Ambiguity: Hide the Presence of Individuals and Their Privacy with Low Information Loss

Hui (Wendy) Wang

Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA
hwang@cs.stevens.edu

Abstract

Publishing a database instance containing individual information poses two kinds of privacy risk: *presence leakage*, by which the attackers can explicitly identify individuals in (or not in) the database, and *association leakage*, by which the attackers can unambiguously associate individuals with sensitive information. However, the existing privacy-preserving data publishing techniques that can protect both presence privacy and association privacy have considerable amounts of information loss, while the techniques that produce better utility fail to protect the presence privacy.

In this paper, we propose a novel technique, *ambiguity*, to protect both presence privacy and association privacy with low information loss. We formally define the privacy model and quantify the privacy guarantee of our ambiguity technique against both presence leakage and association leakage. We investigate the information loss of the ambiguity technique and theoretically prove that it always has less information loss than the generalization-based techniques. We accompany the theory with an efficient algorithm that constructs the ambiguity scheme that provides sufficient protection of both presence privacy and association privacy with low information loss. Extensive experiments demonstrate that compared with generalization approach, our ambiguity technique always achieves better accuracy of data analysis.

1 Introduction

Recent years have witnessed increasing volume of released microdata. The release of microdata offers significant advantages in terms of information availability, which make it particularly suitable for ad hoc analysis in a variety of domains such as public health and population studies. But it also raises concerns about risks of releasing individual's private information. There are two kinds of leakage of private information: *presence leakage*, by which an individual is identified to be in (or not in) the microdata, and *association leakage*, by which an individual is identified to be associated with some sensitive information, e.g., a specific disease. As [12] has proven that *knowing an individual is in the database poses a serious privacy risk*, both presence privacy and association privacy are important and must be well protected.

	Released Data			
	Quasi-identifier			Sensitive
Name	**Age**	**Gender**	**Zipcode**	**Disease**
Alan	45	M	11000	diabetes
Charles	20	M	12000	flu
George	50	M	23000	diarrhea
Henry	60	M	12000	stroke
Alice	20	F	54000	leukemia
Carol	50	F	23000	diabetes
Grace	60	F	23000	leukemia
Helen	60	F	21000	dyspepsia

(a) Microdata

Age	Gender	Zipcode	Disease
[20, 60]	M	[11000,23000]	diabetes
[20, 60]	M	[11000,23000]	flu
[20, 60]	M	[11000,23000]	diarrhea
[20, 60]	M	[11000,23000]	stroke
[20, 60]	F	[21000,54000]	leukemia
[20, 60]	F	[21000,54000]	diabetes
[20, 60]	F	[21000,54000]	leukemia
[20, 60]	F	[21000,54000]	dyspepsia

(b) The 3-diversity table

Figure 1: Examples of original and generalized microdata

Recent study has proven that simply removing explicit identifiers, e.g., name and SSN, is insufficient to defend against both presence and association leakage [16]. Due to the existence of some non-ID attributes, e.g., zipcode, gender and date of birth, in the released

International Conference on Management of Data
COMAD 2008, Mumbai, India, December 17–19, 2008
©Computer Society of India, 2008

Age	GroupID
45	1
20	1
50	1
60	1
20	2
50	2
60	2

(a) The auxiliary table AT_1 on QI=Age

Gender	GroupID
M	1
F	2

(b) The auxiliary table AT_2 on QI=Gender

Zipcode	GroupID
11000	1
12000	1
23000	1
54000	2
23000	2
21000	2

(c) The auxiliary table AT_3 on QI=Zipcode

GroupID	Disease	Frequency
1	diabetes	1
1	flu	1
1	diarrhea	1
1	stroke	1
2	leukemia	2
2	diabetes	1
2	dyspepsia	1

(d) The sensitive table *ST*

Figure 2: An example of *ambiguity* scheme

microdata, by using external data (e.g., public voting registration data), the adversaries can easily re-identify the individuals in the microdata and associate them with private information. Those non-ID attributes are called *quasi-identifier* (QI) attributes.

To address the re-identification problem, *generalization-based* techniques [16, 1, 2, 6, 14] divide tuples into groups (*QI-groups*), and transform the data values of quasi-identifier attributes (*QI-values*) into less specific forms, so that the tuples in the same QI-group always have identical QI-values. By generalization, it guarantees that even if publicly available information is linked with records in a microdata database, every individual will be associated with no less than k sensitive values (i.e. *k-anonymity* [16]), and at least l distinct sensitive values (i.e. *l-diversity* [9]). Figure 1 (b) shows an example of a generalized version of the microdata in Figure 1 (a). It consists of two QI-groups, each containing at least three distinct disease values. Therefore the adversaries cannot distinguish which disease an individual truly has. Furthermore, due to the generalization on QI-values, the adversaries cannot easily conclude whether the record of any specific individual exists in the original microdata. In other words, generalization-based techniques can protect both presence privacy and association privacy. However, formal definition and analysis of presence privacy of generalized-based techniques are missing.

One defect of generalization-based techniques is that generalization often results in considerable amount of information loss, which severely compromises the accuracy of data analysis. For example, consider the 3-diversity microdata in Figure 1 (b). Without additional knowledge, the researcher will assume uniform data distribution in the generalized ranges. Let's look at the following aggregate query:

```
Query Q1:
SELECT count(*) from Released-Microdata
WHERE Disease = stroke AND Age ≥ 45;
```

The query falls into the first QI-group in Figure 1 (b) and returns count = 1 for the age range [20, 60]. Since the range [20, 60] covers forty discrete ages, the answer of query Q will be estimated as $1 * \frac{(60-45)}{(60-20)} = \frac{3}{8}$, which is much less than the real answer 1. The error is caused by the fact that the data in every generalized range may significantly deviate from uniformity as assumed. Therefore, generalization prevents an analyst from correctly understanding the data distribution on even a single attribute.

To address the defect of generalization, *permutation-based* techniques (e.g., *anatomy* [18], *k-permutation* [21]) are proposed recently. The basic idea is that instead of generalization of QI-values, the *exact* QI-values are published in one table, while the sensitive attributes are published in another. Then by lossy join of these two tables, every individual will be associated with all distinct sensitive values in the same QI-group (i.e., these sensitive values are permutated). Compared with generalization-based techniques, by publishing the exact QI-values, permutation-based techniques achieve better accuracy of aggregate queries. However, since revealing the exact quasi-identifier values together enables the adversary to easily confirm the presence of any particular individual, *permutation-based techniques fail to protect presence privacy* [3]. It arises the trade-off issue between privacy and data utility: to achieve better utility, some privacy has to be sacrificed. However, as in many applications, privacy always has higher priority than utility. Users may accept data analysis of reasonable amount of inaccuracy but cannot allow leakage of any private information. Therefore, the question is, is there any scheme that can guard both presence privacy and association privacy as generalization-based techniques but with better utility?

1.1 Our Approach: Ambiguity

In this paper, we propose an innovative technique, *ambiguity*, to protect both presence privacy and association privacy with low information loss. Similar to permutation-based techniques, ambiguity publishes the exact QI-values so that it can provide better utility than generalization-based techniques. However, to protect presence privacy, instead of publishing QI-values together, ambiguity publishes them in separate tables. Specifically, for each QI-attribute, ambiguity releases a corresponding *auxiliary table*. In addition, ambiguity releases a *sensitive table (ST)* that contains the sensitive values and their frequency counts in every QI-group. The QI-group membership is included in all auxiliary tables and the sensitive table. Figure 2 illustrates an example of the ambiguity scheme constructed from the microdata in Figure 1 (a).

How can ambiguity protect the presence privacy? Intuitively, it hides the presence of individuals by breaking the associations of QI-values. When the adversary tries to reconstruct the QI-values, by join of the auxiliary tables, he/she will have multiple candidates of QI-values. Due to the lossy join operation, some of the candidates are false match, i.e., they exist in the external database but not in the microdata. For example, assume the adversary knows that Alan is of (Age=45, Gender=M, Zipcode=11000). All these values are present in the QI-group G_1 (i.e., the tuples of group ID 1) in the released ambiguity scheme in Figure 2. Since these values may either all come from Alan's record or from a few other individuals' records, the adversary only can conclude that Alan's record *may* exist in the original microdata. Since G_1 corresponds to 4 tuples in AT_1, 1 tuple in AT_2, and 3 tuples in AT_3, the adversary will have 4*1*3 = 12 tuples from the results of join of all auxiliary tables on the group ID 1. From the *count* attribute in the sensitive table ST, the adversary knows that G_1 consists of 4 tuples in the original microdata. Thus there are $\binom{12}{4}$ number of choices to pick 4 tuples out of 12 combinations, out of which $\binom{11}{3}$ choices contain Alan's record. Without additional knowledge, the adversary's belief probability of Alan's record is present in the microdata is $Pr(Alan \in T) = \binom{11}{3}/\binom{12}{4} = 4/12 = 1/3$, i.e., the adversary cannot explicitly conclude whether Alan's record exists in the original microdata.

How can ambiguity protect the association privacy? The protection of sensitive associations is accomplished by lossy join of auxiliary tables and the sensitive table. For example, to infer whether the association `(Alan, diabetes)` exists in the original microdata, since the reasoning is dependent on the presence of Alan's record, the adversary has to calculate the probability $Pr((Alan, diabetes) \in T \mid Alan \in T) = \frac{Pr((Alan,diabetes) \in T \cap Alan \in T)}{Pr(Alan \in T)}$. We have shown above that $Pr(Alan \in T) = 1/3$. Then to calculate $Pr((Alan, diabetes) \in T \cap Alan \in T)$, the adversary joins the auxiliary tables and the sensitive table on the first QI-group. The result contains 4*1*3*4=48 tuples, which include all possible associations between QI-values and sensitive values in the first QI-group. Again, from the frequency counts in the sensitive table, the adversary learns that this QI-group consists of 4 tuples. Then his/her probability that Alan is associated with diabetes with the assumption that his record is present in the microdata, i.e., $Pr((Alan, diabetes) \in T \cap Alan \in T)$, is $\binom{47}{3}/\binom{48}{4} = 4/48 = 1/12$. Thus without additional knowledge, the adversary's probability $Pr((Alan, diabetes) \in T \mid Alan \in T) = \frac{1/12}{1/3} = 1/4$, i.e., the adversary cannot explicitly decide whether Alan has diabetes.

Approaches	Presence privacy	Association privacy	Information loss
Generalization	Yes	Yes	Worst of 3
Permutation	Yes	No	Best of 3
Ambiguity (our work)	Yes	Yes	In middle

Figure 3: Comparison of Ambiguity with other techniques

How can Ambiguity achieve less information loss than generalization-based techniques? In this paper, we consider the error of count queries as the information loss. Ambiguity achieves less information loss than generalization-based approaches since it releases the exact QI-values. As a result, the estimation of query results based on ambiguity schemes is more accurate than generalized ranges. For example, for the same query Q_1, it matches the first QI-group in Figure 2. There are four distinct ages, three of which (i.e., 45, 50 and 60) satisfy Age $\geq$ 45. Thus the answer will be estimated as 3/4. Compared with the answer by generalization-based approach, which is 3/8, our approach is much closer to the real answer 1.

A brief comparison of our ambiguity technique to both generalization-based and permutation-based techniques is given in Figure 3. We must note that although the ambiguity technique breaks the correlations between the attributes, which results in worse information loss than the permutation-based approaches, this is what we have to sacrifice for protection of the presence privacy. Furthermore, as we will show in Section 6 that the ambiguity technique always produces less information loss than generalization-based technique, thus it is an effective approach of privacy-preserving data publishing.

1.2 Contributions

In this paper, we comprehensively study the ambiguity technique. First, we formalize the ambiguity methodology in Section 3. Ambiguity releases the QI-values and sensitive values in different tables, so that both presence privacy and association privacy are well protected.

Second, we define both presence privacy and association privacy in a unified framework (Section 4). Specifically, we define both presence and association privacy as probabilities, with association privacy probability *conditionally dependent on* the presence privacy probability. We discuss how to measure both presence and association privacy probability for the ambiguity technique (Section 5).

Third, we investigate the information loss of the ambiguity technique. We theoretically prove that the information loss by ambiguity is always better than generalization-based techniques (Section 6).

Fourth, we develop an algorithm to efficiently generate ambiguity schemes that provide sufficient protection to both presence privacy and association privacy. Furthermore, the algorithm is designed in a greedy fashion so that the amount of information loss is controlled as small as possible (Section 7).

Finally, we use extensive experiments to prove both the efficiency and effectiveness of the ambiguity technique (Section 8). Our experiment results demonstrate that ambiguity always achieves better information loss than generalization-based techniques.

The rest of paper is organized as follows. Section 2 describes the related work. Section 3 introduces the background and defines the notions. Section 9 summarizes the paper.

2 Related Work

Privacy-preserving data publishing has received considerable attention in recent years. There are several techniques to anonymize a microdata database to ensure privacy. Most of these techniques can be categorized into two types: *generalization*-based and *permutation*-based.

Generalization-based techniques To protect privacy, the notion *k-anonymity* [16] is proposed recently. It applies suppression/generalization on QI values, so that the QI-values are replaced with less specific ones (e.g., replace specific age with a range of ages). After generalization, every QI-group consists of at least k tuples that are of the same QI-values (e.g., [14, 2, 11]). As the improvement of k-anonymity, new notions (e.g., l-diversity [9], t-closeness [7], (α, k)-anonymity [17]) have been proposed to provide stronger privacy. Similar to k-anonymity, most of them generalize the QI-values to achieve the privacy goal. However, surprisingly, most of them only pay attention to association privacy. Formal definition and technical discussion of the presence privacy is completely ignored. One exception is δ-presence [12]. It defines the presence privacy as probabilities. In particular, given a released microdata T^*, for any individual t, its presence probability $Pr(t \in T \mid T^*) = \frac{m}{n}$, where m is the number of generalized tuples that match the QI-values of t, and n is the number of tuples in the external public dataset, i.e., the presence probability is dependent on the size of the external public dataset. Compared with our work, we assume the data owner is not aware of which external public datasets will be available to the adversary, which is true in many real-world applications. Under this assumption, it is impractical to use the δ-presence privacy model in our work.

Permutation-based techniques Although generalization-based technique can effectively protect privacy, it brings considerable amount of information loss. The situation gets even worse when the microdata contains a large number of QI attributes: due to curse of dimensionality, it becomes difficult to generalize the data without an unacceptably high amount of information loss [1]. To address this defect, a few permutation-based techniques (e.g. anatomy [18], k-permutation [21], bucketization [10]) are recently proposed to protect privacy without any data perturbation. Anatomy [18] releases all QI and sensitive values separately in two tables. By breaking the association between sensitive values and the QI values, it protects the association privacy. Both k-permutation [21] and bucketization [10] techniques first partition the tuples into buckets. Then they randomly permute the sensitive values within each bucket. The permutation disconnects the association between the sensitive values and the QI attributes and thus guard association privacy. As in common, all these permutation-based approaches release the exact QI values, which enable them to achieve better utility than generalization-based techniques. However, releasing the exact QI values in the same table enables the adversary to easily confirm that a particular individual is included in the microdata [3]. Therefore, permutation-based approaches cannot provide enough protection to presence privacy.

3 Background and Notions

In this section, we introduce the notions that are used in the paper.

Let T be a microdata table. Let A denote the set of attributes $\{A_1, A_2, \ldots, A_m\}$ of T and t$[A_i]$ the value of attribute A_i of tuple t. The attribute set A can be categoried as *ID*, *sensitive*, and *quasi-identifier*. ID attribute is used to uniquely identify the individuals. Typical ID attributes include people's name and social security number (SSN). For most of the cases, ID attributes are removed from the released microdata. Sensitive attributes are the attributes, for example, `disease` or `salary`, whose values are considered as sensitive. We use S to denote the set of all sensitive attributes. In the next discussion, we assume there is only one sensitive attribute. Our work can be easily extended to multiple sensitive attributes.

Next, we formally define quasi-identifier attributes.

Definition 3.1 [Quasi-identifier (QI) Attributes] A set of nonsensitive non-ID attributes $\{A_1, \ldots, A_k\}$ of a table is called *quasi-identifier* (QI) attributes if these attributes can be linked with external data to uniquely identify an individual in the general population.

The quasi-identifier attributes of the microdata in Figure 1 (a) is the set {Gender, Age, ZipCode}. Next, we define *QI-groups*.

Definition 3.2 [QI-group] Given a microdata set T, *QI-groups* are subsets of T such that each tuple in T belongs to exactly one subset. We denote QI-groups as $G_1, G_2, \ldots, G_m$. Specifically, $\cup_{i=1}^{m} G_i = T$, and for any $i \neq j$, $G_i \cap G_j = \oslash$.

As an example, for the microdata in Figure 2, $G_1 = \{t_1, t_2, t_3, t_4\}$, and $G_2 = \{t_5, t_6, t_7, t_8\}$.

Now we are ready to formulate ambiguity.

Definition 3.3 [Ambiguity] Given a microdata T that consists of m QI-attributes and the sensitive attribute S, assume T is partitioned into n QI-groups. Then ambiguity produces m *auxiliary* tables (ATs) and a *sensitive table* (ST). In particular,
(1) Each QI-attribute QI_i ($1 \leq i \leq m$) corresponds to a duplicate-free *auxiliary table* AT_i of schema

$$(QI_i, GroupID)$$

Furthermore, for any QI-group $G_j (1 \leq j \leq n)$ and any tuple $t \in G_j$, there is a tuple $(t[QI_i], j) \in AT_i$, where j is the ID of the QI-group QI_j.
(2) The sensitive attribute S corresponds to a *sensitive table* ST of schema

$$(GroupID, S, Frequency)$$

Furthermore, for any QI-group $G_j (1 \leq j \leq n)$ and any distinct sensitive value s of S in G_j, there is a tuple $(j, s, c) \in ST$, where j is the ID of the QI-group QI_j, and c is the number of tuples $t \in G_j$ such that $t[S] = s$.

Figure 2 shows an ambiguity scheme of microdata in Figure 1 (a). It consists of a sensitive table and three auxiliary tables for three QI-attributes *Age*, *Gender* and *Zipcode* respectively.

4 Privacy Model

In this section, we formally define privacy models of both presence privacy and association privacy. First, to address presence privacy, we define *α-presence*. We use $Pr(t \in T)$ to denote the adversary's belief probability of the record of the individual t in the original microdata table T.

Definition 4.1 [α-presence] Given a private microdata T, let T^* be its anonymized version. We say T^* satisfies *α-presence* if for each tuple $t \in T^*$, $Pr(t \in T) \leq \alpha$.

Next, for association privacy, we define *β-association* as adversary's belief of the association between any individual and any sensitive value. We use (t, s) to denote the association between an individual t and a sensitive value s. Since the inference of any private association of a specific individual is based on the assumption of the presence of his/her record in the original microdata, we define the association privacy probability as *conditionally dependent* on the presence privacy probability. Specifically, we define $Pr((t, s) \in T \mid t \in T)$ to denote the adversary's belief probability of the association (t, s) in T with the assumption that the record of the individual t exists in T.

Definition 4.2 [β-association] Given a private microdata T, let T^* be its released version. We say T^* satisfies *β-association* if for any sensitive association $(t, s) \in T^*$, $Pr((t, s) \in T \mid t \in T) \leq \beta$.

Based on both α-presence and β-association, we are ready now to define *(α, β)-privacy.*

Definition 4.3 [(α, β)-privacy] Given a private microdata T, let T^* be its released version. We say T^* is *(α, β)-private* if it satisfies both α-presence and β-association.

Given a microdata T and two privacy parameters α and β, our goal is to construct an (α, β)-private scheme T^* of T.

5 Privacy of Ambiguity

In this section, we elaborate the details of quantifying both presence and association privacy of an ambiguity scheme. We start from the measurement of the presence privacy.

5.1 Measurement of Presence Privacy

Intuitively the adversary infers the presence of a tuple in the original microdata if all of its QI-values in the external public database exist in the released ambiguity tables. We first define *cover* to address this. We use t^{QI} to denote the QI-values of the tuple t.

Definition 5.1 [Cover] Given a microdata T and an ambiguity scheme T^* (AT_1, ..., AT_k, ST). Let J be the result of $\bowtie_{i=1}^{k} AT_i$, where $\bowtie$ is an equal-join operator. We say a tuple $t \in T$ is *covered* by T^* if $t^{QI} \in J$.

Based on the semantics of equal join, it is straightforward that a tuple is covered if every piece of its QI values is contained in one auxiliary ambiguity table. We use $t[QI_i]$ to indicate the i-th QI value of the tuple t.

Lemma 5.1 (Cover) : Given a microdata T and an ambiguity scheme T^* (AT_1, ..., AT_k, ST), let AT_i be the auxiliary table that contains the i-th QI attribute QI_i. Then a tuple $t \in T$ is *covered* by T^* if and only if there exists a QI-group G_j s.t. for each QI-value QI_i of t, $(t[QI_i], j) \in AT_i$. In particular, we say t is covered by QI-group G_j.

For example, Alice's record is covered by the second QI-group in the ambiguity scheme in Figure 2. However, due to lossy join, the adversary only can conclude that Alice's record *may* exist in the original microdata. In the join result of the auxiliary tables in Figure 2 on group ID 2 (i.e., the group that covers Alice's record). there are 3*1*3 = 9 combinations of QI-attributes, some of them are false match and do not exist in the original microdata. Furthermore, the count frequencies in the sensitive table ST infers that there are four individuals in this QI-group. Therefore there are $\binom{9}{4}$ choices to choose 4 individuals from these 9 combinations, out of which $\binom{8}{3}$ choices contain Alice's record. Thus the probability that Alice's record exists in the microdata is $\binom{8}{3}/\binom{9}{4} = 4/9$. We use $\mid G \mid$ to denote the number of tuples in QI-group G, and $\mid G_{AT_i} \mid$ as the number of tuples of QI-group G in the auxiliary table AT_i. Then in general,

Theorem 5.1 (Presence Probability) : Given a microdata T and an ambiguity scheme $T^*(AT_1, \ldots, AT_k, ST)$, for any individual tuple $t \in T$ that is covered by T^*, let G be the QI-group that covers t. Then the presence probability $Pr(t \in T)$ $= \mid G \mid /(\Pi_{i=1}^{k} \mid G_{AT_i} \mid)$.

Theorem 5.1 shows that the presence probability can be decreased by increasing $\mid G_{AT_i} \mid$, the size of QI-group in AT_i, and/or reducing $\mid G \mid$, the size of QI-group. We follow this principle when we design the ambiguity scheme of low information loss. More details are in Section 7.

5.2 Measurement of Association Privacy

Definition 4.2 has defined the association privacy as a conditional probability. It is straightforward that

$$Pr((t,s) \in T \mid t \in T) = \frac{Pr((t,s) \in T \cap t \in T)}{Pr(t \in T)}$$

We have discussed how to measure $Pr(t \in T)$ in Section 5.1. Next, we discuss how to compute $Pr((t,s) \in T \cap t \in T)$.

Join of the auxiliary tables and the sensitive table on group IDs will result in a table of schema (QI_1, …, QI_k, S, $GroupID$), where QI_i is the ith QI-attribute ($i \leq k$), and S are the sensitive attributes. Due to lossy join on group IDs, in this table, each QI-value is associated with all sensitive values in the same QI-group. For example, by matching Alice's QI-values with the released ambiguity tables in Figure 2, the adversary knows that if Alice's record is present in the original microdata, it must exist in the second QI-group. First, the join of the auxiliary tables and the sensitive table on group ID 2 will construct 3*1*3*(2+1+1) = 36 tuples. Second, the count frequencies in the sensitive table ST indicates that there are four tuples in this group. Therefore, there are $\binom{36}{4}$ choices to choose four tuples. If the adversary assumes Alice's record is present in the microdata and he/she is interested with the association `(Alice, leukemia)`, since the frequency count of leukemia is 2, there will be $\binom{2}{1}*\binom{35}{3}$ choices that contain `(Alice, leukemia)`. Without additional knowledge, the probability $Pr((Alice, leukemia) \in T \cap (Alice \in T))$ is $\binom{2}{1}*\binom{35}{3}/\binom{36}{4}$. This is formally explained in the next lemma. Again, we use $\mid G \mid$ to denote the number of tuples in QI-group G, and $\mid G_{AT_i} \mid$ as the number of tuples of QI-group G in the auxiliary table AT_i.

Lemma 5.2 Given a microdata T and an ambiguity scheme $T^*(AT_1, \ldots, AT_k, ST)$, for any individual tuple $t \in T$ that is covered by T^*, let G be the QI-group that covers t. Let c be the count of the sensitive value s in G. Then the probability $Pr((t,s) \in T \cap t \in T) = c/(\Pi_{i=1}^{k} \mid G_{AT_i} \mid)$.

Now we are ready to measure the association privacy. We use the same denotation as above.

Theorem 5.2 (Association Privacy) : Given a private microdata T and an ambiguity scheme T^* ($AT_1, \ldots, AT_k$, ST), for any tuple $t \in T^*$, let G be its covered QI-group. Then the association privacy $Pr((t,s) \in T \mid t \in T) = c/ \mid G \mid$, where c is the frequency count of the sensitive value s in G.

Theorem 5.2 shows that for the association between any individual and a sensitive value s, its association probability is decided by the frequency of the sensitive value s and the sum of frequency counts of all distinct sensitive values in the QI-group that the individual belongs to. For instance, given the ambiguity tables in Figure 2, $Pr((Alice, leukemia) \in T \mid Alice \in T) =$ 2/4=1/2.

6 Information Loss of Ambiguity

In this paper, as the same as in [13, 18], we consider the error of the result of count queries as information loss. Specifically, let Q be a count query, $Q(T)$ and $Q(T^*)$ be the accurate and approximate result by applying Q on the original microdata T and the released ambiguity table T^*. The relative error $Error = \frac{|Q(T)-Q(T^*)|}{|Q(T)|}$. Next, we explain how ambiguity estimates $Q(T^*)$.

Given the released ambiguity scheme $(AT_1, \ldots, AT_k, ST)$, for any count query Q = count($\sigma_C(AT_1 \bowtie \ldots AT_k \bowtie ST)$), where C is a selection condition statement, we approximate $Q(T^*)$ by applying estimation on every individual table AT_i. Before we explain the details, we first define the notions. Given the ambiguity scheme $(AT_1, \ldots, AT_k, ST)$, and a counting query Q with the selection condition C, we use $C_i (1 \leq i \leq k)$ and C_S to denote the results of applying projection of the scheme of table AT_i and ST on the selection condition

C. We only consider C_i and C_S that are not null. For example, for C = "`Age`$\geq$`55 and Disease=stroke`" on the ambiguity scheme in Figure 2, C_1 (on AT_1) = "`Age`$\geq$`55`", and C_S (on ST) = "`Disease=stroke`".

The pseudo code in Algorithm 1 shows the details of how to approximate the result of counting queries. First, we locate all the QI-groups that satisfy C_S (Line 1). Second, for every returned QI-group G_j, we estimate the count result. In particular, we compute the count result c, i.e., the number of tuples in G_j that satisfies C_S in the sensitive table ST (Line 6). Then for every selection condition C_i on the AT table AT_i $(1 \leq i \leq k)$, we calculate the percentage p of tuples in G_j that satisfy C_i, and adjust the count result accordingly by multiplying c with p (Line 7 - 9). Last, we sum up the adjusted counts for all QI-groups (Line 9).

Note that generalization-based techniques use the same approach to estimate the results of count queries. Their percentage p is defined as the size of the range of the generalized tuples that satisfy the selection condition C over the size of the whole range. An example has been given in Section 1.

Algorithm 1 Algorithm: Estimation of Answers of Counting Queries

```
Require: Ambiguity tables $(AT_1, \ldots, AT_k, ST)$, query $Q$;
Ensure: The estimated answer of $Q$.
 1: $GID \leftarrow \Pi_{GroupID}\sigma_{C_S}(ST)$;
 2: $n \leftarrow 0$, i $\leftarrow 1$;
 3: for all $i \leq k$ do
 4:     $l \leftarrow 0$, $m \leftarrow 0$;
 5:     for all Group ID $j \in GID$ do
 6:         $c \leftarrow$ count($\sigma_{(C_S, GroupID=j)}ST$);
 7:         $l \leftarrow$ count($\sigma_{(C_i, groupID=j)}AT_i$) ;
 8:         $m \leftarrow$ count($\sigma_{(groupID=j)}AT_i$);
 9:         $n \leftarrow n + c * l/m$
10:     $i \leftarrow i + 1$;
11: Return $n$.
```

Example 6.1 We explain how to use Algorithm 1 to estimate the results of count queries by using the ambiguity scheme in Figure 2. For the query Q_2:

```
SELECT count(*) from Released-Microdata
WHERE Age≥50 AND Zipcode=23000 AND Disease=diabetes;
```

Both QI-group 1 and 2 satisfy the condition *disease = diabetes* on ST. For QI-group 1, the count is estimated as $1 * \frac{2}{4} * \frac{1}{3} = \frac{1}{6}$, where $\frac{2}{4}$ corresponds to 2 ages (out of 4) that satisfy `Age`$\geq$ `50` in table AT_1, and $\frac{1}{3}$ corresponds to 1 zipcode (out of 3) that satisfy `Zipcode = 23000` in table AT_2. Similarly, for QI-group 2, the count is estimated as $1 * \frac{2}{3} * \frac{1}{3} = \frac{2}{9}$. The final answer is $\frac{1}{6} + \frac{2}{9} = \frac{7}{18}$.

Estimation of query answers brings information loss. With QI-groups of fixed size, it is straightforward that the fewer tuples in every auxiliary table that satisfy the queries, the worse the information loss will be. However, no matter how worse it is, the information loss by the ambiguity technique is always less than that by generalization-based approaches. We have:

Theorem 6.1 (Information Loss: ambiguity V.S. generalization) : Given a microdata T, let T_G be the generalized table of T. Then there always exists an ambiguity scheme T_A such that for any count query Q, the relative error of answering Q by using T_A is less than that by T_G.

Proof: We construct T_A by following: for any QI-group G_i in T_G, we construct the corresponding ambiguity auxiliary tables and sensitive tables. Then we prove that the union of these auxiliary tables and sensitive tables construct the ambiguity scheme T_A that always achieves less information loss than T_G. For each auxiliary table $AT_i (1 \leq i \leq k)$, and for each QI-group G_j in AT_i, let m_{ij} be the cardinality of G_j in AT_i and l_{ij} be the count result by applying selection condition C_i on G_j in AT_i. Let n_j be the count result by applying C_S on the QI-group G_j in the sensitive table ST. Then the estimation result on G_j in AT_i is $(n_j * l_{ij})/m_{ij}$. Assume the data values in G_j of AT_i are generalized to R_{ij}. Let r_{ij} be the size of the generated range R. Then the estimation result on G_j in AT_i is $n_j * l_{ij}/r_{ij}$. Since for each G_j of AT_i, it is always true that $G_j \subseteq R_{ij}$, i.e., the generalized range of the QI-group always consists of all the tuples in the group, it is straightforward that $r_{ij} \geq m_{ij}$. Therefore for every QI-group in each ambiguity auxiliary table, its estimated result is larger than that by generalization. It follows that $Q(T_A) \geq Q(T_G)$. Consequently the relative error by ambiguity is always less than that by generalization approaches. ▮

We also have experiment results to prove that our ambiguity approach always wins generalization-based approaches with regard to information loss. More details can be found in Section 8.3.

7 Ambiguity Algorithm

In this section, we explain the details of our ambiguity algorithm. The purpose of the algorithm is to construct an (α, β)-private scheme with small information loss (α and β are two given privacy parameters). The essence of the algorithm is to partition the microdata T into multiple non-overlapping QI-groups, each of which meets α-presence (by Theorem 5.1) and β-association (by Theorem 5.2). Since the amount of information loss increases when the size of QI-groups grows, to reduce the information loss, we construct the QI-groups that are of sizes as small as possible. Next, we discuss the details of the ambiguity algorithm. Our algorithm consists of three steps. Algorithm 2 shows the pseudo code of the algorithm.

Algorithm 2 Ambiguity Table Construction

```
Require: A microdata $T$, two privacy parameters $\alpha$ and $\beta$.
Ensure: Ambiguity scheme $(AT_1, \ldots, AT_k, ST)$ that satisfies $\alpha$-presence and $\beta$-association.
   {//Step 1: Bucketization}
 1: for all Attribute $A_i$ do
 2:     Hash the tuples into buckets by values on $A_i$, each bucket per value;
 3: Sort the buckets on sensitive values by their sizes in descending order;
   {//Step 2: Construction of safe QI-groups from the hashed buckets}
 4: $GS \leftarrow \{\}$; {$GS$: stores set of QI groups}
 5: Let $H_{QI_i}$ and $H_s$ be the hashed buckets on QI-attribute $QI_i$ and sensitive attribute $S$;
 6: $HS \leftarrow (\cup H_{QI_i}) \cup H_S$; {$HS$: Unpicked tuples}
 7: repeat
 8:     $G \leftarrow \{\}$;{$G$: a QI-group}
 9:     $Pr \leftarrow 1$; {$Pr$: probability}
10:     for $i$= 0 to $\lceil 1/\beta \rceil$ do
11:         $G \leftarrow G \cup pick(G, HS)$;
12:     $Pr \leftarrow$ CalPPro($G$) (Algorithm 3);
13:     while $Pr > \alpha$ do
14:         $G \leftarrow G \cup pick(G, HS)$ (Algorithm 4);
15:         $Pr \leftarrow$ CalPPro($G$) (Algorithm 3);
16:     Add $G$ into $GS$;
17:     Remove all tuples in $G$ from $HS$. {Update $HS$ for the grouped tuples;}
18: until There are less than $i \leq \lceil 1/\beta \rceil$ non-empty buckets in $H_S$
   {//Step 3: Process the residues}
19: if $\exists\ 0 < k < 1/\beta$ non-empty buckets in $H_S$ then
20:     for all non-empty hash bucket $B$ of $H_s$ do
21:         for all tuple $t \in B$ do
22:             for all QI-group $G \in GS$ do
23:                 if $t[S] \notin G$ and CalPPro($G \cup \{t\} \leq \alpha$) then
24:                     $G \leftarrow G \cup \{t\}$;
25:                     remove $t$ from $H_s$;
26:                     Break;
27: Construct ambiguity scheme $T^*$ based on $GS$;
28: Return $T^*$;
```

Step 1: Bucketize on QI and sensitive values. The first step of ambiguity is to bucketize the values into smaller units, so that the following construction procedure will be more efficient on a smaller search space. Intuitively for each attribute, its values will be bucketized, so that every bucket contains the tuples that are of the same value. The buckets can be constructed by hashing the tuples by their sensitive values (Line 2 of Algorithm 2). Each hashed value corresponds to a bucket. We require that for n distinct values, there exists n hashed buckets, so that different values won't be hashed into the same bucket. After the bucketization, we sort the buckets on the sensitive attributes in descending order by the size of the buckets, i.e., the number of tuples in the buckets (Line 3 of Algorithm 2). The reason for sorting is to put higher priority on sensitive values of large number of occurrences, so that in the later steps of QI-group construction, these values will be picked earlier and scattered more sparsely across QI-groups. As the result the occurrence of these values in each QI-group is minimized. Since small frequency occurrences incur both small presence privacy probability and association privacy probability, such design enables earlier termination of construction of QI-groups that satisfy (α, β)-privacy so that QI-groups are of smaller sizes. Consequently the amount of information loss is reduced. Figure 4 shows the bucketized result of Figure 1. The integer numbers on the right side of → indicate the bucket IDs.

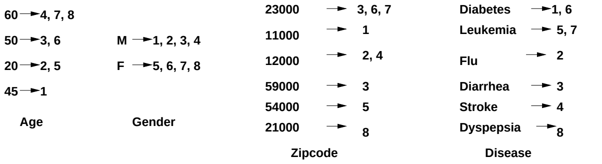


Figure 4: Bucketization

Based on the bucketization result, we can compute the presence probability as following: for QI-group G, let m_i and n be the number of buckets that G covers for the i-th QI-attribute QI_i and the sensitive attribute. Then following Theorem 5.1, the presence probability equals $n/\Pi_{i=1}^{k} m_i$, where k is the number of QI-attributes. For example, the QI-group in Figure 2 that contains both tuple 1 and 2, which covers 2 buckets for *Age*, 1 for *Gender*, 2 for *Zipcode*, and 1 for *Disease*, will result in the presence probability of 1/(2*1*2) = 1/4. Algorithm 3 shows more details. We use H_{QI_i} and H_s to denote the hashed buckets on QI-attribute QI_i and the sensitive attribute S. The reason why we only compute the presence privacy but not association privacy is that we can make the QI-groups meet β-association requirement by controlling the sizes of QI-groups. More details are in Step 2 and Step 3.

Algorithm 3 CalPPro(G): Calculation of Presence Probability of QI-group G

```
1: Let $m_i$ be the number of hash buckets that $G$ covers for $H_{QI_i}$;
2: Let $n$ be the number of hash buckets that $G$ covers for $H_S$;
3: Return $n/\Pi_{i=1}^{k} m_i$.
```

Step 2: Construct (α, β)-private QI-groups from hashed buckets It is straightforward that for each QI-group, the more buckets it covers, the smaller the presence probability will be. Therefore, when we pick the tuples and add them into the QI-group, we always pick the ones that cover the maximum number of buckets, i.e., produce the minimum presence probability. Algorithm 4 shows more details.

Algorithm 4 pick(G, HS): pick a tuple that will cover the maximum number of buckets with the tuples in $G \cup \{t\}$ by using hash buckets HS

```
$max \leftarrow 100000$; $picked \leftarrow$ null;
for all unpicked tuple $t \in T$ do
    Let $m$ be the number of hash buckets in $HS$ that $G \cup \{t\}$ covers;
    if $m < max$ then
        $max \leftarrow m$; $picked \leftarrow t$;
Return $picked$.
```

Given two privacy parameters α and β, we construct QI-groups in a greedy fashion: starting from the buckets consisting of the largest number of unpicked tuples, we pick $\lceil 1/\beta \rceil$ tuples from $\lceil (1/\beta) \rceil$ buckets on the sensitive values, a tuple from a bucket (Line 10 - 11 of Algorithm 2). We pick the tuples by calling *pick*() function (Algorithm 4), so that the picked tuples will cover the maximum number of possible buckets, i.e., produces the minimum presence probability. We calculate the presence probability of the picked $\lceil 1/\beta \rceil$ tuples. If the presence probability does not satisfy the α-presence requirement, we keep picking tuples following the same principle, until the presence probability reaches the threshold α (Line 13 - 15 of Algorithm 2). By this greedy approach, the α-presence requirement will be met early and QI-groups of smaller size will be constructed, which will result in the information loss of smaller amount. We repeat the construction of QI-groups until there are less than $\lceil 1/\beta \rceil$ non-empty buckets, i.e., there are not enough tuples to construct a QI-group of size $\lceil 1/\beta \rceil$.

Step 3: Process the residues After Step 2, there may exist residue tuples that are not assigned to any QI-group. In this step, we assign these residue tuples to the QI-groups that are constructed by Step 2. Adding tuples to the QI-groups will influence both presence and association probabilities. Thus for every residue tuple t, we add it to the QI-group G if: (1) the sensitive value of tuple t is not included in G originally, and (2) the presence probability of the QI-group $G \cup \{t\}$ is less than α (Line 22 - 24 of Algorithm 2). We have:

Theorem 7.1 Given a microdata T, let T^* be the ambiguity scheme that is constructed by Algorithm 2. Then T^* is (α, β)-private.

Proof. The proof of α-presence is from the construction procedure. The construction of QI-groups terminates only when the α-presence is satisfied. Adding residue tuples is also aware of α-presence requirement. Thus the constructed QI-groups always satisfy α-presence. The proof of β-association is the following. Since each bucket corresponds to a unique sensitive value, by our construction approach, every sensitive value in every QI-group has only one occurrence, which results that the sum of frequency counts in every QI-group must be at least $\lceil 1/\beta \rceil$, i.e., step 2 always produces QI-groups that satisfy β-association. Furthermore, adding residue tuples of unique sensitive values to QI-groups by Step 3 only decreases the association probability. Thus the QI-groups still meet the β-association requirement. ▮

Following our construction procedure, the ambiguity scheme has the following privacy property.

Theorem 7.2 (Ambiguity VS. *l*-diversity) : Given a private microdata T, let T^* be the ambiguity scheme that is constructed by Algorithm 2. Then T^* satisfies $(\lceil 1/\beta \rceil)$-diversity.

Proof: By Algorithm 2, since in each QI-group G, every sensitive value only has 1 number of occurrence, and there are at least $\lceil 1/\beta \rceil$ tuples in G, G consists of at least $\lceil 1/\beta \rceil$ distinct sensitive values, i.e., G satisfies $(\lceil 1/\beta \rceil)$-diversity. ▮

8 Experiments

We ran a battery of experiments to evaluate the efficiency and effectiveness of ambiguity technique. In this section, we describe our experiments and analyze the results.

8.1 Experimental Setup

Setup We implement the ambiguity algorithms in C++ and use a workstation running Linux RedHat version 2.6.5 with 1 processor having a speed of 2.8GHz and 2GB of RAM. We use the multi-dimension k-anonymity generalization algorithm implemented by Xiao et al.[18][1].

Dataset We use the *Census* dataset that contains personal information of 500,000 American adults[2]. The details of the dataset are summarized in Figure 5. From the Census dataset, we create two sets of microdata tables, *Occ* and *Sal*, with the *Occ* set using *Occupancy* as sensitive attribute and *Sal* using *Salary*. For each set, we randomly pick 100k, 200k, 300k, 400k and 500k tuples from the full set and correspondently construct tables as *Occ*-n and *Sal*-n ($n = 100$, 200, 300, 400, 500).

To study the impact of distributions, we also generate a set of files with various distributions. We construct 10 datasets *Occ-100k*-d and *Sal-100k*-d ($1 \leq d \leq 5$), each of 100k tuples. The parameter d is used to specify that the sensitive values are distributed to $(100/d)$% of tuples. In other word, d controls the degree of density. The larger the d is, the denser the dataset is.

[1] The source code is downloaded from http://www.cse.cuhk.edu.hk/ taoyf/paper/vldb06.html

[2] http://www.ipums.org/

Attribute	Number of distinct values
Age	78
Gender	2
Education	17
Marital	6
Race	9
Work Class	10
Country	83
Occupation	50
Salary-class	50

Figure 5: Summary of Attributes

Queries We consider the count queries of the form.

```
SELECT $QT_1, \ldots, QT_i$, count(*)
FROM data
WHERE $S$ = $v$
GROUP By $QT_1, \ldots, QT_i$;
```

Each QT_i is a QI-attribute, whereas S is a sensitive attribute. We randomly pick $QT_1, \ldots, QT_i$, vary the value v and create three batches of query workload $Query - i$ ($1 \leq i \leq 3$), where $i = 1, 2$ and 3 correspond to the query selectivity of 1%, 5% and 10%.

8.2 Performance of Generating Ambiguity Scheme

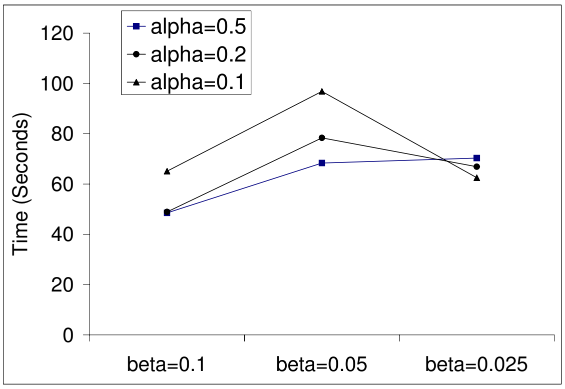


Figure 6: Performance: various α and β values

First, we vary the values of α and β for α-presence and β-association requirements. Figure 6 shows the result of *Occ* datasets with size 500k. It demonstrates that the performance is not linear to either α or β. This is because the time complexity of the ambiguity algorithm equals $t_c * m$, where t_c is the time complexity of function $CalPPro()$ (Algorithm 3) and m is the number of QI-groups. Smaller QI-groups will result in smaller t_c but larger m, i.e., $t_c * m$ is not linear to the size of QI-groups. Therefore although α and β decide the size of QI-groups, they cannot decide the performance. We examined the other *Occ* datasets with different sizes as well as *Sal* datasets and got the similar results. For simplicity, we omit the results.

Second, we examine the performance on datasets of different distributions. Figure 7 shows the result of *Occ* datasets. We observe that the sparser the dataset is, the better performance will be. This is because with sparser datasets, the QI-groups will cover more distinct values and as a result will yield better presence probability. Therefore it will meet the α-presence requirement earlier without additional computation to search for appropriate tuples to add into QI-groups. The same phenomenon also hold for *Sal* datasets.

8.3 Information Loss

We process each query workload $Query - i$ ($i = 1$, 2, 3 correspond to the query selectivity of 1%, 5% and 10%) on the resulting tables and measure the average of the relative errors. As explained in Section 6, for each query, its relative error equals $act \mid - \mid est \mid /act$, where act is its actual result derived from the microdata, and est the estimate computed from the ambiguity/generalized table. The details of measurement of $\mid est \mid$ for both generalized tables and ambiguity technique are explained in Section 6.

The first set of this part of experiments compares the accuracy of query results of ambiguity technique with that of generalization technique on datasets with various β values for β-association. Figure 8 shows the result. As expected, ambiguity always produces better accuracy of query results than generalization. This is because ambiguity preserves a large amount of precision of data values. Furthermore, when β increases, both ambiguity and generalization have growing information loss. This is because larger β results in bigger QI-groups for both ambiguity and generalization, which increases the size of QI-groups for ambiguity and the size of generalized range for generalization. However, since the generalized ranges always grow faster than number of distinct tuples, our ambiguity technique always has much better accuracy than generalization.

Second, we measure the accuracy of queries involving 3, 4 and 5 attributes in the selection conditions. The attributes are chosen randomly. The results are shown in Figure 9 (1). Although for both ambiguity and generalization, the accuracy of query answers degrades when there are more attributes in the selection conditions of queries, ambiguity always wins generalization regarding information loss.

We also measure the impact of the data distribution to the information loss of ambiguity. Figure 9 (2) shows that the denser datasets deliver worse accuracy. The reason is that the dense datasets will produce QI-groups of larger size, which consequently results in worse accuracy of query results. However, Figure 10 shows that ambiguity always has better information loss than generalization technique for various data distributions.

As a brief summary, we showed that our ambiguity

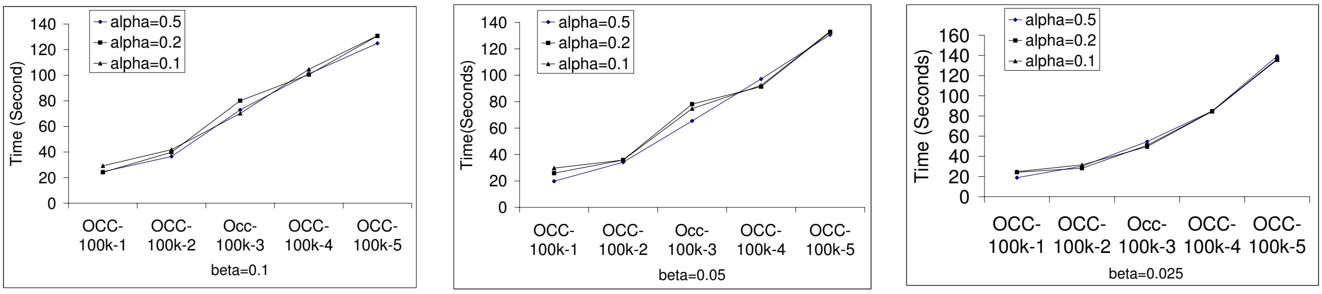
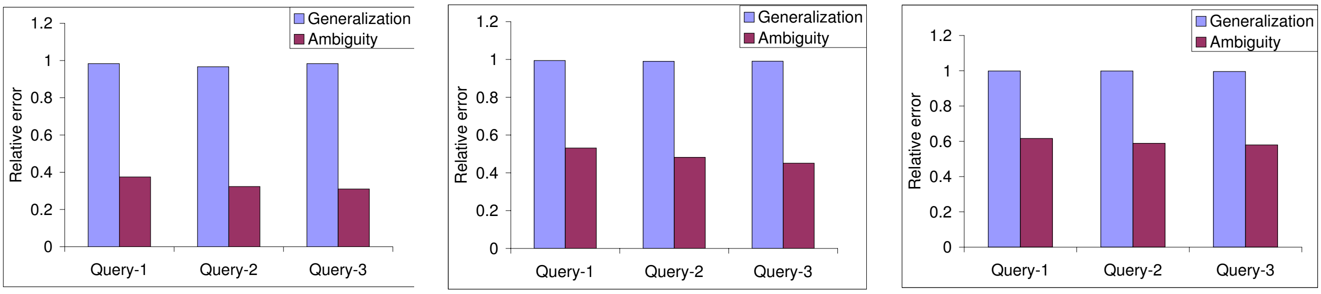
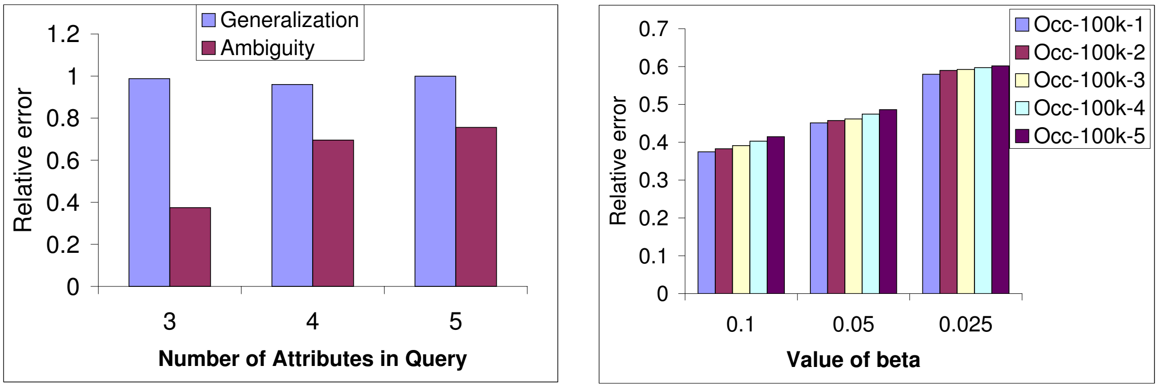


Figure 7: Performance of Ambiguity: various distributions, Occ dataset



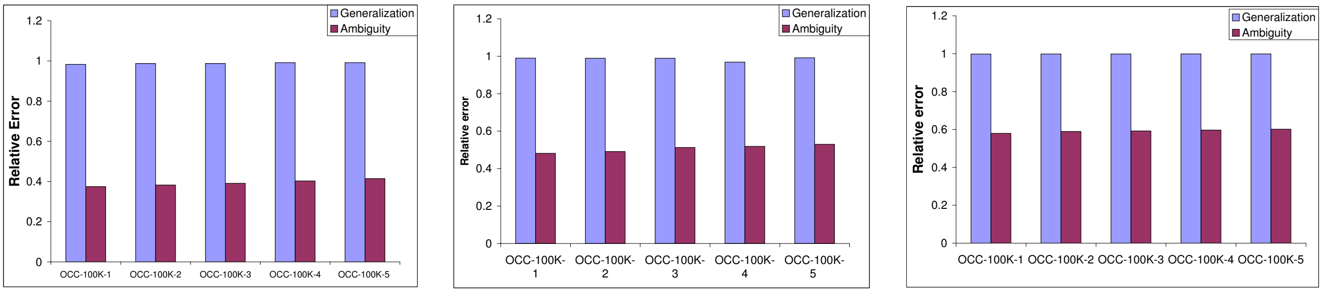
(1)β=0.1 (2)β=0.05 (3)β=0.025

Figure 8: Information Loss: Generalization V.S. Ambiguity, various β values



(1) Various # of attributes in queries (2) Various distributions

Figure 9: Information Loss



(1)β=0.1 (2)β=0.05 (3)β=0.025

Figure 10: Information Loss: Generalization V.S. Ambiguity, various Distributions

technique allows more accurate analysis of aggregate queries. Its information loss is always smaller than generalization.

9 Conclusion

When publishing private databases, two kinds of privacy leakage must be prevented. One is *presence leakage*, which is to identify an individual in (or not in) the microdata. Another is *association leakage*, which is to identify whether an individual is associated with some sensitive information, e.g., a specific disease. Association leakage is dependent on presence leakage. In this paper, we defined α-presence and β-association to address these two kinds of privacy leakage in a unified framework. We developed a novel technique, *ambiguity*, that protects both presence privacy and association privacy. We investigated the information loss of ambiguity and proved that ambiguity always yields better utility than generalization-based techniques. We elaborated our algorithm that efficiently constructs the ambiguity scheme that not only satisfies both α-presence and β-association but also produces small amounts of information loss.

One disadvantage of ambiguity is that it preserves limited amounts of correlations between data values. In the future, we plan to study how to improve the ambiguity technique so that correlations can be reasonably preserved while both α-presence and β-association are still satisfied.

References

[1] Charu C. Aggarwal, “On K-Anonymity and the Curse of Dimensionality”, VLDB 2005.

[2] Roberto J. Bayardo and Rakesh Agrawal, “Data privacy through optimal k-anonymization”, ICDE 2005.

[3] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, Nikos Mamoulis, “Fast Data Anonymization with Low Information Loss”, VLDB 2007.

[4] V. S. Iyengar, “Transforming Data to Satisfy Privacy Constraints”, In Proc. of SIGKDD, 2002.

[5] Kristen LeFevre, David DeWitt, and Raghu Ramakrishnan, “Incognito: Efficient Full-domain K-anonymity”, SIGMOD 2005.

[6] Kristen LeFevre, David DeWitt, and Raghu Ramakrishnan, “Mondrian Multidimensional K-Anonymity”, ICDE 2005.

[7] Ninghui Li, Tiancheng Li, “t-Closeness: Privacy Beyond K-anonymity and l-diversity”, ICDE 2007.

[8] Daniel Kifer, Johannes Gehrke, “Injecting Utility into Anonymized Datasets”, SIGMOD 2006.

[9] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. “l-Diversity: Privacy Beyond k-Anonymity”, ICDE 2006.

[10] David J. Martin, Daniel Kifer, Ashwin Machanavajjhala, Johannes Gehrke, Joseph Y. Halpern, “Worst-Case Background Knowledge for Privacy-Preserving Data Publishing”, ICDE 2007.

[11] Adam Meyerson, Ryan Williams, “On the Complexity of Optimal K-anonymity”, PODS, 2004.

[12] M. Ercan Nergiz, Maurizio Atzori, Christopher W. Clifton, “Hiding the Presence of Individuals from Shared Databases”, SIGMOD’07.

[13] Vibhor Rastogi, Dan Suciu, Sungho Hong, “The Boundary Between Privacy and Utility in Data Publishing”, VLDB 2007.

[14] Pierangela Samarati, Latanya Sweendy, “Generalizing Data to Provide Anonymity when Disclosing Information”, PODS, 1998.

[15] P. Samarati, “Protecting respondents’ identities in microdata release”, In TKDE, 2001.

[16] Latanya Sweeney, “k-anonymity: a model for protecting privacy”, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5):557570, 2002.

[17] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, Ke Wang, “(α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy-Preserving Data Publishing”, SIGKDD, 2006.

[18] Xiaokui Xiao, Yufei Tao, “Anatomy: Simple and Effective Privacy Preservation”, VLDB, 2006.

[19] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, Ada Wai-Chee Fu, “Utility-Based Anonymization Using Local Recoding”, SIGKDD, 2006.

[20] Ke Wang, Benjamin Fung, Philip Yu, ICDM, 2005. “Template-Based Privacy Preservation in Classification Problems”,

[21] Qing Zhang, Nick Koudas, Divesh Srivastava, Ting Yu: ”Aggregate Query Answering on Anonymized Tables”, ICDE 2007.