Preservation of Proximity Privacy in Publishing Numerical Sensitive Data

Jiexing Li Yufei Tao Xiaokui Xiao
Department of Computer Science and Engineering
Chinese University of Hong Kong
Sha Tin, New Territories, Hong Kong
{jxli, taoyf, xkxiao}@cse.cuhk.edu.hk

ABSTRACT

We identify *proximity breach* as a privacy threat specific to numerical sensitive attributes in anonymized data publication. Such breach occurs when an adversary concludes with high confidence that the sensitive value of a victim individual must fall in a short interval — even though the adversary may have low confidence about the victim's actual value.

None of the existing anonymization principles (e.g., k-anonymity, l-diversity, etc.) can effectively prevent proximity breach. We remedy the problem by introducing a novel principle called (ε, m)-anonymity. Intuitively, the principle demands that, given a QI-group G, for every sensitive value x in G, at most $1/m$ of the tuples in G can have sensitive values "similar" to x, where the similarity is controlled by ε. We provide a careful analytical study of the theoretical characteristics of (ε, m)-anonymity, and the corresponding generalization algorithm. Our findings are verified by experiments with real data.

ACM Categories and Subject Descriptors: H3.3 [Information Search and Retrieval]: Retrieval Models.

General Terms: Algorithms, Theory

Keywords: Privacy, Anonymization, Numeric, (ε, m)-anonymity

1. INTRODUCTION

Anonymized data publication has received considerable attention from the research community in recent years, due to the need of preventing "linking attacks" in numerous data-dissemination applications. Consider, for example, that a company wants to contribute its payment records in Table 1a, called the *microdata*, to sociology scientists. Attribute *Salary* is *sensitive*, that is, the publication must ensure that no adversary can accurately infer the salary of any employee. *Age* and *Zipcode* are *quasi-identifier* (QI) attributes, because they can be utilized in a linking attack to recover employees' identities. Assume that an adversary knows Andy's age 17 and Zipcode 12k. Given Table 1a, s/he ascertains that the first tuple must belong to Andy, and hence, Andy's salary must be 1000.

Andy

Age	Zip.	Salary
17	12k	1000
19	13k	1010
20	14k	1020
24	16k	50000
29	21k	16000
34	24k	24000
39	36k	33000
45	39k	31000

(a) The microdata

Group ID	Age	Zip.	Salary
1	[17,24]	[12k,16k]	1000
1	[17,24]	[12k,16k]	1010
1	[17,24]	[12k,16k]	1020
1	[17,24]	[12k,16k]	50000
2	[29,34]	[21k,24k]	16000
2	[29,34]	[21k,24k]	24000
3	[39,45]	[36k,39k]	33000
3	[39,45]	[36k,39k]	31000

(b) Generalization

Table 1: Privacy-preserving publication

Generalization [35, 36] is a popular methodology to thwart linking attacks. It divides the microdata into *QI-groups*, and then transforms the QI-values in each group to a uniform format. Table 1b demonstrates a generalized version of Table 1a (e.g., the age 17 of Andy, for instance, has been *generalized* to an interval $[17, 24]$). The generalization produces 3 QI-groups, as indicated by their group-IDs, such that the tuples in the same group are indistinguishable by the QI-attributes. Given Table 1b, the adversary mentioned earlier can no longer uniquely determine Andy's salary: any tuple of the first QI-group may belong to Andy; hence, his salary may be 1000, 1010, 1020 or 50000.

An anonymized table is considered "adequately protected", if it satisfies an *anonymization principle*. The existing principles for generalization include *k-anonymity* [35, 36], *l-diversity* [27] (and its variants [40, 42]), *variance control* [23], *t-closeness* [26], *(k, e)-anonymity* [44], *(c, k)-safety* [28], *privacy skyline* [12], and *δ-presence* [31]. They achieve different types of privacy protection; therefore, the choice of a principle depends on the needs of the underlying application.

1.1 Motivation: Proximity Breach

The motivation of this work is that, *none of the previous anonymization principles can prevent "proximity breach"*, which is a privacy threat specific to numerical sensitive attributes (such as *Salary* in Table 1b). Intuitively, proximity breach occurs when an adversary concludes with high confidence that the sensitive value of a victim individual must fall in a *short* interval — even though the adversary may have low confidence about the victim's actual value.

SIGMOD'08, June 9–12, 2008, Vancouver, BC, Canada.

For example, as explained earlier, an adversary possessing the QI-values of Andy is able to find out that Andy's record is in the first QI-group of Table 1b. Without further information, s/he assumes that each tuple in the group has an equal chance of being owned by Andy. Thus, s/he concludes that Andy's salary is in the interval $[1000, 1020]$ with 75% probability (although s/he only has 25% chance to discover Andy's real salary 1000). Equivalently, the adversary has arrived at a privacy-intruding claim: "Andy's salary is very likely around 1000".

Somewhat surprisingly, despite its apparent importance in practice, proximity breach has not been addressed in the literature. Specifically, no existing principle targets directly this type of privacy leakage. As a result, even if an anonymized table conforms to such a principle, it may still incur proximity breach. We will provide detailed explanations in Section 3.

1.2 Contributions

We introduce a new anonymization principle, (ε, m)-*anonymity*, which eliminates proximity breach in publishing numeric sensitive attributes. This principle is based on a natural rationale: *given a QI-group G, for every sensitive value x in G, at most $1/m$ of the tuples in G can have sensitive values "similar" to x.* The interpretation of "similarity" is quantified by the parameter ε. We discuss two interpretations that are especially useful in practice. The first one dictates that two values x and y are similar, if their absolute difference is at most ε, i.e., $|y - x| \leq \varepsilon$. The second interpretation, on the other hand, judges similarity in a relative sense: y is similar to x, if $|y - x| \leq \varepsilon \cdot x$.

The two interpretations lead to two instantiations of the proposed principle: absolute (ε, m)-anonymity and relative (ε, m)-anonymity, respectively. Note that ε and m define the degree of protection against proximity breach from different perspectives. The former parameter specifies, for each sensitive value x, the length of its private "neighborhood", whereas $1/m$ limits the probability that an adversary realizes x falling in that neighborhood. Apparently, stronger protection is achieved with a higher ε or higher m.

This paper presents a systematic study of proximity breach. First, we explain why the previous anonymization principles are inadequate for eliminating this privacy threat. Second, we present a careful theoretical analysis, which reveals several important characteristics of the proposed principle (ε, m)-anonymity. In particular, our results clarify the tradeoff between the extent of value similarity (captured by ε) and the risk of privacy breach (controlled by m), and lead to a reliable guideline for selecting ε and m in practice. Finally, utilizing our analytical findings, we develop an efficient algorithm for computing (ε, m)-anonymous generalization.

The rest of the paper is organized as follows. Section 2 clarifies privacy attacks and formulates the concepts underlying (ε, m)-anonymity. Section 3 points out the limitations of the previous anonymization techniques when they are applied to prevent proximity breach. Section 4 discusses the theoretical properties of (ε, m)-anonymity, and Section 5 elaborates the generalization algorithm. Section 6 experimentally evaluates the effectiveness of our solutions. Section 7 concludes the paper with directions for future work.

2. FORMALIZATION

Let T be a microdata table storing the private information of a set of individuals. T has d QI-attributes A_1,..., A_d, and a sensitive attribute (SA) S. We consider that S is numerical, and there is a linear ordering on the values of every QI-attribute A_i $(1 \leq i \leq d)$. The ordering is obvious for a numerical A_i. When A_i is categorical, the ordering juxtaposes, from left to right, the leaf values in the generalization taxonomy on A_i. All attributes have finite and positive domains. For each tuple $t \in T$, $t.A_i$ $(1 \leq i \leq d)$ denotes its value on A_i, and $t.S$ represents its SA value.

2.1 Privacy Attacks

We first clarify two fundamental concepts, and then illustrate the process taken by an adversary to deduce privacy.

DEFINITION 1. *A* **partition** *of a microdata table T is a set of* **buckets** *G_1, ..., G_g, where each bucket is a subset of T, and it holds that $G_1 \cup ... \cup G_g = T$ and $G_i \cap G_j = \emptyset$ for any $i \neq j$.*

DEFINITION 2. *Let $\{G_1, ..., G_g\}$ be a partition of the microdata table T. A* **generalization** *of T is a table T^* having the same schema as T. For each tuple $t \in T$, T^* contains a* **generalized tuple** *t^* such that $t^*.A_i$ $(1 \leq i \leq d)$ is an interval covering $t.A_i$ and $t^*.S$ equals $t.S$. Each G_j $(1 \leq j \leq g)$ defines a* **QI-group** *of T^*, which includes all the tuples generalized from the tuples in G_j. All tuples in the same QI-group are indistinguishable by their QI-values.*

Given T^*, an adversary may use it to infer the SA value $o.S$ of a *victim* individual o. We consider that the adversary possesses full identification information [28], as formulated next.

DEFINITION 3. *The* **background knowledge** *of an adversary includes (i) the identities (e.g., SSN) of the people in the microdata T, (ii) their exact QI-values, and (iii) the QI-group, denoted as G, in the generalized relation T^* that contains the record of the victim o.*

In practice, knowledge of Types (i) and (ii) in Definition 3 can be gleaned from an external database [28, 36]. For example, in the context of Table 1a, an external database can be the list of tax payers (every employee is definitely in the list). By examining the QI-values in T^*, the adversary can identify a small number of "candidate QI-groups" that can contain the record of o. By tackling knowledge of Type (iii), we are dealing with the unfortunate scenario where there is only one candidate QI-group.

After obtaining G (i.e., the QI-group in T^* including the tuple of o), the adversary proceeds to deduce the privacy of o with a probabilistic approach. Specifically, s/he considers that any tuple in G may belong to o with identical likelihood. Let X be a random variable modeling the distribution of $o.S$. After consulting G, the adversary's understanding about $o.S$ is a probability density function (pdf):

$$P[X = v] = n(G, v)/|G| \quad (1)$$

where $n(G, v)$ is the number of tuples in G whose SA values are v. For example, after realizing that Andy's record appears in the first QI-group of Table 1b, an adversary derives $P[X = v] = 25\%$, where X models the salary of Andy, and v can be any salary amount (i.e., 1000, 1010, 1020, or 50000) in that group.

2.2 (ε, m)-Anonymity

Let us associate each tuple t in the microdata T with an interval $I(t)$, indicating that we do not want the public to discover that its sensitive value $t.S$ is inside $I(t)$. We refer to $I(t)$ as the *private neighborhood* of t:

DEFINITION 4. *For a tuple $t \in T$, its* **private neighborhood** *$I(t)$ is a range in the domain of S that contains the sensitive value $t.S$ of t.*

As explained in Section 2.1, after a linking attack, an adversary derives a pdf, as in Equation 1, of a random variable X modeling $t.S$. Hence, s/he believes $X \in I(t)$ with a probability $\sum_{\forall v \in I(t)} P[X = v]$, which is the risk of breaching the proximity requirement enforced by $I(t)$. This risk can be represented in a closed form, as shown below:

DEFINITION 5. *Let t be a tuple in T, and G the QI-group in T^* that t is generalized to. The* **risk of proximity breach** *of t, denoted as $P_{brh}(t)$, equals $x/|G|$, where x is the number of tuples in G whose sensitive values fall in $I(t)$, and $|G|$ the size of G.*

For example, let T, T^* be Tables 1a and 1b respectively. Suppose that t is the tuple of Andy, and $I(t) = [900, 1100]$. Then, the breach risk $P_{brh}(t)$ of t equals 3/4. In the sequel, we use the term *proximity attack* to refer to a linking attack whose purpose is to infer the probability that the SA value of a tuple t is in its private neighborhood $I(t)$.

We now instantiate Definition 4 into two specific types of private neighborhoods, which are particularly important in practice, and can be specified with a single parameter ε.

DEFINITION 6. *For each tuple $t \in T$, its $I(t)$ is an* **absolute ε-neighborhood** *if*

$$I(t) = [t.S - \varepsilon,\ t.S + \varepsilon],$$

where ε is any non-negative value. Similarly, $I(t)$ is a **relative ε-neighborhood** *if*

$$I(t) = [t.S \cdot (1 - \varepsilon), t.S \cdot (1 + \varepsilon)],$$

where ε is a real value in $[0, 1]$.

By specifying an absolute (relative) ε-neighborhood on a tuple t, we indicate our willingness to allow the public to associate t with a sensitive value that has at least absolute (relative) error ε with respect to the real $t.S$.

We are ready to formalize (ε, m)-anonymity, and the problem of (ε, m)-anonymous generalization.

DEFINITION 7. *Given a real value ε and an integer $m \geq 1$, a generalized table T^* fulfills* **absolute (relative) (ε, m)-anonymity**, *if*

$$P_{brh}(t) \leq 1/m \tag{2}$$

for every tuple $t \in T$, where $P_{brh}(t)$ is the risk of proximity breach with $I(t)$ being the absolute (relative) ε-neighborhood.

PROBLEM 1. *Given a microdata table T, a pair of ε and m, the objective of* **absolute (relative) (ε, m)-anonymous generalization** *is to compute a generalization T^* of T that fulfills absolute (relative) (ε, m)-anonymity.*

By increasing the value of ε or m, we are strengthening the protection from proximity breach, however, in *different* ways. Specifically, the effect of raising ε is to enlarge the protection range of each sensitive value, whereas the purpose of elevating m is to lower an adversary's chance of beating that protection.

Some pairs of (ε, m), however, are not achievable by any generalization, i.e., Problem 1 has no solution for such a pair. For example, imagine a fairly large ε such that the absolute ε-neighborhood $I(t)$ of each tuple $t \in T$ covers the entire domain of the sensitive attribute. In this case, given any generalized table, the breach risk $P_{brh}(t)$ always equals 100%, rendering (ε, m) un-achievable for any $m > 1$.

In fact, there is an inherent conflict between ε and m, such that increasing either parameter may force the other to decrease. We refer to the phenomenon as the (ε, m)-*tradeoff*. Understanding the tradeoff is imperative to striking a good balance between the length of the protected neighborhood of a sensitive value and an adversary's chance of breaking the protection. We will analyze this issue in Section 4.

3. INADEQUACY OF THE EXISTING ANONYMIZATION METHODS IN PREVENTING PROXIMITY ATTACKS

This section reveals why the previous solutions to anonymized publication are not sufficient for Problem 1. Section 3.1 first discusses generalization, and then Section 3.2 analyzes perturbation.

3.1 Inadequacy of Known Generalization Principles

The privacy-preservation power of generalization is determined by the adopted anonymization principle. Next, we revisit the principles in the literature, and establish their limitations (in proximity-attack prevention) in two steps. First, for each principle, we will explain why proximity breach may still occur, even if the principle has been properly enforced. For this purpose, our analysis distinguishes these principles according to whether they are designed for categorical sensitive attributes (Section 3.1.1), or numeric ones (Section 3.1.2). Remember that our problem deals with numeric sensitive attributes. Second, in Section 3.1.3, we will argue that these principles cannot be easily adapted to avoid proximity breach.

3.1.1 Principles for Categorical Sensitive Attributes

***k*-anonymity [35, 36].** This is the first anonymization principle in the literature. It requires each QI-group to contain at least k tuples. Due to its pioneering nature, however, k-anonymity places no constraint on the SA (sensitive attribute) values in each QI-group. Absence of such constraints may result in a "homogenous" QI-group [27], where all tuples possess exactly the same SA value. The homogeneity offers virtually no protection against linking attacks: once an adversary realizes that a victim is in a homogeneous QI-group, s/he immediately becomes affirmative about the victim's precise SA value.

***l*-diversity [27] and Its Variants.** Prevention of homogeneity is equivalent to ensuring adequate diversity in the SA values of a QI-group. This is the motivation of l-diversity

[27], which demands at least l "well-represented" SA values in every QI-group. Focusing on categorical sensitive attributes, l-diversity aims at forbidding the public from discovering the *exact* SA value of a tuple.

A similar principle, called (α, k)-*anonymity*, is developed in [40]. Combining k-anonymity and l-diversity, (α, k)-anonymity dictates that, in every QI-group, (i) there are at least k-tuples, and (ii) at most α-percent of the tuples carry an identical SA value. On the other hand, *m-invariance* [42] is a principle originally proposed for re-publication of the microdata, after it has been updated with insertions and deletions. It is a stringent version of l-diversity. Specifically, it requires that each QI-group should have at least m tuples, all of which must have different SA-values.

(c, k)-safety [28] and Skyline-privacy [12]. This principle considers that an adversary may have the so-called *implicational knowledge*. Specifically, each *piece* of the knowledge says that, if an individual o_1 has an SA value v_1, then another individual o_2 has an SA value v_2. (c, k)-safety guarantees that even if an adversary has k pieces of such knowledge, s/he can successfully figure out the *precise* SA value of an individual with probability at most c. This principle achieves stronger privacy protection than l-diversity, because the latter guards only against adversaries with no implication knowledge. Aiming at the same privacy attack as (c, k)-safety (i.e., precise SA reconstruction), *Skyline-privacy* [12] extends (c, k)-safety by considering other types of background knowledge (one of which is implicational knowledge), and offers a way to tune the amount of privacy protection against each type.

Common Drawback of Principles for Categorial Attributes. The goal of all the above principles is to avoid exact sensitive-value reconstruction. This goal is reasonable for a categorical sensitive attribute, where different values do not have any sense of proximity. However, it is inappropriate for numerical sensitive attributes. Specifically, even if an adversary is able to re-build an SA-value with a very small error, this is still not considered as a privacy breach (by the above principles), while it already constitutes a proximity breach. As a concrete example, in Figure 1b, QI-group 1 has four different tuples, three of which are very close to each other. As explained in Section 1.1, equipped with the correct QI-values of Andy, an adversary declares that Andy's salary is in a short interval [1000, 1020] with a high probability 75%. However, the adversary has only a low chance of 25% to discover Andy's actual salary 1000, which is deemed acceptable by the previous principles.

3.1.2 Principles for Numeric Sensitive Attributes

(k, e)-anonymity [44]. Under this principle, each QI-group must have at least k different sensitive values, and the difference between the maximum and minimum values in the group must be at least e. Even with very large k and e, however, a (k, e)-anonymous QI-group may still incur proximity breach. For example, consider a QI-group with size k, where $k-1$ tuples have nearly identical (but still different) sensitive values, and the remaining tuple carries a faraway sensitive value to satisfy the requirement posed by e. Thus, for any of the first $k-1$ tuples, its risk of proximity-breach is as high as $(k-1)/k$, even if its private neighborhood is a short interval.

Variance Control [23]. This is a natural anonymization principle for numerical sensitive attributes. Specifically, it specifies a threshold t, and demands that, in every QI-group, the variance of the sensitive values must be at least t. Unfortunately, no matter how large the variance is, the QI-group may still suffer from proximity breach. Imagine a QI-group G, where $|G|-1$ tuples share the same sensitive value v. We may set the sensitive value of the remaining tuple to be sufficiently different from v, in order to acquire an arbitrarily large variance in G. The first $|G|-1$ tuples incur proximity-breach risk of at least $(|G|-1)/|G|$, regardless of their private neighborhoods.

***t*-closeness [26].** Let f be the distribution of sensitive values in the entire microdata T. The rationale of t-*closeness* [26] is that, in every QI-group G, the distribution f_G of sensitive values should mimic f. Specifically, $EMD(f, f_G)$ must not exceed t, where function $EMD(.)$ measures the *earth mover distance* [26] between f and f_G.

Unfortunately, EMD is not a reliable indicator of proximity-breach risk (given in Definition 5), because the former does not have provable mathematical relation to the latter. To illustrate this, assume a microdata table T with 6 tuples, whose sensitive values are {1, 2, 3, 4000, 5000, 6000}. Let G_1 and G_2 be two QI-groups in a generalization of T. They contain sensitive values {1, 3, 5000} and {2, 4000, 6000}, respectively. Both groups have the same EMD[1] to T. However, the values 1 and 3 in G_1 are more vulnerable to proximity breach than any value in G_2.

δ-presence [31]. This principle achieves privacy protection from a perspective different from all the above principles. Specifically, it prevents an adversary from inferring that an individual is in the microdata. Trivially, if the adversary is only δ (percent) sure that an individual is in the microdata, then any SA-value in the published table can belong to that individual with at most probability δ.

In our settings, however, we tackle the challenge that an adversary *already knows* that an individual is definitely in the microdata. This is indeed the case in many applications. For example, let the microdata be the tax database; then as long as a friend of Andy knows that Andy has a formal job, the friend is certain that Andy must have a record in the microdata. Nevertheless, it is worth mentioning that δ-presence is indeed an option for guarding against proximity attacks, in applications where an adversary does not know about the presence of any individual in the microdata.

3.1.3 Discussion

We are not aware of any simple approach of adapting the existing principles to prevent proximity attacks. There are three primary reasons. First, those principles are proposed for purposes drastically different from ours. This is especially true for the principles (Section 3.1.1) on categorical sensitive attributes. Second, many principles (for example, l-diversity, (c, k)-safety, skyline-privacy, to name just a few) are supported by a set of solid theory that is built upon the intrinsic properties of those principles. Modification of a principle can easily topple its underlying theory, and demand most of its properties to be re-established. This is

[1]Closed formulae for EMD calculation can be found in [26]. With those equations, it can be easily verified that both G and G' have an EMD of 0.1.

really not an easy task, given the complexity of the underlying theoretical derivation.

Third, all the principles reviewed earlier share a crucial feature: *monotonicity*. Namely, let G_1 and G_2 be two sets of tuples both of which fulfill a principle; then $G_1 \cup G_2$ definitely satisfies the principle, too. As proved in Section 5.1, however, (ε, m)-anonymity is *not* monotonic. The difference immediately invalidates the applicability of the previous anonymization algorithms [7, 17, 18, 24, 25, 43, 44] to achieving (ε, m)-anonymity. This fact justifies the necessity of studying the characteristics of (ε, m)-anonymity, and the corresponding generalization algorithm.

3.2 Inadequacy of Perturbation

This subsection discusses another popular methodology of data anonymization called *perturbation* [6, 15, 33]. We will show that although in theory this methodology can be applied to prevent proximity attacks, its applicability in practice is rather limited, since it usually entails considerable information loss.

Given a microdata T, perturbation creates a perturbed relation T' as follows. For each tuple $t \in T$, it generates a random number x between 0 and 1. If $x \leq p$, where p is a parameter called *retention probability*, the SA value $t.S$ of t is retained; otherwise, $t.S$ is replaced with (i.e., *perturbed* to) a random value in the domain of the sensitive attribute. After processing all the tuples in T, the resulting tuples constitute T', which is then published.

The privacy guarantee of perturbation has been very well studied [6, 15, 33]. Adapting the result in [6], it is easy to derive the breach risk $P_{brh}(t)$ (Definition 5) as:

$$P_{brh}(t) = \frac{P[I(t)] \cdot (p + (1-p) \cdot |I(t)|)}{P[I(t)] \cdot p + (1-p) \cdot |I(t)|}. \quad (3)$$

where $P[I(t)]$ is the percentage of the tuples in the microdata T whose sensitive values fall in $I(t)$, and $|I(t)|$ is the length of $I(t)$, assuming that the domain size of the sensitive attribute has been normalized to 1. In particular, note that $P[I(t)]$ depends on the data distribution of T. In general, a smaller p lowers the risk $P_{brh}(t)$, which is expected because a smaller p introduces more noise, and hence, loses more information.

To satisfy $P_{brh}(t) \leq 1/m$ (as required by Definition 7), the highest permissible p varies for different tuples t (call that p as the *best p of t*), as it depends on the SA-value distribution around the SA-value $t.X$ (due to the effect of $P[I(t)]$), and $|I(t)|$ (which may change for different t in the case of relative ε-neighborhood; see Definition 6). Apparently, the publisher must adopt a value of p that cannot exceed the lowest best p of all tuples.

The drawback of this approach is that, even under practical settings, the p that can be used by the publisher must be very low, in which case the perturbed relation T' is too noisy for data analysis. Imagine, for example, that the SA values in the microdata T follow a uniform distribution from 0 to 1. Suppose that our objective is to achieve absolute $(0.1, 4)$-anonymity, namely, $I(t)$ has length 0.1 for every tuple t, and $P_{brh}(t)$ must not exceed 1/4. As $|I(t)| = 0.1$, we know $P[I(t)] = 0.1$ (due to uniformity), and thus, it is easy to obtain that p must be below 7%. In other words, more than 93% of the SA values in the perturbation relation T' are random noise, rendering T' useless for research (as shown in [6], the retention probability p should be at least 20% to produce a T' useful for mining).

Another advantage of our technique over perturbation is that, we can still guarantee adequate privacy protection even when perturbation has failed completely. To illustrate, still assume a T with a uniform distribution on the sensitive attribute, but this time, we want to ensure absolute $(0.1, 5)$-anonymity. It can be verified that p will have to be 0 for perturbation, i.e., *all* the SA values in T' are noise. In fact, according to Theorem 1 (to be presented in Section 4.2), our solution can actually achieve absolute $(0.1, 10)$-anonymity (as will be clear later, the $n/maxsize(T)$ in Theorem 1 evaluates to 10 here), let alone $(0.1, 5)$-anonymity which is less stringent.

4. CHARACTERISTICS OF (EPSILON, M) ANONYMITY

We proceed to study the "(ε, m)-tradeoff" mentioned at the end of Section 2.2. Our results serve two crucial purposes:

- Permit a publisher to determine whether a target level of privacy protection is reachable. For example, if we want to apply an absolute (or relative) ε-neighborhood for each salary amount, is it possible to limit the risk of proximity breach to 10%? If not, what is the lowest possible risk limit? Conversely, if the breach risk must be controlled at 10%, what is the longest absolute (relative) private neighborhood that can be offered to each salary?
- Provide the theoretical foundation for designing an efficient algorithm for finding a good (ε, m)-anonymous generalization.

4.1 A Reduction

Interestingly, although absolute and relative (ε, m)-anonymity are based on different private neighborhoods, they are instances a generic form of proximity privacy: (e_1, e_2, m)-anonymity. The next two definitions formally elaborate this notion.

DEFINITION 8. *For each tuple t in the microdata T, its private neighborhood $I(t)$ is an* **(e_1, e_2)-neighborhood**, *if $I(t) = [t.S - e_1, t.S + e_2]$, where $t.S$ is the SA (sensitive value) of t, and e_1, e_2 two non-negative values.*

DEFINITION 9. *Give two non-negative values e_1, e_2 and an integer $m \geq 1$, a generalized table T^* fulfills* **(e_1, e_2, m)-anonymity**, *if $P_{brh}(t) \leq 1/m$ for every tuple $t \in T$, where $P_{brh}(t)$ is the risk of proximity breach (given in Definition 5) with $I(t)$ being the (e_1, e_2)-neighborhood.*

PROBLEM 2. *Given a microdata table T, and the values of e_1, e_2, m, the objective of* **(e_1, e_2, m)-anonymous generalization** *is to compute a generalization T^* of T that fulfills (e_1, e_2, m)-anonymity.*

Both the absolute and relative versions of Problem 1 are a special case of Problem 2. Obviously, an absolute (ε, m)-anonymous generalization is (e_1, e_2, m)-anonymous, where $e_1 = e_2 = \varepsilon$. Next, we provide a reduction that transforms relative (ε, m)-anonymous generalization to (e_1, e_2, m)-anonymous generalization, where $e_1 = \log \frac{1}{1-\varepsilon}$ and $e_2 = \log(1 + \varepsilon)$. All the "log" in this paper have a base of 2. For example, if the goal is relative $(0.2, 2)$-anonymity,

we can aim at achieving $(0.32, 0.26, 2)$-anonymity, where $0.32 = \log \frac{1}{1-0.2}$ and $0.26 = \log(1+0.2)$.

Given a microdata table T, we convert each SA value v in T to $\log v$. In this way, T is transformed into an alternative table T'. Let T'^* be a $(\log \frac{1}{1-\varepsilon}, \log(1+\varepsilon), m)$-anonymous generalization of T'. For every SA value v in T'^*, we replace it with 2^v. The replacement changes T'^* into a different table T^*. The next lemma shows that T^* is definitely a relative (ε, m)-anonymous generalization of the original microdata T.

LEMMA 1. *For each $t \in T$, $P_{brh}(t) \leq 1/m$, where $P_{brh}(t)$ is calculated from the T^* obtained as described earlier, with $I(t)$ being the relative ε-neighborhood of t.*

PROOF. Let t' be the tuple in T' that is converted from t, and $I(t')$ its (e_1, e_2)-neighborhood with $e_1 = \log \frac{1}{1-\varepsilon}$ and $e_2 = \log(1+\varepsilon)$. Use G' (G) to denote the QI-group in T'^* (T^*) that t' (t) is generalized to. By (e_1, e_2, m)-anonymity, there are at most $|G'|/m$ SA values v in G' that lie in $[t'.S - e_1, t'.S + e_2]$. Hence, 2^v falls in $[2^{t'.S-e_1}, 2^{t'.S+e_2}]$, which is exactly the relative ε-neighborhood $I(t)$ of t. It follows that no more than $|G|/m$ SA values in G are covered by $I(t)$, thus completing the proof. □

The above reduction allows us to analyze the (ε, m)-tradeoff for both absolute and relative (ε, m)-anonymity in a unified framework. In the next two subsections, we will study the characteristics of (e_1, e_2, m)-anonymity, and then, extend the results to (ε, m)-anonymity in Section 4.4.

4.2 Achievable Range of m Given e_1 and e_2

Next, we answer the following question: given a pair of (e_1, e_2), what is the condition on m such that Problem 2 has at least one solution? Let n be the cardinality of the microdata T. For any tuple $t \in T$, $I(t)$ represents its (e_1, e_2)-neighborhood.

DEFINITION 10. *Given a subset G of T, the* **left covering set** *of $t \in G$, denoted as $left(t, G)$, is the set of tuples in G whose SA values are at most $t.S$, and are covered by $I(t)$.*

Given G, the **right covering set** *of $t \in G$, denoted as $right(t, G)$, is the set of tuples in G whose SA values are at least $t.S$, and are covered by $I(t)$.*

As a running example, let T be the microdata in Table 1a, which has eight tuples with SA values (in ascending order) $t_1.S$ =1000, $t_2.S$ =1010, $t_3.S$ =1020, $t_4.S$ =16000, $t_5.S$ =24000, $t_6.S$ =31000, $t_7.S$ =33000, and $t_8.S$ =50000. Let G be the whole T, and assume $e_1 = 20$ and $e_2 = 10000$. Hence, $left(t_3, T) = \{t_1, t_2, t_3\}$, because the SA values of t_1, t_2, t_3 fall in the range of $[t_3.S - e_1, t_3.S] = [0, 1020]$. Similarly, $right(t_3, T) = \{t_3\}$ because no other tuple in T has an SA value in the range $[t_3.S, t_3.S + e_2] = [1020, 11020]$.

For every tuple $t \in G$, we deploy $size(t, G)$ to indicate the cardinality of the more sizable set between $left(t, G)$ and $right(t, G)$, namely:

$$size(t, G) = \max\{|left(t, G)|, |right(t, G)|\}. \quad (4)$$

In our running example, since (as mentioned earlier) $|left(t_3, T)| = 3$ and $|right(t_3, T)| = 1$, we have $size(t_3, T) = \max\{3, 1\} = 3$.

DEFINITION 11. *For any subset G of T, its* **maximum covering-set size** *$maxsize(G)$ is the largest $size(t, G)$ of all tuples $t \in G$. Formally, $maxsize(G) = \max_{\forall t \in G} size(t, G)$.*

In the running example, $maxsize(T) = 3$, because no other tuple $t \in t$ has a $size(t, T)$ higher than $size(t_3, T)$, which equals 3 as explained before. Maximum covering-set size has an interesting property:

LEMMA 2. *Let G_1 and G_2 be two disjoint subsets of T, and $G = G_1 \cup G_2$. Then:*

$$\frac{maxsize(G)}{|G|} \leq \max\{\frac{maxsize(G_1)}{|G_1|}, \frac{maxsize(G_2)}{|G_2|}\}. \quad (5)$$

PROOF. We first show $maxsize(G) \leq maxsize(G_1) + maxsize(G_2)$. Due to symmetry, assume $t \in G_1$, and that $maxsize(G)$ is the size of the left covering set $left(t, G)$ of a tuple $t \in G$. Use S_1 (S_2) to denote the set of tuples in $left(t, G)$ that also belong to G_1 (G_2). Obviously $left(t, G) = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$. Let t' be the tuple in S_2 with the largest SA value. Notice that $S_1 \subseteq left(t, G_1)$ and $S_2 \subseteq left(t', G_2)$. Therefore, $maxsize(G) = |S_1| + |S_2| \leq |left(t, G_1)| + |left(t', G_2)| \leq maxsize(G_1) + maxsize(G_2)$.

Given any subset G of T, we define $\alpha(G) = maxsize(G)/|G|$, and $\alpha(G_1)$, $\alpha(G_2)$ in the same manner. As $maxsize(G) \leq maxsize(G_1) + maxsize(G_2)$, we have $(|G_1| + |G_2|) \cdot \alpha(G) = |G_1| \cdot \alpha(G_1) + |G_2| \cdot \alpha(G_2)$, leading to $\frac{|G_1|}{|G_2|} \cdot (\alpha(G) - \alpha(G_1)) + \alpha(G) \leq \alpha(G_2)$. If $\alpha(G) \leq \alpha(G_1)$, Lemma 2 already holds. If $\alpha(G) > \alpha(G_1)$, the term $\frac{|G_1|}{|G_2|} \cdot (\alpha(G) - \alpha(G_1)) > 0$; hence $\alpha(G) < \alpha(G_2)$. □

In the running example (with $e_1 = 20$ and $e_2 = 10000$), let $G_1 = \{t_1, t_2\}$, and $G_2 = \{t_3, t_4, ..., t_8\}$. Clearly, $G_1 \cup G_2$ equals the entire microdata T. It is easy to verify that $maxsize(G_1) = size(t_1, G_1) = 2$ and $maxsize(G_2) = size(t_5, G_2) = 3$. Hence, the right hand side of Inequality 5 is $\max\{\frac{2}{2}, \frac{3}{6}\} = 1$. As mentioned earlier, $maxsize(T) = 3$; hence, the left hand side of the equality is $\frac{3}{8}$, which indeed bounded by the right hand side.

Based on the previous lemma, we establish an important theorem:

THEOREM 1. *Given a pair of e_1 and e_2, T has at least one (e_1, e_2, m)-anonymous generalization, if and only if $m \leq \lfloor n/maxsize(T) \rfloor$.*

PROOF. The proof consists of two steps. Step 1: Here, the goal is to show that, if $m > \lfloor |T|/maxsize(T) \rfloor$, no generalization T^* can satisfy (e_1, e_2, m)-anonymity. Assume that T^* is created from a partition $\{G_1, ..., G_g\}$ of T (see Definition 2). Hence, $T = \bigcup_{i=1}^{g} G_i$. As a directly corollary of Lemma 2, $maxsize(T)/|T| \leq \max_{i=1}^{g} maxsize(G_i)/|G_i|$. Without loss of generality, assume $maxsize(T)/|T| \leq maxsize(G_1)/|G_1|$. Let t be the tuple in G_1 such that $size(t, G_1) = maxsize(G_1)$. Thus, the proximity-breach risk $P_{brh}(t)$ of t is at least $size(t, G_1)/|G_1| = maxsize(G_1)/|G_1|$, which is at least $maxsize(T)/|T| > 1/m$.

Step 2: Let $g = maxsize(T)$ and $m_{max} = \lfloor |T|/g \rfloor$; the objective is to prove that, as long as $m \leq m_{max}$, there exists at least an (e_1, e_2, m)-generalization T^* of T. Let us sort the tuples in T in ascending order of their SA values, and use t_i $(1 \leq i \leq n)$ to denote the i-th tuple in the sorted list. A crucial observation is that, for any two tuples t_i and t_j, if

$|i-j| \geq g$, $t_i.S$ does not fall in $I(t_j)$, and likewise, $t_j.S$ does not fall in $I(t_i)$.

We divide T into g disjoint buckets G_1, G_2, ..., G_g by assigning tuples into buckets in a round-robin fashion. Specifically, tuple t_i $(1 \leq i \leq n)$ is added to G_j, where $j = (i \bmod g) + 1$. The assignment guarantees that, in any bucket $G \in \{G_1, ..., G_g\}$, (i) there is no tuple whose SA value lies in the (e_1, e_2)-neighborhood of another tuple in G, and (ii) $|G| \geq m_{max}$.

Now, take a generalization T^* of T based on the g buckets obtained earlier. T^* fulfills the property that, given any tuple $t \in T$, $P_{brh}(t) = 1/|G| \leq 1/m_{max}$, where G is the QI-group that t is generalized to. □

In our running example (where $e_1 = 20$ and $e_2 = 10000$), we already know that $maxsize(T) = 3$. Hence, the above theorem indicates that $(20, 10000, m)$-anonymity can be achieved, if and only if $m \leq \lfloor 8/3 \rfloor = 2$.

Finding the Maximum m. At first glance, $maxsize(T)$ seems to depend on both e_1 and e_2, whereas it actually relies on only the larger of e_1 and e_2. We present this observation as a formal lemma.

LEMMA 3. *Let e_{max} equal $\max\{e_1, e_2\}$, and G be any subset of T. Regardless of the concrete values of e_1 and e_2, as long as e_{max} is fixed, $maxsize(G)$ remains the same.*

PROOF. Assume that the tuples in G have been sorted in ascending order of their SA values. Use t_i $(1 \leq i \leq |G|)$ to denote the i-th tuple in the sorted list. Next we will prove the lemma for the case $e_1 \geq e_2$, as the extension to the symmetric case is straightforward.

For any $right(t_i) = \{t_i, ..., t_j\}$ $(1 \leq i \leq j \leq |G|)$, $t_i.S$ is also covered by $I(t_j)$, which implies $right(t_i) \subseteq left(t_j)$. Hence, $maxsize(G)$ is determined by the sizes of left covering sets, which, in turn, are determined by $e_{max} = e_1$. Therefore, e_2 has no influence on $maxsize(G)$. □

To illustrate, recall that in our running example, when $e_1 = 20$ and $e_2 = 10000$, $maxsize(T) = 3$. The above lemma implies that $maxsize(T)$ must also be 3, if e_1 and e_2 swap their values. Indeed, given $e_1 = 10000$ and $e_2 = 20$, $maxsize(T) = size(t_1, T) = right(t_1, T) = 3$.

Based on the above lemma, we develop an algorithm *find-maxsize*, as in Figure 1 for obtaining the $maxsize(G)$ of any subset G of T. Provided that the tuples in G have been sorted in ascending order of their SA values, the algorithm terminates in $O(|G|)$ time.

Now we are ready to settle the question raised at the beginning of this subsection. Given e_1, e_2 and m, we invoke *find-maxsize* by setting G to T and e_{max} to $\max\{e_1, e_2\}$. Then, according to Theorem 1, an (e_1, e_2, m)-anonymous generalization exists, if and only if $m \leq \lfloor |T|/maxsize(T) \rfloor$.

4.3 Achievable e_1 and e_2 Given m

This subsection deals with a problem opposite to the one solved in Section 4.2. Specifically, if we must put an upper bound of $1/m$ to the risk of proximity breach, what are the conditions on e_1 and e_2 so that there is at least one (e_1, e_2, m)-anonymous generalization of the microdata T? In the sequel, we use t_i $(1 \leq i \leq n)$ to denote the tuple having the i-th largest SA value in the microdata T. For any tuple $t \in T$, $I(t)$ is its (e_1, e_2)-neighborhood.

Algorithm *find-maxsize* (G, e_{max})
/* the tuples in G have been sorted in ascending order of their SA values */
1. $maxsize = 1$; $i = 1$; $j = 2$
2. while $(j \leq |G|)$
3. if $t_j.S - t_i.S \leq e_{max}$
/* $t_i.S$ falls in $I(t_j)$ or $t_j.S$ falls in $I(t_i)$ */
4. j++; $maxsize$++
5. else
6. j++; i++
7. return $maxsize$

Figure 1: Computation of $maxsize(G)$

Algorithm *find-e-max* (T, m)
/* the tuples in T have been sorted in ascending order of their SA values */
1. $h = \lfloor |T|/m \rfloor$
2. $e_{max} = \infty$
3. $i = 1$; $j = h + 1$
4. while $(j \leq |T|)$
5. if $(e_{max} > t_j.S - t_i.S)$
6. $e_{max} = t_j.S - t_i.S$
7. i++; j++
8. return e_{max}

Figure 2: Finding the upper bound of e_1, e_2

THEOREM 2. *Given a positive integer m, T has at least one (e_1, e_2, m)-anonymous generalization, if and only if*

$$\max\{e_1, e_2\} < \min_{i=1}^{n-h}(t_{i+h}.S - t_i.S) \quad (6)$$

where $h = \lfloor n/m \rfloor$.

PROOF. Let α be the right hand side of Inequality 6. We establish the theorem in two steps. Step 1: This step will show that, if $e_1 \geq \alpha$, no (e_1, e_2, m)-anonymous generalization of T exists. The extension to the symmetric case where $e_2 \geq \alpha$ is straightforward. Assume, without loss of generality, that α is minimized when i equals j. If $e_1 \geq \alpha$, $t_j.S$ is covered by $I(t_{j+h})$, meaning that $left(t_{j+h})$ includes at least $h + 1$ elements t_j, t_{j+1}, ..., t_{j+h}. Hence, $maxsize(T)$, given in Definition 11, is strictly larger than $h = \lfloor n/m \rfloor$ (notice that $maxsize(T) \geq |left(t_{j+h})|$). It follows that, $m > \lfloor n/maxsize(T) \rfloor$. By Theorem 1, T has no (e_1, e_2, m)-anonymous generalization.

Step 2: Here, the goal is to prove that, as long as $\max\{e_1, e_2\} < \alpha$, we can always find a generalization T^* of T, such that $P_{brh}(t) \leq 1/m$ for any tuple $t \in T$. Observe that, when $\max\{e_1, e_2\} < \alpha$, for any tuples t_x, t_y with $|x - y| \geq h$, $t_x.S$ and $t_y.S$ are not covered by $I(t_x)$ and $I(t_y)$, respectively. Thus, we create a partitioning of T with h buckets G_1, ..., G_h by assigning t_i $(1 \leq i \leq n)$ to G_j, where $j = (i \bmod h) + 1$. This assignment ensures that, in any bucket $G \in \{G_1, ..., G_h\}$, (i) there is no tuple whose SA value lies in the (e_1, e_2)-neighborhood of another tuple in G, and (ii) $|G| \geq \lfloor n/h \rfloor \geq m$. Compute a generalization T^* of T based on the h buckets obtained earlier. Thus, given any $t \in T$, it holds that $P_{brh}(t) \leq 1/|G| \leq 1/m$. □

In other words, after m has been decided, T has (e_1, e_2, m)-anonymous generalization, if and only if both e_1

and e_2 are smaller than a certain "upper bound", which equals the right hand side of Inequality 6. Earlier in Section 4.2, we know that, in our running example, when $e_1 = 20$ and $e_2 = 10000$, (e_1, e_2, m)-anonymity is possible only for $m \leq 2$. Next, let us utilize the above theorem to understand why m cannot reach 3.

Assume $m = 3$. Then, h (in Theorem 1) equals $\lfloor 8/3 \rfloor = 2$. For convenience, let us juxtapose the SA-values in the microdata of our running example in ascending order here: $\{t_1.S = 1000, t_2.S = 1010, t_3.S = 1020, t_4.S = 16000, t_5.S = 24000, t_6.S = 31000, t_7.S = 33000, t_8.S = 50000\}$. In the summation of Inequality 6, for $i = 1$, $t_{i+h}.S - t_i.S = t_3.S - t_1.S = 20$; similarly, given $i = 2$, $t_{i+h}.S - t_i.S = t_4.S - t_2.S = 14900$, and so on. Thus, it is easy to see that the right hand side of the inequality evaluates to 20 (i.e., the minimum is achieved at $i = 1$). Hence, to achieve $(e_1, e_2, 3)$-anonymity, both e_1 and e_2 have to be strictly smaller than 20. This is the reason why, given $e_1 = 20$ and $e_2 = 10000$, m must be smaller than 3.

Finding the Upper Bound of e_1 and e_2. Leveraging Theorem 2, Figure 2 presents an algorithm for computing this upper bound efficiently. Specifically, after the tuples in T have been sorted in ascending order of their SA values, the algorithm terminates in $O(|T|)$ time.

4.4 Selecting the Parameters of (ϵ, m)-Anonymity

Now let us return to (ε, m)-anonymity, and explain how a publisher can utilize the previous results (on (e_1, e_2, m)-anonymity) to decide the parameters ε and m. We distinguish three situations.

Both ε and m Decided. A publisher may already have clear preferences for the length ε of a private neighborhood and the limit $1/m$ of breach risk. In this case, it needs to check whether the (ε, m)-pair is achievable. This can be done easily with Theorem 1. Specifically, if the publisher wants to enforce absolute (ε, m)-anonymity, it can set both e_1 and e_2 directly to ε. The original (ε, m)-anonymity is achievable, if and only if m qualifies the condition in Theorem 1, where $maxsize(T)$ can be obtained with the algorithm *find-maxsize* in Figure 1. On the other hand, in case relative (ε, m)-anonymity is the target, the publisher needs to transform the microdata T to an alternative table T', following the reduction described in Section 4.1. Then, it can equate e_1 to $\log \frac{1}{1-\varepsilon}$, e_2 to $\log(1+\varepsilon)$, obtain $maxsize(T')$. Then, (ε, m)-anonymous generalization of T is possible, if and only if m qualifies Theorem 1 on T'.

ε Decided; Seek m. In this scenario, a publisher has decided the length ε of a private neighborhood, and aims at curbing proximity-breach risk under a bound $1/m$ that is as low as possible. Minimization of $1/m$ is equivalent to maximization of m. This can also be accomplished with Theorem 1. If absolute (ε, m)-anonymity is needed, the largest possible m equals $\lfloor |T|/maxsize(T) \rfloor$, where $maxsize(T)$ is computed with $e_1 = e_2 = \varepsilon$. For relative (ε, m)-anonymity, the maximum m is $\lfloor |T'|/maxsize(T') \rfloor$, where T' is reduced from T, and $maxsize(T')$ is obtained with $e_1 = \log \frac{1}{1-\varepsilon}$ and $e_2 = \log(1+\varepsilon)$.

m Decided; Seek ε. Sometimes the publisher is obliged to ensure a certain upper bound $1/m$ on breach risk. In this situation, it tries to elongate private neighborhoods as much as possible, i.e., maximizing ε. For absolute (ε, m)-anonymity, ε can be set arbitrarily close to the value e_{max} returned by the algorithm *find-e-max* in Figure 2. For relative (ε, m)-anonymity, ε can infinitely approach $1 - 1/2^{e_{max}}$, where e_{max} is returned by *find-e-max* on the T' reduced from T.

5. GENERALIZATION ALGORITHM

The essence of computing a generalization T^* is to decide a partition of T (c.f. Definition 1). As mentioned in Definition 2, each bucket G in the partition determines a QI-group in T^*. The final generalized QI-values in G can be obtained following different strategies. For example, on each QI-attribute A_i $(1 \leq i \leq d)$, the generalized value may simply be the minimum bounding interval of the A_i-values of all the tuples in G [25]; alternatively, one may also require that the generalized value should align with a pre-determined hierarchy on A_i [7].

The quality of a generalization is gauged by a metric of data loss, denoted as *loss*. Specifically, given a generalized tuple $t^* \in T^*$, $loss(t^*)$ returns the amount of information lost by t^*. The objective, therefore, is to minimize $loss(T^*) = \sum_{\forall t^* \in T^*} loss(t^*)$, i.e., the total information loss in generalizing all the tuples in T. Numerous metrics have been proposed in the literature (see a summary in [22] and the references therein). In the following subsections, we will present an algorithm for finding a generalization T^* with small $loss(T^*)$.

5.1 Non-Monotonicity and Predictability

All the privacy preserving principles surveyed in Section 3 have an important property: *monotonicity*. Specifically, this property says that if (the QI-groups decided by) two disjoint subsets G_1 and G_2 of T fulfill a principle, then the union $G_1 \cup G_2$ also satisfies the principle. Monotonicity is the prerequisite of an efficient top-down pruning paradigm for computing a generalization, which underlies nearly all the existing generalization algorithms [7, 17, 24, 25].

Unfortunately, (ε, m)-anonymity does not possess this property. We formally establish this fact with a lemma.

LEMMA 4. *Neither absolute nor relative (ε, m)-anonymity is monotonic.*

PROOF. We prove only the absolute case, since the relative case can be established similarly. It suffices to find a counter-example, where two subsets G_1 and G_2 of T are (absolute) (ε, m)-anonymous, but their union is not. Here is such an example: $\varepsilon = 15$, $m = 2$; G_1 and G_2 contain SA values $\{40, 60\}$ and $\{50, 80\}$ respectively. □

We propose a new concept, *predictability*, which is imperative to designing a fast generalization algorithm for principles disobeying monotonicity.

DEFINITION 12. *An anonymization principle is* **linearly predictable** *if, given any subset G of the microdata T, it is possible to determine in $O(|G|)$ time whether G is* **generalizable**, *i.e, if any generalization of G fulfills that principle.*

Linear predictability requires that we can quickly obtain a yes-or-no answer about the "generalizability" of G, as opposed to its concrete generalization.

LEMMA 5. *Both absolute and relative (ε, m)-anonymity are linearly predictable.*

PROOF. By the reduction in Section 4.1, establishment of the lemma is equivalent to proving that (e_1, e_2, m)-anonymity is linearly predictable. Note that Theorem 1 holds, even if we replace T with any subset G of T. Hence, we can determine whether G has at least one (e_1, e_2, m)-anonymous generalization, after computing $maxsize(G)$ using the algorithm *find-maxsize* (Figure 1). The algorithm terminates in $O(|G|)$ time. □

The above lemma makes a weak assumption: the tuples in the given subset G (whose generalizability is being determined) have been sorted in ascending order of their SA values (as is demanded by *find-maxsize*). As will be elaborated in the next subsection, during the entire process of computing an (ε, m)-anonymous generalization, we need to perform sorting only once, even though the generalizability of numerous subsets must be examined.

5.2 The Algorithm

This section elaborates how to obtain (absolute and relative) (ε, m)-anonymous generalizations of the microdata T. In fact, it suffices to discuss only (e_1, e_2, m)-anonymity computation, due to the reduction in Section 4.1.

After sorting the tuples in T in ascending order of their SA values, our algorithm proceeds in two steps: *splitting* and *partitioning*. The purpose of the splitting step is to reduce the lengths of the generalized QI-values, whereas that of the partitioning phase is to produce the final (e_1, e_2, m)-anonymous partitions. Next, we discuss each phase in detail.

Splitting. This step is motivated by the Mondrian algorithm in [25], and yields a partition of T. Each bucket G of the partition, however, does not necessarily qualify (e_1, e_2, m)-anonymity. Nevertheless, in case G does not, we guarantee that there always exists a way to further partition G into smaller subsets, each of which fulfills (e_1, e_2, m)-anonymity.

The splitting algorithm runs in iterations, and maintains a set S of buckets. Initially, S contains a single bucket that is simply T itself. Then, each partition divides a bucket $G \in S$ into two generalizable buckets (Definition 12) G_1, G_2, removes G from S, and adds G_1, G_2 to S. The algorithm stops when no more such G can be found.

Splitting a bucket G (into G_1, G_2) is performed based on a QI-attribute $A \in \{A_1, A_2, ..., A_d\}$. Specifically, G_1 (G_2) includes all the tuples in G whose A-values are at most (strictly larger than) v, where v is the median of the A-values of all the tuples in G. Among all the QI-attributes, the splitting dimension is the one that (i) leads to non-empty generalizable G_1 and G_2, and (ii), among all dimensions qualifying (i), minimizes the total information loss in generalizing G_1 and G_2 (measured by the metric *loss* presented at the beginning of Section 5).

A bucket split can be accomplished in $O(|G| \cdot d + \lambda \cdot d)$ expected time, where λ is the cost of evaluating *loss*, and is independent of the generalization algorithm. Specifically, for each QI-dimension A_i $(1 \leq i \leq d)$, we invoke the quick-select algorithm [16] to obtain the median A_i-value v in G, the cost of which is $O(|G|)$ in expectation. Then, with a single scan of G, we assign each tuple in G to G_1 or G_2, by comparing its A_i-value to v. Since the order that tuples are inserted to G_1 (G_2) coincides with the order they are scanned in G, they remain sorted in ascending order of their SA values in G_1 (G_2) — recall that the initial bucket T is sorted. Next, we determine the generalizability of G_1 and G_2 in $O(G)$ time (see Lemma 5). If both are generalizable, the quality of generalizing G_1 and G_2 is calculated in $O(\lambda)$ time. After deciding the split dimension, another scan on G is executed to produce the final G_1 and G_2.

Algorithm *anonymity-check* (G, e_1, e_2, m)
/* the tuples in G have been sorted in ascending order of their SA values */
1. $x = 1$; $i = 1$; $j = 2$
2. while ($j \leq |G|$)
3. if $t_i.S < t_x.S - e_1$ then i++ and goto line 2
4. if $t_j.S \leq t_x.S + e_2$ then j++
5. if $j > |G|$ or $t_j.S > t_x.S + e_2$
6. $P_{brh}(t_x) = (j - i)/|G|$
7. if $P_{brh}(t_x) > 1/m$ then return false
/* (e_1, e_2, m)-anonymity is violated */
8. else x++
9. return true

Figure 3: Checking (e_1, e_2, m)-anonymity

Partitioning. Given any bucket G in the set S output by the previous step, the partitioning phase converts it to one or more subsets of T that fulfill (e_1, e_2, m)-anonymity. Towards this purpose, we first check whether G itself satisfies (e_1, e_2, m)-anonymity. If yes, G is retained directly, and no other subsets are created. The checking can be carried out in $O(|G|)$ time, using the algorithm *anonymity-check* in Figure 3.

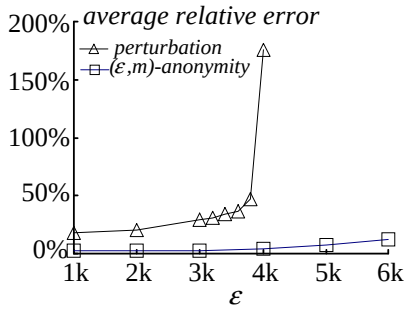
If G violates (e_1, e_2, m)-anonymity, we will partition it into g smaller subsets G_1, G_2, ..., G_g, where g equals $maxsize(G)$, calculated in $O(|G|)$ time by the algorithm *find-maxsize* in Figure 2. The partitioning is achieved following the idea behind the second step in the proof of Theorem 1. Specifically, we scan the tuples in G in ascending order of their SA values, assign the i-th $(1 \leq i \leq |G|)$ tuple to G_j, where $j = (i \bmod g) + 1$. All of the resulting G_1, G_2, ..., G_g are guaranteed to obey (e_1, e_2, m)-anonymity (see the proof of Theorem 1). As no sorting is required to obtain the scanning order, the entire assignment finishes in $O(|G|)$ time.

6. EXPERIMENTS

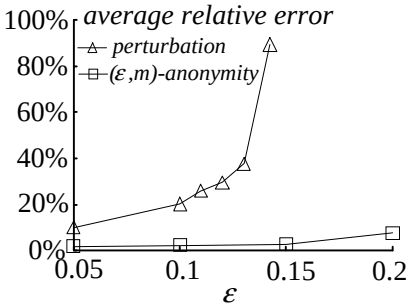
This section experimentally evaluates the effectiveness and efficiency of the proposed technique. Our purposes are twofold. First, we show that our generalization algorithm (presented in Section 5) produces (ε, m)-anonymous tables that permit accurate data analysis. Second, we verify that the algorithm entails small computation cost. Our machine runs a 3 GHz CPU, and has 1 gigabytes memory.

Our experimentation deploys a real database SAL[2] commonly used in the literature. It contains 500k tuples, each of which describes the personal information of an American. SAL includes four integer attributes *Age*, *Birthplace*, *Occupation*, and *Income*, whose domains are $[16, 93]$, $[1, 710]$, $[1, 983]$ and [1k, 100k], respectively. We treat the first three

[2]Downloadable at *http://ipums.org*.

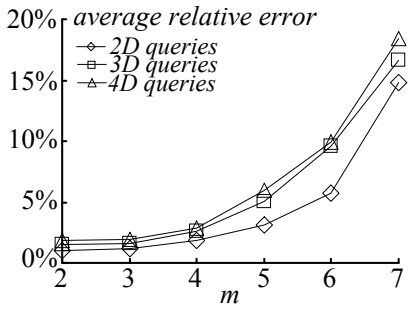


(a) Absolute ε-anonymity

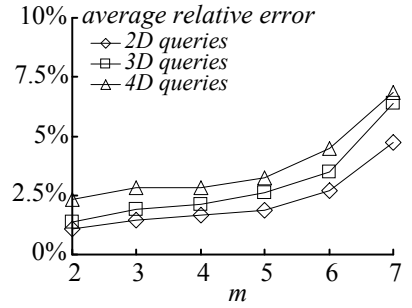


(b) Relative ε-anonymity

Figure 4: Query accuracy vs. ε ($m = 5$, $s = 0.1$)



(a) Absolute 4500-anonymity



(b) Relative 0.125-anonymity

Figure 5: Query accuracy vs. m ($s = 0.1$)

columns as QI-attributes, and *Income* as the sensitive attribute.

As an implementation detail, the function $loss(t^*)$ (for quantifying the amount of information loss; see the beginning of Section 5) employed by our algorithm is

$$\sum_{i=1}^{3} |t^*.A_i|/|A_i|$$

where A_1, A_2, A_3 denote the three QI-attributes in SAL, $|A_i|$ $(1 \leq i \leq 3)$ is the domain size of A_i, and $|t^*.A_i|$ equals the number of A_i-values covered by the generalized value $t^*.A_i$. We organize our results in two parts, which demonstrate the utility of anonymization and the execution time of our algorithm, respectively.

Utility of the Anonymized Data. As reviewed in Section 3, currently the only existing method that can provide adequate protection against proximity attacks is *perturbation*, which is selected as the competitor of our (ε, m)-*anonymity* technique. We compare the usefulness of the data anonymized by the two approaches, under the same privacy requirement. Recall that a crucial parameter of *perturbation* is its retention probability (a higher retention probability offers weaker anonymity protection but enhances data utility). We maximize the utility of *perturbation*, by always using the largest possible retention probability that ensures the degree of privacy preservation mandated by (ε, m)-anonymity. In particular, this largest retention probability is obtained in the way as described in Section 3.2.

We measure the utility of a technique by the error of answering *count queries* on the anonymized data it produces. Each query has the form:

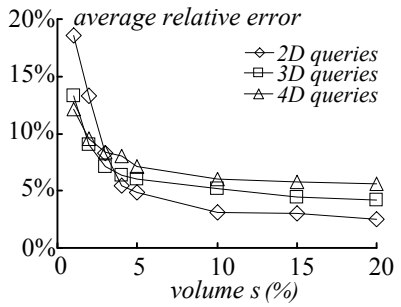
select count(*) from SAL
where $A_1 \in b_1$ *and* $A_2 \in b_2$ *and ... and* $A_w \in b_w$.

Here, w is a parameter called the *query dimensionality*. A_1, ..., A_{w-1} are $w-1$ arbitrary distinct QI-attributes in SAL, but A_w is always *Income*, b_i $(1 \leq i \leq w)$ is a random interval in the domain of A_i. The generation of b_1, ..., b_w is governed by another parameter termed *volume* s, which is a real number in $[0, 1]$, and determines the length (in the number of integers) of b_i $(1 \leq i \leq w)$ as $\lfloor |A_i| \cdot s^{1/w} \rfloor$. Apparently, the query result becomes larger given a higher s. A *workload* consists of 1k queries with the same w and s.

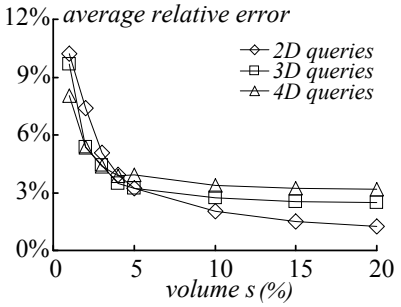
Given an (ε, m)-anonymous relation, we derive the estimated answer of a query using the approach explained in [25]. Given a perturbed relation, the estimated answer is calculated according to the solution of [6]. For both methods, the accuracy of an estimate is gauged as its relative error. Namely, let *act* and *est* be the actual and estimated results respectively; the relative error equals $|act - est|/act$.

The first set of experiments studies the influence of ε (i.e., the length of a private neighborhood) on data utility. Towards this, we set m to 5, namely, the breach risk must be bounded by 20%. We measure the average (per-query) error of *perturbation* and (ε, m)-*anonymity* in answering a workload with $s = 0.1$. Concerning absolute (relative) $(\varepsilon, 5)$-anonymity, Figure 4a (4b) plots the error as a function of ε. *Perturbation* has no result for $\varepsilon >$ 4k and 0.15 in Figures 4a and 4b, respectively. This is because, in those cases, *perturbation simply cannot provide the privacy control guaranteed by* $(\varepsilon, 5)$-*anonymity* (i.e., even the lowest retention probability 0 cannot fulfill the purpose, due to the reason elaborated in Section 3.2).

Evidently, (ε, m)-*anonymity* produces significantly more useful anonymized data than *perturbation*. Both techniques incur higher error as ε increases. This is expected, since a larger ε demands stricter privacy preservation, which reduces data utility. Nevertheless, the error of (ε, m)-*anonymity* is always below 15%. In contrast, the accu-

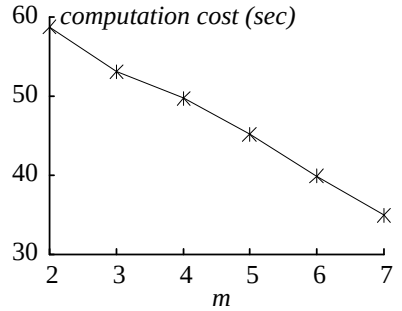


(a) Absolute 4500-anonymity

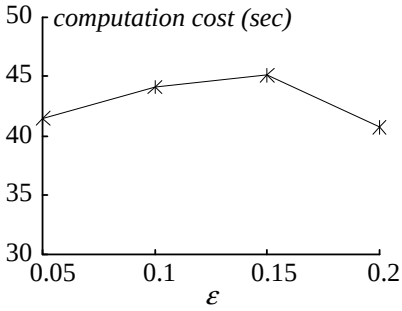


(b) Relative 0.125-anonymity

Figure 6: Query accuracy vs. s ($m = 5$)



(a) Absolute 4500-anonymity



(b) Relative ε-anonymity ($m = 5$)

Figure 7: Computation time

racy of *perturbation* deteriorates drastically as ε approaches 4k (0.15) in Figure 4a (4b). As mentioned earlier, the anonymity requirement of $\varepsilon = 4$k (0.15) is already the limit of the privacy control that can be offered by *perturbation*. Thus, when ε moves close to 4k (0.15), *perturbation* must adopt exceedingly low retention probability, rendering its anonymized data useless for research. Since *perturbation* is by far the worse method in the subsequent experiments, we do not discuss it further.

Next, still using $s = 0.1$, we examine the utility of absolute (4500, m)- and relative (0.125, m)-anonymous generalizations with different m (i.e., adjusting the upper bound of breach risk). For absolute (relative) anonymity, Figure 5a (5b) presents the average error of 2D, 3D, and 4D workloads as a function of m. The error grows with m because a larger m demands tighter anonymity control. Nevertheless, even for the greatest m, the data sanitized by our technique still enjoys fairly good utility, incurring error no more than 20% (8%) in the absolute (relative) case. To study the impact of s (which decides the magnitude of query results), we focus on absolute (4500, 5)- and relative (0.125, 5)-anonymity. Figure 6 plots the average workload error as s changes from 1% to 20%. The error decreases as s increases. This phenomenon is consistent with the existing understanding that generalization provides better support to count queries when query results are larger [25]. The query error of our solution remains small even for the lowest value of s.

Efficiency. Having verified the effectiveness of our technique, we proceed to test its efficiency. Using $\varepsilon = 4500$, Figure 7a demonstrates the cost of computing absolute (ε, m)-anonymous generalization, when m varies from 2 to 7 (the results on relative anonymity are omitted since they have similar behavior). The cost drops as m grows. This is expected, because fewer qualified generalizations exist for a greater m, allowing our algorithm to terminate earlier.

With $m = 5$, Figure 7b shows the computation cost as a function of ε, in finding relative (ε, m)-anonymous generalizations (the absolute case is analogous). Interestingly, as ε increases, the cost initially becomes higher but then decreases monotonically. This phenomenon is due to a pair of contradicting factors that push up and down the running time, respectively. First, as ε grows, the partitioning phase entails larger overhead, as more buckets output by the splitting phase need to be partitioned; this explains the initial growth of the overall cost. On the other hand, when ε escalates, there are fewer possible (ε, m)-anonymous generalizations, thus demanding less search time; this causes the eventual cost descent. In all experiments, our algorithm terminates within a minute.

7. RELATED WORK

The literature of privacy preserving publication has grown considerably in the past few years. The previous works can be loosely classified into two categories. The first one aims at developing effective anonymization principles whose satisfaction guarantees strong privacy protection. The objective of the second category is to design algorithms for obtaining generalized tables that obey an anonymization principle and yet incur small information loss. Since we have discussed in detail the known principles in Section 3, the following survey concentrates on the second category, as well as works that fit neither category nicely.

The existing generalization algorithms can be further divided into *heuristic* and *theoretical*. The main advantage of heuristic algorithms is that they are general, namely, they can be applied to many anonymization principles. Specifically, a genetic algorithm is developed in [20], and the branch-and-bound paradigm is employed on a set-enumeration tree in [7]. Top-down and bottom-up algorithms are presented in [17, 43]. *Incognito* [24] borrows ideas

from frequent item set mining, while *Mondrian* [25] takes a partitioning approach reminiscent of kd-trees. In [18], space filling curves are leveraged to facilitate generalization, and the work of [19] draws an analogy between spatial indexing and generalization. The above approaches minimize a generic metric of information loss, whereas a *workload-aware* method [23] uses a representative workload supplied by users. *Sequential publication* is addressed in [38], and *re-publication* is tacked in [42]. As shown in [39], the previous algorithms may suffer from *minimality attacks*, which can be avoided by introducing some randomization.

Heuristic algorithms work well on practical datasets, but do not have attractive asymptotical performance in the worst case. This motivates studies on theoretical algorithms. Interestingly, all the know theoretical results focus on k-anonymity. Meyerson and Williams [29] are the first to prove the NP-hardness of optimal k-anonymous generalization, and give an $O(k \log k)$-approximation algorithm. Aggarwal et al. [3] reduce the approximation ratio to $O(k)$, which is further improved to $O(\log k)$ by Park and Shim [32]. Unlike these solutions whose approximation ratios are functions of k, Du et al. [13] present a method having a ratio $O(d)$, where d is the number of attributes in the QID. Aggarwal et al. [2] develop constant approximation algorithms.

So far we have focused on generalization, while anonymized publication can also be achieved by other methodologies. Kifer and Gehrke [22] develop *marginal publication*, which releases the anonymized versions of the projections of the microdata on different subsets of attributes. Xiao and Tao [41] advocate *anatomy* that publishes the QI and SA values directly in two different tables. Aggarwal and Yu [1] design the *condensation method*, which releases only selected statistics about each QI-group. Rastogi et al. [33] employ *perturbation*, which has been explained in Section 3.

Finally, besides data publication, anonymity issues arise in many other environments. Some examples include anonymized surveying [5, 15], statistical databases [10, 14, 30], cryptographic computing [21, 34, 37], access control [4, 8, 9], and so on.

8. CONCLUSIONS

Although proximity breach is a natural privacy threat to numerical sensitive data, it has not received dedicated attention in the literature. We eliminate this threat with a new anonymization principle called (ε, m)-anonymity. We present a thorough theoretical analysis, which reveals numerous important characteristics of this principle, and leads to an efficient generalization algorithm. Extensive experiments confirm that our technique produces anonymized datasets that are highly useful in analyzing the original microdata.

This paper lays down a solid foundation for several directions towards further studies on protecting sensitive numeric data. First, while this paper concentrates on microdata that contains only a single sensitive attribute, it is interesting to investigate how the proposed solutions can be extended to support multiple attributes. Second, our discussion assumes one-time publication of a static dataset, whereas it remains open how to ensure (ε, m)-anonymity in multiple re-publications of a dynamic dataset [11]. This is a challenging problem because an adversary may utilize the intricate correlations among various published versions to increase her/his chance of breaching the privacy of an individual.

Acknowledgements

This work was supported by grants CUHK 1202/06 and 4161/07 from the research grant council of HKSAR.

REFERENCES

[1] C. C. Aggarwal and P. S. Yu. A condensation approach to privacy preserving data mining. In *Proc. of Extending Database Technology (EDBT)*, pages 183–199, 2004.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *Proc. of ACM Symposium on Principles of Database Systems (PODS)*, pages 153–162, 2006.

[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *Proc. of International Conference on Database Theory (ICDT)*, pages 246–258, 2005.

[4] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proc. of Very Large Data Bases (VLDB)*, pages 143–154, 2002.

[5] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of ACM Management of Data (SIGMOD)*, pages 439–450, 2000.

[6] R. Agrawal, R. Srikant, and D. Thomas. Privacy preserving olap. In *Proc. of ACM Management of Data (SIGMOD)*, pages 251–262, 2005.

[7] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 217–228, 2005.

[8] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. An access control model supporting periodicity constraints and temporal reasoning. *ACM Transactions on Database Systems (TODS)*, 23(3):231–285, 1998.

[9] E. Bertino and E. Ferrari. Secure and selective dissemination of xml documents. *ACM Trans. Inf. Syst. Secur.*, 5(3):290–331, 2002.

[10] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *Proc. of ACM Symposium on Principles of Database Systems (PODS)*, pages 128–138, 2005.

[11] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li. Secure anonymization for incremental datasets. In *Secure Data Management (SDM)*, pages 48–63, 2006.

[12] B.-C. Chen, R. Ramakrishnan, and K. LeFevre. Privacy skyline: Privacy with multidimensional adversarial knowledge. In *Proc. of Very Large Data Bases (VLDB)*, pages 770–781, 2007.

[13] Y. Du, T. Xia, Y. Tao, D. Zhang, and F. Zhu. On multidimensional k-anonymity with local recoding generalization. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 1422–1424, 2007.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, pages 265–284, 2006.

[15] A. V. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proc. of ACM Symposium on Principles of Database Systems (PODS)*, pages 211–222, 2003.

[16] R. Floyd and R. Rivest. Expected time bounds for selection. In *Communications of the ACM (CACM)*, volume 18, pages 165–172, 1975.

[17] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 205–216, 2005.

[18] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss. In *Proc. of Very Large Data Bases (VLDB)*, pages 758–769, 2007.

[19] T. Iwuchukwu and J. F. Naughton. K-anonymization as spatial indexing: Toward scalable and incremental anonymization. In *Proc. of Very Large Data Bases (VLDB)*, pages 746–757, 2007.

[20] V. Iyengar. Transforming data to satisfy privacy constraints. In *Proc. of ACM Knowledge Discovery and Data Mining (SIGKDD)*, pages 279–288, 2002.

[21] W. Jiang and C. Clifton. A secure distributed framework for achieving k-anonymity. *The VLDB Journal*, 15(4):316–333, 2006.

[22] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *Proc. of ACM Management of Data (SIGMOD)*, pages 217–228, 2006.

[23] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Workload-aware anonymization. In *Proc. of ACM Knowledge Discovery and Data Mining (SIGKDD)*, 2006.

[24] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proc. of ACM Management of Data (SIGMOD)*, pages 49–60, 2005.

[25] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 277–286, 2006.

[26] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.

[27] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proc. of International Conference on Data Engineering (ICDE)*, page 24, 2006.

[28] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern. Worst-case background knowledge in privacy. In *Proc. of International Conference on Data Engineering (ICDE)*, 2007.

[29] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proc. of ACM Symposium on Principles of Database Systems (PODS)*, pages 223–228, 2004.

[30] S. U. Nabar, B. Marthi, K. Kenthapadi, N. Mishra, and R. Motwani. Towards robustness in query auditing. In *Proc. of Very Large Data Bases (VLDB)*, pages 151–162, 2006.

[31] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *Proc. of ACM Management of Data (SIGMOD)*, pages 665–676, 2007.

[32] H. Park and K. Shim. Approximate algorithms for k-anonymity. In *Proc. of ACM Management of Data (SIGMOD)*, pages 67–78, 2007.

[33] V. Rastogi, S. Hong, and D. Suciu. The boundary between privacy and utility in data publishing. In *Proc. of Very Large Data Bases (VLDB)*, pages 531–542, 2007.

[34] J. Rothe. Some facets of complexity theory and cryptography: A five-lecture tutorial. *ACM Computing Surveys*, 34(4):504–549, 2002.

[35] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6):1010–1027, 2001.

[36] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness, and Knowlege-Based Systems*, 10(5):557–570, 2002.

[37] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proc. of ACM Knowledge Discovery and Data Mining (SIGKDD)*, pages 206–215, 2003.

[38] K. Wang and B. C. M. Fung. Anonymizing sequential releases. In *Proc. of ACM Knowledge Discovery and Data Mining (SIGKDD)*, pages 414–423, 2006.

[39] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei. Minimality attack in privacy preserving data publishing. In *Proc. of Very Large Data Bases (VLDB)*, pages 543–554, 2007.

[40] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang. (alpha, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proc. of ACM Knowledge Discovery and Data Mining (SIGKDD)*, pages 754–759, 2006.

[41] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *Proc. of Very Large Data Bases (VLDB)*, pages 139–150, 2006.

[42] X. Xiao and Y. Tao. m-invariance: towards privacy preserving re-publication of dynamic datasets. In *Proc. of ACM Management of Data (SIGMOD)*, pages 689–700, 2007.

[43] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. Utility-based anonymization using local recoding. In *Proc. of ACM Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–790, 2006.

[44] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 116–125, 2007.