Hui Wang. Privacy-preserving data sharing in cloud computing. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 25(3): 401–414 May 2010

Privacy-Preserving Data Sharing in Cloud Computing

Hui Wang (王　慧)

Department of Computer Science, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ, U.S.A.

E-mail: hwang@cs.stevens.edu

Received May 25, 2009; revised March 4, 2010.

Abstract Storing and sharing databases in the cloud of computers raise serious concern of individual privacy. We consider 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 sharing techniques either fail to protect the presence privacy or incur considerable amounts of information loss. 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 *Ambiguity* against both presence leakage and association leakage. We prove both theoretically and empirically that the information loss of *Ambiguity* is always less than the classic generalization-based anonymization technique. We further propose an improved scheme, *PriView*, that can achieve better information loss than *Ambiguity*. We propose efficient algorithms to construct both *Ambiguity* and *PriView* schemes. Extensive experiments demonstrate the effectiveness and efficiency of both *Ambiguity* and *PriView* schemes.

Keywords privacy, data sharing, anonymity, utility, cloud computing

1 Introduction

Recent years have witnessed an emerged paradigm called cloud computing[1] which promises reliable services delivered through next-generation data centers that are built on computation and storage virtualization technologies. Data and programs are being swept up from desktop PCs and corporate server rooms and installed in the cloud of computers. With the aid of cloud computing, consumers no longer need to invest heavily or encounter difficulties in building and maintaining complex IT infrastructure. This is extremely useful for the users, for instance, some small/median-size enterprises, who have limited resources but computational-expensive tasks (e.g., data warehouse and data mining applications) for their data.

Although cloud computing offers the possibility of reliable storage of large volumes of data, efficient query processing, and savings of database administration cost for the data owner, sharing data with a third-party service provider and allowing it to take custody of personal documents raises questions about privacy protection. [2] has given an example scenario that a government agency presents a subpoena or search warrant to the cloud service provider that has possession of individual data. As the service provider is presumably less likely to contest the order, it will release the data without informing the data owners. This situation gets even worse as some service providers secretly sell their hosted data to make profit. As large amounts of data that are stored in the cloud contain personal information and are in non-aggregate format, sharing the data with third-party service providers in the cloud without careful consideration will raise great threat to data privacy.

In this paper, we consider two kinds of leakage of private information: *presence leakage*, by which an individual is identified to be in (or not in) the original dataset, and *association leakage*, by which an individual is identified to be associated with some sensitive information. As [3] 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.

We must note that the data privacy concern in the cloud computing is also shared in the traditional data publishing scenario; for analysis purpose, the data needs to be released in some format that is close to its raw value. Therefore, the privacy attacks in the data publishing scenario can also be applied to data sharing in cloud computing. One typical attack is called *record linkage attack*[4-5]. In particular, removing explicit identifiers such as name and SSN from the released data is insufficient to protect

Regular Paper

personal privacy; the combination of non-identification attributes, for instance, zipcode, gender and date of birth, still can uniquely identify individuals. These attributes are called quasi-identifier (QI) attributes. The QI-attributes are commonly present with individual names and SSNs in the external public datasets, for example, voting registration lists. Then by join of the shared dataset and the external public datasets, the attacker still can restore the identity of individuals as well as their private information.

	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)

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)

Fig.1. Examples of original and generalized dataset. (a) Original dataset. (b) 3-diversity table.

Various techniques have been proposed to defend against the record linkage attack in the context of privacy-preserving data publishing. One of the popular privacy principles is called k-anonymity[4-6]. Specifically, a table is said to satisfy k-anonymity if every record in the table is indistinguishable from at least $k-1$ other records with respect to quasi-identifier attributes. This ensures that no individual can be uniquely identified by record linkage attack. An improved principle called l-diversity, which also catches considerable attention recently, further requires that every group of indistinguishable records must contain at least l distinct sensitive values[7]. Fig.1(b) shows an example of a 3-diversity table.

Generalization[4-5] is a popular methodology to realize k-anonymity and l-diversity. In particular, the dataset is partitioned into groups (called QI-groups). For the records in the same group, their quasi-identifier values (QI-values) are replaced with the identical generalized values so that they are indistinguishable from each other with regard to their QI-values. The generalization-based anonymization technique can protect both presence privacy and association privacy; thus it can be used as a potential solution to privacy-preserving data sharing in the cloud.

However, generalization often results in considerable amount of information loss, which severely compromises the accuracy of data analysis. For example, consider the 3-diversity dataset in Fig.1(b). Without additional knowledge, the researcher will assume uniform data distribution in the generalized ranges. Let us look at the following aggregate query:

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

The query falls into the first QI-group in Fig.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 $1 \times \frac{(60-45)}{(60-20)} = \frac{3}{8}$, which is much less than the real answer 1. The error is caused by the fact that the data distribution in the generalized ranges may significantly deviate from uniformity as assumed. Therefore, generalization may circumvent correct understanding of data distribution on even a single attribute.

To address the defect of generalization, the permutation-based technique (e.g., anatomy[8], k-permutation[9]) are proposed recently. The basic idea is that instead of generalization of QI-values, both the exact QI-values and the sensitive values are published in two different tables. 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 the generalization-based technique, by publishing the exact QI-values, the permutation-based technique achieves 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, the permutation-based technique fails to protect presence privacy[10]. It arises the issue of trade-off between privacy and data utility: to achieve better utility, privacy has to be sacrificed to some extent. However, as in many applications, privacy always has higher priority than utility; users may accept data analysis result of reasonable amount of inaccuracy but cannot allow leakage of any private information. Therefore, it is important to design the anonymization technique that can guard both presence privacy and association privacy as the generalization-based technique 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 the permutation-based technique, *Ambiguity* publishes the exact QI-values so that it can provide better utility than the generalization-based technique. However, to protect presence privacy, instead of publishing QI-values together in a single table, *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 each QI-group. The QI-group membership is included in all auxiliary tables and the sensitive table. Fig.2 illustrates an example of the *Ambiguity* scheme of the dataset in Fig.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, he/she will have multiple candidates of QI-values due to the lossy join of the auxiliary tables. Out of these candidates, some are false match, i.e., they exist in the external database but not in the original dataset. For example, assume that 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 Fig.2. Since these values may all come either 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 dataset. 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 $4 \times 1 \times 3 = 12$ tuples from the join result of all auxiliary tables of 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 dataset. 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 original dataset is $Pr(\text{Alan} \in T) = \binom{11}{3}/\binom{12}{4} = 4/12 = 1/3$.

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 dataset, since the reasoning is dependent on the presence of Alan's record, the adversary has to calculate the probability $Pr((\text{Alan}, \text{diabetes}) \in T \mid \text{Alan} \in T) = \frac{Pr((\text{Alan},\text{diabetes}) \in T \cap \text{Alan} \in T)}{Pr(\text{Alan} \in T)}$. We have shown above that $Pr(\text{Alan} \in T) = 1/3$. Then to calculate $Pr((\text{Alan}, \text{diabetes}) \in T \cap \text{Alan} \in T)$, the adversary joins the auxiliary tables and the sensitive table on the first QI-group. The result contains $4 \times 1 \times 3 \times 4 = 48$ 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 knows 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 original dataset.

How can Ambiguity achieve less information loss than the generalization-based technique? 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 aforementioned query Q_1, it matches the first QI-group in Fig.2. There are four distinct ages, three of which (i.e., 45, 50 and 60) satisfy $Age \geqslant 45$. Thus the answer will be estimated as 3/4. Compared with the answer 3/8 from the generalized table (Fig.1(b)), the query result from the *Ambiguity* scheme is much closer to the real answer 1.

Age	*GID*
45	1
20	1
50	1
60	1
20	2
50	2
60	2

(a)

Gender	*GID*
M	1
F	2

(b)

Zipcode	*GID*
11000	1
12000	1
23000	1
54000	2
23000	2
21000	2

(c)

GID	*Disease*	*Count*
1	diabetes	1
1	flu	1
1	diarrhea	1
1	stroke	1
2	leukemia	2
2	diabetes	1
2	dyspepsia	1

(d)

Fig.2. An example of *Ambiguity* scheme. (a) Auxiliary table AT_1 on QI = *Age*. (b) Auxiliary table AT_2 on QI = *Gender*. (c) Auxiliary table AT_3 on QI = *Zipcode*. (d) Sensitive table ST.

Approaches	Presence Privacy	Association Privacy	Info. Loss
Generalization	Yes	Yes	Worst
Permutation	Yes	No	Best
Ambiguity (our work)	Yes	Yes	Between

Fig.3. Comparison of *Ambiguity* with other techniques.

A brief comparison of our *Ambiguity* technique to both generalization-based and permutation-based techniques is given in Fig.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 the generalization-based technique, it is an effective approach for privacy-preserving data sharing in the cloud.

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 can be 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 the generalization-based technique (Section 6).

Fourth, we develop an efficient algorithm to generate the *Ambiguity* scheme that provides sufficient protection to both presence privacy and association privacy. The algorithm is designed in a greedy fashion so that the amount of information loss is minimized (Section 7).

Fifth, we discuss *PriView*, an extension to *Ambiguity*. In particular, instead of splitting the original dataset into multiple view tables, with each containing a single QI-attribute, *PriView* splits the original dataset into only two view tables, each containing multiple QI-attributes. We analyze the privacy guarantee of *PriView*. Furthermore, we formally prove that *PriView* has better utility than *Ambiguity* (Section 8).

Finally, we use extensive experiments to prove both the efficiency and effectiveness of the *Ambiguity* and the *PriView* techniques (Section 9). Our experimental results demonstrate that both techniques always achieve better information loss than the generalization-based technique.

The rest of paper is organized as follows. Section 2 describes the related work. Section 3 introduces the background and defines the notions. Section 10 summarizes the paper.

2 Related Work

The most related work is in the area of privacy-preserving data publishing. Privacy-preserving data publishing has received considerable attention recently. There are considerable amounts of work on privacy models and anonymization techniques.

The k-anonymity model is one of the earlies privacy-preserving data publishing models. In a k-anonymous table, each record is indistinguishable from at least $k-1$ other records with respect to their quasi-identifier values. As the enhancement to k-anonymity, several privacy principles, for example, l-diversity[7], t-closeness[11] and (α, k)-anonymity[12], have been proposed to provide stronger privacy guarantee. The l-diversity model requires that every QI-group must contain at least l "well-represented" sensitive values. The t-closeness model requires that the distance between the distribution of anonymized dataset and that of the original database must be within t. And (α, k)-anonymity requires that: 1) every quasi-identifier *qid* is shared by at least k records, and 2) the confidence that *qid* is associated with the sensitive value s should be no larger than α, the given threshold. 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[3]. It defines the presence privacy as probabilities. In particular, given a released dataset 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 are 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.

There are several techniques that anonymize datasets to achieve the above privacy principles. Most of these techniques can be categorized into two types: generalization-based and permutation-based.

Generalization-Based Techniques. Generalization is a popular anonymization technique to realize the aforementioned privacy models. By generalization, the quasi-identifier values are replaced with less specific ones (e.g., replace specific age with a range of ages), so that after generalization, the original dataset is partitioned into groups, with each group consisting of at least k tuples that are of the same generalized quasi-identifier values[4,13-14].

Permutation-Based Techniques. Although the generalization-based technique can effectively protect privacy, it brings considerable amount of information loss. The situation gets even worse when the original dataset contains a large number of QI attributes: due to the curse of dimensionality, it becomes difficult to generalize the data without an unacceptably high amount of information loss[15]. To address this defect, a few permutation-based techniques (e.g., anatomy[8], k-permutation[9], bucketization[16]) are recently proposed to protect privacy without any data perturbation. Anatomy[8] releases all QI and sensitive values separately in two tables. By breaking the association between sensitive values and QI-values, it protects the association privacy. Both k-permutation[9] and bucketization[16] 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. All these permutation-based approaches release the exact QI-values, which enable them to achieve better utility than the generalization-based technique. However, releasing the exact QI-values in the same table enables the adversary to easily confirm that a particular individual is included in the original dataset. Therefore, the permutation-based approaches cannot provide enough protection to presence privacy[10]. We refer the readers to [17] for a detailed survey on privacy-preserving data publishing.

3 Background and Notions

Let T be a relational 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 categorized as identifier attributes $\mathcal{ID}$, sensitive attributes $\mathcal{S}$, and quasi-identifier attributes $\mathcal{QI}$. $\mathcal{ID}$ is used to uniquely identify the individuals. Typical $\mathcal{ID}$ attributes include people's name and social security number (SSN). For most of the cases, $\mathcal{ID}$ attributes are removed from the released dataset. Sensitive attributes $\mathcal{S}$ are the attributes, for example, `disease` or `salary`, whose values are considered as sensitive. In the next discussion, we assume there is only one sensitive attribute S. 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 non-sensitive non-ID attributes $\mathcal{QI}$ is called quasi-identifier (QI) attributes if these attributes can be linked with external datasets to uniquely identify an individual in the general population.*

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

Definition 3.2 (QI-Group). *Given a dataset 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 $i \neq j$, $G_i \cap G_j = \varnothing$.*

As an example, for the dataset in Fig.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 dataset 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 ($1 \leqslant i \leqslant m$) corresponds to a duplicate-free auxiliary table AT_i of schema (QI_i, GID). Furthermore, for any QI-group G_j ($1 \leqslant j \leqslant n$) 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 (GID, S, Count). Furthermore, for any QI-group G_j ($1 \leqslant j \leqslant n$) 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 $t[S] = s$.*

Fig.2 shows an *Ambiguity* scheme of the dataset in Fig.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 individual record t in the original dataset T.

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

Next, for association privacy, we define β-association as adversary's belief of the association between individuals and sensitive values. 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 dataset, we define the association privacy probability as conditionally dependent on the presence privacy probability. Specifically, we use $Pr((t,s) \in T \mid t \in T)$ 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 dataset T, let T^* be its released version. We say T^* satisfies β-association if for any sensitive association $(t,s) \in T^*$, $Pr((t,s) \in T \mid t \in T) \leqslant \beta$.*

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 dataset 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 dataset T and two privacy parameters α and β, our goal is to construct an (α,β)-private scheme T^* of T. Both α and β values are pre-defined by the data owners when they sanitize the dataset. We assume that the data owner uses the same value pairs of α and β for all individual records in his/her dataset. It is an interesting direction of future work to consider varying α and β values for different individual records.

5 Privacy of *Ambiguity*

In this section, we elaborate the details of quantifying both presence and association privacy of an *Ambiguity* scheme.

5.1 Measurement of Presence Privacy

To analyze the privacy guarantee of *Ambiguity* for both presence privacy and association privacy, we have to understand the attack first. By accessing the published ST and AT tables, the attacker can try to reason about the *possible* base tables that would yield the same tables as ST and AT by using the same view definitions. Note that hiding the view definitions from the attacker does not help, so we should consider the case where the view definitions are known to the attacker. We formalize this idea next, and identify each possible database as a possible world.

Definition 5.1 (Possible Worlds). *Given the original dataset T and the Ambiguity tables $T^* = \{AT_1,\ldots,AT_m,ST\}$, then the possible worlds PW of $T = \{T' \mid T'$ is a relation of the same schema as D, $\Pi_{\mathcal{QI},\mathcal{S}}(T') = \Pi_{\mathcal{QI}},\mathcal{S}(\bowtie_{i=1}^{m} AT_i \bowtie ST).\}$*

Fig.4 shows a subset of possible worlds of the *Ambiguity* tables in Fig.2. Out of all these possible worlds, only a subset contains the correct tuples as in the original dataset. This subset of possible worlds can help the attacker infer the existence of individuals and their private information. We call these worlds interesting worlds. The formal definition is as follows:

Definition 5.2 (Interesting Worlds). *Given the original dataset T and the Ambiguity tables $T^* = \{AT_1,\ldots,AT_m,ST\}$, let PW be the possible worlds constructed from T^*. The interesting worlds IW^t of an individual tuple $t \in T$ is defined as $IW^t = \{T' \mid T' \in PW$, and $t \in T'.\}$*

By the definition, only possible worlds 1 and 3 in Fig.4 are the interesting worlds of Alan (*Age* = 45, *Gender* = M, *Zipcode* = 11000). We assume that every possible world and interesting world are equally likely. Then for any individual tuple, we define its presence probability and association probability as follows:

Definition 5.3 (Presence Probability). *Given the original dataset T and the Ambiguity tables $T^* = \{AT_1,\ldots,AT_m,ST\}$, let PW be the possible worlds of T^*. For each individual QI-value $t \in T$, let IW^t be the interesting worlds of t, then the presence probability $Pr(t \in T) = |IW^t|/|PW|$.*

We next discuss how to infer $|IW^t|$ and $|PW|$, the size of possible worlds and interesting worlds. Intuitively the adversary will infer the presence of a tuple in the original dataset 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.

Age	*Gen.*	*Zip*	*Disease*
45	M	11000	diabetes
20	M	12000	flu
50	M	23000	diarrhea
60	M	12000	stroke
20	F	54000	leukemia
50	F	23000	diabetes
60	F	23000	leukemia
60	F	21000	dyspepsia

(a)

Age	*Gen.*	*Zip*	*Disease*
45	M	23000	diarrhea
20	M	11000	diabetes
50	M	12000	flu
60	M	12000	stroke
20	F	23000	diabetes
50	F	54000	leukemia
60	F	23000	leukemia
60	F	21000	dyspepsia

(b)

Age	*Gen.*	*Zip*	*Disease*
45	M	11000	diabetes
20	M	12000	flu
50	M	12000	stroke
60	M	23000	diarrhea
20	F	21000	dyspepsia
50	F	23000	diabetes
60	F	23000	leukemia
60	F	54000	leukemia

(c)

Fig.4. Example of possible worlds. (a) Possible world 1. (b) Possible world 2. (c) Possible world 3.

Definition 5.4 (Cover). *Given a dataset T and an Ambiguity table T^* $(AT_1, \ldots, AT_m, ST)$, we say a tuple $t \in T$ is* covered *by T^* if $t^{QI} \in \bowtie_{i=1}^{m} AT_i$, where $\bowtie$ is an equal-join operator.*

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 at least one auxiliary *Ambiguity* table. We use $t[QI_i]$ to indicate the i-th QI-value of the tuple t.

Lemma 1 (Cover). *Given a dataset T and an Ambiguity table T^* $(AT_1, \ldots, AT_m, ST)$, let AT_i $(1 \leqslant i \leqslant m)$ 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 the QI-group G_j.*

For example, Alice's record is covered by the second QI-group in the *Ambiguity* scheme in Fig.2. In the join result of the auxiliary tables in Fig.2 of group ID 2 (i.e., the group that covers Alice's record), there are $3 \times 1 \times 3 = 9$ combinations of QI-attributes, some of them are false match and do not exist in the original dataset. Furthermore, the frequency count 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 (i.e., the possible worlds), out of which $\binom{8}{3}$ choices contain Alice's record (i.e., the interesting worlds). Thus the probability that Alice's record exists in the original dataset is $\binom{8}{3}/\binom{9}{4} = 4/9$. We use $|G|$ to denote the number of tuples in QI-group G, and $|G_{AT_i}|$ as the number of tuples of QI-group G in the auxiliary table AT_i. Then in general:

Theorem 1 (Presence Probability). *Given a dataset T and an Ambiguity table $T^*(AT_1, \ldots, AT_m, ST)$, 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 $Pr(t \in T) = |G|/(\Pi_{i=1}^{m}|G_{AT_i}|)$.*

Proof. Given the tuple t, the number of possible worlds $|PW|$ equals $\binom{(\Pi_{i=1}^{m}|G_{AT_i}|)}{|G|}$. Out of these possible worlds, the number of interesting worlds of t, $|IW^t|$ equals $\binom{(\Pi_{i=1}^{m}|G_{AT_i}|-1)}{(|G|-1)}$. Thus $Pr(t \in T) = |IW^t|/|PW| = |G|/(\Pi_{i=1}^{m}|G_{AT_i}|)$. □

Theorem 1 shows that the presence probability can be improved by increasing $|G_{AT_i}|$, the size of QI-group in AT_i, and/or reducing $|G|$, 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 $Pr((t, s) \in T \mid t \in T)$. 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 discussed how to measure $Pr(t \in T)$ in Subsection 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 $(QI_1, \ldots, QI_m, S, GID)$, where QI_i is the i-th QI-attribute $(1 \leqslant i \leqslant m)$, and S is the sensitive attribute. 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 Fig.2, the adversary knows that if Alice's record is present in the original dataset, 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 $3 \times 1 \times 3 \times (2+1+1) = 36$ tuples. Second, the frequency count 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 as the possible worlds. If the adversary assumes Alice's record is present in the original dataset and he/she is interested with the association `(Alice, leukemia)`, since the frequency count of leukemia is 2, there will be $\binom{2}{1} \times \binom{35}{3}$ choices that contain `(Alice, leukemia)`. Without additional knowledge, the probability $Pr((\text{Alice}, \text{leukemia}) \in T \cap (\text{Alice} \in T))$ is $\binom{2}{1} \times \binom{35}{3}/\binom{36}{4}$. This is formally explained in the next lemma. Again, we use $|G|$ to denote the number of tuples in QI-group G, and $|G_{AT_i}|$ as the number of tuples of QI-group G in the auxiliary table AT_i.

Lemma 2. *Given a dataset T and an Ambiguity scheme $T^*(AT_1, \ldots, AT_m, ST)$, 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 frequency count of the sensitive value s in G. Then the probability $Pr((t, s) \in T \cap t \in T) = c/(\Pi_{i=1}^{m}|G_{AT_i}|)$.*

Proof. Given the tuple (t, s), the number of possible worlds $|PW|$ equals $\binom{(\Pi_{i=1}^{m}|G_{AT_i}| \times |G|)}{|G|}$. Out of these possible worlds, the number of interesting worlds of (t, s) $|IW^{(t,s)}|$ equals $\binom{\binom{c}{1} \times (|G| \times \Pi_{i=1}^{m}|G_{AT_i}|-1)}{(|G|-1)}$. Thus $Pr(t \in T) = |IW^t|/|PW| = c \times |G|/(\Pi_{i=1}^{m}|G_{AT_i}| \times |G|) = c/(\Pi_{i=1}^{m}|G_{AT_i}|)$. □

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

Theorem 2. (Association Privacy). *Given a dataset T and an Ambiguity scheme T^* $(AT_1, \ldots, AT_m, ST)$, for any tuple $t \in T^*$, let G be its covered QI-group. Then the association privacy $Pr((t, s) \in T \mid t \in T) = c/|G|$, where c is the frequency count of the sensitive value s in G.*

Proof. Lemma 2 has shown that $Pr((t, s) \in T \cap t \in T) = c/(\Pi_{i=1}^{m}|G_{AT_i}|)$, and Theorem 1 has proven that

$Pr(t \in T) = |G|/(\Pi_{i=1}^{m}|G_{AT_i}|)$. Thus $Pr((t,s) \in T \mid t \in T) = \frac{Pr((t,s) \in T \cap t \in T)}{Pr(t \in T)} = c/|G|$. □

Theorem 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 Fig.2, $Pr((\text{Alice}, \text{leukemia}) \in T \mid \text{Alice} \in T) = 2/4 = 1/2$.

6 Information Loss of *Ambiguity*

In this paper, as the same as in [8, 18], we consider the error 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 dataset 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 $(AT_1, \ldots, AT_m, ST)$, for any count query $Q = \text{count}(\sigma_C(AT_1 \bowtie \cdots \bowtie AT_m \bowtie ST))$, 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 $(AT_1, \ldots, AT_m, ST)$, and a counting query Q with the selection condition C, we use C_i $(1 \leqslant i \leqslant m)$ 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* $\geqslant$ 55 and *Disease* = stroke" on the *Ambiguity* scheme in Fig.2, C_1 (on AT_1) = "*Age* $\geqslant$ 55", and C_S (on ST) = "*Disease* = stroke".

The pseudo code in Fig.5 shows the details of how to approximate the result of counting queries. First, we locate all the QI-groups that satisfy C_S. Second, for 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. Then for every selection condition C_i on the AT table AT_i $(1 \leqslant i \leqslant m)$, 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. Last, we sum up the adjusted counts for all QI-groups.

```
Input:  Ambiguity tables (AT_1, ..., AT_m, ST), query Q;
Output:  the estimated answer of Q.
GID ← Π_GID σ_{C_S}(ST);
n ← 0, i ← 1;
For i ⩽ m
    l ← 0, k ← 0;
    For Group ID j ∈ GID
        c ← count(σ_{(C_S, GID=j)} ST);
        l ← count(σ_{(C_i, GID=j)} AT_i);
        k ← count(σ_{(GID=j)} AT_i);
        n ← n + c × l/k
    i ← i + 1;
Return n.
```

Fig.5. Algorithm: estimation of answers of counting queries.

Note that the generalization-based technique uses 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 is given in Section 1.

We explain how to use the algorithm in Fig.5 to estimate the results of count queries by using the *Ambiguity* scheme in Fig.2. For query Q_2:

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

Both QI-groups 1 and 2 satisfy the condition *Disease* = diabetes on ST. For QI-group 1, the count is estimated as $1 \times \frac{2}{4} \times \frac{1}{3} = \frac{1}{6}$, where $\frac{2}{4}$ corresponds to 2 ages (out of 4) that satisfy *Age* $\geqslant$ 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 $1 \times \frac{2}{3} \times \frac{1}{3} = \frac{2}{9}$. 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 the generalization-based approach. We have:

Theorem 3 (Information Loss: *Ambiguity* vs. Generalization). *Given a dataset T, let T_G be the table of T anonymized by generalization. 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 $(1 \leqslant i \leqslant m)$, and for each QI-group G_j in AT_i, let k_{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 $(n_j \times l_{ij})/k_{ij}$. 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 $n_j \times l_{ij}/r_{ij}$. 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 $r_{ij} \geqslant k_{ij}$. Therefore for every QI-group in each *Ambiguity* auxiliary table, its estimated result is larger than that by generalization. It follows that $Q(T_A) \geqslant Q(T_G)$. Consequently the relative error by *Ambiguity* is always less than that by generalization approaches. □

We also have experimental results to prove that our *Ambiguity* approach always wins generalization-based approaches with regard to information loss. More details can be found in Subsection 9.3.

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 dataset T into multiple non-overlapping QI-groups, each of which meets α-presence (by Theorem 1) and β-association (by Theorem 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.

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. Each hashed value corresponds to a bucket. We require that for n distinct values, there exists n hashed buckets, so that different values will not 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. 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 multiple QI-groups, and thus 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 (α, β)-privacy QI-groups with smaller sizes. Consequently the amount of information loss is reduced. Fig.6 shows the bucketized result of Fig.1. The integer numbers on the right side of → indicate the bucket IDs.

60 → 4, 7, 8		23000 → 3, 6, 7	Diabetes → 1, 6
50 → 3, 6	M → 1,2, 3, 4	11000 → 1	Leukemia → 5, 7
20 → 2, 5	F → 5, 6, 7, 8	12000 → 2, 4	Flu → 2
45 → 1		59000 → 3	Diarrhea → 3
		54000 → 5	Stroke → 4
Age	*Gender*	21000 → 8	Dyspepsia → 8
		Zipcode	*Disease*

Fig.6. Bucketization.

Based on the bucketization result, we can compute the presence probability as follows: for QI-group G, let k_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 1, the presence probability equals $n/\Pi_{i=1}^{m} k_i$, where m is the number of QI-attributes. For example, the QI-group in Fig.2 that contains both tuples 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 $1/(2 \times 1 \times 2) = 1/4$. The pseudo code in Fig.8 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.

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. The pseudo code in Fig.7 shows more details.

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,

```
max ← 100000; picked ← null;
For all unpicked tuple t ∈ T {
    m ← No. of hash buckets in HS that G ∪ {t} covers;
    If m < max
        max ← m; picked ← t;}
Return picked.
```

Fig.7. *pick*(G, *HS*): pick a tuple that will cover the max. number of buckets with the tuples in $G \cup \{t\}$ by using hash buckets *HS*.

we pick $\lceil 1/\beta \rceil$ tuples from $\lceil (1/\beta) \rceil$ buckets on the sensitive values, a tuple from a bucket. We pick the tuples by calling *pick*() function (Fig.7), 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. 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. We have:

Theorem 4. *Given a dataset T, let T^* be the Ambiguity scheme that is constructed by Ambiguity algorithm. Then T^* is (α, β)-private.*

Proof. Since the construction of QI-groups terminates only when the α-presence is satisfied, and adding residue tuples is also aware of α-presence requirement, 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 5. (*Ambiguity* vs. l-Diversity). *Given a dataset T, let T^* be the Ambiguity scheme that is constructed by our algorithm. Then T^* satisfies $\lceil 1/\beta \rceil$-diversity.*

Proof. In our *Ambiguity* algorithm, 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 Extension: *PriView*

As shown in Section 6, the information loss by *Ambiguity* is always less than that by the generalization-based anonymization technique. However, due to lossy join of multiple released auxiliary tables and the sensitive table, the information loss may still be high. In this section, we discuss *PriView*, an extension to *Ambiguity*, that incurs smaller information loss. In particular, instead of publishing multiple auxiliary tables, each containing a single QI-attribute, we release only two view tables, each containing multiple QI-attributes. Fig.9 shows an example of *PriView* tables of the original dataset shown in Fig.1(a). Formally:

Definition 8.1 (*PriView*). *Given a dataset T that consists of m QI-attributes $\mathcal{QI}$ and the sensitive attribute $\mathcal{S}$, PriView includes an auxiliary table (AT) and a sensitive table (ST). In particular*:

1) *the auxiliary table AT is of schema (QI, GID), where $QI \subset \mathcal{QI}$,*

2) *the sensitive table ST is of schema $(GID, QI', \mathcal{S}, Count)$, where (i) $QI' \cup QI = \mathcal{QI}$, and (ii) $QI' \cap QI = \varnothing$.*

8.1 Privacy Analysis

Similar to *Ambiguity*, *PriView* protects both presence and association privacy by lossy join of the AT

$k_i \leftarrow$ `No. buckets that` G `covers for` H_{QI_i};
$n \leftarrow$ `No. buckets that` G `covers for` H_S;
`Return` $n/\Pi_{i=1}^{m} k_i$.

Fig.8. *CalPPro*(G): calculation of presence probability of QI-group G.

Age	*Gender*	*GID*
45	M	1
20	M	1
50	M	1
60	M	1
20	F	2
50	F	2
60	F	2
60	F	2

(a)

GID	*Zipcode*	*Disease*	*Count*
1	11000	diabetes	1
1	12000	flu	1
1	23000	diarrhea	1
1	12000	stroke	1
2	54000	leukemia	1
2	23000	diabetes	1
2	23000	leukemia	1
2	21000	dyspepsia	1

(b)

Fig.9. Example of *PriView* tables. (a) Auxiliary table AT. (b) Sensitive table ST.

and ST tables. Then by the similar reasoning as in Theorem 1 and Theorem 2, we have:

Theorem 6 (Presence and Association Probabilities). *Given the original dataset T and the PriView tables $\{ST, AT\}$, for each individual tuple $t \in T$, let G be the QI-group that covers t. Then the presence probability $Pr(t \in T) = 1/|G|$, and $Pr((t,s) \in T|t \in T) = c/|G|$, where c is the frequency count of the sensitive value s in G.*

Proof. First, we explain the details of how to infer $Pr(t \in T)$. It is straightforward that the total number of possible worlds constructed from QI-group G equals $\binom{|G_{AT}|\times|G_{ST}|}{|G|}$, where $|G_{AT}|$ and $|G_{ST}|$ are the sizes of the QI-group G in AT and ST tables. Out of these possible worlds, the total number of interesting worlds of QI-value t equals $\binom{|G_{AT}|\times|G_{ST}|-1}{|G|-1}$. Thus $Pr(t \in T) = |G|/(|G_{AT}| \times |G_{ST}|)$. Since $|G_{AT}| = |G|$, $Pr(t \in T) = 1/|G|$. Similarly, for the association probability $Pr((t,s) \in T \,|\, t \in T)$, the total number of possible worlds equals $\binom{c\times|G_{AT}|\times|G_{ST}|}{|G|}$, and the total number of interesting worlds of QI-value t equals $\binom{c\times|G_{AT}|\times|G_{ST}|-1}{|G|-1}$, thus $Pr((t,s) \in T) = c/|G|$. □

8.2 Information Loss of *PriView*

As in *Ambiguity*, we still consider the accuracy of count queries as the utility for *PriView*. The only difference between *Ambiguity* and *PriView* is that *PriView* only considers the join of two tables, while *Ambiguity* may consider more than two tables. Thus we adapt Fig.5 in Section 6 to *PriView* by changing the input to two tables $\{AT, ST\}$. And we have:

Theorem 7. (Information Loss: *PriView* vs. *Ambiguity*). *Given a dataset T, let T_A be the released Ambiguity tables of T. Then there always exists a PriView scheme T_P such that for any count query Q, the relative error of answering Q by using T_P is no more than that by T_A.*

Proof. Given the *Ambiguity* table T_A, we construct the *PriView* scheme T_P by following: first, we pick an auxiliary table in T_A to join with the sensitive table in T_A. Let the join result be ST. Second, we join the rest of unpicked auxiliary tables in T_A and let the join result be AT. Then $\{AT, ST\}$ is the *PriView* scheme T_P that we are looking for. Compared with *Ambiguity*, to evaluate count queries on join of tables, *PriView* only needs one lossy join, while *Ambiguity* contains $m-1 \geqslant 1$ lossy joins, where m is the number of QI-attributes. Thus *PriView* always produces smaller information loss than *Ambiguity*. □

Our experimental results also show that *PriView* always incur much less information loss than *Ambiguity*, More details can be found in Section 9.

8.3 Algorithm

Intuitively, given m QI-attributes, *PriView* has $2^m - 2$ possible schemes. However, due to the fact that the more QI-attributes being put into the same table, the less the information loss incurred by lossy join, we do not need to consider all possible schemes. Thus we only need to consider the schemes in which the AT table contains $m-1$ QI-attributes, while ST contains the remaining one QI-attribute. In other words, we only have m possible *PriView* schemes to consider. This optimization dramatically reduces the search space. Out of these m schemes, we pick the one that potentially returns the smallest information loss. To achieve this goal, we follow the same principle as *Ambiguity* algorithm: we construct the QI-groups that of sizes as small as possible. Thus we adapt the *Ambiguity* algorithm (Section 7) to *PriView*. In particular, instead of bucketizing on each individual QI-attribute QI_i, we bucketize on $(QI_i, \mathcal{S})$, where $\mathcal{S}$ is the set of sensitive attributes.

9 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.

9 Experimental Setup

Setup. We implement the *Ambiguity* algorithm in C++. We use a workstation running Linux RedHat version 2.6.5 with 1 processor of speed 2.8GHz and 2GB RAM. We use the multi-dimension k-anonymity generalization algorithm implemented by Xiao *et al.*[8]①.

Dataset. We use the *Census* dataset that contains personal information of 500 000 American adults②. The details of the dataset are summarized in Fig.10.

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

Fig.10. Summary of attributes.

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

②http://www.ipums.org/

From the *Census* dataset, we create two datasets, *Occ* and *Sal*, with the *Occ* set using *Occupancy* as sensitive attribute and *Sal* using *Salary*. For each set, we randomly pick 100 K, 200 K, 300 K, 400 K and 500 K tuples from the full set and construct tables as *Occ*-n and *Sal*-n (n = 100 K, 200 K, 300 K, 400 K, 500 K).

To study the impact of distributions to anonymization, we also generate a set of files with various distributions. We construct 10 datasets *Occ*-100 K-d and *Sal*-100 K-d ($1 \leqslant d \leqslant 5$), each of 100 K 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, the denser the dataset.

Queries. We consider the count queries of the form.

```
SELECT QT_1, ..., QT_i, count(*)
FROM data
WHERE S = v
GROUP By QT_1, ..., 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 $Query - i$ ($1 \leqslant i \leqslant 3$), where $i = 1, 2$ and 3 correspond to the query selectivity of 1%, 5% and 10%.

9.2 Performance of Generating *Ambiguity* Scheme

First, we vary the values of α and β for the α-presence and β-association requirements. Fig.11 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 $t_c \times m$, where t_c is the time complexity of function *CalPPro*() (Fig.8) and m is the number of QI-groups. Smaller QI-groups will result in smaller t_c but larger m, i.e., $t_c \times m$ 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 examine 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. Fig.12 shows the result of *Occ* datasets. We observe that the sparser the dataset is, the better the 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 be added into QI-groups. The same phenomenon also hold for *Sal* datasets.

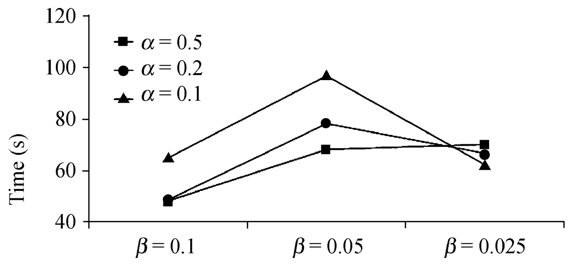


Fig.11. Performance: various α and β values.

9.3 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 $(|act| - |est|)/|act|$, where *act* is its actual result derived from the dataset, and *est* the estimate computed from the *Ambiguity*, *PriView*, and generalized table. The details of measurement of $|est|$ for both generalized tables and *Ambiguity* technique are explained in Section 6. The answer estimation measurement on the *PriView* scheme is the same as that of *Ambiguity*.

The first set of this part of experiments compares the accuracy of query results of *PriView*, *Ambiguity* technique and generalization technique regarding different query configurations. Fig.13(a) shows the comparison result for queries of different selectivity. We observe that the information loss decreases when the queries are more selective. We also measure the accuracy of queries involving 3, 4 and 5 attributes in the selection conditions. The attributes are chosen randomly. The

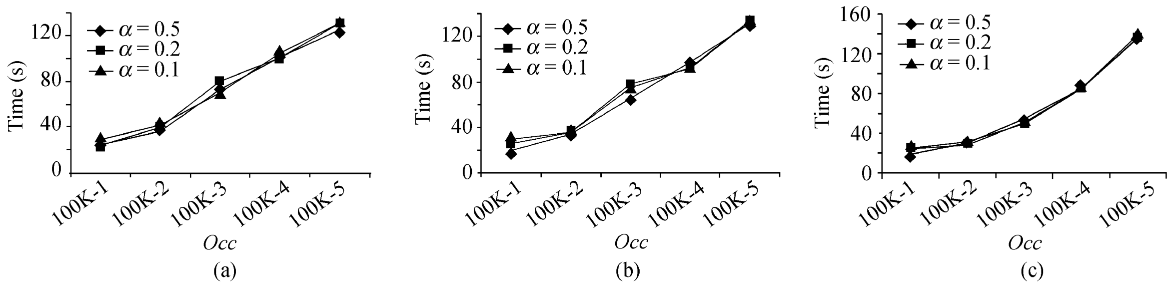


Fig.12. Performance of *Ambiguity*: various distributions, *Occ* dataset. (a) $\beta = 0.1$. (b) $\beta = 0.05$. (c) $\beta = 0.025$.

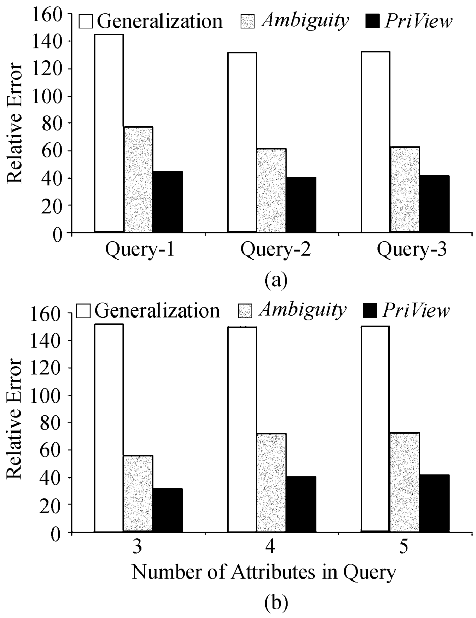


Fig.13. Information loss. (a) Different queries. (b) Various numbers of attributes in queries.

results are shown in Fig.13(b). It is not surprising that the information loss will increase when there are more attributes in queries. For both sets of experiments, as expected, *PriView* always wins the other two approaches, while *Ambiguity* always produces better accuracy of query results than the generalization-based approach.

Second, we measure the impact of β values to information loss. Fig.14 shows that when β increases, all three approaches have decreasing information loss. This is because larger β results in smaller QI-groups, which decreases the size of QI-groups for both *Ambiguity* and *PriView*, and the size of generalized ranges for generalization-based approaches. However, since the generalized ranges always grow faster than the number of distinct tuples, our *Ambiguity* and *PriView* techniques have much better accuracy than the generalization-based approach.

We also measure the impact of the data distribution to the information loss of *Ambiguity* and *PriView*. Fig.15(a) shows that for *Ambiguity*, 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. The same results also hold for *PriView* (Fig.15(b)).

As a brief summary, we showed that our *Ambiguity* technique allows more accurate analysis of aggregate queries. Its information loss is always smaller than generalization. Moreover, the extension *PriView* incurs smaller information loss than *Ambiguity*.

10 Conclusion

Storing private databases in the cloud of computers and sharing them with third-party service providers

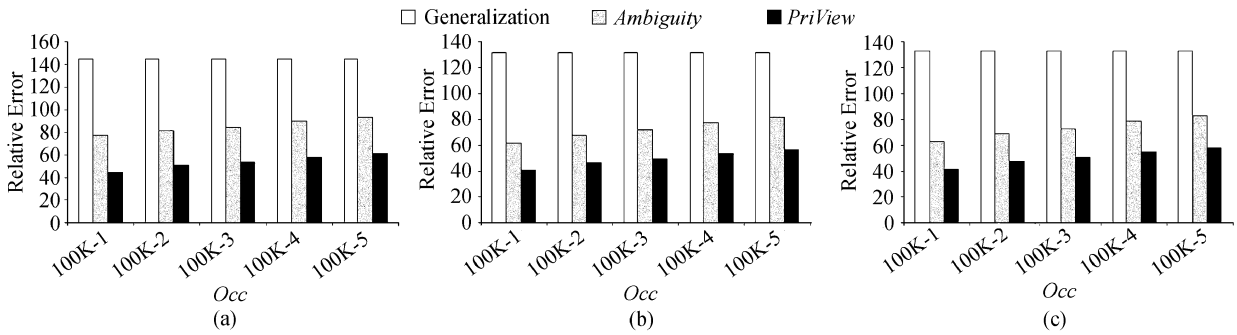


Fig.14. Information loss; various β values. (a) $\beta = 0.025$. (b) $\beta = 0.05$. (c) $\beta = 0.1$.

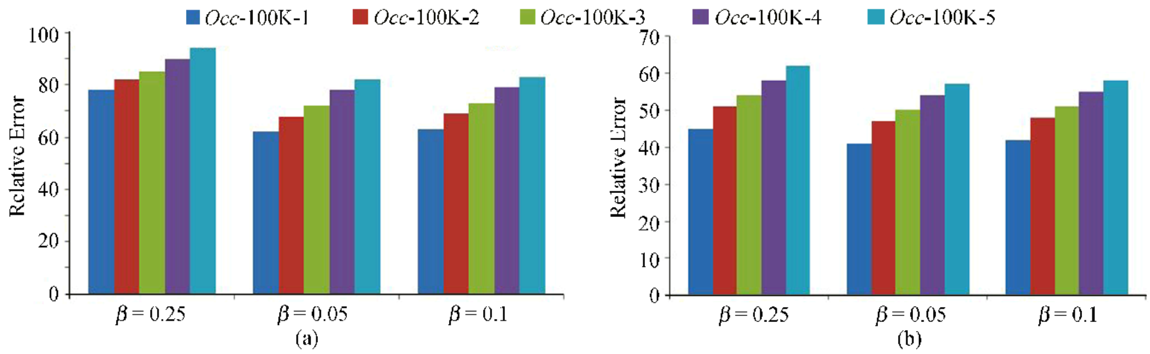


Fig.15. Information loss; various datasets. (a) *Ambiguity*. (b) *PriView*.

raise serious concern of privacy. We considered two kinds of privacy leakage, presence leakage, which is to identify an individual in (or not in) the dataset, and association leakage, which is to identify whether an individual is associated with some sensitive information, e.g., a specific disease. 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 the generalization-based technique. 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. We also proposed *PriView* that better preserves the correlations between data values than *Ambiguity*. In the future, we plan to adapt both *Ambiguity* and *PriView* to the dynamic datasets.

References

[1] Weiss A. Computing in the clouds. *NetWorker*, Dec. 2007, 11(4): 16-25.

[2] Hayes B. Cloud computing. *Communications of the ACM*, 2008, 51(7): 9-11.

[3] Nergiz M E, Atzori M, Clifton C W. Hiding the presence of individuals from shared databases. In *Proc. ACM's Special Interest Group on Management of Data* (*SIGMOD 2007*), Beijing, China, June 11-17, 2007, pp.665-676.

[4] Samarati P, Sweeney L. Generalizing data to provide anonymity when disclosing information. In *Proc. ACM International Conference on Principles of Database Systems* (*PODS*), Seattle, USA, June 1-4, 1998, p.188.

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

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

[7] Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M. l-diversity: Privacy beyond k-anonymity. In *Proc. International Conference on Data Engineering Conference* (*ICDE*), Atlanta, USA, Apr. 2006, p.24.

[8] Xiao X, Tao Y. Anatomy: Simple and effective privacy preservation. In *Proc. Very Large Data Base Conference* (*VLDB*), Seoul, Korea, Sept. 12-15, 2006, pp.139-150.

[9] Zhang Q, Koudas N, Srivastava D, Yu T. Aggregate query answering on anonymized tables. In *Proc. International Conference on Data Engineering Conference* (*ICDE*), Istanbul, Turkey, Apr. 15-20, 2007, pp.116-125.

[10] Ghinita G, Karras P, Kalnis P, Mamoulis N. Fast data anonymization with low information loss. In *Proc. Very Large Data Base Conference* (*VLDB*), Vienna, Austria, Sept. 23-27, 2007, pp.758-769.

[11] Li N, Li T. t-Closeness: Privacy beyond K-anonymity and l-diversity. *In Proc. International Conference on Data Engineering Conference* (*ICDE*), Istanbul, Turkey, April 15-20, 2007, pp.106-115.

[12] Wong R C W, Li J, Fu A W C, Wang K. (α, k)-anonymity: An enhanced k-anonymity model for privacy-preserving data publishing. In *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (*SIGKDD*), Philadelphia, USA, Aug. 20-23, 2006, pp.754-759.

[13] Bayardo R J, Agrawal R. Data privacy through optimal k-anonymization. In *Proc. International Conference on Data Engineering Conference* (*ICDE*), Tokyo, Japan, Apr. 5-8, 2005, pp.217-228.

[14] Meyerson A, Williams R. On the complexity of optimal k-anonymity. In *Proc. ACM International Conference on Principles of Database Systems* (*PODS*), Paris, France, June 14-16, 2004, pp.223-228.

[15] Aggarwal C C. On k-anonymity and the curse of dimensionality. In *Proc. Very Large Data Base Conference* (*VLDB*), Trondheim, Norway, June 14-16, 2005, pp.901-909.

[16] Martin D J, Kifer D, Machanavajjhala A, Gehrke J, Halpern J Y. Worst-case background knowledge for privacy-preserving data publishing. In *Proc. International Conference on Data Engineering Conference* (*ICDE*), Istanbul, Turkey, April 15-20, 2007, pp.126-135.

[17] Fung B C M, Wang K, Chen R, Yu P S. Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys* (*CSUR*), December 2010, 42(4). (to appear).

[18] Rastogi V, Suciu D, Hong S. The boundary between privacy and utility in data publishing. In *Proc. Very Large Data Base Conference* (*VLDB*), Vienna, Austria, Sept. 23-27, 2007, pp.531-542.

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

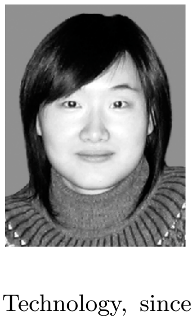
[20] Iyengar V S. Transforming data to satisfy privacy constraints. In *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (*SIGKDD*), Edmonton, Canada, July 23-26, 2002, pp.279-288.

[21] Xu J, Wang W, Pei J, Wang X, Shi B, Fu A W C. Utility-based anonymization using local recoding. In *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (*SIGKDD*), Philadelphia, USA, Aug. 20-23, 2006, pp.785-790.

[22] Kifer D, Gehrke J. Injecting utility into anonymized datasets. In *Proc. ACM's Special Interest Group on Management Of Data* (*SIGMOD*), Chicago, USA, June 27-29, 2006, pp.217-228.

[23] LeFevre K, DeWitt D, Ramakrishnan R. Incognito: Efficient full-domain k-anonymity. In *Proc. ACM's Special Interest Group on Management Of Data* (*SIGMOD*), Baltimore, USA, June 14-16, 2005, pp.49-60.

[24] LeFevre K, DeWitt D, Ramakrishnan R. Mondrian multidimensional k-anonymity. In *Proc. International Conference on Data Engineering Conference* (*ICDE*), Tokyo, Japan, Apr. 5-8, 2005, p.25.



Hui Wang received the B.S. degree in computer science from Wuhan University in 1998, the M.S. degree in computer science from University of British Columbia in 2002, and the Ph.D. degree in computer science from University of British Columbia in 2007. She has been an assistant professor in the Computer Science Department, Stevens Institute of Technology, since 2008. Her research interests include data management, database security, data privacy, and semi-structured databases.