Personalized Privacy Preservation

Xiaokui Xiao Yufei Tao

Department of Computer Science
City University of Hong Kong
Tat Chee Avenue, Hong Kong
{xkxiao, taoyf}@cs.cityu.edu.hk

ABSTRACT

We study generalization for preserving privacy in publication of sensitive data. The existing methods focus on a universal approach that exerts the same amount of preservation for all persons, without catering for their concrete needs. The consequence is that we may be offering insufficient protection to a subset of people, while applying excessive privacy control to another subset.

Motivated by this, we present a new generalization framework based on the concept of *personalized anonymity*. Our technique performs the minimum generalization for satisfying everybody's requirements, and thus, retains the largest amount of information from the microdata. We carry out a careful theoretical study that leads to valuable insight into the behavior of alternative solutions. In particular, our analysis mathematically reveals the circumstances where the previous work fails to protect privacy, and establishes the superiority of the proposed solutions. The theoretical findings are verified with extensive experiments.

1. INTRODUCTION

It is often necessary to publish personal information for research purposes. For example, a hospital may release patients' diagnosis records so that researchers can study the characteristics of various diseases. The raw data, also called *microdata*, contains the identities (e.g. names) of individuals, which are not released to protect their privacy. However, there may exist other attributes that can be used, in combination with an external database, to recover the personal identities.

For example, assume that the hospital publishes the table in Figure 1a, which does not explicitly indicate the names of patients. However, if an adversary has access to the voter registration list in Figure 1b, s/he can easily discover the identities of all patients by joining the two tables on {*Age*, *Sex*, *Zipcode*}. These three attributes are, therefore, the *quasi-identifier* (QI) attributes.

Generalization [5, 7, 8, 9, 11, 13, 15, 17] is a common approach to avoid the above problem, by transforming the QI values into less specific forms so that they no longer uniquely represent individuals. In particular, a table is k*-anonymous* [13, 15] if the QI values of each tuple are identical to those of at least $k-1$ other tuples. Figure 1c shows an example of 2-anonymous generalization for Figure 1a. Even with the voter registration list, an adversary can only infer that Andy may be the person involved in the first 2 tuples of Figure 1c, or equivalently, the real disease of Andy is discovered only with probability 50%. In general, k-anonymity guarantees that an individual can be associated with her/his real tuple with a probability at most $1/k$.

1.1 Motivation

k-anonymity has several drawbacks. First, *a k-anonymous table may allow an adversary to derive the sensitive information of an individual with 100% confidence.* Assume that an adversary attempts to infer the disease of Joe, knowing his age 12, sex, and zipcode 22000. From the published table in Figure 1c, s/he knows that Joe may correspond to tuple 5 or 6 (the QI values of the other tuples do not cover those of Joe). The diseases of both tuples are *pneumonia*; hence, the adversary can declare (with 100% confidence) that Joe must have contracted *pneumonia*. The phenomenon is caused by the fact that, k-anonymity only prevents association between individuals and tuples, instead of association between individuals and *sensitive values*. Unfortunately, it is the second type of association that leads to privacy breach.

Second, *a k-anonymous table may lose considerable information from the microdata.* Consider a researcher who wants to obtain, from the table of Figure 1c, an estimate for the number of female patients above the age of 30. It suffices to examine tuples 7-10, because they are the only tuples that may qualify the query condition. Given only the fact that the original ages of the 4 tuples are in $[21, 60]$, the researcher derives the estimate by assuming a uniform age distribution. This leads to an estimate of $4 \times \frac{60-30}{60-20} = 3$, which significantly deviates from the actual result 1 (see Figure 1a). The serious error arises because Mary has a much larger age than the other patients; hence, combining her age with another age incurs substantial information loss. Observe that the same problem also exists on attribute *Zipcode* with respect to tuple 7. Specifically, Linda's exceedingly-large zipcode decides the loose zipcode-range $[30000, 60000]$ for tuples 7-10[1].

Third, *k-anonymity does not take into account personal anonymity requirements.* As mentioned earlier, from Figures 1b and 1c, an adversary learns that Andy must have suffered from either *gastric-ulcer* or *dyspepsia*, which is acceptable according to

SIGMOD 2006, June 27–29, 2006, Chicago, Illinois, USA.

[1]Although the table of Figure 1c demands only 2-anonymity, it is reasonable to make the QI values of tuples 7-10 identical. This is because, by "single-dimension encoding" generalization [9], no two intervals on the same attribute should intersect. For instance, if tuples 7, 8 (in Figure 1a) and tuples 9, 10 are generalized to two separate groups, then the zipcode-ranges of the two groups will intersect. Similarly, combining tuples 7, 9 and tuples 8, 10 in two groups respectively causes intersection on *Age*.

row #	Age	Sex	Zipcode	Disease	guarding node
1 (Andy)	5	M	12000	gastric ulcer	*stomach disease*
2 (Bill)	9	M	14000	dyspepsia	*dyspepsia*
3 (Ken)	6	M	18000	pneumonia	*respiratory infection*
4 (Nash)	8	M	19000	bronchitis	*bronchitis*
5 (Joe)	12	M	22000	pneumonia	*pneumonia*
6 (Sam)	19	M	24000	pneumonia	*pneumonia*
7 (Linda)	21	F	58000	flu	*Ø*
8 (Jame)	26	F	36000	gastritis	*gastritis*
9 (Sarah)	28	F	37000	pneumonia	*respiratory infection*
10 (Mary)	56	F	33000	flu	*flu*

(a) Microdata

Name	Age	Sex	Zipcode
Andy	5	M	12000
Bill	9	M	14000
Ken	6	M	18000
Nash	8	M	19000
Mike	*7*	*M*	*17000*
Joe	12	M	22000
Sam	19	M	24000
Linda	21	F	58000
Jane	26	F	36000
Sarah	28	F	37000
Mary	56	F	33000

(b) Voter registration list

row #	Age	Sex	Zipcode	Disease
1	[1, 10]	M	[10001, 15000]	gastric ulcer
2	[1, 10]	M	[10001, 15000]	dyspepsia
3	[1, 10]	M	[15001, 20000]	pneumonia
4	[1, 10]	M	[15001, 20000]	bronchitis
5	[11, 20]	M	[20001, 25000]	pneumonia
6	[11, 20]	M	[20001, 25000]	pneumonia
7	[21, 60]	F	[30000, 60000]	flu
8	[21, 60]	F	[30000, 60000]	gastritis
9	[21, 60]	F	[30000, 60000]	pneumonia
10	[21, 60]	F	[30000, 60000]	flu

(c) A 2-anonymous table

Figure 1: Microdata, external source, and quasi-identifier generalization

2-anonymity. However, Andy may not want anyone to think (with high confidence) "Andy must have some stomach problem"; this cannot be guaranteed in Figure 1c, since both *gastric-ulcer* and *dyspepsia* are stomach diseases. On the other hand, it is possible that Linda regards *flu* as a common disease, and agrees to release her true diagnosis result (to enhance the effectiveness of research). In this case, it is not necessary to apply any generalization on tuple 7. Such preference variations are not captured by k-anonymity.

1.2 Contributions

In this paper, we develop a novel privacy preserving technique that overcomes the above problems. The core of our solutions is the concept of *personalized anonymity*, i.e., a person can specify the degree of privacy protection for her/his sensitive values. To illustrate the concept, consider Figure 2, which demonstrates a simple taxonomy on attribute *Disease*. The taxonomy is accessible by the public, and organizes all diseases as leaves of a tree. An intermediate node carries a name summarizing the diseases in its subtree. Some part of the tree is omitted since it is not relevant to our discussion.

A personal preference can be easily solicited from an individual when s/he is supplying her/his data. In our approach, a preference is formulated through a node in the taxonomy. As an example, for tuple 1 in Figure 1a, Andy may specify node *stomach-disease* (the "guarding node" for his privacy, which will be formalized in the next section). Thus, nobody should be able to infer, with significant confidence, that he suffered from any disease (i.e., *gastric-ulcer*, *dyspepsia*, or *gastritis*) in the subtree of the node. In other words, in Andy's opinion, allowing the public to associate him with *dyspepsia* or *gastritis* is as serious as revealing his true disease.

On the other hand, for tuple 7 in Figure 1a, Linda may specify $\emptyset$, which is an implicit node underneath all the leaves of the taxonomy. The empty-set preference implies that she is willing to release her actual diagnosis result *flu*; therefore, tuple 7 can be published directly. In general, *flu* may not be "sensitive" for many people, such that it is often not necessary to apply any privacy protection to this value.

In fact, personalization is an inherent notion of privacy preservation whose objective is to protect the interests of individuals at the first place. Somewhat surprisingly, so far the literature has focused on a universal approach that exerts the same amount of privacy preserving for all persons, without catering for their concrete needs. The consequence is that we may be offering insufficient protection to a subset of people (such as Andy in the above example), while applying excessive privacy control to another subset (including, for instance, Linda). Our method is more flexible, since it decides the minimum amount of necessary generalization for satisfying everybody's needs, and hence, retains the maximum amount of information from the microdata.

We present a careful study for the problem of personalized

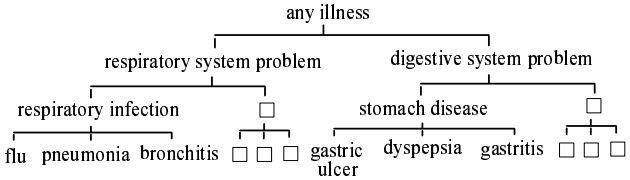


Figure 2: The taxonomy of attribute *Disease*

anonymity. First, we formalize the concepts that underlie a new framework of computing privacy-conscious information taking into account individual preferences. As opposed to k-anonymity, our approach applies *direct protection against the association between individuals and their sensitive values*.

As a second step, we analyze the theory behind our methodology, and derive formulae for quantifying privacy-breach likelihood. These equations mathematically reveal the scenarios where k-anonymity can/cannot ensure safe data publication. In particular, we prove that, unlike our approach, *k-anonymity (even its improved version "l-diversity" [11]) cannot guarantee privacy protection if an individual may correspond to multiple tuples in the microdata*. This is a serious defect due to the large amount of such data in practice that requires privacy control. For example, the table of Figure 1a may contain numerous records for a person if s/he has been sick for several times.

Finally, we develop an algorithm for finding a generalized table that preserves a large amount of information in the microdata without violating any privacy constraint. Utilizing several interesting problem characteristics, the algorithm optimizes the degrees of generalization on QI- and sensitive attributes, respectively. Extensive experiments verify that the output tables of our algorithm permit highly accurate data analysis.

The rest of the paper is organized as follows. Section 2 formalizes the general methodology of personalized anonymity. Section 3 provides its theoretical foundation, and reveals important insight into the behavior of alternative approaches. Section 4 explains an algorithm for deriving a generalized table. Section 5 experimentally evaluates the effectiveness of our solutions. Section 6 surveys the previous work related to ours, and Section 7 concludes the paper with directions for future work.

2. PERSONALIZED ANONYMITY

Let T be a relation storing private information about a set of individuals. The attributes in T are classified in 4 categories: (i) an *identifier attribute* $\mathcal{A}^i$ which uniquely identifies a person, and must be removed when T is released to the public, (ii) a *sensitive attribute* $\mathcal{A}^s$ (e.g., *Disease* in Figure 1a), whose values may be confidential for an individual (subject to her/his preferences), (iii) d *quasi-identifier* (QI) *attributes* $\mathcal{A}_1^{qi}$, ..., $\mathcal{A}_d^{qi}$, whose values can be published, but may reveal a personal identify with the aid of ex-

ternal information (*Age*, *Sex*, *Zipcode* in Figure 1a), and (iv) other attributes that are not relevant to our discussion.

We require that $\mathcal{A}^s$ should be categorical, whereas the other attributes can be either numerical or categorical. All the attributes have finite domains. Following the previous work [5, 7, 8, 9, 11, 13, 15, 17], we assume that each categorical attribute A is accompanied by a *taxonomy* (as in Figure 2 for *Disease*), which indicates the publicly-known hierarchy among the possible values of A.

Our objective is to compute a generalized table T^* such that (i) it contains all the attributes of T except $\mathcal{A}^i$, (ii) it has a generalized tuple for *every* tuple in T, (iii) it preserves as much information of T as possible, and (iv) its publication does not cause any privacy breach, as formulated in the next section.

2.1 Personal Privacy Requirements

We start by defining a subtree in the taxonomy of $\mathcal{A}^s$.

DEFINITION 1 ($\mathcal{A}^s$ SUBTREE). *For any node x in the taxonomy of $\mathcal{A}^s$, we represent its* **subtree** *as* SUBTR(x), *which includes x itself, and the part of the taxonomy under it.*

A tuple $t \in T$ defines an *association* between an individual o (identified by $t.\mathcal{A}^i$) and a sensitive value $v = t.\mathcal{A}^s$. We denote the association as $\{o, v\}$. To formulate her/his privacy preference, o specifies a *guarding node* as follows:

DEFINITION 2 (GUARDING NODE). *For a tuple $t \in T$, its* **guarding node $t.\mathcal{GN}$** *is a node on the path from the root to $t.\mathcal{A}^s$ in the taxonomy of $\mathcal{A}^s$.*

Through $t.\mathcal{GN}$, o indicates that s/he does not want the public to associate her/him with any leaf $\mathcal{A}^s$ value in SUBTR$(t.\mathcal{GN})$. Specifically, assume that SUBTR$(t.\mathcal{GN})$ contains x leaf values v_1, v_2, ..., v_x. The privacy requirement of $t.\mathcal{GN}$ is *breached* if an adversary thinks that any of the associations $\{o, v_1\}$, ..., $\{o, v_x\}$ exists in T.

DEFINITION 3 (BREACH PROBABILITY). *For a tuple $t \in T$, its* **breach probability $\mathbf{P}_{breach}(t)$** *equals the probability that an adversary can infer from T^* that any of the associations $\{o, v_1\}$, ..., $\{o, v_x\}$ exists in T, where v_1, ..., v_x are the leaf values in* SUBTR$(t.\mathcal{GN})$.

The published table T^* should guarantee that, for all $t \in T$, $\mathbf{P}_{breach}(t)$ is at most p_{breach}, which is a system parameter specifying the amount of confidentiality control.

Figure 1a demonstrates the guarding nodes selected by the individuals involved in the microdata. For example, let t be tuple 3 ($t.\mathcal{A}^i$ = *Ken* and $t.\mathcal{A}^s$ = *pneumonia*). The guarding node *respiratory-infection* of t indicates that nobody can infer, with high confidence, that Ken suffered from a disease under *respiratory-infection* in the taxonomy of Figure 2. $\mathbf{P}_{breach}(t)$ is the probability that an adversary can infer that *any* of the following 3 associations exists in T: {Ken, *flu*}, {Ken, *pneumonia*}, {Ken, *bronchitis*}.

On the other hand, Ken does not care if somebody conjectures, with any probability, that he contracted *gastric-ulcer* (not in SUBTR$(t.\mathcal{GN})$), since it is very different from his true diagnosis result. In general, the higher $t.\mathcal{GN}$ is in the taxonomy, the stronger privacy must be guaranteed.

Guarding nodes depend entirely on personal preferences, and are not determined by the sensitive values. For instance, Joe and Sam (who, as with Ken, contracted *pneumonia*) set their guarding nodes simply to *pneumonia* (tuples 5, 6 in Figure 1a), implying that they do not mind being associated with *flu* or *bronchitis*. Specially, if a patient believes that disclosing $t.\mathcal{A}^s$ to the public does not violate her/his privacy, s/he may simply set $t.\mathcal{GN}$ to $\emptyset$.

2.2 Generalization

We first clarify two fundamental concepts.

DEFINITION 4. (PARTITION / GENERAL DOMAIN) *If attribute $\mathcal{A}$ is numeric, a* **partition** *is a continuous interval in the domain of $\mathcal{A}$. Otherwise, a* **partition** *consists of all the leaves in the subtree of a node in the taxonomy of $\mathcal{A}$. In any case, a* **general domain** *of $\mathcal{A}$ is a set of disjoint* partitions *whose union forms the original domain of $\mathcal{A}$.*

By a simple transformation, we can use the interval representation for the general domains of both numeric and categorical attributes. Notice that, when $\mathcal{A}$ is categorical, a general domain is determined by a set of nodes in the taxonomy of $\mathcal{A}$, whose subtrees do not overlap, but cover all the leaves. (For instance, in Figure 2, nodes *respiratory-system-problem* and *digestion-system-problem* decide a general domain of *Disease*.) Clearly, $\mathcal{A}$ can be converted to a numeric attribute by imposing a 1D ordering on the leaves of its taxonomy: the left-most leaf is mapped to value 1, its neighbor to 2, and so on. Thus, a partition of $\mathcal{A}$ can be denoted as an interval. For example, the partition corresponding to *respiratory-system-problem* in Figure 2 is an interval of $[1, 6]$.

DEFINITION 5 (GENERALIZATION). *A general domain of an attribute $\mathcal{A}$ uniquely decides a* **generalization function**. *Given a value v in the original domain of $\mathcal{A}$, the function returns the only partition in the general domain that contains v. The partition is the* **generalized value** *of v.*

Clearly, $\mathcal{A}$ can have many generalization functions, since its values can be partitioned into numerous general domains.

For each tuple $t \in T$, we use t^* to represent its generalized tuple in T^*. The generalization is performed in two steps. The first step, the *QI-generalization*, is identical to conventional generalization in [5, 7, 8, 17]. Specifically, we choose a generalization function for every QI attribute $\mathcal{A}_i^{qi}$ ($1 \leq i \leq d$), and obtain the generalized value $t^*.\mathcal{A}_i^{qi}$ for all tuples $t \in T$ (t^* retains the sensitive value of t at this step). Then, the generalized tuples are divided into *QI-groups*, defined as follows.

DEFINITION 6 (QI-GROUP). *After* QI-generalization, *a* **QI-group** *consists of the tuples with identical values on all the QI attributes.* **The i-th QI-value** *($1 \leq i \leq d$) of the QI-group equals $t.A_i^{qi}$, where t is an arbitrary tuple in the QI-group.*

In the second step, *SA-generalization* (SA stands for "sensitive attribute"), we consider each QI-group in turn, and select a *tailored* generalization function on $\mathcal{A}^s$. Note that, unlike the previous step where all tuples are processed with identical generalization functions, SA-generalization uses a *different* function for each group. This strategy achieves less information loss, by allowing each group to decide the amount of necessary generalization.

Figure 3 shows a possible result of our entire generalization scheme for Figure 1a. The table contains 5 QI-groups: the first one includes tuples 1-4, the second involves tuples 5-6, the third only tuple 7, the fourth tuples 8-9, and the fifth group consists of the last tuple. Note that the sensitive value *flu* of tuple 7 is retained directly, while the same disease of tuple 10 is generalized to *respiratory-infection*. This is legal because, as mentioned earlier, SA-generalization may choose a different generalization function for each QI-group.

None of the existing methods permits SA-generalization. In fact, as demonstrated in Section 5, SA-generalization may produce a table that allows more accurate analysis about the correlation between the sensitive attribute $\mathcal{A}^s$ and other attributes. The reason

row #	Age	Sex	Zipcode	Disease
1 (Andy)	[1, 10]	M	[10001, 20000]	gastric ulcer
2 (Bill)	[1, 10]	M	[10001, 20000]	dyspepsia
3 (Ken)	[1, 10]	M	[10001, 20000]	respiratory infection
4 (Nash)	[1, 10]	M	[10001, 20000]	respiratory infection
5 (Joe)	[11, 20]	M	[20001, 25000]	respiratory infection
6 (Sam)	[11, 20]	M	[20001, 25000]	respiratory infection
7 (Linda)	21	F	58000	flu
8 (Jane)	[26, 30]	F	[35001, 40000]	gastritis
9 (Sarah)	[26, 30]	F	[35001, 40000]	pneumonia
10 (Mary)	56	F	33000	respiratory infection

Figure 3: A possible result of our generalization scheme

is that, although SA-generalization results in less precise values on $\mathcal{A}^s$, it retains more information on the QI attributes.

In Figure 1a, for example, considerable *Age* precision will be lost by generalizing the QI values of Mary (tuple 10), as discussed in Section 1.1. An alternative approach is to generalize her disease *flu* to *respiratory-infection*, leaving the other QI values intact. As shown in Figure 3, this leads to an age-interval $[26, 30]$ for tuples 8-9 that is much tighter than their age representation $[21, 60]$ in Figure 1c. If we publish the table in Figure 3, an adversary can find out that *flu* is the real disease of Mary only with probability 1/3 (*flu* is the guarding node set by Mary), as explained in Section 2.3. Intuitively, this is because 3 different diseases exist in the subtree of *respiratory-infection* (the sensitive value of tuple 10 in Figure 3).

2.3 Combinatorial Process of Privacy Attack

Consider an adversary who attempts to infer the sensitive data of an individual o from T^*. In the worst case, s/he has all the QI values $o.\mathcal{A}_1^{qi}$, ..., $o.\mathcal{A}_d^{qi}$ of o. Therefore, s/he inspects only those tuples $t^* \in T^*$ whose QI value $t^*.\mathcal{A}_i^{qi}$ covers $o.\mathcal{A}_i^{qi}$, for all $i \in [1, d]$.

These tuples must form a QI-group. That is, if t^* and t'^* are two such tuples, then $t^*.\mathcal{A}_i^{qi} = t'^*.\mathcal{A}_i^{qi}$ for all $i \in [1, d]$. Actually, if, for instance, $t^*.\mathcal{A}_1^{qi} \neq t'^*.\mathcal{A}_1^{qi}$, the two values are different partitions in the general domain of $\mathcal{A}_1^{qi}$ that both contain $o.\mathcal{A}_1^{qi}$, violating the requirement that all partitions are disjoint.

DEFINITION 7. (ESSENTIAL QI-GROUP / $\mathcal{S}_{real}$). *Given an individual o, the* **essential QI-group $\mathcal{EG}(o)$** *is the only QI-group in T^* whose i-th QI-value covers $o.\mathcal{A}_i^{qi}$, for all $i \in [1, d]$. We use $\mathcal{S}_{real}(o)$ to refer to the set of individuals, who have tuples in T generalized to $\mathcal{EG}(o)$.*

Note that $\mathcal{S}_{real}(o)$ is unknown to an adversary. To derive $\mathcal{S}_{real}(o)$, the adversary must resort to an external dataset, and retrieve a set $\mathcal{S}_{ext}(o)$ of persons that may be concerned in $\mathcal{EG}(o)$. $\mathcal{S}_{ext}(o)$ is defined as follows.

DEFINITION 8 (EXTERNAL INDIVIDUAL SET $\mathcal{S}_{ext}$). *Given an essential QI-group $\mathcal{EG}(o)$, and an external database DB_{ext}, $\mathcal{S}_{ext}(o)$ consists of the people $o' \in DB_{ext}$, such that $o'.\mathcal{A}_i^{qi}$ ($1 \leq i \leq d$) is covered by the i-th QI-value of $\mathcal{EG}(o)$.*

To illustrate the above concepts, assume that an adversary tries to infer the disease of Ken from Figure 3, having his age 6, sex, and zipcode 18000. The essential QI-group $\mathcal{EG}$(Ken) consists of tuples 1-4, i.e., $\mathcal{S}_{real}$(Ken) equals {Andy, Bill, Ken, Nash}. Attempting to derive $\mathcal{S}_{real}$(Ken), the adversary consults the external database in Figure 1b, and obtains $\mathcal{S}_{ext}$(Ken) = {Andy, Bill, Ken, Nash, Mike}.

In general

$$\mathcal{S}_{real}(o) \subseteq \mathcal{S}_{ext}(o) \quad (1)$$

This is a reasonable condition underlying all the previous work. For instance, if Ken does not appear in the voter registration list, his privacy is trivially preserved. In fact, under the circumstances where an arbitrary number of individuals in T may be missing in the external source, the adversary can infer little information, because all tuples of the essential QI-group may actually correspond to the missing individuals.

Next, the adversary adopts a combinatorial approach to infer the A^s value of individual o. We elaborate the approach by distinguishing two cases in Sections 2.3.1 and 2.3.2, respectively. The subsequent discussion uses m, n to represent the sizes of $\mathcal{EG}(o)$ and $\mathcal{S}_{ext}(o)$, respectively. Also, we denote the tuples in $\mathcal{EG}(o)$ as t_1^*, ..., t_m^*, whose original versions in the microdata are t_1, ..., t_m, respectively.

2.3.1 Primary Case

We first consider the case where $T.\mathcal{A}^i$ is the primary key of T, i.e., each individual has at most one tuple in T. This is the only scenario addressed in the previous work [5, 7, 8, 9, 11, 13, 15, 17].

DEFINITION 9. (PRIMARY POSSIBLE RECONSTRUCTION). *In the Primary Case, given an individual o, a* **possible reconstruction** *of the essential QI-group $\mathcal{EG}(o)$ includes*

- *m distinct persons o_1, ..., o_m, who constitute a subset of $\mathcal{S}_{ext}(o)$, i.e., o_j ($1 \leq j \leq m$) is taken as the owner of t_j;*
- *m leaf sensitive values v_1, ..., v_m, such that v_j ($1 \leq j \leq m$) is in* SUBTR($t_j^*.\mathcal{A}^s$)*, i.e., v_j is taken as the real sensitive value of t_j.*

EXAMPLE 1. We explain the definition by continuing our example, where the adversary has derived $\mathcal{S}_{ext}$(Ken) = {Andy, Bill, Ken, Nash, Mike}. As mentioned earlier, m = 4, n = 5, and t_1^*, ..., t_4^* are tuples 1-4 in Figure 3, respectively.

To obtain a possible reconstruction, the adversary first assigns o_1, ..., o_4 to 4 different persons in $\mathcal{S}_{ext}$(Ken). As a possible assignment, o_1 = Mike, o_2 = Nash, o_3 = Andy, and o_4 = Ken. Then, the adversary sets v_1 to *gastric-ulcer*, which is the only potential value of v_1, because $t_1^*.\mathcal{A}^s$ = *gastric-ulcer* is a leaf node in the *Disease*-taxonomy. For the same reason, v_2 must be *dyspepsia*. On the other hand, v_3 (v_4) can be any of the 3 leaf diseases under $t_3^*.\mathcal{A}^s$ ($t_4^*.\mathcal{A}^s$) = *respiratory-infection*. The possible reconstruction is completed by assuming, for instance, v_3 = *flu* and v_4 = *bronchitis*.

According to the reconstruction, the adversary thinks that Mike, Nash, Andy, Ken contracted *gastric-ulcer*, *dyspepsia*, *flu*, and *bronchitis*, respectively. Note that a reconstruction most likely is not equivalent to the microdata (where Mike does not even exist); instead, it is only a conjecture by the adversary. Nevertheless, the previous reconstruction violates the privacy requirement enforced by the guarding node of tuple 3 in Figure 1a (i.e., Ken does not want people to think that he had any respiratory infection). Interestingly, the breach happens when Ken is associated with tuple 4, *instead of his original tuple 3* in the microdata.

It is important to understand the probabilistic nature of possible reconstructions. In fact, o_1, ..., o_4 can be decided in Permu$(5, 4)$ = 120 ways[2]. For each decision, by the reasoning explained earlier, v_1 and v_2 are fixed, but 3^2 = 9 choices exist for setting v_3 and v_4. Hence, there exist totally 120 $\times$ 9 = 1080 possible reconstructions.

432 reconstructions breach the privacy requirement of tuple 3 in Figure 1. Specifically, a reconstruction is breaching if and only if either o_3 or o_4 equals Ken. If o_3 = Ken, then there are Permu$(4, 3)$

[2]Permu(x, y) equals the number of permutations by taking y objects out of a set of x objects.

= 24 choices to formulate o_1, o_2, o_4, and 9 possibilities to determine v_1, ..., v_4, leading to 24 × 9 = 216 reconstructions. Symmetrically, if o_4 = Ken, there exist another 216 breaching reconstructions.

Without further information, the adversary assumes that each reconstruction happens with identical likelihood. Hence, the breach probability of tuple 3 in the microdata equals 432/1080 = 2/5. □

2.3.2 Non-primary Case

We proceed to analyze the case where $T.\mathcal{A}^i$ is not the primary key of T, namely, each individual can appear an arbitrary number of times in T. No previous work has addressed this scenario before.

DEFINITION 10. (NONPRIMARY POSSIBLE RECONSTRUCTION). *In the Non-primary Case, given an individual o, a* **possible reconstruction** *of the essential QI-group $\mathcal{EG}(o)$ includes*

- *a multi-set of individuals $\{o_1, ..., o_m\}$ (perhaps with duplicates), where the distinct elements constitute a subset of $\mathcal{S}_{ext}(o)$;*
- *m leaf sensitive values $v_1, ..., v_m$, such that v_j ($1 \leq j \leq m$) is in* SUBTR($t_j^*.\mathcal{A}^s$).

EXAMPLE 2. Let us revisit the situation where the adversary has obtained $\mathcal{S}_{ext}$(Ken) = {Andy, Bill, Ken, Nash, Mike}. The values of m, n, t_1^*, ..., and t_4^* are the same as in Example 1.

In a possible reconstruction, the adversary may set all of o_1, ..., o_4 to Ken (which is not allowed in the Primary Case). The way that v_1, ..., v_4 are decided is identical to that in Example 1; let us again assume v_1 = *gastric-ulcer*, v_2 = *dyspepsia*, v_3 = *flu*, and v_4 = *bronchitis*. By this reconstruction, the adversary thinks that Ken contracted all the 4 diseases. Evidently, the conjecture does not correctly reflect the microdata, but it causes a privacy breach for tuple 3 in Figure 1a.

Since each of o_1, ..., o_4 can independently be any of {Andy, Bill, Ken, Nash, Mike}, 5^4 = 625 choices exist for deciding o_1, ..., o_4. Given each decision, due to the reasons presented in Example 1, there are 9 ways to formulate v_1, ..., v_4. Therefore, the total number of possible reconstructions equals 625 × 9 = 5625.

A reconstruction breaches the privacy constraint of tuple 3 in the microdata, if and only if Ken is assigned to o_3 or o_4. If o_3 = Ken, o_1, o_2, o_4 may be any person in $\mathcal{S}_{ext}$(Ken), and hence, can be assigned in 5^3 = 125 manners. Regardless of the assignment, v_1, ..., v_4 may be set in 9 ways, resulting in 125 × 9 = 1125 different reconstructions. Similarly, another 1125 exist if o_4 = Ken, but some of them (where o_3 = o_4 = Ken) have been counted twice. Specifically, if o_3 = o_4 = Ken, there are 25 possibilities for determining o_1 and o_2, whereas, for each possibility, 9 choices exist for deciding v_1, ..., v_4. Hence, the number of double-counted reconstructions equals 25 × 9 = 225.

Therefore, totally $1125 + 1125 - 225 = 2025$ reconstructions breach the privacy of tuple 3 in Figure 1a. Thus, the breach probability of the tuple equals 2025/5625 = 9/25. □

Deriving a breach probability through the above procedures is quite cumbersome. In the next section, we present closed formulae that return the probability directly. Then, it will become simple to verify that publishing the table of Figure 3 allows no tuple in Figure 1a to be breached with a probability more than 50%.

3. THEORETICAL FOUNDATION

In this section, we solve the probability $\mathbf{P}_{breach}(t_{tar})$ formulated in Definition 3, where t_{tar} is an arbitrary tuple in T (the subscript means "target"). Obviously, if the guarding node $t_{tar}.\mathcal{GN}$ of t_{tar} is $\emptyset$, $\mathbf{P}_{breach}(t_{tar}) = 0$, i.e., no privacy control is required. Next, we focus on $t_{tar}.\mathcal{GN} \neq \emptyset$.

Section 3.1 first clarifies the notations and their properties, which will be used in our derivation. Then, Section 3.2 solves $\mathbf{P}_{breach}(t_{tar})$ into closed formulae, based on which Section 3.3 points out the defects of k-anonymity.

3.1 Notations and Basic Properties

Following the notations in Section 2.3, we use o_{tar} to denote the person identified by $t_{tar}.\mathcal{A}^i$, and t_{tar}^* for the generalized tuple of t_{tar}. Furthermore, let m be the size of the corresponding essential QI-group $\mathcal{EG}(o_{tar})$ (Definition 7), whose tuples are represented as t_1^*, ..., t_m^* (one of which is t_{tar}^*), respectively. $\mathcal{S}_{real}(o_{tar})$ refers to the set of individuals whose records (in the microdata T) are generalized to $\mathcal{EG}(o_{tar})$. Finally, we deploy n for the cardinality of $\mathcal{S}_{ext}(o_{tar})$ (Definition 8).

As a direct corollary of Formula 1, we have:

$$n \geq |\mathcal{S}_{real}(o_{tar})| \tag{2}$$

In the Primary Case, $|\mathcal{S}_{real}(o_{tar})|$ always equals m, since every tuple in $\mathcal{EG}(o_{tar})$ is owned by a distinct person. In the Non-primary case, however, $|\mathcal{S}_{real}(o_{tar})|$ may be any value in $[1, m]$. Furthermore, regardless of the size of $\mathcal{EG}(o_{tar})$, $|\mathcal{S}_{real}(o_{tar})|$ can take the minimum value 1, which happens if all the tuples in $\mathcal{EG}(o_{tar})$ belong to the same person.

We introduce b as the number of tuples t_j^* ($1 \leq j \leq m$) in $\mathcal{EG}(o_{tar})$, such that SUBTR($t_j^*.\mathcal{A}^s$) overlaps SUBTR($t_{tar}.\mathcal{GN}$). For example, assume that t_{tar} is tuple 1 of Figure 1a, i.e., $t_{tar}.\mathcal{GN}$ = *stomach-disease*. Thus, in Figure 3, $\mathcal{EG}(o_{tar})$ involves tuples 1-4, and m = 4. Since SUBTR($t_{tar}.\mathcal{GN}$) overlaps the subtrees of the $\mathcal{A}^s$ values of tuples 1 and 2 in $\mathcal{EG}(o_{tar})$, we have b = 2.

We define two functions $\mathcal{F}_{subsize}$ and $\mathcal{F}_{percent}$ related to the tuples $t^* \in T^*$. Specifically, $\mathcal{F}_{subsize}(t^*)$ equals the number of leaf values in SUBTR($t^*.\mathcal{A}^s$) (e.g., $\mathcal{F}_{subsize}(t^*)$ = 3 if $t^*.\mathcal{A}^s$ = *respiratory-infection*). On the other hand:

- $\mathcal{F}_{percent}(t^*, t_{tar})$ equals the *percentage* of the leaf values in SUBTR($t^*.\mathcal{A}^s$) that are also in SUBTR($t_{tar}.\mathcal{GN}$).

Thus, it follows that:

- $\mathcal{F}_{percent}(t^*, t_{tar}) = 1$, if $t^*.\mathcal{A}^s$ is in SUBTR($t_{tar}.\mathcal{GN}$);
- $\mathcal{F}_{percent}(t^*, t_{tar}) = 0$, if SUBTR($t^*.\mathcal{A}^s$) is disjoint with SUBTR($t_{tar}.\mathcal{GN}$).

We illustrate $\mathcal{F}_{percent}$ assuming $t_{tar}.\mathcal{GN}$ = *respiratory-infection*. If $t^*.\mathcal{A}^s$ = *respiratory-system-problem*, then $\mathcal{F}_{percent}(t^*, t_{tar})$ = 50%, because $t^*.\mathcal{A}^s$ has 6 leaf diseases, and half of them lie in SUBTR($t_{tar}.\mathcal{GN}$). As another example, if $t^*.\mathcal{A}^s$ is *flu*, which is in SUBTR($t_{tar}.\mathcal{GN}$); therefore, $\mathcal{F}_{percent}(t^*, t_{tar})$ = 100%. Finally, given $t^*.\mathcal{A}^s$ = *stomach-disease* (whose subtree is disjoint with SUBTR($t_{tar}.\mathcal{GN}$)), $\mathcal{F}_{percent}(t^*, t_{tar})$ = 0.

LEMMA 1. *For all tuples t_j^* ($1 \leq j \leq m$) in $\mathcal{EG}(o_{tar})$, $\mathcal{F}_{percent}(t_j^*, t_{tar})$ equals 0 or a constant.*

PROOF. (Sketch) By symmetry, it suffices to prove the lemma for j = 1. As mentioned earlier, if SUBTR($t_1^*.\mathcal{A}^s$) does not overlap SUBTR($t_{tar}.\mathcal{GN}$), $\mathcal{F}_{percent}(t_1^*, t_{tar}) = 0$. Otherwise, we distinguish two scenarios: (i) $t_1^*.\mathcal{A}^s$ is an ancestor of $t_{tar}.\mathcal{GN}$, or (ii) it is in SUBTR($t_{tar}.\mathcal{GN}$). Due to the space constraint, we discuss only the first scenario.

Consider any other tuple t_j^* ($2 \leq j \leq m$). If SUBTR($t_j^*.\mathcal{A}^s$) is disjoint with SUBTR($t_{tar}.\mathcal{GN}$), $\mathcal{F}_{percent}(t_j^*, t_{tar}) = 0$. If not,

we will show that $t_j^*.\mathcal{A}^s = t_1^*.\mathcal{A}^s$, and therefore, $\mathcal{F}_{percent}(t_j^*, t_{tar}) = \mathcal{F}_{percent}(t_1^*, t_{tar})$. Assume, on the contrary, $t_j^*.\mathcal{A}^s \neq t_1^*.\mathcal{A}^s$. Recall that SUBTR($t_1^*.\mathcal{A}^s$) covers the entire SUBTR($t_{tar}.\mathcal{GN}$). Hence, if $t_j^*.\mathcal{A}^s$ has a subtree overlapping SUBTR($t_{tar}.\mathcal{GN}$), $t_j^*.\mathcal{A}^s$ and $t_1^*.\mathcal{A}^s$ become two intersecting partitions in the general domain of $\mathcal{A}^s$. This is not possible, because all the partitions must be disjoint. □

Therefore, in the sequel, we avoid the notation of $\mathcal{F}_{percent}$ by using c to represent the non-zero value of $\mathcal{F}_{percent}(t_1^*, t_{tar})$, ..., $\mathcal{F}_{percent}(t_m^*, t_{tar})$.

3.2 Derivation of the Breach Probability

As clarified in Section 2.3, to infer the $\mathcal{A}^s$ value of o_{tar}, an adversary reconstructs $\mathcal{EG}(o_{tar})$ according to Definition 9 (or 10) in the primal (or non-primal) scenario. In any case, we use n_{recon} to capture the total number of possible reconstructions, and n_{breach} for the number of reconstructions violating the privacy constraint enforced by $t_{tar}.\mathcal{GN}$. It follows that

$$\mathbf{P}_{breach}(t_{tar}) = n_{breach}/n_{recon} \quad (3)$$

The next two theorems solve $\mathbf{P}_{breach}(t_{tar})$ for the primal and non-primary cases, respectively.

THEOREM 1. *In the Primary Case,* $\mathbf{P}_{breach}(t_{tar}) =$

$$\begin{cases} b/n & \text{if } t_{tar}^*.\mathcal{A}^s \text{ is in SUBTR}(t_{tar}.\mathcal{GN}) \\ b \cdot c/n & \text{otherwise} \end{cases}$$

PROOF. (Sketch) We focus on the scenario where $t_{tar}^*.\mathcal{A}^s$ is in SUBTR($t_{tar}.\mathcal{GN}$), as the reasoning extends to the other scenario as well. There are Permu(n, m) ways of setting o_1, ..., o_m (defined in Definition 9) to m persons in $\mathcal{S}_{ext}(o_{tar})$, which has size n. Independently, there exist $\mathcal{F}_{subsize}(t_j^*)$ choices for each v_j ($1 \leq j \leq m$). As a result, $n_{recon} = \text{Permu}(n, m) \cdot \prod_{j=1}^{m} \mathcal{F}_{subsize}(t_j^*)$.

Let t_1^*, ..., t_b^* be all the tuples in $\mathcal{EG}(o_{tar})$, such that the subtrees of their $\mathcal{A}^s$ values overlap SUBTR($t_{tar}.\mathcal{GN}$). In a possible reconstruction violating the privacy requirement of t_{tar}, o_{tar} must be selected as one of o_1, ..., o_b. For each selection, the other elements of o_1, ..., o_m can be set to $m-1$ individuals in $\mathcal{S}_{ext}(o_{tar})$ in Permu($n-1, m-1$) manners. Hence:

$$n_{breach} = b \cdot \text{Permu}(n-1, m-1) \cdot \prod_{j=1}^{m} \mathcal{F}_{subsize}(t_j^*)$$

Then, Equation 3 can be solved as $\mathbf{P}_{breach}(t_{tar}) = b/n$. □

EXAMPLE 3. We illustrate the theorem using Figures 1a, 1b, and 3. Assume t_{tar} (or t_{tar}^*) to be tuple 3 in Figure 1a (or Figure 3). Thus, $t_{tar}^*.\mathcal{A}^s$ = $t_{tar}.\mathcal{GN}$ = *respiratory-infection*, and $\mathcal{EG}(o_{tar})$ involves the first 4 tuples of Figure 3. According to Figure 1b, Andy, Bill, Ken, Nash, Mike are potentially involved in $\mathcal{EG}(o_{tar})$, rendering $n = 5$. Furthermore, $b = 2$, because the subtrees of the $\mathcal{A}^s$ values in tuples 3, 4 (Figure 3) overlap SUBTR($t_{tar}.\mathcal{GN}$). Since $t_{tar}^*.\mathcal{A}^s$ is in SUBTR($t_{tar}.\mathcal{GN}$), by Theorem 1, $\mathbf{P}_{breach}(t_{tar}) = b/n$ = 2/5, confirming the analysis in Example 1.

To demonstrate the second case of the theorem, let t_{tar} (or t_{tar}^*) be tuple 5 in Figure 1a (or Figure 3). Namely, $t_{tar}^*.\mathcal{A}^s$ = *respiratory-infection*, $t_{tar}.\mathcal{GN}$ = *pneumonia*, and $\mathcal{EG}(o_{tar})$ consists of tuples 5, 6 of Figure 3. Only Joe and Sam in Figure 1b can be involved in $\mathcal{EG}(o_{tar})$, leading to $n = 2$. Furthermore, $b = 2$, because the $\mathcal{A}^s$ values of both tuples in $\mathcal{EG}(o_{tar})$ have subtrees overlapping SUBTR($t_{tar}.\mathcal{GN}$). In particular, the subtree of the sensitive value in tuple 5 (or 6) of Figure 3 has 3 leaf diseases, one of which is in SUBTR($t_{tar}.\mathcal{GN}$). Hence, c equals 1/3. Since $t_{tar}^*.\mathcal{A}^s$ is not in SUBTR($t_{tar}.\mathcal{GN}$), $\mathbf{P}_{breach}(t_{tar}) = b \cdot c/n$ = 1/3. □

THEOREM 2. *In the Non-primary Case,* $\mathbf{P}_{breach}(t_{tar}) =$

$$\begin{cases} 1-(1-1/n)^b & \text{if } t_{tar}^*.\mathcal{A}^s \text{ is in SUBTR}(t_{tar}.\mathcal{GN}) \\ 1-(1-c/n)^b & \text{otherwise} \end{cases}$$

PROOF. (Sketch) Again, we discuss only the case that $t_{tar}^*.\mathcal{A}^s$ is in SUBTR($t_{tar}.\mathcal{GN}$). Since each o_j ($1 \leq j \leq m$) in Definition 10 can be set to any of the n individuals in $\mathcal{S}_{ext}(o_{tar})$, and independently, there are $\mathcal{F}_{subsize}(t_j^*)$ choices for each v_j, the total number of possible reconstructions is $n_{recon} = n^m \cdot \prod_{j=1}^{m} \mathcal{F}_{subsize}(t_j^*)$.

Let t_1^*, ..., t_b^* be all the tuples in $\mathcal{EG}(o_{tar})$, such that the subtrees of their $\mathcal{A}^s$ values overlap SUBTR($t_{tar}.\mathcal{GN}$). Since $t^*.\mathcal{A}^s$ is in SUBTR($t_{tar}.\mathcal{GN}$), the $\mathcal{A}^s$ values of t_1^*, ..., t_b^* must also be in SUBTR($t_{tar}.\mathcal{GN}$), according to Lemma 1. In any reconstruction that does *not* cause privacy breach on t_{tar}, o_{tar} must *not* be any of o_1, ..., o_b. In that case, each of o_1, ..., o_b can be assigned to any of the other $n-1$ individuals in $\mathcal{S}_{ext}(o_{tar})$, resulting in $(n-1)^b$ different assignments. For each assignment, o_{b+1}, ..., o_m can be set to any person (including o_{tar}) in $\mathcal{S}_{ext}(o_{tar})$ in n^{m-b} ways. Hence:

$$n_{breach} = n_{recon} - (n-1)^b \cdot n^{m-b} \cdot \prod_{j=1}^{m} \mathcal{F}_{subsize}(t_j^*)$$

Combining the above analysis with Equation 3, we obtain $\mathbf{P}_{breach}(t_{tar}) = 1-(1-1/n)^b$. □

EXAMPLE 4. Let t_{tar} be tuple 3 of Figure 1a. As explained in Example 3, n = 5, b = 2, and $t_{tar}^*.\mathcal{A}^s$ is in SUBTR($t_{tar}.\mathcal{GN}$). Theorem 2 shows that $\mathbf{P}_{breach}(t_{tar})$ is $1-(1-1/5)^2$ = 9 / 25, which is consistent with the derivation in Example 2.

To demonstrate the second case, assume t_{tar} to be tuple 5 in Figure 1a. As mentioned in Example 3, n = 2, b = 2, c = 1/3, and $t_{tar}^*.\mathcal{A}^s$ is not in SUBTR($t_{tar}.\mathcal{GN}$). Thus, $\mathbf{P}_{breach}(t_{tar})$ is $1-(1-1/(3\times 2))^2$ = 11/36. □

3.3 Drawbacks of k-anonymity

A k-anonymous table is obtained only with QI-generalization, i.e., all the $\mathcal{A}^s$ values in the original table T are directly retained. k-anonymity does not consider personal privacy preferences, which is equivalent to setting the guarding node of each tuple $t \in T$ directly to $t.\mathcal{A}^s$. Hence, k-anonymity can be regarded as a special case of our personalized technique.

All the above concepts (e.g., "essential QI-group" and "possible reconstructions") extend to k-anonymity in a natural manner. Therefore, Theorems 1 and 2 also capture the privacy protection quality of k-anonymity. In fact, only the first case (i.e., $t_{tar}^*.\mathcal{A}^s$ is in SUBTR($t_{tar}.\mathcal{GN}$)) of each theorem is necessary, because $t_{tar}^*.\mathcal{A}^s$ = $t_{tar}.\mathcal{GN}$ (= $t_{tar}.\mathcal{A}^s$) always holds. Furthermore, b has a simpler interpretation: it is the number of tuples in $\mathcal{EG}(o_{tar})$ that have the same $\mathcal{A}^s$ value as t_{tar}^*. Next, we use the theorems to explain when and why k-anonymity fails to guarantee safe publication, *even in the scenario with no personal preferences*.

We start with the Primary Case. k-anonymity guarantees that the size m of each QI-group must be at least k. Let us consider the worst scenario, where the adversary has a "perfect" external database such that $\mathcal{S}_{ext}(o_{tar}) = \mathcal{S}_{real}(o_{tar})$, i.e., the external source does not contain any person irrelevant to the microdata. Thus, in Theorem 1, n equals $|\mathcal{S}_{real}(o_{tar})|$, which (for the Primary Case) is equivalent to m. Hence, the breach probability evaluates to b/m. The value of b, however, may reach m, if all the tuples in $\mathcal{EG}(o_{tar})$

have the same $\mathcal{A}^s$ value. When this happens, the breach probability equals 100%, i.e., an adversary can infer the exact information of o_{tar} with full confidence (as is the case explained in Section 1.1 for Joe).

In fact, k-anonymity provides strong protection only if the external database consulted by an adversary may include many individuals that do not exist in the microdata, so that n is by far larger than $|\mathcal{S}_{real}(o_{tar})| = m$. In particular, if the ratio between n and m exceeds b, the breach probability b/n in Theorem 1 is at most $1/m$, which, in turn, is at most $1/k$, i.e., the target protection level of k-anonymity.

Machanavajjhala et al. [11] also observed the above problem, and partially solved it with a new concept of "l-diversity". The essence of l-diversity is to ensure that the sensitive values in each QI-group are sufficiently diverse. Consider that we group the tuples in the QI-group by their sensitive values, and call each resulting group a "sub-group". Assume that p percent of the tuples in the QI-group appear in the largest sub-group. l-diversity ensures that[3] p is at most p_{breach}, the highest permissible breach probability.

Theorem 1 theoretically confirms that the strategy of l-diversity indeed works. In fact, if n equals m, the breach probability b/n is exactly the percentage of tuples in $\mathcal{EG}(o_{tar})$ having the sensitive value $t^*_{tar}.\mathcal{A}^s$ (in other words, l-diversity essentially guarantees $b/m \leq p \leq p_{breach}$). Since, by Inequality 2, n is at least $|\mathcal{S}_{real}(o_{tar})|$ (= m in the Primary Case), l-diversity ensures that b/n is at most p_{breach} for all tuples.

In the Non-primary Case, however, $|\mathcal{S}_{real}(o_{tar})|$ is no longer m; instead, as mentioned in Section 3.1, $|\mathcal{S}_{real}(o_{tar})|$ does not depend on m any more, and can always be 1 regardless of m. As a result, *neither k-anonymity and l-diversity can guarantee low breach probability*. In the worst case, both techniques allow an adversary to obtain the sensitive value of o_{tar} with 100% probability. This happens when o_{tar} is the only person in both $\mathcal{S}_{real}(o_{tar})$ and $\mathcal{S}_{ext}(o_{tar})$, i.e., all the tuples in $\mathcal{EG}(o_{tar})$ concern o_{tar}, and no other individual in the external source can be involved in $\mathcal{EG}(o_{tar})$. As a result, n equals 1, and, by Theorem 2, the breach probability is 1.

What is neglected by k-anonymity and l-diversity? The effect of $|\mathcal{S}_{real}(o_{tar})|$! As discussed earlier, k-anonymity ensures $m \geq k$, and l-diversity guarantees $b/m \leq p_{breach}$, but neither m nor b/m is a component in deriving the breach probability (see Theorem 2). In particular, a major component n is not captured — n can be very small, no matter how large (or small) m (or b/m) is.

4. GENERALIZATION ALGORITHM

Let v be a value in the domain of attribute $\mathcal{A}$. We use $\mathcal{IL}_{value}(v^*)$ to capture the (amount of) information loss in generalizing v to v^*, which is a partition in the corresponding general domain of $\mathcal{A}$ (Definition 5). Formally,

$$\mathcal{IL}_{value}(v^*) = \frac{(\text{the number of values in } v^*) - 1}{\text{the number of values in the domain of } \mathcal{A}} \quad (4)$$

For instance, if the domain of *Age* is $[1, 60]$, generalizing age 5 to $[1, 10]$ has information loss $\mathcal{IL}_{value}([1, 10])$ = (10 − 1) / 60. Similarly, since the taxonomy of *Disease* has 12 leaves, generalizing *flu* to *respiratory-infection* results in $\mathcal{IL}_{value}$(*respiratory-infection*) = $(3 - 1)/12$, where 3 is the number of leaves under *respiratory-infection*. Obviously, if v is not generalized (i.e., $v = v^*$), $\mathcal{IL}_{value}(v^*)$ equals 0, i.e., no information is lost.

[3] l-diversity has other requirements, if an adversary's "background knowledge" is taken into account [11]. We do not consider this complication in this work.

Algorithm *Greedy-Personalized-Generalization*
Input: the microdata T, and the guarding nodes of all tuples
Output: the publishable relation T^*

1. for every QI-attribute $\mathcal{A}^{qi}_i$ $(1 \leq i \leq d)$
2. initialize a generalization function f_i with a single partition covering the entire domain of $\mathcal{A}^{qi}_i$ (see Definitions 4 and 5)
3. T^* = the relation after applying QI-generalization on T according to $S = \{f_1, ..., f_d\}$
4. G' = the only QI-group in T^*
5. *SA-generalization* (G') //Figure 5
   /* at this point, T^* becomes publishable */
6. while (true)
7. $T^*_{best} = T^*$; $S_{best} = S$
8. for every possible $S' = \{f'_1, ..., f'_d\}$ obtained from S with a "single split" (see the explanation in Section 4.1)
9. T'^* = the relation after applying QI-generalization on T according to S'
10. for each QI-group $G' \in T'^*$
11. *SA-generalization* (G') //Figure 5
    /* at this point, T'^* becomes publishable */
12. if $\mathcal{IL}_{table}(T'^*) < \mathcal{IL}_{table}(T^*_{best})$
13. $T^*_{best} = T'^*$; $S_{best} = S'$
14. if ($T^*_{best} = T^*$) then go to Line 17 //no next round
15. else
16. $T^* = T^*_{best}$; $S = S_{best}$ //prepare for the next round
17. return T^*_{best}

Figure 4: Algorithm for computing personalized generalization

The overall information loss $\mathcal{IL}_{tuple}(t^*)$ of a generalized tuple t^* equals

$$w^s \cdot \mathcal{IL}_{value}(t^*.\mathcal{A}^s) + \sum_{i=1}^{d} w^{qi}_i \cdot \mathcal{IL}_{value}(t^*.\mathcal{A}^{qi}_i) \quad (5)$$

where w^{qi}_1, ..., w^{qi}_d, and w^s are positive system parameters, specifying the penalty factor of sacrificing precision on each attribute. Obviously, SA-generalization can be easily disabled by setting $w^s = \infty$, i.e., even the least generalization on $\mathcal{A}^s$ entails infinite information loss.

The total information loss $\mathcal{IL}_{table}(T^*)$ of the entire (generalized) relation T^* is given by

$$\mathcal{IL}_{table}(T^*) = \sum_{\forall t^* \in T^*} \mathcal{IL}_{tuple}(t^*) \quad (6)$$

Next, leveraging the findings of the previous section, we propose an algorithm for computing a generalized table T^* with small $\mathcal{IL}_{table}(T^*)$ which guarantees $\mathbf{P}_{breach}(t) \leq p_{breach}$ for each $t \in T$.

4.1 The Greedy Framework

As elaborated in Section 2.2, our generalization scheme includes two steps. The first phase applies QI-generalization on T, using a set of generalization functions $S = \{f_1, ..., f_d\}$ on the d QI-attributes, respectively. Then, the second step produces the final T^* by performing SA-generalization on the resulting QI-groups, employing a specialized generalization function for each QI-group. Hence, the quality of T^* depends on (i) the choice of S, and (ii) the effectiveness of SA-generalization. We provide a solution for settling the first issue in this subsection, and deal with (ii) in Section 4.2.

A generalization function f_i $(1 \leq i \leq d)$ is decided by a general domain of $\mathcal{A}^{qi}_i$ (Definition 5), which, in turn, is determined by a set of partitions in the original domain of $\mathcal{A}^{qi}_i$ (Definition 4). There-

fore, selecting S is equivalent to finding the appropriate partitions of each f_i. Figure 4 presents a greedy algorithm for achieving this purpose (the pseudocode also explains the framework of calculating T^*).

At Lines 1-2, we obtain the simplest f_i ($1 \leq i \leq d$), which contains a single partition, covering the entire domain of $\mathcal{A}_i^{qi}$. Using such f_1, ..., f_d, Line 3 carries out QI-generalization on T, which, apparently, results in a single QI-group. Next, the algorithm invokes *SA-generalization* (elaborated in the next section) on the QI-group (Lines 4-5), which yields a publishable T^*.

The subsequent execution proceeds in *rounds*. Specifically, each round slightly refines *one* of f_1, ..., f_d, and leads to a new T^* with lower information loss. Before explaining the details, we must clarify the refinement of a function, e.g., f_1, without loss of generality.

Refining a generalization function. Refining f_1 means splitting one of its partitions once. For instance, assume that f_1 is on a numeric attribute *Age* with domain $[1, 60]$, and is determined by partitions $[1, 30]$ and $[31, 60]$. Partition $[1, 30]$ may be split into $[1, x]$ and $[x+1, 30]$, for any $x \in [1, 29]$, i.e., $[1, 30]$ can be split in 29 ways. Similarly, there are also 29 options for splitting $[31, 60]$. Therefore, by a single split, f_1 can be refined into 58 possible generalization functions.

The situation is different, if f_1 concerns a categorical attribute, e.g., *Disease* (strictly speaking, *Disease* is not a QI-attribute in Figure 1c; but no confusion should be caused by borrowing it to illustrate the refinement of f_1). For example, suppose that *respiratory-system-problem* is one of the partitions (in the taxonomy of Figure 2) deciding f_1. Using the transformation stated in Section 2.2, we can represent *respiratory-system-problem* with an interval $[1, 6]$ (by converting the leaf nodes under the partition to values 1-6, respectively). Note that, it is not possible to split the partition into, for instance, $[1, 2]$ and $[3, 6]$. As formulated in Definition 4, each partition of a categorical attribute must be a node in the corresponding taxonomy. Here, $[1, 2]$ cannot be mapped to any node in Figure 2. In fact, there is only one possible split for *respiratory-system-problem*, i.e., breaking its interval $[1, 6]$ to sub-intervals $[1, 3]$ and $[4, 6]$.

In general, the number of possible refinements for a categorical f_1 equals exactly the number of non-leaf partitions of f_1. For example, assuming that f_1 is determined by *respiratory-system-problem* and *digestive-system-problem*, we can refine it into 2 different generalization functions with a single split.

A round of the greedy algorithm. We are ready to elaborate each round of the algorithm in Figure 4. Before a round starts, the algorithm has obtained a publishable table T^*, with a set of QI-generalization functions $S = \{f_1, ..., f_d\}$. At the beginning of the round, we duplicate T^* and S into T^*_{best} and S_{best}, respectively (Line 7).

Next, the algorithm examines (Line 8) all possible sets of refined functions $S' = \{f'_1, ..., f'_d\}$, obtained by performing one split over a single function in S (i.e., S' shares $d-1$ identical functions with S). Given an S', Lines 9-11 perform QI- and SA-generalizations to calculate a publishable T'^*, in the same manner as Lines 3-5, except that multiple QI-groups may be produced after the QI-generalization. If T'^* incurs smaller information loss (computed with Equation 6) than our current best solution T^*_{best} (Line 12), T'^* and S' replace T^*_{best} and S_{best} respectively (Line 13).

We provide a heuristic to reduce computation time. Since S' differs from S in only one element, the QI-generalization based on S' can be computed incrementally from that based on S (which is available from the previous round). Furthermore, if the same QI-group G results from both QI-generalizations, its SA-generation does not need to be re-computed. Similarly, in deriving the information loss $\mathcal{IL}_{table}(T'^*)$, the contribution of the tuples in G needs not be re-calculated, either.

The round finishes, after all S' has been considered. Line 14 checks if a better solution (compared to the one discovered prior to this round) has been found. If not, the algorithm terminates by returning T^*_{best}. Otherwise, another round is executed, after setting T^* (or S) to T^*_{best} (or S_{best}) at Line 16.

4.2 Optimal SA-generalization

Let G' be an arbitrary QI-group output by performing QI-generalization on T. Without loss of generality, assume that G' contains m tuples t'_1, ..., t'_m. We use G to denote the set of corresponding tuples $\{t_1, ..., t_m\}$ in the microdata T. Specifically, for each $j \in [1, m]$, $t'_j.\mathcal{A}^s = t_j.\mathcal{A}^s$, whereas $t'_j.\mathcal{A}_i^{qi}$ generalizes $t_j.\mathcal{A}_i^{qi}$ ($1 \leq i \leq d$).

We aim at applying SA-generation on G' to derive $G^* = \{t^*_1, ..., t^*_m\}$, which achieves two objectives. As discussed in Sections 2.2 and 3, $\mathbf{P}_{breach}(t_j)$ ($1 \leq j \leq m$) depends only on G^* (which is the essential QI-group of the individual that t_j belongs to). Hence, as the first objective, G^* must ensure $\mathbf{P}_{breach}(t_j) \leq p_{breach}$.

The second objective is to minimize

$$\sum_{j=1}^{m} \mathcal{IL}_{value}(t^*_j.\mathcal{A}^s) \tag{7}$$

where $\mathcal{IL}_{value}$ is given in Equation 4. Given the fact that the QI-values of t^*_1, ..., t^*_m have been finalized (before the SA-generalization), fulfilling the second objective essentially minimizes $\sum_{j=1}^{m} \mathcal{IL}_{tuple}(t^*_j)$, where $\mathcal{IL}_{tuple}$ is defined in Equation 5. Therefore, after carrying out SA-generalization on all the QI-groups (produced by QI-generalization) in the same manner, the resulting publishable T^* minimizes $\mathcal{IL}_{table}(T^*)$ of Equation 6.

LEMMA 2. *For any tuples t_x and t_y ($1 \leq x, y \leq m$), if $t_x.\mathcal{GN}$ is in* SUBTR($t_y.\mathcal{GN}$), *then $\mathbf{P}_{breach}(t_x) \leq \mathbf{P}_{breach}(t_y)$ regardless of the SA-generalization applied.*

PROOF. Let b_x (or b_y) be the number of tuples t^*_j ($1 \leq j \leq m$) such that SUBTR($t^*_j.\mathcal{A}^s$) overlaps SUBTR($t_x.\mathcal{GN}$) (or SUBTR($t_y.\mathcal{GN}$)). Since $t_x.\mathcal{GN}$ is in SUBTR($t_y.\mathcal{GN}$), $b_x \leq b_y$. By Theorems 1 and 2, we have $\mathbf{P}_{breach}(t_x) \leq \mathbf{P}_{breach}(t_y)$ (the values of c and n are equivalent in computing the two probabilities). □

Therefore, in searching for the optimal SA-generalization, we can avoid checking the breach probabilities of the tuples like t_x in Lemma 2, because they must be adequately protected once the privacy information of the other tuples is secured.

LEMMA 3. *For any tuple t_j ($1 \leq j \leq m$), if $\mathbf{P}_{breach}(t_j) > p_{breach}$ before SA-generalization, then $t^*_j.\mathcal{A}^s$ must be an ancestor of $t_j.\mathcal{GN}$ after SA-generalization.*

PROOF. (Sketch) Obviously, $\mathbf{P}_{breach}(t_j)$ must have decreased after SA-generalization since it eventually drops below p_{breach}. Assume, on the contrary, that the final $t^*_j.\mathcal{A}^s$ is in SUBTR($t_j.\mathcal{GN}$). Consider the values of b, c, and n in calculating $\mathbf{P}_{breach}(t_j)$ with Theorem 1 or 2. Both c and n remain the same before and after the SA-generalization. Since SA-generalization never reduces b, $\mathbf{P}_{breach}(t_j)$ cannot have decreased after the SA-generalization, leading to a contradiction. □

Based on the above properties, Figure 5 shows an algorithm that finds the optimal SA-generalization for the given QI-group G'.

Algorithm *SA-generalization* (G')
Input: a QI-group G' with tuples t'_1, ..., t'_m after QI-generalization
Output: a set G^* of tuples t^*_1, ..., t^*_m in the final publishable T^*
1. G = the set of tuples t_1, ..., t_m in T generalized to G'; G^* = $\{t'_1, ..., t'_m\}$
2. S_{prob} = the set of tuples $t \in G$ such that $t.\mathcal{GN}$ is not in the subtree of the guarding node of any other tuple in G
3. S_{bad} = the set of tuples $t \in G$ satisfying $\mathbf{P}_{breach}(t) > p_{breach}$ /* In the Primary Case, $\mathbf{P}_{breach}(t)$ is computed from Theorem 1, replacing n with the size of G. In the Non-primary Case, the computation is based on Theorem 2, replacing n with the number of distinct individuals involved in G. */
4. for each tuple $t \in S_{bad}$
5. $t^*.\mathcal{A}^s$ = the parent of $t.\mathcal{GN}$ //t^* is the tuple in G^* corresponding to t
6. for each tuple $t'^* \in G^*$ such that $t'^* \neq t^*$
7. if $t'^*.\mathcal{A}^s$ is in SUBTR($t^*.\mathcal{A}^s$)
8. $t'^*.\mathcal{A}^s = t^*.\mathcal{A}^s$
9. while there is a tuple $t \in S_{prob}$ satisfying $\mathbf{P}_{breach}(t) > p_{breach}$
10. if $t^*.\mathcal{A}^s$ is the root of the taxonomy
11. return NULL //no possible SA-generalization
12. $t^*.\mathcal{A}^s$ = the parent of $t^*.\mathcal{A}^s$

Lines 13-15 are identical to Lines 6-8

Figure 5: Algorithm for finding the optimal SA-generalization

Line 1 initializes two sets G and G^*. G collects all tuples t_1, ..., t_m in T generalized to G', while G^* = G'. Line 2 creates a set S_{prob} as follows. For each tuple $t \in G$, if its guarding node $t.\mathcal{GN}$ is not in SUBTR($t'.\mathcal{GN}$) of any other tuple $t' \in G$, t is added to S_{prob}. By Lemma 2, once the privacy requirements of the tuples in S_{prob} are satisfied, the requirements of the other tuples are also fulfilled.

For each tuple $t \in S_{prob}$, the algorithm calculates $\mathbf{P}_{breach}(t)$ according to Theorem 1 or 2 (based on the current, non-generalized, A^s values in G^*). If $\mathbf{P}_{breach}(t)$ is larger than p_{breach}, t is placed in a set S_{bad} (Line 3), that is, S_{bad} includes the tuples in S_{prob} whose privacy constraints have not been satisfied after QI-generalization.

Next, we consider each tuple $t \in S_{bad}$ in turn (Line 4). Let t^* be its corresponding tuple in G^*. According to Lemma 3, we can immediately set $t^*.\mathcal{A}^s$ to the parent of $t.\mathcal{GN}$ (Line 5). After this, $t^*.\mathcal{A}^s$ may become an ancestor of $t'^*.\mathcal{A}^s$ of another tuple $t'^* \in G^*$. This is not allowed because, otherwise, $t^*.\mathcal{A}^s$ and $t'^*.\mathcal{A}^s$ become two overlapping partitions in the general domain of $\mathcal{A}^s$. To remedy this problem, we must also generalize $t'^*.\mathcal{A}^s$ to $t^*.\mathcal{A}^s$ (Lines 6-8).

The algorithm terminates (Line 9) if $\mathbf{P}_{breach}(t)$ does not exceed p_{breach} for any tuple $t \in S_{prob}$. Otherwise ($\mathbf{P}_{breach}(t) > p_{breach}$ for some tuple t), we must decrease $\mathbf{P}_{breach}(t)$ by generalizing $t^*.\mathcal{A}^s$ further (t^* is the tuple in G^* corresponding to t). If $t^*.\mathcal{A}^s$ is already the root of the taxonomy (Line 10), the algorithm returns, reporting that no appropriate SA-generalization can be found (Line 11). In fact, in this case, the $\mathcal{A}^s$ values of all tuples in G^* have been generalized to the root, so that no more generalization is possible.

If $t^*.\mathcal{A}^s$ is not the root, we raise $t^*.\mathcal{A}^s$ "one level up" in the taxonomy, by replacing it with its parent (Line 12). After this, the $\mathcal{A}^s$ values of some other tuples may also need to be raised, due to the reasoning for Lines 6-8. These changes may increase the breach probabilities of some tuples. Hence, the algorithm returns to Line 9 to check whether any probability is above p_{breach}. If yes, the above procedures are repeated.

The computation of $\mathbf{P}_{breach}(t)$ deserves further clarification. The value of n in Theorems 1 and 2 is unavailable when T^* is being computed (i.e., we do not know which external database will be consulted by an adversary). Hence, as a conservative approach, we replace n with its lower bound $|\mathcal{S}_{real}(o_{tar})|$ (Inequality 2). If the breach probability computed with this lower bound is at most p_{breach}, then the actual breach probability derived by an adversary will definitely be bounded by p_{breach}.

The following theorem proves that Figure 5 produces an SA-generalization that minimizes Equation 7.

THEOREM 3. *Let t^*_1, ..., t^*_m be the tuples returned by the algorithm in Figure 5, and t'^*_1, ..., t'^*_m be the tuples obtained by any alternative SA-generalization that prevents privacy breach. For any $j \in [1, m]$, $t^*_j.\mathcal{A}^s$ must be in* SUBTR($t'^*_j.\mathcal{A}^s$)*, namely, $\mathcal{IL}_{value}(t^*_j.\mathcal{A}^s) \leq \mathcal{IL}_{value}(t'^*_j.\mathcal{A}^s)$.*

PROOF. (Sketch) On the contrary, assume that there exists a hypothetic SA-generalization that violates the theorem. That is, for some $j \in [1, m]$, $t^*_j.\mathcal{A}^s$ is an ancestor of $t'^*_j.\mathcal{A}^s$. Consider the moment, during the algorithm of Figure 5, when the $\mathcal{A}^s$ value of t_j is generalized for the first time. This generalization may happen at Line 5, 8, or 15; in particular, if it is due to Line 8 or 15, the generalization is caused by generalizing the $\mathcal{A}^s$ value of another tuple. Let t be t_j in case of Line 5, or in case of Line 8 or 15, let it be the tuple that causes the generalization of t_j. Now consider the path from $t^*_j.\mathcal{A}^s$ to $t.\mathcal{A}^s$. It can be shown that, for any SA-generalization that ensures privacy protection, none of the nodes on this path can appear as the final $\mathcal{A}^s$ value of any tuple. This, however, contradicts our assumption, because $t'^*_j.\mathcal{A}^s$, obtained by the hypothetic SA-generalization, lies on this path. □

5. EXPERIMENTS

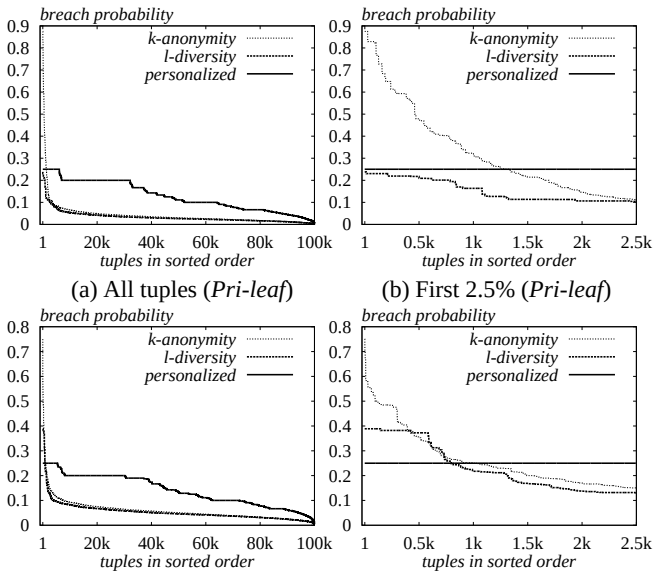
This section experimentally evaluates the effectiveness of our technique using a popular dataset[4] in the literature [5, 7, 8, 9, 11]. The dataset contains a relation with 100k tuples, each storing information of an American adult. The relation has 6 columns: *Age*, *Education*, *Gender*, *Marital-status*, *Occupation*, and *Income*. The first two columns are numerical, whereas *Gender*, *Marital-status*, *Occupation* are categorical; these 5 columns are the QI attributes.

Income is the sensitive attribute, and its values fall in the range of [0, 50k). We categorize the column as follows. First, the domain is evenly divided into 50 ranges (i.e., each has a length of 1k), which constitute the leaves of the taxonomy. Then, every 5 consecutive leaves are grouped as the child of a level-2 node. Recursively, every two level-2 nodes are grouped under a level-3 node. This results in five level-3 nodes, which are the children of the root. Note that the fanouts (5, 2, 5 at levels 2, 3, 4, respectively) are chosen simply to create a balanced taxonomy; other fanouts may also be used, without affecting the experiment results significantly.

We add a unique *ID* to each tuple to obtain a "primary relation" (each individual has exactly one record). Personal references are generated in two ways, leading to datasets *Pri-leaf* and *Pri-mixed*. Specifically, in *Pri-leaf*, the guarding node of each tuple is identical to its sensitive value (i.e., all guarding nodes are leaves of the taxonomy), simulating the scenario where no personal privacy preference is allowed. In *Pri-mixed*, tuples are randomly divided into 3 groups which account for 10%, 30%, and 60% of the relation, respectively. For each tuple in the first (or second) group, its guarding node is the parent of its sensitive value (or is $\emptyset$). The guarding nodes of the tuples in the last group are their sensitive values.

We also synthesize a "non-primary relation" as follows. First, 50k arbitrary persons are sampled from the primary relation, and added to the non-primary relation. Then, among these 50k persons, we extract three disjoint subsets, each containing $50k/3$ random persons. For each person o in the first subset, we create a tuple in

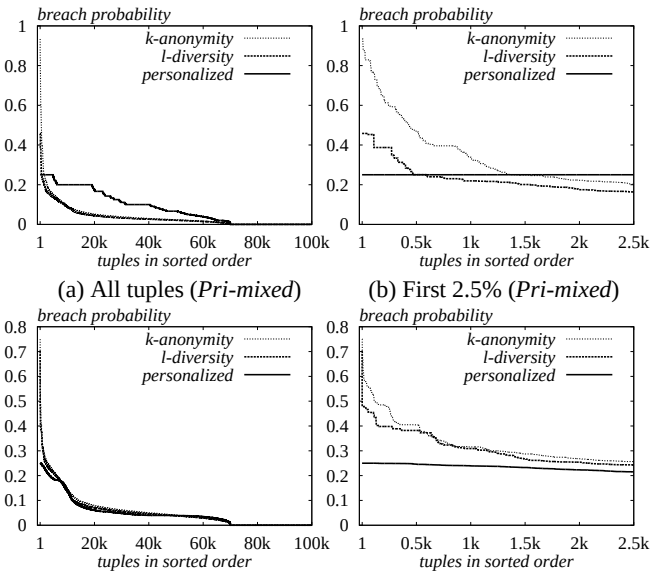
[4]The dataset can be downloaded at *http://www.ipums.org*.



(a) All tuples (*Pri-leaf*) (b) First 2.5% (*Pri-leaf*)

(c) All tuples (*Nonpri-leaf*) (d) First 2.5% (*Nonpri-leaf*)

Figure 6: Tuple breach probabilities (no personalization)



(a) All tuples (*Pri-mixed*) (b) First 2.5% (*Pri-mixed*)

(c) All tuples (*Nonpri-mixed*) (d) First 2.5% (*Nonpri-mixed*)

Figure 7: Tuple breach probabilities (with personalization)

the non-primary relation whose *ID* and QI values are equivalent to those of o, but its sensitive value is generated following the distribution of *Income* in the primary relation. Hence, o corresponds to 2 tuples in the non-primary relation. Similarly, for each person in the second subset, two tuples are created as described earlier, i.e., the person corresponds to 3 tuples. The last subset is directly included in the final relation, whose cardinality is, therefore, 100k. Based on the non-primary relation, we generate personal preferences in the same manner as in *Pri-leaf* and *Pri-mixed*, leading to datasets *Nonpri-leaf* and *Nonpri-mixed*, respectively.

The maximum permissible breach probability p_{breach} is fixed to 0.25. As mentioned in Section 4, our generalization algorithm requires penalty factors w_1^{qi}, ..., w_5^{qi} (for the 5 QI attributes) and w^s. In all cases, w_1^{qi}, ..., w_5^{qi} equal 1. The value of w^s will be varied in different experiments. All the experiments are performed using a Pentium IV CPU at 3.4Ghz.

5.1 Quality of Privacy Protection

In this section, we compare the quality of privacy protection offered by k-anonymity, l-diversity (which improves k-anonymity as mentioned in Section 3.3), and our *personalized* approach. The value of k for k-anonymity equals $1/p_{breach}$ = 4. As with *personalized*, l-diversity takes a single parameter[5] p_{breach} = 0.25. The value of w^s is fixed to 1. In the following experiments, each breach probability is computed from Theorem 1 or 2, replacing n with its lower bound in Equation 2.

In the first experiment, we use the 3 methods to generalize dataset *Pri-leaf*, respectively. In each case, we compute the breach probability of each original tuple with respect to the generalized table. For each method, the probabilities of all tuples are sorted in descending order, as demonstrated in Figure 6a. Since the 3 curves differ primarily in the behavior of the tuples in the first 2.5% of the corresponding sorted lists, Figure 6b plots the probabilities for only these tuples.

k-anonymity cannot achieve the required level of protection, because the breach probabilities of some tuples are significantly higher than p_{breach} = 0.25. As mentioned in Section 1, k-anonymity prevents accurate association between individuals and tuples, but does not provide direct protection against association between individuals and sensitive values. Both l-diversity and *personalized* guarantee adequate preservation. Interestingly, the curve of *personalized* is above that of l-diversity, which indicates that *personalized* performs less generalization, i.e., just enough to meet the privacy requirements. Indeed, the table output by *personalized* has information loss 3123 (calculated with Equation 6), as opposed to 184845 for the l-diverse table.

Figures 6c and 6d illustrate similar results for dataset *Nonpri-leaf*. As predicted in Section 3.3, neither k-anonymity nor l-diversity can satisfy the privacy constraints of all tuples. In particular, k-anonymity (l-diversity) allows some tuples to be inferred with a probability higher than 70% (40%), whereas *Personalized* still guarantees that the breach probabilities of all tuples are bounded by p_{breach}.

In the previous experiments, the guarding node of each tuple equals its original sensitive value, i.e., the no-personalization scenario assumed by the previous work. Next, we study personalized scenarios, by repeating the same experiments on datasets *Pri-* and *Nonpri-mixed*, respectively. For k-anonymity and l-diversity (which are not aware of personal preferences), their generalized tables are identical to those for *Pri-* and *Nonpri-leaf*, respectively.

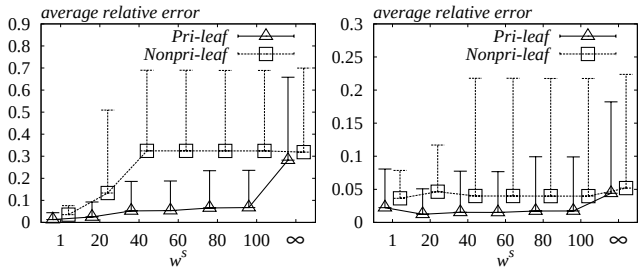
Figure 7 presents the results. Our technique is again the only one that can achieve the required protection degree for an entire dataset. For k-anonymity and l-diversity, as expected, more tuples have breach probabilities above p_{breach}, compared to the results on non-personalized datasets (Figure 6). In Figures 7a and 7c, for each solution, 30% of the tuples have breach probability 0, because they are the tuples whose guarding nodes are $\emptyset$.

In summary, we showed that our solution guarantees privacy preserving in all cases. We also confirmed the finding in Section 3.3 that k-anonymity, as well as its improved version l-diversity, fails to satisfy the privacy requirements in most scenarios. Hence, the two methods are omitted in the subsequent experiments.

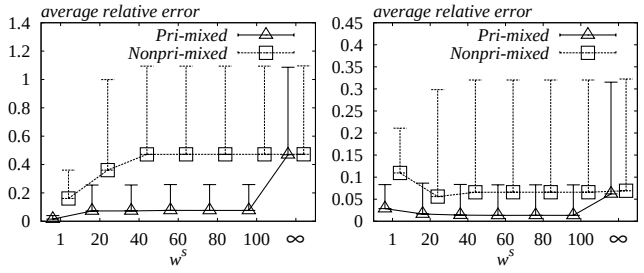
5.2 Accuracy of Data Analysis

In this section, we aim at establishing the fact that SA-generalization is beneficial since, compared to pure quasi-identifer

[5]The parameter p_{breach} has the same functionality as the notation c in [11]. Since we do not consider an adversary's background knowledge, the other parameters of l-diversity are inapplicable.



(a) No personalization, r = 0.1 (b) No personalization, r = 0.9



(c) With personalization, r = 0.1 (d) With personalization, r = 0.9

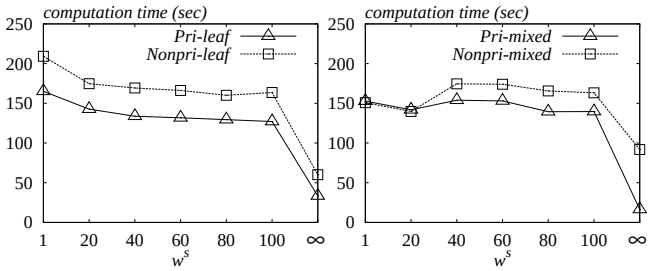
Figure 8: Accuracy of aggregate queries

generalization, it results in generalized tables that permit more accurate data analysis.

We consider *aggregate reasoning* as the goal of data analysis, and examine two types of COUNT-queries related to the sensitive attribute *Income*. A type-1 query retrieves the number of tuples satisfying ik $\leq$ *Income* $<$ (i + 5)k, where i is a random integer in $[0, 45]$. A type-2 query returns the number of tuples qualifying two conditions simultaneously. The first condition is ik $\leq$ *Income* $<$ (i + 10)k, for some $i \in [0, 40]$. The other condition concerns a QI attribute $\mathcal{A}$ arbitrarily selected. If $\mathcal{A}$ is numeric (i.e., *Age* or *Education*), the condition is a range predicate $x \leq \mathcal{A} \leq y$, where $[x, y]$ is a random interval covering 20% of the domain of $\mathcal{A}$. If $\mathcal{A}$ is *Gender* or *Marital-status*, the condition is an equality predicate $\mathcal{A} = x$, where x is an arbitrary value in the domain of $\mathcal{A}$. Finally, if $\mathcal{A}$ is *Occupation*, the condition is also $\mathcal{A} = x$, but x is a random level-1 node in the taxonomy of *Occupation*. The selectivities of all queries are at least 1%.

Given a dataset T, we compute a generalized table T^* with the algorithm in Figure 4, use it to obtain estimated query results, and examine their relative error. Specifically, if est is an estimated result, its relative error equals $|act - est|/act$, where act is the actual query result from T. To derive est, we compute, for each tuple $t^* \in T^*$, the probability p that t^* satisfies the query, after which est is set to the sum of such probabilities of all tuples. Let t be the original tuple in T of t^*. For a type-1 query, p equals the probability that the sensitive value $t.\mathcal{A}^s$ of t falls in the query interval, assuming that $t.\mathcal{A}^s$ is uniformly distributed in $t^*.\mathcal{A}^s$. For a type-2 query, following the same idea, we first obtain the probabilities that t satisfies the two query conditions respectively, using the uniform assumption; then, p equals the product of the two probabilities.

Answering a type-1 query accurately requires retaining as much information on $\mathcal{A}^s$ as possible, while answering type-2 queries accurately demands retaining sufficient information on all attributes. SA-generalization reduces the precision of sensitive values, while preserving more information on the other attributes; hence, it favors type-2 queries. Pure QI-generalization, on the other hand, allows type-1 queries to be precisely processed (i.e., no error), since it retains complete information on $\mathcal{A}^s$. As a tradeoff, it applies much more generalization on the QI attributes, and thus, cannot support type-2 queries as well as SA-generalization.



(a) No personalization (b) With personalization

Figure 9: Cost of personalized generalization

Each *workload* contains 10000 queries. We vary the number of queries of each type using a parameter $r \in [0, 1]$. Specifically, r equals the percentage of type-1 queries in a workload. We inspect two values 0.1 and 0.9 of r; r = 0.1 (0.9) leads to a workload where type-2 (-1) queries are significantly more frequent.

In the experiment of Figure 8a, we set r to 0.1, and deploy the non-personalized datasets *Pri-leaf* and *Nonpri-leaf*. We create generalized tables for a wide spectrum of w^s from 1 to 100. Recall that a larger w^s indicates a higher information-loss penalty of generalizing the sensitive attribute. We also create a generalized table using $w^s = \infty$, which is equivalent to disabling SA-generalization, as mentioned in Section 4. In other words, the performance at $w^s = \infty$ represents the effectiveness of pure QI-generalization.

Figure 8a plots the average error of a workload as a function of w^s. For each average, we also demonstrate the sum of the average and the standard deviation of the errors (in the workload). The sum, depicted as the top of a vertical bar, equals approximately the 8500-highest error. For both *Pri-leaf* and *Nonpri-leaf*, the best w^s equals 1, i.e., we should treat all the attributes equally in generalization. Pure QI-generalization, on the other hand, does not permit robust analysis since it leads to average error nearly 30% (5 times that of w^s = 1), and huge variance. This is expected because a workload with r = 0.1 is populated mostly with type-2 queries, for which pure QI-generalization has poor performance, due to the reasons explained earlier.

Figure 8b shows the results of the same experiment for r = 0.9, i.e., most queries in a workload are type-1 queries. All values of w^s result in average error below 5%. Although pure QI-generalization has small average error, the accuracy of its estimated answers again has significant variance. This phenomenon happens because the generalized table computed by $w^s = \infty$ incurs huge error for type-2 queries (even though type-1 queries can be perfectly processed). On the other hand, the table computed by $w^s = 1$ performs reasonably well for both query types, leading to small average error and variance.

Figures 8c and 8d illustrate the results of the above experiments on the personalized datasets *Pri-mixed* and *Nonpri-mixed* respectively. These results are consistent with those for the non-personalized datasets. In both diagrams, the average error and variance are similar when w^s varies between 40 and 100, because the generalized tables obtained with these values are almost identical.

To summarize, we demonstrated that SA-generalization should be considered in practice. Our experiments suggest that it is reasonable to treat all attributes equally in generalization, which leads to a more useful table for analysis than pure QI-generalization in most cases.

5.3 Computation Cost

Finally, we evaluate the overhead of performing personalized generalization. Figure 9 shows the execution time of our algorithm (Figure 4) in producing the generalized tables used in the experi-

ments of Figure 8, as a function of w^s. The algorithm terminates in less than 4 minutes in all cases. Except for minor random irregularities (of *Pri-mixed* in Figure 9b), the cost decreases as w^s increases. This is because, the higher w^s, the less SA-generalization is possible such that the function of Figure 5 entails smaller overhead.

6. RELATED WORK

Since the introduction of k-anonymity in [13, 15], numerous algorithms [5, 7, 8, 9, 10, 13, 15, 17] have been proposed to obtain k-anonymous tables. These algorithms can be divided into two categories, according to the constraints imposed on generalization. The first category employs "full-domain generalization" [13], which assumes a hierarchy on each QI attribute, and requires that all the partitions in a general domain should be at the same level of the hierarchy. For example, if the value *pneumonia* in Figure 2 is generalized to *respiratory-infection*, then *gastric-ulcer* must also be generalized to *stomach-disease*. Such a constraint is adopted by the binary search algorithm in [13], the exhaustive search method [15], and the apriori-like dynamic programming approach [9], all of which minimize information loss based on various metrics.

The second category (i.e., "full-subtree recoding" as termed in [9]) drops the same-level requirement mentioned earlier, since it often leads to unnecessary information loss [8]. Following this idea, Iyengar [8] develops a genetic algorithm, whereas greedy algorithms are proposed in [7] and [17], based on top-down and bottom-up generalization, respectively. These approaches, however, do not minimize information loss. Bayardo and Agrawal [5] remedy the problem with the power-set search strategy. Our work also belongs to this category, but significantly extends it to incorporate customized privacy needs.

Several other works investigate the characteristics of k-anonymity. For example, Aggarwal [2] discusses the curse of dimensionality related to k-anonymity. In particular, he shows that it is not possible to create even a 2-anonymous table in high dimensional space without considerable information loss. Yao et al. [18] propose a solution for checking whether a set of views violate k-anonymity. Zhong et al. [19] devise a protocol for obtaining k-anonymous tables in distributed environments.

Machanavajjhala et al. [11] observe the first drawback of k-anonymity discussed in Section 1. They propose l-diversity to enhance privacy protection. However, as analyzed in Section 3.3, for the Non-primary Case, this approach may still allow an adversary to discover sensitive data with full confidence.

Recently, Wang et al. [16] present a method which computes the publishable information, by taking into account a set of "templates" specified by data owners. These templates formulate individuals' privacy constraints in the form of association rules. Focusing on the Primary Case, the authors of [16] develop an algorithm that generates the releasable data using "suppression" [3, 12] (as opposed to generalization in this paper).

Finally, it is worth mentioning that privacy preservation can also be achieved using other methodologies, including data perturbation [4], query result perturbation [6], and other earlier solutions proposed in the area of statistics [1].

7. CONCLUSIONS

The existing generalization methods are inadequate because they cannot guarantee privacy protection in all cases, and often incur unnecessary information loss by performing excessive generalization. In this paper, we propose the concept of personalized anonymity, and develop a new generalization framework that takes into account customized privacy requirements. Our technique successfully prevents privacy intrusion even in scenarios where the existing approaches fail, and results in generalized tables that permit accurate aggregate analysis.

This work also lays down a solid theoretical foundation for developing alternative generalization strategies. For instance, the greedy algorithm presented in this paper is not optimal, in the sense that it does not necessarily achieve the lowest information loss. Finding the optimal solution is a challenging problem. As another example, in practice, the recipients of the published data are often specialized users (e.g. scientists), who may explicitly specify the analytical tasks (such as association rule mining [14]) required. This information may be utilized to release a table that is highly effective for those tasks, without breaching the privacy constraints formulated by data owners.

Acknowledgements

This work was fully supported by Grant CityU 1163/04E from the Research Grant Council of the HKSAR government. The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

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

[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT*, pages 246–258, 2005.

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

[5] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE*, pages 217–228, 2005.

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

[7] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *ICDE*, pages 205–216, 2005.

[8] V. Iyengar. Transforming data to satisfy privacy constraints. In *SIGKDD*, pages 279–288, 2002.

[9] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD*, pages 49–60, 2005.

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

[11] A. Machanavajjhala, J. Gehrke, and D. Kifer. l-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[12] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS*, pages 223–228, 2004.

[13] P. Samarati. Protecting respondents' identities in microdata release. *TKDE*, 13(6):1010–1027, 2001.

[14] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *SIGMOD*, pages 1–12, 1996.

[15] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[16] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping attacker's confidence: An alternative to k-anonymization. *To appear in Kowledge and Information Systems*.

[17] K. Wang, P. S. Yu, and S. Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *ICDM*, pages 249–256, 2004.

[18] C. Yao, X. S. Wang, and S. Jajodia. Checking for k-anonymity violation by views. In *VLDB*, pages 910–921, 2005.

[19] S. Zhong, Z. Yang, and R. N. Wright. Privacy-enhancing k-anonymization of customer data. In *PODS*, pages 139–147, 2005.