibility results. We take a bounded-rationality approach to this problem and consider a setting where voters have "*coarse*" beliefs (a notion that has gained popularity in the behavioral economics literature). In particular, we construct good voting rules that satisfy a notion of strategy-proofness with respect to coarse i.i.d. beliefs, thus circumventing the existing impossibility results.

BIOGRAPHICAL SKETCH

Edward Lui received his B.Sc. in Computer Science and Mathematics from the University of British Columbia in 2009. He then went to Cornell University to obtain his Ph.D. in Computer Science.

This document is dedicated to my family.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor Rafael Pass for advising and supporting me during my years here at Cornell University, which allowed me to write this thesis. I also thank my other committee members Johannes Gehrke and Dexter Kozen for taking the time to serve on my committee. I also thank all my collaborators (in alphabetical order by last name) for working with me on our papers: Kai-Min Chung, Johannes Gehrke, Michael Hay, Samantha Leung, Mohammad Mahmoody, and Rafael Pass. Finally, I also thank my family and friends for their support.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vi

1 Introduction **1**
1.1 Data Privacy . . . 1
1.1.1 Zero-Knowledge Privacy . . . 2
1.1.2 Crowd-Blending Privacy . . . 7
1.1.3 Tailored Differential Privacy and Outlier Privacy . . . 11
1.2 Voting . . . 15
1.3 Outline . . . 18

2 Zero-Knowledge Privacy **19**
2.1 Introduction . . . 19
2.1.1 Our Results . . . 19
2.2 Zero-Knowledge Privacy . . . 21
2.2.1 Definitions . . . 21
2.2.2 Differential Privacy vs. Zero-Knowledge Privacy . . . 32
2.2.3 Revisiting the Democrats vs. Republicans Example . . . 34
2.3 Characterizing Zero-Knowledge Privacy . . . 37
2.3.1 Simple Examples of Zero-Knowledge Private Mechanisms . . . 43
2.4 Answering a Class of Queries Simultaneously . . . 49
2.4.1 Sample Complexity of a Class of Fraction Queries . . . 54
2.4.2 Constructing Zero-Knowledge Private Mechanisms for a Class of Fraction Queries . . . 60
2.5 Zero-Knowledge Private Release of Graph Properties . . . 62

3 Crowd-Blending Privacy **70**
3.1 Introduction . . . 70
3.1.1 New Database Mechanisms . . . 72
3.1.2 From Crowd-Blending Privacy to Zero-Knowledge Privacy . . . 73
3.2 Preliminaries and Existing Privacy Definitions . . . 75
3.3 Crowd-Blending Privacy – A New Privacy Definition . . . 78
3.3.1 Examples of Crowd-Blending Private Mechanisms . . . 82
3.3.2 Discussion of Composition . . . 84
3.4 Privately Releasing Synthetic Data Points in $\mathbb{R}^d$ for Computing Smooth Functions . . . 87
3.5 Our Main Theorem . . . 93
3.5.1 Our Main Theorem Extended to Robust Sampling . . . 104

4 Outlier Privacy **118**
4.1 Introduction . . . 118
4.1.1 Our Results . . . 120
4.2 Outlier Privacy . . . 126
4.2.1 Simple Outlier Privacy . . . 129
4.2.2 Simultaneously Achieving Simple Outlier Privacy and Differential Privacy . . . 134
4.2.3 Staircase Outlier Privacy . . . 138
4.2.4 Examples of Outlier Private Histogram Algorithms for General $\epsilon(\cdot), \delta(\cdot)$ . . . 143
4.2.5 Comparing the Staircase Algorithm and the Algorithms for General $\epsilon(\cdot), \delta(\cdot)$ . . . 149
4.3 Simultaneously Achieving Simple Outlier Privacy and Distributional Differential Privacy . . . 151

5 Voting with Coarse Beliefs **160**
5.1 Introduction . . . 160
5.1.1 Our Construction . . . 163
5.1.2 Other Related Work . . . 168
5.2 Preliminaries . . . 169
5.2.1 Strategy-Proofness with respect to a Set of Beliefs . . . 171
5.3 Large-Scale Strategy-Proof Voting w.r.t. Coarse i.i.d. Beliefs . . . 173
5.3.1 Our General Framework . . . 175
5.3.2 Examples of our General Framework . . . 179
5.3.3 Achieving Actual Strategy-Proofness via the Punishing Voting Rule . . . 183

A Appendix for Chapter 2 **187**

B Appendix for Chapter 5 **189**
B.1 Background Information on the Gibbard-Satterthwaite Theorem . . . 189
B.2 Forming a Belief from Observations . . . 191
B.3 Proofs for Section 5.3.1 and 5.3.2 . . . 193
B.4 More Examples of our General Framework . . . 201
B.5 Proofs for Section 5.3.3 . . . 206

Bibliography **211**

CHAPTER 1

INTRODUCTION

1.1 Data Privacy

Data privacy is a fundamental problem in today's information age. Large amounts of data are collected from people by government agencies, search engines, social networking systems, hospitals, financial institutions, and other organizations. There are huge social benefits in analyzing this data. However, it is important to protect the privacy of the people that contributed their data; organizations need to make sure that sensitive information about individuals is not leaked to the people analyzing the data.

Many privacy definitions and schemes for releasing data have been proposed in the past (see [13] and [30] for surveys). However, many of them have been shown to be insufficient due to realistic attacks on such schemes (e.g., see [46]). The notion of *differential privacy* [22, 19], however, has remained strong and resilient to these attacks. Differential privacy requires that when one person's data is added or removed from the database, the output distribution of the database access mechanism changes very little (by at most an ϵ amount, where a specific notion of closeness of distributions is used). Differential privacy has quickly become the standard definition of privacy, and mechanisms for releasing a variety of functions (including histogram queries, principal component analysis, learning, and many more; see [18, 20] for a survey) have been developed.

1.1.1 Zero-Knowledge Privacy

As we shall argue, however, although differential privacy provides a strong privacy guarantee, there are realistic social network settings where these guarantees might not be strong enough. Roughly speaking, differential privacy says that whether you're in the database or not is inconsequential for your privacy (i.e., the output of the database mechanism is essentially the same). But this does not mean your privacy is protected; the information provided by your *friends* might already breach your privacy.

Alternatively, differential privacy can be rephrased as requiring that an adversary does not learn much more about an individual from the mechanism than what she could learn from knowing everyone else in the database (see the appendix of [22] for a formalization of this statement). Such a privacy guarantee is not sufficiently strong in the setting of social networks where an individual's *friends* are strongly correlated with the individual; in essence, "If I know your friends, I know you." (Indeed, a recent study [42] indicates that an individual's sexual orientation can be accurately predicted just by looking at the person's Facebook friends.) We now give a concrete example to illustrate how a differentially private mechanism can violate the privacy of individuals in a social network setting.

Example 1 (Democrats vs. Republicans)**.** Consider a social network of n people that are grouped into cliques of size 200. In each clique, either at least 80% of the people are Democrats, or at least 80% are Republicans. However, assume that the number of Democrats overall is roughly the same as the number of Republicans. Now, consider a mechanism that computes the proportion (in $[0, 1]$) of Democrats in each clique and adds just enough Laplacian noise to satisfy ϵ-differential privacy for a small ϵ, say $\epsilon = 0.1$. For example, to achieve ϵ-differential privacy, it suffices to

add $Lap(\frac{1}{200\epsilon})$ noise[1] to each clique independently, since if a single person changes his or her political preference, the proportion for the person's clique changes by $\frac{1}{200}$ (see Proposition 1 in [22]).

Since the mechanism satisfies ϵ-differential privacy for a small ϵ, one may think that it is safe to release such information without violating the privacy of any particular person. That is, the released data should not allow us to guess correctly with probability significantly greater than $\frac{1}{2}$ whether a particular person is a Democrat or a Republican. However, this is not the case. With $\epsilon = 0.1$, $Lap(\frac{1}{200\epsilon})$ is a small amount of noise, so with high probability, the data released will tell us the main political preference for any particular clique. An adversary that knows which clique a person is in will be able to correctly guess the political preference of that person with probability close to 80%.

For a more detailed explanation and analysis of the above example, see Appendix A.

Remark. In the above example, we assume that the graph structure of the social network is known and that the adversary can identify which clique an individual is in. Such information is commonly available: Graph structures of (anonymized) social networks are often released; these may include a predefined or natural clustering of the people (nodes) into cliques. Furthermore, an adversary may often also figure out the identity of various nodes in the graph (e.g., see [2, 39]); in fact, by participating in the social network before the anonymized graph is published, an adversary can even target specific individuals of his or her choice (see [2]).

Differential privacy says that the output of the mechanism does not depend

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

much on any particular individual's data in the database. Thus, in the above example, a person has little reason not to truthfully report his political preference. However, this does not necessarily imply that the mechanism does not violate the person's privacy. In situations where a social network provides auxiliary information about an individual, that person's privacy can be violated even if he decides to not have his information included.

It is already known that differential privacy may not provide a strong enough privacy guarantee when an adversary has specific auxiliary information about an individual. For example, it was pointed out in [19] that if an adversary knows the auxiliary information "person A is two inches shorter than the average American woman", and if a differentially private mechanism accurately releases the average height of American women, then the adversary learns person A's height (which is assumed to be sensitive information in this example). In this example, the adversary has very specific auxiliary information about an individual that is usually hard to obtain. However, in the Democrats vs. Republicans example, the auxiliary information (the graph and clique structure) about individuals is more general and more easily accessible. Since social network settings contain large amounts of auxiliary information and correlation between individuals, differential privacy is usually not strong enough in such settings.

One may argue that there are versions of differential privacy that protect the privacy of groups of individuals, and that the mechanism in the Democrats vs. Republicans example does not satisfy these stronger definitions of privacy. While this is true, the main point here is that differential privacy will not protect the privacy of an individual, even though the definition is designed for individual privacy. Furthermore, even if we had used a differentially private mechanism that ensures

privacy for groups of size 200 (i.e., the size of each clique), it might still be possible to deduce information about an individual by looking at the *friends of the friends* of the individual; this includes a significantly larger number of individuals.[2]

Towards a Zero-Knowledge Definition of Privacy

In 1977, Dalenius [16] stated a privacy goal for statistical databases: anything about an individual that can be learned from the database can also be learned without access to the database. This would be a very desirable notion of privacy. Unfortunately, Dwork and Naor [19, 23] demonstrated a general impossibility result showing that a formalization of Dalenius's goal along the lines of semantic security for cryptosystems cannot be achieved, assuming that the database gives any non-trivial utility.

Our aim is to provide a privacy definition along the lines of Dalenius, and more precisely, relying on the notion of *zero-knowledge* from cryptography. In this context, the traditional notion of zero-knowledge says that an adversary gains essentially "zero additional knowledge" by accessing the mechanism. More precisely, whatever an adversary can compute by accessing the mechanism can essentially also be computed without accessing the mechanism. A mechanism satisfying this property would be private but utterly useless, since the mechanism provides essentially no information. The whole point of releasing data is to provide utility; thus, this extreme notion of zero-knowledge, which we now call "complete zero-knowledge", is not very applicable in this setting.

Intuitively, we want the mechanism to not release any additional information

[2]The number of "friends of friends" is usually larger than the square of the number of friends (see [64]).

beyond some *"aggregate information"* that is considered acceptable to release. To capture this requirement, we use the notion of a "simulator" from zero-knowledge, and we require that a simulator with the acceptable aggregate information can essentially compute whatever an adversary can compute by accessing the mechanism. Our zero-knowledge privacy definition is thus stated relative to some class of algorithms providing acceptable aggregate information.

Aggregate Information

The question is how to define appropriate classes of aggregate information. We focus on the case where the aggregate information is any information that can be obtained from k random samples/rows (each of which corresponds to one individual's data) of the database, where the data of the person the adversary wants to attack has been concealed. The value of k can be carefully chosen so that the aggregate information obtained does not allow one to infer (much) information about the concealed data. The simulator is given this aggregate information and has to compute what the adversary essentially computes, even though the adversary has access to the mechanism. This ensures that the mechanism does not release any additional information beyond this "k random sample" aggregate information given to the simulator.

Differential privacy can be described using our zero-knowledge privacy definition by considering simulators that are given aggregate information consisting of the data of all but one individual in the database; this is the same as aggregate information consisting of "k random samples" with $k = n$, where n is the number of rows in the database (recall that the data of the individual the adversary wants to attack is concealed), which we formally prove later. For k less than n, such as

$k = \sqrt{n}$, we obtain notions of privacy that are stronger than differential privacy. For example, we later show that the mechanism in the Democrats vs. Republicans example does not satisfy our zero-knowledge privacy definition when $k = o(n)$ and n is sufficiently large.

We may also consider more general models of aggregate information that are specific to graphs representing social networks; in this context we focus on random samples with some exploration of the neighborhood of each sample.

In Chapter 2, we define and investigate *zero-knowledge privacy*, and demonstrate that it can be meaningfully achieved for tasks such as computing averages, fractions, histograms, and a variety of graph parameters and properties, such as average degree and distance to connectivity. Our results are obtained by establishing a connection between zero-knowledge privacy and sample complexity, and by leveraging recent sublinear time algorithms. Chapter 2 is based on the following work: [32].

1.1.2 Crowd-Blending Privacy

Privacy from Random Sampling of Data. Both differential privacy and zero-knowledge privacy provide strong privacy guarantees. However, for certain tasks, mechanisms satisfying these privacy definitions have to add a lot of "noise", thus lowering the utility of the released data. Also, many of these mechanisms run in exponential time (e.g., [25, 6]), so efficiency is also an issue. This leaves open the question of whether there exists a practical approach to sanitizing data, without harming utility too much.

One approach for circumventing the above-mentioned issues is to rely on the

fact that in many cases of interest, the data to be sanitized has been collected via *random sampling* from some underlying population. Intuitively, this initial random sampling already provides some basic privacy guarantees, and may thus help us in decreasing the amount of noise added during sanitization. Indeed, there are several results in the literature indicating that random sampling helps in providing privacy: In [10] the authors quantify the level of the privacy that may be obtained from just random sampling of data (without any further sanitization); in [65] the authors consider a certain type of "sample-and-aggregate" mechanism for achieving differential privacy (but the sampling technique here is more elaborate than just random sampling from a population); a result in [43] shows that random pre-sampling can be used to amplify the privacy level of a differentially private mechanism; finally, in a manuscript [51], the authors demonstrate that a random pre-sampling step applied to a particular mechanism leads to a differentially private mechanism.

In this thesis, we continue the investigation of using random sampling as a means to achieve privacy. In particular, our goal is to provide a *general* definition of privacy that allows us to achieve both differential and zero-knowledge privacy in situations where the data is collected using random sampling from some population. In order to be realistic, we allow the random sampling during data collection to be *biased*, and an adversary may even know whether certain individuals were sampled or not. (Although the mechanisms in the earlier papers rely on random sampling, the random sampling is usually thought of as being part of the sanitization procedure and thus the mechanisms are only analyzed under the assumption that the sampling has been done "ideally".) Additionally, we will require that the privacy notion is meaningful in its own right, also without any pre-sampling; we believe this requirement is crucial for guaranteeing a strong fall-back guarantee

even in case the result of the pre-sampling is leaked (and thus the attacker knows exactly who was sampled).

Towards a Weaker Notion of Privacy

We aim to develop a new privacy definition that allows us to design mechanisms that have greater utility or efficiency than differentially private mechanisms, but still provide a meaningful notion of privacy; furthermore, we want mechanisms satisfying the new definition to achieve differential and zero-knowledge privacy when the underlying data was collected via biased random sampling from some population. To this end, we begin by reconsidering some older notions of privacy.

k-Anonymity and Blending in a Crowd. k-anonymity [75] is a privacy definition specifically for releasing data tables, where a data table is simply a table of records (rows), each of which has values for the attributes (columns) of the table. Roughly speaking, a released data table satisfies *k-anonymity* if every record in the table is the same as $k-1$ other records in the table with respect to certain "identifying" attributes (chosen beforehand). k-anonymity imposes constraints on the syntax of the released data table, but does not consider the way the released data table was computed from the underlying database; this issue has led to several practical attacks against the notion of k-anonymity (e.g., see [78, 80]). k-anonymity can be viewed as being based on the intuition of *"blending in a crowd"*, since the records in the released output are required to "blend" with other records. Intuitively, in many cases, if an individual blends in a crowd of many people in the database, then the individual's privacy is sufficiently protected. However, as demonstrated by known attacks, k-anonymity does not properly capture this in-

tuition as it does not impose any restrictions on the algorithm/mechanism used to generate the released output. Indeed, one of the key insights behind the notion of differential privacy was that privacy should be a property of the sanitization mechanism and not just the output of it.

Relying on this insight, we aim to develop a privacy notion that captures what it means for a mechanism to guarantee that individuals "blend in a crowd". (Another definition partly based on the intuition of blending in a crowd is (c, t)-*isolation* [11], which requires adversaries to be unable to isolate an individual, represented by a data point in $\mathbb{R}^d$, by roughly determining the individual's location in $\mathbb{R}^d$; we formalize the intuition of blending in a crowd in a very different way.)

In Chapter 3, we introduce a new definition of privacy called *crowd-blending privacy* that strictly relaxes the notion of differential privacy. Roughly speaking, k-crowd blending private sanitization of a database requires that each individual i in the database "blends" with k other individuals j in the database, in the sense that the output of the sanitizer is "indistinguishable" if i's data is replaced by j's.

We demonstrate crowd-blending private mechanisms for histograms and for releasing synthetic data points, achieving strictly better utility than what is possible using differentially private mechanisms. Additionally, we demonstrate that if a crowd-blending private mechanism is combined with a "pre-sampling" step, where the individuals in the database are randomly drawn from some underlying population (as is often the case during data collection), then the combined mechanism satisfies not only differential privacy, but also the stronger notion of zero-knowledge privacy. This holds even if the pre-sampling is slightly biased and an adversary knows whether certain individuals were sampled or not. Taken together, our results yield a practical approach for collecting and privately releasing

data while ensuring higher utility than previous approaches. Chapter 3 is based on the following work: [31].

1.1.3 Tailored Differential Privacy and Outlier Privacy

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 [33], has been partly investigated in a mechanism design setting (e.g., see [33, 27, 52, 70, 67]), 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 thesis, 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 thesis, 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 2 (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 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 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 would be 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.

In Chapter 4, we introduce our 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 thesis, 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. Chapter 4 is based on the following work: [53].

1.2 Voting

So far, this thesis has investigated the problem of data privacy. A related problem is how to get people to honestly report their data to begin with. In this thesis, we will focus on the specific problem of voting, which we now discuss.

People have long desired to have a good voting rule that is *strategy-proof*—that is, the voters would not want to lie about their true preferences. Unfortunately, the celebrated Gibbard-Satterthwaite theorem [34, 71] shows that if there are at least three possible candidates, then any deterministic strategy-proof voting rule has to be dictatorial—that is, there exists a fixed voter whose top choice is always the winner. Although the Gibbard-Satterthwaite theorem only applies to *deterministic* voting rules, Gibbard later generalized the Gibbard-Satterthwaite theorem to *randomized* voting rules [35]. In particular, Gibbard showed that any randomized strategy-proof voting rule has to be a probability distribution over *unilateral rules* and *duple rules*, where a unilateral rule depends only on a single voter, and a duple rule chooses only between two possible candidates; furthermore, if the voting rule satisfies the natural condition of *Pareto efficiency*—that is, the voting rule *never* chooses a candidate y that is dominated by some other candidate x by every voter—then the voting rule must be a probability distribution over dictatorial voting rules.

The notion of strategy-proofness, however, is quite strong. It requires voters to truthfully report their preferences, *no matter* what preferences the other voters have (and in particular, even if the voter *knows* exactly the preferences of everyone else). One may thus hope that these impossibility results can be circumvented by relaxing this requirement. For instance, for the case of "large-scale" voting, it makes sense to assume that each voter has some belief about the preferences of the

other voters, and additionally that these preferences are independent and identically distributed (i.i.d.)—we refer to such a notion as *strategy-proofness w.r.t. (with respect to) i.i.d. beliefs*. Unfortunately, this weakening does not make things much better: A result by McLennan [57] shows that if an *anonymous*[3] voting rule (with at least three candidates) is strategy-proof w.r.t. *all* i.i.d. beliefs and is also Pareto efficient, then the voting rule must be a random dictatorship—that is, a uniformly random voter's top choice is chosen as the winner. Furthermore, in [49], Leung, Lui, and Pass strengthen McLennan's result by showing that relaxing Pareto efficiency to ϵ-Pareto efficiency (where Pareto efficiency can be violated with probability ϵ) does not help, even for rather large values of ϵ, and even for a significantly weaker notion of ϵ-Pareto efficiency. Thus, even for an extremely weak notion of what it means to be a "reasonable" voting rule, strategy-proofness w.r.t. all i.i.d. beliefs cannot be achieved.

Can bounded-rationality help? In this thesis, we consider using notions of "bounded-rationality" (see e.g., [74]) to overcome the above impossibility results. An initial approach in this direction was considered by Bartholdi, Tovey, and Trick [3], who suggested that although voting manipulations exist, they may be "hard to find". However, a more recent line of research [45, 15, 69, 28, 29, 79, 17, 41, 59, 58] has demonstrated that instances where manipulation is possible are relatively common and furthermore, successful manipulation can be efficiently computed (if the manipulator has complete knowledge of everyone else's preferences). Nevertheless, it may still be conceivable that such computational approaches may be applicable if we restrict to strategy-proofness w.r.t. i.i.d. beliefs. However, we do not pursue this path here.

[3]A voting rule is anonymous if it does not depend on the identity of the voters.

A different approach suggested by Birrell and Pass [5] relaxes strategy-proofness to *approximate* strategy-proofness, where a voting rule is ϵ-strategy-proof if no voter can gain more than ϵ in expected utility by lying. However, although Birrell and Pass [5] present positive results for the case where $\epsilon = O(1/n)$, they also show that Gibbard's result [35] extends when $\epsilon = o(1/n^2)$. While it may be reasonable to assume that (bounded-rational) voters do not care about "small" differences in expected utility, in some settings a gain of $1/n^2$ may be too much. Additionally, Carroll [7] demonstrates that a variant of McLennan's result [57] holds even if we just consider $o(1/n^{3/2})$-strategy-proofness w.r.t. i.i.d beliefs, as long as we restrict to *deterministic* voting rules.

In Chapter 5, we take a bounded-rationality approach to this problem and consider a setting where voters have "coarse" beliefs. A belief is said to be *α-coarse* if the probabilities in the belief are restricted to lie on a uniform discretization of $[0, 1]$ with "mesh size" at least α. We consider strategy-proofness w.r.t. coarse i.i.d. beliefs, and we focus on "large-scale" voting where the number of voters n is sufficiently large but is still polynomially-related to $1/\alpha$, where α is the coarseness parameter. A voting rule is said to be *large-scale strategy-proof w.r.t. coarse i.i.d. beliefs* if there exists a polynomial $p(\cdot)$ such that for every coarseness parameter $\alpha > 0$, and every $n \geq p(1/\alpha)$, no voter having an α-coarse i.i.d. belief can improve her expected utility by lying about her preferences.

We construct good voting rules that are large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, thus circumventing the above impossibility results. In particular, we construct anonymous ϵ-Pareto efficient voting rules that are large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponentially small in the number of voters. One of our voting rules is a variant of the well-known *instant-runoff* voting

rule, which is used in many elections throughout the world. Chapter 5 is based on the following work: [50].

1.3 Outline

Chapter 2 contains our work on zero-knowledge privacy. Chapter 3 contains our work on crowd-blending privacy. Chapter 4 contains our work on tailored differential privacy and outlier privacy. Finally, Chapter 5 contains our work on voting.

CHAPTER 2

ZERO-KNOWLEDGE PRIVACY

2.1 Introduction

In this chapter, we present our work on zero-knowledge privacy.

2.1.1 Our Results

We consider two different settings for releasing information. In the first setting, we consider statistical (row) databases in a setting where an adversary might have auxiliary information, such as from a social network, and we focus on releasing traditional statistics (e.g., averages, fractions, histograms, etc.) from a database. As explained earlier, differential privacy may not be strong enough in such a setting, so we use our zero-knowledge privacy definition instead. In the second setting, we consider graphs with personal data that represent social networks, and we focus on releasing information directly related to a social network, such as properties of the graph structure.

Setting #1. Computing functions on databases with zero-knowledge privacy: In this setting, we focus on computing functions mapping databases to $\mathbb{R}^m$. We give a characterization of the functions that can be released with zero-knowledge privacy in terms of their *sample complexity*—i.e., how accurate the function can be approximated using random samples from the input database. More precisely, functions with low sample complexity can be computed accurately by a zero-knowledge private mechanism, and vice versa. (It is already known that

functions with low sample complexity can be computed with differential privacy (see [22]), but here we show that the stronger notion of zero-knowledge privacy can be achieved.) In this result, the zero-knowledge private mechanism we construct simply adds Laplacian noise appropriately calibrated to the sample complexity of the function.

Many common queries on statistical databases have low sample complexity, including averages, fraction queries, counting queries, and coarse histogram queries. (In general, it would seem that any "meaningful" query function for statistical databases should have relatively low sample complexity if we think of the rows of the database as random samples from some large underlying population.) We also show that for functions with low sample complexity, we can use differentially private mechanisms to construct zero-knowledge private mechanisms. Using this result, we construct zero-knowledge private mechanisms for such functions while providing decent utility guarantees. All of these results can be found in Section 2.3.

We also consider mechanisms that answer a class of queries simultaneously, and we generalize the notion of sample complexity to classes of query functions. By showing that a class of fraction queries with low VC dimension has low sample complexity, we are able to use existing differentially private mechanisms for classes of fraction queries to construct zero-knowledge private mechanisms, resulting in improved accuracy for fraction queries. These results can be found in Section 2.4.

Setting #2. Releasing graph structure information with zero-knowledge privacy: In this setting, we consider a graph representing a social network, and we focus on privately releasing information about the structure of the graph. We

use our zero-knowledge privacy definition, since the released information can be combined with auxiliary information such as an adversary's knowledge and/or previously released data (e.g., graph structure information) to breach the privacy of individuals.

The connection between sample complexity and zero-knowledge privacy highlights an interesting connection between *sublinear time algorithms* and privacy. As it turns out, many of the recently developed sublinear algorithms on graphs proceed by picking random samples (vertices) and performing some local exploration; we are able to leverage these algorithms to privately release graph structure information, such as average degree and distance to properties such as connectivity and cycle-freeness. We discuss these results in Section 2.5.

2.2 Zero-Knowledge Privacy

2.2.1 Definitions

Let $\mathcal{D}$ be the collection of all databases whose rows are elements (e.g., tuples) from some data universe X. For convenience, we will assume that X contains an element $\perp$, which can be used to conceal the true value of a row. Given a database D, let $|D|$ denote the number of rows in D. For any integer n, let $[n]$ denote the set $\{1, \ldots, n\}$. For any database $D \in \mathcal{D}$, any integer $i \in [|D|]$, and any element $v \in X$, let (D_{-i}, v) denote the database D with row i replaced by the element v.

In this chapter, mechanisms, adversaries, and simulators are simply randomized algorithms that play certain roles in our definitions. Let San be a mechanism that

operates on databases in $\mathcal{D}$. For any database $D \in \mathcal{D}$, any adversary A, and any $z \in \mathcal{B}^*$, let $Out_A(A(z) \leftrightarrow San(D))$ denote the random variable representing the output of A on input z after interacting with the mechanism San operating on the database D. Note that San can be interactive or non-interactive. If San is non-interactive, then $San(D)$ sends information (e.g., a sanitized database) to A and then halts immediately; the adversary A then tries to breach the privacy of some individual in the database D.

Let agg be any class of randomized algorithms that provide aggregate information to simulators, as described in Section 1.1.1. We refer to agg as a *model of aggregate information*.

Definition 1. We say that San is **ϵ-zero-knowledge private with respect to** agg if there exists a $T \in agg$ such that for every adversary A, there exists a simulator S such that for every database $D \in X^n$, every $z \in \mathcal{B}^*$, every integer $i \in [n]$, and every $W \subseteq \mathcal{B}^*$, the following hold:

- $\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W] \leq e^\epsilon \cdot \Pr[S(z, T(D_{-i}, \perp), i, n) \in W]$
- $\Pr[S(z, T(D_{-i}, \perp), i, n) \in W] \leq e^\epsilon \cdot \Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]$

The probabilities are over the random coins of San and A, and T and S, respectively.

Intuitively, the above definition says that whatever an adversary can compute by accessing the mechanism can essentially also be computed without accessing the mechanism but with certain aggregate information (specified by agg). The adversary in the latter scenario is represented by the simulator S. The definition requires that the adversary's output distribution is close to that of the simulator.

This ensures that the mechanism essentially does not release any additional information beyond what is allowed by agg. When the algorithm T provides aggregate information to the simulator S, the data of individual i is concealed so that the aggregate information does not depend directly on individual i's data. However, in the setting of social networks, the aggregate information may still depend on people's data that are correlated with individual i in reality, such as the data of individual i's friends. Thus, the role played by agg is very important in the context of social networks.

To measure the closeness of the adversary's output and the simulator's output, we use the same closeness measure as in differential privacy (as opposed to, say, statistical difference) for the same reasons. As explained in [22], consider a mechanism that outputs the contents of a randomly chosen row. Suppose agg is defined so that it includes the algorithm that simply outputs its input $(D_{-i}, \perp)$ to the simulator (which is the case of differential privacy; see Section 1.1.1 and 2.2.2). Then, a simulator can also choose a random row and then simulate the adversary with the chosen row sent to the simulated adversary. The real adversary's output will be very close to the simulator's output in statistical difference ($1/n$ to be precise); however, it is clear that the mechanism always leaks private information about some individual.

Remark. Our ϵ-zero-knowledge privacy definition can be easily extended to (ϵ, δ)-zero-knowledge privacy, where we also allow an additive error of δ on the RHS of the inequalities. We can further extend our definition to (c, ϵ, δ)-zero-knowledge privacy to protect the privacy of any group of c individuals simultaneously. To obtain this more general definition, we would change "$i \in [n]$" to "$I \subseteq [n]$ with $1 \leq |I| \leq c$", and "$S(z, (D_{-i}, \perp), i, n)$" to "$S(z, (D_{-I}, \vec{\perp}), I, n)$", where $(D_{-I}, \vec{\perp})$ denotes the database D with the rows at positions I replaced by $\perp$. We use this

more general definition when we consider group privacy.

Remark. In our zero-knowledge privacy definition, we consider computationally unbounded simulators. We can also consider PPT simulators by requiring that the mechanism San and the adversary A are PPT algorithms, and agg is a class of PPT algorithms. All of these algorithms would be PPT in n, the size of the database. With minor modifications, the results of this chapter would still hold in this case.

The choice of agg determines the type and amount of aggregate information given to the simulator, and should be decided based on the context in which the zero-knowledge privacy definition is used. The aggregate information should not depend much on data that is highly correlated with the data of a single person, since such aggregate information may be used to breach the privacy of that person. For example, in the context of social networks, such aggregate information should not depend much on any person and the people closely connected to that person, such as his or her friends. By choosing agg carefully, we ensure that the mechanism essentially does not release any additional information beyond what is considered acceptable. We first consider the model of aggregate information where T in the definition of zero-knowledge privacy chooses $k(n)$ random samples. Let $k : \mathbb{N} \to \mathbb{N}$ be any function.

- $RS(k(\cdot)) = k(\cdot)$ random samples: the class of algorithms T such that on input a database $D \in X^n$, T chooses $k(n)$ random samples (rows) from D uniformly without replacement, and then performs any computation on these samples without reading any of the other rows of D. Note that with such samples, T can emulate choosing $k(n)$ random samples with replacement, or a combination of without replacement and with replacement.

$k(n)$ should be carefully chosen so that the aggregate information obtained does not allow one to infer (much) information about the concealed data. For $k(n) = 0$, the simulator is given no aggregate information at all, which is the case of complete zero-knowledge. For $k(n) = n$, the simulator is given all the rows of the original database except for the target individual i, which is the case of differential privacy (as we prove later). For $k(n)$ strictly in between 0 and n, we obtain notions of privacy that are stronger than differential privacy. For example, one can consider $k(n) = o(n)$, such as $k(n) = \sqrt{n}$.

In the setting of a social network, $k(n)$ can be chosen so that when $k(n)$ random samples are chosen from $(D_{-i}, \perp)$, with very high probability, for (almost) all individuals j, very few of the $k(n)$ chosen samples will be in individual j's local neighborhood in the social network graph. This way, the aggregate information released by the mechanism depends very little on data that is highly correlated with the data of a single individual. The choice of $k(n)$ would depend on various properties of the graph structure, such as clustering coefficient, edge density, and degree distribution. The choice of $k(n)$ would also depend on the amount of correlation between the data of adjacent or close vertices (individuals) in the graph, and the type of information released by the mechanism. In this model of aggregate information, vertices (individuals) in the graph with more adjacent vertices (e.g., representing friends) may have less privacy than those with fewer adjacent vertices. However, this is often the case in social networks, where having more links/connections to other people may result in less privacy.

One can also consider other models of aggregate information, such as the class of algorithms T such that on input a database $D \in X^n$, T reads each row with at most a certain probability, say $\frac{k(n)}{n}$. This class of algorithms, which we call "$k(\cdot)$

adaptive samples" and denote by $AS(k(\cdot))$, is more general and contains $RS(k(\cdot))$. However, there are some "bad" mechanisms that are zero-knowledge private with respect to $AS(k(\cdot))$ but intuitively violate the privacy of individuals. We now give an example of such a mechanism.

Example 3. Recall the Democrats vs. Republicans example in the introduction. Now, consider a new mechanism that chooses a clique uniformly at random, computes the proportion of Democrats in the chosen clique, adds $Lap(\frac{1}{200\epsilon})$ noise to the computed proportion, and then outputs the clique number/identifier and the noisy proportion. For the same reasons as in the Democrats vs. Republicans example, this mechanism clearly violates the privacy of the individuals in the chosen clique. However, this mechanism is still ϵ-zero-knowledge private with respect to $AS(k(\cdot))$ as long as $k(n)$ isn't too small.

Intuitively, a simulator with $T \in AS(k(\cdot))$ providing aggregate information can simulate the mechanism by doing the same thing the mechanism does, since the mechanism reads each row with probability $\frac{200}{n}$ (each clique is chosen with probability $\frac{200}{n}$, since there are $\frac{n}{200}$ cliques). This works as long as $\frac{200}{n} \leq \frac{k(n)}{n}$, since T is only allowed to read each row with probability at most $\frac{k(n)}{n}$. We assume that T can easily determine which rows belong to a particular clique; for example, the rows of the database can be ordered so that individuals belonging to the same clique appear consecutively in the database, or the nodes in the published social network graph can have distinct labels in $\{1, \ldots, n\}$, and the political preference for node i is stored in row i of the database.

In Section 2.5, we consider other models of aggregate information that take more into consideration the graph structure of a social network. Note that zero-knowledge privacy does not necessarily guarantee that the privacy of every indi-

vidual is completely protected. Zero-knowledge privacy is defined with respect to a model of aggregate information, and such aggregate information may still leak some sensitive information about an individual in certain scenarios.

Composition: Just as for differentially private mechanisms, mechanisms that are ϵ-zero-knowledge private with respect to $RS(k(\cdot))$ also compose nicely.

Proposition 2. *Suppose San_1 is ϵ_1-zero-knowledge private with respect to $RS(k_1(\cdot))$ and San_2 is ϵ_2-zero-knowledge private with respect to $RS(k_2(\cdot))$. Then, the mechanism San obtained by (sequentially) composing San_1 with San_2 is $(\epsilon_1 + \epsilon_2)$-zero-knowledge private with respect to $RS((k_1 + k_2)(\cdot))$.*

Proof. Let $k(n) = k_1(n) + k_2(n)$, and let $T_1 \in RS(k_1(\cdot))$ and $T_2 \in RS(k_2(\cdot))$ be the aggregate information algorithms guaranteed by the zero-knowledge privacy of San_1 and San_2, respectively. Let T be an algorithm in $RS(k(\cdot))$ that, on input a database $D \in X^n$, chooses $k_1(n)$ random samples as in T_1, chooses $k_2(n)$ random samples as in T_2, runs T_1 and T_2 on D separately using the chosen samples, and then outputs $(T_1(D), T_2(D))$. Let A be any adversary. It is easy to decompose A into two adversaries A_1 and A_2, where A_j is the part of A that interacts with San_j. The output of A_1 contains information describing the state (including the work tape) of A after finishing its interaction with San_1. A_2 expects its input z to be the output of A_1 so that it can start interacting with San_2 with the same information A would have at this point of the interaction. Let S_j be the (guaranteed) simulator for San_j and A_j.

Let S be a simulator that, on input $(z, T(D_{-i}, \perp), i, n) = (z, (T_1(D_{-i}, \perp), T_2(D_{-i}, \perp)), i, n)$, first runs the simulator S_1 on input $(z, T_1(D_{-i}, \perp), i, n)$ to get $z' := S_1(z, T_1(D_{-i}, \perp), i, n)$, and then runs the simulator S_2 on input $(z', T_2(D_{-i}, \perp), i, n)$.

Let $D \in X^n$, $z \in \mathcal{B}^*$, $i \in [n]$, and $W \subseteq \mathcal{B}^*$. Let $Y = Supp(S_1(z, T_1(D_{-i}, \perp), i, n))$. We note that $Y = Supp(Out_{A_1}(A_1(z) \leftrightarrow San_1(D)))$. Now, observe that

$$\begin{aligned}
&\left|\ln\left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]}{\Pr[S(z, T(D_{-i}, \perp), i, n) \in W]}\right)\right| \\
\leq &\left|\ln\left(\frac{\sum_{z' \in Y} \Pr[Out_{A_2}(A_2(z') \leftrightarrow San_2(D)) \in W] \Pr[Out_{A_1}(A_1(z) \leftrightarrow San_1(D)) = z']}{\sum_{z' \in Y} \Pr[S_2(z', T_2(D_{-i}, \perp), i, n) \in W] \Pr[S_1(z, T_1(D_{-i}, \perp), i, n) = z']}\right)\right| \\
\leq &\ \epsilon_1 + \epsilon_2.
\end{aligned}$$

□

Group Privacy: A nice feature of differential privacy is that ϵ-differential privacy implies $(c, c\epsilon)$-differential privacy for groups of size c (see [19] and the appendix in [22]). We have a similar group privacy guarantee for ϵ-zero-knowledge privacy.

Proposition 3. *Suppose San is ϵ-zero-knowledge private with respect to agg. Then, for every $c \geq 1$, San is also $(c, (2c-1)\epsilon)$-zero-knowledge private with respect to agg.*

Proof. Let T be the algorithm in *agg* guaranteed by the ϵ-zero-knowledge privacy of *San*. Let $c \geq 1$. Consider any adversary A, and let S be the simulator for A and *San*. Let $D \in X^n$, $z \in \mathcal{B}^*$, $I \subseteq [n]$ with $1 \leq |I| \leq c$, and $W \subseteq \mathcal{B}^*$. Let i be any integer in I. Then, by the ϵ-zero-knowledge privacy of *San*, we have

$$\left|\ln\left(\frac{\Pr[S(z, T(D_{-I}, \vec{\perp}), i, n) \in W]}{\Pr[Out_A(A(z) \leftrightarrow San(D_{-(I\setminus\{i\})}, \vec{\perp})) \in W]}\right)\right| \leq \epsilon. \tag{1}$$

We later show that ϵ-zero-knowledge privacy implies 2ϵ-differential privacy (Proposition 7), so *San* is 2ϵ-differentially private and thus $(c-1, 2(c-1)\epsilon)$-differentially private. As a result, we have

$$\left|\ln\left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D_{-(I\setminus\{i\})}, \vec{\perp})) \in W]}{\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]}\right)\right| \leq 2(c-1)\epsilon. \tag{2}$$

Combining (1) and (2) from above, we get

$$\left|\ln\left(\frac{\Pr[S(z,T(D_{-I},\vec{\perp}),i,n)\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D))\in W]}\right)\right|\leq(2c-1)\epsilon.$$

□

It can be easily shown that (ϵ,δ)-differential privacy implies $(0,e^\epsilon-1+\delta)$-differential privacy (see [21] or Section 2.2.2 for the definition of (ϵ,δ)-differential privacy), which implies $(c,0,c(e^\epsilon-1+\delta))$-differential privacy for groups of size c. We have a similar group privacy guarantee for (ϵ,δ)-zero-knowledge privacy.

Proposition 4. *Suppose San is (ϵ,δ)-zero-knowledge private with respect to agg. Then, for every $c\geq 1$, San is also $(c,0,(2c-1)(e^\epsilon-1+\delta))$-zero-knowledge private with respect to agg.*

Proof. Let T be the algorithm in *agg* guaranteed by the (ϵ,δ)-zero-knowledge privacy of *San*. Let $c\geq 1$. Consider any adversary A, and let S be the simulator for A and *San*. Then, for every database $D\in X^n$, $z\in\mathcal{B}^*$, $i\in[n]$, and $W\subseteq\mathcal{B}^*$, we have

$$\begin{aligned}&\Pr[Out_A(A(z)\leftrightarrow San(D))\in W]\leq e^\epsilon\cdot\Pr[S(z,T(D_{-i},\perp),i,n)\in W]+\delta\\&\leq\Pr[S(z,T(D_{-i},\perp),i,n)\in W]+(e^\epsilon-1)+\delta,\text{ and}\end{aligned}$$

$$\begin{aligned}&\Pr[S(z,T(D_{-i},\perp),i,n)\in W]\leq e^\epsilon\cdot\Pr[Out_A(A(z)\leftrightarrow San(D))\in W]+\delta\\&\leq\Pr[Out_A(A(z)\leftrightarrow San(D))\in W]+(e^\epsilon-1)+\delta,\text{ so}\end{aligned}$$

$$|\Pr[Out_A(A(z)\leftrightarrow San(D))\in W]-\Pr[S(z,T(D_{-i},\perp),i,n)\in W]|\leq e^\epsilon-1+\delta.$$

Let $D \in X^n$, $z \in \mathcal{B}^*$, $I \subseteq [n]$ with $1 \leq |I| \leq c$, and $W \subseteq \mathcal{B}^*$. Let i be any integer in I. Then, we have

$$|\Pr[S(z, T(D_{-I}, \vec{\perp}), i, n) \in W] - \Pr[Out_A(A(z) \leftrightarrow San(D_{-(I\setminus\{i\})}, \vec{\perp})) \in W]| \leq e^\epsilon - 1 + \delta. \tag{1}$$

Also, for every pair of databases $D', D'' \in X^n$ differing in one row, say row j, we have

$$\begin{aligned}
&|\Pr[Out_A(A(z) \leftrightarrow San(D')) \in W] - \Pr[Out_A(A(z) \leftrightarrow San(D'')) \in W]| \\
&\leq |\Pr[Out_A(A(z) \leftrightarrow San(D')) \in W] - \Pr[S(z, T(D_{-j}, \perp), j, n) \in W]| \\
&\quad + |\Pr[S(z, T(D_{-j}, \perp), j, n) \in W] - \Pr[Out_A(A(z) \leftrightarrow San(D'')) \in W]| \\
&\leq 2(e^\epsilon - 1 + \delta).
\end{aligned}$$

Now, we note that the database $(D_{-(I\setminus\{i\})}, \vec{\perp})$ differs from the database D in at most $c-1$ rows. By considering a sequence of at most c databases where the first database is $(D_{-(I\setminus\{i\})}, \vec{\perp})$, the last database is D, and adjacent databases differ in only one row (and thus are "$2(e^\epsilon - 1 + \delta)$-close" to one another), we have

$$\begin{aligned}
&|\Pr[Out_A(A(z) \leftrightarrow San(D_{-(I\setminus\{i\})}, \vec{\perp})) \in W] - \Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]| \\
&\leq 2(c-1)(e^\epsilon - 1 + \delta).
\end{aligned} \tag{2}$$

Combining (1) and (2) from above yields the result. □

For $agg = RS(k(\cdot))$, we also have the following group privacy guarantee for (ϵ, δ)-zero-knowledge privacy.

Proposition 5. *Suppose San is (ϵ, δ)-zero-knowledge private with respect to $RS(k(\cdot))$. Then, for every $c \geq 1$, San is also $(c, \epsilon, \delta + e^\epsilon(c-1)\frac{k(n)}{n})$-zero-knowledge private with respect to $RS(k(\cdot))$.*

Intuitively, for $k(n)$ sufficiently smaller than n, (ϵ, δ)-zero-knowledge privacy with respect to $RS(k(\cdot))$ actually implies some notion of group privacy, since the algorithm T (in the privacy definition) chooses each row with probability $k(n)/n$. Thus, T chooses any row of a fixed group of c rows with probability at most $ck(n)/n$. If this probability is very small, then the output of T and thus the simulator S does not depend much on any group of c rows.

Proof. Fix $c \geq 1$. Since San is (ϵ, δ)-zero-knowledge private with respect to $RS(k(\cdot))$, there exists a $T \in RS(k(\cdot))$ such that for every adversary A, there exists a simulator S such that for every $D \in X^n$, $z \in \mathcal{B}^*$, $i \in [n]$, and $W \subseteq \mathcal{B}^*$, we have

$$\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W] \leq e^\epsilon \Pr[S(z, T(D_{-i}, \perp), i, n) \in W] + \delta \quad \text{and}$$

$$\Pr[S(z, T(D_{-i}, \perp), i, n) \in W] \leq e^\epsilon \Pr[Out_A(A(z) \leftrightarrow San(D)) \in W] + \delta.$$

Let A be any adversary, and let S be the simulator guaranteed by the zero-knowledge privacy of San. Let S' be a simulator that, on input $(z, T(D_{-I}, \vec{\perp}), I, n)$, outputs $S(z, T(D_{-I}, \vec{\perp}), i, n)$, where i is the smallest integer in I. Let $D \in X^n$, $z \in \mathcal{B}^*$, $I \subseteq [n]$ with $1 \leq |I| \leq c$, and $W \subseteq \mathcal{B}^*$. Let i be the smallest integer in I.

Let E be the event that T reads a row at any of the positions specified by $I \setminus \{i\}$ (the input of T is inferred from context). We note that conditioned on $\overline{E}$, $T(D_{-i}, \perp)$ and $T(D_{-I}, \vec{\perp})$ have the same distribution. Since $T \in RS(k(\cdot))$ and $|I \setminus \{i\}| \leq c - 1$, we have $\Pr[E] \leq (c-1) \cdot \frac{k(n)}{n}$ when T is run on any database

$D' \in X^n$. Now, observe that

$$\begin{aligned}
&\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W] \\
&\leq e^\epsilon \cdot \Pr[S(z, T(D_{-i}, \perp), i, n) \in W] + \delta \\
&= e^\epsilon \cdot (\Pr[S(z, T(D_{-i}, \perp), i, n) \in W \mid \overline{E}] \cdot \Pr[\overline{E}] + \Pr[S(z, T(D_{-i}, \perp), i, n) \in W \mid E] \cdot \Pr[E]) + \delta \\
&\leq e^\epsilon \cdot (\Pr[S'(z, T(D_{-I}, \vec{\perp}), I, n) \in W \mid \overline{E}] \cdot \Pr[\overline{E}] + \Pr[E]) + \delta \\
&\leq e^\epsilon \cdot \left(\Pr[S'(z, T(D_{-I}, \vec{\perp}), I, n) \in W] + (c-1) \cdot \frac{k(n)}{n} \right) + \delta \\
&= e^\epsilon \cdot \Pr[S'(z, T(D_{-I}, \vec{\perp}), I, n) \in W] + e^\epsilon (c-1) \cdot \frac{k(n)}{n} + \delta.
\end{aligned}$$

We also have

$$\begin{aligned}
&\Pr[S'(z, T(D_{-I}, \vec{\perp}), I, n) \in W] \\
&= \Pr[S'(z, T(D_{-I}, \vec{\perp}), I, n) \in W \mid \overline{E}] \cdot \Pr[\overline{E}] + \Pr[S'(z, T(D_{-I}, \vec{\perp}), I, n) \in W \mid E] \cdot \Pr[E] \\
&\leq \Pr[S(z, T(D_{-i}, \perp), i, n) \in W \mid \overline{E}] \cdot \Pr[\overline{E}] + \Pr[E] \\
&\leq \Pr[S(z, T(D_{-i}, \perp), i, n) \in W] + (c-1) \cdot \frac{k(n)}{n} \\
&\leq e^\epsilon \cdot \Pr[Out_A(A(z) \leftrightarrow San(D)) \in W] + \delta + e^\epsilon (c-1) \cdot \frac{k(n)}{n}.
\end{aligned}$$

□

2.2.2 Differential Privacy vs. Zero-Knowledge Privacy

In this section, we compare differential privacy to our zero-knowledge privacy definition. We first state the definition of differential privacy in a form similar to our zero-knowledge privacy definition in order to more easily compare the two. For any pair of databases $D, D' \in X^n$, let $H(D, D')$ denote the number of rows in which D and D' differ, comparing row-wise.

Definition 6. We say that San is **ϵ-differentially private** if for every adversary A, every $z \in \mathcal{B}^*$, every pair of databases $D, D' \in X^n$ with $H(D, D') \leq 1$, and

every $W \subseteq \mathcal{B}^*$, we have

$$\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W] \leq e^{\epsilon} \cdot \Pr[Out_A(A(z) \leftrightarrow San(D')) \in W],$$

where the probabilities are over the random coins of San and A. For (c, ϵ)-**differential privacy** (for groups of size c), the "$H(D, D') \leq 1$" is changed to "$H(D, D') \leq c$". For (ϵ, δ)-**differential privacy**, we allow an additive error of δ on the RHS of the inequality in the definition.

Proposition 7. *Suppose San is ϵ-zero-knowledge private with respect to any class agg. Then, San is 2ϵ-differentially private.*

Proof. Let A be any adversary, let $z \in \mathcal{B}^*$, let $D', D'' \in X^n$ with $H(D', D'') \leq 1$, and let $W \subseteq \mathcal{B}^*$. Since $H(D', D'') \leq 1$, there exists an integer $i \in [n]$ such that $D'_{-i} = D''_{-i}$. Since San is ϵ-zero-knowledge private with respect to agg, there exists a $T \in agg$ and a simulator S such that for every database $D \in X^n$, we have

$$\left| \ln \left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]}{\Pr[S(z, T(D_{-i}, \perp), i, n) \in W]} \right) \right| \leq \epsilon.$$

Now, observe that

$$\begin{aligned}
&\left| \ln \left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D')) \in W]}{\Pr[Out_A(A(z) \leftrightarrow San(D'')) \in W]} \right) \right| \\
\leq &\left| \ln \left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D')) \in W]}{\Pr[S(z, T(D'_{-i}, \perp), i, n) \in W]} \right) \right| + \left| \ln \left(\frac{\Pr[S(z, T(D'_{-i}, \perp), i, n) \in W]}{\Pr[Out_A(A(z) \leftrightarrow San(D'')) \in W]} \right) \right| \\
\leq &\epsilon + \left| \ln \left(\frac{\Pr[S(z, T(D''_{-i}, \perp), i, n) \in W]}{\Pr[Out_A(A(z) \leftrightarrow San(D'')) \in W]} \right) \right| \leq 2\epsilon.
\end{aligned}$$

□

Proposition 8. *Suppose San is ϵ-differentially private. Then, San is ϵ-zero-knowledge private with respect to $RS(n)$.*

Proof. Let T be an algorithm in $RS(n)$ that, on input a database $D' \in X^n$, chooses n "random" samples from D' without replacement (i.e., chooses all the rows of the

database), and then outputs the whole database D'. Let A be any adversary. Let S be the simulator that, on input $(z, (D_{-i}, \perp), i, n)$, simulates the interaction between $A(z)$ and $San(D_{-i}, \perp)$, and outputs whatever A outputs in the simulated interaction. Thus, we have $S(z, T(D_{-i}, \perp), i, n) = Out_A(A(z) \leftrightarrow San(D_{-i}, \perp))$. Let $D \in X^n$, $z \in \mathcal{B}^*$, $i \in [n]$, and $W \subseteq \mathcal{B}^*$. Since San is ϵ-differentially private and $H(D, (D_{-i}, \perp)) \leq 1$, we have

$$\left|\ln\left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]}{\Pr[S(z, T(D_{-i}, \perp), i, n) \in W]}\right)\right| = \left|\ln\left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]}{\Pr[Out_A(A(z) \leftrightarrow San(D_{-i}, \perp)) \in W]}\right)\right|$$
$$\leq \epsilon.$$

□

Remark. If we consider PPT simulators in the definition of zero-knowledge privacy instead of computationally unbounded simulators, then we require San in Proposition 8 to be PPT as well.

Combining Propositions 7 and 8, we see that our zero-knowledge privacy definition includes differential privacy as a special case (up to a factor of 2 for ϵ).

2.2.3 Revisiting the Democrats vs. Republicans Example

Recall the Democrats vs. Republicans example in the introduction. The mechanism in the example is ϵ-differentially private for some small ϵ, even though the privacy of individuals is clearly violated. However, the mechanism is not zero-knowledge private in general. Suppose that the people's political preferences are stored in a database $D \in X^n$.

Proposition 9. *Fix $\epsilon > 0$, $c \geq 1$, and any function $k(\cdot)$ such that $k(n) = o(n)$. Let San be a mechanism that on input $D \in X^n$ computes the proportion of Democrats*

in each clique and adds $Lap(\frac{c}{200\epsilon})$ *noise to each proportion independently. Then,* San *is* (c, ϵ)*-differentially private, but for every constant* $\epsilon' > 0$ *and every sufficiently large* n*,* San *is not* ϵ'*-zero-knowledge private with respect to* $RS(k(\cdot))$*.*

Intuitively, San is not ϵ'-zero-knowledge private with respect to $RS(k(\cdot))$ because for sufficiently large n, an adversary having only $k(n) = o(n)$ random samples would not have any samples in many of the cliques, so the adversary would know nothing about many of the cliques. Therefore, the adversary does gain knowledge by accessing the mechanism, which gives some information about every clique since the amount of noise added to each clique is constant.

Proof. We note that when a single person changes his or her political preference, the vector of proportions of Democrats changes by $\frac{1}{200}$ in L_1 distance. Thus, by Proposition 1 in [22], San is (ϵ/c)-differentially private, which implies that San is (c, ϵ)-differential private (see [19] and the appendix in [22]), as required.

Let $\epsilon' > 0$. Let A be the adversary that simply outputs whatever the mechanism releases. To obtain a contradiction, suppose there exists a $T \in RS(k(\cdot))$ and a simulator S for A satisfying the required condition in the definition of ϵ'-zero-knowledge privacy. Recall that there are 200 people in each clique. Let $\lambda = \frac{c}{200\epsilon}$, let $K \geq 600\epsilon'\lambda$ be a constant such that 200 divides K, and let $n \geq K$ such that 200 divides n. Let $W = (\mathbb{R}_{\leq 0})^{K/200} \times \mathbb{R}^{n/200-K/200}$, and let $z \in \mathcal{B}^*$. Then, for every $D \in X^n$, we have

$$\left| \ln \left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]}{\Pr[S(z, T(D_{-1}, \perp), 1, n) \in W]} \right) \right| \leq \epsilon'.$$

Without loss of generality, suppose that the rows of a database in X^n are ordered so that the first 200 rows correspond to 200 people in the same clique, and

the next 200 rows correspond to 200 people in the same clique, and so on. Let D_1 be the database $(0,\ldots,0)$ of size n, and let D_2 be the database $(1^K,0,\ldots,0)$ of size n, where $1^K=(1,\ldots,1)$ is of size K. Let $X_1,\ldots,X_{n/200}\sim Lap(\lambda)$ (independently), and let $X=(X_1,\ldots,X_{n/200})$. Now, observe the following:

$$\ln\left(\frac{\Pr[Out_A(A(z)\leftrightarrow San(D_1))\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)=\ln\left(\frac{\Pr[(0^{n/200})+X\in W]}{\Pr[(1^{K/200},0,\ldots,0)+X\in W]}\right)$$

$$=\ln\left(\frac{\prod_{j=1}^{K/200}\Pr[X_j\in\mathbb{R}_{\leq 0}]}{\prod_{j=1}^{K/200}\Pr[X_j\in(-\infty,-1]]}\right)=\ln\left(\frac{(\frac{1}{2})^{K/200}}{\prod_{j=1}^{K/200}F_\lambda(-1)}\right)=\ln\left(\frac{(\frac{1}{2})^{K/200}}{\prod_{j=1}^{K/200}(\frac{1}{2}e^{-1/\lambda})}\right)$$

$$=\ln(e^{\frac{K}{200\cdot\lambda}})=\frac{K}{200\cdot\lambda}\geq\frac{600\epsilon'\lambda}{200\cdot\lambda}\geq 3\epsilon', \tag{1}$$

where $F_\lambda(x)=\frac{1}{2}e^{x/\lambda}$ is the cumulative distribution function of $Lap(\lambda)$ for $x\leq 0$. Let E_K denote the event that T does not read any of the first K rows of its input (the input of T is inferred from context). Now, observe that

$$\ln\left(\frac{\Pr[Out_A(A(z)\leftrightarrow San(D_1))\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)$$

$$=\ln\left(\frac{\Pr[Out_A(A(z)\leftrightarrow San(D_1))\in W]}{\Pr[S(z,T((D_1)_{-1},\perp),1,n)\in W]}\right)+\ln\left(\frac{\Pr[S(z,T((D_1)_{-1},\perp),1,n)\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)$$

$$\leq\epsilon'+\ln\left(\frac{\Pr[S(z,T((D_1)_{-1},\perp),1,n)\in W\mid E_K]\Pr[E_K]+\Pr[S(z,T((D_1)_{-1},\perp),1,n)\in W\mid\overline{E_K}]\Pr[\overline{E_K}]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)$$

$$\leq\epsilon'+\ln\left(\frac{\Pr[S(z,T((D_2)_{-1},\perp),1,n)\in W\mid E_K]\cdot\Pr[E_K]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}+\frac{\Pr[\overline{E_K}]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)$$

$$\leq\epsilon'+\ln\left(\frac{\Pr[S(z,T((D_2)_{-1},\perp),1,n)\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}+\frac{\Pr[\overline{E_K}]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)$$

Since $T\in RS(k(\cdot))$ and $k(n)=o(n)$, we have the numerator $\Pr[\overline{E_K}]\to 0$ as $n\to\infty$. However, the denominator $\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]=(\frac{1}{2})^{K/200}e^{-K/(200\lambda)}$ (partly computed earlier) is a constant. Since ln is continuous and $\frac{\Pr[S(z,T((D_2)_{-1},\perp),1,n)\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\in[e^{-\epsilon'},e^{\epsilon'}]$ for all n, we have that for sufficiently large n,

$$\epsilon'+\ln\left(\frac{\Pr[S(z,T((D_2)_{-1},\perp),1,n)\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}+\frac{\Pr[\overline{E_K}]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)$$

$$\leq\epsilon'+\ln\left(\frac{\Pr[S(z,T((D_2)_{-1},\perp),1,n)\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)+\frac{\epsilon'}{2}\ \leq\ \epsilon'+\epsilon'+\frac{\epsilon'}{2}\ \leq\ \frac{5\epsilon'}{2}.$$

Thus, for sufficiently large n, we have $\ln\left(\frac{\Pr[Out_A(A(z)\leftrightarrow San(D_1))\in W]}{\Pr[Out_A(A(z)\leftrightarrow San(D_2))\in W]}\right)\leq\frac{5\epsilon'}{2}$, which contradicts (1) above. □

Remark. In the Democrats vs. Republicans example, even if San adds $Lap(\frac{1}{\epsilon})$ noise to achieve $(200, \epsilon)$-differential privacy so that the privacy of each clique (and thus each person) is protected, the mechanism would still fail to be ϵ'-zero-knowledge private with respect to $RS(k(\cdot))$ for any constant $\epsilon' > 0$ when n is sufficiently large (see Proposition 9). Thus, zero-knowledge privacy with respect to $RS(k(\cdot))$ with $k(n) = o(n)$ seems to provide an unnecessarily strong privacy guarantee in this particular example. However, this is mainly because the clique size is fixed and known to be 200, and we have assumed that the only correlation between people's political preferences that exists is within a clique. In a more realistic social network, there would be cliques of various sizes, and the correlation between people's data would be more complicated. For example, an adversary knowing your friends' friends may still be able to infer a lot of information about you.

2.3 Characterizing Zero-Knowledge Privacy

In this section, we focus on constructing zero-knowledge private mechanisms that compute a function mapping databases in X^n to $\mathbb{R}^m$, and we characterize the set of functions that can be computed with zero-knowledge privacy. These are precisely the functions with low sample complexity, i.e., can be approximated (accurately) using only limited information from the database, such as k random samples.

We quantify the error in approximating a function $g : X^n \to \mathbb{R}^m$ using L_1 distance. Let the L_1-sensitivity of g be defined by $\Delta(g) = \max\{||g(D') - g(D'')||_1 : D', D'' \in X^n \text{ s.t. } H(D', D'') \leq 1\}$. Let $\mathcal{C}$ be any class of randomized algorithms.

Definition 10. A function $g : X^n \to \mathbb{R}^m$ is said to have (δ, β)**-sample complex-**

ity with respect to $\mathcal{C}$ if there exists an algorithm $T \in \mathcal{C}$ such that for every database $D \in X^n$, we have $T(D) \in \mathbb{R}^m$ and

$$\Pr[||T(D) - g(D)||_1 \leq \delta] \geq 1 - \beta.$$

T is said to be a (δ, β)**-sampler for g with respect to $\mathcal{C}$**.

Remark. If we consider PPT simulators in the definition of zero-knowledge privacy instead of computationally unbounded simulators, then we would require here that $\mathcal{C}$ is a class of PPT algorithms (PPT in n, the size of the database). Thus, in the definition of (δ, β)-sample complexity, we would consider a family of functions (one for each value of n) that can be computed in PPT, and the sampler T would be PPT in n.

It was shown in [22] that functions with low sample complexity with respect to $RS(k(\cdot))$ have low sensitivity as well.

Lemma 11 ([22])**.** *Suppose $g : X^n \to \mathbb{R}^m$ has (δ, β)-sample complexity with respect to $RS(k(\cdot))$ for some $\beta < \frac{1-k(n)/n}{2}$. Then, $\Delta(g) \leq 2\delta$.*

As mentioned in [22], the converse of the above lemma is not true, i.e., not all functions with low sensitivity have low sample complexity (see [22] for an example). This should be no surprise, since functions with low sensitivity have accurate differentially private mechanisms, while functions with low sample complexity have accurate zero-knowledge private mechanisms. We already know that zero-knowledge privacy is stronger than differential privacy, as illustrated by the Democrats vs. Republicans example.

We now state how the sample complexity of a function is related to the amount of noise a mechanism needs to add to the function value in order to achieve a certain level of zero-knowledge privacy.

Proposition 12. *Suppose* $g : X^n \to [a,b]^m$ *has* (δ, β)*-sample complexity with respect to some* $\mathcal{C}$*. Then, the mechanism* $San(D) = g(D) + (X_1, \ldots, X_m)$*, where* $X_j \sim Lap(\lambda)$ *for* $j = 1, \ldots, m$ *independently, is* $\ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}})$*-zero-knowledge private with respect to* $\mathcal{C}$*.*

Intuitively, *San* should be zero-knowledge private because a simulator can simulate *San* by first approximating $g(D)$ by running a sampler $T \in \mathcal{C}$ for g, and then adding the same amount of noise as *San*; the error in approximating $g(D)$ is blurred by the added noise so that the simulator's output distribution is close to *San*'s output distribution.

Proof. Let T be a (δ, β)-sampler for g with respect to $\mathcal{C}$. Let A be any adversary. Let S be a simulator that, on input $(z, T(D_{-i}, \perp), i, n)$, first checks whether $T(D_{-i}, \perp)$ is in $[a,b]^m$; if not, S projects $T(D_{-i}, \perp)$ onto the set $[a,b]^m$ (with respect to L_1 distance) so that the accuracy of $T(D_{-i}, \perp)$ is improved and $||g(D) - T(D_{-i}, \perp)||_1 \leq (b-a)m$ always holds, which we use later. From here on, $T(D_{-i}, \perp)$ is treated as a random variable that reflects the possible modification S may perform. The simulator S computes $T(D_{-i}, \perp) + (X_1, \ldots, X_m)$, which we will denote using the random variable $S'(z, T(D_{-i}, \perp), i, n)$. S then simulates the computation of $A(z)$ with $S'(z, T(D_{-i}, \perp), i, n)$ sent to A as a message, and outputs whatever A outputs.

Let $D \in X^n$, $z \in \mathcal{B}^*$, $i \in [n]$. Fix $x \in T(D_{-i}, \perp)$ and $s \in \mathbb{R}^m$. Then, we have

$$\begin{aligned}
&\max\left\{\frac{f_\lambda(s-g(D))}{f_\lambda(s-x)}, \frac{f_\lambda(s-x)}{f_\lambda(s-g(D))}\right\} \\
=\ &\max\left\{e^{(\frac{1}{\lambda}\cdot(||s-x||_1-||s-g(D)||_1))}, e^{(\frac{1}{\lambda}\cdot(||s-g(D)||_1-||s-x||_1))}\right\} \\
\leq\ & e^{(\frac{1}{\lambda}\cdot||g(D)-x||_1)} \leq e^{(\frac{1}{\lambda}\cdot(||g(D)-g(D_{-i},\perp)||_1+||g(D_{-i},\perp)-x||_1))} \leq e^{(\frac{1}{\lambda}\cdot(\Delta(g)+||g(D_{-i},\perp)-x||_1))}.
\end{aligned} \tag{1}$$

Since $||g(D)-x||_1 \leq (b-a)m$ always holds, we also have

$$\max\left\{\frac{f_\lambda(s-g(D))}{f_\lambda(s-x)}, \frac{f_\lambda(s-x)}{f_\lambda(s-g(D))}\right\} \leq e^{(\frac{1}{\lambda}\cdot||g(D)-x||_1)} \leq e^{\frac{(b-a)m}{\lambda}}. \tag{2}$$

Since T is a (δ,β)-sampler for g, we have $\Pr[||g(D_{-i},\perp)-T(D_{-i},\perp)||_1 \leq \delta] \geq 1-\beta$. Thus, using (1) and (2) above, we have

$$\ln\left(\frac{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}{f_\lambda(s-g(D))}\right) \leq \ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}}).$$

Now, using (1) and (2) again, we also have

$$\begin{aligned}
&\ln\left(\frac{f_\lambda(s-g(D))}{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}\right) \\
=\ &-\ln\left(\frac{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}{f_\lambda(s-g(D))}\right) \\
\leq\ &-\ln((1-\beta)e^{-\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{-\frac{(b-a)m}{\lambda}}) = \ln(((1-\beta)e^{-\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{-\frac{(b-a)m}{\lambda}})^{-1}) \\
\leq\ &\ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}}),
\end{aligned}$$

where the last inequality follows from the fact that the function $f(x) = x^{-1}$ is convex for $x > 0$. Then, for every $s \in \mathbb{R}^n$, we have

$$\begin{aligned}
&\left|\ln\left(\frac{\Pr[San(D)=s]}{\Pr[S'(z,T(D_{-i},\perp),i,n)=s]}\right)\right| \\
=\ &\left|\ln\left(\frac{f_\lambda(s-g(D))}{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}\right)\right| \\
\leq\ &\ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}}).
\end{aligned}$$

Thus, for every $W \subseteq \mathcal{B}^*$, we have $\left|\ln\left(\frac{\Pr[Out_A(A(z)\leftrightarrow San(D))\in W]}{\Pr[S(z,T(D_{-i},\perp),i,n)\in W]}\right)\right| \leq \ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}})$. □

Corollary 13. *Suppose $g : X^n \to [a,b]^m$ has (δ,β)-sample complexity with respect to $RS(k(\cdot))$ for some $\beta < \frac{1-k(n)/n}{2}$. Then, the mechanism $San(D) = g(D) + (X_1,\ldots,X_m)$, where $X_j \sim Lap(\lambda)$ for $j = 1,\ldots,m$ independently, is $\ln((1-\beta)e^{\frac{3\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}})$-zero-knowledge private with respect to $RS(k(\cdot))$.*

Proof. This follows from combining Proposition 12 and Lemma 11. □

Using Proposition 12, we can recover the basic mechanism in [22] that is ϵ-differentially private.

Corollary 14. *Let $g : X^n \to [a,b]^m$ and $\epsilon > 0$. A mechanism San for g that adds $Lap(\frac{\Delta(g)}{\epsilon})$ noise to $g(D)$ is ϵ-zero-knowledge private with respect to $RS(n)$.*

Proof. We note that every function $g : X^n \to \mathbb{R}^m$ has $(0,0)$-sample complexity with respect to $RS(n)$. The corollary follows by applying Proposition 12. □

We now show how the zero-knowledge privacy and utility properties of a mechanism computing a function is related to the sample complexity of the function. A class of algorithms agg is said to be *closed under postprocessing* if for any $T \in agg$ and any algorithm M, the composition of M and T (i.e., the algorithm that first runs T on the input and then runs M on the output of T) is also in agg. We note that $RS(k(\cdot))$ is closed under postprocessing.

Proposition 15. *Let agg be any class of algorithms that is closed under postprocessing, and suppose a function $g : X^n \to \mathbb{R}^m$ has a mechanism San such that the following hold:*

- *Utility:* $\Pr[||San(D) - g(D)||_1 \leq \delta] \geq 1 - \beta \quad$ *for every* $D \in X^n$
- *Privacy: San is* ϵ*-zero-knowledge private with respect to agg.*

Then, g has $(\delta, \frac{\beta+(e^\epsilon-1)}{e^\epsilon})$*-sample complexity with respect to agg.*

The intuition is that the zero-knowledge privacy of San guarantees that San can be simulated by a simulator S that is given aggregate information provided by some algorithm $T \in agg$. Thus, an algorithm that runs T and then S will be able to approximate g with accuracy similar to that of San.

Proof. Let A be an adversary that simply outputs whatever San releases. Since San is ϵ-zero-knowledge private with respect to agg, there exists a $B \in agg$ and a simulator S such that for every $D \in X^n$, $z \in \mathcal{B}^*$, and $t \in \mathcal{B}^*$, we have

$$\left|\ln\left(\frac{\Pr[Out_A(A(z) \leftrightarrow San(D)) = t]}{\Pr[S(z, B(D_{-1}, \perp), 1, n) = t]}\right)\right| \leq \epsilon. \tag{2.1}$$

Fix $z \in \mathcal{B}^*$. Let T be an algorithm that, on input $D \in X^n$, first runs B on $(D_{-1}, \perp)$, then runs S on $(z, B(D_{-1}, \perp), 1, n)$, and then outputs $S(z, B(D_{-1}, \perp), 1, n)$. Since $B \in agg$, S is an algorithm, and agg is closed under postprocessing, we have that T is in agg. Let $D \in X^n$. We have

$$\begin{aligned}
&Pr[||T(D) - g(D)||_1 \leq \delta] \;=\; \Pr[||S(z, B(D_{-1}, \perp), 1, n) - g(D)||_1 \leq \delta] \\
&= \sum_{t \in Supp(S(z, B(D_{-1}, \perp), 1, n))} \Pr[||t - g(D)||_1 \leq \delta] \cdot \Pr[S(z, B(D_{-1}, \perp), 1, n) = t] \\
&\geq \sum_{t \in Supp(Out_A(A(z) \leftrightarrow San(D)))} \Pr[||t - g(D)||_1 \leq \delta] \cdot \frac{1}{e^\epsilon} \Pr[Out_A(A(z) \leftrightarrow San(D)) = t] \\
&= \frac{1}{e^\epsilon} \Pr[||Out_A(A(z) \leftrightarrow San(D)) - g(D)||_1 \leq \delta] \;=\; \frac{1}{e^\epsilon} \Pr[||San(D) - g(D)||_1 \leq \delta] \\
&\geq \frac{1}{e^\epsilon}(1 - \beta) \;=\; 1 - \frac{\beta + (e^\epsilon - 1)}{e^\epsilon},
\end{aligned}$$

where the first inequality is due to (2.1). Thus, T is a $(\delta, \frac{\beta+(e^\epsilon-1)}{e^\epsilon})$-sampler for g with respect to agg. □

2.3.1 Simple Examples of Zero-Knowledge Private Mechanisms

In this section, we show how to construct some simple examples of zero-knowledge private mechanisms with respect to $RS(k(\cdot))$.

Example 4 (Averages). Let $n \geq 1$, $k = k(n)$. Let $avg : [0,1]^n \to [0,1]$ be defined by $avg(D) = \frac{\sum_{i=1}^{n} D_i}{n}$, and let $San(D) = avg(D) + Lap(\lambda)$, where $\lambda > 0$. Let T be an algorithm that, on input a database $D \in [0,1]^n$, chooses k random samples from D uniformly, and then outputs the average of the k random samples. By Hoeffding's inequality, we have $\Pr[|T(D) - avg(D)| \leq \delta] \geq 1 - 2e^{-2k\delta^2}$. Thus, avg has $(\delta, 2e^{-2k\delta^2})$-sample complexity with respect to $RS(k(\cdot))$. By Proposition 12, San is $\ln(e^{\frac{1}{\lambda}(\frac{1}{n}+\delta)} + 2e^{\frac{1}{\lambda}-2k\delta^2})$-zero-knowledge private with respect to $RS(k(\cdot))$.

Let $\epsilon \in (0,1]$. We choose $\delta = \frac{1}{k^{1/3}}$ and $\lambda = \frac{1}{\epsilon}(\frac{1}{n} + \delta) = \frac{1}{\epsilon}(\frac{1}{n} + \frac{1}{k^{1/3}})$ so that $\ln(e^{\frac{1}{\lambda}(\frac{1}{n}+\delta)} + 2e^{\frac{1}{\lambda}-2k\delta^2}) = \ln(e^{\epsilon} + 2e^{\frac{\epsilon}{1/n+k^{-1/3}}-2k^{1/3}}) \leq \ln(e^{\epsilon} + 2e^{-k^{1/3}}) \leq \epsilon + 2e^{-k^{1/3}}$. Thus, we have the following result:

- By adding $Lap(\frac{1}{\epsilon}(\frac{1}{n} + \frac{1}{k^{1/3}})) = Lap(O(\frac{1}{\epsilon k^{1/3}}))$ noise to $avg(D)$, San is $(\epsilon + 2e^{-k^{1/3}})$-zero-knowledge private with respect to $RS(k(\cdot))$.

Our example mechanism for computing averages comes from the general connection between sample complexity and zero-knowledge privacy (Proposition 12), which holds for any model agg of aggregate information. For computing averages, we can actually construct a mechanism with better utility by choosing $k(n)$ random samples without replacement from the input database $D \in X^n$ and then running a differentially private mechanism on the chosen samples. We will show that such a mechanism is zero-knowledge private with respect to $RS(k(\cdot))$ and has

even better privacy parameters than the differentially private mechanism, due to the initial sampling step.

In general, this "Sample and DP-Sanitize" method works for query functions that can be approximated using random samples (e.g., averages, fractions, and histograms), and allows us to convert differentially private mechanisms to zero-knowledge private mechanisms with respect to $RS(k(\cdot))$. We now show what privacy guarantees are obtained by the Sample and DP-Sanitize method.

Proposition 16 (Sample and DP-Sanitize)**.** *Let San_{DP} be any (ϵ, δ)-differentially private mechanism. Let San be any mechanism that, on input $D \in X^n$, chooses $k = k(n)$ random samples* without replacement*, runs San_{DP} on the chosen samples, and then performs any computation on the output of San_{DP} without reading the input database D again. Then, the following hold:*

- *San is (ϵ, δ)-zero-knowledge private with respect to $RS(k(\cdot))$.*
- *San is $(2\ln(1 + \frac{k}{n}(e^\epsilon - 1)), (2 + \frac{k}{n}(e^\epsilon - 1))\frac{k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$.*
- *If $\epsilon \leq 1$, then San is $(\frac{4k}{n}\epsilon, \frac{4k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$.*

Intuitively, *San* is zero-knowledge private with respect to $RS(k(\cdot))$ because San_{DP} is differentially private and is only run on k random samples; also, *San* has better privacy parameters than those of San_{DP} because of the extra noise added from choosing only k random samples.

Proof. We observe that the mechanism *San* itself is in $RS(k(\cdot))$. Thus, let $T = San$. Let $n \geq 1$, $D \in X^n$, and $i \in [n]$.

We first show that San is (ϵ, δ)-zero-knowledge private with respect to $RS(k(\cdot))$. Consider $San(D)$ and $T(D_{-i}, \perp) = San(D_{-i}, \perp)$. We note that $San(D)$ and $San(D_{-i}, \perp)$ have the same output distribution when both San's choose the same k random samples and the samples do not contain row i. When both San's choose the same k random samples and the samples do contain row i, the two databases formed by the chosen samples of the two San's (respectively) will differ in exactly one row. Since both San's run an (ϵ, δ)-differentially private mechanism, namely San_{DP}, on the chosen samples, and since San does not use the original input database in its computation afterwards, it is easy to see that the output distribution of the two San's satisfy the closeness condition in the (ϵ, δ)-zero-knowledge privacy definition. Since the simulator S in the privacy definition gets $T(D_{-i}, \perp) = San(D_{-i}, \perp)$ as one of its inputs, it is easy to show that San is (ϵ, δ)-zero-knowledge private with respect to $RS(k(\cdot))$.

We now show that San is $(2\ln(1+\frac{k}{n}(e^{\epsilon}-1)), (2+\frac{k}{n}(e^{\epsilon}-1))\frac{k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$. If $k = n$, then this follows from the fact that San is (ϵ, δ)-zero-knowledge private with respect to $RS(k(\cdot))$. Thus, we now assume that $k \leq n-1$. We will show that $San(D)$ and $T(D_{-i}, \perp) = San(D_{-i}, \perp)$ are "$(2\ln(1+\frac{k}{n}(e^{\epsilon}-1)), (2+\frac{k}{n}(e^{\epsilon}-1))\frac{k}{n}\delta)$"-close. Abusing notation, let $San(D_{-i})$ denote the output of San on input D_{-i} but San chooses $k = k(n)$ random samples (without replacement) instead of $k(|D_{-i}|) = k(n-1)$ random samples. Our strategy is to show that $San(D)$ is close to $San(D_{-i})$ and $San(D_{-i})$ is close to $San(D_{-i}, \perp)$. Let $W \subseteq \{0,1\}^*$, and let E be the event that row i is chosen when

San chooses k random samples. Observe that

$$\begin{aligned}
&\Pr[San(D) \in W] = \Pr[San(D) \in W \mid E]\Pr[E] + \Pr[San(D) \in W \mid \overline{E}]\Pr[\overline{E}] \\
&\leq (e^{\epsilon}\Pr[San(D_{-i}) \in W] + \delta)\Pr[E] + \Pr[San(D_{-i}) \in W](1 - \Pr[E]) \\
&= (1 + \Pr[E](e^{\epsilon} - 1))\Pr[San(D_{-i}) \in W] + \Pr[E]\delta.
\end{aligned}$$

We also have

$$\begin{aligned}
&\Pr[San(D) \in W] = \Pr[San(D) \in W \mid E]\Pr[E] + \Pr[San(D) \in W \mid \overline{E}]\Pr[\overline{E}] \\
&\geq e^{-\epsilon}(\Pr[San(D_{-i}) \in W] - \delta)\Pr[E] + \Pr[San(D_{-i}) \in W](1 - \Pr[E]) \\
&= \Pr[San(D_{-i}) \in W](\Pr[E]e^{-\epsilon} + (1 - \Pr[E])) - e^{-\epsilon}\Pr[E]\delta \\
&\implies \Pr[San(D_{-i}) \in W] \\
&\quad \leq \frac{1}{\Pr[E]e^{-\epsilon} + (1 - \Pr[E])}\Pr[San(D) \in W] + \frac{1}{\Pr[E] + (1 - \Pr[E])e^{\epsilon}}\Pr[E]\delta \\
&\quad \leq (\Pr[E]e^{\epsilon} + (1 - \Pr[E]))\Pr[San(D) \in W] + \Pr[E]\delta \\
&\quad \leq (1 + \Pr[E](e^{\epsilon} - 1))\Pr[San(D) \in W] + \Pr[E]\delta,
\end{aligned}$$

where the second last inequality follows from the fact that the function $f(x) = \frac{1}{x}$ is convex for $x > 0$. Thus, we have the following:

- $\Pr[San(D) \in W] \leq (1 + \Pr[E](e^{\epsilon} - 1))\Pr[San(D_{-i}) \in W] + \Pr[E]\delta$
- $\Pr[San(D_{-i}) \in W] \leq (1 + \Pr[E](e^{\epsilon} - 1))\Pr[San(D) \in W] + \Pr[E]\delta$

Using the same argument as above but with $(D_{-i}, \perp)$ in place of D, we get the following:

- $\Pr[San(D_{-i}, \perp) \in W] \leq (1 + \Pr[E](e^{\epsilon} - 1))\Pr[San(D_{-i}) \in W] + \Pr[E]\delta$
- $\Pr[San(D_{-i}) \in W] \leq (1 + \Pr[E](e^{\epsilon} - 1))\Pr[San(D_{-i}, \perp) \in W] + \Pr[E]\delta$

Combining the results above and noting that $\Pr[E] = \frac{k}{n}$, we have the following:

- $\Pr[San(D) \in W] \leq (1 + \frac{k}{n}(e^\epsilon - 1))^2 \Pr[San(D_{-i}, \perp) \in W] + (2 + \frac{k}{n}(e^\epsilon - 1))\frac{k}{n}\delta$
- $\Pr[San(D_{-i}, \perp) \in W] \leq (1 + \frac{k}{n}(e^\epsilon - 1))^2 \Pr[San(D) \in W] + (2 + \frac{k}{n}(e^\epsilon - 1))\frac{k}{n}\delta$

It easily follows that San is $(2\ln(1 + \frac{k}{n}(e^\epsilon - 1)), (2 + \frac{k}{n}(e^\epsilon - 1))\frac{k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$. Now, suppose that $\epsilon \leq 1$. Then, one can easily verify that $e^\epsilon - 1 \leq 2\epsilon$, so $1 + \frac{k}{n}(e^\epsilon - 1) \leq e^{\frac{2k}{n}\epsilon}$ and $2 + \frac{k}{n}(e^\epsilon - 1) \leq 4$. Thus, San is $(\frac{4k}{n}\epsilon, \frac{4k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$. □

We now use the Sample and DP-Sanitize method (Proposition 16) to construct some zero-knowledge private mechanisms.

In the examples below, let $n \geq 1$, $k = k(n)$, and $\epsilon, \beta \in (0, 1]$. If $D \in X^n$ is a database, let $\widehat{D}$ be a random variable representing a database formed by choosing k random samples from D uniformly without replacement.

Example 5 (Improved Accuracy for Averages). Let $X = [0, 1]$, and let $avg(D) = \frac{\sum_{i=1}^{|D|} D_i}{|D|}$. Let $San(D) = avg(\widehat{D}) + Lap(\frac{1}{\epsilon k})$ for $D \in X^n$. Then, by Proposition 16, San is $\frac{4k}{n}\epsilon$-zero-knowledge private with respect to $RS(k(\cdot))$.

Also, by Hoeffding's inequality (which still holds when the sampling is done without replacement as opposed to with replacement (e.g., see [40])), we have $\Pr[|avg(\widehat{D}) - avg(D)| \geq \alpha] \leq 2e^{-2k\alpha^2}$, and the RHS is $\leq \frac{\beta}{2}$ if $\alpha \geq \frac{1}{\sqrt{k}}\sqrt{\frac{1}{2}\ln(\frac{4}{\beta})}$; thus, we have $\Pr[|avg(\widehat{D}) - avg(D)| \geq \frac{1}{\sqrt{k}}\sqrt{\frac{1}{2}\ln(\frac{4}{\beta})}] \leq \frac{\beta}{2}$. One can also easily verify that for $Y \sim Lap(\frac{1}{\epsilon k})$, we have $\Pr[|Y| \geq \alpha] = e^{-\epsilon k\alpha}$, and the RHS is $\leq \frac{\beta}{2}$ if $\alpha \geq \frac{1}{\epsilon k}\ln(\frac{2}{\beta})$; thus, we have $\Pr[|Y| \geq \frac{1}{\epsilon k}\ln(\frac{2}{\beta})] \leq \frac{\beta}{2}$. Thus, by the union bound, we have the following result:

- For $D \in X^n$, $San(D) = avg(\widehat{D}) + Lap(\frac{1}{\epsilon k})$ approximates $avg(D)$ to within an additive error of $\frac{1}{\sqrt{k}}\sqrt{\frac{1}{2}\ln(\frac{4}{\beta})} + \frac{1}{\epsilon k}\ln(\frac{2}{\beta})$ with probability at least $1-\beta$, and is $\frac{4k}{n}\epsilon$-zero-knowledge private with respect to $RS(k(\cdot))$.

This mechanism is usually more accurate than the mechanism in the earlier example for averages, which adds at least $Lap(\frac{1}{\epsilon k^{1/3}})$ noise and thus is accurate to within an additive error of $\frac{1}{\epsilon k^{1/3}}\ln(\frac{2}{\beta})$ with probability at most $1-\beta$.

Example 6 (**Fraction Queries: Fraction of rows satisfying some property P).** Let $P : X \to \mathcal{B}$ be the predicate representing some property of a row. Let $frac_P(D) = \frac{\sum_{i=1}^{|D|} P(D_i)}{|D|}$, which is the fraction of rows satisfying property P. Since $frac_P(D)$ can be viewed as the average of the numbers $\{P(D_i)\}_{i=1}^n$, we can get the same result as in the example for averages:

- For $D \in X^n$, $San(D) = frac(\widehat{D}) + Lap(\frac{1}{\epsilon k})$ approximates $frac(D)$ to within an additive error of $\frac{1}{\sqrt{k}}\sqrt{\frac{1}{2}\ln(\frac{4}{\beta})} + \frac{1}{\epsilon k}\ln(\frac{2}{\beta})$ with probability at least $1-\beta$, and is $\frac{4k}{n}\epsilon$-zero-knowledge private with respect to $RS(k(\cdot))$.

Example 7 (**Counting Queries: Number of rows satisfying some property P).** Let $P : X \to \mathcal{B}$ be the predicate representing some property of a row. Let $count(D) = \sum_{i=1}^n P(D_i)$, which is the number of rows satisfying property P. Since $g(D)$ is simply a fraction query but scaled by a factor of n, we can get the same result as in the example for fraction queries except that the error is scaled by a factor of n:

- For $D \in X^n$, $San(D) = n \cdot (frac(\widehat{D}) + Lap(\frac{1}{\epsilon k}))$ approximates $count(D) = n \cdot frac_P(D)$ to within an additive error of $\frac{n}{\sqrt{k}}\sqrt{\frac{1}{2}\ln(\frac{4}{\beta})} + \frac{n}{\epsilon k}\ln(\frac{2}{\beta})$ with probability at least $1-\beta$, and is $\frac{4k}{n}\epsilon$-zero-knowledge private with respect to $RS(k(\cdot))$.

Example 8 (Histograms). Let $B_1, \ldots, B_m$ be any partition of X with m blocks. We refer to each B_i as a bin. Let $hist(D) = (b_1, \ldots, b_n)$, where $b_i = |\{j \in [n] : D_j \in B_i\}|$ is the number of rows of D that belong to bin B_i. Given a database D, let $\widetilde{D}_1, \ldots, \widetilde{D}_m$ be independent random variables representing databases formed by choosing $\frac{k}{m}$ random samples from the database D uniformly without replacement.

We can construct a zero-knowledge private mechanism (with respect to $RS(k(\cdot))$) that computes the histogram with respect to the bins $B_1, \ldots, B_m$, by composing San_i for $i = 1, \ldots, m$, where San_i is any zero-knowledge private mechanism (with respect to $RS(\frac{1}{m}k(\cdot))$) for estimating the number of rows in the i^{th} bin, and then applying our composition result (Propsition 2). Using our mechanism for counting queries, we can define San_i so that it approximates the number of rows in the i^{th} bin to within an additive error of $\frac{n\sqrt{m}}{\sqrt{k}}\sqrt{\frac{1}{2}\ln(\frac{4m}{\beta})} + \frac{nm}{\epsilon k}\ln(\frac{2m}{\beta})$ with probability at least $1 - \frac{\beta}{m}$, and is $\frac{4k}{nm}\epsilon$-zero-knowledge private with respect to $RS(\frac{1}{m}k(\cdot))$. Then, applying the union bound and our composition result (Propsition 2), we get the following result:

- For $D \in X^n$, $San(D) = (San_1(\widetilde{D}_1), \ldots, San_m(\widetilde{D}_m))$ approximates $hist(D)$ to within an error (with respect to L_1 distance) of $\frac{nm^{3/2}}{\sqrt{k}}\sqrt{\frac{1}{2}\ln(\frac{4m}{\beta})} + \frac{nm^2}{\epsilon k}\ln(\frac{2m}{\beta})$ with probability at least $1 - \beta$, and is $\frac{4k}{n}\epsilon$-zero-knowledge private with respect to $RS(k(\cdot))$.

2.4 Answering a Class of Queries Simultaneously

In this section, we consider mechanisms that answer a class of query functions simultaneously. We generalize the notion of sample complexity (with respect to $RS(k(\cdot))$) to classes of query functions and show a connection between differen-

tial privacy and zero-knowledge privacy for any class of query functions with low sample complexity. In particular, we show that for any class $\mathcal{Q}$ of query functions that can be approximated simultaneously using random samples, any differentially private mechanism that is "useful" for $\mathcal{Q}$ can be converted to a zero-knowledge private mechanism that is useful for $\mathcal{Q}$, similar to the Sample and DP-Sanitize method. We also show that any class of fraction queries with low VC dimension can be approximated simultaneously using random samples, so we can use existing differentially private mechanisms (e.g., the ones in [6] and [25]) to obtain zero-knowledge private mechanisms for any class of fraction queries with low VC dimension.

Let X^* denote the set of all databases whose rows are elements from the data universe X. In this section, a query is a function from X^* to $\mathbb{R}^m$ for some m.

In this section, we consider mechanisms that answer a class $\mathcal{Q}$ of queries simulataneously by outputting a "synopsis" (e.g., a synthetic database) that allows us to answer all the queries in $\mathcal{Q}$. A *synopsis* is a pair $(\widetilde{D}, R)$, where $\widetilde{D}$ is any data structure (containing data), and R is a description of any deterministic "query-answering" algorithm that, on input a data structure $\widetilde{D}$ and a query $q : X^* \to \mathbb{R}^m$, answers the query by reading $\widetilde{D}$ and outputting some vector in $\mathbb{R}^m$.

Let R_{DB} be the usual query-answering algorithm for databases that, on input a database $D \in X^*$ and a query $q : X^* \to \mathbb{R}^m$, answers with $q(D)$. If D is a database, then (D, R_{DB}) is an example of a synopsis. If $\widehat{D}$ is a database obtained by choosing k random samples from D, then another example of a synopsis is $(\widehat{D}, R_{\widehat{D}})$, where $R_{\widehat{D}}$ is a query-answering algorithm that approximates a given counting query q on the larger database D by computing $q(\widehat{D})$ and then scaling the answer by $\frac{|D|}{k}$ (to compensate for the fact that $\widehat{D}$ contained only k random samples from D).

Let $\mathcal{Q}$ be any class of queries that map databases in X^* to vectors in $\mathbb{R}^m$ for some m. We now define what it means for two synopses to be close to one another with respect to $\mathcal{Q}$.

Definition 17. Two synopses $(\widetilde{D}, R)$ and $(\widetilde{D}', R')$ are said to be α**-close with respect to** $\mathcal{Q}$ if $\sup_{q \in \mathcal{Q}} ||R(\widetilde{D}, q) - R'(\widetilde{D}', q)||_1 \leq \alpha$.

Intuitively, two synopses are α-close to one another with respect to $\mathcal{Q}$ if they are "α-indistinguishable" by $\mathcal{Q}$, i.e., no query in $\mathcal{Q}$ can be used to distinguish the two synopses by more than α. Thus, if two synopses are close to one another with respect to $\mathcal{Q}$, then we can use one synopsis to approximate the other synopsis's answers to queries in $\mathcal{Q}$. We want to construct mechanisms that, on input a database D, outputs a synopsis that is close to the synopsis (D, R_{DB}), so that we can use the synopsis to accurately answer queries in $\mathcal{Q}$ on the database D. We define the usefulness/utility of a mechanism from this perspective of closeness of synopses.

Definition 18. A mechanism San is (α, β)**-useful with respect to** $\mathcal{Q}$ **for databases of size** n if for every input database $D \in X^n$, with probability at least $1 - \beta$ (over the random coins of San), $San(D)$ outputs a synopsis $(\widetilde{D}, R)$ that is α-close to (D, R_{DB}) with respect to $\mathcal{Q}$.

We now generalize the notion of sample complexity with respect to $RS(k(\cdot))$ to classes of query functions. Intuitively, a class $\mathcal{Q}$ of queries has low sample complexity if the queries can be approximated *simultaneously* using k random samples from the input database.

Definition 19. A class $\mathcal{Q}$ of queries is said to have (k, α, β)**-sample complexity for databases of size** n **with converter** $f : \mathcal{Q} \times \mathbb{R}^m \to \mathbb{R}^m$ if for every database

$D \in X^n$, if we choose k random samples from the database D *without replacement* and form a database $\widehat{D}$ consisting of the chosen samples, then with probability at least $1 - \beta$, we have $\sup_{q \in \mathcal{Q}} ||q(D) - f(q, q(\widehat{D}))||_1 \leq \alpha$.

The converter f in the above definition is used to convert the answer to a query q on the database $\widehat{D}$ to an answer to the same query q on the original database D. When $\mathcal{Q}$ is a class of queries computing averages or the fraction of rows satisfying some predicate, f would normally be the function $f(q, \vec{x}) = \vec{x}$. When $\mathcal{Q}$ is a class of queries computing sums, f would normally be the function $f(q, \vec{x}) = \frac{n}{k}\vec{x}$, since the database $\widehat{D}$ consists of only k random samples from the original database D, which has n rows.

We now show that for any class $\mathcal{Q}$ of queries with low sample complexity, we can convert any useful differentially private mechanism to a useful zero-knowledge private mechanism (with respect to $RS(k(\cdot))$).

Proposition 20. *Let $n \geq 1$, $k = k(n)$. Suppose a class $\mathcal{Q}$ of queries has (k, α_1, β_1)-sample complexity for databases of size n with a converter $f : \mathcal{Q} \times X^m \to X^m$ such that $||f(q, x) - f(q, y)||_1 \leq L||x - y||_1$ for every $x, y \in \mathbb{R}^m, q \in \mathcal{Q}$, where L is a non-negative real constant. Let $\epsilon \in (0, 1]$, and let San_{DP} be any (ϵ, δ)-differentially private mechanism that is (α_2, β_2)-useful with respect to $\mathcal{Q}$ for databases of size k.*

Then, using San_{DP}, we can construct a mechanism San_{ZK} that is $(\frac{4k}{n}\epsilon, \frac{4k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$ and is $(\alpha_1 + L\alpha_2, \beta_1 + \beta_2)$-useful with respect to $\mathcal{Q}$ for databases of size n.

Proof. Let San_{ZK} be the mechanism that, on input a database $D \in X^n$, first chooses k random samples without replacement from D, and then forms the database $\widehat{D}$ using the samples. Then, San_{ZK} runs the mechanism San_{DP} on

$\widehat{D}$ to get a synopsis $(\widetilde{D}, R)$ that is α-close to $(\widehat{D}, R_{DB})$ with respect to $\mathcal{Q}$. Then, San_{ZK} outputs the synopsis $(\widetilde{D}, R')$, where R' is the algorithm that, on input the data structure $\widetilde{D}$ and a query q, first runs R on $(\widetilde{D}, q)$, then converts the answer $R(\widetilde{D}, q)$ to an answer on the original database D by computing $f(q, R(\widetilde{D}, q))$, and then outputs $f(q, R(\widetilde{D}, q))$.

By Proposition 16, San_{ZK} is $(\frac{4k}{n}\epsilon, \frac{4k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$.

We now show that San_{ZK} is $(\alpha_1 + L\alpha_2, \beta_1 + \beta_2)$-useful with respect to $\mathcal{Q}$ for databases of size n. Let $D \in X^n$. Since $\mathcal{Q}$ has (k, α_1, β_1)-sample complexity for databases of size n with the converter f, we have that with probability at least $1 - \beta_1$, $San_{ZK}(D)$ forms a database $\widehat{D} \in X^k$ such that $\sup_{q \in \mathcal{Q}} ||q(D) - f(q, q(\widehat{D}))||_1 \leq \alpha_1$. Since San_{DP} is (α_2, β_2)-useful with respect to $\mathcal{Q}$ for databases of size k, we also have that for every database $\widehat{D} \in X^k$, with probability at least $1 - \beta_2$, the mechanism $San_{DP}(\widehat{D})$ run by $San_{ZK}(D)$ outputs a synopsis $(\widetilde{D}, R)$ such that $\sup_{q \in \mathcal{Q}} ||q(\widehat{D}) - R(\widetilde{D}, q)||_1 \leq \alpha_2$.

Thus, with probability at least $1 - (\beta_1 + \beta_2)$, the synopsis $(\widetilde{D}, R')$ that $San_{ZK}(D)$ outputs satisfies $\sup_{q \in \mathcal{Q}} ||q(D) - R'(\widetilde{D}, q)||_1 = \sup_{q \in \mathcal{Q}} ||q(D) - f(q, R(\widetilde{D}, q))||_1 \leq \sup_{q \in \mathcal{Q}} (||q(D) - f(q, q(\widehat{D}))||_1 + ||f(q, q(\widehat{D})) - f(q, R(\widetilde{D}, q))||_1) \leq \alpha_1 + \sup_{q \in \mathcal{Q}} (L||q(\widehat{D}) - R(\widetilde{D}, q)||_1) \leq \alpha_1 + L\alpha_2$. Thus, with probability at least $1 - (\beta_1 + \beta_2)$, San_{ZK} outputs a synopsis $(\widetilde{D}, R')$ that is $(\alpha_1 + L\alpha_2)$-close to (D, R_{DB}) with respect to $\mathcal{Q}$, as required. $\square$

2.4.1 Sample Complexity of a Class of Fraction Queries

There already exist differentially private mechanisms for classes of fraction queries (e.g., the ones in [6] and [25]). To use these mechanisms in Proposition 20, we will show that any class of fraction queries with low VC dimension has low sample complexity.

If the sampling in the definition of sample complexity were done *with* replacement as opposed to *without* replacement, then we could use known learning theory results to show that any class of fraction queries with low VC dimension has low sample complexity. However, the privacy guarantees of the above proposition rely on the fact that the sampling is done without replacement, since the proof uses Proposition 16, which needs this requirement. If the sampling is done with replacement, we are unable to achieve as good privacy parameters.

Our strategy is still to use known learning theory results, but we will adapt known proofs of the results as necessary so that we can use the results to show that any class of fraction queries with low VC dimension has low sample complexity, where the sampling is done without replacement.

A fraction query is a query q of the form $q(D) = \frac{|\{D_i \,:\, i \in [|D|],\ \phi(D_i)=1|}{|D|}$, where D_i is the i^{th} row of the database D, and $\phi : X \to \{0,1\}$ is some predicate. Thus, a fraction query corresponds to some predicate, and for any class $\mathcal{Q}$ of fraction queries, we can consider the class $\widehat{\mathcal{Q}}$ of predicates that correspond to the fraction queries in $\mathcal{Q}$.

We now review some terminology from learning theory. Let $\widehat{\mathcal{Q}}$ be any class of predicates, and let S be any finite subset of X. The restriction of $\widehat{\mathcal{Q}}$ to S, denoted $\widehat{\mathcal{Q}}|_S$, is the set $\{\phi|_S : S \to \{0,1\} \mid \phi \in \widehat{\mathcal{Q}}\}$, i.e., the set of restrictions

to S of all predicates in $\widehat{\mathcal{Q}}$. The growth function $\Pi_{\widehat{\mathcal{Q}}} : \mathbb{N} \to \mathbb{N}$ of $\widehat{\mathcal{Q}}$ is defined by $\Pi_{\widehat{\mathcal{Q}}}(m) = \max_{S' \subseteq X, |S'|=m} |\widehat{\mathcal{Q}}|_{S'}|$. We note that $\Pi_{\widehat{\mathcal{Q}}}(m) \leq 2^m$ for every $m \in \mathbb{N}$, since for any finite $S' \subseteq X$, there are only $2^{|S'|}$ functions from S' to $\{0, 1\}$.

We say that $\widehat{\mathcal{Q}}$ *shatters* S if $|\widehat{\mathcal{Q}}|_S| = 2^{|S|}$, i.e., for every predicate $\phi : S \to \{0, 1\}$, there exists a predicate $\phi' \in \widehat{\mathcal{Q}}$ such that $\phi'|_S = \phi$. The Vapnik-Chervonenkis dimension (VC dimension) of $\widehat{\mathcal{Q}}$ is the size of the largest finite set $S \subseteq X$ shattered by $\widehat{\mathcal{Q}}$, or ∞ if the largest doesn't exist. Equivalently, the VC dimension of $\widehat{\mathcal{Q}}$ is the largest non-negative integer m such that $\Pi_{\widehat{\mathcal{Q}}}(m) = 2^m$, or ∞ if the largest doesn't exist. We note that if $\widehat{\mathcal{Q}}$ is finite, then the VC dimension of $\widehat{\mathcal{Q}}$ is at most $\log_2 |\widehat{\mathcal{Q}}|$, since if a finite set $S \subseteq X$ is shattered by $\widehat{\mathcal{Q}}$, then $|\widehat{\mathcal{Q}}|_S| = 2^{|S|}$, so $\widehat{\mathcal{Q}}$ must contain at least $2^{|S|}$ predicates.

For convenience, when we refer to the VC dimension of a class $\mathcal{Q}$ of fraction queries, we are actually referring to the VC dimension of the class $\widehat{\mathcal{Q}}$ of predicates that corresponds to $\mathcal{Q}$. We now prove a lemma that describes how well k random samples chosen *without* replacement can simultaneously approximate a class of fraction queries. This lemma is similar to a known result in learning theory (e.g., see Theorem 4.3 in [1]).

Lemma 21. *Let $\mathcal{Q}$ be any class of fraction queries, and let $\widehat{\mathcal{Q}}$ be the corresponding class of predicates. Then, for every database $D \in X^n$, $\alpha > 0$, and $k \geq 0$, we have*

$$\Pr[|q(D) - q(\widehat{D})| \geq \alpha \text{ for some } q \in \mathcal{Q}] \leq 4\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}},$$

where the probability is over the choice of the database $\widehat{D}$ formed by choosing k random samples without replacement *from the database D.*

Proof. Fix $D \in X^n$, $\alpha > 0$, $k \geq 0$. Let B be the event that $|q(D) - q(\widehat{D})| \geq \alpha$ for some $q \in \mathcal{Q}$, where $\widehat{D}$ is formed by choosing k random samples without replacement

from the database D. Now, consider choosing another set of k random samples without replacement from the database D, and denote the samples by $\widetilde{D}$. $\widetilde{D}$ is chosen independently of $\widehat{D}$, so $\widetilde{D}$ and $\widehat{D}$ may contain overlapping samples.

Let B' be the event that $|q(\widehat{D}) - q(\widetilde{D})| \geq \frac{\alpha}{2}$ for some $q \in \mathcal{Q}$, where $\widehat{D}$ and $\widetilde{D}$ are chosen as above. Our goal is to bound $\Pr[B]$, and we do so by showing $\Pr[B] \leq 2\Pr[B']$ and bounding $\Pr[B']$ instead. We first note that if $k < \frac{4}{\alpha^2}$, then the RHS of the inequality in the lemma is ≥ 1, and so the lemma holds trivially. Thus, we can assume that $k \geq \frac{4}{\alpha^2}$.

We now show that $\Pr[B] \leq 2\Pr[B']$. To do this, we first show that $\Pr[B' \mid B] \geq \frac{1}{2}$. Suppose event B occurs so that $|q(D) - q(\widehat{D})| \geq \alpha$ for some fixed $\widehat{D}$ and $q \in \mathcal{Q}$. We will show that $\Pr[|q(\widetilde{D}) - q(D)| \leq \frac{\alpha}{2}] \geq \frac{1}{2}$, so with probability $\geq \frac{1}{2}$, the event B' also occurs, since $|q(\widetilde{D}) - q(D)| \leq \frac{\alpha}{2}$ and $|q(D) - q(\widehat{D})| \geq \alpha$ imply that $|q(\widehat{D}) - q(\widetilde{D})| \geq \frac{\alpha}{2}$. Now, by Hoeffding's inequality, we have $\Pr[|q(\widetilde{D}) - q(D)| \leq \frac{\alpha}{2}] \geq 1 - 2e^{-2k(\frac{\alpha}{2})^2}$, and the RHS is $\geq \frac{1}{2}$ if and only if $k \geq \frac{2\ln 4}{\alpha^2}$, which holds since $k \geq \frac{4}{\alpha^2}$. We have shown that $\Pr[B' \mid B] \geq \frac{1}{2}$. Thus, $\frac{\Pr[B']}{\Pr[B]} \geq \frac{\Pr[B' \text{ and } B]}{\Pr[B]} = \Pr[B' \mid B] \geq \frac{1}{2}$, so $\Pr[B] \leq 2\Pr[B']$.

We will now bound $\Pr[B']$. Consider the following process. Choose $\widehat{D}$ and $\widetilde{D}$ as before, and then perform the following swapping process. Let Y be the set of samples of D that were chosen to be in *both* $\widehat{D}$ and $\widetilde{D}$. Regarding $\widehat{D}$ and $\widetilde{D}$ as sets as samples, we arbitrarily pair (using any fixed deterministic algorithm) each sample in $\widehat{D} \setminus Y$ with a sample in $\widetilde{D} \setminus Y$ so that we have a (perfect) matching between $\widehat{D} \setminus Y$ and $\widetilde{D} \setminus Y$; we also pair each sample in $\widehat{D} \cap Y$ with the corresponding (equal) sample in $\widetilde{D} \cap Y$. Then, for each matched pair x, y, we swap x and y with probability $\frac{1}{2}$. Let $\widehat{D}'$ and $\widetilde{D}'$ denote the resulting $\widehat{D}$ and $\widetilde{D}$. It is easy to see that the sets $\widehat{D}'$ and $\widetilde{D}'$ are identically distributed to $\widehat{D}$ and $\widetilde{D}$. (The main

difference between this proof and classic proofs (e.g., see [1]) of the corresponding learning theory result is in this swapping procedure, where we do the swapping in a particular way to ensure that $\widehat{D}'$ and $\widetilde{D}'$ are identically distributed to $\widehat{D}$ and $\widetilde{D}$ even though our sampling is done without replacement.)

Let B'' be the event that $|q(\widehat{D}') - q(\widetilde{D}')| \geq \frac{\alpha}{2}$ for some $q \in \mathcal{Q}$. Then $\Pr[B'] = \Pr[B'']$, so it suffices to bound $\Pr[B'']$. We will show that $\Pr[B''] \leq 2\Pi_Q(2k)e^{-\frac{k\alpha^2}{8}}$ by showing that $\Pr[B'' \mid \widehat{D}, \widetilde{D}] \leq 2\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}}$ for every fixed $\widehat{D}$ and $\widetilde{D}$ that can be sampled while generating $\widehat{D}'$ and $\widetilde{D}'$.

To this end, fix $\widehat{D}$ and $\widetilde{D}$. Let $t = |(\widehat{\mathcal{Q}}|_{\widehat{D}' \cup \widetilde{D}'})|$, where $\widehat{D}' \cup \widetilde{D}'$ is regarded as a set of elements in X. We note that $t \leq \Pi_{\widehat{\mathcal{Q}}}(2k)$, since $|\widehat{D}' \cup \widetilde{D}'| \leq 2k$. Since $|(\widehat{\mathcal{Q}}|_{\widehat{D}' \cup \widetilde{D}'})| = t$, we can choose t predicates $\phi_1, \ldots, \phi_t \in \widehat{\mathcal{Q}}$ such that for any predicate $\phi \in \widehat{\mathcal{Q}}$, there exists an $i \in \{1, \ldots, t\}$ such that $\phi|_{\widehat{D}' \cup \widetilde{D}'} = \phi_i|_{\widehat{D}' \cup \widetilde{D}'}$, i.e., $\phi(x) = \phi_i(x)$ for every $x \in \widehat{D}' \cup \widetilde{D}'$.

Let $q_1, \ldots, q_t$ be the fraction queries in $\mathcal{Q}$ that correspond to the predicates $\phi_1, \ldots, \phi_t$. Then, for any fraction query $q \in \mathcal{Q}$, there exists an $i \in \{1, \ldots, t\}$ such that $q(\widehat{D}') = q_i(\widehat{D}')$ and $q(\widetilde{D}') = q_i(\widetilde{D}')$. Thus, B'' occurs if and only if $|q_i(\widehat{D}') - q_i(\widetilde{D}')| \geq \frac{\alpha}{2}$ for some $i \in \{1, \ldots, t\}$. Then, by the union bound, we have the following:

$$\begin{aligned}\Pr[B'' \mid \widehat{D}, \widetilde{D}] &\leq t \max_{1 \leq i \leq t} \Pr[|q_i(\widehat{D}') - q_i(\widetilde{D}')| \geq \frac{\alpha}{2} \mid \widehat{D}, \widetilde{D}] \\ &\leq \Pi_Q(2k) \max_{1 \leq i \leq t} \Pr[|q_i(\widehat{D}') - q_i(\widetilde{D}')| \geq \frac{\alpha}{2} \mid \widehat{D}, \widetilde{D}].\end{aligned}$$

Fix an $i \in \{1, \ldots, t\}$. For convenience, we order the elements in $\widehat{D}$, $\widetilde{D}$, $\widehat{D}'$, and $\widetilde{D}'$ as $\widehat{D}_1, \ldots, \widehat{D}_k$, $\widetilde{D}_1, \ldots, \widetilde{D}_k$, $\widehat{D}'_1, \ldots, \widehat{D}'_k$, and $\widetilde{D}'_1, \ldots, \widetilde{D}'_k$, respectively, so that for every $j \in \{1, \ldots, k\}$, $\widehat{D}_j$ is matched with $\widetilde{D}_j$, and $\widehat{D}'_j$ is matched with $\widetilde{D}'_j$ according

to the pairing scheme described above. Now, observe that

$$\begin{aligned}
&\Pr[|q_i(\widehat{D}') - q_i(\widetilde{D}')| \geq \frac{\alpha}{2} \mid \widehat{D}, \widetilde{D}] \\
&= \Pr[|\frac{1}{k}\sum_{j=1}^{k} \phi_i(\widehat{D}'_j) - \frac{1}{k}\sum_{j=1}^{k} \phi_i(\widetilde{D}'_j)| \geq \frac{\alpha}{2} \mid \widehat{D}, \widetilde{D}] \\
&= \Pr[|\frac{1}{k}\sum_{j=1}^{k} (\phi_i(\widehat{D}'_j) - \phi_i(\widetilde{D}'_j))| \geq \frac{\alpha}{2} \mid \widehat{D}, \widetilde{D}] \\
&= \Pr[|\frac{1}{k}\sum_{j=1}^{k} (|\phi_i(\widehat{D}_j) - \phi_i(\widetilde{D}_j)| \cdot z_j)| \geq \frac{\alpha}{2} \mid \widehat{D}, \widetilde{D}], \quad z_j \leftarrow \{-1, 1\} \\
&\leq 2e^{-\frac{k\alpha^2}{8}},
\end{aligned}$$

where $z_j \leftarrow \{-1, 1\}$ means that z_j is sampled uniformly from $\{-1, 1\}$, and the last inequality follows from Hoeffding's inequality. Thus, $\Pr[B'' \mid \widehat{D}, \widetilde{D}] \leq 2\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}}$, so $\Pr[B''] \leq 2\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}}$. Therefore, $\Pr[B] \leq 2\Pr[B'] = 2\Pr[B''] \leq 4\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}}$, as required. □

The above lemma gives an upper bound on the probability that k random samples (chosen without replacement) does not simultaneously approximate a class $\mathcal{Q}$ of fraction queries well, and the upper bound involves $\Pi_{\widehat{\mathcal{Q}}}(2k)$. The following lemma gives an upper bound on $\Pi_{\widehat{\mathcal{Q}}}(2k)$ in terms of the VC dimension of $\widehat{\mathcal{Q}}$.

Lemma 22. *Let $\widehat{\mathcal{Q}}$ be any class of predicates with finite VC dimension $d \geq 1$. Then, for every integer $m \geq d$, we have $\Pi_{\widehat{\mathcal{Q}}}(m) \leq (\frac{em}{d})^d$.*

Proof. This lemma is a well-known result in learning theory and is a corollary of "Sauer's lemma", which states that for any class $\widehat{\mathcal{Q}}$ of predicates with finite VC dimension d, $\Pi_{\widehat{\mathcal{Q}}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}$ for all nonnegative integers m (e.g., see [72]). A proof of Sauer's lemma, as well as this lemma, can be found in [1]. □

We can now combine Lemmas 21 and 22 to get the following proposition.

Proposition 23. *Let $\mathcal{Q}$ be any class of fraction queries with finite VC dimension $d \geq 1$. Then, for every $n \geq 1$, $k \geq d/2$, and $\beta \in (0, 1]$, $\mathcal{Q}$ has (k, α, β)-sample complexity for databases of size n with the converter $f(q, x) = x$, where $\alpha = \frac{2\sqrt{2}}{\sqrt{k}}\sqrt{d\ln(\frac{2ek}{d}) + \ln(\frac{4}{\beta})}$.*

Also, for every $n \geq 1$ and $\alpha, \beta \in (0, 1]$, $\mathcal{Q}$ has (k, α, β)-sample complexity for databases of size n with converter $f(q, x) = x$, where k is any non-negative integer satisfying $k \geq \frac{16}{\alpha^2}(2d\ln(\frac{6}{\alpha}) + \ln(\frac{4}{\beta}))$.

Proof. Let $\widehat{\mathcal{Q}}$ be the class of predicates that corresponds to the class $\mathcal{Q}$ of fraction queries. Let $n \geq 1$, $k \geq d/2$, $\alpha > 0$, $\beta \in (0, 1]$, and $D \in X^n$. By Lemma 21, we have $\Pr[|q(D) - q(\widehat{D})| \geq \alpha \text{ for some } q \in \mathcal{Q}] \leq 4\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}}$, where $\widehat{D}$ is a database formed by choosing k random samples without replacement from the database D. Thus, $\mathcal{Q}$ has $(k, \alpha, 4\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}})$-sample complexity for databases of size n with converter $f(q, x) = x$.

Rearranging the inequality $4\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}} \leq \beta$, we get $\alpha \geq \sqrt{\frac{8}{k}(\ln(\Pi_{\widehat{\mathcal{Q}}}(2k)) + \ln(\frac{4}{\beta}))}$. We have $\Pi_{\widehat{\mathcal{Q}}}(2k) \leq (\frac{2ek}{d})^d$ by Lemma 22, so $\alpha \geq \frac{2\sqrt{2}}{\sqrt{k}}\sqrt{d\ln(\frac{2ek}{d}) + \ln(\frac{4}{\beta})}$ implies that $4\Pi_{\widehat{\mathcal{Q}}}(2k)e^{-\frac{k\alpha^2}{8}} \leq \beta$. Thus, $\mathcal{Q}$ has (k, α', β)-sample complexity for databases of size n with converter $f(q, x) = x$, where $\alpha' = \frac{2\sqrt{2}}{\sqrt{k}}\sqrt{d\ln(\frac{2ek}{d}) + \ln(\frac{4}{\beta})}$, as required.

Now, let $\alpha \in (0, 1]$ and k be any non-negative integer satisfying $k \geq \frac{16}{\alpha^2}(2d\ln(\frac{6}{\alpha}) + \ln(\frac{4}{\beta}))$. We note that $k \geq d/2$ still holds. Thus, from the argument above, to show that $\mathcal{Q}$ has (k, α, β)-sample complexity for databases of size n with converter $f(q, x) = x$, it suffices to show that $\alpha \geq \frac{2\sqrt{2}}{\sqrt{k}}\sqrt{d\ln(\frac{2ek}{d}) + \ln(\frac{4}{\beta})}$ holds. Rearranging $\alpha \geq \frac{2\sqrt{2}}{\sqrt{k}}\sqrt{d\ln(\frac{2ek}{d}) + \ln(\frac{4}{\beta})}$, we get $k \geq \frac{8}{\alpha^2}(d\ln(k) + d\ln(\frac{2e}{d}) + \ln(\frac{4}{\beta}))$.

We now use the inequality $\ln a \leq ab + \ln\frac{1}{b} - 1$, which holds for all $a, b > 0$; this can be easily shown by using the inequality $1 + x \leq e^x$ (which holds for all

$x \in \mathbb{R}$), setting x to $ab-1$, and rearranging the inequality. Applying the inequality $\ln a \leq ab + \ln \frac{1}{b} - 1$ with $a = k$ and $b = \frac{\alpha^2}{16d}$, we have $\ln k \leq \frac{\alpha^2}{16d}k + \ln(\frac{16d}{\alpha^2}) - 1 = \frac{\alpha^2}{16d}k + \ln(\frac{16d}{e\alpha^2})$.

Thus, it suffices to show that $k \geq \frac{8}{\alpha^2}(d\frac{\alpha^2}{16d}k + d\ln(\frac{16d}{e\alpha^2}) + d\ln(\frac{2e}{d}) + \ln(\frac{4}{\beta}))$. Rearranging this inequality, we get $k \geq \frac{16}{\alpha^2}(2d\ln(\frac{4\sqrt{2}}{\alpha}) + \ln(\frac{4}{\beta}))$, which holds by definition of k. □

2.4.2 Constructing Zero-Knowledge Private Mechanisms for a Class of Fraction Queries

Proposition 23 gives us a bound on the sample complexity of any class $\mathcal{Q}$ of fraction queries in terms of its VC dimension. Proposition 20 allows us to convert differentially private mechanism to zero-knowledge private mechanisms for any class of queries with low sample complexity. Thus, we now combine these two propositions with existing differentially private mechanisms for classes of fraction queries with low VC dimension.

The following proposition is obtained by using the differentially private mechanism in [6].

Proposition 24. *Let $\mathcal{Q}$ be any class of fraction queries with finite VC dimension $d \geq 1$, and suppose the data universe X is finite. Let $n \geq 1$ and $\epsilon, \alpha, \beta \in (0, 1]$. Then, for every integer $k = k(n)$ satisfying $k \geq O(\frac{(\log|X|)d\log(1/\alpha)+\log(1/\beta)}{\alpha^3\epsilon})$, there exists a mechanism San that is (α, β)-useful with respect to $\mathcal{Q}$ for databases of size n, and is $\frac{4k}{n}\epsilon$-zero-knowledge private with respect to $RS(k(\cdot))$.*

Proof. By Proposition 23, $\mathcal{Q}$ has $(k', \frac{\alpha}{2}, \frac{\beta}{2})$-sample complexity for databases of size

n with the converter $f(q,x) = x$, where k' is any non-negative integer satisfying $k' \geq \frac{64}{\alpha^2}(2d\ln(\frac{12}{\alpha})+\ln(\frac{8}{\beta}))$. We note that $|f(q,x)-f(q,y)| \leq 1\cdot|x-y|$ for every $x, y \in \mathbb{R}, q \in \mathcal{Q}$. Let San_{DP} be the ϵ-differentially private mechanism in [6] that is $(\frac{\alpha}{2}, \frac{\beta}{2})$-useful with respect to $\mathcal{Q}$ for databases of size $k'' \geq O(\frac{(\log|X|)d\log(1/\alpha)}{\alpha^3\epsilon} + \frac{\log(1/\beta)}{\alpha\epsilon})$. (This result in [6] actually assumes that $X = \{0,1\}^{d'}$ for some d', but as mentioned in the paper, the result can be easily extended to any finite set X.)

Let $k \geq \max\{\frac{64}{\alpha^2}(2d\ln(\frac{12}{\alpha}) + \ln(\frac{8}{\beta})), O(\frac{(\log|X|)d\log(1/\alpha)}{\alpha^3\epsilon} + \frac{\log(1/\beta)}{\alpha\epsilon})\} = O(\frac{(\log|X|)d\log(1/\alpha)+\log(1/\beta)}{\alpha^3\epsilon})$. Then, by Proposition 20, we can use San_{DP} to construct a mechanism San_{ZK} that is $(\frac{4k}{n}\epsilon, \frac{4k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$ and is (α,β)-useful with respect to $\mathcal{Q}$ for databases of size n. □

We now use the differentially private mechanism in [25] to obtain the following proposition.

Proposition 25. *Let $\mathcal{Q}$ be any finite class of fraction queries with finite VC dimension $d \geq 1$, and suppose the data universe X is finite. Let $n \geq 1$, $\epsilon \in (0,1]$, and $\kappa \geq 1$. Then, for every integer $k = k(n)$ satisfying $k \geq \frac{d}{2}$, there exists a mechanism San that is $(\tilde{O}(\frac{\sqrt{d+\kappa}}{\sqrt{k}} + \frac{\sqrt{\log|X|}(\log|\mathcal{Q}|)\kappa^{3/2}}{\epsilon\sqrt{k}}), e^{-\kappa})$-useful with respect to $\mathcal{Q}$ for databases of size n, and is $(\frac{4k}{n}\epsilon, \frac{4k}{n}e^{-\kappa})$-zero-knowledge private with respect to $RS(k(\cdot))$.*

Proof. Let $k \geq \frac{d}{2}$. Then, by Proposition 23, $\mathcal{Q}$ has $(k, \alpha, \frac{1}{2}e^{-\kappa})$-sample complexity for databases of size n with the converter $f(q,x) = x$, where $\alpha = \frac{2\sqrt{2}}{\sqrt{k}}\sqrt{d\ln(\frac{2ek}{d}) + \ln(8e^{\kappa})} = \tilde{O}(\frac{\sqrt{d+\kappa}}{\sqrt{k}})$. We note that $|f(q,x)-f(q,y)| \leq 1\cdot|x-y|$ for every $x, y \in \mathbb{R}, q \in \mathcal{Q}$. Let San_{DP} be the $(\epsilon, e^{-\kappa})$-differentially private mechanism in [25] that is $(\tilde{O}(\frac{\sqrt{\log|X|}(\log|\mathcal{Q}|)\kappa^{3/2}}{\epsilon\sqrt{k}}), \frac{1}{2}e^{-\kappa})$-useful with respect to $\mathcal{Q}$ for databases of size k.

Then, by Proposition 20, we can use San_{DP} to construct a mechanism San_{ZK} that is $(\frac{4k}{n}\epsilon, \frac{4k}{n}\delta)$-zero-knowledge private with respect to $RS(k(\cdot))$ and is $(\tilde{O}(\frac{\sqrt{d+\kappa}}{\sqrt{k}} + \frac{\sqrt{\log|X|}(\log|\mathcal{Q}|)\kappa^{3/2}}{\epsilon\sqrt{k}}), e^{-\kappa})$-useful with respect to $\mathcal{Q}$ for databases of size n. □

In general, Propositions 23 and 20 can be used to convert useful differentially private mechanisms to useful zero-knowledge private mechanisms for classes of fraction queries with low VC dimension.

Recall that if $\mathcal{Q}$ is a finite class of fraction queries, then the VC dimension of $\mathcal{Q}$ is at most $\log_2|\mathcal{Q}|$. Thus, we can replace the VC dimension d in Proposition 25 by $\log|\mathcal{Q}|$ (we can also do this in Proposition 24 if we assume $\mathcal{Q}$ is finite). However, it is possible that the VC dimension of a class $\mathcal{Q}$ of fraction queries is substantially smaller than $\log_2|\mathcal{Q}|$ (e.g., see [1]), especially if $\mathcal{Q}$ is infinite.

2.5 Zero-Knowledge Private Release of Graph Properties

In this section, we first generalize statistical (row) databases to graphs with personal data so that we can model a social network and privately release information that is dependent on the graph structure. We then discuss how to model privacy in a social network, and we construct a sample of zero-knowledge private mechanisms that release certain information about the graph structure of a social network.

We represent a social network using a graph whose vertices correspond to people (or other social entities) and whose edges correspond to social links between them, and a vertex can have certain personal data associated with it. There are various types of information about a social network one may want to release, such as information about the people's data, information about the structure of the social

network, and/or information that is dependent on both. In general, we want to ensure privacy of each person's personal data as well as the person's links to other people (i.e., the list of people the person is linked to via edges).

To formally model privacy in social networks, let $\mathcal{G}_n$ be a class of graphs on n vertices where each vertex includes personal data. (When we refer to a graph $G \in \mathcal{G}_n$, the graph always includes the personal data of each vertex.) The graph structure is represented by an adjacency matrix, and each vertex's personal data is represented by an element in X. For the privacy of individuals, we use our zero-knowledge privacy definition with some minor modifications:

- ϵ-zero-knowledge privacy is defined as before except we change "database $D \in X^n$" to "graph $D \in \mathcal{G}_n$", and we define $(D_{-i}, \perp)$ to be the graph D except the personal data of vertex i is replaced by $\perp$ and all the edges incident to vertex i are removed (by setting the corresponding entries in the adjacency matrix to 0); thus $(D_{-i}, \perp)$ is essentially D with person i's personal data and links removed.

We now consider functions $g : \mathcal{G}_n \to \mathbb{R}^m$, and we redefine the L_1-sensitivity of g to be $\Delta(g) = \max\{||g(D') - g(D'')||_1 : D', D'' \in \mathcal{G}_n \text{ s.t. } (D'_{-i}, \perp) = (D''_{-i}, \perp) \text{ for some } i \in [n]\}$. We also redefine $RS(k(\cdot))$ so that the algorithms in $RS(k(\cdot))$ are given a graph $D \in \mathcal{G}_n$ and are allowed to choose $k(n)$ random vertices without replacement and read their personal data; however, the algorithms are not allowed to read the structure of the graph, i.e., the adjacency matrix. It is easy to verify that all our previous results still hold when we consider functions $g : \mathcal{G}_n \to \mathbb{R}^m$ on graphs and use the new definition of $\Delta(g)$ and $RS(k(\cdot))$.

Since a social network has more structure than a statistical database containing

a list of values, we now consider more general models of aggregate information that allow us to release more information about social networks:

- $RSE(k(\cdot), s) = k(\cdot)$ random samples with exploration of s vertices: the class of algorithms T such that on input a graph $D \in \mathcal{G}_n$, T chooses $k(n)$ random vertices uniformly with or without replacement (or a combination of both). For each sampled vertex v, T is allowed to explore the graph locally at v until s vertices (including the sampled vertex) have been visited. The data of any visited vertex can be read. (RSE stands for "random samples with exploration".)
- $RSN(k(\cdot), d) = k(\cdot)$ random samples with neighborhood of radius d: same as $RSE(k(\cdot), s)$ except that while exploring locally, instead of exploring until s vertices have been visited, T is allowed to explore up to a distance of d from the sampled vertex. (RSN stands for "random samples with neighborhood".)

Note that these models of aggregate information include $RS(k(\cdot))$ as a special case. We can also consider variants of these models where instead of allowing the data of any visited vertex to be read, only the data of the $k(n)$ randomly chosen vertices can be read. (The data of the "explored" vertices cannot be read.)

Remark. In the above models, vertices (people) in the graph with high degree may be visited with higher probability than those with low degree. Thus, the privacy of these people may be less protected. However, this is often the case in social networks, where people with very many friends will naturally have less privacy than those with few friends.

We now show how to combine Proposition 12 (the connection between sample complexity and zero-knowledge privacy) with recent sublinear time algorithms to

privately release information about the graph structure of a social network. For simplicity, we assume that the degree of every vertex is bounded by some constant $d_{\max}$ (which is often the case in a social network anyway).[1]

Let $\mathcal{G}_n$ be the set of all graphs on n vertices where every vertex has degree at most $d_{\max}$. We assume that $d_{\max}$ is publicly known. Let $M = \frac{d_{\max} n}{2}$ be an upper bound on the number of edges of a graph in $\mathcal{G}_n$. For any graph $G \in \mathcal{G}$, the (relative) distance from G to the some property Π, denoted $dist(G, \Pi)$, is the least number of edges that need to be modified (added/removed) in G in order to make it satisfy property Π, divided by M.

Theorem 26. *Let Conn, Eul, and CycF be the property of being connected, Eulerian*[2]*, and cycle-free, respectively. Let $\bar{d}(G)$ denote the average degree of a vertex in G. Let $\epsilon, \delta > 0$, and let $K \in \mathbb{Z}^+$. Then, for the class of graphs $\mathcal{G}_n$, we have the following results:*

1. *The mechanism $San(G) = dist(G, Conn) + Lap(\frac{2/n+\delta}{\epsilon})$ is $\epsilon + e^{-(K-\epsilon/\delta)}$-zero-knowledge private with respect to $RSE(k(\cdot), s)$, where $k(n) = O(\frac{K}{(\delta d_{\max})^2})$ and $s = O(\frac{1}{\delta d_{\max}})$.*
2. *The mechanism $San(G) = dist(G, Eul) + Lap(\frac{4/n+\delta}{\epsilon})$ is $\epsilon + e^{-(K-\epsilon/\delta)}$-zero-knowledge private with respect to $RSE(k(\cdot), s)$, where $k(n) = O(\frac{K}{(\delta d_{\max})^2})$ and $s = O(\frac{1}{\delta d_{\max}})$.*
3. *The mechanism $San(G) = dist(G, CycF) + Lap(\frac{2/n+\delta}{\epsilon})$ is $\epsilon + e^{-(K-\epsilon/\delta)}$-zero-knowledge private with respect to $RSE(k(\cdot), s)$, where $k(n) = O(\frac{K}{\delta^2})$ and $s = O(\frac{1}{\delta d_{\max}})$.*

[1] Weaker results can still be established without this assumption.

[2] A graph G is Eulerian if there exists a path in G that traverses every edge of G exactly once.

4. *The mechanism* $San(G) = \bar{d}(G) + Lap(\frac{2d_{\max}/n+\delta L}{\epsilon})$ *is* $\epsilon + e^{-(K-\epsilon/\delta)}$*-zero-knowledge private with respect to* $RSN(k(\cdot), 2)$*, where* $k(n) = O(K\sqrt{n}\log^2 n \cdot \frac{1}{\delta^{9/2}}\log(\frac{1}{\delta}))$*. Here, we further assume that* $\delta \in (0, \frac{1}{2})$ *and every graph in* $\mathcal{G}$ *has no isolated vertices and the average degree of a vertex is bounded by* L*.*

The results of the above theorem are obtained by combining Proposition 12 (the connection between sample complexity and zero-knowledge privacy) with sublinear time algorithms from [56] (for results 1, 2, and 3) and [38] (for result 4). Intuitively, the sublinear algorithms give bounds on the sample complexity of the functions ($dist(G, Conn)$, etc.) with respect to $RSE(k(\cdot), s)$ or $RSN(k(\cdot), d)$.

Proof.

Distance approximation to connectivity: Let $San(G) = dist(G, Conn) + Lap(\lambda)$, where $\lambda = \frac{2/n+\delta}{\epsilon}$. In [56], Marko and Ron have given an algorithm that approximates the distance to connectivity to within an additive error δ with probability at least $\frac{2}{3}$. The algorithm does this by randomly choosing $O(\frac{1}{(\delta d_{\max})^2})$ vertices, and for each chosen vertex, exploring the graph locally from the vertex until at most $O(\frac{1}{\delta d_{\max}})$ vertices have been reached. Here is the algorithm from [56] (modified slightly to fit this context):

1. Uniformly and independently sample $t = \frac{32}{(\delta d_{\max})^2}$ vertices from G. Let S be the multiset of the sampled vertices.
2. For every $v \in S$, perform a BFS starting from v until $\frac{4}{\delta d_{\max}}$ vertices have been reached or v's connected component has been found. Let $\widehat{n}_v$ be the number of vertices in v's connected component in case it was found. Otherwise $\widehat{n}_v = \infty$.
3. Let $\widehat{C} = \frac{n}{t}\sum_{v \in S}(\frac{1}{\widehat{n}_v})$ and output $\frac{1}{M}(\widehat{C} - 1)$.

By running the above algorithm $O(K)$ times and outputting the median value, we can increase the success probability to $1-e^{-K}$. Thus, $dist(G, Conn)$ has $(\delta, 1-e^{-K})$ sample complexity with respect to $RSE(k(\cdot), s)$, where $k(n) = O(\frac{K}{(\delta d_{\max})^2})$ and $s = O(\frac{1}{\delta d_{\max}})$. By Proposition 12, San is $\ln(e^{\frac{2/n+\delta}{\lambda}} + e^{\frac{1}{\lambda}-K})$-zero-knowledge private with respect to $RSE(k(\cdot), s)$.

Now, observe that $\ln(e^{\frac{2/n+\delta}{\lambda}} + e^{\frac{1}{\lambda}-K}) \leq \ln(e^{\epsilon} + e^{\epsilon/\delta-K}) \leq \epsilon + e^{\epsilon/\delta-K} = \epsilon + e^{-(K-\epsilon/\delta)}$. Thus, we have the following result:

- The mechanism $San(G) = dist(G, Conn) + Lap(\frac{2/n+\delta}{\epsilon})$ is $\epsilon + e^{-(K-\epsilon/\delta)}$-zero-knowledge private with respect to $RSE(k(\cdot), s)$, where $k(n) = O(\frac{K}{(\delta d_{\max})^2})$ and $s = O(\frac{1}{\delta d_{\max}})$.

Distance approximation to being Eulerian: Let $San(G) = dist(G, Eul) + Lap(\lambda)$, where $\lambda = \frac{4/n+\delta}{\epsilon}$. In [56], Marko and Ron have given an algorithm that approximates the distance to being Eulerian to within an additive error δ with probability at least $\frac{2}{3}$. The algorithm does this by randomly choosing $O(\frac{1}{(\delta d_{\max})^2})$ vertices, and for each chosen vertex, exploring the graph locally from the vertex until at most $O(\frac{1}{\delta d_{\max}})$ vertices have been reached.

By a similar analysis as in the "distance approximation to connectivity" example, we get the following result:

- The mechanism $San(G) = dist(G, Eul) + Lap(\frac{4/n+\delta}{\epsilon})$ is $\epsilon + e^{-(K-\epsilon/\delta)}$-zero-knowledge private with respect to $RSE(k(\cdot), s)$, where $k(n) = O(\frac{K}{(\delta d_{\max})^2})$ and $s = O(\frac{1}{\delta d_{\max}})$.

Distance approximation to cycle freeness: Let $San(G) = dist(G, CycF) + Lap(\lambda)$, where $\lambda = \frac{2/n+\delta}{\epsilon}$. In [56], Marko and Ron have given an algorithm that approxi-

mates the distance to being cycle-free to within an additive error δ with probability at least $\frac{2}{3}$. The algorithm does this by randomly choosing $O(\frac{1}{\delta^2})$ vertices, and for each chosen vertex, exploring the graph locally from the vertex until at most $O(\frac{1}{\delta d_{\max}})$ vertices have been reached.

By a similar analysis as in the "distance approximation to connectivity" example, we get the following result:

- The mechanism $San(G) = dist(G, CycF) + Lap(\frac{2/n+\delta}{\epsilon})$ is $\epsilon + e^{-(K-\epsilon/\delta)}$-zero-knowledge private with respect to $RSE(k(\cdot), s)$, where $k(n) = O(\frac{K}{\delta^2})$ and $s = O(\frac{1}{\delta d_{\max}})$.

Approximating the average degree of a graph: Let $San(G) = \bar{d}(G) + Lap(\lambda)$, where $\lambda = \frac{2d_{\max}/n+\delta L}{\epsilon}$. In [38], Goldreich and Ron have shown that $\bar{d}(G)$ can be approximated by an algorithm (which needs the extra assumptions stated in the above theorem) to within a multiplicative error of $(1+\delta)$ with probability at least $\frac{2}{3}$, by randomly choosing $O(\sqrt{n}\log^2 n \cdot \frac{1}{\delta^{9/2}}\log(\frac{1}{\delta}))$ vertices, and for each chosen vertex, exploring the graph locally from the vertex up to a distance of 2. By running the approximation algorithm $O(K)$ times and outputting the median value, we can increase the success probability to $1 - 2^{-K}$.

Such an algorithm is a $(\delta L, 1 - 2^{-K})$-sampler for $\bar{d}(G)$ with respect to $RSN(k(\cdot), 2)$, where $k(n) = O(K\sqrt{n}\log^2 n \cdot \frac{1}{\delta^{9/2}}\log(\frac{1}{\delta}))$. By Proposition 12, San is $\ln(e^{\frac{2d_{\max}/n+\delta L}{\lambda}} + e^{\frac{L}{\lambda}-K})$-zero-knowledge private with respect to $RSN(k(\cdot), 2)$.

Now, observe that $\ln(e^{\frac{2d_{\max}/n+\delta L}{\lambda}} + e^{\frac{L}{\lambda}-K}) \leq \ln(e^{\epsilon} + e^{\epsilon/\delta-K}) \leq \epsilon + e^{-(K-\epsilon/\delta)}$. Thus, we have the following result:

- The mechanism $San(G) = \bar{d}(G) + Lap(\frac{2d_{\max}/n+\delta L}{\epsilon})$ is $\epsilon + e^{-(K-\epsilon/\delta)}$-

zero-knowledge private with respect to $RSN(k(\cdot), 2)$, where $k(n) = O(K\sqrt{n}\log^2 n \cdot \frac{1}{\delta^{9/2}}\log(\frac{1}{\delta}))$.

□

There are already many (non-private) sublinear time algorithms for computing information about graphs whose accuracy is proved formally (e.g., see [38, 12, 56, 36, 44, 37, 68]) or demonstrated empirically (e.g, see [48, 47]). We leave for future work to investigate whether these (or other) sublinear algorithms can be used to get zero-knowledge private mechanisms.

CHAPTER 3

CROWD-BLENDING PRIVACY

3.1 Introduction

In this chapter, we present our work on crowd-blending privacy.

Crowd-Blending Privacy – A New Privacy Definition. Let us now turn to describing our new privacy definition, which we call *crowd-blending privacy*. We say that an individual *blends* with another individual with respect to a mechanism San if the two individuals are *indistinguishable by the mechanism* San, i.e., whenever we have a database containing either one or both of the individuals, we can replace one of the individual's data with the other individual's data, and the mechanism's output distribution remains essentially the same. We say that an individual t *blends in a crowd of* k *people in the database* D *with respect to the mechanism* San if there exist at least $k-1$ other individuals in the database D that blend with individual t with respect to San. The intuition behind this notion is that if an individual t blends in a crowd of k people in the database, then the mechanism essentially does not release any information about individual t beyond the general characteristics of the crowd of k people; in particular, the mechanism does not release any personal information that is specific to individual t and no one else.

Roughly speaking, we say that a mechanism San is *crowd-blending private* if the following property holds: For every database and every individual in the database, either the individual *blends in a crowd of* k *people in the database with respect to* San, or the mechanism San *essentially ignores the individual's data*.

We do not claim that crowd-blending privacy provides sufficiently strong privacy protection in *all* scenarios: the key weakening with respect to differential privacy is that an attacker who knows the data of everyone in an individual i's crowd (except i) may learn information about individual i, as long as this information is "general" in the sense that it applies to the entire crowd. For instance, if the attacker knows everyone in the crowd of individual i, it may deduce that i has, say, three children, as long as everyone in i's crowd has three children. Although to some extent, this may be viewed as a privacy violation (that would not be allowed by the notion of differential privacy), we would argue that the attribute leaked about individual i is "non-sensitive" as it is shared by a sufficiently large crowd. Thus, we view this weakening as desirable in many contexts as it allows us to trade privacy of "non-sensitive information" for improved utility.

A potentially more serious deficiency of the definition is that (in contrast to differential and zero-knowledge privacy) crowd-blending privacy is not closed under composition: San_1 and San_2 may both be crowd-blending private, but the crowds for an individual with respect to San_1 and San_2 could be essentially disjoint, making the individual's crowd for the combination of San_1 and San_2 very small. Although we view composition as an important property of a privacy definition, our goal here is to study the weakest possible "meaningful" definition of "stand-alone" privacy that when combined with pre-sampling leads to strong privacy notions (such as differential and zero-knowledge privacy) that themselves are closed under composition.

3.1.1 New Database Mechanisms

As it turns out, achieving crowd-blending privacy is significantly easier than achieving differential privacy, and crowd-blending private mechanisms may yield significantly higher utility than differentially private ones.

Privately Releasing Histograms with No Noise for Sufficiently Large Counts. We show that we can release histograms with crowd-blending privacy where no noise is added to bins with a sufficiently large count (and only a small amount of noise is added to bins with a small count). Intuitively, individuals in the same bin blend with each other; thus, the individuals that belong to a bin with a sufficiently large count already blend in a crowd, so no noise needs to be added to the bin. It is easy to see that it is impossible to release the exact count of a bin in a histogram while satisfying differential privacy or zero-knowledge privacy. Using crowd-blending privacy, we can overcome this limitation (for bins with a sufficiently large count) and achieve better utility. These results can be found in Section 3.3.1.

Privately Releasing Synthetic Data Points in $\mathbb{R}^d$ for Computing Smooth Functions. Given a class $\mathcal{C}$ of counting queries whose size is not too large, it is shown in [6] how to release a synthetic database for approximating all the queries in $\mathcal{C}$ simultaneously while satisfying differential privacy; however, the mechanism is not necessarily efficient. It is known that it is impossible (assuming the existence of one-way functions) to *efficiently* and privately release a synthetic database for approximating certain classes of counting queries, such as the class of all 2-way marginals (see [76, 24]). However, these query functions are non-smooth in the

sense that even slightly changing one row of the input database can affect the output of the query functions quite a lot. Here, we focus on efficiently and privately releasing synthetic data for approximating *all* "smooth" functions $g : (\mathbb{R}^d)^* \to \mathbb{R}^m$.

Roughly speaking, a function $g : (\mathbb{R}^d)^* \to \mathbb{R}^m$ is *smooth* if the value of g does not change much when we perturb the data points of the input slightly. We show that we can *efficiently* release synthetic data points in $\mathbb{R}^d$ for approximating *all* smooth functions simultaneously while satisfying crowd-blending privacy. On the other hand, we show that there are smooth functions that cannot even be approximated with non-trivial utility from any synthetic data that has been released with differential privacy (even if the differentially private mechanism is *inefficient*). These results can be found in Section 3.4.

3.1.2 From Crowd-Blending Privacy to Zero-Knowledge Privacy

Our main technical result shows that if we combine a crowd-blending private mechanism with a natural pre-sampling step, then the combined algorithm satisfies zero-knowledge privacy (and thus differential privacy as well). We envision the pre-sampling step as being part of the data collection process, where individuals in some population are sampled and asked for their data. Thus, if data is collected using random sampling of individuals from some population, and next sanitized using a crowd-blending private mechanism, then the resulting process ensures zero-knowledge privacy.

We first prove our main theorem for the case where the pre-sampling step samples each individual in the population with probability p independently. In reality,

the sampling performed during data collection may be slightly biased or done slightly incorrectly, and an adversary may know whether certain individuals were sampled or not. Thus, we next extend our main theorem to also handle the case where the sampling probability is not necessarily the same for everybody, but the sampling is still "robust" in the sense that most individuals are sampled independently with probability in between p and p' (this probability can even depend on the individual's data), where p and p' are relatively close to one another, while the remaining individuals are sampled independently with arbitrary probability. As a result, we have that in scenarios where data has been collected using any robust sampling, we may release data which both ensures strong utility guarantees and satisfies very strong notions of privacy (i.e., zero-knowledge privacy and differential privacy). In particular, this methodology can allow us to achieve zero-knowledge privacy and differential privacy while guaranteeing utility that is better than that of previous methods (such as for releasing histograms or synthetic data points as described above). Our main theorems can be found in Section 3.5.

It is worthwhile to note that the particular mechanism considered in [51] (which in fact is a particular mechanism for achieving k-anonymity) can easily be shown to satisfy crowd-blending privacy; as a result, their main result can be derived (and significantly strengthened) as a corollary of our main theorem.[1] (See Section 3.3.1 and 3.5 for more details.)

[1] As mentioned, none of the earlier work using random pre-sampling focus on the case when the sampling is biased; furthermore, even for the case of perfect random sampling, the authors of [51] were not able to provide a closed form expression of the level of differential privacy achieved by their mechanism, whereas a closed form expression can be directly obtained by applying our main theorem.

3.2 Preliminaries and Existing Privacy Definitions

A *database* is a finite *multiset* of data values, where a data value is simply an element of some fixed set X, which we refer to as the *data universe*. Each data value in a database belongs to an individual, so we also refer to a data value in a database as an *individual* in the database. For convenience, we will sometimes order the individuals in a database in an arbitrary way and think of the database as an element of X^*, i.e., a vector with components in X (the components are referred to as the *rows* of the database). Given a database D and a data value $v \in X$, let (D, v) denote the database $D \uplus \{v\}$. A (database) *mechanism* is simply an algorithm that operates on databases.

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 (see [22, 19]) can now be defined in the following manner:

Definition 27 ([22, 19])**.** A mechanism San is said to be **ϵ-differentially private** if for every pair of databases D and D' differing in only one data value, we have $San(D) \approx_{\epsilon} San(D')$.

There are two definitions in the literature for "a pair of databases D and D' differing in only one data value", leading to two slightly different definitions of

differential privacy. In one definition, it is required that D contains D' and has exactly one more data value than D'. In the other definition, it is required that $|D| = |D'|$, $|D \setminus D'| = 1$, and $|D' \setminus D| = 1$. Intuitively, differential privacy protects the privacy of an individual t by requiring the output distribution of the mechanism to be essentially the same regardless of whether individual t's data is included in the database or not (or regardless of what data value individual t has).

We now begin describing zero-knowledge privacy, which is a privacy definition introduced in [32] that is strictly stronger than differential privacy. In the definition of zero-knowledge privacy, *adversaries* and *simulators* are simply randomized algorithms that play certain roles in the definition. Let San be any mechanism. For any database D, any adversary A, and any auxiliary information $z \in \mathcal{B}^*$, let $Out_A(A(z) \leftrightarrow San(D))$ denote the output of A on input z after interacting with the mechanism San operating on the database D. San can be interactive or non-interactive. If San is non-interactive, then $San(D)$ simply sends its output (e.g., sanitized data) to A and then halts immediately.

Let agg be any class of randomized algorithms. agg is normally a class of randomized aggregation functions that provide aggregate information to simulators, as described in the introduction.

Definition 28 ([32])**.** A mechanism San is said to be (ϵ, δ)**-zero-knowledge private with respect to** agg if there exists a $T \in agg$ such that for every adversary A, there exists a simulator S such that for every database D, every individual $t \in D$, and every auxiliary information $z \in \mathcal{B}^*$, we have

$$Out_A(A(z) \leftrightarrow San(D)) \approx_{\epsilon,\delta} S(z, T(D \setminus \{t\}), |D|).$$

Intuitively, zero-knowledge privacy requires that whatever an adversary can

compute about individual t by accessing (i.e., interacting with) the mechanism can also be essentially computed without accessing the mechanism but with certain aggregate information about the remaining individuals; this aggregate information is provided by an algorithm in agg. The adversary in the latter scenario is represented by the simulator S. This ensures that the adversary essentially does not learn any additional information about individual t beyond the aggregate information provided by an algorithm in agg on the remaining individuals.

agg is normally some class of randomized aggregation functions, such as the class of all functions T that draws r random samples from the input database and performs any computation (e.g., computes the average or simply outputs the samples) on the r random samples (note that in the definition, T is applied to $D \setminus \{t\}$ instead of D so that the aggregate information from T does not depend directly on individual t's data). Zero-knowledge privacy with respect to this class of aggregation functions ensures that an adversary essentially does not learn anything more about an individual beyond some "r random sample aggregate information" of the other individuals. One can also consider zero-knowledge privacy with respect to other classes of aggregation functions, such as the class of (randomized) functions that first sample each row of the input database with probability p (or in between p and p') independently and then performs any computation on the samples. We will actually use such classes of aggregation functions when we prove our main theorems later. It can be easily shown that zero-knowledge privacy (with respect to any class agg) implies differential privacy (see [32]).

In the original definition of zero-knowledge privacy in [32], T operates on $(D \setminus \{t\}, \perp)$ instead of $D \setminus \{t\}$, where $\perp$ is any arbitrary element of the data universe X. The main point is that the database that T is applied to does not include individual

t's data value (otherwise, a lot of information about individual t could possibly be leaked). Thus, using $T(D \setminus \{t\})$ in the definition also makes sense, and we choose to use $T(D \setminus \{t\})$ in this chapter for convenience.[2] This version of zero-knowledge privacy still implies differential privacy (essentially the same "hybrid/transitivity" proof from [32] works).

3.3 Crowd-Blending Privacy – A New Privacy Definition

We now begin to formally define our new privacy definition. Given $t, t' \in X$, $\epsilon \geq 0$, and a mechanism San, we say that t and t' are *ϵ-indistinguishable* by San, denoted $t \approx_{\epsilon,San} t'$, if $San(D,t) \approx_\epsilon San(D,t')$ for every database D. Intuitively, t and t' are indistinguishable by San if for any database containing t, we can replace the t by t' and the output distribution of San remains essentially the same. Usually, t and t' are the data values of two individuals, and if t and t' are indistinguishable by San, then this roughly means that San cannot distinguish these two individuals regardless of who else is in the database. If t and t' are ϵ-indistinguishable by San, we also loosely say that t *blends* with t' (with respect to San). We now describe what it means for an individual to blend in a crowd of people in the database (with respect to a mechanism).

Definition 29. Let D be any database. An individual $t \in D$ **ϵ-blends in a crowd of k people in D with respect to the mechanism** San if $|\{t' \in D : t' \approx_{\epsilon,San} t\}| \geq k$.

In the above definition, $\{t' \in D : t' \approx_{\epsilon,San} t\}$ should be regarded as a multiset.

[2]Even if we used $T(D \setminus \{t\}, \perp)$ instead of $T(D \setminus \{t\})$, our results would still hold with only minor modifications and slight differences in privacy parameters. Recall that differential privacy also has two versions of its definition.

When the mechanism San is clear from context, we shall simply omit the "with respect to the mechanism San". Intuitively, an individual $t \in D$ blends in a crowd of k people in D if t is indistinguishable by San from at least $k-1$ other individuals in D. Note that by the definition of two individuals being indistinguishable by San, $t \in D$ must be indistinguishable by San from each of these $k-1$ other individuals *regardless of what the database is*, as opposed to only when the database is D. (A weaker requirement would be that for each of these $k-1$ other individuals t', t and t' only need to be "indistinguishable by San with respect to D", i.e., if we take D and replace t by t' or vice versa, the output distributions of San on D and the modified D are essentially the same; we leave investigating this and other possible weaker requirements for future work.) We are now ready to state our new privacy definition.

Definition 30 (Crowd-blending privacy)**.** A mechanism San is (k, ϵ)**-crowd-blending private** if for every database D and every individual $t \in D$, either t ϵ-blends in a crowd of k people in D, or $San(D) \approx_\epsilon San(D \setminus \{t\})$ (or both).

Crowd-blending privacy requires that for every individual t in the database, either t blends in a crowd of k people in the database, or the mechanism essentially ignores individual t's data (the latter case is captured by $San(D) \approx_\epsilon San(D \setminus \{t\})$ in the definition). When an individual t blends in a crowd of k people in the database, the mechanism essentially does not release any information about individual t beyond the general characteristics of the crowd of k people. This is because the mechanism cannot distinguish individual t from the people in the crowd of k people, i.e., individual t's data can be changed to the data of another person in the crowd of k people and the output distribution of the mechanism remains essentially the same. A consequence is that the mechanism does not release any personally identifying information about individual t.

As mentioned in the introduction, crowd-blending privacy is not closed under composition (we later give an example in Section 3.3.2); however, we note that the privacy guarantee of blending in a crowd of k people in the database (described above) holds regardless of the amount of auxiliary information the adversary has (i.e., the definition is agnostic to the adversary's auxiliary information). Additionally, as mentioned previously, we show in Section 3.5 that when crowd-blending privacy is combined with "robust pre-sampling", we get zero-knowledge privacy and thus differential privacy as well, both of which satisfy composition in a natural way. Thus, as long as robust sampling is used during data collection before running a crowd-blending private mechanism on the collected data, independent releases from crowd-blending private mechanisms do compose and satisfy zero-knowledge privacy and differential privacy. (We also mention that one can compose a crowd-blending private mechanism with a differentially private mechanism to obtain a crowd-blending private mechanism; see Section 3.3.2 for details.)

Relationship with Differential Privacy. Differential privacy implies crowd-blending privacy.

Proposition 31 (Differential privacy $\implies$ Crowd-blending privacy)**.** *Let San be any ϵ-differentially private mechanism. Then, San is (k, ϵ)-crowd-blending private for every integer $k \geq 1$.*

Proof. This immediately follows from the two privacy definitions. □

(k, ϵ)-crowd-blending privacy for some integer k does not imply differential privacy in general; this will be clear from the examples of crowd-blending private mechanisms that we give later. Crowd-blending privacy requires that for every

database D and every individual $t \in D$, at least one of two conditions hold. The second condition $San(D) \approx_\epsilon San(D \setminus \{t\})$ is similar to the condition required in differential privacy. Thus, we can view crowd-blending privacy as a relaxation of differential privacy. If we remove the first condition "t ϵ-blends in a crowd of k people in D" from crowd-blending privacy, we clearly get the same definition as differential privacy. If we remove the second condition instead, it turns out that we also get differential privacy. (When we remove the second condition $San(D) \approx_\epsilon San(D \setminus \{t\})$, we also change the definition to only consider databases of size at least k, since otherwise it would be impossible for individual t to blend in a crowd of k people in the database.)

Proposition 32 (Removing the condition $San(D) \approx_\epsilon San(D \setminus \{t\})$ in crowd-blending privacy results in differential privacy)**.** *Let San be any mechanism, let $\epsilon \geq 0$, and let k be any integer ≥ 2. Then, San is ϵ-differentially private*[3] *if and only if San satisfies the property that for every database D of size at least k and every individual $t \in D$, t ϵ-blends in a crowd of k people in D with respect to San.*

Proof. If *San* is ϵ-differentially private, then for every database D of size at least k and every individual $t \in D$, t is ϵ-indistinguishable by *San* from every individual in D, so t ϵ-blends in a crowd of k people in D.

Now, suppose *San* is not ϵ-differentially private. Then, there exist a database D and a pair of data values $t, t' \in X$ such that $San(D, t) \not\approx_\epsilon San(D, t')$. Now, consider a database D' consisting of an individual with data value t and $k-1$ individuals with data value t'. Since $San(D, t) \not\approx_\epsilon San(D, t')$, t and t' are not

[3]Here, we are using the version of differential privacy that considers a pair of databases of equal size.

ϵ-indistinguishable by San, so the individual $t \in D'$ does not ϵ-blend in a crowd of k people in D'. □

3.3.1 Examples of Crowd-Blending Private Mechanisms

Given a partition P of the data universe X, and given a database D, one can compute the histogram with respect to the partition P using the database D; the histogram specifies for each block of the partition (which we refer to as a "bin") the number of individuals in D that belong to the block (which we refer to as the "count" of the bin). We first give an example of a crowd-blending private mechanism that computes a histogram and suppresses (i.e., sets to 0) bin counts that are considered too small.

Example 9 (Histogram with suppression of small counts)**.** Let P be any partition of X. Fix $k \in \mathbb{Z}_{\geq 0}$. Let San be a mechanism that, on input a database D, computes the histogram with respect to the partition P using the database D, suppresses each bin count that is $< k$ (by setting the count to 0), and then releases the resulting histogram.

Then, San is $(k, 0)$-crowd-blending private. To see this, we note that an individual t in a database D is 0-indistinguishable by San from all the individuals in D that belong to the same bin as t. If there are at least k such people, then individual t blends with k people in D; otherwise, we have $San(D) \approx_0 San(D \setminus \{t\})$ since San suppresses each bin count that is $< k$.

It is easy to see that it is impossible to release the exact count of a bin while satisfying differential privacy. Thus, crowd-blending privacy is indeed weaker than differential privacy. For crowd-blending privacy, we can actually get better utility

by adding a bit of noise to bins with low counts instead of completely suppressing them.

Example 10 (Histogram with noise for small counts and no noise for large counts)**.** Let P be any partition of X. Fix $\epsilon > 0$ and $k \in \mathbb{Z}_{\geq 0}$. Let San be a mechanism that, on input a database D, computes the histogram with respect to the partition P using the database D. Then, San replaces each bin count $i < k$ with $A(i)$, where A is any (randomized) algorithm that satisfies $A(j) \approx_\epsilon A(j-1)$ for every $0 < j < k$ ($A(i)$ is normally a noisy version of i). San then releases the noisy histogram.

Then, San is (k, ϵ)-crowd-blending private. To see this, we note that an individual t in a database D is ϵ-indistinguishable (in fact, 0-indistinguishable) by San from all the individuals in D that belong to the same bin as t. If there are at least k such people, then individual t blends with k people in D, as required. If not, then we have $San(D) \approx_\epsilon San(D \setminus \{t\})$, since the histogram when using the database D is the same as the histogram when using the database $D \setminus \{t\}$ except for individual t's bin, which differs by one; however, San replaces the count i for individual t's bin with $A(i)$, and the algorithm A satisfies $A(i) \approx_\epsilon A(i-1)$, so $San(D) \approx_\epsilon San(D \setminus \{t\})$, as required.

We can choose the algorithm A to be $A(j) = j + Lap(\frac{1}{\epsilon})$, where $Lap(\lambda)$ is (a random variable with) the Laplace distribution with probability density function $f_\lambda(x) = \frac{1}{2\lambda} e^{|x|/\lambda}$. The proof that $A(j) \approx_\epsilon A(j-1)$ for every $0 < j < k$ is simple and can be implicitly found in [22].

The differentially private mechanism in [22] for computing histograms has to add noise to every bin, while our mechanism here only adds noise to the bins that have a count that is $< k$.

Example 11 (Sanitizing a database by generalizing records safely)**.** Many mechanisms for achieving k-anonymity involve "generalizing" the records in the input table by replacing specific values with more general values, such as replacing a specific age with an age range. If this is not done carefully, the privacy of individuals can be breached, as shown by many attacks in the past (e.g., see [78, 80]). Most of these mechanisms do not satisfy crowd-blending privacy. However, if the generalization of records is done carefully, achieving crowd-blending privacy may be possible.

One example is the mechanism of [51]: Let Y be any set, and let $f : X \to Y$ be any function. We think of Y as a set of possible "generalized records", and f is a function that maps a record to its generalized version. Let San be a mechanism that, on input a database D, applies the function f to each individual in D; let $f(D)$ be the multi-set of images in Y. San then removes each record in $f(D)$ that appears fewer than k times in $f(D)$, and then outputs the result. It is easy to see that San is $(k, 0)$-crowd-blending private. To see this, we note that an individual t in a database D is 0-indistinguishable by San from all the individuals in D that also get mapped to $f(t)$. If there are at least k such people, then individual t blends with k people in D; otherwise, we have $San(D) \approx_0 San(D \setminus \{t\})$ since San removes each record in $f(D)$ that appears fewer than k times in $f(D)$.

3.3.2 Discussion of Composition

Unfortunately, crowd-blending private mechanisms do not necessarily compose, as we now show:

Proposition 33. *Let $X = \{1, 2, 3\}$ be the data universe, and let $k \in \mathbb{Z}^+$ and*

$\epsilon > 0$. *Let* San_1 *and* San_2 *be the histogram mechanism in the "Histogram with noise for small counts and no noise for large counts" example with partitions* $P_1 = \{\{1,2\},\{3\}\}$ *and* $P_2 = \{\{1\},\{2,3\}\}$*, respectively. As shown in the example,* San_1 *and* San_2 *are both* (k,ϵ)*-crowd-blending private.*

Let San be the composition of San_1 *and* San_2*, i.e.,* $San(D) = (San_1(D), San_2(D))$ *for every database* D*. Then, for every* $k' > 1$ *and every* $\epsilon' \geq 0$*, San is not* (k',ϵ')*-crowd-blending private.*

Proof. Fix $k' > 1$ and $\epsilon' \geq 0$. Let D be the database containing exactly $k-1$ individuals with data value 1, exactly 1 individual with data value 2, and exactly $k-1$ individuals with data value 3. Let t be the individual in D with data value 2.

We claim that individual t is not ϵ'-indistinguishable by San from any individual in D other than himself/herself. To see this, we note that if t changes his/her data value to 1, then the number of individuals in the database that belong to the block $\{2,3\}$ of the partition P_2 decreases from k to $k-1$; since San_2 adds noise to counts that are $< k$ but does not add noise to counts that are $\geq k$, the output distribution of San_2 changes completely and $San(D) \approx_{\epsilon'} San(D \setminus \{t\}, 1)$ clearly does not hold. If t changes his/her data value to 3, then the number of individuals in the database that belong to the block $\{1,2\}$ of the partition P_1 decreases from k to $k-1$; since San_1 adds noise to counts that are $< k$ but does not add noise to counts that are $\geq k$, the output distribution of San_1 changes completely and $San(D) \approx_{\epsilon'} San(D \setminus \{t\}, 3)$ clearly does not hold. Thus, individual t is not ϵ'-indistinguishable by San from any individual in D other than himself/herself, so individual t does not ϵ'-blend in a crowd of k' people in the database D.

We now claim that $San(D) \not\approx_{\epsilon'} San(D \setminus \{t\})$. To see this, we note that when the database is D, San_1 does not add noise to the bin $\{1, 2\}$ of the histogram it computes, since the count of the bin is k. However, when the database is $D \setminus \{t\}$, San_1 does add noise to the bin $\{1, 2\}$, since the count of the bin is $k - 1$. Thus, $San(D) \not\approx_{\epsilon'} San(D \setminus \{t\})$ clearly does not hold.

It follows that San is not (k', ϵ')-crowd-blending private. □

Although crowd-blending private mechanisms do not necessarily compose, one can compose via concatenation a crowd-blending private mechanism with a differentially private mechanism to obtain a crowd-blending private mechanism.

Proposition 34. *Let San_1 be any (k, ϵ_1)-crowd-blending private mechanism, and let San_2 be any ϵ_2-differentially private mechanism. Then, the mechanism $San(D) = (San_1(D), San_2(D))$ is $(k, \epsilon_1 + 2\epsilon_2)$-crowd-blending private.*

Proof. Let D be any database and t be any individual in D. Since San_1 is (k, ϵ_1)-crowd-blending private, either t ϵ_1-blends in a crowd of k people in D with respect to San_1, or $San_1(D) \approx_{\epsilon_1} San_1(D \setminus \{t\})$.

In the former case, we have $|t' \in D : t' \approx_{\epsilon_1, San_1} t| \geq k$; now, we note that if $t' \in D$ satisfies $t' \approx_{\epsilon, San_1} t$, then t' also satisfies $t' \approx_{\epsilon_1 + 2\epsilon_2, San} t$ since for every database D', we have

$$San(D', t') = (San_1(D', t'), San_2(D', t')) \approx_{\epsilon_1 + 2\epsilon_2} (San_1(D', t), San_2(D', t)) = San(D', t).$$

(The factor of 2 in $2\epsilon_2$ appears when we use a "hybrid/transitivity" argument: Since San_2 is ϵ_2-differentially private, we have $San_2(D', t') \approx_{\epsilon_2} San_2(D') \approx_{\epsilon_2} San(D', t)$, so $San_2(D', t') \approx_{2\epsilon_2} San(D', t)$.) Thus, individual t $(\epsilon_1 + 2\epsilon_2)$-blends in a crowd of k people in D with respect to San, as required.

In the latter case, we have $San_1(D) \approx_{\epsilon_1} San_1(D \setminus \{t\})$, so

$$San(D) = (San_1(D), San_2(D)) \approx_{\epsilon_1+\epsilon_2} (San_1(D \setminus \{t\}), San_2(D \setminus \{t\})) = San(D \setminus \{t\}),$$

as required. □

3.4 Privately Releasing Synthetic Data Points in $\mathbb{R}^d$ for Computing Smooth Functions

Roughly speaking, a function $g : (\mathbb{R}^d)^* \to \mathbb{R}^m$ is smooth if the value of g does not change much when we perturb the data points of the input slightly. In this section, we show that we can *efficiently* release synthetic data points in $\mathbb{R}^d$ for approximating *all* smooth functions simultaneously while satisfying crowd-blending privacy. On the other hand, we show that there are smooth functions that cannot even be approximated with non-trivial utility from synthetic data that has been released with differential privacy (even if the differentially private mechanism is *inefficient*).

In this section, the data universe X is any bounded subset of $\mathbb{R}^d$ for some positive integer d, and the input databases of mechanisms are elements of X^*. We consider mechanisms that always output a synthetic database where each row is a data point in $\mathbb{R}^d$. We loosely use the term "synthetic data/database" to mean that the data/database was outputted by a mechanism but still has the same format as the original input data/database. Given a database/vector D, let D_i denote the i^{th} row/component of D. We now state the definition of smoothness of a function $g : (\mathbb{R}^d)^* \to \mathbb{R}^m$.

Definition 35. Let $M : \mathbb{Z}_{\geq 0} \to \mathbb{R}^+$ and $K : \mathbb{Z}_{\geq 0} \to \mathbb{R}^+$ be functions. A function

$g : (\mathbb{R}^d)^* \to \mathbb{R}^m$ is said to be $(M(\cdot), K(\cdot))$**-smooth** if for every pair of databases $D, D' \in X^*$ of equal size n such that $||D_i - D'_i||_1 \leq M(n)$ for every $i \in [n]$, we have $||g(D) - g(D')||_1 \leq K(n)$.

Roughly speaking, a function is $(M(\cdot), K(\cdot))$-smooth if the value of the function changes by at most a distance of $K(n)$ when the data points in a database of size n are perturbed by at most a distance of $M(n)$. For example, the function that computes the mean of the data points is $(M(\cdot), M(\cdot))$-smooth for every function $M : \mathbb{Z}_{\geq 0} \to \mathbb{R}^+$. In practice, *outliers* are often removed before computing certain statistics on the data points, since outliers often cause the statistics to be less meaningful. Thus, when we consider the utility of a mechanism, we will consider how well the synthetic database released by the mechanism can be used to accurately approximate smooth functions with an outlier removal preprocessing step.

We now discuss how we decide whether a data point is an outlier or not. For the rest of the section, we fix a bounded data universe $X \subseteq \mathbb{R}^d$, a partition P of X, and an integer $k \geq 1$. Given a database D, an individual t in D is said to be an *outlier in* D (with respect to the partition P and the threshold k) if the block of P containing t contains fewer than k data points from D. We now describe what it means for a mechanism to be useful for a class of functions with outlier removal preprocessing.

Definition 36. Let *San* be any mechanism that always outputs a database whose rows are data points in $\mathbb{R}^d$. Let $\mathcal{C}$ be any class of functions of the form $g : (\mathbb{R}^d)^* \to \mathbb{R}^m$. *San* is said to be $(\alpha(\cdot), \beta(\cdot))$**-useful for** $\mathcal{C}$ **with outlier removal preprocessing** if for every database $D \in X^*$, if we let $\widehat{D}$ be the database D with all outliers removed and $\widehat{n} = |\widehat{D}|$, then with probability at least $1 - \beta(\widehat{n})$,

$San(D)$ outputs a synthetic database $\widetilde{D}$ such that

$$||g(\widetilde{D}) - g(\widehat{D})||_1 \leq \alpha(\widehat{n}) \quad \text{for every } g \in \mathcal{C}.$$

We now give an example of a crowd-blending private mechanism that releases synthetic data points in $\mathbb{R}^d$ for approximating all smooth functions with outlier removal preprocessing. Given a subset $A \subseteq \mathbb{R}^d$, let the diameter of A, denoted $diam(A)$, be defined by $diam(A) = \sup_{x,y \in A} ||x - y||_1$.

Example 12 (Releasing noisy data points in $\mathbb{R}^d$ for approximating all smooth functions with outlier removal preprocessing)**.** Let $\epsilon > 0$. Let San be a mechanism that, on input a database D, looks at each data point $\vec{x}$ in D and does the following: If $\vec{x}$ is an outlier in D, San simply deletes $\vec{x}$. Otherwise, San replaces $\vec{x}$ with $A_B(\vec{x})$, where B is the block of the partition P that contains $\vec{x}$, and A_B is any (randomized) algorithm that satisfies $A_B(\vec{y}) \approx_\epsilon A_B(\vec{z})$ for every pair of vectors $\vec{y}, \vec{z} \in B$ ($A_B(\vec{x})$ is normally a noisy version of $\vec{x}$). San then releases all the noisy data points.

Then, San is (k, ϵ)-crowd-blending private. To see this, let D be any database and let t be any individual in D. If t is an outlier in D, then we have $San(D) = San(D \backslash \{t\})$, since San simply deletes all outliers and the removal of t from D does not change whether the other individuals are outliers or not; thus, San is (k, ϵ)-crowd-blending private, as required. Thus, we now assume t is not an outlier in D. Then, let B be the block of the partition P that contains t. We note that individual t is ϵ-indistinguishable by San from each individual $t' \in D$ in the block B, since for every database D', we have $San(D', t) \approx_\epsilon San(D', t')$ since $A_B(t) \approx_\epsilon A_B(t')$. Since t is not an outlier in D, there are at least k people in D that belong to the block B, so t ϵ-blends in a crowd of k people in D, as required.

For each block B of the partition P, we can choose the algorithm A_B to be

$A_B(\vec{y}) = \vec{y} + Lap(\frac{diam(B)}{\epsilon})^d$, where $Lap(\frac{diam(B)}{\epsilon})^d$ is a random vector with d components, each of which is independently distributed as $Lap(\frac{diam(B)}{\epsilon})$. Using techniques/results found in [22], it is easy to show that $A_B(\vec{y}) \approx_\epsilon A_B(\vec{z})$ for every pair of vectors $\vec{y}, \vec{z} \in B$.

Remark. Even though *San* essentially runs a differentially private mechanism within each block (that does not contain too few data points), it is not the case that the only information that remains for a block is the number of data points that belong to the block. This is because there can be many data points within a block, and if *San* adds Laplacian noise to each data point as above, the general distribution of data points and many statistics are preserved in expectation and would also be reasonably accurate with high probability. Outputting just the number of data points within a block does not tell us such distributional and statistical information. Thus, we do not get the same result if *San* simply outputs the number of data points within each block like a histogram.

We now show that the above crowd-blending private mechanism with $A_B(\vec{y}) = \vec{y} + Lap(\frac{diam(B)}{\epsilon})^d$ is useful for all smooth functions with outlier removal preprocessing.

Proposition 37. *Let $\epsilon > 0$ and $L > 0$, and let $M : \mathbb{Z}_{\geq 0} \to \mathbb{R}^+$ and $K : \mathbb{Z}_{\geq 0} \to \mathbb{R}^+$ be arbitrary functions. Suppose $diam(B) \leq L$ for every block B of the partition P. Let San be the mechanism in the above example with $A_B(\vec{y}) = \vec{y} + Lap(\frac{diam(B)}{\epsilon})^d$. Then, San is $(K(\cdot), \beta(\cdot))$-useful for the class $\mathcal{C}$ of all $(M(\cdot), K(\cdot))$-smooth functions with outlier removal preprocessing, where $\beta(\widehat{n}) = d\widehat{n}e^{-\frac{\epsilon M(\widehat{n})}{dL}}$.*

Proof. Let $D \in X^*$, let $\widehat{D}$ be the database D with all outliers removed, and let $\widehat{n} = |\widehat{D}|$. Let $\widetilde{D} = San(D)$. Since $San(D)$ simply removes all outliers in D, we

have $\widetilde{D} = San(\widehat{D})$. Now, we note that $|\widetilde{D}| = \widehat{n}$ and for every $i \in [\widehat{n}]$, we have $\widetilde{D}_i = \widehat{D}_i + Lap(\frac{diam(B_i)}{\epsilon})^d$, where B_i is the block of P that contains $\widehat{D}_i$. Let $\lambda = \frac{L}{\epsilon}$ so that $\frac{diam(B_i)}{\epsilon} \leq \lambda$ for every $i \in [\widehat{n}]$. From the p.d.f. or c.d.f. of $Lap(\lambda)$, it is easy to verify that for every $\delta \geq 0$, we have $\Pr_{X \sim Lap(\lambda)}[|X| \leq \delta] = 1 - e^{-\frac{\delta}{\lambda}}$, so $\Pr_{X \sim Lap(\lambda)^d}[||X||_1 \leq \delta] \geq 1 - de^{-\frac{\delta}{d\lambda}}$ by a union bound. Then, for every $i \in [\widehat{n}]$, we have

$$\begin{aligned}
\Pr\left[||\widetilde{D}_i - \widehat{D}_i||_1 \leq M(\widehat{n})\right] &= \Pr_{X \sim Lap(\frac{diam(B_i)}{\epsilon})^d}[||X||_1 \leq M(\widehat{n})] \\
&\geq \Pr_{X \sim Lap(\lambda)^d}[||X||_1 \leq M(\widehat{n})] \\
&\geq 1 - de^{-\frac{\epsilon M(\widehat{n})}{dL}}.
\end{aligned}$$

Then, by a union bound, with probability at least $1 - \widehat{n}de^{-\frac{\epsilon M(\widehat{n})}{dL}}$, we have $||\widetilde{D}_i - \widehat{D}_i||_1 \leq M(\widehat{n})$ for every $i \in [\widehat{n}]$. Then, with probability at least $1 - \widehat{n}de^{-\frac{\epsilon M(\widehat{n})}{dL}}$, we have $||g(\widetilde{D}) - g(\widehat{D})||_1 \leq K(\widehat{n})$ for every $(M(\cdot), K(\cdot))$-smooth function $g : (\mathbb{R}^d)^* \to \mathbb{R}^m$ by definition of $(M(\cdot), K(\cdot))$-smooth. Thus, San is $(K(\cdot), \beta(\cdot))$-useful for the class $\mathcal{C}$ of all $(M(\cdot), K(\cdot))$-smooth functions with outlier removal preprocessing. $\square$

We note that $\beta(\widehat{n}) = d\widehat{n}e^{-\frac{\epsilon M(\widehat{n})}{dL}}$ can be made to be negligible by choosing $M(\widehat{n}) = \Omega(\widehat{n}^\kappa)$ for any $\kappa > 0$. We also note that the mechanism in the proposition can clearly be implemented efficiently. We now show that there exist $(M(\cdot), K(\cdot))$-smooth functions that cannot be computed with non-trivial utility from synthetic data released by a differentially private mechanism, regardless of the running time of the mechanism.

Proposition 38. *Let $g : (\mathbb{R}^d)^* \to \mathbb{R}^d$ be the function defined by $g(D) = D_1$, which is clearly $(M(\cdot), M(\cdot))$-smooth for every function $M : \mathbb{Z}_{\geq 0} \to \mathbb{R}^+$. Let $\epsilon \geq 0$, and let San be any (possibly inefficient) ϵ-differentially private mechanism that always outputs a database where each row is a data point in $\mathbb{R}^d$. Then, for every $\delta > 0$,*

San is not even $(\frac{diam(X)}{4}, \frac{1}{1+e^\epsilon} - \delta)$*-useful for the function g with outlier removal preprocessing.*

Proof. Let $\vec{x}$ and $\vec{y}$ be any pair of data points in X such that $||\vec{x}-\vec{y}||_1 \geq \frac{3}{4}diam(X)$. Let D be the database consisting of exactly $k+1$ copies of $\vec{x}$ followed by exactly k copies of $\vec{y}$, and let D' be the same as D except that the first row is changed from $\vec{x}$ to $\vec{y}$. Then, both D and D' do not contain any outliers.

Let $\delta > 0$. To obtain a contradiction, suppose San is $(\frac{diam(X)}{4}, \frac{1}{1+e^\epsilon} - \delta)$-useful for the function g with outlier removal preprocessing. Then, San is also $(\frac{diam(X)}{4}, \frac{1}{1+e^\epsilon})$-useful for g with outlier removal preprocessing. Then, we have

$$\Pr[||San(D)_1 - \vec{x}||_1 \leq diam(X)/4] \geq 1 - \frac{1}{1+e^\epsilon} = \frac{e^\epsilon}{1+e^\epsilon}.$$

Since San is ϵ-differentially private and D and D' differ by only one row, we have $San(D) \approx_\epsilon San(D')$, so

$$\begin{aligned}\Pr[||San(D')_1 - \vec{x}||_1 \leq diam(X)/4] &\geq e^{-\epsilon} \cdot \Pr[||San(D)_1 - \vec{x}||_1 \leq diam(X)/4] \\ &\geq \frac{1}{1+e^\epsilon}.\end{aligned}$$

Since $||\vec{x} - \vec{y}||_1 \geq \frac{3}{4}diam(X)$, if $||San(D')_1 - \vec{x}||_1 \leq diam(X)/4$ holds, then $||San(D')_1 - \vec{y}||_1 \leq diam(X)/4$ does not hold. It follows that

$$\begin{aligned}\Pr[||g(San(D')) - g(D')||_1 \leq diam(X)/4] = &\Pr[||San(D')_1 - \vec{y}||_1 \leq diam(X)/4] \\ &\leq 1 - \Pr[||San(D')_1 - \vec{x}||_1 \leq diam(X)/4] \\ &\leq 1 - \frac{1}{1+e^\epsilon} \\ &< 1 - \left(\frac{1}{1+e^\epsilon} - \delta\right).\end{aligned}$$

This contradicts our assumption that San is $(\frac{diam(X)}{4}, \frac{1}{1+e^\epsilon} - \delta)$-useful for g with outlier removal preprocessing. □

In Proposition 38, we note that the image of X^* under g is X (recall that the databases that we run mechanisms on are elements of X^*) and $\frac{1}{1+e^\epsilon} \approx \frac{1}{2}$ when ϵ is small, and so requiring San to be $(\frac{diam(X)}{4}, \frac{1}{1+e^\epsilon} - \delta)$-useful for g is only requiring San to possibly provide non-trivial utility; however, the proposition says that San cannot even satisfy this non-triviality requirement. If we apply Proposition 37 to the same function g, we see that for every function $M : \mathbb{Z}_{\geq 0} \to \mathbb{R}^+$, the crowd-blending private mechanism is $(M(\cdot), \beta(\cdot))$-useful for g with outlier removal preprocessing, where $\beta(\widehat{n}) = d\widehat{n}e^{-\frac{\epsilon M(\widehat{n})}{dL}}$ and L is a bound on the diameter of every block of the partition P. The utility guarantee of this result is non-trivial in many situations. Thus, it is possible to release synthetic data points for approximating smooth functions while satisfying crowd-blending privacy, but doing this while satisfying differential privacy is impossible in general.

3.5 Our Main Theorem

In this section, we prove our main theorem that says that when we combine a crowd-blending private mechanism with a natural *pre-sampling* step, the combined algorithm is zero-knowledge private (and thus differentially private as well). The pre-sampling step should be thought of as being part of the data collection process, where individuals in some population are sampled and asked for their data. A crowd-blending private mechanism is then run on the samples to release useful information while preserving privacy.

We first prove our main theorem for the case where the pre-sampling step samples each individual in the population with probability p independently. In reality, the sampling performed during data collection may be slightly *biased* or done

slightly incorrectly, and an adversary may know whether certain individuals were sampled or not. Thus, we later extend our main theorem to the case where the sampling probability is not necessarily the same for everybody, but the sampling is still *robust* in the sense that most individuals are sampled independently with probability in between p and p' (this probability can even depend on the individual's data), where p and p' are relatively close to one another, while the remaining individuals are sampled independently with arbitrary probability.

We begin with some necessary terminology and notation. A *population* is a collection of *individuals*, where an individual is simply represented by a data value in the data universe X. Thus, a population is actually a multiset of data values, which is the same as a database. (If we want individuals to have unique data values, we can easily modify X to include personal/unique identifiers.) Given a population $\mathcal{P}$ and a real number $p \in [0, 1]$, let $Sam(\mathcal{P}, p)$ be the outcome of sampling each individual in $\mathcal{P}$ with probability p independently.

Although zero-knowledge privacy was originally defined for mechanisms operating on *databases*, one can also consider mechanisms operating on *populations*, since there is essentially no difference between the way we model populations and databases. (In the definition of zero-knowledge privacy, we simply change "database" to "population" and D to $\mathcal{P}$.) We now describe a class of (randomized) aggregation functions that we will use in the definition of zero-knowledge privacy.

- $iidRS(p)$ = i.i.d. random sampling with probability p : the class of algorithms T such that on input a population $\mathcal{P}$, T chooses each individual in $\mathcal{P}$ with probability p independently, and then performs any computation on the data

of the chosen individuals.[4]

We now state and prove the basic version of our main theorem.

Theorem 39 (Sampling + Crowd-Blending Privacy ⇒ Zero-Knowledge Privacy)**.** *Let San be any (k, ϵ)-crowd-blending private mechanism with $k \geq 2$, and let $p \in (0, 1)$. Then, the algorithm San_{zk} defined by $San_{zk}(\mathcal{P}) = San(Sam(\mathcal{P}, p))$ for any population $\mathcal{P}$ is $(\epsilon_{zk}, \delta_{zk})$-zero-knowledge private[5] with respect to $iidRS(p)$, where*

$$\epsilon_{zk} = \ln\left(p \cdot \left(\frac{2-p}{1-p}e^{\epsilon}\right) + (1-p)\right) \quad \text{and} \quad \delta_{zk} = e^{-\Omega(k\cdot(1-p)^2)}.$$

To prove Theorem 39, we will first prove two supporting lemmas. The first lemma essentially says that if an individual t blends with (i.e., is indistinguishable by *San* from) many people in the population, then t's privacy is protected when we sample from the population and run *San* on the samples:

Lemma 40 (Protection of individuals that blend with many people in the population)**.** *Let San be any mechanism, $\mathcal{P}$ be any population, $p \in (0, 1)$, and $\epsilon \geq 0$. Let t be any individual in $\mathcal{P}$, and let A be any non-empty subset of $\mathcal{P} \setminus \{t\}$ such that $t' \approx_{\epsilon, San} t$ for every individual $t' \in A$. Let $n = |A|$. Then, we have*

$$San(Sam(\mathcal{P}, p)) \approx_{\epsilon_{final}, \delta_{final}} San(Sam(\mathcal{P} \setminus \{t\}, p)),$$

where $\epsilon_{final} = \ln(p \cdot (\frac{2-p}{1-p}e^{\epsilon}) + (1-p))$ and $\delta_{final} = e^{-\Omega((n+1)p(1-p)^2)}$.

In the lemma, A is any non-empty set of individuals in $\mathcal{P} \setminus \{t\}$ that blend with individual t. (We could set A to be the set of *all* individuals in $\mathcal{P} \setminus \{t\}$ that blend

[4]To make zero-knowledge privacy compose naturally for this type of aggregate information, we can extend $iidRS(p)$ to $iidRS(p, r)$, where T is now allowed to perform r rounds of sampling before performing any computation on the sampled data. It is not hard to see that zero-knowledge privacy with respect to $iidRS(p, r)$ composes in a natural way.

[5]The constant hidden by the $\Omega(\cdot)$ in δ_{zk} can be easily computed; however, we did not try to optimize the constant in any way.

with individual t, but leaving A more general allows us to more easily extend the lemma to the case of "robust" sampling later.) We note that δ_{final} is smaller when $n = |A|$ is larger, i.e., when t blends with more people. Intuitively, if an individual t is indistinguishable by San from many other people in the population, then t's presence or absence in the population does not affect the output of $San(Sam(\cdot, p))$ much, since the people indistinguishable from t can essentially take the place of t in almost any situation (and the output of San would essentially be the same). Since it does not matter much whether individual t is in the population or not, it follows that t's privacy is protected.

The proof of the lemma *roughly* works as follows: Consider two scenarios, one where individual t is in the population (i.e., $San(Sam(\mathcal{P}, p))$ in the lemma), and one where individual t has been removed from the population (i.e., $San(Sam(\mathcal{P} \setminus \{t\}, p))$ in the lemma). Our goal is to show that the output of San is essentially the same in the two scenarios, i.e., $San(Sam(\mathcal{P}, p)) \approx_{\epsilon_{final}, \delta_{final}} San(Sam(\mathcal{P} \setminus \{t\}, p))$. Conditional on individual t not being sampled in the first scenario, the two scenarios are exactly the same, as desired. Thus, we now always condition on individual t being sampled in the first scenario. In the lemma, A is a set of individuals in the population (excluding t) that are indistinguishable from t by San. Let $\widetilde{m}$ denote the number of people in A that are sampled. The proof involves showing the following two properties:

1. $\widetilde{m}$ is relatively smooth near its expectation: For every integer m near the expectation of $\widetilde{m}$, $\Pr[\widetilde{m} = m]$ is relatively close to $\Pr[\widetilde{m} = m + 1]$.
2. For every integer $m \in \{0, \ldots, n-1\}$, the output of San in the first scenario conditioned on $\widetilde{m} = m$ (and t being sampled) is essentially the same as the output of San in the second scenario conditioned on $\widetilde{m} = m + 1$.

For the first property, we note that $\widetilde{m}$ follows a binomial distribution, which can be shown to be relatively smooth near its expectation. To show the second property, we note that when we condition on $\widetilde{m} = m$ (and t being sampled) in the first scenario, m random samples are drawn uniformly from A (one at a time) without replacement, and also $t \notin A$ is sampled for sure (and the remaining individuals are sampled independently with probability p). This is very similar to the second scenario conditioned on $\widetilde{m} = m + 1$, where $m + 1$ random samples are drawn uniformly from A without replacement, since if we replace the $(m + 1)^{th}$ sample by t, we get back the first scenario conditioned on $\widetilde{m} = m$ (and t being sampled). Since the $(m+1)^{th}$ sample is indistinguishable from t by San, the output of San is essentially the same in both scenarios.

Using the two properties above, one can show that when $\widetilde{m}$ is close to its expectation, the output of San is essentially the same in both scenarios. δ_{final} in the lemma captures the probability of the bad event where $\widetilde{m}$ is not close to its expectation, which we bound by essentially using a Chernoff bound. We now give the formal proof of Lemma 40.

Proof of Lemma 40. Let $\epsilon_{Sam} > 0$, $\widehat{D} = Sam(\mathcal{P}, p)$, $\widetilde{D} = Sam(\mathcal{P} \setminus \{t\}, p)$, $\widetilde{m} = |\widetilde{D} \cap A|$, and $Y \subseteq \{0,1\}^*$. Let E be the event that t is sampled when $\widehat{D}$ is chosen. We first observe that

$$\begin{aligned}\Pr[San(\widehat{D}) \in Y] &= \Pr[San(\widehat{D}) \in Y \mid E] \cdot \Pr[E] + \Pr[San(\widehat{D}) \in Y \mid \overline{E}] \cdot \Pr[\overline{E}] \\ &= \Pr[San(\widetilde{D} \cup \{t\}) \in Y] \cdot p + \Pr[San(\widetilde{D}) \in Y] \cdot (1-p). \qquad (1)\end{aligned}$$

We will now show that for every $m \in \{0, \ldots, n-1\}$, we have

$$\left| \ln \left(\frac{\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} = m]}{\Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} = m+1]} \right) \right| \leq \epsilon. \qquad (2)$$

Fix $m \in \{0, \ldots, n-1\}$. Let $\mathcal{P}_{-t,-A} = (\mathcal{P} \setminus \{t\}) \setminus A$. We note that for $j \in \{0, \ldots, n\}$, the conditional distribution of $\widetilde{D}$ given $\widetilde{m} = j$ is equal to $Sam(\mathcal{P}_{-t,-A}, p) \cup A_j$, where A_j is the outcome of choosing j random samples uniformly without replacement from A. Then, using the fact that $t' \approx_{\epsilon,San} t$ for every individual $t' \in A$, we have

$$\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} = m] = \Pr[San(Sam(\mathcal{P}_{-t,-A}, p) \cup A_m \cup \{t\}) \in Y]$$
$$\leq e^{\epsilon} \Pr[San(Sam(\mathcal{P}_{-t,-A}, p) \cup A_{m+1}) \in Y] = e^{\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} = m+1].$$

Similarly, we also have

$$\Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} = m+1] = \Pr[San(Sam(\mathcal{P}_{-t,-A}, p) \cup A_{m+1}) \in Y]$$
$$\leq e^{\epsilon} \Pr[San(Sam(\mathcal{P}_{-t,-A}, p) \cup A_m \cup \{t\}) \in Y] = e^{\epsilon} \Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} = m].$$

Thus, we have shown (2).

Now, we observe that for every $m \in \{0, \ldots, n-1\}$, if $m+1 \leq (n+1)p \cdot \frac{e^{\epsilon_{Sam}}}{pe^{\epsilon_{Sam}} + (1-p)}$, then

$$\frac{\Pr[\widetilde{m} = m]}{\Pr[\widetilde{m} = m+1]} = \frac{\binom{n}{m} p^m (1-p)^{n-m}}{\binom{n}{m+1} p^{m+1} (1-p)^{n-(m+1)}} = \frac{m+1}{n-m} \frac{1-p}{p} \leq e^{\epsilon_{Sam}}. \quad (3)$$

Let $\alpha = \frac{e^{\epsilon_{Sam}}}{pe^{\epsilon_{Sam}} + (1-p)}$ and $\delta_{Sam} = \Pr[\widetilde{m} + 1 > (n+1)p \cdot \alpha]$. Now, using (3) and (2) (and the fact that $m = n$ does not satisfy $m+1 \leq (n+1)p \cdot \alpha$), we have

$$\Pr[San(\widetilde{D} \cup \{t\}) \in Y]$$
$$\leq \sum_{\substack{m \in \{0,\ldots,n\} \\ m+1 \leq (n+1)p \cdot \alpha}} \Pr[\widetilde{m} = m] \Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} = m] + \Pr[\widetilde{m} + 1 > (n+1)p \cdot \alpha]$$
$$\leq \sum_{\substack{m \in \{0,\ldots,n\} \\ m+1 \leq (n+1)p \cdot \alpha}} e^{\epsilon_{Sam}} \Pr[\widetilde{m} = m+1] \cdot e^{\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} = m+1] + \delta_{Sam}$$
$$\leq e^{\epsilon + \epsilon_{Sam}} \Pr[San(\widetilde{D}) \in Y] + \delta_{Sam}. \quad (4)$$

Let $\epsilon_{total} = \max\{\ln(pe^{\epsilon+\epsilon_{Sam}} + (1-p)), \ln(\frac{1}{1-p})\}$. Combining (1) and (4), we have

$$\begin{aligned}\Pr[San(\widehat{D}) \in Y] &\leq (e^{\epsilon+\epsilon_{Sam}} \Pr[San(\widetilde{D}) \in Y] + \delta_{Sam}) \cdot p + \Pr[San(\widetilde{D}) \in Y] \cdot (1-p) \\ &\leq e^{\epsilon_{total}} \Pr[San(\widetilde{D}) \in Y] + p \cdot \delta_{Sam}.\end{aligned}$$

By (1), we also have

$$\Pr[San(\widehat{D}) \in Y] \geq (1-p) \Pr[San(\widetilde{D}) \in Y] \geq e^{-\epsilon_{total}} \Pr[San(\widetilde{D}) \in Y].$$

Thus, we have $San(\widehat{D}) \approx_{\epsilon_{total}, p\cdot\delta_{Sam}} San(\widetilde{D})$.

Now, we set $\epsilon_{Sam} = \ln(\frac{2-p}{1-p})$. Then, we have

$$\begin{aligned}\epsilon_{total} &= \max\{\ln(pe^{\epsilon+\epsilon_{Sam}} + (1-p)), \ln(\frac{1}{1-p})\} \\ &= \max\{\ln(p \cdot (\frac{2-p}{1-p}e^{\epsilon}) + (1-p)), \ln(\frac{1}{1-p})\} \\ &= \ln(p \cdot (\frac{2-p}{1-p}e^{\epsilon}) + (1-p))\end{aligned}$$

and

$$\begin{aligned}p \cdot \delta_{Sam} &= p \cdot \Pr[\widetilde{m} + 1 > (n+1)p \cdot (2-p)] \\ &\leq \Pr[\widetilde{m} + Bin(1,p) > (n+1)p \cdot (2-p)] \\ &\leq e^{-\Omega((n+1)p(1-p)^2)},\end{aligned}$$

where $Bin(1,p)$ is a binomial random variable with 1 trial and success probability p, and the last inequality follows from a multiplicative Chernoff bound. □

We now show how pre-sampling combined with a crowd-blending private mechanism can protect the privacy of individuals who blend with (i.e., are indistinguishable by *San* from) few people in the population.

Lemma 41 (Protection of individuals that blend with few people in the population)**.** *Let San be any (k,ϵ)-crowd-blending private mechanism with $k \geq 2$, let*

$\mathcal{P}$ be any population, and let $p \in (0,1)$. Let t be any individual in $\mathcal{P}$, and let $n = |\{t' \in \mathcal{P} \setminus \{t\} : t' \approx_{\epsilon,San} t\}|$. Then, if $n \leq \frac{k-1}{p(2-p)}$, we have

$$San(Sam(\mathcal{P}, p)) \approx_{\epsilon_{final},\delta_{final}} San(Sam(\mathcal{P} \setminus \{t\}, p)),$$

where $\epsilon_{final} = \ln(pe^{\epsilon} + (1-p))$ and $\delta_{final} = pe^{-\Omega(k\cdot(1-p)^2)}$.

The proof of the lemma *roughly* works as follows: In the lemma, n is the number of people in the population that individual t blends with, and is assumed to be small. We will show that when we remove individual t from the population, the output of San does not change much.

Consider two scenarios, one where individual t is in the population, and one where individual t has been removed from the population. Conditional on individual t not being sampled in the first scenario, the two scenarios are exactly the same, as desired. Thus, we now always condition on individual t being sampled in the first scenario. Since individual t blends with few people in the population, we have that with very high probability, the database obtained from sampling from the population would contain fewer than k people that blend with individual t; since San is (k,ϵ)-crowd-blending private and individual t does not blend in a crowd of k people in the database, San must essentially ignore individual t's data; thus, the first scenario is essentially the same as the second scenario, since individual t's data is essentially ignored anyway. δ_{final} in the lemma captures the probability of the bad event where the database obtained from sampling actually contains k people that blend with individual t. We now give the formal proof of Lemma 41.

Proof of Lemma 41. Suppose $n \leq \frac{k-1}{p(2-p)}$. Let $\widehat{D} = Sam(\mathcal{P}, p)$, $\widetilde{D} = Sam(\mathcal{P} \setminus \{t\}, p)$, and $Y \subseteq \{0,1\}^*$. Let $A = \{t' \in \mathcal{P} \setminus \{t\} : t' \approx_{\epsilon,San} t\}$, so $n = |A|$. Let $\widetilde{m} = |\widetilde{D} \cap A|$, and let E be the event that individual t is in $\widehat{D}$ when $\widehat{D}$ is chosen.

We first note that

$$\begin{aligned}\Pr[San(\widehat{D}) \in Y] &= \Pr[San(\widehat{D}) \in Y \mid \overline{E}] \cdot \Pr[\overline{E}] + \Pr[San(\widehat{D}) \in Y \mid E] \cdot \Pr[E] \\ &= \Pr[San(\widetilde{D}) \in Y] \cdot (1-p) + \Pr[San(\widetilde{D} \cup \{t\}) \in Y] \cdot p. \end{aligned} \tag{1}$$

Since San is (k, ϵ)-crowd-blending private, we have

$$\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k-1] \leq e^{\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k-1]$$

and

$$\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k-1] \geq e^{-\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k-1].$$

Then, we have

$$\begin{aligned} & \Pr[San(\widetilde{D} \cup \{t\}) \in Y] \\ \leq\ & \Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k-1] \Pr[\widetilde{m} < k-1] + \Pr[\widetilde{m} \geq k-1] \\ \leq\ & e^{\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k-1] \Pr[\widetilde{m} < k-1] + \Pr[\widetilde{m} \geq k-1] \\ \leq\ & e^{\epsilon} \Pr[San(\widetilde{D}) \in Y] + \Pr[\widetilde{m} \geq k-1], \end{aligned} \tag{2}$$

and

$$\begin{aligned} \Pr[San(\widetilde{D} \cup \{t\}) \in Y] \geq\ & \Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k-1] \Pr[\widetilde{m} < k-1] \\ \geq\ & e^{-\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k-1] \Pr[\widetilde{m} < k-1] \\ \geq\ & e^{-\epsilon} (\Pr[San(\widetilde{D}) \in Y] - \Pr[\widetilde{m} \geq k-1]) \\ =\ & e^{-\epsilon} \Pr[San(\widetilde{D}) \in Y] - e^{-\epsilon} \Pr[\widetilde{m} \geq k-1]. \end{aligned} \tag{3}$$

Now, combining (1) and (2), we have

$$\Pr[San(\widehat{D}) \in Y] \leq (pe^{\epsilon} + (1-p)) \Pr[San(\widetilde{D}) \in Y] + p \Pr[\widetilde{m} \geq k-1]. \tag{4}$$

Also, combining (1) and (3), we have

$$\Pr[San(\widehat{D}) \in Y] \geq (pe^{-\epsilon} + (1-p)) \Pr[San(\widetilde{D}) \in Y] - e^{-\epsilon} p \Pr[\widetilde{m} \geq k-1].$$

Rearranging this inequality, we get

$$\begin{aligned}\Pr[San(\widetilde{D}) \in Y] &\leq \frac{1}{pe^{-\epsilon} + (1-p)} \Pr[San(\widehat{D}) \in Y] + \frac{e^{-\epsilon}}{pe^{-\epsilon} + (1-p)} p \Pr[\widetilde{m} \geq k-1] \\ &\leq (pe^{\epsilon} + (1-p)) \Pr[San(\widehat{D}) \in Y] + p \Pr[\widetilde{m} \geq k-1], \end{aligned} \tag{5}$$

where the last inequality follows from the fact that the function $f(x) = \frac{1}{x}$ is convex for $x > 0$, so $\frac{1}{pe^{-\epsilon}+(1-p)} \leq pe^{\epsilon} + (1-p)$.

Let $\tau = \frac{k-1}{p(2-p)}$. Then, we have $n \leq \tau$. The lemma now follows from (4), (5), and the inequality

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

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

We are now ready to prove Theorem 39. The proof roughly works as follows: By definition of $iidRS(p)$, a simulator in the definition of zero-knowledge privacy is able to obtain the aggregate information $Sam(\mathcal{P} \setminus \{t\}, p)$. With $Sam(\mathcal{P} \setminus \{t\}, p)$, the simulator can easily compute $San(Sam(\mathcal{P} \setminus \{t\}, p))$, which it can then use to simulate the computation of the given adversary. It is not hard to see that the simulation works if $San(Sam(\mathcal{P}, p)) \approx_{\epsilon_{zk}, \delta_{zk}} San(Sam(\mathcal{P} \setminus \{t\}, p))$ holds. Thus, consider any population $\mathcal{P}$ and any individual $t \in \mathcal{P}$. Recall that Lemma 40 protects the privacy of individuals that blend with many people in $\mathcal{P}$, while Lemma 41 protects the privacy of individuals that blend with few people in $\mathcal{P}$. Thus, if

individual t blends with many people in $\mathcal{P}$, we use Lemma 40; otherwise, we use Lemma 41. It then follows that $San(Sam(\mathcal{P}, p)) \approx_{\epsilon_{zk}, \delta_{zk}} San(Sam(\mathcal{P} \setminus \{t\}, p))$, as required. We now give the formal proof of Theorem 39.

Proof of Theorem 39. We first note that $Sam(\cdot, p) \in iidRS(p)$. Thus, we can let $T = Sam(\cdot, p)$ in the definition of zero-knowledge privacy with respect to $iidRS(p)$. Let A be any adversary. We will describe how to construct a simulator S for A. Let $\mathcal{P}$ be any population, t be any individual in $\mathcal{P}$, and $z \in \{0,1\}^*$. Since the simulator S is given $T(\mathcal{P} \setminus \{t\}) = Sam(\mathcal{P} \setminus \{t\}, p)$ and z as part of its input, S can easily compute $San_{zk}(\mathcal{P} \setminus \{t\}) = San(Sam(\mathcal{P} \setminus \{t\}, p))$ and then simulate the computation of the adversary A that is given $San_{zk}(\mathcal{P} \setminus \{t\})$ and the auxiliary information z; the simulator S then outputs whatever A outputs.

Now, we note that if $San_{zk}(\mathcal{P}) \approx_{\epsilon_{zk}, \delta_{zk}} San_{zk}(\mathcal{P} \setminus \{t\})$, then $Out_A(A(z) \leftrightarrow San_{zk}(\mathcal{P})) \approx_{\epsilon_{zk}, \delta_{zk}} S(z, T(\mathcal{P} \setminus \{t\}), |\mathcal{P}|)$. Thus, to show that San_{zk} is $(\epsilon_{zk}, \delta_{zk})$-zero-knowledge private with respect to $iidRS(p)$, it suffices to show that $San_{zk}(\mathcal{P}) \approx_{\epsilon_{zk}, \delta_{zk}} San_{zk}(\mathcal{P} \setminus \{t\})$, i.e.,

$$San(Sam(\mathcal{P}, p)) \approx_{\epsilon_{zk}, \delta_{zk}} San(Sam(\mathcal{P} \setminus \{t\}, p)).$$

To this end, let $A = \{t' \in \mathcal{P} \setminus \{t\} : t' \approx_{\epsilon, San} t\}$ and $n = |A|$. Let $\tau = \frac{k-1}{p(2-p)}$. We will consider two cases: $n > \tau$ and $n \leq \tau$.

Suppose $n > \tau$. By Lemma 40, we have

$$San(Sam(\mathcal{P}, p)) \approx_{\epsilon_1, \delta_1} San(Sam(\mathcal{P} \setminus \{t\}, p)),$$

where $\epsilon_1 = \ln(p \cdot (\frac{2-p}{1-p} e^{\epsilon}) + (1-p)) = \epsilon_{zk}$ and $\delta_1 = e^{-\Omega((n+1)p(1-p)^2)} \leq e^{-\Omega(k \cdot (1-p)^2)} = \delta_{zk}$.

Now, suppose $n \leq \tau$. By Lemma 41, we have

$$San(Sam(\mathcal{P}, p)) \approx_{\epsilon_2, \delta_2} San(Sam(\mathcal{P} \setminus \{t\}, p)),$$

where $\epsilon_2 = \ln(pe^{\epsilon} + (1-p)) \leq \epsilon_1 = \epsilon_{zk}$ and $\delta_2 = pe^{-\Omega(k \cdot (1-p)^2)} \leq \delta_{zk}$.

It follows that

$$San(Sam(\mathcal{P}, p)) \approx_{\epsilon_{zk}, \delta_{zk}} San(Sam(\mathcal{P} \setminus \{t\}, p)),$$

as required. □

3.5.1 Our Main Theorem Extended to Robust Sampling

We now extend our main theorem to the case where the sampling probability is not necessarily the same for everybody, but the sampling is still "robust" in the sense that most individuals are sampled independently with probability in between p and p' (this probability can even depend on the individual's data), where p and p' are relatively close to one another (i.e., $\frac{p'}{p}$ is not too large), while the remaining individuals are sampled independently with arbitrary probability.

We begin with some more notation. Given a population $\mathcal{P}$ and a function $\pi : X \to [0, 1]$, let $Sam(\mathcal{P}, \pi)$ be the outcome of sampling each individual t in $\mathcal{P}$ with probability $\pi(t)$ independently. We note that for $Sam(\mathcal{P}, \pi)$, two individuals in $\mathcal{P}$ with the same data value in X 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 value in X. Thus, for convenience, we now define a population to be a subset of the data universe X instead of being a multiset of data values in X. Then, each individual in a population would have a unique data value in X, so π does not have to assign

the same sampling probability to two different individuals. We now describe a class of aggregation functions that we will use in the definition of zero-knowledge privacy.

- $iRS(p, p', \ell)$ = independent random sampling with probability in between p and p' except for ℓ individuals: the class of algorithms T such that on input a population $\mathcal{P}$, T independently chooses each individual $t \in \mathcal{P}$ with some probability $p_t \in [0, 1]$ (possibly dependent on t's data), but all except for at most ℓ individuals in $\mathcal{P}$ must be chosen with probability in $\{0\} \cup [p, p']$; T then performs any computation on the chosen individuals' data.

We now state the extended version of our main theorem.

Theorem 42 (Robust Sampling + Crowd-Blending Privacy $\Rightarrow$ Zero-Knowledge Privacy)**.** *Let San be any (k, ϵ)-crowd-blending private mechanism with $k \geq 2$, let $0 < p \leq p' < 1$, let $\pi : X \to [0, 1]$ be any function, let $\ell = |\{x \in X : \pi(x) \notin \{0\} \cup [p, p']\}|$, and let $p_{\max} = \sup_{x \in X} \pi(x)$. Suppose $\ell < k - 1$.*

Then, the algorithm San_{zk} defined by $San_{zk}(\mathcal{P}) = San(Sam(\mathcal{P}, \pi))$ for any population $\mathcal{P}$ is $(\epsilon_{zk}, \delta_{zk})$-zero-knowledge private with respect to $iRS(p, p', \ell)$, where

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

In the theorem, ℓ represents the number of individuals that are sampled with probability outside of $\{0\} \cup [p, p']$. We prove the theorem by extending Lemmas 40 and 41 to the case of "robust" sampling. We first describe *some* of the main changes to the lemmas and their proofs, and then we give the formal proof of Theorem 42.

Let us first consider Lemma 40, which protects the privacy of individuals that blend with many people in the population. Like before, consider two scenarios, one where individual t is in the population, and one where individual t has been removed. Let $\widetilde{m}$ denote the number of people in A that are sampled (recall that A is a set of individuals that blend with individual t). Recall that in the proof of Lemma 40, we had to show two properties: (1) $\widetilde{m}$ is relatively smooth near its expectation, and (2) the output of San in the first scenario conditioned on $\widetilde{m} = m$ (and t being sampled) is essentially the same as the output of San in the second scenario conditioned on $\widetilde{m} = m + 1$.

For the first property, we used the fact that the binomial distribution is relatively smooth near its expectation. Here, since the sampling is no longer i.i.d. but is still robust, we need the Poisson binomial distribution (the sum of independent Bernoulli trials, where the success probabilities are not necessarily the same) to be relatively smooth near its expectation. This can be shown as long as the success probabilities are all relatively close to one another; this is ensured by changing the lemma so that everyone in the set A is required to have a sampling probability in $[p, p']$.

For the second property, we used the fact that when we condition on $\widetilde{m} = m+1$ in the second scenario, we are drawing $m + 1$ random samples from A (one at a time) uniformly without replacement, and if we replace the $(m + 1)^{th}$ sample by t, we get the first scenario conditioned on $\widetilde{m} = m$ and t being sampled. This idea still works in the new setting where the sampling probabilities are no longer the same, since there is still a "draw-by-draw" selection procedure for drawing samples from A (one at a time) in a way so that right after drawing the j^{th} sample, the distribution of samples we currently have is the same as if we have conditioned on

$\widetilde{m} = j$ (e.g., see Section 3 in [14]).

We now consider Lemma 41, which protects the privacy of individuals that blend with few people in the population. The extension of Lemma 41 to robust sampling redefines what is meant by "few people", since even if an individual blends with few people, many of them could be sampled with probability 1. With this modification, the proof of the extended lemma is similar to the proof of the original lemma.

When we prove the extended theorem using the extended lemmas, when we are trying to show that privacy holds for individual t, we look at how many people blend with t that are sampled with probability in $[p, p']$ (in particular, we exclude the ℓ people that are sampled with probability outside of $\{0\} \cup [p, p']$); similar to before, if this number is large, we use the extended version of Lemma 40; otherwise, we use the extended version of Lemma 41.

We now give the formal proof of Theorem 42. We begin by proving a lemma about the smoothness of the Poisson binomial distribution[6] near its expectation, which will be used later in the proof of Lemma 44.

Lemma 43 (Smoothness of the Poisson binomial distribution near its expectation)**.** *Let $\mathcal{P}$ be any population, $0 < p \leq p' < 1$, $\pi : X \to [0, 1]$ be any function, and $\epsilon_{Sam} > 0$. Let A be any non-empty 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 $\Pr[\widetilde{m} = m] \leq \frac{p'}{p} \frac{1-p}{1-p'} e^{\epsilon_{Sam}} \Pr[\widetilde{m} = m+1]$.*

[6]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* $\Pr[\widetilde{m}=m] \geq \frac{p}{p'}\frac{1-p'}{1-p}e^{-\epsilon_{Sam}} \Pr[\widetilde{m}=m+1]$.

Proof. Fix $m \in \{0,\ldots,n-1\}$. Given an individual i in $\mathcal{P}$, let $p_i = \pi(i)$, and let $w_i = \frac{p_i}{1-p_i}$. Given any set A' of individuals in $\mathcal{P}$ and any integer m', let $Q(A',m') = \sum_{B\subseteq A',|B|=m'} \prod_{i\in B} w_i$. Then, we have

$$\frac{\Pr[\widetilde{m}=m]}{\Pr[\widetilde{m}=m+1]} = \frac{\sum_{B\subseteq A,|B|=m}(\prod_{i\in B} p_i)(\prod_{i\in A\setminus B}(1-p_i))}{\sum_{B\subseteq A,|B|=m+1}(\prod_{i\in B} p_i)(\prod_{i\in A\setminus B}(1-p_i))} = \frac{Q(A,m)}{Q(A,m+1)}. \quad (1)$$

We will show that for every $j \in A$,

$$\frac{\partial}{\partial w_j}\left(\frac{Q(A,m)}{Q(A,m+1)}\right) \leq 0. \quad (2)$$

Fix $j \in A$. We note that for any integer m', we have $\frac{\partial}{\partial w_j}Q(A,m') = Q(A\setminus\{j\},m'-1)$ and $Q(A,m') = Q(A\setminus\{j\},m') + w_j Q(A\setminus\{j\},m'-1)$. Then, we observe that

$$\frac{\partial}{\partial w_j}\left(\frac{Q(A,m)}{Q(A,m+1)}\right) = \frac{Q(A,m+1)\cdot Q(A\setminus\{j\},m-1) - Q(A,m)\cdot Q(A\setminus\{j\},m)}{Q(A,m+1)^2}.$$

Now, using the equalities $Q(A,m+1) = Q(A\setminus\{j\},m+1) + w_j Q(A\setminus\{j\},m)$ and $Q(A,m) = Q(A\setminus\{j\},m) + w_j Q(A\setminus\{j\},m-1)$, we get

$$\begin{aligned}&\frac{\partial}{\partial w_j}\left(\frac{Q(A,m)}{Q(A,m+1)}\right)\\ &= \frac{Q(A\setminus\{j\},m+1)\cdot Q(A\setminus\{j\},m-1) - Q(A\setminus\{j\},m)\cdot Q(A\setminus\{j\},m)}{Q(A,m+1)^2}. \quad (3)\end{aligned}$$

We will show that this expression is at most 0 by showing that the numerator $Q(A\setminus\{j\},m+1)\cdot Q(A\setminus\{j\},m-1) - Q(A\setminus\{j\},m)\cdot Q(A\setminus\{j\},m)$ is at most 0. If $m=0$, then $Q(A\setminus\{j\},m-1)=0$, so the numerator is clearly at most 0. Thus, we now assume $m \geq 1$. Consider the full expansion of $Q(A\setminus\{j\},m+1)\cdot Q(A\setminus\{j\},m-1)$ and $Q(A\setminus\{j\},m)\cdot Q(A\setminus\{j\},m)$. Each term of both expansions is of

the form $w_{i_1}^2 \cdots w_{i_r}^2 w_{j_1} \cdots w_{j_s}$, where the indices $i_1, \ldots, i_r, j_1, \ldots, j_s$ are all distinct, and $2r + s = 2m$. For example, a term $w_{i_1}^2 \cdots w_{i_r}^2 w_{j_1} \cdots w_{j_s}$ that appears in the expansion of $Q(A \setminus \{j\}, m+1) \cdot Q(A \setminus \{j\}, m-1)$ is obtained if both $Q(A \setminus \{j\}, m+1)$ and $Q(A \setminus \{j\}, m-1)$ choose $w_{i_1}, \ldots, w_{i_r}$, $Q(A \setminus \{j\}, m+1)$ chooses $m+1-r$ of the factors $w_{j_1}, \ldots, w_{j_s}$, and $Q(A \setminus \{j\}, m-1)$ chooses the remaining factors in $w_{j_1}, \ldots, w_{j_s}$.

Now, consider a term of the form $w_{i_1}^2 \cdots w_{i_r}^2 w_{j_1} \cdots w_{j_s}$, where the indices $i_1, \ldots, i_r, j_1, \ldots, j_s$ are all distinct, and $2r + s = 2m$. It suffices to show that the number of times this term appears in (the full expansion of) $Q(A \setminus \{j\}, m+1) \cdot Q(A \setminus \{j\}, m-1)$ is at most the number of times it appears in $Q(A \setminus \{j\}, m) \cdot Q(A \setminus \{j\}, m)$. If $r > m-1$, then this term appears 0 times in $Q(A \setminus \{j\}, m+1) \cdot Q(A \setminus \{j\}, m-1)$, since $Q(A \setminus \{j\}, m-1)$ needs to choose more than $m-1$ factors in $w_{i_1}, \ldots, w_{i_r}$ but it can only choose at most $m-1$; thus, the numerator in (3) is at most 0, as required. If $r \leq m-1$, then this term appears $\binom{s}{m+1-r}$ times in $Q(A \setminus \{j\}, m+1) \cdot Q(A \setminus \{j\}, m-1)$ and $\binom{s}{m-r}$ times in $Q(A \setminus \{j\}, m) \cdot Q(A \setminus \{j\}, m)$. Now, we note that $\binom{s}{m+1-r} \leq \binom{s}{m-r}$, since $s = 2(m-r)$ and $\binom{2(m-r)}{m+1-r} \leq \binom{2(m-r)}{m-r}$, as required.

Now, from (1),(2), and the fact that $\pi(a) \in [p, p']$ for every $a \in A$, it follows that

$$\frac{\Pr[\widetilde{m} = m]}{\Pr[\widetilde{m} = m+1]} = \frac{Q(A,m)}{Q(A,m+1)} \leq \frac{\sum_{B \subseteq A, |B|=m} \prod_{i \in B} \frac{p}{1-p}}{\sum_{B \subseteq A, |B|=m+1} \prod_{i \in B} \frac{p}{1-p}} = \frac{1-p}{p} \frac{m+1}{n-m} \quad (4)$$

and

$$\frac{\Pr[\widetilde{m} = m]}{\Pr[\widetilde{m} = m+1]} = \frac{Q(A,m)}{Q(A,m+1)} \geq \frac{\sum_{B \subseteq A, |B|=m} \prod_{i \in B} \frac{p'}{1-p'}}{\sum_{B \subseteq A, |B|=m+1} \prod_{i \in B} \frac{p'}{1-p'}} = \frac{1-p'}{p'} \frac{m+1}{n-m}. \quad (5)$$

If $m+1 \leq (n+1)\bar{p} \cdot \frac{e^{\epsilon_{Sam}}}{\bar{p}e^{\epsilon_{Sam}}+(1-\bar{p})}$, then from (4) we have

$$\frac{\Pr[\widetilde{m}=m]}{\Pr[\widetilde{m}=m+1]} \leq \frac{1-p}{p}\frac{m+1}{n-m} = \frac{1-p}{p}\frac{\bar{p}}{1-\bar{p}}\left(\frac{1-\bar{p}}{\bar{p}}\frac{m+1}{n-m}\right) \leq \frac{\bar{p}}{p}\frac{1-p}{1-\bar{p}}e^{\epsilon_{Sam}}$$
$$\leq \frac{p'}{p}\frac{1-p}{1-p'}e^{\epsilon_{Sam}}.$$

If $m+1 \geq (n+1)\bar{p} \cdot \frac{1}{\bar{p}+(1-\bar{p})e^{\epsilon_{Sam}}}$, then from (5) we have

$$\frac{\Pr[\widetilde{m}=m]}{\Pr[\widetilde{m}=m+1]} \geq \frac{1-p'}{p'}\frac{m+1}{n-m} = \frac{1-p'}{p'}\frac{\bar{p}}{1-\bar{p}}\left(\frac{1-\bar{p}}{\bar{p}}\frac{m+1}{n-m}\right) \geq \frac{\bar{p}}{p'}\frac{1-p'}{1-\bar{p}}e^{-\epsilon_{Sam}}$$
$$\geq \frac{p}{p'}\frac{1-p'}{1-p}e^{-\epsilon_{Sam}}.$$

□

We now prove a lemma that essentially says that if an individual blends with many people in the population, then the individual's privacy is protected when we robustly sample from the population and run *San* on the samples. This lemma is essentially the extension of Lemma 40 to robust sampling.

Lemma 44 (Protection of individuals that blend with many people in the population that have a good sampling probability)**.** *Let San be any mechanism, $\mathcal{P}$ be any population, $0 < p \leq p' < 1$, $\pi : X \to [0,1]$ be any function, and $\epsilon \geq 0$. Let t be any individual in $\mathcal{P}$, and let A be any non-empty subset of $\mathcal{P} \setminus \{t\}$ such that for every individual $t' \in A$, we have $t' \approx_{\epsilon,San} t$ and $\pi(t') \in [p,p']$. Let $n = |A|$, $p_t = \pi(t)$, and $\bar{p} = \frac{1}{n}\sum_{t' \in A}\pi(t')$. Then, we have*

$$San(Sam(\mathcal{P},\pi)) \approx_{\epsilon_{final},\delta_{final}} San(Sam(\mathcal{P} \setminus \{t\},\pi)),$$

where $\epsilon_{final} = \ln(p_t \cdot (\frac{p'}{p}\frac{(1-p)(2-p)}{(1-p')^2}e^{\epsilon}) + (1-p_t))$ and $\delta_{final} = \max\{\frac{p_t}{p}, \frac{p_t}{1-p'}\} \cdot e^{-\Omega((n+1)\bar{p}(1-\bar{p})^2)}$.

Proof. Let $\epsilon_{Sam} > 0$, $\widehat{D} = Sam(\mathcal{P},\pi)$, $\widetilde{D} = Sam(\mathcal{P} \setminus \{t\},\pi)$, $\widetilde{m} = |\widetilde{D} \cap A|$, and $Y \subseteq \{0,1\}^*$. Let E be the event that t is sampled when $\widehat{D}$ is chosen.

We first show that for every $m \in \{0, \ldots, n-1\}$, we have

$$\left| \ln \left(\frac{\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} = m]}{\Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} = m+1]} \right) \right| \leq \epsilon. \tag{1}$$

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 the conditional distribution of $Sam(A, \pi)$ given $|Sam(A, \pi)| = j$ (e.g., see Section 3 in [14]). 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 the conditional distribution of $Sam(A, \pi)$ given $|Sam(A, \pi)| = j$.

Now, fix $m \in \{0, \ldots, n-1\}$. Let $\mathcal{D}_m = Sam(\mathcal{P} \setminus (A \cup \{t\}), \pi) \cup \{X_1, \ldots, X_m\}$. Then, for every $D \subseteq \mathcal{P}$, we have $\Pr[\widetilde{D} \cup \{t\} = D \mid \widetilde{m} = m] = \Pr[\mathcal{D}_m \cup \{t\} = D]$ and $\Pr[\widetilde{D} = D \mid \widetilde{m} = m+1] = \Pr[\mathcal{D}_m \cup \{X_{m+1}\} = D]$. Then, using the fact that $t' \approx_{\epsilon, San} t$ for every individual $t' \in A$, we have

$$\begin{aligned} &\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} = m] = \Pr[San(\mathcal{D}_m \cup \{t\}) \in Y] \\ &\leq e^{\epsilon} \Pr[San(\mathcal{D}_m \cup \{X_{m+1}\}) \in Y] = e^{\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} = m+1]. \end{aligned}$$

Similarly, we also have

$$\begin{aligned} &\Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} = m+1] = \Pr[San(\mathcal{D}_m \cup \{X_{m+1}\}) \in Y] \\ &\leq e^{\epsilon} \Pr[San(\mathcal{D}_m \cup \{t\}) \in Y] = e^{\epsilon} \Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} = m]. \end{aligned}$$

Thus, we have shown (1).

Now, we observe that

$$\begin{aligned} \Pr[San(\widehat{D}) \in Y] &= \Pr[San(\widehat{D}) \in Y \mid E] \cdot \Pr[E] + \Pr[San(\widehat{D}) \in Y \mid \overline{E}] \cdot \Pr[\overline{E}] \\ &= \Pr[San(\widetilde{D} \cup \{t\}) \in Y] \cdot p_t + \Pr[San(\widetilde{D}) \in Y] \cdot (1 - p_t). \end{aligned} \tag{2}$$

Let $\alpha = \frac{e^{\epsilon_{Sam}}}{\bar{p}e^{\epsilon_{Sam}}+(1-\bar{p})}$ and $\beta = \frac{1}{\bar{p}+(1-\bar{p})e^{\epsilon_{Sam}}}$, and let $\delta_{Sam} = \max\{\Pr[\widetilde{m}+1 > (n+1)\bar{p}\cdot\alpha], \Pr[\widetilde{m} < (n+1)\bar{p}\cdot\beta]\}$. By Lemma 43 and (1) (and the fact that $m = n$ does not satisfy $m+1 \leq (n+1)\bar{p}\cdot\alpha$), we have

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

and

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

Let $\epsilon_{total} = \ln(p_t\cdot(\frac{p'}{p}\frac{1-p}{1-p'}e^{\epsilon+\epsilon_{Sam}}) + (1-p_t))$. Now, combining (2) and (3), we have

$$
\begin{aligned}
\Pr[San(\widehat{D})\in Y] &\leq (p_t\cdot(\frac{p'}{p}\frac{1-p}{1-p'}e^{\epsilon+\epsilon_{Sam}}) + (1-p_t))\Pr[San(\widetilde{D})\in Y] + p_t\cdot\delta_{Sam} \\
&= e^{\epsilon_{total}}\Pr[San(\widetilde{D})\in Y] + p_t\cdot\delta_{Sam}.
\end{aligned}
$$

Combining (2) and (4), we also have

$$
\begin{aligned}
&\Pr[San(\widehat{D}) \in Y] \\
&\geq (p_t \cdot (\frac{p}{p'}\frac{1-p'}{1-p}e^{-(\epsilon+\epsilon_{Sam})}) + (1-p_t)) \Pr[San(\widetilde{D}) \in Y] - p_t \cdot (\frac{p}{p'}\frac{1-p'}{1-p}e^{-(\epsilon+\epsilon_{Sam})}) \cdot \delta_{Sam} \\
&\Longrightarrow \Pr[San(\widetilde{D}) \in Y] \\
&\leq \frac{1}{p_t \cdot (\frac{p}{p'}\frac{1-p'}{1-p}e^{-(\epsilon+\epsilon_{Sam})}) + (1-p_t)} \Pr[San(\widehat{D}) \in Y] + \frac{\frac{p}{p'}\frac{1-p'}{1-p}e^{-(\epsilon+\epsilon_{Sam})}}{p_t \cdot (\frac{p}{p'}\frac{1-p'}{1-p}e^{-(\epsilon+\epsilon_{Sam})}) + (1-p_t)} p_t \cdot \delta_{Sam} \\
&\leq (p_t \cdot (\frac{p'}{p} \cdot \frac{1-p}{1-p'} \cdot e^{\epsilon+\epsilon_{Sam}}) + (1-p_t)) \Pr[San(\widehat{D}) \in Y] + p_t \cdot \delta_{Sam} \\
&= e^{\epsilon_{total}} \Pr[San(\widehat{D}) \in Y] + p_t \cdot \delta_{Sam},
\end{aligned}
$$

where the last inequality follows from the fact that the function $f(x) = \frac{1}{x}$ is convex for $x > 0$. Thus, we have $San(\widehat{D}) \approx_{\epsilon_{total}, p_t \cdot \delta_{Sam}} San(\widetilde{D})$.

Now, we set $\epsilon_{Sam} = \ln(\frac{2-\bar{p}}{1-\bar{p}})$. Then, we have

$$
\begin{aligned}
\epsilon_{total} &= \ln(p_t \cdot (\frac{p'}{p}\frac{(1-p)(2-\bar{p})}{(1-p')(1-\bar{p})}e^{\epsilon}) + (1-p_t)) \leq \ln(p_t \cdot (\frac{p'}{p}\frac{(1-p)(2-p)}{(1-p')^2}e^{\epsilon}) + (1-p_t)) \\
&= \epsilon_{final}
\end{aligned}
$$

and

$$
\begin{aligned}
&p_t \cdot \delta_{Sam} \\
&= p_t \cdot \max\{\Pr[\widetilde{m} + 1 > (n+1)\bar{p} \cdot \frac{e^{\epsilon_{Sam}}}{\bar{p}e^{\epsilon_{Sam}} + (1-\bar{p})}], \Pr[\widetilde{m} < (n+1)\bar{p} \cdot \frac{1}{\bar{p} + (1-\bar{p})e^{\epsilon_{Sam}}}]\} \\
&= p_t \cdot \max\{\Pr[\widetilde{m} + 1 > (n+1)\bar{p} \cdot (2-\bar{p})], \Pr[\widetilde{m} < (n+1)\bar{p} \cdot \frac{1}{2}]\} \\
&\leq p_t \cdot \max\{\frac{1}{\bar{p}} \Pr[\widetilde{m} + Bin(1,\bar{p}) > (n+1)\bar{p} \cdot (2-\bar{p})], \frac{1}{1-\bar{p}} \Pr[\widetilde{m} + Bin(1,\bar{p}) < (n+1)\bar{p} \cdot \frac{1}{2}]\} \\
&\leq p_t \cdot \max\{\frac{1}{\bar{p}} e^{-\Omega((n+1)\bar{p}(1-\bar{p})^2)}, \frac{1}{1-\bar{p}} e^{-\Omega((n+1)\bar{p})}\} \\
&\leq \max\{\frac{p_t}{p}, \frac{p_t}{1-p'}\} \cdot e^{-\Omega((n+1)\bar{p}(1-\bar{p})^2)} \\
&= \delta_{final},
\end{aligned}
$$

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. □

We now show how pre-sampling combined with a crowd-blending private mechanism can protect the privacy of individuals who blend with few people in the population. The following lemma is essentially the extension of Lemma 41 to robust sampling. This lemma is stated in a somewhat more general form that allows us to use it to prove Theorem 42 later.

Lemma 45 (Protection of individuals that blend with few people in the population)**.** *Let San be any (k, ϵ)-crowd-blending private mechanism with $k \geq 2$, let $\mathcal{P}$ be any population, and let $\pi : X \to [0, 1]$ be any function. Let t be any individual in $\mathcal{P}$, and let A be any non-empty subset of $\mathcal{P} \setminus \{t\}$ such that for every individual $t' \in A$, we have $t' \approx_{\epsilon,San} t$. Let $n = |A|$, $s = |\{t' \in \mathcal{P} \setminus \{t\} : t' \approx_{\epsilon,San} t \text{ and } t' \notin A\}|$, $p_t = \pi(t)$, and $\bar{p} = \frac{1}{n}\sum_{t' \in A} \pi(t')$. Then, if $s < k - 1$, $\bar{p} > 0$, and $n \leq \frac{k-s-1}{\bar{p}(2-\bar{p})}$, then we have*

$$San(Sam(\mathcal{P}, \pi)) \approx_{\epsilon_{final},\delta_{final}} San(Sam(\mathcal{P} \setminus \{t\}, \pi))$$

where $\epsilon_{final} = \ln(p_t e^{\epsilon} + (1 - p_t))$ and $\delta_{final} = p_t e^{-\Omega((k-s)\cdot(1-\bar{p})^2)}$.

Proof. Suppose $s < k - 1$, $\bar{p} > 0$, and $n \leq \frac{k-s-1}{\bar{p}(2-\bar{p})}$. Let $\widehat{D} = Sam(\mathcal{P}, \pi)$, $\widetilde{D} = Sam(\mathcal{P} \setminus \{t\}, \pi)$, and $Y \subseteq \{0, 1\}^*$. Let $\widetilde{m} = |\widetilde{D} \cap A|$, and let E be the event that individual t is in $\widehat{D}$ when $\widehat{D}$ is chosen. We first note that

$$\begin{aligned}\Pr[San(\widehat{D}) \in Y] &= \Pr[San(\widehat{D}) \in Y \mid \overline{E}] \cdot \Pr[\overline{E}] + \Pr[San(\widehat{D}) \in Y \mid E] \cdot \Pr[E] \\ &= \Pr[San(\widetilde{D}) \in Y] \cdot (1 - p_t) + \Pr[San(\widetilde{D} \cup \{t\}) \in Y] \cdot p_t. \quad (1)\end{aligned}$$

We note that if $\widetilde{m} < k - s - 1$, then t ϵ-blends with fewer than k people in $\widetilde{D} \cup \{t\}$, and since San is (k, ϵ)-crowd-blending private, we have

$$\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k - s - 1] \leq e^{\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k - s - 1]$$

and

$$\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k-s-1] \geq e^{-\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k-s-1].$$

Then, we have

$$\begin{aligned}
&\Pr[San(\widetilde{D} \cup \{t\}) \in Y] \\
\leq\ &\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k-s-1] \Pr[\widetilde{m} < k-s-1] + \Pr[\widetilde{m} \geq k-s-1] \\
\leq\ &e^{\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k-s-1] \Pr[\widetilde{m} < k-s-1] + \Pr[\widetilde{m} \geq k-s-1] \\
\leq\ &e^{\epsilon} \Pr[San(\widetilde{D}) \in Y] + \Pr[\widetilde{m} \geq k-s-1], \qquad (2)
\end{aligned}$$

and

$$\begin{aligned}
\Pr[San(\widetilde{D} \cup \{t\}) \in Y] \geq\ &\Pr[San(\widetilde{D} \cup \{t\}) \in Y \mid \widetilde{m} < k-s-1] \Pr[\widetilde{m} < k-s-1] \\
&\geq e^{-\epsilon} \Pr[San(\widetilde{D}) \in Y \mid \widetilde{m} < k-s-1] \Pr[\widetilde{m} < k-s-1] \\
&\geq e^{-\epsilon} (\Pr[San(\widetilde{D}) \in Y] - \Pr[\widetilde{m} \geq k-s-1]) \\
&= e^{-\epsilon} \Pr[San(\widetilde{D}) \in Y] - e^{-\epsilon} \Pr[\widetilde{m} \geq k-s-1]. \qquad (3)
\end{aligned}$$

Now, combining (1) and (2), we have

$$\Pr[San(\widehat{D}) \in Y] \leq (p_t e^{\epsilon} + (1-p_t)) \Pr[San(\widetilde{D}) \in Y] + p_t \Pr[\widetilde{m} \geq k-s-1]. \quad (4)$$

Also, combining (1) and (3), we have

$$\Pr[San(\widehat{D}) \in Y] \geq (p_t e^{-\epsilon} + (1-p_t)) \Pr[San(\widetilde{D}) \in Y] - e^{-\epsilon} p_t \Pr[\widetilde{m} \geq k-s-1].$$

Rearranging this inequality, we get

$$\begin{aligned}
&\Pr[San(\widetilde{D}) \in Y] \\
\leq\ &\frac{1}{p_t e^{-\epsilon} + (1-p_t)} \Pr[San(\widehat{D}) \in Y] + \frac{e^{-\epsilon}}{p_t e^{-\epsilon} + (1-p_t)} p_t \Pr[\widetilde{m} \geq k-s-1] \\
\leq\ &(p_t e^{\epsilon} + (1-p_t)) \Pr[San(\widehat{D}) \in Y] + p_t \Pr[\widetilde{m} \geq k-s-1], \qquad (5)
\end{aligned}$$

where the last inequality follows from the fact that the function $f(x) = \frac{1}{x}$ is convex for $x > 0$, so $\frac{1}{p_t e^{-\epsilon} + (1-p_t)} \leq p_t e^{\epsilon} + (1 - p_t)$.

Let $\tau = \frac{k-s-1}{\bar{p}(2-\bar{p})}$. Then, we have $n \leq \tau$. The lemma now follows from (4), (5), and the inequality

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

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 $\tau\bar{p}$). □

Using the new lemmas (Lemmas 44 and 45), we can now prove Theorem 42 in a way similar to Theorem 39.

Proof of Theorem 42. We first note that $Sam(\cdot, \pi) \in iRS(p, p', l)$. Thus, we can let $T = Sam(\cdot, \pi)$ in the definition of zero-knowledge privacy with respect to $iRS(p, p', l)$. Let A be any adversary. We will describe how to construct a simulator S for A. Let $\mathcal{P}$ be any population, t be any individual in $\mathcal{P}$, and $z \in \{0,1\}^*$. Since the simulator S is given $T(\mathcal{P} \setminus \{t\}) = Sam(\mathcal{P} \setminus \{t\}, \pi)$ and z as part of its input, S can easily compute $San_{zk}(\mathcal{P} \setminus \{t\}) = San(Sam(\mathcal{P} \setminus \{t\}, \pi))$ and then simulate the computation of the adversary A that is given $San_{zk}(\mathcal{P} \setminus \{t\})$ and the auxiliary input z; the simulator S then outputs whatever A outputs.

Now, we note that if $San_{zk}(\mathcal{P}) \approx_{\epsilon_{zk}, \delta_{zk}} San_{zk}(\mathcal{P} \setminus \{t\})$, then $Out_A(A(z) \leftrightarrow San_{zk}(\mathcal{P})) \approx_{\epsilon_{zk}, \delta_{zk}} S(z, T(\mathcal{P} \setminus \{t\}), |\mathcal{P}|)$. Thus, to show that San_{zk} is $(\epsilon_{zk}, \delta_{zk})$-

zero-knowledge private with respect to $iRS(p, p', l)$, it suffices to show that $San_{zk}(\mathcal{P}) \approx_{\epsilon_{zk},\delta_{zk}} San_{zk}(\mathcal{P} \setminus \{t\})$, i.e.,

$$San(Sam(\mathcal{P}, \pi)) \approx_{\epsilon_{zk},\delta_{zk}} San(Sam(\mathcal{P} \setminus \{t\}, \pi)).$$

To this end, let $A = \{t' \in \mathcal{P} \setminus \{t\} : t' \approx_{\epsilon,San} t \text{ and } \pi(t') \in [p, p']\}$, $n = |A|$, $p_t = \pi(t)$, $\bar{p} = \frac{1}{n}\sum_{t' \in A} \pi(t')$, and $s = |\{t' \in \mathcal{P} \setminus \{t\} : t' \approx_{\epsilon,San} t \text{ and } t' \notin A\}|$. It is easy to see that without loss of generality, we can assume that $\mathcal{P}$ satisfies the property that $\pi(t') \neq 0$ for every $t' \in \mathcal{P}$. We note that $s \leq l$, which we use later in some of the inequalities below. Let $\tau = \frac{k-s-1}{\bar{p}(2-\bar{p})}$. We will consider two cases: $n > \tau$ and $n \leq \tau$.

Suppose $n > \tau$. By Lemma 44, we have

$$San(Sam(\mathcal{P}, \pi)) \approx_{\epsilon_1,\delta_1} San(Sam(\mathcal{P} \setminus \{t\}, \pi)),$$

where $\epsilon_1 = \ln(p_t \cdot (\frac{p'}{p}\frac{(1-p)(2-p)}{(1-p')^2}e^{\epsilon}) + (1-p_t)) \leq \ln(p_{\max} \cdot (\frac{p'}{p}\frac{(1-p)(2-p)}{(1-p')^2}e^{\epsilon}) + (1-p_{\max})) = \epsilon_{zk}$ and $\delta_1 = \max\{\frac{p_t}{p}, \frac{p_t}{1-p'}\} \cdot e^{-\Omega((n+1)\bar{p}(1-\bar{p})^2)} \leq \max\{\frac{p_{\max}}{p}, \frac{p_{\max}}{1-p'}\} \cdot e^{-\Omega((k-l)(1-p')^2)} = \delta_{zk}$.

Now, suppose $n \leq \tau$. By Lemma 45, we have

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

where $\epsilon_2 = \ln(p_t e^{\epsilon} + (1 - p_t)) \leq \epsilon_1 \leq \epsilon_{zk}$ and $\delta_2 = p_t e^{-\Omega((k-s)\cdot(1-\bar{p})^2)} \leq p_{\max} e^{-\Omega((k-l)\cdot(1-p')^2)} \leq \delta_{zk}$.

It follows that

$$San(Sam(\mathcal{P}, \pi)) \approx_{\epsilon_{zk},\delta_{zk}} San(Sam(\mathcal{P} \setminus \{t\}, \pi)),$$

as required. □

CHAPTER 4

OUTLIER PRIVACY

4.1 Introduction

In this chapter, we present our work on tailored differential privacy and outlier privacy. 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 [9] 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 [63]). 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 2, 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 2. 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.

4.1.1 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 [31]. 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 [31] 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 [32]) 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 [4], 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 [4] 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 chapter 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 [31]) 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 [31] 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.

4.2 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 ([22, 19]) can now be defined in the following manner:

Definition 46 ((ϵ, δ)-differential privacy [22, 19])**.** 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 47 (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 chapter, we focus on a specific instance of tailored differential privacy, which we call *outlier privacy*. Outlier privacy tailors an individual's privacy pa-

rameter 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 48 (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 49 (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 50 ($(\epsilon(\cdot), \delta(\cdot))$-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.

4.2.1 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 51 ((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 2 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 52. *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 13 (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 53. *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 14 (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 54. *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 [22]), 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 4.3.

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 55. *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 [31], 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 chapter, 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 [31], which we now recall.

Definition 56 (Crowd-blending privacy [31])**.** 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 57. *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. □

4.2.2 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 58 ((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 59 ((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 15 (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 60. *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 16 (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 61. *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 de-

scribed 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 17 (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 [22]).

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.

4.2.3 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 62 (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 18 (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 63. *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 $\mathcal{M}(D \backslash \{t\}) \approx_{0,\ell e^{-\alpha}/2} \mathcal{M}'(D \backslash \{t\})$. 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 63) 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 63, 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.

4.2.4 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 64 (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 19 (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 65 (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

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

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

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

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

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

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 [65]).
- 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 [65]).

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.

4.2.5 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 [65]) 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.

4.3 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 [4]. 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 66 (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 [4] 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 [4] 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 67. *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 67, 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 67, let us make some remarks. Our result (Theorem 67) is somewhat similar to the result in [31] 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 [31], 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 [4], 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 [4] 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 67. We begin by stating a lemma about the smoothness of the Poisson binomial distribution[1] near its expectation, which has appeared in [31], and will be used later in the proof of Lemma 69.

Lemma 68 (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 $\Pr[\widetilde{m} = m] \leq \frac{p'}{p}\frac{1-p}{1-p'}e^{\epsilon_{Sam}} \Pr[\widetilde{m} = m+1]$.*
- *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 68 can be found in the full version of [31]. We now prove a lemma that roughly says that if an individual is $\mathcal{M}$-equivalent to many people

[1]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.

in the population, then the individual's privacy is protected.

Lemma 69. *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 [14]). 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 68 and

(1) (and the fact that $m = n$ does not satisfy $m + 1 \leq (n+1)\bar{p} \cdot \alpha$), we have

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

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 $\epsilon_{Sam} \leq \ln 2$). $\square$

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 70. *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

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

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 $\tau\bar{p}$). $\square$

We will now use the above lemmas to prove Theorem 67.

of Theorem 67. 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 69, 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 70, 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. □

CHAPTER 5

VOTING WITH COARSE BELIEFS

5.1 Introduction

In this chapter, we present our work on voting. We here consider a new approach to bounded-rationality in voting: we assume that voters have "coarse" beliefs.

Strategy-proof voting w.r.t. coarse i.i.d. beliefs. Several celebrated works in the behavioral economics literature (e.g., see [60, 61]) indicate that humans "think through categories" and that a more appropriate model of human behavior is obtained by restricting players to have "coarse" beliefs, where the probabilities are restricted to some coarse set (e.g., a discretization of $[0, 1]$) instead of a continuous interval. In this thesis, we focus on such "coarse" beliefs: we say that a belief is α-coarse if the probabilities (the player assigns to states) are restricted to lie on a uniform discretization of $[0, 1]$ with "mesh size" at least α. Coarse beliefs are very natural. For example, any belief with rational probabilities is an α-coarse belief for some $\alpha > 0$. Also, many natural methods for forming a belief from observations yield α-coarse beliefs where α is inversely proportional to the number of observations; such methods include taking empirical frequencies, as well as using a Dirichlet distribution and updating it when samples or data are observed. We note that even if people form their beliefs using some complicated formula and for instance obtain a belief of the form "event A happens with probability $1/\sqrt{2}$", behavioral experiments (see, e.g., [55, 62]) suggest that people often "round" such beliefs and interpret them using some coarse measure (e.g., event A happens with "very high"/"high"/"medium"/"low"/"very low" probability).

In this thesis, we consider strategy-proofness w.r.t. coarse i.i.d. beliefs. We focus on "large-scale" voting, where the number of voters n is sufficiently large but is still polynomially-related to $1/\alpha$, where α is the coarseness parameter.

Definition 71 (Informal)**.** A voting rule is *large-scale strategy-proof w.r.t. coarse i.i.d. beliefs* if there exists a polynomial $p(\cdot)$ such that for every coarseness parameter $\alpha > 0$, and every $n \geq p(1/\alpha)$, no voter having an α-coarse i.i.d. belief can improve her expected utility by lying about her preferences.

In this thesis, we construct "good" anonymous ϵ-Pareto efficient voting rules that are large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponentially small in the number of voters. For example, one of the voting rules we construct is a modification of the plurality rule, and it chooses a candidate that is guaranteed to be the best or close to the best candidate in terms of the number of votes.

Theorem 72 (Informal)**.** *We construct "good" anonymous ϵ-Pareto efficient voting rules that are large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponentially small in the number of voters.*

Since we are interested in large-scale voting, where we envision the number of voters to exceed 10000, we do not consider the fact that the voting rule only achieves ϵ-Pareto efficiency (as opposed to "exact" Pareto efficiency) unappealing; the probability of Pareto efficiency being violated is on the order of 2^{-100}.

Relaxing the coarse i.i.d. belief assumption. So far we have assumed that each voter has an α-coarse i.i.d. belief. It is well-known that the i.i.d. assumption is seemingly strong in the context of voting. To illustrate this, let us recall an

example from Chamberlain and Rothschild [8]: Consider a simple majority-rule election with two candidates A and B. If a voter believes that each of the other voters will vote for candidate A with probability *exactly* $p = 0.51$, then in a large-scale election, the voter will be essentially certain that candidate A will win (the probability of him casting the pivotal vote will be on the order of e^{-n}, where n is the number of voters). On the other hand, if a voter is *uncertain* about the probability p that the other voters will vote for candidate A (e.g., p is drawn from some distribution over $[0.49, 0.53]$), then this voter may believe that both candidates have a significant chance of winning the election and that the probability of him casting the pivotal vote will be on the order of $1/n$. Note that in the latter case (when the voter is uncertain about p), he no longer has an i.i.d. belief about the preferences of the other voters (conditioned on p, the belief is indeed i.i.d., but the combined process of first sampling p and then sampling $n - 1$ independent preferences (according to p) for the other voters does not result in an i.i.d. belief; see [8] for more discussion on this).

We note, however, that the belief considered above is a *distribution* over i.i.d. beliefs: we first sample a belief, and then independently sample preferences for the other voters according to this belief. Since our notion of large-scale strategy proofness requires strategy-proofness w.r.t. *all* coarse i.i.d. beliefs, it directly follows that our notion implies strategy-proofness w.r.t. to *all distributions* over coarse i.i.d. beliefs (e.g., the uniform distribution over a discretization of $[0.49, 0.53]$ in the above example).

Another seemingly strong aspect of i.i.d. beliefs is that a voter believes that each of the other voters' preferences is drawn from the *same* distribution ϕ (e.g., the distribution determined by p in the above example). Again, this assumption

can be relaxed by allowing the voter to have a *distribution* over possible ϕ's, and a new ϕ is sampled for each of the other voters when sampling their preferences. Such a distribution over possible ϕ's can be collapsed to a single distribution over preferences. In the case of *coarse* i.i.d. beliefs, as long as the distribution over possible ϕ's has finite support and does not depend on the number of voters, the collapsed distribution will be a *coarse* i.i.d. belief. Thus, our notion of large-scale strategy proofness w.r.t. coarse i.i.d. beliefs also directly implies strategy-proofness in this more complicated model. This more complicated model can be used to model situations where a voter believes that the voting population is separated into a constant number of communities, and each of the communities has a different distribution ϕ that is used to generate the community's preferences. For simplicity of presentation, we will state our definitions and results in the more simple model.

5.1.1 Our Construction

Our construction proceeds in two steps. We first show how to construct voting rules that satisfy exact Pareto efficiency but only a notion of large-scale ϵ-strategy-proofness w.r.t coarse i.i.d. beliefs—that is, voters can gain at most ϵ in expected utility by lying—where ϵ is exponentially small in the number of voters n. In a second step, we then show how to transform these voting rules into ones that satisfy actual strategy-proofness w.r.t coarse i.i.d. beliefs (this, however, comes at the cost of achieving only ϵ-Pareto efficiency, where ϵ is exponentially small).

Step 1: Achieving ϵ-strategy-proofness. We now explain (at a high level) how we obtain voting rules that are large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponentially small. To provide some intuition, let us

first consider the plurality rule, which simply chooses the candidate with the most top-choice votes. The plurality rule is not large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs for an exponentially small ϵ. For example, suppose that a voter has the preference ordering $c > a > b$, but she believes that each of the other voters has either the preference ordering $a > b > c$, or the preference ordering $b > a > c$, each with probability 1/2. Such a belief is α-coarse for every $\alpha \leq 1/2$. Now, we observe that according to her belief, her top choice c will certainly not be the winner, so she may want to lie and report her second top choice a as her top choice instead; it can be shown that by doing so, she can increase her expected utility by $\Omega(1/\sqrt{n})$.[1] In this example, the problem is that the voter believes her top choice c will certainly not be the winner, and by lying, she can make it more likely that the plurality rule will choose her second top choice a instead of her last choice b.

We now show how to modify the plurality rule in (what we consider) a natural way to make it large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponentially small. Recall that each voter submits a preference ordering over the entire set of candidates. Our *Repeated Plurality Elimination* voting rule proceeds as follows. We first count the number of top-choice votes for each candidate. Then, we eliminate the "non-great" candidates—these are the candidates whose number of top-choice votes is not within some margin, say $\approx \sqrt{n}$, of the number of top-choice votes of the best candidate. Then, we restrict the voters' preference orderings to the remaining candidates and repeat the elimination process until no more candidates can be eliminated—that is, all the remaining candidates are within the margin of the best candidate. When no more candidates can be eliminated, we run the traditional plurality rule (without a "margin") on the remaining candidates.

[1] For example, this can be shown by using the analytical tools in [7] to establish that with probability $\Omega(1/\sqrt{n})$, the number of top-choice votes for a is equal to that of b, in which case the voter can lie to make a the winner.

Intuitively, these modifications solve the specific issue given above where the voter lies and reports her second top choice a as her top choice, since by Chernoff bounds, with extremely high probability w.r.t. her belief, the number of top-choice votes for candidates a and b will be within the margin $\approx \sqrt{n}$ of each other, while candidate c will be outside the margin; in this case, the voter's lie has no effect, since candidate c will be eliminated while candidates a and b will move onto the next iteration. Thus, the voter might as well tell the truth and report candidate c as her top choice, since candidate c will be eliminated anyway and her second top choice a will become the top choice after restricting the voters' preferences to the remaining candidates. In our voting rule, the elimination process is repeated because after some candidates are eliminated and the top-choice votes are recounted, the same issue may still be present among the remaining candidates.

More generally, to prove that our Repeated Plurality Elimination voting rule is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, we consider a voter i with a coarse belief, and we roughly proceed in two steps. In the first step, we show that voter i believes that she only has an exponentially small chance of influencing which candidates are outside the "margin of the best remaining candidate" and thus will be eliminated. Roughly speaking, to show this, we note that a candidate x's expected count (w.r.t. to voter i's belief) is either a) *equal to the best remaining candidate's expected count*, or b) *different from the best remaining candidate's expected count*. In case a, by Chernoff bounds, the candidate x's actual count will be within the margin with overwhelming probability. In case b, we use the fact that voter i's belief is *coarse* to show that the candidate x's expected count is separated from the best remaining candidate's expected count by a sufficiently large gap, and thus by Chernoff bounds, the candidate x's actual count will be outside the margin with overwhelming probability. In the second step of our proof, we show that voter

i believes that at the end of the elimination process (where all remaining candidates are within the margin of the best candidate), all the remaining candidates will have *exactly the same* expected count with overwhelming probability. In such a situation where all the (remaining) candidates have the same expected count, the plurality rule is actually strategy-proof, which is intuitively why it is okay to run the plurality rule on the remaining candidates at the end. We note that even though voter i only has an exponentially small chance of influencing which candidates get eliminated, voter i does have a reasonable chance (i.e., $\Omega(1/\sqrt{n})$ probability) of influencing which remaining candidate gets chosen by the plurality rule at the end. However, voter i wants to be truthful because of what we have shown in the second step of our proof.

We also consider a variant of the above Repeated Plurality Elimination voting rule, which we refer to as the *approximate instant-runoff* voting rule: at each iteration, instead of eliminating all the candidates that are not "great", we instead eliminate all the candidates that are "close" to the *worst* candidate, with the following exception: if elimination would cause all the candidates to be eliminated (i.e., all the candidates are close to the worst one), we select the winner using the plurality rule. This voting rule is very similar to the widely used *instant-runoff* voting rule. Instant-runoff voting, as well as variations of it, are used in many elections throughout the world (e.g., see [77]). Instant-runoff voting is identical to our "approximate instant-runoff" voting rule with the exception that at each iteration only the candidate with the actual *least* number of top-choice votes is eliminated (as opposed to eliminating all the candidates that are close to it).

More generally, we develop a general framework for constructing voting rules that are large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponen-

tially small, and show how both of the above voting rules (as well as several other voting rules) are natural instances of our framework. All of these voting rules satisfy *exact* Pareto efficiency.

Step 2: Achieving actual strategy-proofness. In the second step of our construction, we provide a general technique for converting voting rules that are large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where $\epsilon = o(1/n^2)$, into voting rules that are large-scale (actual) strategy-proof w.r.t. coarse i.i.d. beliefs. (Note that the plurality rule is only ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs for $\epsilon = \Omega(1/\sqrt{n})$, so this technique cannot be applied to the plurality rule.) In fact, such a technique was already provided in [5] in the context of strategy-proofness without beliefs (and a variant of it was also explored in [66] and [26] in more general mechanism design contexts), and we here extend the analysis to the context of strategy-proofness with beliefs. The idea (from [5]) is to combine in a randomized way an ϵ-strategy-proof voting rule with a so-called "punishing" voting rule that is *strictly* strategy-proof (i.e., voters are strictly better off by truthfully reporting their preferences). The punishing voting rule may not be Pareto efficient, but the combination is done in such a way that the punishing voting rule is only run with tiny probability; this suffices for ensuring that the final voting rule satisfies actual strategy-proofness w.r.t. coarse i.i.d. beliefs. Using this technique, we transform our voting rules into ones that satisfy actual strategy-proofness w.r.t coarse i.i.d. beliefs while satisfying ϵ-Pareto efficiency, where ϵ is exponentially small. This technique actually requires the utility functions of the voters to be coarse, so we will add this assumption to our definition of large-scale strategy-proofness w.r.t. coarse i.i.d. beliefs; a utility function is α-coarse if for every pair of candidates, the utility assigned to the two candidates are either the same or separated by a gap of at least α.

Discussion. Although the random dictatorship voting rule is already Pareto efficient and strategy-proof, the voting rules we construct are arguably much better than random dictatorship. For example, the random dictatorship voting rule is "very random" and can possibly choose a candidate that all voters rank last except for one voter. On the other hand, our Repeated Plurality Elimination voting rule is deterministic, and the punishing voting rule is only run with exponentially small probability; furthermore, our Repeated Plurality Elimination rule is guaranteed to choose a candidate that is the best or close to the best in terms of the number of top-choice votes.

If we ignore our use of the punishing voting rule (which is only run with exponentially small probability), our voting rules (e.g., our approximate instant-runoff voting rule) are all quite natural and very similar to what is used in elections throughout the world. Thus, our results provide some intuition for why strategic misreporting of preferences might not be occurring much in these elections.

We note that our voting rules are not monotone—that is, improving the ranking of a candidate in some voter's preference can decrease the chance of that candidate winning. This is because improving the ranking of a candidate in some voter's preference can change which candidates get eliminated, which then changes the number of top-choice votes each candidate has. This side effect can also occur in the classic instant-runoff voting rule, which is also not monotone.

5.1.2 Other Related Work

In this chapter, we consider strategy-proofness with respect to a restricted class of beliefs. There have been other papers that also consider strategy-proofness with

respect to a restricted class of beliefs. In [54], Majumdar and Sen show that a large class of voting rules are strategy-proof w.r.t. the uniform belief where the other voters' preferences are uniformly distributed. The authors also show that it is not possible to construct a reasonable deterministic voting rule that is strategy-proof w.r.t. any of a large set of beliefs where the voters' preferences are independent of each other; this further suggests that the consideration of independent preferences is not sufficient and that it is appropriate to further assume that the preferences are identically distributed. In [73], Shen used Beta distributions to model the beliefs of voters (in a way that is different from how we model beliefs) in the context of approval voting, and showed that voters may still have incentives to lie. In contrast to the above two papers—which consider very specific types of beliefs—our focus here is on defining a *general* class of natural beliefs for which strategy-proof voting can be achieved.

5.2 Preliminaries

Given an integer $k \in \mathbb{N}$, let $[k] = \{1, \ldots, k\}$. Let $\mathcal{C}$ be any finite set of *candidates* (or *alternatives*). A *preference ordering* on $\mathcal{C}$ is a strict total order on the set of candidates $\mathcal{C}$; let $\mathcal{P}$ denote the set of all preference orderings on $\mathcal{C}$. Given a subset $A \subseteq \mathcal{C}$ of candidates, let $L(A)$ denote the set of preference orderings (i.e., strict total orders) on A. Given a preference ordering P and a pair of candidates $x, y \in \mathcal{C}$, we shall write xPy to mean that x *is (strictly) preferred over* y *in* P, i.e., x is ranked higher than y according to P. Given a preference ordering P, let $top(P)$ denote the highest-ranked candidate according to P, i.e., $top(P)$ is the candidate x in $\mathcal{C}$ such that xPy for every $y \in \mathcal{C} \setminus \{x\}$.

Throughout this chapter, we will use n to denote the number of voters, and m to denote the number of candidates in $\mathcal{C}$; we will often treat m as a constant. A *preference profile* is a vector of length n whose components are preference orderings in $\mathcal{P}$; that is, a preference profile is simply an element of $\mathcal{P}^n$ which specifies the (submitted) preference orderings of n voters. Let $\mathcal{P}^* = \bigcup_{n \in \mathbb{N}} \mathcal{P}^n$.

Given a finite set S, let $\Delta(S)$ denote the set of all probability distributions over S. A (randomized) *voting rule* is a function $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ (or $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ if v works for any number of voters) that maps preference profiles to probability distributions over candidates; intuitively, $v(\vec{P})$ is a distribution over $\mathcal{C}$ that specifies the probability that each candidate is selected when the submitted votes form the preference profile $\vec{P}$. A voting rule v is said to be *deterministic* if for every preference profile $\vec{P}$, the distribution $v(\vec{P})$ assigns probability 1 to some candidate. A voting rule v is said to be *anonymous* if v does not depend on the order in which the preference orderings appear in the input, i.e., $v(P_1, \ldots, P_n) = v(P_{\sigma(1)}, \ldots, P_{\sigma(n)})$ for every preference profile $(P_1, \ldots, P_n) \in \mathcal{P}^n$ and every permutation $\sigma : [n] \to [n]$. In this chapter, we will only consider anonymous voting rules; most common voting rules are indeed anonymous, and one can argue that anonymous voting rules are more fair and democratic than non-anonymous ones.

Given a (randomized) voting rule $v : \mathcal{P}^n \to \Delta(\mathcal{C})$, a candidate $x \in \mathcal{C}$, and a preference profile $\vec{P}$, let $v(x, \vec{P})$ be the probability mass assigned to x by the distribution $v(\vec{P})$; we also refer to $v(x, \vec{P})$ as the *selection probability of x with respect to v and $\vec{P}$*, since $v(x, \vec{P})$ is the probability that candidate x is selected by the voting rule v when the input preference profile is $\vec{P}$. A *utility function* is a function $u : \mathcal{C} \to [0, 1]$ that assigns a real number in $[0, 1]$ to each candidate in $\mathcal{C}$.[2]

[2]It is not important that the codomain of the utility function u is $[0, 1]$; as long as the codomain is bounded, the results of this chapter still hold with minor modifications.

Given a preference ordering P and a utility function u, we say that u is *consistent with* P if for every pair of candidates $x, y \in \mathcal{C}$, we have $u(x) > u(y)$ if and only if xPy.

A voting rule is *Pareto efficient* if it never chooses a Pareto dominated candidate, i.e., a candidate y such that all the voters prefer x over y for some candidate x. A slight relaxation of Pareto efficiency is ϵ-*Pareto efficiency*, where we allow the voting rule to choose a Pareto dominated candidate with probability at most ϵ.

Definition 73 (ϵ-Pareto efficiency)**.** A voting rule $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ is ϵ-*Pareto efficient* if for every pair of candidates $x, y \in \mathcal{C}$ and every preference profile $\vec{P} = (P_1, \ldots, P_n) \in \mathcal{P}^n$ such that xP_iy for every $i \in [n]$, we have $v(y, \vec{P}) \leq \epsilon$.

See Appendix B.1 for background information on the Gibbard-Satterthwaite theorem [34, 71] and Gibbard's generalization of the Gibbard-Satterthwaite theorem to randomized voting rules [35].

5.2.1 Strategy-Proofness with respect to a Set of Beliefs

Gibbard's generalization [35] of the Gibbard-Satterthwaite theorem shows that when there are at least three candidates, we cannot even construct good *randomized* voting rules that are strategy-proof. Given this impossibility result, let us consider relaxed notions of strategy-proofness. We observe that strategy-proofness requires that no voter would want to lie about her true preference even if the voter *knows* the submitted preferences of *all* the other voters. However, in many realistic scenarios, a voter is *uncertain* about how other voters will vote, and she would only lie if she *believes* that she can gain utility in expectation by lying. As a result, we consider

a relaxed notion of strategy-proofness where we consider the voter's *belief* of how the other voters will vote. The standard notion of strategy-proofness requires that no voter would want to lie regardless of what her belief is. To weaken the notion of strategy-proof, one can require that no voter would want to lie as long as her belief belongs in a certain set of beliefs. Let us now move to formalizing these notions.

In this chapter, we will only consider beliefs that are *i.i.d.* (independent and identically distributed), meaning that for each belief, the other voters' preference orderings are sampled independently from some distribution ϕ over preference orderings. Thus, for simplicity, we define a *belief* to be a probability distribution over the set $\mathcal{P}$ of preference orderings, representing a voter's belief that each of the other voters will have a preference ordering drawn independently from this distribution. We now state the definition of *strategy-proof with respect to a set of beliefs*.

Definition 74 (Strategy-proof w.r.t. a set Φ of beliefs)**.** A voting rule $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ is *strategy-proof w.r.t. a set Φ of beliefs* if for every $i \in [n]$, every pair of preference orderings $P_i, P_i' \in \mathcal{P}$, every belief $\phi \in \Phi$, and every utility function u_i that is consistent with P_i, we have

$$\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))],$$

where $\vec{P}_{-i} \sim \phi^{n-1}$.

5.3 Large-Scale Strategy-Proof Voting w.r.t. Coarse i.i.d. Beliefs

In this section, we first define the notion of "coarse" i.i.d. beliefs; then, we introduce the notion of *large-scale strategy-proof w.r.t. coarse i.i.d. beliefs*. We then develop a general framework for constructing voting rules that are large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, and we then use the general framework to obtain many examples of good voting rules. We then show how to transform these voting rules into ones that are *actually* large-scale strategy-proof w.r.t. coarse i.i.d. beliefs.

Let us begin by introducing the notion of a coarse i.i.d. belief. Roughly speaking, an i.i.d. belief ϕ is α-coarse if the probability masses assigned by ϕ are restricted to lie on a uniform discretization of $[0, 1]$ with "mesh size" at least α. More precisely, an i.i.d. belief ϕ is said to be α-*coarse* if the probability masses assigned by ϕ are multiples of some number $\beta \geq \alpha$, i.e., there exists a number $\beta \geq \alpha$ such that for every preference ordering $P \in \mathcal{P}$, we have $\phi(P) = i\beta$ for some integer i. Coarse i.i.d. beliefs are quite natural due to many reasons. For example, if a human were to describe or represent her belief (as a distribution over preference orderings), the probabilities would almost certainly be rational numbers (e.g., it is very strange to believe that a certain preference ordering has probability $1/\pi$ of occurring), and an i.i.d. belief with rational probabilities is an α-coarse i.i.d. belief for some $\alpha > 0$. Also, many common and natural ways of forming a belief also result in a coarse i.i.d. belief. For example, one can use empirical frequencies or a Dirichlet distribution to form a belief from observed samples of preferences. Both of these methods yield α-coarse i.i.d. beliefs, where α is inversely proportional to the number of observations. See Appendix B.2 for more information. We can also consider α-coarse utility functions. A utility function $u : \mathcal{C} \to [0, 1]$ is said to be α-*coarse* if for every

pair of candidates $x, y \in \mathcal{C}$, we have $u(x) = u(y)$ or $|u(x) - u(y)| \geq \alpha$. We only need the utility functions to be coarse for the "punishing" voting rule that we will use later.

Large-scale strategy-proof w.r.t. coarse i.i.d. beliefs. Let us now introduce the notion of *large-scale strategy-proof w.r.t. coarse i.i.d. beliefs*, which is a notion of strategy-proof where the voters have coarse i.i.d. beliefs and there are sufficiently (but still polynomially) many voters.

Definition 75 (Large-scale strategy-proof w.r.t. coarse i.i.d. beliefs)**.** A voting rule $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ is *large-scale strategy-proof w.r.t. coarse i.i.d. beliefs* if there exists a polynomial $p(\cdot)$ such that for every $\alpha > 0$, every $n \geq p(\frac{1}{\alpha})$, every $i \in [n]$, every pair of preference orderings $P_i, P_i' \in \mathcal{P}$, every α-coarse i.i.d. belief ϕ_i, and every α-coarse utility function u_i that is consistent with P_i, we have

$$\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))],$$

where $\vec{P}_{-i} \sim {\phi_i}^{n-1}$; when this holds, we may refer to the above polynomial $p(\cdot)$ as the *rate* of the voting rule v.

In the above definition, α controls the coarseness of the belief, and $p(1/\alpha)$ controls how many voters are required in order to achieve truthfulness; we need n to be sufficiently large because as the i.i.d. beliefs become less and less coarse, the set of beliefs considered becomes closer and closer to the set of all i.i.d. beliefs, which we later show is impossible to construct good voting rules for. The rate $p(\cdot)$ captures how many voters are needed relative to the coarseness of the beliefs.

As mentioned in the introduction, we can consider a slightly more realistic model where each voter has a distribution over α-coarse i.i.d. beliefs, and when

computing expected utility for a voter, a *single* α-coarse i.i.d. belief is sampled from this distribution, and then this sampled belief is used to generate *all* the other voters' preferences in an i.i.d. manner. Our results still hold in this more realistic model; this easily follows from the definition of large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, where it is required that strategy-proofness holds for *every* α-coarse belief, so strategy-proofness also holds if we sample a random α-coarse i.i.d. belief from a distribution.

We now define a relaxed version of large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, where we allow a voter to gain at most $\epsilon(n)$ in expected utility.

Definition 76 (Large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs)**.** A voting rule $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ is *large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs* if there exists a polynomial $p(\cdot)$ such that for every $\alpha > 0$, every $n \geq p(\frac{1}{\alpha})$, every $i \in [n]$, every pair of preference orderings $P_i, P_i' \in \mathcal{P}$, every α-coarse i.i.d. belief ϕ_i, and every α-coarse utility function u_i that is consistent with P_i, we have

$$\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))] - \epsilon(n),$$

where $\vec{P}_{-i} \sim {\phi_i}^{n-1}$; when this holds, we may refer to the above polynomial $p(\cdot)$ as the *rate* of the voting rule v.

5.3.1 Our General Framework

In this section, we develop a general framework for constructing large-scale ϵ-strategy-proof voting rules w.r.t. coarse i.i.d. beliefs. Later, we will show that as long as $\epsilon = o(1/n^2)$, we can transform such voting rules into ones that satisfy *actual* large-scale strategy-proofness w.r.t coarse i.i.d. beliefs. Before we describe the general framework in detail, let us describe an example for motivation.

Recall that the plurality rule simply chooses the candidate with the most top-choice votes. The plurality rule is simple, very commonly used, and intuitively has good efficiency (e.g., it is Pareto efficient). Unfortunately, the plurality rule is not large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs for any exponentially small ϵ, only for $\epsilon = \Omega(1/\sqrt{n})$. However, it is not hard to see that the plurality rule is actually strategy-proof w.r.t. beliefs where all the candidates have the same probability of being the top choice of a voter's preference ordering. Can one design an "elimination rule" that eliminates candidates in a way so that (1) a voter with a coarse i.i.d. belief will believe that she only has an exponentially small chance of affecting which candidates get eliminated, and (2) once these candidates are eliminated from her belief, all the remaining candidates will have the same probability of being the top choice? Intuitively, by running such an elimination rule and then running the plurality rule on the remaining candidates, the combined voting rule is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponentially small. Such an elimination rule exists; it repeatedly eliminates the candidates whose number of top-choice votes is not "close" to the highest number of top-choice votes. We will later show that this elimination rule satisfies the two required properties.

The above example can be viewed as an instantiation of a more general framework for constructing voting rules that are large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, which we now describe. The general framework consists of an "elimination rule" and a "selection rule" satisfying certain properties. The elimination rule will choose a subset of the candidates, and then the selection rule will select a winner from this subset. As long as certain properties are satisfied, the elimination rule combined with the selection rule will be large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs. Let us now informally describe the general procedure and the requirements.

On input a preference profile, we do the following:

Stage 1: Run an "elimination rule" that, on input a preference profile, eliminates a subset of the candidates, leaving a subset $A \subseteq \mathcal{C}$ remaining. We require the following: a single voter with a coarse i.i.d. belief ϕ has little influence on the choice A of the elimination rule when the other voters' preferences are distributed according to ϕ; furthermore, with high probability, the restriction of the belief ϕ to the remaining candidates A results in a belief in some set Φ'_A.

Stage 2: Run a "selection rule" on the preference profile restricted to the remaining candidates A. We require that the selection rule is strategy-proof w.r.t. the set Φ'_A of beliefs from Stage 1.

Intuitively, the above procedure is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs because a voter i with a coarse i.i.d. belief ϕ will believe that she has little influence on the choice A of the elimination rule, and since the restriction of ϕ to A is a belief for which the selection rule is strategy-proof, voter i cannot gain much by lying. We note that even though a voter has little influence on Stage 1, the voter can still have quite a lot of influence on Stage 2 (and thus on the voting rule as a whole), but since the selection rule is strategy-proof w.r.t. the set Φ'_A of beliefs from Stage 1, the voter would want to be truthful.

We now describe the framework more formally. An *elimination rule* is a function $f : \mathcal{P}^* \to \Delta(2^{\mathcal{C}})$ that, on input a preference profile $\vec{P}$, outputs a non-empty subset $A \subseteq \mathcal{C}$ representing the *remaining* candidates after elimination. Recall that given a subset $A \subseteq \mathcal{C}$ of candidates, we use $L(A)$ to denote the set of preference orderings on A. A *selection rule* is a collection of functions

$\{s_A : (L(A))^* \to \Delta(A)\}_{A \subseteq \mathcal{C}, A \neq \emptyset}$, one for each non-empty subset $A \subseteq \mathcal{C}$, such that for every $A \subseteq \mathcal{C}$, s_A is a voting rule for the set of candidates A. Given a selection rule $s = \{s_A\}_{A \subseteq \mathcal{C}, A \neq \emptyset}$ and a preference profile $\vec{P}$ whose components are preference orderings over A, let $s(\vec{P}) = s_A(\vec{P})$.

Given a preference profile $\vec{P}$ and a non-empty subset $A \subseteq \mathcal{C}$ of candidates, let the *restriction of $\vec{P}$ to A*, denoted $\vec{P}|_A$, be the preference profile obtained by removing all the candidates in $\vec{P}$ that are not in A, while preserving the ordering of the remaining candidates. Given an i.i.d. belief ϕ and a non-empty subset $A \subseteq \mathcal{C}$ of candidates, let the *restriction of ϕ to A*, denoted $\phi|_A$, be the belief (i.e., distribution over preference orderings on A) $P|_A$, where $P \sim \phi$. We now state our theorem that precisely describes our general framework.

Theorem 77 (Our general framework)**.** *Let $f : \mathcal{P}^* \to \Delta(2^{\mathcal{C}})$ be any elimination rule, and let $s = \{s_A\}_{A \subseteq \mathcal{C}, A \neq \emptyset}$ be any selection rule. Let $\delta : \mathbb{N} \to \mathbb{R}$ be any function. Suppose there exists a polynomial $p(\cdot)$ such that for every $\alpha > 0$ and every $n \geq p(\frac{1}{\alpha})$, the following holds:*

- *For every $i \in [n]$ and every α-coarse i.i.d. belief ϕ_i, there exists a non-empty subset $A \subseteq \mathcal{C}$ of candidates such that the following conditions hold:*
  - *For every $P_i \in \mathcal{P}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses (to keep) A with probability at least $1 - \delta(n)$ over the randomness of $\vec{P}_{-i} \sim {\phi_i}^{n-1}$ and f.*
  - *s_A is strategy-proof w.r.t. the restricted belief $\phi_i|_A$.*

Then, the voting rule $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ defined by $v(\vec{P}) = s(\vec{P}|_{f(\vec{P})})$ is large-scale 2δ-strategy-proof w.r.t. coarse i.i.d. beliefs, and the rate of v is the polynomial $p(\cdot)$.

See Appendix B.3 for the proof of Theorem 77.

5.3.2 Examples of our General Framework

In this section, we provide some examples of our general framework. Recall that the plurality rule simply chooses the candidate with the most top-choice votes (breaking ties in some way). We now describe a modified plurality rule in the format of our general framework; this voting rule is the same as the repeated plurality elimination voting rule described earlier and in the introduction of this chapter.

Example 20 (Repeated Plurality Elimination + Plurality Selection). Let $0 < \delta < 1/2$, and let $v_{pl} : \mathcal{P}^* \to \Delta(\mathcal{C})$ be a voting rule defined as follows; on input a preference profile $\vec{P} \in \mathcal{P}^n$, v_{pl} does the following:

Stage 1: Repeatedly do the following until no more candidates are eliminated: count the number of top-choice votes for each candidate, and eliminate all the candidates that have a count that is not within $n^{1/2+\delta}$ of the highest count among the remaining candidates; restrict the preference profile to the set of remaining candidates.

Stage 2: Run the plurality rule for the remaining candidates, i.e., on the preference profile restricted to the set of remaining candidates.

Using our general framework, we now show that v_{pl} is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs (where ϵ is exponentially small), and also satisfies certain efficiency properties.

Theorem 78. *Let $0 < \delta < 1/2$, and let v_{pl} be the voting rule defined above. Then, v_{pl} satisfies the following properties:*

1. *v_{pl} is large-scale $(e^{-\Omega(n^{2\delta})})$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$.*
2. *v_{pl} is Pareto efficient.*
3. *v_{pl} is $n^{1/2+\delta}$-close to optimal in the sense that v_{pl} always chooses a candidate $c \in \mathcal{C}$ such that the number of top-choice votes for c is within $n^{1/2+\delta}$ of the highest number of top-choice votes among the candidates.*

Let us explain at a high level how the voting rule v_{pl} satisfies the required conditions in our general framework to establish Property 1 in the above theorem. The idea of the proof is essentially the same as that described in the introduction of this chapter, but here we describe the proof idea in the context of our general framework, whereas in the introduction we did not discuss our general framework for simplicity.) To apply our general framework, we roughly proceed as follows. Consider a voter i with a coarse belief. We need to show that there exists a set A of candidates (dependent on voter i's coarse belief) such that regardless of what preference ordering voter i submits, the set of remaining candidates after Stage 1 (the elimination stage) is precisely A with overwhelming probability. Roughly speaking, to show this, we consider any round in Stage 1, and we note that a remaining candidate x's expected count (w.r.t. to voter i's belief) is either a) *equal to the best remaining candidate's expected count*, or b) *different from the best remaining candidate's expected count*. In case a, by Chernoff bounds, the candidate x's actual count will be within the $n^{1/2+\delta}$ margin with overwhelming probability, and thus will not be eliminated in this round. In case b, we use the fact that voter i's belief is *coarse* to show that the candidate x's expected count is separated

from the best remaining candidate's expected count by a sufficiently large gap, and thus by Chernoff bounds, the candidate x's actual count will be outside the $n^{1/2+\delta}$ margin with overwhelming probability, and thus will be eliminated in this round. Roughly speaking, the required set A of candidates is simply the set of candidates that are expected (with overwhelming probability) to belong to case a throughout all the rounds in Stage 1. Now, applying the union bound over the constant number of rounds in Stage 1, we get our desired result.

To apply our general framework, we also need to show that the plurality rule run in Stage 2 is strategy-proof w.r.t. voter i's coarse belief restricted to the remaining set A of candidates. From the above analysis, it is not hard to see that by definition of A, if voter i's belief is restricted to A, all the candidates (in A) have the same expected count. Now, we note that the plurality rule is strategy-proof w.r.t. any belief where all the candidates have the same expected count, which gives us our desired result. We note that even though voter i only has an exponentially small chance of influencing which candidates get eliminated in Stage 1, voter i does have a reasonable chance (i.e., $\Omega(1/\sqrt{n})$ probability) of influencing which remaining candidate gets chosen by the plurality rule in Stage 2. However, voter i wants to be truthful because the plurality rule is strategy-proof w.r.t. voter i's coarse belief restricted to A.

See Appendix B.3 for the full proof of Theorem 78. Recall that we will later combine this voting rule with a "punishing" voting rule to obtain a voting rule that is large-scale (actual) strategy-proof w.r.t. coarse i.i.d. beliefs.

Example 21 (Approximate Instant-Runoff Voting). The standard instant-runoff voting rule repeats the following until a candidate has been chosen as the winner: count the number of top-choice votes for each candidate, and eliminate the

candidate with the least number of top-choice votes (breaking ties in some way); restrict the preference profile to the set of remaining candidates, and if there is only one candidate remaining, choose the candidate to be the winner.

It is not hard to see that the standard instant-runoff voting rule is not large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is reasonably small. However, we can slightly modify the standard instant-runoff voting rule to obtain an approximate version that is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where ϵ is exponentially small. In each iteration, instead of eliminating only the candidate with the least number of top-choice votes, we eliminate all the candidates that have a count that is close to the least number of top-choice votes; however, we stop right before all the remaining candidates are about to be eliminated, and then we choose the candidate with the most top-choice votes. Let us now put our approximate instant-runoff voting rule in the format of our general framework.

Let $0 < \delta < 1/2$, and let $v_{irv} : \mathcal{P}^* \to \Delta(\mathcal{C})$ be a voting rule defined as follows; on input a preference profile $\vec{P} \in \mathcal{P}^n$, v_{irv} does the following:

Stage 1: Repeat the following: Count the number of top-choice votes for each candidate, and eliminate all the candidates that have a count that is within $n^{1/2+\delta}$ of the least number of top-choice votes, unless doing so would eliminate all the remaining candidates, in which case we simply stop and proceed to Stage 2; restrict the preference profile to the set of remaining candidates.

Stage 2: Run the plurality rule for the remaining candidates, i.e., on the preference profile restricted to the set of remaining candidates.

Using our general framework, we now show that v_{irv} is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs (where ϵ is exponentially small), and also satisfies certain

efficiency properties.

Theorem 79. *Let $0 < \delta < 1/2$, and let v_{irv} be the voting rule defined above. Then, v_{irv} satisfies the following properties:*

1. *v_{irv} is large-scale $(e^{-\Omega(n^{2\delta})})$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$.*
2. *v_{irv} is Pareto efficient.*

See Appendix B.3 for the proof of Theorem 79. Recall that we will later combine this voting rule with a "punishing" voting rule to obtain a voting rule that is large-scale (actual) strategy-proof w.r.t. coarse i.i.d. beliefs.

In Appendix B.4, we provide some more examples of our general framework.

5.3.3 Achieving Actual Strategy-Proofness via the Punishing Voting Rule

In this section, we show how to transform our voting rules into ones that are actually strategy-proof w.r.t. coarse i.i.d. beliefs. We do this by providing a general technique for converting voting rules that are large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where $\epsilon = o(1/n^2)$, into voting rules that are large-scale (actual) strategy-proof w.r.t. coarse i.i.d. beliefs. The idea is to combine in a randomized way an ϵ-strategy-proof voting rule with a "punishing" voting rule that is "strictly strategy-proof". The punishing voting rule is defined as follows:

- Let $v_{punish} : \mathcal{P}^n \to \Delta(\mathcal{C})$ be the voting rule that chooses a voter $i \in [n]$ uniformly at random and then chooses the j^{th} top choice of voter i with probability proportional to $m-j$, i.e., with probability $(m-j)/\sum_{\ell=1}^{m}(m-\ell)$.

We now show that v_{punish} is *strictly* strategy-proof in the sense that if a voter lies about her preference ordering, her expected utility will be strictly less than what it would be if she submitted her true preference ordering, and the difference in the two expected utilities is at least $\Omega(\alpha/n)$, where α is the coarseness of the utility function.

Lemma 80. *The voting rule v_{punish} is "strictly strategy-proof" in the following sense: For every $\alpha > 0$, every $i \in [n]$, every pair of preference orderings $P_i, P_i' \in \mathcal{P}$ with $P_i \neq P_i'$, every $\vec{P}_{-i} \in \mathcal{P}^{n-1}$, and every α-coarse utility function u_i that is consistent with P_i, we have*

$$\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i'))] + \Omega(\alpha/n).$$

See Appendix B.5 for the proof of Lemma 80. We now show that if we take a voting rule v that is large-scale ϵ-strategy proof w.r.t. coarse i.i.d. beliefs, where $\epsilon = o(1/n^2)$, and "mix" it with the punishing voting rule v_{punish} by running v with probability $1-q$ and v_{punish} with probability q for some appropriately chosen $q = \Omega(n^2 \cdot \epsilon(n))$, then the "mixed" voting rule is large-scale (actual) strategy-proof w.r.t. coarse i.i.d. beliefs.

Lemma 81. *Let $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ be any voting rule that is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where $\epsilon(n) = o(1/n^2)$, and let $p(\cdot)$ be the rate of v. Let v_{mix} be the voting rule that runs v with probability $1-q(n)$ and runs v_{punish} with probability $q(n)$, where $q(n) = \Omega(n^2 \cdot \epsilon(n))$. Then, v_{mix} is large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p_{new}(x) = \max\{x, p(x)\}$.*

See Appendix B.5 for the proof of Lemma 81. We now combine the punishing voting rule with our general framework (Theorem 77) to obtain a new general framework for obtaining voting rules that are large-scale (actual) strategy-proof w.r.t. coarse i.i.d. beliefs.

Theorem 82. *Let $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ be a voting rule as defined in Theorem 77 with corresponding function $\delta(n) = o(1/n^2)$ and polynomial $p(\cdot)$. Let v_{mix} be the voting rule that runs v with probability $1 - q(n)$ and runs v_{punish} with probability $q(n)$, where $q(n) = \Omega(n^2 \cdot \delta(n))$. Then, v_{mix} is large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p_{new}(x) = \max\{x, p(x)\}$.*

Proof. The theorem follows by combining Theorem 77 with Lemma 81. □

Using the punishing voting rule, we can also transform our previous voting rules into ones that are large-scale (actual) strategy-proof w.r.t. coarse i.i.d. beliefs, and ϵ-Pareto efficient, where ϵ is exponentially small.

Theorem 83 (Repeated Plurality Elimination + Plurality Selection). *There exists a constant $C > 0$ such that the following holds. Let $0 < \delta < 1/2$, and let v_{pl} be the voting rule in Theorem 78. Let v'_{pl} be the voting rule that runs v_{pl} with probability $1 - e^{-Cn^{2\delta}}$ and runs v_{punish} with probability $e^{-Cn^{2\delta}}$. Then, v'_{pl} satisfies the following properties:*

1. *v'_{pl} is large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$.*
2. *v'_{pl} is $e^{-\Omega(n^{2\delta})}$-Pareto efficient.*
3. *With probability at least $1 - e^{-\Omega(n^{2\delta})}$, v'_{pl} is $n^{1/2+\delta}$-close to optimal in the sense that v'_{pl} chooses a candidate $c \in \mathcal{C}$ such that the number of top-choice*

votes for c is within $n^{1/2+\delta}$ of the highest number of top-choice votes among the candidates.

Proof. The theorem immediately follows by combining Theorem 78 with Lemma 81. □

Theorem 84 (**Approximate Instant-Runoff Voting).** *There exists a constant $C > 0$ such that the following holds. Let $0 < \delta < 1/2$, and let v_{irv} be the voting rule in Theorem 79. Let v'_{irv} be the voting rule that runs v_{irv} with probability $1-e^{-Cn^{2\delta}}$ and runs v_{punish} with probability $e^{-Cn^{2\delta}}$. Then, v'_{irv} satisfies the following properties:*

1. *v'_{irv} is large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$.*
2. *v'_{irv} is $e^{-\Omega(n^{2\delta})}$-Pareto efficient.*

Proof. The theorem immediately follows by combining Theorem 79 with Lemma 81. □

APPENDIX A

APPENDIX FOR CHAPTER 2

Example 22 (A more detailed explanation and analysis of the Democrats vs. Republicans example)**.** Consider a a social network of n people that are grouped into cliques of size c. (For simplicity, assume that c divides n.) In each clique, either most people are Democrats, or most people are Republicans. To model this situation, we first let $\alpha \in [0, 0.2]$. For each clique, we choose a number p in $[0, \alpha] \cup [1 - \alpha, 1]$ randomly and uniformly, and we decide that each person in the clique is a Democrat with probability p, or a Republican with probability $1 - p$. This gives us a probability distribution over databases, each with a binary attribute $X = \{0, 1\}$ and n rows, where each row states the political preference of a single person; a value of 1 represents Democrat, while a value of 0 represents Republican.

Now, let $g : X^n \to \mathbb{R}^{n/c}$ be the function that computes the proportion of Democrats in each clique. Let San be the mechanism that, on input a database $D \in X^n$, first computes $g(D)$ and then adds $Lap(\frac{1}{c\epsilon})$ noise to each component of $g(D)$. San then releases this vector of noisy proportions. The L_1-sensitivity (see [22]) $\Delta(g)$ of the function g being computed is $1/c$, since if a single person changes his or her political preference, the value of g changes only by $1/c$ in one of the components (cliques). Recall from [22] that a mechanism that computes a function $h(D)$ and then adds $Lap(\frac{\Delta(h)}{\epsilon})$ noise to each component of $h(D)$ is ϵ-differentially private. Thus, San is ϵ-differentially private, so for small ϵ, one may think that it is safe to release such information without violating the privacy of any particular person. That is, the released data should not allow us to guess correctly with probability significantly greater than $1/2$ whether a particular person is a Democrat or a Republican. However, this is not the case.

To see this, suppose we know which clique some person i is in. We look at the data released by *San* to obtain the noisy proportion $\hat{p}$ for the clique person i is in. If $\hat{p} \geq 0.5$, we guess that person i's clique mostly consists of Democrats, so we guess that person i is a Democrat; otherwise, we guess that person i's clique mostly consists of Republicans, so we guess that person i is a Republican. Since *San* adds $Lap(\frac{1}{c\epsilon})$ noise to the true proportion p of person i's clique, we have $\Pr[\hat{p} - p \geq \frac{1}{2} - \alpha] = \Pr[p - \hat{p} \geq \frac{1}{2} - \alpha] = F(-(\frac{1}{2} - \alpha)) = \frac{1}{2}e^{-(\frac{1}{2}-\alpha)c\epsilon}$, where $F(x) = \frac{1}{2}e^{xc\epsilon}$ is the cumulative distribution function of the Laplace distribution $Lap(\frac{1}{c\epsilon})$ for $x < 0$.

We note that if $p \in [0, \alpha]$, then $\hat{p} - p < \frac{1}{2} - \alpha$ implies that our guess for person i's clique is correct, so this occurs with probability at least $1 - \frac{1}{2}e^{-(\frac{1}{2}-\alpha)c\epsilon}$. Similarly, if $p \in [1-\alpha, 1]$, then $p - \hat{p} < \frac{1}{2} - \alpha$ implies that our guess for person i's clique is correct, so this occurs with probability at least $1 - \frac{1}{2}e^{-(\frac{1}{2}-\alpha)c\epsilon}$. In both cases, our guess for person i's clique is correct with probability at least $1 - \frac{1}{2}e^{-(\frac{1}{2}-\alpha)c\epsilon}$. Therefore, our guess for person i herself is correct with probability at least $(1 - \frac{1}{2}e^{-(\frac{1}{2}-\alpha)c\epsilon})(1 - \alpha)$.

With $\epsilon = 0.1$, $\alpha = 0.2$, and $c = 200$, our guess for person i is correct with probability at least $(1 - \frac{1}{2}e^{-(\frac{1}{2}-\alpha)c\epsilon})(1 - \alpha) \approx 0.799$. This is significantly higher than $0.5 \cdot e^{\epsilon} = 0.5 \cdot e^{0.1} \approx 0.553$, which one might think is supposed to be an upper bound on the probability that our guess is correct, since *San* satisfies ϵ-differential privacy with $\epsilon = 0.1$ (see the appendix in [22]; the 0.5 comes from guessing randomly).

With $\epsilon = 0.01$, $\alpha = 0.2$, and $c = 200$, our guess for person i is correct with probability at least $(1 - \frac{1}{2}e^{-(\frac{1}{2}-\alpha)c\epsilon})(1 - \alpha) \approx 0.580$. This is still a lot higher than $0.5 \cdot e^{\epsilon} = 0.5 \cdot e^{0.01} \approx 0.505$.

APPENDIX B

APPENDIX FOR CHAPTER 5

B.1 Background Information on the Gibbard-Satterthwaite Theorem

Roughly speaking, a voting rule is said to be *strategy-proof* if no voter can gain utility in expectation by lying about her true preferences. We now give the formal definition of strategy-proof.

Definition 85 (Strategy-proof)**.** A voting rule $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ is *strategy-proof* if for every $i \in [n]$, every preference profile $\vec{P}_{-i} \in \mathcal{P}^{n-1}$, every pair of preference orderings $P_i, P_i' \in \mathcal{P}$, and every utility function u_i that is consistent with P_i, we have

$$\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))].$$

It is desirable for a voting rule to be strategy-proof, since we can then expect voters to honestly submit their true preferences, and thus the candidate chosen by the voting rule will better reflect the voters' true preferences. Unfortunately, if there are at least three candidates, then it is not possible for a deterministic and onto voting rule to be strategy-proof unless it is *dictatorial*, i.e., there exists some voter i such that the voting rule simply always chooses voter i's top choice. This was shown independently by Gibbard [34] and Satterthwaite [71], and is known as the Gibbard-Satterthwaite theorem.

Theorem 86 (Gibbard-Satterthwaite [34, 71])**.** *Suppose there are at least three candidates, i.e., $|\mathcal{C}| \geq 3$. Let $v : \mathcal{P}^n \to \mathcal{C}$ be any deterministic voting rule that is*

onto and strategy-proof. Then, v is dictatorial, i.e., there exists an $i \in [n]$ such that $v(P_1, \ldots, P_n) = top(P_i)$ for every preference profile $(P_1, \ldots, P_n) \in \mathcal{P}^n$.

The Gibbard-Satterthwaite theorem considers voting rules that are *deterministic*. However, several years later, Gibbard [35] generalized the Gibbard-Satterthwaite theorem to *randomized* voting rules. Before we state Gibbard's generalized impossibility result, let us state some required definitions. A (randomized) voting rule $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ is said to be *unilateral* if it only depends on the preference of a single voter, i.e., there exists an $i \in [n]$ such that $v(\vec{P}) = v(\vec{P'})$ for every $\vec{P} = (P_1, \ldots, P_n), \vec{P'} = (P'_1, \ldots, P'_n) \in \mathcal{P}^n$ such that $P_i = P'_i$. A (randomized) voting rule $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ is said to be *duple* if v always chooses some candidate from a fixed set of two candidates, i.e., there exist candidates $x, y \in \mathcal{C}$ such that $v(z, \vec{P}) = 0$ for every $z \in \mathcal{C} \setminus \{x, y\}$ and $\vec{P} \in \mathcal{P}^n$.

Intuitively, when there are at least three candidates, both unilateral rules and duple rules are undesirable, since the former only consider a single voter's preference, and the latter essentially ignore all but two candidates. Gibbard's generalized impossibility result [35] states that any randomized strategy-proof voting rule is a probability distribution over unilateral rules and duple rules.

Theorem 87 (Gibbard [35])**.** *Let $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ be any randomized voting rule that is strategy-proof. Then, v is a distribution over unilateral rules and duple rules, i.e., there exist randomized voting rules $v_1, \ldots, v_t$ and weights $\alpha_1, \ldots, \alpha_t \in (0, 1]$ with $\sum_{i=1}^{t} \alpha_i = 1$, such that each v_i is unilateral or duple, and $v(x, \vec{P}) = \alpha_1 v_1(x, \vec{P}) + \cdots + \alpha_t v_t(x, \vec{P})$ for every $\vec{P} \in \mathcal{P}^n$ and $x \in \mathcal{C}$.*

A corollary of Gibbard's impossibility result is that if a randomized voting rule is strategy-proof and Pareto efficient, then it is a probability distribution over

dictatorial voting rules.

Corollary 88 (Gibbard [35])**.** *Let $v : \mathcal{P}^n \to \Delta(\mathcal{C})$ be any randomized voting rule that is strategy-proof and Pareto efficient. Then, v is a distribution over dictatorial voting rules, i.e., there exist dictatorial voting rules $v_1, \ldots, v_t$ and weights $\alpha_1, \ldots, \alpha_t \in (0, 1]$ with $\sum_{i=1}^{t} \alpha_i = 1$, such that $v(x, \vec{P}) = \alpha_1 v_1(x, \vec{P}) + \cdots + \alpha_t v_t(x, \vec{P})$ for every $\vec{P} \in \mathcal{P}^n$ and $x \in \mathcal{C}$.*

B.2 Forming a Belief from Observations

We now describe how forming a belief from observations using empirical frequencies or a Dirichlet distribution yields α-coarse beliefs, where α is inversely proportional to the number of observations.

Forming a belief using empirical frequencies. Consider a voter that forms a belief ϕ based on ℓ observations $P_1, \ldots, P_\ell$ by simply setting $\phi(P)$ to be the relative frequency of P in $P_1, \ldots, P_\ell$, i.e., $\phi(P) = \frac{|\{j \in [\ell] : P_j = P\}|}{\ell}$. We see that the resulting belief ϕ is $(1/\ell)$-coarse.

Forming a belief using a Dirichlet distribution. Let us first describe a common method of forming a belief based on observations of preferences. We begin with some initial distribution (e.g., the uniform distribution) over the set of all beliefs, and as we make observations, we update this distribution using Bayes' Rule. At any time, our distribution over beliefs can be used to form a single belief by taking the expectation of the distribution over beliefs; equivalently, the single belief is the resulting distribution over preferences obtained by first sampling a

belief from the distribution over beliefs, and then sampling a preference from the sampled belief.

At the beginning when no samples of preferences have been observed yet, we are indifferent between different possible beliefs, so we start with the uniform distribution over the set $\Delta(\mathcal{P})$ of all beliefs. Then, given an observation of a preference ordering P_1, we update the uniform distribution over $\Delta(\mathcal{P})$ by conditioning on the event that the sample P_1 is observed. Upon further observations $P_2, \ldots, P_\ell$, we update the current distribution over $\Delta(\mathcal{P})$ by conditioning on each of the observations $P_2, \ldots, P_\ell$ separately in sequence. The resulting distribution over beliefs can be "collapsed" to give us a single belief as described above.

The distributions over beliefs that we obtain can be described by the *Dirichlet distribution*. The Dirichlet distribution $Dir(\vec{\alpha})$ of order $K \geq 2$ with parameters $\vec{\alpha} = (\alpha_1, \ldots, \alpha_K) > 0$ has a pdf given by $f_{\vec{\alpha}}(x_1, \ldots, x_K) \sim \prod_{i=1}^{K} x_i^{\alpha_i - 1}$ for every $(x_1, \ldots, x_K) \in \mathbb{R}^K$ such that $\sum_{i=1}^{K} x_i = 1$, and is 0 elsewhere. In our context of updating beliefs, we fix an arbitrary ordering of the preferences in $\mathcal{P}$, and we let $K = |\mathcal{P}|$, so $(x_1, \ldots, x_K)$ (with $\sum_{i=1}^{K} x_i = 1$) are the probability masses describing a belief. The uniform distribution over the set of all beliefs is the Dirichlet distribution $Dir(1, \ldots, 1)$. It is known that if the current distribution over beliefs is $Dir(\vec{\alpha})$ and we observe $P_1, \ldots, P_\ell$, then the resulting distribution over beliefs obtained by conditioning on $P_1, \ldots, P_\ell$ is $Dir(\vec{\alpha} + \vec{c})$, where $\vec{c}$ is the vector of counts representing how many times each preference ordering appears in the observations $P_1, \ldots, P_\ell$. It is also known that for any $\vec{\alpha'} = (\alpha'_1, \ldots, \alpha'_K)$, the expectation of $Dir(\vec{\alpha'})$ is $\frac{1}{\sum_{i=1}^{K} \alpha'_i} \cdot (\alpha'_1, \ldots, \alpha'_K)$. Let $\vec{\alpha} = (1, \ldots, 1)$ (vector of K 1's), and let $\vec{\alpha'} = \vec{\alpha} + \vec{c}$, where $\vec{c}$ is as described above. Noting that $||\vec{c}||_1 = \ell$ (since there are ℓ observations), the expectation of $Dir(\vec{\alpha'})$ is $\frac{1}{K+\ell} \cdot (\alpha'_1, \ldots, \alpha'_K)$. Since the belief

formed from $Dir(\vec{\alpha'})$ is the expectation of $Dir(\vec{\alpha'})$, and since the α'_i's are integers, we see that the obtained belief is $\frac{1}{K+\ell}$-coarse.

B.3 Proofs for Section 5.3.1 and 5.3.2

Theorem 77. *Let $f : \mathcal{P}^* \to \Delta(2^{\mathcal{C}})$ be any elimination rule, and let $s = \{s_A\}_{A\subseteq\mathcal{C},A\neq\emptyset}$ be any selection rule. Let $\delta : \mathbb{N} \to \mathbb{R}$ be any function. Suppose there exists a polynomial $p(\cdot)$ such that for every $\alpha > 0$ and every $n \geq p(\frac{1}{\alpha})$, the following holds:*

- *For every $i \in [n]$ and every α-coarse i.i.d. belief ϕ_i, there exists a non-empty subset $A \subseteq \mathcal{C}$ of candidates such that the following conditions hold:*
  - *For every $P_i \in \mathcal{P}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses (to keep) A with probability at least $1 - \delta(n)$ over the randomness of $\vec{P}_{-i} \sim {\phi_i}^{n-1}$ and f.*
  - *s_A is strategy-proof w.r.t. the restricted belief $\phi_i|_A$.*

Then, the voting rule $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ defined by $v(\vec{P}) = s(\vec{P}|_{f(\vec{P})})$ is large-scale 2δ-strategy-proof w.r.t. coarse i.i.d. beliefs, and the rate of v is the polynomial $p(\cdot)$.

Proof. Let $\alpha > 0$, let $n \geq p(1/\alpha)$, let $i \in [n]$, let $P_i, P'_i \in \mathcal{P}$, let ϕ_i be any α-coarse i.i.d. belief, and let u_i be any utility function that is consistent with P_i. Let $\vec{P}_{-i} \sim {\phi_i}^{n-1}$. We will show that

$$\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v(\vec{P}_{-i}, P'_i))] - 2\delta(n). \tag{1}$$

Let A be the set of candidates guaranteed by the assumptions of the theorem statement. Consider an alternate voting rule $v' : \mathcal{P}^* \to \Delta(\mathcal{C})$ defined by $v'(\vec{P}) = s(\vec{P}|_A)$. Since the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses A with probability at least $1-\delta(n)$, it is easy to see that for every $P \in \mathcal{P}$, we have $||v(\vec{P}_{-i}, P) - v'(\vec{P}_{-i}, P)||_1 \leq \delta(n)$. Thus, we have

$$\begin{aligned}
&|\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] - \mathcal{E}[u_i(v'(\vec{P}_{-i}, P_i))]| \\
&= \left| \sum_{c \in \mathcal{C}} \Pr[v(\vec{P}_{-i}, P_i) = c] \cdot u_i(c) - \sum_{c \in \mathcal{C}} \Pr[v'(\vec{P}_{-i}, P_i) = c] \cdot u_i(c) \right| \\
&\leq \sum_{c \in \mathcal{C}} |\Pr[v(\vec{P}_{-i}, P_i) = c] - \Pr[v'(\vec{P}_{-i}, P_i) = c]| \cdot |u_i(c)| \\
&\leq \delta(n). \qquad (2)
\end{aligned}$$

Similarly, we also have

$$|\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))] - \mathcal{E}[u_i(v'(\vec{P}_{-i}, P_i'))]| \leq \delta(n). \qquad (3)$$

Since s_A is strategy-proof w.r.t. $\phi_i|_A$, we have $\mathcal{E}[u_i(v'(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v'(\vec{P}_{-i}, P_i'))]$. Combining this with (2) and (3) yields (1), as required. □

Theorem 78. *Let $0 < \delta < 1/2$, and let v_{pl} be the voting rule defined above Theorem 78 in the body of the paper. Then, v_{pl} satisfies the following properties:*

1. *v_{pl} is large-scale $(e^{-\Omega(n^{2\delta})})$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta) \rceil})$.*
2. *v_{pl} is Pareto efficient.*
3. *v_{pl} is $n^{1/2+\delta}$-close to optimal in the sense that v_{pl} always chooses a candidate $c \in \mathcal{C}$ such that the number of top-choice votes for c is within $n^{1/2+\delta}$ of the highest number of top-choice votes among the candidates.*

Proof. Property 3 clearly follows from the definition of v_{pl}. We will now show Property 2. Let $\vec{P} \in \mathcal{P}^n$ be a preference profile such that every voter in $\vec{P}$ prefers candidate x over candidate y. We note that in order for candidate y to be chosen as the winner, candidate y must be in the set of remaining candidates in Stage 2. However, when this occurs, candidate x would also be in the set of remaining candidates in Stage 2, since candidate x always has a count that is higher than that of candidate y. Thus, candidate y would have no top-choice votes in Stage 2, so it cannot be chosen as the winner by the plurality rule in Stage 2. We have now shown Property 2.

We will now show Property 1. We will use our general framework, i.e., Theorem 77. The elimination rule $f : \mathcal{P}^* \to \Delta(2^{\mathcal{C}})$ corresponds to Stage 1, i.e., it chooses to keep the candidates that are remaining at the end of Stage 1. The selection rule $s = \{s_A\}_{A \subseteq \mathcal{C}, A \neq \emptyset}$ runs the plurality rule on the remaining candidates with respect to the restricted preference profile. For each non-empty $A \subseteq \mathcal{C}$, let Φ'_A be the set of beliefs ϕ (over the set of all preference orderings on A) where every candidate in A has the same probability of being the top choice. It is not hard to verify that the plurality rule, and thus the selection rule, is strategy-proof with respect to each Φ'_A. Let $p(x) = (3x+1)^{\lceil 1/(1/2-\delta) \rceil}$, let $\alpha > 0$, let $n \geq p(1/\alpha)$, let $i \in [n]$, and let ϕ_i be any α-coarse i.i.d. belief.

Given a preference ordering P and a candidate $x \in \mathcal{C}$, let $points(x, P)$ be 1 if x is the top choice in P, and 0 otherwise. Let A be the set of candidates remaining after the following procedure:

- Let $S = \mathcal{C}$, and repeatedly do the following until no more candidates are eliminated: Eliminate all the candidates $a \in S$ such that $\mathcal{E}_{P \sim \phi_i}[points(a, P|_S)] < \max_{a' \in S} \mathcal{E}_{P \sim \phi_i}[points(a', P|_S)]$, and let S be the set of remaining candidates.

We will show that the following conditions hold:

- For every $P_i \in \mathcal{P}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses (to keep) A with probability at least $1 - e^{-\Omega(n^{2\delta})}$ over the randomness of $\vec{P}_{-i} \sim {\phi_i}^{n-1}$ and f.
- The restriction of ϕ_i to A results in a belief in Φ'_A.

The second condition holds because from the definition of A, we see that for every $a \in A$, we have $\Pr_{P\sim\phi_i}[top(P|_A) = a] = \mathcal{E}_{P\sim\phi_i}[points(a, P|_A)] = \max_{a'\in A} \mathcal{E}_{P\sim\phi_i}[points(a', P|_A)]$. Thus, we now show the first condition.

Let $P_i \in \mathcal{P}$. Consider the execution of one iteration of the loop in Stage 1. Suppose the current set of remaining candidates is S and we are currently at Stage 1. Let $M = \max_{a'\in S} \mathcal{E}_{P\sim\phi_i}[points(a', P|_S)]$. Let E be the set of candidates $a \in S$ such that $\mathcal{E}_{P\sim\phi_i}[points(a, P|_S)] = M$. We first show that for each candidate $y \in S \setminus E$, we have

$$\mathcal{E}_{P\sim\phi_i}[points(y, P|_S)] \leq M - \alpha. \tag{1}$$

It suffices to show that for every $x, y \in S$, we have $|\mathcal{E}_{P\sim\phi_i}[points(x, P|_S)] - \mathcal{E}_{P\sim\phi_i}[points(y, P|_S)]| = 0$ or $|\mathcal{E}_{P\sim\phi_i}[points(x, P|_S)] - \mathcal{E}_{P\sim\phi_i}[points(y, P|_S)]| \geq \alpha$. To see this, let $x, y \in S$, and observe that

$$\begin{aligned}
&|\mathcal{E}_{P\sim\phi_i}[points(x, P|_S)] - \mathcal{E}_{P\sim\phi_i}[points(y, P|_S)]| \\
&= \left|\sum_{P\in\mathcal{P}} \phi_i(P) \cdot points(x, P|_S) - \sum_{P\in\mathcal{P}} \phi_i(P) \cdot points(y, P|_S)\right| \\
&= \left|\sum_{P\in\mathcal{P}} \phi_i(P) \cdot (points(x, P|_S) - points(y, P|_S))\right|. \qquad (2)
\end{aligned}$$

Since ϕ_i is α-coarse, there exists a $\beta \geq \alpha$ such that for every $P \in \mathcal{P}$, $\phi_i(P)$ is a multiple of β. Thus, each term of the sum in (2) is a multiple of β, and so the

sum is also a multiple of β. Thus, the entire expression in (2) is either 0 or at least $\beta \geq \alpha$, as required. Thus, we have shown (1).

Let $P_{i'} \sim \phi_i$ independently for every $i' \in [n] \setminus \{i\}$, and let $score_{-i}(x) = \sum_{i' \in [n] \setminus \{i\}} points(x, P_{i'}|_S)$ for every $x \in S$. By a Chernoff bound, for each $x \in S$, we have

$$\Pr[|score_{-i}(x) - \mathcal{E}[score_{-i}(x)]| \geq n^{1/2+\delta}/4] \leq e^{-\Omega(n^{2\delta})}.$$

Now, by the union bound, we have

$$\Pr[\exists x \in S : |score_{-i}(x) - \mathcal{E}[score_{-i}(x)]| \geq n^{1/2+\delta}/4] \leq m \cdot e^{-\Omega(n^{2\delta})} = e^{-\Omega(n^{2\delta})}. \tag{3}$$

Since $\mathcal{E}[score_{-i}(x)] = (n-1) \cdot M$ for every $x \in E$, it follows from (3) that

$$\Pr[\exists x, y \in E : |score_{-i}(x) - score_{-i}(y)| \geq n^{1/2+\delta}/2] \leq e^{-\Omega(n^{2\delta})}. \tag{4}$$

From (1), we have $\mathcal{E}_{P \sim \phi_i}[points(y, P|_S)] \leq M - \alpha$ for every $y \in S \setminus E$. Thus, $\mathcal{E}[score_{-i}(y)] = (n-1) \cdot \mathcal{E}_{P \sim \phi_i}[points(y, P|_S)] \leq (n-1)M - (n-1)\alpha < (n-1)M - 2n^{1/2+\delta}$ for every $y \in S \setminus E$, so it also follows from (3) that

$$\Pr[\exists x \in S \setminus E, y \in E : score_{-i}(x) \geq score_{-i}(y) - n^{1/2+\delta} - 1] \leq e^{-\Omega(n^{2\delta})}. \tag{5}$$

Since the elimination rule f eliminates precisely the candidates that have a score (i.e., count) that is not within $n^{1/2+\delta}$ of the maximum score among the candidates, and since voter i's preference ordering P_i adds at most 1 to the score of a candidate, we see (from (4), (5), and the union bound) that with probability at least $1 - e^{-\Omega(n^{2\delta})}$, precisely the candidates in $S \setminus E$ will be eliminated in the current iteration of Stage 1. Thus, at each iteration, with probability at least $1 - e^{-\Omega(n^{2\delta})}$, the set of candidates that get eliminated in the iteration precisely matches the set of candidates that would be eliminated in the procedure used to define A. Thus, by

the union bound, we have that with probability at least $1-m\cdot e^{-\Omega(n^{2\delta})} = 1-e^{-\Omega(n^{2\delta})}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses (to keep) A.

Now, by Theorem 77, v_{pl} is large-scale $e^{-\Omega(n^{2\delta})}$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$. □

Theorem 79. *Let $0 < \delta < 1/2$, and let v_{irv} be the voting rule defined above Theorem 79 in the body of the paper. Then, v_{irv} satisfies the following properties:*

1. *v_{irv} is large-scale $(e^{-\Omega(n^{2\delta})})$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$.*
2. *v_{irv} is Pareto efficient.*

Proof. We first show Property 2. Let $\vec{P} \in \mathcal{P}^n$ be a preference profile such that every voter in $\vec{P}$ prefers candidate x over candidate y. We note that in order for candidate y to be chosen as the winner, candidate y must be in the set of remaining candidates in Stage 2. However, when this occurs, candidate x would also be in the set of remaining candidates in Stage 2, since candidate x always has a count that is higher than that of candidate y. Thus, candidate y would have no top-choice votes in Stage 2, so it cannot be chosen as the winner by the plurality rule in Stage 2. We have now shown Property 2.

We will now show Property 1. We will use our general framework, i.e., Theorem 77. The elimination rule $f : \mathcal{P}^* \to \Delta(2^{\mathcal{C}})$ corresponds to Stage 1, i.e., it chooses to keep the candidates that are remaining at the end of Stage 1. The selection rule $s = \{s_A\}_{A\subseteq\mathcal{C}, A\neq\emptyset}$ runs the plurality rule on the remaining candidates with respect to the restricted preference profile. For each non-empty $A \subseteq \mathcal{C}$, let Φ'_A be the set of beliefs ϕ (over the set of all preference orderings on A) where every candidate

in A has the same probability of being the top choice. It is not hard to verify that the plurality rule, and thus the selection rule, is strategy-proof with respect to each Φ'_A. Let $p(x) = (3x+1)^{\lceil 1/(1/2-\delta)\rceil}$, let $\alpha > 0$, let $n \geq p(1/\alpha)$, let $i \in [n]$, and let ϕ_i be any α-coarse i.i.d. belief.

Given a preference ordering P and a candidate $x \in \mathcal{C}$, let $points(x, P)$ be 1 if x is the top choice in P, and 0 otherwise. Let A be the set of candidates remaining after the following procedure:

- Initialize $S := \mathcal{C}$, and repeat the following: Eliminate all the candidates $a \in S$ such that $\mathcal{E}_{P\sim\phi_i}[points(a, P|_S)] = \min_{a'\in S} \mathcal{E}_{P\sim\phi_i}[points(a', P|_S)]$, unless this would eliminate all the remaining candidates, in which case we stop and exit the repeat loop without eliminating any of the remaining candidates. Let S be the new set of remaining candidates.

We will show that the following conditions hold:

- For every $P_i \in \mathcal{P}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses (to keep) A with probability at least $1 - e^{-\Omega(n^{2\delta})}$ over the randomness of $\vec{P}_{-i} \sim {\phi_i}^{n-1}$ and f.
- The restriction of ϕ_i to A results in a belief in Φ'_A.

The second condition holds because from the definition of A, we see that for every $a \in A$, we have $\Pr_{P\sim\phi_i}[top(P|_A) = a] = \mathcal{E}_{P\sim\phi_i}[points(a, P|_A)] = \max_{a'\in A} \mathcal{E}_{P\sim\phi_i}[points(a', P|_A)]$. Thus, we now show the first condition.

Let $P_i \in \mathcal{P}$. Consider the execution of one iteration of the loop in Stage 1. Suppose the current set of remaining candidates is S and we are currently at Stage 1. Let $M = \min_{a'\in S} \mathcal{E}_{P\sim\phi_i}[points(a', P|_S)]$. Let E be the set of candidates

$a \in S$ such that $\mathcal{E}_{P\sim\phi_i}[points(a, P|_S)] = M$. We first show that for each candidate $y \in S \setminus E$, we have

$$\mathcal{E}_{P\sim\phi_i}[points(y, P|_S)] \geq M + \alpha. \tag{1}$$

It suffices to show that for every $x, y \in S$, we have $|\mathcal{E}_{P\sim\phi_i}[points(x, P|_S)] - \mathcal{E}_{P\sim\phi_i}[points(y, P|_S)]| = 0$ or $|\mathcal{E}_{P\sim\phi_i}[points(x, P|_S)] - \mathcal{E}_{P\sim\phi_i}[points(y, P|_S)]| \geq \alpha$. To this end, let $x, y \in S$, and observe that

$$\begin{aligned}
&|\mathcal{E}_{P\sim\phi_i}[points(x, P|_S)] - \mathcal{E}_{P\sim\phi_i}[points(y, P|_S)]| \\
&= \left|\sum_{P\in\mathcal{P}} \phi_i(P) \cdot points(x, P|_S) - \sum_{P\in\mathcal{P}} \phi_i(P) \cdot points(y, P|_S)\right| \\
&= \left|\sum_{P\in\mathcal{P}} \phi_i(P) \cdot (points(x, P|_S) - points(y, P|_S))\right|. \qquad (2)
\end{aligned}$$

Since ϕ_i is α-coarse, there exists a $\beta \geq \alpha$ such that for every $P \in \mathcal{P}$, $\phi_i(P)$ is a multiple of β. Thus, each term of the sum in (2) is a multiple of β, and so the sum is also a multiple of β. Thus, the entire expression in (2) is either 0 or at least $\beta \geq \alpha$, as required. Thus, we have shown (1).

Let $P_{i'} \sim \phi_i$ independently for every $i' \in [n] \setminus \{i\}$, and let $score_{-i}(x) = \sum_{i'\in[n]\setminus\{i\}} points(x, P_{i'}|_S)$ for every $x \in S$. By a Chernoff bound, for each $x \in S$, we have

$$\Pr[|score_{-i}(x) - \mathcal{E}[score_{-i}(x)]| \geq n^{1/2+\delta}/4] \leq e^{-\Omega(n^{2\delta})}.$$

Now, by the union bound, we have

$$\Pr[\exists x \in S : |score_{-i}(x) - \mathcal{E}[score_{-i}(x)]| \geq n^{1/2+\delta}/4] \leq m \cdot e^{-\Omega(n^{2\delta})} = e^{-\Omega(n^{2\delta})}. \tag{3}$$

Since $\mathcal{E}[score_{-i}(x)] = (n-1) \cdot M$ for every $x \in E$, it follows from (3) that

$$\Pr[\exists x, y \in E : |score_{-i}(x) - score_{-i}(y)| \geq n^{1/2+\delta}/2] \leq e^{-\Omega(n^{2\delta})}. \tag{4}$$

From (1), we have $\mathcal{E}_{P\sim\phi_i}[points(y, P|_S)] \geq M + \alpha$ for every $y \in S \setminus E$. Thus, $\mathcal{E}[score_{-i}(y)] = (n-1)\cdot\mathcal{E}_{P\sim\phi_i}[points(y, P|_S)] \geq (n-1)M + (n-1)\alpha > (n-1)M + 2n^{1/2+\delta}$ for every $y \in S \setminus E$, so it also follows from (3) that

$$\Pr[\exists x \in S \setminus E, y \in E : score_{-i}(x) \leq score_{-i}(y) + n^{1/2+\delta} + 1] \leq e^{-\Omega(n^{2\delta})}. \quad (5)$$

Since the elimination rule f eliminates precisely the candidates that have a score (i.e., count) that is not within $n^{1/2+\delta}$ of the minimum score among the candidates, and since voter i's preference ordering P_i adds at most 1 to the score of a candidate, we see (from (4), (5), and the union bound) that with probability at least $1 - e^{-\Omega(n^{2\delta})}$, precisely the candidates in E will be eliminated in the current iteration of Stage 1. Thus, at each iteration, with probability at least $1 - e^{-\Omega(n^{2\delta})}$, the set of candidates that get eliminated in the iteration precisely matches the set of candidates that would be eliminated in the procedure used to define A. Thus, by the union bound, we have that with probability at least $1-m\cdot e^{-\Omega(n^{2\delta})} = 1-e^{-\Omega(n^{2\delta})}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses (to keep) A.

Now, by Theorem 77, v_{irv} is large-scale $e^{-\Omega(n^{2\delta})}$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$. □

B.4 More Examples of our General Framework

A (positional) *scoring rule* is a voting rule where each candidate x receives a certain number of points from each voter i depending on the position of x in voter i's preference ordering, and the candidate with the highest total score wins (breaking ties in some way). A scoring rule has a *points vector* $(p_1, \ldots, p_m) \in \mathbb{N}^m$ associated with it; for each voter i with submitted preference ordering P_i, the j^{th} top candidate in P_i receives p_j points. There are many well-known examples of

scoring rules, such as the following:

- Plurality: The *plurality* voting rule chooses the candidate with the most top-choice votes (breaking ties in some way). This is simply a scoring rule with the points vector $(1, 0, \ldots, 0) \in \mathbb{N}^m$.
- Borda count: The *Borda count* voting rule is a scoring rule with the points vector $(m, m-1, \ldots, 1) \in \mathbb{N}^m$ (recall that m is the number of candidates).

Example 23 (Scoring Rule Elimination + Input-Independent Selection). Let $0 < \delta < 1/2$, and let $v_{score} : \mathcal{P}^* \to \Delta(\mathcal{C})$ be any voting rule defined as follows; on input a preference profile $\vec{P} \in \mathcal{P}^n$, v_{score} does the following:

Stage 1: Use a scoring rule to compute the scores of the candidates, and then eliminate all the candidates with a score that is not within $n^{1/2+\delta}$ of the highest score among the candidates.

Stage 2: Choose a winner (deterministically or randomly) from the remaining candidates in any way that does not depend on the input preference profile.

Using our general framework, we now show that v_{score} is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs (where ϵ is exponentially small), and also satisfies certain efficiency properties.

Theorem 89. *Let $0 < \delta < 1/2$, and let v_{score} be the voting rule defined above. Then, v_{score} satisfies the following properties:*

1. *v_{score} is large-scale $(e^{-\Omega(n^{2\delta})})$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta) \rceil})$.*

2. *v_{score} is Pareto efficient if the points vector of the scoring rule is strictly decreasing, or if the scoring rule is the plurality rule and n is sufficiently large.*

3. *v_{score} is $n^{1/2+\delta}$-close to optimal in the sense that v_{score} always chooses a candidate $c \in \mathcal{C}$ such that the score of c is within $n^{1/2+\delta}$ of the highest score among the candidates.*

Proof. Property 3 clearly follows from the definition of v_{score}. We will now show Property 2. Let $\vec{P} \in \mathcal{P}^n$ be a preference profile such that every voter in $\vec{P}$ prefers candidate x over candidate y. It suffices to show that candidate y will be eliminated by v_{score}, i.e., the score of y is not within $n^{1/2+\delta}$ of the maximum score among the candidates. If the points vector of the scoring rule is strictly decreasing, then the score of x is at least n more than the score of y (since the points in the points vector are integers), as required. On the other hand, if the scoring rule is the plurality rule, then the score of y is 0 while the maximum score among the candidates is at least $n/(|\mathcal{C}|-1)$; when n is sufficiently large, the score of y is not within $n^{1/2+\delta}$ of the maximum score among the candidates, as required. We have shown Property 2.

We will now show Property 1. We will use our general framework, i.e., Theorem 77. The elimination rule $f : \mathcal{P}^* \to \Delta(2^{\mathcal{C}})$ corresponds to Stage 1, i.e., f chooses to keep the candidates that are within $n^{1/2+\delta}$ of the maximum score among the candidates. The selection rule $s = \{s_A\}_{A \subseteq \mathcal{C}, A \neq \emptyset}$ is the rule used in Stage 2. Clearly, for every non-empty $A \subseteq \mathcal{C}$, s_A is strategy-proof with respect to the set of all beliefs. Let $p(x) = (3x+1)^{\lceil 1/(1/2-\delta) \rceil}$, let $\alpha > 0$, let $n \geq p(1/\alpha)$, let $i \in [n]$, and let ϕ_i be any α-coarse i.i.d. belief.

Let $(p_1, \ldots, p_m) \in \mathbb{N}^m$ be the points vector associated with v_{score}. Given a

preference ordering P and a candidate $x \in \mathcal{C}$, let $points(x, P)$ be the number of points candidate x would receive from a voter with submitted preference ordering P, i.e., $points(x, P) = p_j$, where j is the position of candidate x in P, with the topmost position being position 1.

Let $M = \max_{a' \in \mathcal{C}} \mathcal{E}_{P \sim \phi_i}[points(a', P)]$. Let A be the set of candidates $a \in \mathcal{C}$ such that $\mathcal{E}_{P \sim \phi_i}[points(a, P)] = M$. We will show that the following holds:

- For every $P_i \in \mathcal{P}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses (to keep) A with probability at least $1 - e^{-\Omega(n^{2\delta})}$ over the randomness of $\vec{P}_{-i} \sim {\phi_i}^{n-1}$ and f.

Let $P_i \in \mathcal{P}$. We first show that for each candidate $y \in \mathcal{C} \setminus A$, we have

$$\mathcal{E}_{P \sim \phi_i}[points(y, P)] \leq M - \alpha. \tag{1}$$

It suffices to show that for every $x, y \in \mathcal{C}$, we have $|\mathcal{E}_{P \sim \phi_i}[points(x, P)] - \mathcal{E}_{P \sim \phi_i}[points(y, P)]| = 0$ or $|\mathcal{E}_{P \sim \phi_i}[points(x, P)] - \mathcal{E}_{P \sim \phi_i}[points(y, P)]| \geq \alpha$. To see this, let $x, y \in \mathcal{C}$, and observe that

$$\begin{aligned}
&|\mathcal{E}_{P \sim \phi_i}[points(x, P)] - \mathcal{E}_{P \sim \phi_i}[points(y, P)]| \\
&= \left| \sum_{P \in \mathcal{P}} \phi_i(P) \cdot points(x, P) - \sum_{P \in \mathcal{P}} \phi_i(P) \cdot points(y, P) \right| \\
&= \left| \sum_{P \in \mathcal{P}} \phi_i(P) \cdot (points(x, P) - points(y, P)) \right|. \qquad (2)
\end{aligned}$$

Since ϕ_i is α-coarse, there exists a $\beta \geq \alpha$ such that for every $P \in \mathcal{P}$, $\phi_i(P)$ is a multiple of β. Thus, each term of the sum in (2) is a multiple of β, and so the sum is also a multiple of β. Thus, the entire expression in (2) is either 0 or at least $\beta \geq \alpha$, as required. Thus, we have shown (1).

Let $P_{i'} \sim \phi_i$ independently for every $i' \in [n] \setminus \{i\}$, and let $score_{-i}(x) = \sum_{i' \in [n] \setminus \{i\}} points(x, P_{i'})$ for every $x \in \mathcal{C}$. By a Chernoff bound, for each $y \in \mathcal{C}$,

we have

$$\Pr[|score_{-i}(y) - \mathcal{E}[score_{-i}(y)]| \geq n^{1/2+\delta}/4] \leq e^{-\Omega(n^{2\delta})}.$$

Now, by the union bound, we have

$$\Pr[\exists x \in \mathcal{C} : |score_{-i}(x) - \mathcal{E}[score_{-i}(x)]| \geq n^{1/2+\delta}/4] \leq m \cdot e^{-\Omega(n^{2\delta})} = e^{-\Omega(n^{2\delta})}. \tag{3}$$

Since $\mathcal{E}[score_{-i}(x)] = (n-1) \cdot M$ for every $x \in A$, it follows from (3) that

$$\Pr[\exists x, y \in A : |score_{-i}(x) - score_{-i}(y)| \geq n^{1/2+\delta}/2] \leq e^{-\Omega(n^{2\delta})}. \tag{4}$$

From (1), we have $\mathcal{E}_{P\sim\phi_i}[points(y, P)] \leq M - \alpha$ for every $y \in \mathcal{C} \setminus A$. Thus, $\mathcal{E}[score_{-i}(y)] = (n-1) \cdot \mathcal{E}_{P\sim\phi_i}[points(y, P)] \leq (n-1)M - (n-1)\alpha < (n-1)M - 2n^{1/2+\delta}$ for every $y \in \mathcal{C} \setminus A$, so it also follows from (3) that

$$\Pr[\exists x \in \mathcal{C} \setminus A, y \in A : score_{-i}(x) \geq score_{-i}(y) - n^{1/2+\delta} - \max_{j\in[m]} p_j] \leq e^{-\Omega(n^{2\delta})}. \tag{5}$$

Since the elimination rule f eliminates precisely the candidates that have a score not within $n^{1/2+\delta}$ of the maximum score among the candidates, and since voter i's preference ordering P_i adds at most $\max_{j\in[m]} p_j$ to the score of any candidate, we see (from (4), (5), and the union bound) that with probability at least $1 - e^{-\Omega(n^{2\delta})}$, the elimination rule $f(\vec{P}_{-i}, P_i)$ chooses to keep A.

Now, by Theorem 77, v_{score} is large-scale $e^{-\Omega(n^{2\delta})}$-strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p(x) = O(x^{\lceil 1/(1/2-\delta)\rceil})$. □

Using a strategy-proof voting rule in Stage 2 instead of input-independent selection. In Stage 2, a winner is chosen in any way that does not depend on the input preference profile. However, it is easy to see that one

can run any strategy-proof (in the traditional sense) voting rule on the (preference profile restricted to the) remaining candidates in Stage 2, since such a selection rule clearly still satisfies the requirements of our general framework.

Using plurality in Stage 2 when there are only two candidates remaining. Whenever Stage 1 eliminates all but two candidates, the voting rule v_{score} can actually run the plurality rule on the two remaining candidates in Stage 2 instead of choosing a winner in a way that does not depend on the input preference profile. This is because the plurality rule is strategy-proof when there are only two candidates, and so the selection rule clearly still satisfies the requirements of our general framework. This improvement to the voting rule v_{score} can be especially useful when it is widely believed that there are two "strong" candidates that are much more preferred by the voters than the other candidates.

B.5 Proofs for Section 5.3.3

Lemma 80. *The voting rule v_{punish} is "strictly strategy-proof" in the following sense: For every $\alpha > 0$, every $i \in [n]$, every pair of preference orderings $P_i, P_i' \in \mathcal{P}$ with $P_i \neq P_i'$, every $\vec{P}_{-i} \in \mathcal{P}^{n-1}$, and every α-coarse utility function u_i that is consistent with P_i, we have*

$$\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i'))] + \Omega(\alpha/n).$$

The proof of Lemma 80 *roughly* works as follows. If a voter lies about her preference by swapping two adjacent candidates in her preference ordering, then with probability $1/n$, the voter will be chosen, and she will lose a constant amount

of expected utility; this is because the less preferred candidate is now higher and thus will be chosen with higher probability, while the more preferred candidate is now lower and thus will be chosen with lower probability (and the utilities assigned to the two swapped candidates have an α gap between them, since the utility function is α-coarse). We show that we can obtain any (false) preference ordering from the true preference ordering by performing a sequence of swaps of adjacent candidates, where the less preferred candidate (according to the true preference) is always swapped upwards; each of these swaps causes the voter to lose $\Omega(\alpha/n)$ expected utility, as described earlier. Thus, the lemma holds.

Proof. Let $\alpha > 0$, let $i \in [n]$, let $P_i, P_i' \in \mathcal{P}$ with $P_i \neq P_i'$, let $\vec{P}_{-i} \in \mathcal{P}^{n-1}$, and let $u_i : \mathcal{C} \to [0,1]$ be any α-coarse utility function that is consistent with P_i. We will show that

$$\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i'))] + \Omega(\alpha/n). \tag{1}$$

Let $\vec{P} = (\vec{P}_{-i}, P_i)$ and $\vec{P}' = (\vec{P}_{-i}, P_i')$. Let $a_1, \ldots, a_m$ be the ordering of the candidates in the preference ordering P_i', with a_1 being the top (highest-ranked) candidate in P_i'. We observe that $\vec{P}'$ can be obtained from $\vec{P}$ by performing the following sequence of swaps of adjacent candidates in voter i's preference ordering (similar to how bubble sort works): We first take the candidate a_1 in the preference ordering P_i and move a_1 to the top position by repeatedly swapping a_1 with the candidate directly above; this makes the top candidate of the resulting preference ordering coincide with the top candidate of P_i'. We then take the candidate a_2 in the resulting preference ordering and move a_2 to the second top position by repeatedly swapping the candidate with the candidate directly above; this makes the top two candidates of the resulting preference ordering coincide with the top two candidates of P'. We then take the candidate a_3 in the resulting preference

ordering and move the candidate to the third top position by repeatedly swapping the candidate with the candidate directly above. It is easy to see that by continuing this process in the natural way, we will eventually get the preference ordering P_i'.

We now analyze how the expected utility $\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, \cdot))]$ changes as we perform the swaps to get from P_i to P_i' for voter i's preference ordering. We note that for each swap, we are swapping a pair of adjacent candidates, say x and y with x on top of y (before the swap), such that the preference ordering P_i ranks x higher than y. Let Q_i and Q_i' denote the two preference orderings for voter i before and after such a swap, respectively. Now, we observe that

$$\begin{aligned}
&\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, Q_i'))] - \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, Q_i))] \\
&= \frac{1}{n}[u_i(x) \cdot v_{punish}(x, (\vec{P}_{-i}, Q_i')) + u_i(y) \cdot v_{punish}(y, (\vec{P}_{-i}, Q_i'))] \\
&\quad - \frac{1}{n}[u_i(x) \cdot v_{punish}(x, (\vec{P}_{-i}, Q_i)) + u_i(y) \cdot v_{punish}(y, (\vec{P}_{-i}, Q_i))] \\
&= \frac{1}{n}u_i(x) \cdot [v_{punish}(x, (\vec{P}_{-i}, Q_i')) - v_{punish}(x, (\vec{P}_{-i}, Q_i))] \\
&\quad + \frac{1}{n}u_i(y) \cdot [v_{punish}(y, (\vec{P}_{-i}, Q_i')) - v_{punish}(y, (\vec{P}_{-i}, Q_i))] \\
&= \frac{1}{n}u_i(x) \cdot \left(-\frac{1}{\sum_{k=1}^{m}(m-k)}\right) + \frac{1}{n}u_i(y) \cdot \left(\frac{1}{\sum_{k=1}^{m}(m-k)}\right) \\
&= \frac{1}{n}(u_i(y) - u_i(x)) \cdot \left(\frac{2}{m(m-1)}\right).
\end{aligned}$$

Since the preference ordering P_i ranks x higher than y, and since the utility function u_i is consistent with P_i, we have $u_i(x) > u_i(y)$, so $u_i(y) - u_i(x) \leq -\alpha$ (since u_i is α-coarse). Thus, we have

$$\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, Q_i'))] - \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, Q_i))] \leq -\Omega(\alpha/n).$$

Thus, the expected utility $\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, \cdot))]$ goes down by at least $\Omega(\alpha/n)$ each time we perform a swap in the sequence of swaps to get from P_i to P_i' for voter i's preference ordering. This implies that $\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i))] \geq$

$\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i'))] + \Omega(\alpha/n)$, which shows (1), as required. This completes the proof of the lemma. □

Lemma 81. *Let $v : \mathcal{P}^* \to \Delta(\mathcal{C})$ be any voting rule that is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, where $\epsilon(n) = o(1/n^2)$, and let $p(\cdot)$ be the rate of v. Let v_{mix} be the voting rule that runs v with probability $1-q(n)$ and runs v_{punish} with probability $q(n)$, where $q(n) = \Omega(n^2 \cdot \epsilon(n))$. Then, v_{mix} is large-scale strategy-proof w.r.t. coarse i.i.d. beliefs, with rate $p_{new}(x) = \max\{x, p(x)\}$.*

The proof of Lemma 81 roughly works as follows. By Lemma 80, if a voter lies about her preference, she will gain at most $\epsilon = o(1/n^2)$ expected utility if v_{mix} runs the voting rule v, but she will lose at least $\Omega(\alpha/n)$ expected utility if v_{mix} runs the voting rule v_{punish}, where α is the coarseness of her utility function. The probability q that v_{mix} runs v_{punish} is appropriately chosen so that overall, the voter does not gain any expected utility from lying.

Proof. Let $p_{new}(x) = \max\{x, p(x)\}$. Let $\alpha > 0$, let $n \geq p_{new}(1/\alpha)$, let $i \in [n]$, let $P_i, P_i' \in \mathcal{P}$ with $P_i \neq P_i'$, let ϕ_i be any α-coarse i.i.d. belief, and let u_i be any α-coarse utility function that is consistent with P_i. Let $\vec{P}_{-i} \sim {\phi_i}^{n-1}$. Since v is large-scale ϵ-strategy-proof w.r.t. coarse i.i.d. beliefs, we have

$$\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))] - \epsilon(n).$$

By Lemma 80, we also have

$$\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i))] \geq \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i'))] + \alpha/n.$$

Now, we observe that

$$
\begin{aligned}
&\mathcal{E}[u_i(v_{mix}(\vec{P}_{-i}, P_i))] \\
&= (1 - q(n)) \cdot \mathcal{E}[u_i(v(\vec{P}_{-i}, P_i))] + q(n) \cdot \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i))] \\
&\geq (1 - q(n)) \cdot (\mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))] - \epsilon(n)) + q(n) \cdot (\mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i'))] + \alpha/n) \\
&= (1 - q(n)) \cdot \mathcal{E}[u_i(v(\vec{P}_{-i}, P_i'))] + q(n) \cdot \mathcal{E}[u_i(v_{punish}(\vec{P}_{-i}, P_i'))] - (1 - q(n)) \cdot \epsilon(n) + q(n)\alpha/n \\
&= \mathcal{E}[u_i(v_{mix}(\vec{P}_{-i}, P_i'))] - (1 - q(n)) \cdot \epsilon(n) + q(n)\alpha/n. \qquad (1)
\end{aligned}
$$

Now, we observe that by choosing $q(n) = \Omega(n^2 \cdot \epsilon(n))$ appropriately, we have $-(1 - q(n)) \cdot \epsilon(n) + q(n)\alpha/n \geq -\epsilon(n) + q(n)/(n^2) \geq 0$ (since $\alpha \geq 1/n$), and the lemma follows. □

BIBLIOGRAPHY

[1] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[2] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07: Proc. of the 16th international conference on World Wide Web*, pages 181–190, 2007.

[3] J. J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):pp. 227–241, 1989.

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

[5] Eleanor Birrell and Rafael Pass. Approximately strategy-proof voting. In *IJCAI*, pages 67–72, 2011.

[6] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *STOC '08: Proc. of the 40th annual ACM symposium on Theory of computing*, pages 609–618, 2008.

[7] Gabriel Caroll. A quantitative approach to incentives: Application to voting rules. Manuscript, 2013.

[8] Gary Chamberlain and Michael Rothschild. A note on the probability of casting a decisive vote. *Journal of Economic Theory*, 25(1):152 – 162, 1981.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[10] Kamalika Chaudhuri and Nina Mishra. When random sampling preserves privacy. In *CRYPTO'06*, pages 198–213, 2006.

[11] Shuchi Chawla, Cynthia Dwork, Frank McSherry, Adam Smith, and Hoeteck Wee. Toward privacy in public databases. In *Second Theory of Cryptography Conference (TCC 2005)*, pages 363–385, 2005.

[12] Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM J. Comput.*, 34(6):1370–1379, 2005.

[13] Bee-Chung Chen, Daniel Kifer, Kristen LeFevre, and Ashwin Machanavajjhala. Privacy-preserving data publishing. *Foundations and Trends in Databases*, 2(1-2):1–167, 2009.

[14] Xiang-Hui Chen, Arthur P. Dempster, and Jun S. Liu. Weighted finite population sampling to maximize entropy. *Biometrika*, 81(3):pp. 457–469, 1994.

[15] Vincent Conitzer and Tuomas Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 627–634, 2006.

[16] Tor Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[17] Shahar Dobzinski and Ariel D. Procaccia. Frequent manipulability of elections: The case of two voters. In *Proceedings of the 4th International Workshop on Internet and Network Economics*, WINE '08, pages 653–664, 2008.

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

[19] Cynthia Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

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

[21] Cynthia Dwork, Krishnaram Kenthapadi, Frank Mcsherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *In EUROCRYPT*, pages 486–503, 2006.

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

[23] Cynthia Dwork and Moni Naor. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy, 2008.

[24] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 381–390, 2009.

[25] Cynthia Dwork, Guy Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proc. of the 51st Annual IEEE Symposium on Foundations of Computer Science*, 2010.

[26] Amos Fiat, Anna R. Karlin, Elias Koutsoupias, and Angelina Vidali. Approaching utopia: Strong truthfulness and externality-resistant mechanisms. In *Proceedings of the 4th Innovations in Theoretical Computer Science*, ITCS, 2013.

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

[28] E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 243 –249, oct. 2008.

[29] Ehud Friedgut, Gil Kalai, Nathan Keller, and Noam Nisan. A quantitative version of the gibbard-satterthwaite theorem for three alternatives. *SIAM J. Comput.*, 40(3):934–952, 2011.

[30] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42(4):1–53, 2010.

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

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

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

[34] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):pp. 587–601, 1973.

[35] Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–81, 1977.

[36] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *Proc. of the 29th annual ACM symposium on Theory of computing*, pages 406–415, 1997.

[37] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. In *Proc. of the 30th annual ACM Symposium on Theory of Computing*, pages 289–298, 1998.

[38] Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Struct. Algorithms*, 32(4):473–493, 2008.

[39] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1:102–114, August 2008.

[40] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[41] Marcus Isaksson, Guy Kindler, and Elchanan Mossel. The geometry of manipulation – A quantitative proof of the gibbard-satterthwaite theorem. *Combinatorica*, 32(2):221–250, March 2012.

[42] Carter Jernigan and Behram Mistree. Gaydar. `http://www.telegraph.co.uk/technology/facebook/6213590/Gay-men-can-be-identified-by-their-Facebook-friends.html`, 2009.

[43] S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Foundations of Computer Science, 2008*, pages 531–540, 2008.

[44] Tali Kaufman, Michael Krivelevich, and Dana Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM J. Comput.*, 33(6):1441–1483, 2004.

[45] JerryS. Kelly. Almost all social choice rules are highly manipulable, but a few aren't. *Social Choice and Welfare*, 10:161–175, 1993.

[46] Daniel Kifer. Attacks on privacy and definetti's theorem. In *SIGMOD Conference*, pages 127–138, 2009.

[47] V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J-H Cui, and A. G. Percus. Reducing large internet topologies for faster simulations. In *IFIP NETWORKING*, 2005.

[48] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *KDD '06: Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, 2006.

[49] Samantha Leung, Edward Lui, and Rafael Pass. Stronger impossibility results for strategy-proof voting with i.i.d. beliefs. Manuscript, 2015.

[50] Samantha Leung, Edward Lui, and Rafael Pass. Voting with coarse beliefs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 61–61, New York, NY, USA, 2015. ACM.

[51] Ninghui Li, Wahbeh H. Qardaji, and Dong Su. Provably private data anonymization: Or, k-anonymity meets differential privacy. Manuscript, 2011.

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

[53] Edward Lui and Rafael Pass. Outlier privacy. In Yevgeniy Dodis and Jesper-Buus Nielsen, editors, *Theory of Cryptography*, volume 9015 of *Lecture Notes in Computer Science*, pages 277–305. Springer Berlin Heidelberg, 2015.

[54] Dipjyoti Majumdar and Arunava Sen. Ordinally bayesian incentive compatible voting rules. *Econometrica*, 72(2):523–540, 2004.

[55] Barbara C. Malt, Brian H. Ross, and Gregory L. Murphy. Predicting features for members of natural categories when categorization is uncertain. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(3):646 – 661, 1995.

[56] Sharon Marko and Dana Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Trans. Algorithms*, 5(2):1–28, 2009.

[57] Andrew McLennan. Manipulation in elections with uncertain preferences. *Journal of Mathematical Economics*, 47(3):370–375, 2011.

[58] Elchanan Mossel and Miklós Z. Rácz. Election manipulation: The average case. *SIGecom Exchanges*, 11(2):22–24, 2012.

[59] Elchanan Mossel and Miklós Z. Rácz. A quantitative gibbard-satterthwaite theorem without neutrality. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 1041–1060, 2012.

[60] S. Mullainathan. Thinking through categories. *NBER working paper*, 2002.

[61] Sendhil Mullainathan, Joshua Schwartzstein, and Andrei Shleifer. Coarse thinking and persuasion. *The Quarterly Journal of Economics*, 123(2):577–619, 2008.

[62] G.L. Murphy and B.H. Ross. Predictions from uncertain categorizations. *Cognitive Psychology*, 27(2):148 – 193, 1994.

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

[64] M. E. J. Newman. Ego-centered networks and the ripple effect. *Social Networks*, 25(1):83 – 95, 2003.

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

[66] Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. Approximately optimal mechanism design via differential privacy. In *ITCS*, pages 203–213, 2012.

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

[68] Michal Parnas and Dana Ron. Testing the diameter of graphs. *Random Struct. Algorithms*, 20(2):165–183, 2002.

[69] Ariel D. Procaccia and Jeffrey S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *J. Artif. Int. Res.*, 28(1):157–181, 2007.

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

[71] Mark Allen Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, April 1975.

[72] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145 – 147, 1972.

[73] Emily Shen. *Pattern Matching Encryption, Strategic Equivalence of Range Voting and Approval Voting, and Statistical Robustness of Voting Rules*. PhD thesis, Massachusetts Institute of Technology, 2013.

[74] Herbert A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.

[75] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, October 2002.

[76] Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating private synthetic data. In *Proceedings of the 8th conference on Theory of cryptography*, TCC'11, pages 400–416, 2011.

[77] Wikipedia. Instant-runoff voting - wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Instant-runoff_voting`, 2014. Accessed: 2014-02-09.

[78] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, and Jian Pei. Minimality attack in privacy preserving data publishing. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 543–554. VLDB Endowment, 2007.

[79] Lirong Xia and Vincent Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *ACM Conference on Electronic Commerce*, pages 99–108, 2008.

[80] Lei Zhang, Sushil Jajodia, and Alexander Brodsky. Information disclosure under realistic assumptions: privacy versus optimality. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 573–583. ACM, 2007.