CERIAS Tech Report 2010-27

Provably Private Data Anonymization: Or, k-Anonymity Meets Differential Privacy

by Ninghui Li, Wahbeh Qardaji, Dong Su
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Provably Private Data Anonymization: Or, *k*-Anonymity Meets Differential Privacy

Ninghui Li
Purdue University
305 N. University Street, West
Lafayette, IN 47907, USA
ninghui@cs.purdue.edu

Wahbeh Qardaji
Purdue University
305 N. University Street, West
Lafayette, IN 47907, USA
wqardaji@cs.purdue.edu

Dong Su
Purdue University
305 N. University Street, West
Lafayette, IN 47907, USA
su17@cs.purdue.edu

ABSTRACT

Privacy-preserving microdata publishing currently lacks a solid theoretical foundation. Most existing techniques are developed to satisfy syntactic privacy notions such as k-anonymity, which fails to provide strong privacy guarantees. The recently proposed notion of differential privacy has been widely accepted as a sound privacy foundation for statistical query answering. However, no general practical microdata publishing techniques are known to satisfy differential privacy. In this paper, we start to bridge this gap. We first analyze k-anonymization methods and show how they fail to provide sufficient protection against re-identification, which it was designed to protect. We then prove that, k-anonymization methods, when done "safely", and when preceded with a random sampling step, can satisfy (ϵ, δ)-differential privacy with reasonable parameters. This result is, to our knowledge, the first to link k-anonymity with differential privacy and illustrates that "hiding in a crowd of k" indeed offers privacy guarantees. This naturally leads to future research in designing "safe" and practical k-anonymization methods. We observe that our result gives an alternative approach to output perturbation for satisfying differential privacy: namely, adding a random sampling step in the beginning and pruning results that are too sensitive to changing a single tuple. This approach may be applicable to settings other than microdata publishing. We also show that adding a random-sampling step can greatly amplify the level of privacy guarantee provided by a differentially-private algorithm. This result makes it much easier to provide strong privacy guarantees when one wishes to publish a portion of the raw data. Finally, we show that current definitions of (ϵ, δ)-differential privacy require δ to be very small to provide sufficient privacy protection when publishing microdata, making the notion impractical in some scenarios. To address this problem, we introduce a notion called f-smooth (ϵ, δ)-differential privacy.

1. INTRODUCTION

In an age where the minute details of our life are recorded and stored in databases, a clear paradox is emerging between the need to preserve the privacy of individuals and the need to use these collected data for research, public policy formulation, and other purposes. We consider the scenario where a trusted curator gathers sensitive information from a large number of respondents. The curator may then use the data in two different ways. In the first way, the curator's goal is to learn (and release to the public) statistical facts about the underlying population, while protecting the privacy of respondents. This is known as privacy-preserving data analysis, statistical disclosure control, inference control, or privacy-preserving data mining. In the second way, the curator publishes a sanitized (or, "anonymized") version of the dataset so that other parties can use the data to perform any analysis they are interested in. This is called privacy-preserving microdata publishing.

There are two settings for privacy-preserving data analysis: interactive and non-interactive. In the interactive setting, the curator provides a mechanism with which users may pose queries about the data, and get (possibly noisy) answers. This setting was extensively studied in the 1980's and 1990's. This field, however, sees a renewed and growing interest in recent years, fueled by the development of ϵ-Differential Privacy and its variants [10, 11]. These notions are intuitive and have been proven to provide semantic privacy; they are currently widely accepted as the privacy notion of choice in the interactive setting. In the non-interactive setting, the curator computes and publishes some statistics of the collected dataset.

In this paper, we focus on privacy-preserving microdata publishing, where sanitization or anonymization is applied to the input dataset before publishing. Traditionally, sanitization employed perturbation and data modification techniques, and may also include some accompanying synopses and statistics. Over the last decade or so, there has been extensive work in database community on various data anonymization methods. Many techniques employ generalization (or recoding), i.e., replacing a data value with a less precise value that is semantically consistent, and suppression, i.e., removing tuples that stand out.

Despite the number of anonymization techniques proposed, the field of privacy-preserving microdata publishing currently lacks a solid theoretical foundation. Sweeny and Samarati [30, 29, 27, 28] introduced the well-known k-anonymity notion. It is a syntactic property on anonymized datasets: when only certain attributes, known as quasi-identifiers (QIDs) are considered, each tuple in the anonymized dataset should appear at least k times. The k-anonymity notion has generally be considered too weak. Many privacy notions have been introduced since then, e.g., l-diversity [22] and t-closeness [20]. We often see that a paper introduces a new privacy definition motivated by weaknesses of existing definitions, only to be attacked by a later paper.

Given this current state of the art, one may argue that given the difficulties in establishing a strong privacy definition in privacy-preserving microdata publishing, we should give up the approach

and focus on interactive private data analysis. We argue that the interactive setting cannot replace microdata publishing for the reasons given below, and privacy-preserving microdata publishing remains necessary.

First, the interactive setting is unsuitable when the number of data users is large. In an interactive mechanism, the approach to satisfy differential privacy is to add an appropriate level of noise to each query result. With each query, the overall level of privacy for the dataset decreases as more information is being leaked. Hence, the standard approach is to have a privacy budget, a portion of which is used by answering each new query. When the privacy budget is used up after answering a number of queries, the curator can no longer effectively answer any new queries. This would be unacceptable in situations where a dataset needs to serve the general public such as when the census bureau provides the census data to the public. Because the curator cannot be sure whether any two data users are colluding or not, the privacy budget has to be for *all users*. When there are many users, the issue of dividing the privacy budget among them becomes a thorny problem in itself. In any case, each user can get only a very small budget that is unlikely to be sufficient.

Second, even when a data user has a generous privacy budget, perhaps because there are few data users, this privacy-budget paradigm can become a hinderance to creative research. In a production setting where the queries one wants answered are well-known, this paradigm may be satisfactory. However, when one wants to explore the data and test various hypothesis, the privacy budget can become a serous limitation. For many analysis tasks, effective mechanisms that satisfy differential privacy do not yet exist. Even when one uses only queries where effective mechanisms exist, privacy budgets become a finite and non-replaceable resource that will be consumed by each query, essentially limiting one's access to the data. The research value of data shared under a privacy budget is significantly diminished.

Furthermore, the interactive approach puts lots of resource constraints on the curator and requires a close coupling between the curator and data users. The curator must run all the client queries and algorithms on its servers, which represents a significant overhead. The data users, on the other hand, must reveal to the curator the queries they want to make, which may be considered confidential or sensitive by the data users.

Finally, there are many situations in which answering statistical queries simply does not achieve the purpose of sharing the data. One example is Netflix's release of movie rating data for researchers to develop algorithms based on the data. Given an algorithm, it is possible to make it differentially private, as shown in [24]. However, the purpose here is to let researchers develop new algorithms. It is unclear how this can be done without access to some data. Another example comes from a real-world challenge that was introduced by the Chief Information Officer (CIO) of a large supermarket chain in the United States. The CIO identified the problem of generating production-like data for the purpose of testing systems under development as a major challenge facing CIOs today. The test data must be as close to production data as possible, yet one cannot use production data because it includes sensitive customer information that must be protected according to laws and regulations. Privacy-preserving microdata publishing techniques can help in these situations.

Given these reasons, we argue that privacy-preserving microdata publishing should be studied as a complementary technique of interactive private query answering. Another question one may ask is whether *it really matters in practice* if existing data anonymization techniques satisfy only, say, k-anonymity, but not differential privacy. We argue that the answer is yes. Privacy breaches do occur and have serious consequences. A number of attacks have been publicized as a result of failed anonymization attempts. One of the most widely publicized was the AOL's release of search logs in 2006. Two New York Time journalists [4] were able to re-identify Thelma Arnold, a 62 year old women living in Lilburn, Ga. from the supposed anonymized search logs. This was deemed by CNNMoney to be one of the "101 Dumbest Moments in Business" and landed the company in a class action lawsuit. We show that existing data anonymization privacy notions fail to prevent similar disclosures that are recognized as serious privacy compromises.

While it is natural to apply differential privacy also to the microdata publishing domain, the existing results remain limited. Some are computationally expensive, such as [25, 6]. Other apply the noise-addition techniques to specific domains where one essentially publishes counts of certain items, such as keywords in search logs [23, 15, 18].

In this paper, we aim at developing sound theoretical foundations for microdata anonymization methods. We start by carefully examining the k-anonymity notion to find out why it fails to provide strong privacy protection. We first observe that the k-anonymity notion, even when all attributes are treated as quasi-identifiers, is unable to prevent re-identification. For example, the trivial method of randomly choosing some tuples from the input dataset and duplicating each chosen tuple k times satisfies k-anonymity, even though it publishes a significant portion of the input tuples unchanged, enabling easy re-identification of them. Furthermore, almost all k-anonymization methods that have been proposed in the literature are vulnerable because the way they compute the generation scheme to be applied to the input tuples makes the generalization overly dependent on tuples that contain extreme values, leaking information about these tuples.

One way to avoid the above vulnerability of existing k-anonymization methods is to use a generalization scheme that is independent of the input dataset. That is, the algorithm applies a fixed generation scheme to the input tuples and then suppresses any tuple that appears less than k times. Such a "safe" k-anonymization algorithm has no apparent privacy weaknesses, and intuitively provides some level of privacy protection, as each tuple is indeed "hiding in a crowd of at least k". Unfortunately, the algorithm still does not satisfy differential privacy, simply because the algorithm is deterministic. The desire to understand and formalize what kind of privacy protection is offered by such an algorithm is the initial motivation for this research.

The contributions of this paper are as follows:

- We prove that by adding a random sampling step, the above "safe" k-anonymization algorithm provides (ϵ, δ)-differential privacy with reasonable parameters.

  In the literature, k-anonymity and differential privacy have been viewed as very different privacy guarantees. k-anonymity is syntactic and weak, and differential privacy is algorithmic and provides semantic privacy guarantees. Ours is, to our knowledge, the first result that links k-anonymization with differential privacy. It illustrates that "hiding in a crowd of k" indeed offers privacy guarantees.

  We also observe that existing techniques for satisfying differential privacy rely almost exclusively on output perturbation, that is, adding noise to the query outputs. Our result suggests an alternative approach to satisfy differential privacy. Rather than adding noises to the output, one can add a random sampling step in the beginning and prune results that are too sensitive to changes of individual tuples (i.e., tuples that

violate k-anonymity).

- We show that random sampling has a big privacy amplification effect for ϵ-differential privacy. For example, applying an algorithm that achieves $(\ln 2)$-differential privacy on dataset sampled with 0.1 probability can achieve overall $(\ln 1.1)$-differential privacy. This result applies in interactive query answering as well as in microdata publishing.

  We note that random sampling is a step that is either already undertaken or can be naturally added in many microdata publishing scenarios. For example, the census bureau publishes a 1-percent microdata sample. Netflix can also use random sampling to select which tuples to be released.

- Finally, we note that the notion of (ϵ, δ)-differential privacy, when applied to microdata publishing, is somewhat problematic, because the probability that a privacy breach occurs is not bounded by δ, but rather by δn, where n is the number of tuples in the output dataset. While one can always choose a very small δ, this may result in datasets of poor quality. To address this problem, we introduce the notion of f-smooth (ϵ, δ)-differential privacy, where f is a function such that $f(1) = 1$ and f is monotonically decreasing. This notion essentially requires an algorithm to satisfy (ϵ, δ)-differential privacy for many pairs of (ϵ, δ)'s, and while ϵ increases, δ decreases. We show that this notion is composable and that it is satisfied by the approach of sampling + safe k-anonymization.

The rest of the paper is organized as follows. We give a brief introduction of different privacy in Section 2, and analyze the pitfalls of k-anonymity in Section 3. In Section 4 we prove our main result that random sampling + safe k-anonymization satisfies (ϵ, δ)-differential privacy and discuss its implications. We explore the privacy amplification benefit of random sampling in Section 5 and introduce our notion of f-smooth (ϵ, δ)-differential privacy in Section 6. Finally, we discuss related work in Section 7 and conclude in Section 8.

2. DIFFERENTIAL PRIVACY

Differential privacy formalizes the following protection objective: if a disclosure occurs when an individual participates in the database, then the same disclosure also occurs with similar probability (within a small multiplicative factor) even when the individual does not participate. More formally, differential privacy requires that, given two input datasets that differ only in one tuple, the output distributions of the algorithm on these two datasets should be close. There are two ways to interpret the notion that two datasets differ in only one tuple. The standard interpretation is that one dataset contains one more tuple than the other. An alternative is to have the two datasets contain the same number of tuples, with all but one being the same. In this paper we adopt the standard interpretation.

DEFINITION 1. ***[ϵ-Differential Privacy [10, 12] (ϵ-DP)]:*** *A randomized algorithm $\mathcal{A}$ gives ϵ-differential privacy if for any dataset D, any tuple $t \in D$, and any $S \in Range(\mathcal{A})$,*

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t}) = S]} \leq e^{\epsilon} \quad (1)$$

Note that in this paper we always assume that datasets such as D are multisets, i.e., the same tuple may appear multiple times. We use D_{-t} to denote the dataset with one copy of t removed from D, and define $0/0$ to be 1, and $p/0$ to be ∞ when $p > 0$.

We point out that because inequality 1 bounds the ratio on both sides, Definition 1 also ensures that for any tuple t', the ratio $\frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{+t'})=S]}$ is bounded in $[e^{-\epsilon}, e^{\epsilon}]$, where $D_{+t'}$ denotes the dataset resulted from adding t' to D.

Intuitively, ϵ-DP offers strong privacy protection. If $\mathcal{A}$ satisfies ϵ-DP, one can claim that publishing $\mathcal{A}(D)$ protects the privacy of every tuple t in D, because even if one leaves t out of the dataset, in which case the privacy of t can be considered to be protected, one may still publish S with a similar probability.

Differential privacy provides worst-case privacy guarantee in at least two senses. First, inequality (1) must hold for *every* dataset D and *every* tuple t in D, meaning that the privacy for every tuple in every dataset is protected. Second, inequality (1) requires that the e^{ϵ} bound holds for *every* possible outcome S, even if S occurs only with very low probability. These are desirable features, since as we have seen in past privacy incidents, compromising the privacy of a single individual may already be viewed unacceptable, and should be avoided [4, 26].

In practice, ϵ-DP can be too strong to be satisfiable in some scenarios. A commonly used relaxation is to allow the output S's that violate inequality (1) to occur with a small error probability δ. We use the following formulation.

DEFINITION 2 ((ϵ, δ)-DIFFERENTIAL PRIVACY ((ϵ, δ)-DP)). *A randomized algorithm $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy, if for any dataset D and any $t \in D$, the following holds with probability at least $1 - \delta$:*

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t}) = S]} \leq e^{\epsilon}$$

More precisely, let O denote the set of all S's in $Range(\mathcal{A})$ such that the above does not hold. Then $\Pr[\mathcal{A}(D) \in O] < \delta$ and $\Pr[\mathcal{A}(D_{-t}) \in O] < \delta$.

Some papers use another slightly weaker version of relaxation, which is implied by Definition 2. See, for example, [17] for more details.

The (ϵ, δ)-DP notion allows that for some output S, the ratio $\frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t})=S]}$ does not need to be bounded. In fact, it is possible that for some S, $Pr[\mathcal{A}(D) = S] > 0$ and $Pr[\mathcal{A}(D_{-t}) = S] = 0$. If any such S occurs as output, one can immediately tell that the tuple t is in the input. In other words, (ϵ, δ)-DP gives up one aspect of ϵ-DP's worst-case protection. Under (ϵ, δ)-DP, total privacy compromises for some tuples can occur; however, the total probability that these "bad" outputs occur is bounded by δ. In Section 6, we will explore the potential problem with this relaxation and propose a way to strengthen it.

3. AN ANALYSIS OF K-ANONYMITY

In this section, we examine k-anonymity and explore its limitations and potential merits as a principle for privacy protection.

3.1 Difficulties of Choosing QIDs

The development of k-anonymity was motivated by an early and well publicized privacy incident [30]. The Group Insurance Commission (GIC) published a supposedly anonymized dataset recording the medical visits of patients managed under the plan. While the obvious personal identifiers were removed, the published data included zip code, date of birth, and gender, which

are sufficient to uniquely identify a significant fraction of the population. Sweeney [30] showed that by correlating this data with the publicly available Voter Registration List for Cambridge Massachusetts, medical visits for many individuals can be easily identified, including those of William Weld, a former governor of Massachusetts. We note that even without access to the public voter registration list, the same privacy breaches can occur. Many individuals' birthdate, gender and zip code are public information. This is especially the case with the advent of social media, including Facebook, where users share seemingly innocuous personal information to the public. The GIC re-identification attack directly motivated the development of the k-anonymity privacy notion.

DEFINITION 3. ***[k-Anonymity] [30]:*** *A published table satisfies k-anonymity relative to a set of QID attributes if and only if when the table is projected to include only the QIDs, every tuple appears at least k times.*

A first problem with this definition is that it requires the division of all attributes into quasi-identifiers (QIDs) and sensitive attributes (SA), where the adversary is assumed to know the QIDs, but not SAs. This separation, however, is very hard to obtain in practice. Even though only some attributes are used in the GIC incident, it is difficult to assume that they are the only QIDs. Other attributes in the GIC data include visit date, diagnosis, etc. There may well exist an adversary who knows this information about some individuals, and if with this knowledge these individuals' record can be re-identified, it is still a serious privacy breach. The same difficulty is true for publishing any kind of census, medical, or transactional data. When publishing anonymized microdata, one has to defend against all kinds of adversaries, some know one set of attributes, and others know a different set. An attribute about one individual may be known by some adversaries, and unknown (and thus should be considered sensitive) for other adversaries.

This separation of QIDs and SAs has indirectly led to criticisms of k-anonymity and the development of privacy concepts that focus on attribute disclosure, such as ℓ-diversity [22] and t-closeness [20].

In this paper we consider a strengthened version of k-anonymity by treating all attributes as QIDs. This is stronger than using any subset of attributes as QIDs. This strengthened version of k-anonymity avoids making assumption about the adversary's background knowledge in terms of which attributes are known and what are not. This has been used in the context of anonymizing transaction data [16].

3.2 k-Anonymity Does Not Prevent Re-identification

The notion of k-anonymity is motivated by re-identification attacks, and intuitively it should provide some protection against re-identification, especially if we use the strengthened version that treats all attributes as QIDs. When satisfying this notion, each tuple in the output is blended in a group of at least k tuples that are the same. This follows the appealing principle that "privacy means hiding in a crowd". The intuition is that as there are at least $k-1$ other tuples that look exactly the same, one cannot re-identify which tuple in the output corresponds to an individual with probability over $1/k$. Unfortunately, this intuition turns out to be wrong. Consider the following class of algorithms.

ALGORITHM 1. ***[k-Duplication]:*** *First, select $1/k$'th portion of the input tuples by some method. For example, one method is to group the tuples into clusters of k tuples each, and randomly select one tuple in each cluster as the representative for that cluster. Then, duplicate each selected tuple k times.*

Obviously the outputs of any k-Duplication algorithm satisfies k-anonymity. However, it is very difficult to claim that it provides privacy protection, as it essentially publishes a portion of the input dataset unchanged. One can point to a group of k identical tuples and say which individual these tuples are from. One may object that the above algorithm is too artificial, and argue that practical k-anonymization algorithms are not subject to such re-identification attacks. To illustrate the weakness of k-anonymity more clearly, consider the following anonymization scheme, which represents several proposed algorithms for k-anonymity [7, 19].

ALGORITHM 2. ***[Clustering and Local Recoding (CLR)]:*** *First, group input tuples into clusters such that each cluster has at least k tuples. For example, one method of grouping is the Mondrian algorithm [19]. One could also use some clustering method based on some distance measurement (e.g., [7]). Then, for each tuple, replace each attribute value with a generalized value that represents all values for that attribute in the cluster.*

CLR algorithms are vulnerable when some tuples contain extreme values. Even if the output satisfies k-anonymity, the generalized value depends on the extreme values of some tuples; hence from the output an adversary can infer that one's tuple is in the dataset and can thus infer these values. For example, suppose the dataset records the net worth of some individuals in a town. Further suppose that it is known that only one individual in the town has net worth over \$10 million. When given a $(k = 20)$-anonymized output dataset containing one group of tuples that all have $[900K, 35M]$ as the generalized net worth value, what can one conclude? At least the following: the rich individual is in the dataset; the individual's tuple is in the group; and the individual's net worth is \$35 million. It would be difficult to say that because in the output dataset, there are at least 19 other tuples that are exactly the same, then the individual cannot be re-identified with probability $1/20$. The same attack applies to almost every existing k-anonymization algorithm.

We hope that these arguments are convincing enough for the conclusion that k-anonymity (even when all attributes are treated as QID) does not provide adequate protection. One natural question is: Is this because the intuition "hidden in a crowd of size at least k" fails to provide privacy protection, or is it because the definition of k-anonymity fails to correctly capture "hidden in a crowd of size at least k"?

We believe that the answer is the latter. And we now explain why. The notion of k-anonymity implicitly assumes that there is a one-to-one relationship g between the input tuples and the output tuples, i.e., given input D, the output dataset is $\{g(t) \mid t \in D\}$. When there are k output tuples that are the same, there must be k input tuples; hence any input tuple is hidden within a group of k input tuples. However, while almost all k-anonymization algorithms do have an implicit association relation between input tuples and output tuples, this association relation itself can be overly dependent on a few input tuples. For instance, consider the example above with the extreme value. All the 20 output tuples in the group with $[900K, 35M]$ obviously depend on the single input tuple with value $35M$. Hence it is not just $g(t_1)$ that contains information about t_1, but so does $g(t_2)$, $g(t_3)$, etc. As a result, satisfying only k-anonymity in the output does not prevent re-identification.

Another way to look at this limitation of the notion of k-anonymity is that it is *syntactic*, in that one only needs to check the syntactic properties of each output individually to ensure that it is satisfied; one does not need consider the behavior of the data anonymization algorithm on other inputs. Hence an output that satisfies k-anonymity can still reveal information about individual

tuples.

A natural conjecture is that if a k-anonymization algorithm uses a mapping that does not overly depends on any individual tuple, then such an algorithm provides some level of privacy protection. Indeed, trying to formalize this intuition was the original motivation of the results reported in the current paper.

3.3 A "Safe" k-Anonymization Algorithm

We now consider a "safe" k-anonymization algorithm, in which the mapping from input tuples to output tuples does not depend on the input dataset at all.

ALGORITHM 3. ***[Data-independent_Generalization + k-Suppression (k-DGS)]:*** *This algorithm uses a global recoding scheme (i.e., a generalization scheme) that does not depend on the particular input dataset D. It has two steps. In the first step, one applies this recoding scheme to each tuple in the input. In the second step, one suppresses any tuple that appear less than k times.*

The k-DGS algorithm's outputs satisfy k-anonymity. Intuitively this algorithm provides some level of privacy protection, and the level of privacy protection increases with larger values of k. If any individual's tuple is published, there must be at least $k-1$ other tuples in the *input* database that are the same under the recoding scheme; furthermore, the recoding scheme does not depend on the dataset, and one only sees the results of the recoding. Hence in this input dataset, the individual is hidden in a crowd of at least k.

Can one formally show that the k-DGS algorithm offers privacy protection? The fact that this algorithm satisfies k-anonymity is not helpful, since algorithms that obviously do not protect privacy also satisfy k-anonymity. At the same time, the k-DGS algorithm does not satisfy (ϵ, δ)-DP for any $\delta < 1$. The reason is simple. Let g be the global recoding scheme. Choose D such that it contains $n > k$ tuples t' such that $g(t') = g(t)$. Then, the k-DGS algorithm will output different numbers of $g(t)$ for D and D_{-t}. For D, the output would contain n copies of $g(t)$, and for D_{-t}, the output would contain $n-1$ copies of $g(t)$. Thus, the ratio of probabilities for an output with n copies of $g(t)$ is unbounded.

We note that an adversary is able to tell whether an output S is resulted from D or D_{-t} because the adversary knows exactly how many copies of t' are in D such that $g(t') = g(t)$. A natural consideration is: "Can we make the adversary's knowledge about this less certain?" Observing an output that contains, say, 25 copies of tuples $g(t)$, an adversary would be unable to tell whether the input is D or D_{-t} if he is uncertain whether D contains 25 or 26 copies of t' such that $g(t') = g(t)$.

4. K-ANONYMITY MEETS DIFFERENTIAL PRIVACY

We have shown that k-DGS does not satisfy (ϵ, δ)-DP, because (ϵ, δ)-DP assumes an adversary that knows precise knowledge of what other tuples are in D. A method to add uncertainty to what an adversary knows about the dataset is to first perform a random sampling step. This leads us to the following algorithm:

ALGORITHM 4. ***[β-Sampling + Data-independent_Generalization + k-Suppression (k, β)-SDGS]*** *The algorithm has three steps. In the first step, it samples from the input dataset with probability β, that is, each tuple in the input dataset is chosen with probability β. In the second step, it applies a data-independent generalization procedure to each tuple. In the third and final step, it suppresses any tuple that appears less than k times.*

Below, we show that this algorithm satisfies (ϵ, δ)-differential privacy for a small δ with reasonable values of k and β. We use $f(j; n, \beta)$ to denote the probability mass function for the binomial distribution; that is, $f(j; n, \beta)$ gives the probability of getting exactly j successes in n trials where each trial succeeds with probability β. And we use $F(j; n, \beta)$ to denote the accumulative probability mass function; that is, $F(j; n, \beta) = \sum_{i=0}^{j} f(j; n, \beta)$.

THEOREM 1. *The (k, β)-SDGS algorithm (when $0 < \beta < 1$) satisfies (ϵ, δ)-differential privacy for any $\epsilon \geq -\ln(1-\beta)$, and $\delta = d(k, \beta, \epsilon)$, where the function d is defined as*

$$d(k, \beta, \epsilon) = \max_{n: n \geq \left\lceil \frac{k}{\gamma} - 1 \right\rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta),$$

where $\gamma = \frac{(e^{\epsilon} - 1 + \beta)}{e^{\epsilon}}$.

PROOF. Let $\mathcal{A}$ denote the algorithm, and g be the data-independent generalization procedure in the algorithm. For any dataset D, any tuple $t \in D$, and for any output S. For any $\epsilon \geq -\ln(1-\beta)$, we want to compute the probability by which

$$e^{-\epsilon} \leq \frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t}) = S]} \leq e^{\epsilon} \tag{2}$$

is violated.

Let n be the number of t' in D such that $g(t') = g(t)$. Let j be the number of times that $g(t)$ appears in S. Note that as the only difference between D and D_{-t} is that D has one extra copy of t, we have.

$$\frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t})) = S]} = \frac{\Pr[\mathcal{A}(D) \text{ has } j \text{ copies of g(}t\text{)}]}{\Pr[\mathcal{A}(D_{-t}) \text{ has } j \text{ copies of g(}t\text{)}]}$$

Because any tuple that appears less than k times is suppressed, either $j \geq k$, or $j = 0$. When $j = 0$, we have

$$\frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t})) = S]} = \frac{F(k-1; n, \beta)}{F(k-1; n-1, \beta)} = \frac{\sum_{i=0}^{k-1} f(i; n, \beta)}{\sum_{i=0}^{k-1} f(i; n-1, \beta)}$$

Clearly, $F(k-1; n, \beta)$ is always less than $F(k-1; n-1, \beta)$;[1] hence $\frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t}))=S]} < e^{\epsilon}$. Furthermore, we note that $\forall i \in [0..k-1]$, $\frac{f(i;n,\beta)}{f(i;n-1,\beta)} = \frac{n(1-\beta)}{n-i} \geq (1-\beta)$. Hence $\frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t}))=S]} \geq (1-\beta)$. Because $\epsilon \geq -\ln(1-\beta)$, we have $e^{-\epsilon} \leq 1-\beta$; hence under the case when $j = 0$, inequality (2) is satisfied.

When $j \geq k$, we have

$$\frac{\Pr[\mathcal{A}(D) = S]}{\Pr[\mathcal{A}(D_{-t})) = S]} = \frac{f(j; n, \beta)}{f(j; n-1, \beta)} = \begin{cases} \frac{n(1-\beta)}{n-j} & n \geq j \\ 1 & n < j \end{cases}$$

The choice of n can be arbitrary because it is determined by the choice of D. The value of j is determined by the choice of S. For some values of j, inequality (2) is violated. We want to compute the probabilities of these bad j's occurring. From the above, we know when $j > n$, the outcome is good. We now consider the bad outcomes when $j \leq n$.

Note that because $\epsilon \geq -\ln(1-\beta)$, we have $-\epsilon \leq \ln(1-\beta)$, and

$$\frac{n(1-\beta)}{n-j} > 1 - \beta \geq e^{-\epsilon}.$$

Hence we only need to consider what j's make $\frac{n(1-\beta)}{n-j} > e^{\epsilon}$. This

[1] Let X_i's be random variables that take the value 1 with probability β, and 0 with probability $1-\beta$. $F(k-1; n-1, \beta)$ is the probability that the sum of $n-1$ such X's $\leq k-1$, and $F(k-1; n, \beta)$ is the probability that the sum of n such X's is $\leq k-1$.

occurs when $j > \frac{(e^\epsilon - 1 + \beta)n}{e^\epsilon}$. Let $\gamma = \frac{(e^\epsilon - 1 + \beta)}{e^\epsilon}$, then this occurs when $j > \gamma n$.

So far our analysis has shown that a bad outcome S for an input D would satisfy the condition $j \geq k$ and $n \geq j > \gamma n$. Now we need to compute the probability that $\mathcal{A}(D)$ gives a bad outcome, and the probability that $\mathcal{A}(D_{-t})$ gives a bad outcome. The former is given below:

$$\max_n \sum_{j:(j \geq k \wedge j > \gamma n)}^{n} f(j; n, \beta) \tag{3}$$

And the latter is

$$\max_n \sum_{j:(j \geq k \wedge j > \gamma n)}^{n-1} f(j; n-1, \beta)$$

As the latter is smaller than the former, we only need to bound the former.

Let $n_m = \left\lceil \frac{k}{\gamma} - 1 \right\rceil$, we now show that when $n \leq n_m$, $\sum_{j:(j \geq k \wedge j > \gamma n)}^{n} f(j; n, \beta)$ increases when n increases. Note that the choice of n_m satisfies the condition that $\gamma n_m < k$ and $\gamma(n_m + 1) \geq k$. Observe that when $n \leq n_m$, the condition $(j \geq k \wedge j > \gamma n)$ becomes $j \geq k$. The function $\sum_{j:j \geq k}^{n} f(j; n, \beta)$ is monotonically increasing with respect to n.

When $n \geq n_m$, the condition $(j \geq k \wedge j > \gamma n)$ becomes $j > \gamma n$. (In fact, when $n = n_m + 1$, the smallest j to satisfy $j > \gamma n$ is $k + 1$.) Hence the error probability is bounded by

$$\delta = d(k, \beta, \epsilon) = \max_{n: n \geq \left\lceil \frac{k}{\gamma} - 1 \right\rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta), \text{ where } \gamma = \frac{(e^\epsilon - 1 + \beta)}{e^\epsilon}.$$

□

The formula for the function $d(k, \beta, \epsilon)$ is quite complicated. We now examine it in more details. We want to find $n \geq \left\lceil \frac{k}{\gamma} - 1 \right\rceil$ that maximizes $\sum_{j > \gamma n}^{n} f(j; n, \beta)$. We first observe that $\gamma > \beta$ because

$$\gamma - \beta = \frac{(e^\epsilon - 1 + \beta)}{e^\epsilon} - \beta = \frac{(e^\epsilon - 1)(1 - \beta)}{e^\epsilon} > 0$$

That is, $\sum_{j > \gamma n}^{n} f(j; n, \beta)$ sums up the tail binomial distribution probabilities for the portion of the tail beyond γn, as shown in Figure 1. Following the intuition behind the law of large numbers, the larger the value of n, the smaller this tail probability. Note that the Chernoff bound of this tail probability decreases exponentially with respect to n. Hence intuitively, choosing the smallest value of n, i.e., $n = n_m = \left\lceil \frac{k}{\gamma} - 1 \right\rceil$ should maximize the formula. Unfortunately, due to the discrete nature of the binomial distribution, the maximum value may not be reached at n_m, but instead at one of the next few local maximal points $\left\lceil \frac{k+1}{\gamma} - 1 \right\rceil$, $\left\lceil \frac{k+2}{\gamma} - 1 \right\rceil$, $\cdots$. Thus we are unable to further simplify the representation of the function $d(k, \beta, \epsilon)$.

Below we show the numerical values of the function $d(k, \beta, \epsilon)$ for different parameters. The function d relates the four parameters $\epsilon, \beta, k, \delta$ by requiring $\delta = d(k, \beta, \epsilon)$. Note that the other requirement is that $\epsilon \geq -\ln(1 - \beta)$. Among the four parameters, ϵ and δ define the level of privacy protection, while k and β affect the quality of anonymized data.

In Table 1, we fix $k = 20$ and report the values of δ under different ϵ and β values. The table shows that the values of δ can be very small, which is what we need. We note that with fixed k and β, δ decreases as ϵ increases, which states that the error probability gets smaller when one relaxes the ϵ-bound on the probability ratio. In other words, the more serious a privacy breach, the more unlikely it occurs. We exploit this phenomenon in Section 6 to define f-smooth (ϵ, δ)-DP. The table also shows that with fixed k and ϵ, δ decreases as β decreases, meaning that a smaller sampling probability improves the privacy protection. In Section 5, we will see another instance of the effect of smaller sampling probability means better privacy protection.

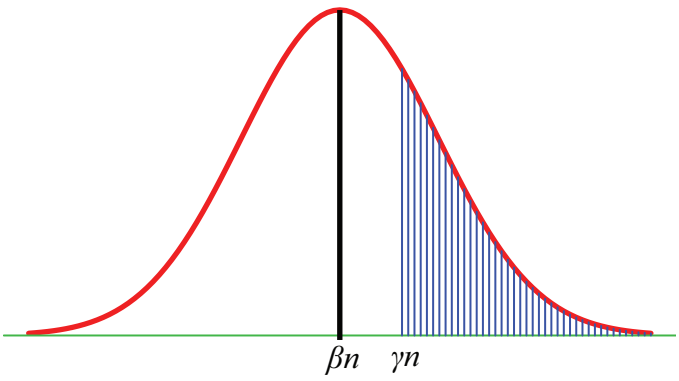

Figure 1: A graph showing the relationship between βn and γn on a binomial curve

In Figure 2, we show the results from examining the relationship between ϵ and δ when we vary $k \in \{5, 10, 20, 30, 50\}$ under fixed $\beta = 0.2$. We plot $\frac{1}{\delta}$ against ϵ for values of $\epsilon > -\ln(1 - \beta)$. The figure indicates a negative correlation between ϵ and δ. Furthermore, increasing k has a close to exponential effect of improving privacy protection. For example, when $\epsilon = 2$, increasing k by 10 roughly decreases δ by 10^{-5}.

In Figure 3, we show the results from examining the effect of varying $\beta \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$ under a fixed value of $k = 20$. This shows that decreasing β also dramatically improve the privacy protection. The two figures indicate the intricate relationship between privacy and utility.

In Figure 4, we explore this phenomenon that increasing k and decreasing β both improve privacy protection. Starting from $(k = 15, \beta = 0.05)$, each time we double β and find a value k that gives a similar level of privacy protection. We finds that k increases from 15 to 22 (for $\beta = 0.1$), 35 (for $\beta = 0.2$), and 60 (for $\beta = 0.4$).

In Figure 5, we examine the quality of privacy protection for very small k's (from 1 to 5). We choose a very small sampling probability of $\beta = 0.025$. Not surprisingly, when $k = 1$, the privacy protection is entirely from the sampling effect, as the obtained δ value is less than β. However, when $k \geq 2$, we start seeing privacy protection effect from k-anonymization, with δ (< 0.001) significantly smaller than $\beta = 0.025$ when $\epsilon = 2$.

Finally, in figure 6 we show the relationship between the privacy parameter ϵ and the utility parameter k if we set the requirement that $\delta \leq 10^{-6}$. The figure shows that smaller values of ϵ can be satisfied for larger values of k. Furthermore, the effect of β over ϵ is quite substantial.

Implications of Our Result. Theorem 1 shows that k-anonymization using a data-independent generalization approach, when preceded by a random sampling step, can satisfy (ϵ, δ)-DP with reasonable parameters. In the literature, k-anonymity and differential privacy have been viewed as very different privacy guarantees. k-anonymity is syntactic and weak, and differential privacy is algorithmic and provides semantic privacy guarantees. Our result is, to our knowledge, the first to link k-anonymization with differential privacy. This suggests that the "hiding in a crowd of k"

		ϵ					
		0.25	0.5	0.75	1.0	1.5	2.0
β	0.05	6.83×10^{-10}	2.50×10^{-14}	3.19×10^{-17}	1.76×10^{-19}	3.97×10^{-22}	2.00×10^{-24}
	0.1	4.19×10^{-06}	1.61×10^{-09}	3.44×10^{-12}	4.07×10^{-14}	3.22×10^{-16}	1.89×10^{-18}
	0.2	2.16×10^{-03}	8.02×10^{-06}	1.89×10^{-07}	6.03×10^{-09}	4.79×10^{-11}	1.59×10^{-12}

Table 1: A table showing the relationship between β and ϵ in determining the value of δ when k is fixed. In the above $k = 20$, and each cell in the table reports the value of δ under the given values of β and ϵ

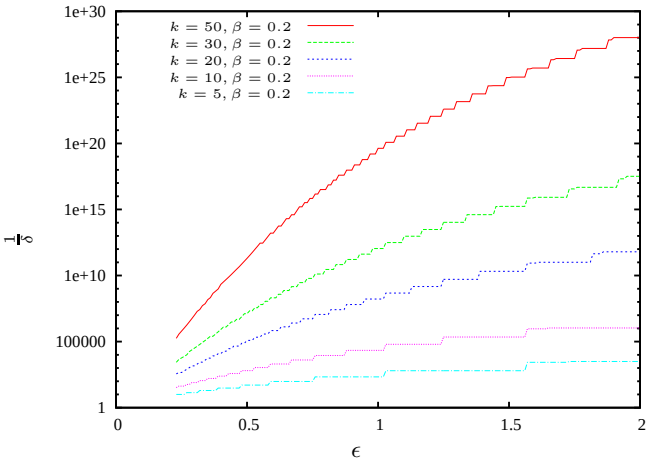

Figure 2: A graph showing the relationship between ϵ and $\frac{1}{\delta}$ if we vary the values of k under fixed β

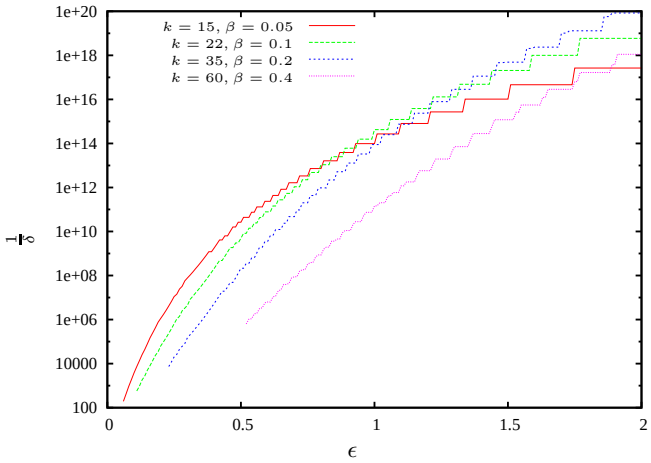

Figure 4: A graph showing the relationship between the values of k needed to achieve roughly the same δ if we double β.

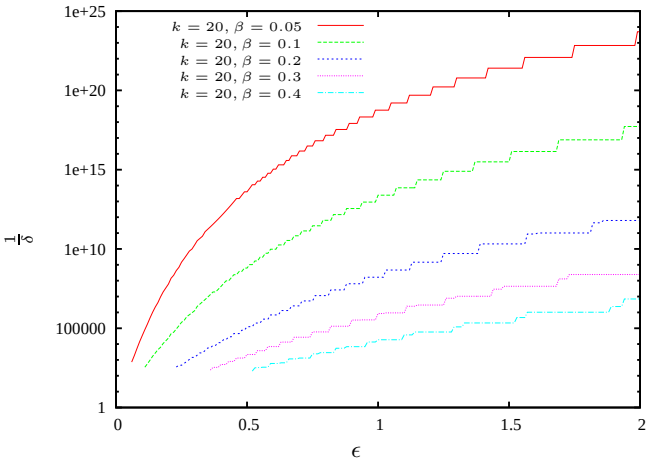

Figure 3: A graph showing the relationship between ϵ and $\frac{1}{\delta}$ if we vary the values of β under fixed k.

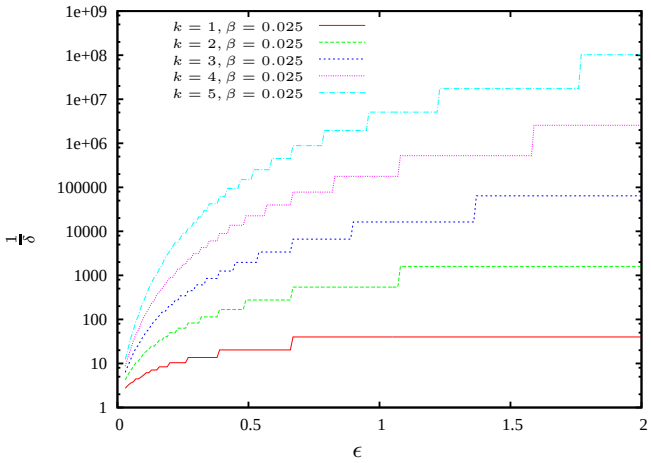

Figure 5: A graph showing the relationship between ϵ and $\frac{1}{\delta}$ with small k's, varying k and fixing β.

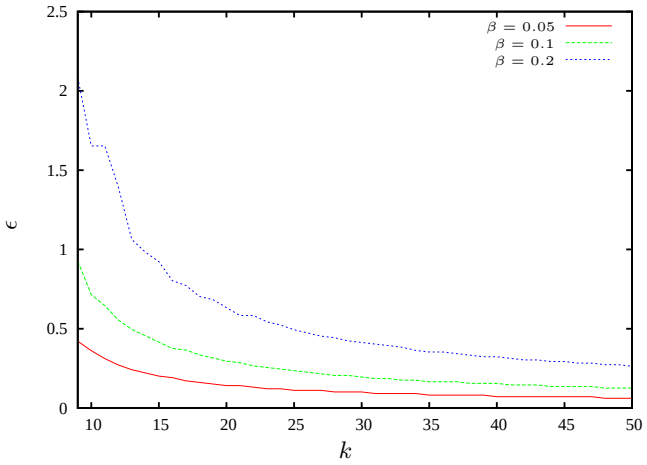

Figure 6: A graph showing the value of ϵ satisfied by a given k if $\delta \leq 10^{-6}$ with varying sampling probabilities.

privacy principle indeed offers some privacy guarantees when used correctly. We note that this principle is used widely in contexts other than privacy-preserving publishing of relational data, including location privacy and publishing of social network data, network packets, and other types of data.

We also observe that existing techniques for satisfying differential privacy rely almost exclusively on output perturbation, that is, adding noises to the query outputs. Our result suggests an alternative approach to satisfy differential privacy. Rather than adding noises to the output, one can add a random sampling step in the beginning and prune results that are too sensitive to changes of individual tuples (i.e., tuples that violate k-anonymity); this also achieves differential privacy. In Section 5, we show further application of this random sampling step for differential privacy. An intriguing question is whether other input perturbation techniques can be used to satisfy differential privacy as well.

Towards practical k-Anonymization. While Theorem 1 gives a microdata anonymization method that satisfies differential privacy, one may argue that the algorithm is impractical. Requiring the usage of a data-independent global generalization scheme may be too restrictive. Furthermore, it is well known that k-anonymity suffers from the curse of dimensionality [1]. We now discuss how these issues can be addressed. The comprehensive development of these ideas is beyond the scope of this paper.

First, rather than having to use a data-independent global generalization scheme, one could compute a generalization scheme using the dataset. One just needs to ensure that the generalization scheme does not depend too much on any individual tuple. Here is the sketch of an algorithm that satisfies (ϵ, δ)-DP. Given a dataset D, one first computes a generalization scheme in a way that satisfies ϵ-DP. One then samples from D with probability β, apples the generalization scheme to the sampled tuples, and finally suppresses any tuple that appears less than k times. Our proof of Theorem 1 can be extended to show that the overall algorithm satisfies (ϵ, δ)-DP for suitable parameters.

Second, the curse of dimensionality can be dealt with by vertical partitioning high-dimensional data [21]. The idea is to group attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. One can anonymize and publish the different columns separately. That is, one tuple is divided into several segments by the columns, and each segment is published in a different output table, with each segment treated as a new tuple. This reduces the dimensionality of the data. If the grouping of attributes into columns can be done in a way that satisfies differential privacy, each tuple affects at most a small number of columns, and each column is published in a way that satisfies differential privacy, then the overall output can satisfy differential privacy.

5. DIFFERENTIAL PRIVACY WITH UNCERTAIN BACKGROUND KNOWLEDGE

In the previous section, we illustrated how adding a sampling step can make a deterministic k-anonymization algorithm satisfy differential privacy, as it adds uncertainty to the adversary's background knowledge. Intuitively, if an algorithm satisfies (ϵ, δ)-DP with a random sampling step, it should provide privacy protection without the sampling step, when it is reasonable to assume that the adversaries do not have precise knowledge about all tuples in the dataset. We note that other k-anonymization techniques would not satisfy (ϵ, δ)-DP, even with the random sampling step.

In this paper, we introduce a new privacy definition, called $(\beta, \epsilon, \delta)$-Uncertain Differential Privacy ($(\beta, \epsilon, \delta)$-UDP for short) and show that it is composable. An algorithm satisfies $(\beta, \epsilon, \delta)$-UDP, if preceding it with a sampling step with probability β, it can satisfy (ϵ, δ)-DP. We allow $\delta = 0$, in which case one gets a variant of ϵ-DP. We believe that this notion can serve as a starting point for developing privacy notions when it is reasonable to assume the adversaries do not have precise knowledge of the dataset. Further development along this direction is future work we plan to pursue.

In this section, we prove one result related to the ($(\beta, \epsilon, \delta)$-UDP notion, that is, sampling can amplify privacy protection for a differentially private mechanism.

DEFINITION 4. *An algorithm $\mathcal{A}$ gives $(\beta, \epsilon, \delta)$-UDP if and only if the algorithm $\mathcal{A}^\beta$ gives (ϵ, δ)-DP, where $\mathcal{A}^\beta$ denotes the algorithm to first sample with probability β, and then apply $\mathcal{A}$. And when $\delta = 0$, we say the algorithm satisfies (β, ϵ)-UDP.*

We show that this notion of $(\beta, \epsilon, \delta)$-UDP is composable.

THEOREM 2. *Given $\mathcal{A}_1$ that satisfies $(\beta, \epsilon_1, \delta_1)$-UDP and $\mathcal{A}_2$ that satisfies $(\beta, \epsilon_2, \delta_2)$-UDP. Then $(\mathcal{A}_1; \mathcal{A}_2)$, where ; denotes concatenation, satisfies $(\beta, \epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-UDP.*

PROOF. For any $D, t, S = S_1; S_2$, $(\mathcal{A}_1; \mathcal{A}_2)(D) = S_1; S_2$ when $\mathcal{A}(D) = S_1$ and $\mathcal{A}_2(D) = S_2$. We have

$$\frac{\Pr[(\mathcal{A}_1;\mathcal{A}_2)^\beta(D)=S]}{\Pr[(\mathcal{A}_1;\mathcal{A}_2)^\beta(D_{-t})=S]} = \frac{\Pr[(\mathcal{A}_1^\beta(D);\mathcal{A}_2^\beta(D))=(S_1;S_2)]}{\Pr[(\mathcal{A}_1^\beta(D_{-t});\mathcal{A}_2^\beta(D_{-t}))=(S_1;S_2)]} = \frac{\Pr[\mathcal{A}_1^\beta(D)=S_1]}{\Pr[\mathcal{A}_1^\beta(D_{-t})=S_1]} \frac{\Pr[\mathcal{A}_2^\beta(D)=S_2]}{\Pr[\mathcal{A}_2^\beta(D_{-t})=S_2]}$$

The above follows from the fact that $\mathcal{A}_1^\beta$ and $\mathcal{A}_2^\beta$ are conditionally independent given D. Hence with probability at least $1-(\delta_1+\delta_2)$, we have

$$e^{-(\epsilon_1+\epsilon_2)} \leq \frac{\Pr[\mathcal{A}^\beta(D) = S]}{\Pr[\mathcal{A}^\beta(D_{-t}) = S]} \leq e^{(\epsilon_1+\epsilon_2)}$$

□

An interesting feature of the (β, ϵ)-UDP notion is that there is a connection between the privacy parameter ϵ and the sampling rate β. The following theorem shows that by employing a smaller sampling rate, one can achieve a strong privacy protection (i.e., a smaller privacy parameter ϵ).

THEOREM 3. *Any algorithm that satisfies (β_1, ϵ)-UDP also satisfies (β_2, ϵ')-UDP for any $\beta_2 < \beta_1$, where* $\epsilon' = \ln\left(1 + \left(\frac{\beta_2}{\beta_1}(e^\epsilon - 1)\right)\right)$.

PROOF. Let $\beta = \frac{\beta_2}{\beta_1}$. The algorithm $\mathcal{A}^{\beta_2}$ can be viewed as first sampling with probability β, then followed by applying the algorithm $\mathcal{A}^{\beta_1}$.

We use $\Lambda_\beta(D)$ to denote the process of sampling from D with sampling rate β. For any D, t, S, we have

$$\frac{\Pr[\mathcal{A}^{\beta_2}(D) = S]}{\Pr[\mathcal{A}^{\beta_2}(D_{-t}) = S]} = \frac{\sum_T \Pr[\Lambda_\beta(D) = T]\Pr[\mathcal{A}^{\beta_1}(T) = S]}{\sum_T \Pr[\Lambda_\beta(D_{-t}) = T]\Pr[\mathcal{A}^{\beta_1}(T) = S]} = \frac{Z}{X}$$

To analyze Z, we note that all the T's that resulted from sampling from D with probability β can be divided into those that do not contain t (i.e., t is not sampled), and those that contain t (i.e., t is sampled). For a T in the former case, we have

$$\begin{aligned}\Pr[\Lambda_\beta(D) = T] &= (1-\beta)\Pr[\Lambda_\beta(D) = T | t \text{ not sampled in } T] \\ &= (1-\beta)\Pr[\Lambda_\beta(D_{-t}) = T]\end{aligned}$$

For a T in the latter case, we have

$$\begin{aligned}\Pr[\Lambda_\beta(D) = T] &= \beta\Pr[\Lambda_\beta(D) = T | t \text{ sampled in } T] \\ &= \beta\Pr[\Lambda_\beta(D_{-t}) = T_{-t}]\end{aligned}$$

Hence we have

$$\begin{aligned}Z = &\sum_{T: t\notin T}(1-\beta)\Pr[\Lambda_\beta(D_{-t}) = T]\Pr[\mathcal{A}^{\beta_1}(T) = S] \\ &+ \sum_{T: t\in T}\beta\Pr[\Lambda_\beta(D_{-t}) = T_{-t}]\Pr[\mathcal{A}^{\beta_1}(T) = S] \\ = &\ (1-\beta)X + \beta\sum_{T'}\Pr[\Lambda_\beta(D_{-t}) = T']\Pr[\mathcal{A}^{\beta_1}(T'_{+t}) = S] \\ = &\ (1-\beta)X + \beta Y\end{aligned}$$

Next, we bound the following ratio:

$$\frac{Y}{X} = \frac{\sum_{T'}\Pr[\Lambda_\beta(D_{-t}) = T']\Pr[\mathcal{A}^{\beta_1}(T'_{+t}) = S]}{\sum_T \Pr[\Lambda_\beta(D_{-t}) = T]\Pr[\mathcal{A}^{\beta_1}(T) = S]}$$

That $\mathcal{A}$ satisfies (β_1, ϵ)-UDP means that $\forall_T \forall_t e^{-\epsilon} \le \frac{\Pr[\mathcal{A}^{\beta_1}(T_{+t}) = S]}{\Pr[\mathcal{A}^{\beta_1}(T) = S]} \le e^\epsilon$. Hence we have $e^{-\epsilon} \le \frac{Y}{X} \le e^\epsilon$, and therefore

$$1 - \beta(1 - e^{-\epsilon}) \le \frac{Z}{X} \le 1 + \beta(e^\epsilon - 1)$$

Hence we have $e^{-\epsilon'} \le \frac{Z}{X} \le e^{\epsilon'}$ for

$$\epsilon' = \max(\ln(1 + \beta(e^\epsilon - 1)), -\ln(1 - \beta(1 - e^{-\epsilon})))$$

We now show that $\epsilon' = \ln(1 + \beta(e^\epsilon - 1))$.

$$\begin{aligned}&\ln(1 + \beta(e^\epsilon - 1) \ge -\ln(1 - \beta(1 - e^{-\epsilon}) \\ \Leftrightarrow\ &1 + \beta(e^\epsilon - 1) \ge \frac{1}{1 - \beta(1 - e^{-\epsilon})} \\ \Leftrightarrow\ &(1 + \beta(e^\epsilon - 1))(1 - \beta(1 - e^{-\epsilon})) - 1 \ge 0 \\ \Leftrightarrow\ &(e^\epsilon + e^{-\epsilon} - 2)(\beta - \beta^2) \ge 0\end{aligned}$$

□

A natural corollary of the above theorem is the following.

COROLLARY 4. *Given an algorithm $\mathcal{A}$ that satisfies ϵ-DP, it satisfies (β, ϵ')-UDP for $\epsilon' = \ln(1 + \beta(e^\epsilon - 1))$.*

An equivalent way to write $\epsilon' = \ln(1 + \beta(e^\epsilon - 1))$ is

$$\frac{e^{\epsilon'} - 1}{e^\epsilon - 1} = \beta.$$

To see the effect of this. Consider the case with sampling probability $\beta = 0.1$, and $e^\epsilon = 2$, then $e^{\epsilon'} = 1.1$. Hence, when one publishes a randomly sampled dataset, then one can use much relaxed parameters. With a 0.1 sample rate, then an algorithm that achieves $(\ln 2)$-DP can achieve overall $(\ln 1.1)$-DP.

Discussions. We have shown that using random sampling as a preprocessing step can amplify the privacy protection power of any algorithm that satisfies ϵ-DP, or equivalently, reduce the potential information leakage. This can be very useful in the interactive setting as well as non-interactive setting. When one samples from a very large dataset and uses the sampled dataset to answer statistical queries, such as in the case of using sensus data, one can add much less noise to satisfy the same privacy requirements because of the sampling step.

We are unable to prove the result in Theorem 3 for the case that $\delta \neq 0$, because of the difficulty in dealing with how the error probability δ changes. How reducing the sample probability would boost the privacy protection with an error probability δ thus remains an open question.

6. SMOOTH ERROR BOUND

So far in this paper we have adopted the relaxation of ϵ-differential privacy by allowing an error probability δ. A natural question is how small does such δ need to be? It is known that in order for (ϵ, δ)-DP to provide strong-enough privacy protection, it must be that $\delta \ll \frac{1}{n}$, where n is the number of records in the dataset. In particular, when applying an algorithm that satisfies (ϵ, δ)-DP, the probability that a privacy breach occurs is not bounded by δ, and depends on the size of dataset. In this section, we analyze this issue. To achieve very small δ's one has to choose large k's, small β's, or both, which result in low-quality outputs. To avoid having to use very small δ's, we introduce an approach to make the (ϵ, δ) relaxation more robust by providing a smoother bound on the probability of bad outcomes.

Consider the following algorithm.

ALGORITHM 5. ***[δ-Sampling]:*** *The δ-sampling algorithm goes through each tuple t in the input dataset. For each t, it publishes t with probability δ, and suppress t with probability $1 - \delta$.*

The above algorithm satisfies $(0, \delta)$-DP. However, given a database of n tuples, each corresponding to one individual, this algorithm will publish on average $n\delta$ tuples unchanged. Suppose that n is 50 million, and $\delta = 10^{-7}$, then the algorithm publishes on average 5 tuples unchanged, completely compromising the privacy of these tuples. This example illustrates that satisfying (ϵ, δ)-DP alone does not guarantee that a privacy breach can occur with probability at most δ. In fact, it guarantees that for any tuple, a privacy breach can occur with probability at most δ. However, when there are many tuples, it may be that with high probability the privacy of *some* tuple will be breached.

We note that this issue of (ϵ, δ)-DP is because when a violation of the inequality $e^{-\epsilon} \le \frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t})=S]} \le e^\epsilon$ occurs, no bound is placed on $\frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D_{-t})=S]}$. In fact, there could be an S such that $\Pr[\mathcal{A}(D_1) = S] = \delta$ and $\Pr[\mathcal{A}(D_2) = S] = 0$, causing complete privacy compromise when S occurs. In many cases, when an algorithm outputs S that violates the e^ϵ bound, the ratio $\frac{\Pr[\mathcal{A}(D_1)=S]}{\Pr[\mathcal{A}(D_2)=S]}$ often does not shoot up to ∞, but rather climb gradually, with higher ratios increasingly unlikely.

We use this observation to propose a better method of relaxing ϵ-DP. In particular, we need a better bound on the error probability, δ, in order to ensure that a disclosure like the one previously mentioned does not occur. We thus introduce a smoother bound for δ as follows.

DEFINITION 5. ***[f-smooth (ϵ,δ)-Differential Privacy]*** *A randomized algorithm $\mathcal{A}$ gives f-smooth (ϵ_0,δ_0)-differential privacy for a function f such that $f(1)=1$ and f is monotonically decreasing, if and only if for any $\epsilon \geq \epsilon_0$, $\mathcal{A}$ satisfies $\left(\epsilon, \delta_0 f\left(\frac{\epsilon}{\epsilon_0}\right)\right)$-differential privacy.*

When f is a constant function, we get (ϵ,δ)-DP. Ideally, we want $f(x)$ to decrease fast and goes to 0 when x goes to ∞.

That an algorithm $\mathcal{A}$ satisfiers f-smooth (ϵ_0,δ_0)-differential privacy means that $\mathcal{A}$ provides (ϵ,δ)-differential privacy simultaneously for an infinite number of (ϵ,δ) pairs, including (ϵ_0,δ_0), $(1.1\epsilon_0, \delta_0 f(1.1))$, $(1.2\epsilon_0, \delta_0 f(1.2))$, and so on.

A benefit of f-smooth (ϵ,δ)-differential privacy is that it avoids total privacy disclosure when the function $f(y)\to 0$ when $y\to\infty$. In fact, the δ-sampling method in Algorithm 5 does not satisfy f-smooth (ϵ,δ)-differential privacy for any such f.

The following theorem shows that the notion of f-smooth differential privacy is composable.

THEOREM 5. *Given $\mathcal{A}_1$ that satisfies f_1-smooth (ϵ_1,δ_1)-DP, and $\mathcal{A}_2$ that satisfies f_2-smooth (ϵ_1,δ_1)-DP, $(\mathcal{A}_1;\mathcal{A}_2)$ satisfies $\max(f_1,f_2)$-smooth $(\epsilon_1+\epsilon_2,\delta_1+\delta_2)$-DP.*

PROOF. For any D and $t\in D$, we say that S_1 is a bad outcome for $\mathcal{A}_1$ if $e^{-x\epsilon_1} \leq \frac{\Pr[A_1(D)=S_1]}{\Pr[A_1(D_{-t})=S_1]} \leq e^{x\epsilon_1}$ is violated, and define bad outcomes for $\mathcal{A}_2$ similarly. We have,

$$\forall x\geq 1 \Pr\left[A_1(D)\in\{S_1 : S_1 \text{ bad for } \mathcal{A}_1\}\right] < \delta_1 f_1(x)$$

and

$$\forall x\geq 1 \Pr\left[A_2(D)\in\{S_2 : S_2 \text{ bad for } \mathcal{A}_2\}\right] < \delta_2 f_2(x)$$

Note that for any D and $t\in D$, any $x\geq 1$, and any (S_1,S_2):

$$\begin{aligned}\frac{\Pr[\mathcal{A}_1;\mathcal{A}_2(D)=(S_1;S_2)]}{\Pr[\mathcal{A}_1;\mathcal{A}_2(D_{-t})=(S_1;S_2)]} &= \frac{\Pr[\mathcal{A}_1(D)=S_1]\Pr[\mathcal{A}_2(D)=S_2]}{\Pr[\mathcal{A}_2(D)=S_2]\Pr[\mathcal{A}_2(D_{-t})=S_2]}\\ &= \frac{\Pr[\mathcal{A}_1(D)=S_1]}{\Pr[\mathcal{A}_2(D)=S_2]}\frac{\Pr[\mathcal{A}_2(D)=S_2]}{\Pr[\mathcal{A}_2(D_{-t})=S_2]}\end{aligned}$$

For any D and $t\in D$, and any $x\geq 1$, we say (S_1,S_2) is a bad outcome when the following is violated

$$e^{-x(\epsilon_1+\epsilon_2)} \leq \frac{\Pr[(\mathcal{A}_1;\mathcal{A}_2)(D)=(S_1;S_2)]}{\Pr[(\mathcal{A}_1;\mathcal{A}_2)(D_{-t})=(S_1;S_2)]} \leq e^{x(\epsilon_1+\epsilon_2)}$$

Observe that (S_1,S_2) is a bad outcome only when either S_1 is a bad outcome for A_1. Therefore,

$$\begin{aligned}&\Pr\left[(A_1;A_2)(D)\in\{(S_1;S_2) : (S_1;S_2) \text{ is a bad outcome}\}\right]\\ \leq\ & 1-\Pr\left[\mathcal{A}_1(D)\text{is not bad }\right]\Pr\left[\mathcal{A}_2(D)\text{ is not bad}\right]\\ \leq\ & 1-(1-\delta_1 f_1(x))(1-\delta_2 f_2(x))\\ \leq\ & \delta_1 f_1(x)+\delta_2 f_2(x)\\ \leq\ & (\delta_1+\delta_2)\max(f_1,f_2)(x)\end{aligned}$$

□

In what follows, we show that the (β,k)-SDGS algorithm satisfies f-smooth (ϵ,δ)-DP.

THEOREM 6. *The (k,β)-SDGS algorithm satisfies the f-smooth bound (ϵ_0,δ_0)-differential privacy, for any $\epsilon_0 \geq -\ln(1-\beta)$, $\delta_0 = e^{-k\left(\ln\left(\frac{\gamma(1)}{\beta}\right)-\frac{\gamma(1)-\beta}{\gamma(1)}\right)}$, and $f(z)=\frac{1}{\delta_0}e^{-k\left(\ln\left(\frac{\gamma(z)}{\beta}\right)-\frac{\gamma(z)-\beta}{\gamma(z)}\right)}$, where $\gamma(z)=\frac{(e^{\epsilon_0 z}-1+\beta)}{e^{\epsilon_0 z}}$.*

PROOF. Recall that the (k,β)-SDGS algorithm satisfies (ϵ,δ)-differential privacy with

$$\delta = d(k,\beta,\epsilon) = \max_{n:n\geq\left\lceil\frac{k}{\gamma}-1\right\rceil}\sum_{j>\gamma n}^{n} f(j;n,\beta),$$

where $\gamma=\frac{(e^{\epsilon}-1+\beta)}{e^{\epsilon}}$.

Note that we are overloading γ to also be a function $\gamma(z)=\frac{(e^{\epsilon_0 z}-1+\beta)}{e^{\epsilon_0 z}}$, since we want to study the behavior of δ with different values. Note that $\gamma(1)=\frac{(e^{\epsilon_0}-1+\beta)}{e^{\epsilon_0}}$, and $\gamma=\frac{(e^{\epsilon}-1+\beta)}{e^{\epsilon}}=\gamma\left(\frac{\epsilon}{\epsilon_0}\right)$.

In order to prove the f-smooth bound for a function f, we cannot just use numerical methods to compute the function d for different values. Instead, we use the following Chernoff bound as an upperbound of the above. Given X to be the sum of n independent binary random variables (i.e. for the binomial distribution), and let $\mu=\mathbb{E}[X]$, then the Chernoff bound states that we have

$$\Pr[X\geq(1+d)\mu] \leq e^{-\mu((1+d)\ln(1+d)-d)}$$

To make it more convenient to use the Chernoff bound, we note that

$$\delta \approx \max_{n:n\geq\frac{k}{\gamma}}\sum_{j\geq\gamma n}^{n} f(j;n,\beta).$$

We use the Chernoff bound to bound $\tau(n)=\sum_{j\geq\gamma n}^{n} f(j;n,\beta)$, in which case we have $\mu=\beta n$:

$$\begin{aligned}\tau(n) &= \textstyle\sum_{j\geq\gamma n}^{n} f(j;n,\beta)\\ &\leq \Pr\left[X\geq\left(1+\left(\frac{\gamma}{\beta}-1\right)\right)\beta n\right]\\ &\leq e^{-\beta n\left(\frac{\gamma}{\beta}\ln\left(\frac{\gamma}{\beta}\right)-\left(\frac{\gamma}{\beta}-1\right)\right)}\\ &= e^{-n\left(\gamma\ln\left(\frac{\gamma}{\beta}\right)-(\gamma-\beta)\right)}\end{aligned}$$

Note that $\left(\gamma\ln\left(\frac{\gamma}{\beta}\right)-(\gamma-\beta)\right) > 0$ for all positive γ and β's. (See, for example, http://www.wolframalpha.com/.) Hence the above function is monotonically decreasing when n increases. Hence

$$\begin{aligned}\delta &= d(k,\beta,\epsilon) =\\ &\leq e^{-\frac{k}{\gamma}\left(\gamma\ln\left(\frac{\gamma}{\beta}\right)-(\gamma-\beta)\right)}\\ &= e^{-k\left(\ln\left(\frac{\gamma}{\beta}\right)-\frac{(\gamma-\beta)}{\gamma}\right)}\end{aligned}$$

□

In order to better understand the above function, $f(z)$, we plot it for various values of ϵ_0. The resulting graph is shown in Figure 7. The value of δ was chosen to make $f(1)=1$. As expected, the function decays exponentially fast when $z>1$. As ϵ increases, the decay of the function is slower. We note that this observation only holds for values of $\epsilon>0.5$. For $\epsilon=0.25$, the curve cuts through the other curves. We are unsure of the exact reason underlying this behavior. However, given the complexity of the f function, this does not seem too strange.

7. RELATED WORK

The vast majority of the literature on privacy-preserving data publishing consider privacy notions that are weaker than differential privacy. These approaches typically assume an adversary that knows only some aspects of the dataset (background knowledge) and tries to prevent it from learning some other aspects. One can always attack such a privacy notion by changing either what the adversary already knows, or changing what the adversary tries to learn. The most prominent among these notions is k-anonymity [30, 29]. Some follow-up notions include l-diversity [22] and t-closeness [20]. In this paper, we analyze the weaknesses of k-anonymity in detail, and argue that a separation between QIDs and sensitive attributes are difficult to obtain

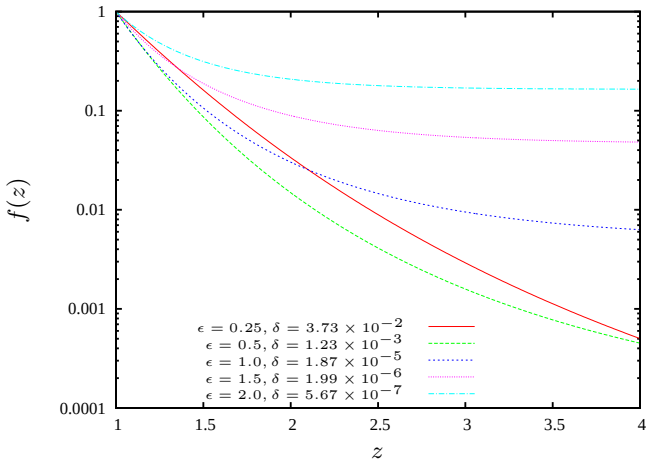

Figure 7: A graph showing the relationship between $f(z)$ and z for $(k = 20, \beta) - SDGS$ under the smooth bound

in practice, undermining the foundation of privacy notions such as l-diversity, t-closeness, and other ones centered on attribute disclosure prevention. Differential privacy and similar algorithmic privacy notions offer a more sound foundation for privacy-preserving data publishing.

From 2003 to 2006, the notion of differential privacy was developed in a series of works [9, 14, 5, 12, 10]. It represents a major breakthrough in privacy-preserving data analysis. Most results on differential privacy are for answering statistical queries, rather than publishing microdata. A survey on these results can be found in [11]. The seminal work of Dwork et al. [12] is based on output perturbation, i.e., adding noises to query results. This approach lends itself rather nicely to statistical results, but its does not solve the general microdata publishing problem. Our approach can be seen as complimenting this approach by considering a method of input perturbation rather than output perturbation to add uncertainty.

In an attempt to make differential privacy more amenable to more sensitive queries, several relaxations have been developed, including (ϵ, δ)-differential privacy [9, 14, 5, 12]. We use (ϵ, δ)-differential privacy in our paper. Moreover, we also point out its pitfalls when being applied in microdata publishing, and propose f-smooth (ϵ, δ)-differential privacy.

Random sampling [2, 3] has been studied as a method for privacy preserving data mining, where privacy notions other than differential privacy were used. To our knowledge, the only work that considered sampling as a method for differential privacy is that of Chaudhuri et al. [8]. In this work, however, sampling is used by itself, rather than as a pre-processing step to a "safe" k-anonymization algorithm or another differentially private algorithm. Therefore, in their results the disclosure probability δ is lower-bounded by the sampling probability, β. Our results show that sampling when combined with other methods can be much more powerful.

Another approach to privacy-preserving microdata publishing is data synthesizing. In [25], McSherry and Talwar proposed an exponential mechanism for releasing data with differential privacy. However, their mechanism is not feasible in practice because it takes time exponential to the size of the possible outputs. Blum et al. [6] considered synthetic data generation that is useful for a particular class of queries. Their approach uses the exponential mechanism [25] and therefore also suffers from the computational constraints. Recently, Dwork et al. [13] obtained a number of theoretical results on the boundary between computational feasibility and unfeasibility for different utility measures. Our work is more practical oriented, as we prove a class of anonymization algorithms satisfy (ϵ, δ)-DP.

There exists some work on publishing microdata while satisfy (ϵ, δ)-DP or its variant. Machanavajjhala et al. [23] introduced a variant of (ϵ, δ)-DP called (ϵ, δ)-probabilistic differential privacy and showed that it is satisfied by a synthetic data generation method for the problem of releasing the commuting patterns of the population in the United States. Korolova et al. [18] considered publishing search queries and clicks that achieves (ϵ, δ)-differential privacy. A similar approach for releasing query logs with differential privacy was proposed by Gotz et al. [15]. These approaches essentially apply the output perturbation technique in differential privacy to microdata publishing scenarios that can be reduced to histogram publishing at their core. Our work differs in the following. First, we do not use output perturbation. Second, we directly link k-anonymization to differential privacy.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we take a major step towards practical solutions for publishing microdata while providing rigorous privacy protection. In this research, we take the approach of starting from both k-anonymization and differential privacy and trying to meet in the middle. On the one hand, we identify weaknesses in the k-anonymity notion and existing k-anonymization methods and fix these weaknesses. On the other hand, we try to relax differential privacy so that it can be satisfied by the improved k-anonymization method. The key insight underlying our result is that random sampling can be used to bridge this gap between k-anonymization and differential privacy. This result is, to our knowledge, the first to link k-anonymity with differential privacy. A natural direction for future work is to develop more practical k-anonymization techniques that can be proven to satisfy differential privacy. One promising approach is to make existing k-anonymization algorithm "safe" so that they satisfy $(\beta, \epsilon, \delta)$-UDP. Then they satisfy (ϵ, δ)-DP when a random sampling step is added. One open problem is how to compute a global recoding scheme that works well for a dataset in a differentially privacy way. Alternatively, we are also looking at the problem of how to make partitioning-based schemes (such as Mondrian [19]) safe.

We also observe that existing techniques for satisfying differential privacy rely almost exclusively on output perturbation, that is, adding noise to the query outputs. Our result suggests that random sampling could play an important role in satisfying differential privacy as well. Not only it can enable a deterministic k-anonymization algorithm to satisfy (ϵ, δ)-DP, it can also amplify the privacy protection for algorithms that already satisfy ϵ-DP. We believe that the effect of random sampling, and perhaps other input perturbation methods, on differential privacy should be further investigated.

Finally, we observe that current definitions of (ϵ, δ)-differential privacy require δ to be very small to provide sufficient privacy protection when publishing microdata, making the notion impractical in some scenarios. We have introduced a notion called f-smooth (ϵ, δ)-differential privacy and showed that it can be satisfied by random sampling plus safe k-anonymization. It would be interesting to examine whether other methods that satisfy (ϵ, δ)-DP satisfy this smooth version of (ϵ, δ)-DP.

9. REFERENCES

[1] C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, pages 901–909, 2005.

[2] R. Agrawal, R. Srikant, and D. Thomas. Privacy preserving olap. In *SIGMOD*, pages 251–262, 2005.

[3] S. Agrawal and J. R. Haritsa. A framework for high-accuracy privacy-preserving mining. In *ICDE*, pages 193–204, 2005.

[4] M. Barbaro and J. Tom Zeller. A face is exposed for aol searcher no. 4417749. New York Times, Aug 2006.

[5] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, New York, NY, USA, 2005. ACM.

[6] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618, 2008.

[7] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *Proceedings of the 12th international conference on Database systems for advanced applications*, DASFAA'07, pages 188–200, 2007.

[8] K. Chaudhuri and N. Mishra. When random sampling preserves privacy. In *CRYPTO*, pages 198–213, 2006.

[9] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210, 2003.

[10] C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[11] C. Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[13] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.

[14] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *CRYPTO*, pages 528–544, 2004.

[15] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Privacy in search logs. *CoRR*, abs/0904.0682, 2009.

[16] Y. He and J. F. Naughton. Anonymization of set-valued data via top-down, local generalization. In *VLDB*, page ss, 2009.

[17] S. P. Kasiviswanathan and A. Smith. A note on differential privacy: Defining resistance to arbitrary side information. *CoRR*, abs/0803.3946, 2008.

[18] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW*, pages 171–180, 2009.

[19] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*, page 25, 2006.

[20] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, pages 106–115, 2007.

[21] T. Li and N. Li. Slicing: A new approach to privacy preserving data publishing. Accepted to appear in IEEE Transactions on Knowledge and Data Engineering (TKDE).

[22] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *ICDE*, page 24, 2006.

[23] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, pages 277–286, 2008.

[24] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the net. In *KDD*, pages 627–636, 2009.

[25] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

[26] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *S&P*, pages 111–125, 2008.

[27] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13:1010–1027, November 2001.

[28] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, 1998.

[29] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, 2002.

[30] L. Sweeney. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.