NON-CENTRALIZED DISTINCT L-DIVERSITY

Chi Hong Cheong[1], Dan Wu[2], and Man Hon Wong[3]

[1,3]Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Hong Kong
{chcheong, mhwong}@cse.cuhk.edu.hk
[2]Network and Telecommunication Division
Guangdong Electric Power Design Institute, China
wudan@gedi.com.cn

ABSTRACT

This paper considers the non-centralized version of privacy preserving data publishing (PPDP), which refers to generating published tables from multiple non-centralized private tables owned by different data holders. Traditional solutions to PPDP on a single centralized dataset cannot be directly applied to this problem. Even if every published table satisfies a traditional privacy preserving requirement individually, an adversary who can collect multiple published tables may be able to deduce some private information that violates the satisfied requirement. Due to privacy reasons, the data holders cannot share information with each other to cooperate on the data publishing issues. In this paper, we propose non-centralized distinct l-diversity and an algorithm to generate published tables. Our algorithm does not rely on any communications between the data holders but only collects published tables released by other data holders. Experiments on real datasets are conducted to show that the algorithm is feasible to real applications.

KEYWORDS

Non-centralized, Privacy Preserving Data Publishing, Database, Conditional Independence

1. INTRODUCTION

Traditional solutions to the problem of privacy preserving data publishing (PPDP), such as *k*-anonymity [1], *l*-diversity [2], and *t*-closeness [3], only guarantee that privacy is preserved if a published table is created from a centralized dataset. Recently, many researchers have shown that if there are multiple published tables created from a centralized dataset (which can be either dynamic or static), privacy may not be preserved even if every published table satisfies a certain privacy preserving requirement individually [4][5][6][7][8]. This problem is called serial data publishing (on a centralized dataset). In this paper, we consider a similar but different problem: data publishing on non-centralized datasets. The non-centralized datasets that we refer to are multiple datasets that have a common sensitive attribute with the same domain and store information about the same group of individuals, but they are owned by different data holders that cannot share information with each other due to privacy reasons. We are going to show that privacy may not be preserved even if all published tables generated from such multiple non-centralized datasets satisfy certain privacy preserving requirements for data publishing individually. It is illustrated in the following example.

Table 1 shows two private tables, namely $PriT_R$ and $PriT_C$, which are solely owned by the Revenue Department and the Censor Department respectively. Both $PriT_R$ and $PriT_C$ store income information of the same group of individuals. Some data miners may find such information useful for their manpower researches. However, due to privacy reasons or even enforced by law, private tables $PriT_R$ and $PriT_C$ cannot be released directly to any other parties.

DOI: 10.5121/ijdms.2012.4201

Instead, the Revenue Department and the Censor Department release two published tables, denoted as T_{age} and $T_{ZIP\ code}$, which are created by generalizing or suppressing the attributes of $PriT_R$ and $PriT_C$ respectively. An example is shown in Table 2, where T_{age} and $T_{ZIP\ code}$ individually satisfy distinct 2-diversity [2], which is a traditional privacy preserving requirement for data publishing. Even if an adversary can obtain access to either T_{age} or $T_{ZIP\ code}$ and possess the identification information (the age and the ZIP code) of an individual (the victim), the adversary cannot deduce the exact sensitive attribute value of the victim. At best, the adversary can only narrow down the number of possible sensitive attribute values to two. However, ensuring that T_{age} and $T_{ZIP\ code}$ satisfy distinct 2-diversity individually is not enough to preserve privacy. We will show that if an adversary can obtain access to both T_{age} and $T_{ZIP\ code}$, the adversary may be able to deduce some information that cannot be deduced from either T_{age} or $T_{ZIP\ code}$ individually, and such information may cause privacy breaches. Suppose an adversary knows the following identification information of the victim.

Table 1. Two private tables $PriT_R$ and $PriT_C$.

Name	Age	Gender	Salary Class
Alice	60	F	High
Bob	50	M	High
Carlo	55	M	High
Diana	50	F	High
Eva	48	F	Middle
Fred	43	M	Middle
George	25	M	Middle
Helen	37	F	Middle
Ivan	57	M	Low
Janice	57	F	Low
Kate	18	F	Low
Leslie	22	F	Low

(a) Private table $PriT_R$

Name	ZIP Code	Salary Class
Alice	25434	High
Bob	27343	High
Carlo	19343	High
Diana	17234	High
Eva	28544	Middle
Fred	24453	Middle
George	26211	Middle
Helen	23094	Middle
Ivan	29454	Low
Janice	12845	Low
Kate	15341	Low
Leslie	22093	Low

(b) Private table $PriT_C$

Table 2. Two published tables T_{age} and $T_{ZIP\ code}$.

Age	Salary Class
≥ 40	High
≥ 40	High
≥ 40	High
≥ 40	High
≥ 40	Middle
≥ 40	Middle
≥ 40	Low
≥ 40	Low
< 40	Middle
< 40	Middle
< 40	Low
< 40	Low

(a) Published Table T_{age}

ZIP Code	Salary Class
≥ 20k	High
≥ 20k	High
≥ 20k	High
≥ 20k	High
≥ 20k	Middle
≥ 20k	Middle
≥ 20k	Low
≥ 20k	Low
< 20k	Middle
< 20k	Middle
< 20k	Low
< 20k	Low

(b) Published Table $T_{ZIP\ code}$

- Both T_{age} and $T_{ZIP\ code}$ contain a tuple that corresponds to the victim.
- The victim's age < 40 and the victim's ZIP Code < 20k.

From T_{age} (Table 2(a)) and the fact that the victim's age < 40, the adversary deduces that the salary class of the victim is either “Low” or “Middle”. From $T_{ZIP\ code}$ (Table 2(b)) and the fact that the victim's ZIP code < 20k, the adversary deduces that the salary class of the victim is either “Low” or ”High”. Since both T_{age} and $T_{ZIP\ code}$ have a tuple that corresponds to the victim, the adversary can combine the above results and deduce that the salary class of the victim must be ”Low”. In other words, the adversary can deduce the exact sensitive value of the victim. Hence, 2-diversity is violated even if both T_{age} and $T_{ZIP\ code}$ are 2-diverse tables individually.

Table 3. The centralized table $PriT_{R,C}$ (for centralized data publishing only).

Name	**Age**	**Gender**	**ZIP Code**	**Salary Class**
Alice	60	F	25434	High
Bob	50	M	27343	High
Carlo	55	M	19343	High
Diana	50	F	17234	High
Eva	48	F	28544	Middle
Fred	43	M	24453	Middle
George	25	M	26211	Middle
Helen	37	F	23094	Middle
Ivan	57	M	29454	Low
Janice	57	F	12845	Low
Kate	18	F	15341	Low
Leslie	22	F	22093	Low

Table 4. A published table $T_{age,\ ZIP\ code}$ (for centralized data publishing only).

Age	**ZIP Code**	**Salary Class**
≥ 40	≥ 20k	High
≥ 40	≥ 20k	High
≥ 40	≥ 20k	Middle
≥ 40	≥ 20k	Middle
≥ 40	≥ 20k	Low
≥ 40	< 20k	High
≥ 40	< 20k	High
≥ 40	< 20k	Low
< 40	≥ 20k	Middle
< 40	≥ 20k	Middle
< 40	≥ 20k	Low
< 40	< 20k	Low

The Revenue Department and the Censor Department not only need to ensure that their published tables, T_{age} and $T_{ZIP\ code}$, satisfy a certain privacy preserving requirement individually, but also have to ensure that no privacy breaches will occur if both published tables are obtained by an adversary. The solution is trivial if the Revenue Department and the Censor Department can share their private tables, $PriT_R$ and $PriT_C$, with each other. It is because they can first combine $PriT_R$ and $PriT_C$ together to form a centralized table $PriT_{R,C}$ as shown in Table 3 by matching the name attributes of the two private tables. Then, they can construct published table $T_{age,\ ZIP\ code}$ as shown in Table 4 by generalizing the age and the ZIP code attributes of the centralized table $PriT_{R,C}$. Both T_{age} and $T_{ZIP\ code}$ can be obtained by suppressing an attribute in $T_{age,\ ZIP\ code}$. If $T_{age,\ ZIP\ code}$ satisfies a certain privacy preserving requirement, releasing both T_{age} and $T_{ZIP\ code}$ does not cause privacy breaches. It is because the private information deduced from an adversary who can obtain access to both T_{age} and $T_{ZIP\ code}$ must be no more than the private information stored in $T_{age,\ ZIP\ code}$. However, in our example, $T_{age,\ ZIP\ code}$ (Table 4) is only a 1-diverse table. Hence, the Revenue Department and the Censor Department know that the adversary may be able to deduce the exact sensitive attribute value of an individual if the adversary can obtain access to both T_{age} and $T_{ZIP\ code}$. Therefore, they will not release T_{age} and $T_{ZIP\ code}$ together and thus privacy breaches are avoided.

Unfortunately, the above solution may not be applicable in practice. It is because due to privacy reasons or even enforced by law, the Revenue Department and the Censor Department cannot share their private tables with each other (though both departments belong to the government). It follows that no one can precisely construct $PriT_{R,C}$ and generate $T_{age,\ ZIP\ code}$. Hence, the above solution cannot be applied when such a restriction is enforced. With such a restriction, this problem becomes non-centralized data publishing. The solution to this problem is not trivial since the Revenue Department and the Censor Department cannot share information with each other, but they still have to generate T_{age} and $T_{ZIP\ code}$ that do not cause privacy breach even if there is an adversary who can obtain access to both T_{age} and $T_{ZIP\ code}$.

The objective of this paper is to present a way to release published tables that will not cause privacy breaches in non-centralized environment. Firstly, we propose two approaches for a data holder to determine the private information that will be deduced by an adversary who can get access to all non-centralized published tables if the data holder releases a published table. Since there may be correlations among the attributes in different published tables, two approaches are needed. The first approach is for the scenario that the adversary knows such correlations. This approach determines the private information deduced by such an adversary by considering how the adversary utilizes the correlations to deduce the private information. The second approach is for the scenario that the adversary does not know such correlations. This approach determines the private information deduced by such an adversary by considering how the adversary makes a conditional independent assumption to deduce private information. For both approaches, the data holder does not need to communicate with other data holders. Instead, the data holder only needs to collect published tables generated by other data holders. Secondly, we propose how to adopt distinct *l*-diversity [2], which is a privacy preserving requirement for centralized data publishing, to non-centralized data publishing. It is so called non-centralized distinct *l*-diversity. Thirdly, we propose an algorithm for a data holder to generate published tables that satisfy non-centralized distinct *l*-diversity. Our algorithm is modified from Incognito [13], which is an efficient and widely-used algorithm for (centralized) data publishing. Theorems are presented to show the correctness of the proposed algorithm. Experiments on real datasets are conducted to show that the proposed algorithm is feasible to real applications.

This paper is organized as follows. Section 2 presents the related work. Section 3 presents the table models used in this paper. Section 4 discusses the private information deduced from multiple published tables in a non-centralized environment. Section 5 presents the non-centralized version of distinct *l*-diversity and an algorithm for non-centralized PPDP. Section 6 presents the experiment results. Finally, conclusions are presented in Section 7.

2. RELATED WORK

This paper focuses on data publishing on non-centralized datasets. There has been extensive research in PPDP, such as *k*-anonymity [1], *l*-diversity [2], and *t*-closeness [3]. However, they focus on the cases for a single centralized dataset and a single published table. In the previous section, we have shown that these solutions cannot be directly applied to the cases for multiple published tables.

Recently, the problem of serial data publishing has attracted a lot of attentions. These research works can be classified into serial data publishing on a dynamic dataset and serial data publishing on a static dataset. Most of the works focus on the cases for a dynamic dataset (or multiple instances of data), which includes an incremental dataset [4][5] (i.e. allowing insertions of new tuples), a dataset with insertions and deletions [6], a dataset with insertions, deletions, and updates [7], a dataset with sensitive values that change over time [8]. Some other works focus on the cases for a static dataset, such as [9]. Nevertheless, all such works cannot be applied to our problem. It is because they require the data holder to possess a dataset or a series of dataset instances that contains all information. However, in the non-centralized datasets problem, no one possesses all information.

We propose a statistical approach to determine the private information deduced by an adversary and the results are stored in a probabilistic table. Many previous works also use probabilistic approach [10] or deal with probabilistic databases [11][12].

Our proposed algorithm that generates published tables is modified from Incognito [13]. Similar to many previous works [1][14][15][16], published tables are generated by generalizing and removing the quasi-identifier attributes of the private table, while remaining the sensitive attribute unchanged.

In the non-centralized data publishing problem, no one (neither the data holders nor the adversaries) is able to combine all the private/published tables and deduce the actual correlations among the quasi-identifier attributes in different private/published tables. Without such correlations, even if an adversary can get access to two or more published tables, the adversary can only deduce information stored in the tables individually, but not the "combined" information reflected by these tables. Nevertheless, the adversary can make the conditional independence assumption, which is not uncommon in database research works [17][18][19][20] and other research works [21], such as Bayesian analysis [22][23], to deduce the "combined" information reflected by the published tables.

This paper focuses on PPDP, which focuses on how to publish data. There is another related area called privacy preserving data mining (PPDM), which focuses on how to perform data mining on the data that is modified to preserve privacy [24][25][26].

3. TABLE MODELS

In this section, we present the private table model (Section 3.1) and the published table model (Section 3.2).

3.1. Private Table

A private table stores private information about a group of individuals. It is solely owned by a data holder and is never released to other parties directly. In this paper, we focus on the case that there are multiple non-centralized private tables that store information about the same group of individuals. For example, Table 1 shows two private tables, namely $PriT_R$ and $PriT_C$, which are solely owned by the Revenue Department and the Censor Department respectively. Both $PriT_R$ and $PriT_C$ store income information about a group of individuals. A tuple in the private table corresponds to an individual. A private table consists of three sets of attributes:

- **Explicit-identifier attributes**. Each of them can uniquely identify an individual in the private table, such as the name.
- **Quasi-identifier attributes** [1]. It is a set of attributes that may uniquely identify an individual in the private table when some of such attributes are considered together and linked to an external dataset, such as the age and the gender.
- **A sensitive attribute**. It stores sensitive information about an individual, such as the salary class. In this paper, we consider the private tables that share a common sensitive attribute and store information about the same group of individuals.

3.2. Published Table

Some parties, say data miners, may find the information stored in the private table useful for research or other purposes. However, due to privacy reasons or even enforced by law, the data holder cannot release the private table to the data miners directly. Hence, the data holder needs to generate a table with less information and release it to the data miners instead. Such a table is called a published table. A popular approach to construct a published table from a private table is to suppress the explicit identifier attributes, generalize or suppress the quasi-identifier attributes, and retain the sensitive attribute of the private table [1][14][15][16]. For example, T_{age} and $T_{ZIP\ code}$ in Table 2 are two published tables for private tables $PriT_R$ and $PriT_C$ respectively. This paper considers non-centralized data publishing, which refers to generating published tables from multiple non-centralized private tables, where each private table has a published table. Let n be the number of private tables. Then, there will be n published tables. Let T_i be the published table for the private table $PriT_i$, where $1 \leq i \leq n$. A published table T_i consists of two sets of attributes:

- **Quasi-identifier attributes**. Each of them is generalized from a quasi-identifier attribute in the corresponding private table. Some quasi-identifier attributes may be suppressed and thus not all quasi-identifier attributes in the private table appear in the published table. Let m be the number of quasi-identifier attributes among all published tables. The quasi-identifier attributes are denoted as A_i for $1 \leq i \leq m$. In this paper, duplicate quasi-identifier attributes among the published tables are discarded. A group of tuples with the same quasi-identifier attribute values form an equivalence class (EC) (or a QI-group). For example, T_{age} in Table 2(a) has one quasi-identifier attribute *age* and two ECs, namely $EC_{age \geq 40}$ and $EC_{age < 40}$.
- **A sensitive attribute**. All published tables share a common sensitive attribute S, which is the same as the sensitive attribute shared by all private tables.

4. Private Information Deduced from Multiple Published Tables

In this section, we discuss the private information deduced by an adversary who can obtain access to multiple published tables for non-centralized data publishing. Data holders have to know what private information can be deduced by such an adversary in order to generate published tables that do not cause privacy breaches. Section 4.1 shows that some private information can be deduced by simple counting on each published table individually. Section 4.2 shows that more private information can be deduced by considering multiple published tables together with (Section 4.2.1) or without (Section 4.2.2) knowing the correlation among the quasi-identifier attributes. Section 4.3 shows that the deduced private information can be systematically stored in a probabilistic table, which is needed for our proposed algorithm to generate non-centralized published tables.

4.1. Private Information Deduced by Simple Counting on Each Published Tables

Consider a victim who has a tuple in both published tables T_{age} and $T_{ZIP\ code}$ depicted in Table 2. Let h, m, and l be the events that the salary classes of the victim are “High”, “Middle”, and “Low” respectively. Let P(H), P(M), and P(L) be the probabilities that the salary classes of the victim are “High”, “Middle”, and “Low” respectively. We assume that the private tables are consistent and hence P(H) can be computed by counting the number of tuples with sensitive attribute values = “High” in either T_{age} or $T_{ZIP\ code}$. In our example, there are four out of 12 tuples with values equal to “High” in each of T_{age} and $T_{ZIP\ code}$, which store information about the same group of individuals and the sensitive attribute values are the same. Hence, we have P(H) = 4/12 = 0.333. Similarly, we have P(M) = P(L) = 0.333.

Consider an adversary who (1) can obtain access to both T_{age} and $T_{ZIP\ code}$, (2) knows that there is a tuple in both T_{age} and $T_{ZIP\ code}$ that corresponds to the victim, and (3) knows that the victim's age ≥ 40 and the victim's ZIP code ≥ 20k. The adversary can deduce some private information, in terms of conditional probabilities, about the victim from each of T_{age} and $T_{ZIP\ code}$ individually. First we consider T_{age}. Since the victim's age ≥ 40, the tuple that corresponds to the victim must be one of the first eight tuples. Among these eight tuples, four of them have the sensitive attribute value = "High". Hence, the conditional probability that the salary class of the victim is "High" given that the victim's age ≥ 40 is 4/8 = 0.5. Such a conditional probability is denoted as $P(H|A \geq 40)$, where A be the age of the victim. Similarly, the adversary can deduce conditional probabilities $P(M|A \geq 40) = 0.25$ and $P(L|A \geq 40) = 0.25$ from T_{age}. Next we consider $T_{ZIP\ code}$. Let Z be the ZIP code of the victim. Similarly, the adversary can deduce $P(H|Z \geq 20\text{k}) = 0.25$, $P(M|Z \geq 20\text{k}) = 0.5$, and $P(L|Z \geq 20\text{k}) = 0.25$ from $T_{ZIP\ code}$.

4.2. Private Information Deduced from Multiple Published Tables

As mentioned in Section 1, although published tables T_{age} and $T_{ZIP\ code}$ (Table 2) satisfy the 2-diversity requirement individually, an adversary who can obtain access to both T_{age} and $T_{ZIP\ code}$ may be able to deduce further private information that violates such a requirement. We propose some approaches to determine such further private information deduced by the adversary. The proposed approach can be used to verify whether releasing a non-centralized published table cause privacy breaches, which is an important step for our proposed algorithm to generate non-centralized published tables.

In Section 4.1, we presented the conditional probabilities deduced from each of T_{age} and $T_{ZIP\ code}$ by the simple counting approach. The conditions of such conditional probabilities involve either A or Z, but not both. As an example, the conditional probabilities deduced in the previous subsection are $P(H|A \geq 40)$ and $P(H|Z \geq 20\text{k})$. However, the adversary actually wants to compute $P(H|A \geq 40, Z \geq 20\text{k})$, which is the conditional probability that the sensitive attribute value of the victim is high given that $A \geq 40$ and $Z \geq 20\text{k}$. It is because the adversary knows the age and the ZIP code of the victim and is able to obtain access to both T_{age} and $T_{ZIP\ code}$.

Although the age and the ZIP code attributes are in different private tables hold by different data holders, it is not surprising that there is a correlation between the two attributes. For example, a city may have more elder people than younger people. Hence, the probability that an individual who lives in such a city is an elderly is higher. There are two ways to compute $P(H|A \geq 40, Z \geq 20\text{k})$. The first way is for the adversary who does not know the QI attribute correlation. Such an adversary can make the conditional independent assumption to compute $P(H|A \geq 40, Z \geq 20\text{k})$. The second way is for the adversary who does not know the QI attribute correlation. Such an adversary can use a statistical approach to compute $P(H|A \geq 40, Z \geq 20\text{k})$. In the rest of this subsection, we will discuss these two ways to compute $P(H|A \geq 40, Z \geq 20\text{k})$.

4.2.1 Without Knowing the Quasi-Identifier Attribute Correlation

Consider an adversary who does not know the correlation between the age and the ZIP code attributes. It seems that the adversary can compute the $P(H|A \geq 40, Z \geq 20\text{k})$ value by assuming that the age and the ZIP code attributes are independent, and then use the independence property $P(A \geq 40, Z \geq 20\text{k}) = P(H|A \geq 40) \cdot P(Z \geq 20\text{k})$. Unfortunately, such an assumption cannot be made. It is because the knowledge about the ZIP code of an individual may be updated by knowing the age of the individual. It is illustrated in the following example. Suppose the adversary does not know the age of the individual. In $T_{ZIP\ code}$, there are 12 tuples and eight of them have ZIP code $\geq$ 20k. Hence, the probability that the individual's ZIP code $\geq$ 20k is 2/3. Such a probability is denoted as p. On the other hand, suppose the adversary knows that the individual's age < 40. From T_{age}, the adversary knows the individual's tuple must be one of the last four tuples. Hence, the salary class of the individual is either "Low" or "Middle" with equal chance. With such information, the adversary can eliminate the tuples with sensitive attribute value = "High" in $T_{ZIP\ code}$. Among the remaining eight tuples, six of them have ZIP code $\geq$ 20k. Hence, the adversary now has $p = 3/4$. In other words, by knowing the victim's age < 40, the value of p is updated from 2/3 to 3/4. This counter example explains why the adversary cannot assume that the age and the ZIP code attributes are independent.

In fact, to compute $\mathrm{P}(H|A \geq 40, Z \geq 20\mathrm{k})$, the adversary who does not know the QI attribute correlation can make the following assumption: the age and ZIP code attributes are conditionally independent given the value of the salary class attribute. In probability theory, conditional independence for events E_1 and E_2 is defined as follows: E_1 and E_2 are two conditionally independent events given event G if and only if $\mathrm{P}(E_1,E_2|G) = \mathrm{P}(E_1|G) \cdot \mathrm{P}(E_2|G)$.

Here we use an example to illustrate how to make the conditional independence assumption in our problem. If the salary class of the individual is given in the first place, the knowledge about the ZIP code of an individual is independent of the knowledge of the age of the individual. In other words, knowing the ZIP code (or the age) of the individual will not update the knowledge about the age (or the ZIP code) of the individual. It is illustrated in the following example. Suppose the information that the salary class of the victim is either "Low" or "Middle" with equal chance is given in the first place. With such information, the adversary has $p = 3/4$ already even if we do not know the individual's age and only consider $T_{ZIP\ code}$. Hence, even if the adversary now knows that the individual's age < 40, the value of p will not be updated. This is also true if the salary class of the victim equals to other value(s). It is because the salary class attribute is the only connection between the age and the ZIP code attributes if the adversary does not know the QI attribute correlation. By knowing the age attribute, one can update the knowledge about the salary class attribute, which can be used to update the knowledge about the ZIP code attribute. If such an updated knowledge about the salary class attribute is given in the first place, it will update the knowledge about the ZIP code of the individual regardless of the knowledge about the age of the individual.

$$\frac{P(H,A\geq40,Z\geq20k)}{P(A\geq40,Z\geq20k)} \tag{1}$$

$$\frac{P(H,A\geq40,Z\geq20k)}{P(H,A\geq40,Z\geq20k)+ P(M,A\geq40,Z\geq20k)+ P(L,A\geq40,Z\geq20k)} \tag{2}$$

$$\frac{P(A\geq40,Z\geq20k|H)\cdot P(H)}{P(A\geq40,Z\geq20k|H)\cdot P(H)+P(A\geq40,Z\geq20k|M)\cdot P(M)+P(A\geq40,Z\geq20k|L)\cdot P(L)} \tag{3}$$

$$\frac{P(A\geq40|H)\cdot P(Z\geq20k|H)\cdot P(H)}{P(A\geq40|H)\cdot P(Z\geq20k|H)\cdot P(H)+P(A\geq40|M)\cdot P(Z\geq20k|M)\cdot P(M)+P(A\geq40|L)\cdot P(Z\geq20k|L)\cdot P(L)} \tag{4}$$

With such an assumption, the adversary can compute $\mathrm{P}(H|A \geq 40, Z \geq 20\mathrm{k})$ as follows.

- By the definition of conditional probability, $\mathrm{P}(H|A \geq 40 , Z \geq 20\mathrm{k})$ can be expressed as Equation (1) .
- The salary class has only three possible values, by the law of alternatives, Equation (1) can be expressed as Equation (2).
- By the product rule of conditional probabilities, Equation (2) can be expressed as (3).
- By the definition of conditional independence, Equation (3) can be expressed as Equation (4).
- By simple counting on T_{age} and $T_{ZIP\ code}$ individually, the adversary can deduce $\mathrm{P}(A \geq 40|H) = 0.5$, $\mathrm{P}(Z \geq 20\mathrm{k}|H) = 0.25$, $\mathrm{P}(H) = 0.333$, $\mathrm{P}(A \geq 40|M) = 0.25$, $\mathrm{P}(Z \geq 20\mathrm{k}|M) = 0.5$, $\mathrm{P}(M) = 0.333$, $\mathrm{P}(A \geq 40|L) = 0.25$, $\mathrm{P}(Z \geq 20\mathrm{k}|L) = 0.25$, and $\mathrm{P}(L) = 0.333$. By substituting these values into the Equation (4), the adversary can obtain: $\mathrm{P}(H|A \geq 40, Z \geq 20\mathrm{k}) = 0.4$.

Similarly, the adversary can also calculate the conditional probabilities for other sensitive attribute values: $\mathrm{P}(M|A \geq 40, Z \geq 20\mathrm{k}) = 0.4$ and $\mathrm{P}(L|A \geq 40, Z \geq 20\mathrm{k}) = 0.2$.

In general, suppose there are n published tables, namely T_1, ..., T_n, which are generated from n private tables $PriT_1$, ..., $PriT_n$ respectively. Further suppose there are m quasi-identifier attributes, namely A_1, ..., A_m, which are distributed among these n published tables. The n published tables can be formulated as:

$$T_1 = \{A_1, \ldots, A_{r_1}, S\}, T_2 = \{A_{r_1+1}, \ldots, A_{r_2}, S\}, \ldots, T_n = \{A_{r_{n-1}+1}, \ldots, A_m, S\},$$

where $1 \leq r_1 < r_2 < \ldots < r_{n-1} < m$.

The conditional independence assumption for such a general case is stated as follows.

Assumption 1. The quasi-identifier attributes among different published tables, i.e. $A_1, \ldots, A_{r_1}$ in T_1, $A_{r_1+1}, \ldots, A_{r_2}$ in T_2, ..., and $A_{r_{n-1}+1}, \ldots, A_m$ in T_n, are conditionally independent given the value of sensitive attribute *S*.

Assumption 1 is needed by the adversary who does not know the QI attribute correlation to deduce private information from multiple published tables by considered them together. Note that if the adversary knows the QI attribute correlation, the adversary does not need this assumption to deduce the required private information (see the next part of this subsection).

The idea of Assumption 1 is that if the value of *S* is given in the first place, the knowledge about the A_1, ..., A_m values of an individual are independent to each other. In other words, knowing the A_i value of the individual will not update the knowledge about the A_j value of the individual for $1 \leq i \neq j \leq m$. In probability theory, conditional independence for events E_1, ..., E_m is defined as follows.

Definition 1. (Conditional independence for *m* events) E_1, ..., E_m are conditionally independent events given event *G* if and only if $P(E_1, \ldots, E_m|G) = \prod_{i=1}^{m} P(E_i|G)$ for $m \geq 2$.

Consider the worst case that the adversary can obtain access to all *n* published tables and know the values of all *m* quasi-identifier attributes for the victim. Let S_k be a possible value of *S* for $1 \leq k \leq s$.

Here shows that the adversary is able to compute P($S_k|A_1 \geq v_1$, ..., $A_m \geq v_m$), which is conditional probability that the sensitive attribute value of the victim is S_k given that $A_1 \geq v_1$, ..., $A_m \geq v_m$.

P($S_k|A_1 \geq v_1$, ..., $A_m \geq v_m$) can be expressed as an expression containing terms that can be calculated by simple counting on the published tables (such an expression is the general case for Equation (4). By the definition of conditional probability, it can be expressed as

$$\frac{P(S_k, A_1 \geq v_1, \ldots, A_m \geq v_m)}{P(A_1 \geq v_1, \ldots, A_m \geq v_m)}$$

By the law of alternatives, the above can be expressed as

$$\frac{P(S_k, A_1 \geq v_1, \ldots, A_m \geq v_m)}{\sum_{j=1}^{s} P(S_j, A_1 \geq v_1, \ldots, A_m \geq v_m)}$$

By the product rule of conditional probabilities, the above can be expressed as

$$\frac{P(A_1 \geq v_1, \ldots, A_m \geq v_m|S_k) \cdot P(S_k)}{\sum_{j=1}^{s} P(A_1 \geq v_1, \ldots, A_m \geq v_m|S_j) \cdot P(S_j)}$$

By Assumption 1 and Definition 1, the above can be expressed as

$$\frac{(\prod_{i=1}^{n} p_{i,k}) \cdot P(S_k)}{\sum_{j=1}^{s}((\prod_{i=1}^{n} p_{i,j}) \cdot P(S_j))} \qquad (5)$$

where $p_{1,k} = P(A_1 \geq v_1, \ldots, A_{r_1} \geq v_{r_1}|S_k)$, $p_{2,k} = P\left(A_{r_1+1} \geq v_{r_1+1}, \ldots, A_{r_2} \geq v_{r_2}\middle|S_k\right)$, $\ldots, p_{n,k} = P\left(A_{r_{n-1}+1} \geq v_{r_{n-1}+1}, \ldots, A_m \geq v_m\middle|S_k\right)$, and v_i is a value in the domain of A_i for $1 \leq i \leq m$. (Recall that *m* is the number of QI attributes, *n* is the number of published tables, and *s* is the number of possible sensitive values.)

All the terms in Equation (5) can be deduced easily. As described earlier, the probabilities P(S_1), ..., P(S_s) can be calculated by simple counting on any one of T_1, ..., T_n; the conditional probabilities $p_{1,k}$, ... $p_{n,k}$ can be calculated by simple counting on each of T_1, ..., T_n respectively. Therefore, the adversary is able to compute P(S_k | $A_1 \geq v_1$, ..., $A_m \geq v_m$).

Similarly, the adversary can also calculate conditional probabilities from P(S_k|$A_1 \geq v_1, ..., A_{m-1} \geq v_{m-1}, A_m < v_m$) to P($S_k$|$A_1 < v_1, ..., A_m < v_m$).

Table 5. A published table $T_{age,\ ZIP\ code}$ (for centralized data publishing only).

Age	Salary Class	Number of Tuples
≥ 40	High	3
	Middle	3
	Low	4
< 40	High	4
	Middle	4
	Low	3

(a) Published table T'_{age}

ZIP Code	Salary Class	Number of Tuples
≥ 20k	High	4
	Middle	3
	Low	2
< 20k	High	3
	Middle	4
	Low	5

(b) Published table $T'_{ZIP\ code}$

4.2.2 Knowing the Quasi-Identifier Attribute Correlation

Table 5 shows two published tables T'_{age} and $T'_{ZIP\ code}$ with different presentations such that they store the number of tuples having each sensitive value for each EC. Such a presentation is more convenient to illustrate our idea presented in this subsection.

Consider an adversary who can get access to T'_{age} and $T'_{ZIP\ code}$ and knows the correlation between the age and the ZIP code attributes. Such a correlation can be expressed in a probability equation, e.g. P($A \geq 40$|$Z \geq 20$k) = 4/9. We assume that such a correlation is consistent with the private tables and the published tables.

Let $PriT'_{age}$ and $PriT'_{ZIP\ code}$ be the private tables of T'_{age} and $T'_{ZIP\ code}$ respectively. Since no one can get access to both $PriT'_{age}$ and $PriT'_{ZIP\ code}$, no one can construct a table that precisely combines the information about age, ZIP code, and salary in both $PriT'_{age}$ and $PriT'_{ZIP\ code}$. Such a table is denoted as $T'_{age,\ ZIP\ code}$ and it is an imaginary table only (it cannot be constructed precisely). Nevertheless, the adversary can still deduce the following information about the imaginary table.

- There are four tuples in $EC_{A \geq 40,\ Z \geq 20k}$ in the imaginary table. It is because there are nine tuples in $EC_{Z \geq 20k}$ in $T'_{ZIP\ code}$ and P($A \geq 40$|$Z \geq 20$k) = 4/9.
- The maximum number of tuples in $EC_{A \geq 40,\ Z \geq 20k}$ in the imaginary table having sensitive value "High" is three. It is because such a number is the minimum of the number of tuples in $EC_{A \geq 40}$ in T'_{age} and the number of tuples in $EC_{Z \geq 20k}$ in $T'_{ZIP\ code}$ having sensitive value "High", which are three and four respectively.
- Similarly, the maximum numbers of tuples in $EC_{A \geq 40,\ Z \geq 20k}$ in the imaginary table having sensitive value "Middle" and "Low" are three and two respectively.

From the above information, although the adversary cannot construct the imaginary table precisely, the adversary can compute conditional probabilities P(H|$A \geq 40$, $Z \geq 20$k), P(M|$A \geq 40$, $Z \geq 20$k), and P(L|$A \geq 40$, $Z \geq 20$k) as follows.

The adversary first constructs all possible instances of the imaginary table. Let h, m, and l be the number of tuples in an imaginary table instance having sensitive value of "High", "Middle", and "Low"' respectively.

The adversary can write a simple program to find out all possible values of h, m, and l. Table 6 shows such possible values for our example. We observe that there are 10 possible imaginary table instances with equal chances. For the first instance, there are three out of four tuples in $EC_{A \geq 40,\ Z \geq 20k}$ with sensitive value "High". The value of P(H|$A \geq 40$, $Z \geq 20$k) can be computed by considering all instances as follows:

$$P(H|A \geq 40, Z \geq 20k) = \frac{1}{10} \cdot \left(\frac{3}{4} + \frac{3}{4} + \frac{2}{4} + \frac{2}{4} + \frac{2}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4}\right) = \frac{3}{8}$$

Similarly, we have P($M|A \geq 40$, $Z \geq 20$k) = 3/8 and P($M|A \geq 40$, $Z \geq 20$k) = 1/4.

Table 6. The possible values of *h*, *m*, and *l* for all ten possible imaginary table instances.

Instance	***h***	***m***	***l***
1	3	1	0
2	3	0	1
3	2	2	0
4	2	1	1
5	2	0	2
6	1	3	0
7	1	2	1
8	1	1	2
9	0	3	1
10	0	2	2

Table 7. Probabilistic table $PT_{age,ZIP\ code}$.

Age	**ZIP Code**	**Salary Class**	**Probability**
≥ 40	≥ 20k	High	0.4
≥ 40	≥ 20k	Middle	0.4
≥ 40	≥ 20k	Low	0.2
≥ 40	< 20k	High	0.667
≥ 40	< 20k	Middle	0
≥ 40	< 20k	Low	0.333
< 40	≥ 20k	High	0
< 40	≥ 20k	Middle	0.667
< 40	≥ 20k	Low	0.333
< 40	< 20k	High	0
< 40	< 20k	Middle	0
< 40	< 20k	Low	1

It may not be possible for a data holder to know what the QI attribute correlation will be obtained by an adversary. Nevertheless, the data holder can still use the above equation to make the published table satisfy non-centralized *l*-diversity regardless of the QI attribute correlation possessed by the adversary. More details will be explained in the next subsection.

4.3. Probabilistic Table

The conditional probabilities computed in the previous subsections can be stored systematically in a table called a probabilistic table, which allows data holders to estimate the information deduced by an adversary in order to verify whether releasing a published table will cause privacy breaches. A probabilistic table for *n* published tables is denoted as $PT_{1\sim n}$. It consists of three sets of attributes:

- **Quasi-identifier attributes**. The quasi-identifier attributes of all previous published tables, i.e., A_1, ..., A_m. Same as an equivalence class of a published table, an equivalence class of a probabilistic table is defined as a collection of tuples with the same quasi-identifier attribute values.

- **A sensitive attribute**. It is the common sensitive attribute of all previous published tables. Let S be the number of possible sensitive attribute values. Each equivalence class has exactly S tuples, which contain sensitive attribute values S_1, ..., S_s.
- **A probability attribute**. The probability attribute of the tuple with $A_1 \geq v_1$, ..., $A_m \geq v_m$ and $S = S_k$ stores the $P(S_k|A_1 \geq v_1, ..., A_m \geq v_m)$ value computed in Equation 5. Similarly, the values of other conditional probabilities from $P(S_k|A_1 \geq v_1, ..., A_{m-1} \geq v_{m-1}, A_m < v_m)$ to $P(S_k|A_1 < v_1, ..., A_m < v_m)$ for $k = 1, .., s$ are also stored in this attribute.

As an example, Table 7 shows $PT_{age,\ ZIP\ code}$, which is a probabilistic table for published tables T_{age} and $T_{ZIP\ code}$ in Table 2. The probabilistic table $PT_{age,\ ZIP\ code}$ contains quasi-identifier attributes “Age” and “ZIP Code”, a sensitive attribute “Salary Class”, and an attribute storing probability values. The first row of $PT_{age,\ ZIP\ code}$ indicates that given the victim's age ≥ 40 and the victim's ZIP code ≥ 20k, the conditional probability that the salary class of the victim is “High” is 0.4.

In practice, an adversary may only be able to obtain access to some of the n published tables and/or may only know that some of the m quasi-identifier attribute values that correspond to the victim. Let A_1, ..., $A_{m'}$ be such quasi-identifier attributes in the published tables that can be accessed by the adversary, where $m' < m$. In this case, we cannot use $PT_{1\sim n}$ to represent the information deduced by such an adversary. It is because the adversary does not possess enough information to compute those conditional probabilities in $PT_{1\sim n}$. Instead, we use another probabilistic table that only contains quasi-identifier attributes A_1, ..., $A_{m'}$, a sensitive attribute S, and a probability attribute to represent the information deduced by the adversary who ignores the quasi-identifier attributes $A_{m'+1}$, ..., A_m that he/she is not knowledgeable about. Nevertheless, for preserving privacy, we only need to consider the worst case: the adversary is able to access all n published tables. Hence, we do not need to explicitly construct all possible probabilistic tables.

5. Privacy Preserving Requirement and Algorithm for Non-Centralized Data Publishing

In this section, we present the non-centralized version of distinct l-diversity and an algorithm for non-centralized PPDP. Section 5.1 proposes how to adopt distinct l-diversity as a privacy preserving requirement for non-centralized data publishing. Section 5.2 proposes an algorithm, which is modified from Incognito [13], to generate a published table that satisfies non-centralized distinct l-diversity. Section 5.3 proves the correctness of applying the modified version of Incognito to the non-centralized data publishing problem.

5.1. Non-centralized Distinct *l*-diversity

Suppose there are n data holders, denoted as DH_1, ..., and DH_n. Each of them solely owns a private table, which have a common sensitive attribute S with the same domain and store information about the same group of individuals. Let Pri_i be the private table owned by DH_i for $1 \leq i \leq n$. For research or other purpose, a published version of the private table (the published table) is released by the data holders. Let T_i be the published table released by DH_i for $1 \leq i \leq n$. Each published table contains the common sensitive attribute and some quasi-identifer attributes. Let A_1, ..., A_m be the m quasi-identifier attributes in these n published table. Without loss of generality, we assume that DH_n is the last data holder to release its published table T_n, which contains quasi-identifier attributes $A_{m'+1}$, ..., A_m, where $m' < m$. DH_n has to ensure that there are no privacy breaches for releasing T_n if it is obtained by an adversary who possesses T_1, ..., T_{n-1} and the identification information of the individuals in these published tables.

We propose that distinct l-diversity [2], which is a privacy preserving requirement for centralized data publishing, can also be used for non-centralized data publishing. In other words, it can be used by DH_n to ensure that releasing T_n does not cause privacy breaches. It is explained as follows. Consider an equivalence class in probabilistic table $PT_{1\sim n}$. The number of tuples having non-zero probability attribute values can be regarded as the number of possible sensitive attribute values of the victim deduced by an adversary who possesses the identification information of the victim. It shares the same semantic of the *l* value in distinct *l*-diversity. Hence, we can define distinct *l*-diversity for non-centralized data publishing under our proposed framework as follows.

Defintion 2. (Distinct *l*-diversity for an equivalence class) An equivalence class in probabilistic table $PT_{1\sim n}$ satisfies distinct *l*-diversity if and only if it has at least *l* tuples having non-zero probability attribute values.

For example, the four equivalence classes in $PT_{age,\ ZIP\ code}$ (Table 7) satisfy distinct *l*-diversity for $l = 3, 2, 2, 1$ (from the top to the bottom).

According to distinct *l*-diversity for centralized data publishing, a published table satisfies distinct *l*-diversity if and only if all of its equivalence classes satisfy distinct *l*-diversity. Here we have the same definition for non-centralized data publishing under our proposed framework. It is formally defined as follows.

Definition 3. (Distinct *l*-diversity for $PT_{1\sim n}$) A probabilistic table $PT_{1\sim n}$ satisfies distinct *l*-diversity if and only if all of its equivalence classes satisfy distinct *l*-diversity.

For example, $PT_{age,\ ZIP\ code}$ (Table 7) only satisfies distinct *l*-diversity for $l = 1$.

$PT_{1\sim n}$ represents the worst case that an adversary can obtain access to all *n* published tables and knows the values of all *m* quasi-identifier attributes of all individuals. According to the definition of distinct *l*-diversity [2], if $PT_{1\sim n}$ satisfies distinct *l*-diversity, we can say that no privacy breaches will occur for such a worst case. However, in practice, other possible cases may happen such that the adversary may only be able to obtain access to *n'* published tables and/or may only know *m'* quasi-identifier values about the victim, where $n' < n$ and $m' < m$. Theorems 1 and 2 show that if $PT_{1\sim n}$ (the worst case) satisfies distinct *l*-diversity, probabilistic tables for all other possible cases will also satisfy distinct *l*-diversity.

Theorem 1. If the probabilistic table $PT_{1\sim n}$ that is constructed by the adversary who can obtain access to all *n* published tables satisfies distinct *l*-diversity, the probabilistic table $PT_{1\sim n'}$ that is constructed by the adversary who can only obtain access to *n'* published tables also satisfies distinct *l*-diversity, where $n' < n$. *(Proof skipped due to page limit)*

Theorem 2. If the probabilistic table $PT_{1\sim n}$ that is constructed by the adversary who possesses the values of all *m* quasi-identifier attributes of the victim satisfies distinct *l*-diversity, the probabilistic table $PT'_{1\sim n}$ that is constructed by the adversary who only possesses the values of $m' < m$ quasi-identifier attributes of the victim also satisfies distinct *l*-diversity. *(Proof skipped due to page limit)*

Theorems 1 and 2 imply that no privacy breach will occur for all other possible cases if the worst case does not cause privacy breach.

Next, we consider published table T_n. The reason for data holder DH_n to construct $PT_{1\sim n}$ is to check whether privacy breaches will occur for releasing published table T_n. We say that no privacy breaches will occur for releasing published table T_n if probabilistic table $PT_{1\sim n}$ satisfy distinct *l*-diversity. It is because according to Definition 3 and Theorems 1 and 2, if $PT_{1\sim n}$ satisfies distinct *l*-diversity, no privacy breaches will occur for the case that an adversary can obtain access to all/some published tables, which may include T_n. Hence, no privacy breaches will occur for releasing published table T_n.

The relationship among T_n, $PT_{1\sim n}$, and (non-centralized) distinct *l*-diversity is formally defined as follows.

Definition 4. (Non-centralized distinct *l*-diversity for T_n) If probabilistic table $PT_{1\sim n}$ satisfies distinct *l*-diversity, we say that T_n satisfies non-centralized distinct *l*-diversity.

Note that in order for T_n to satisfy non-centralized distinct l-diversity with $l = l'$, previous published tables T_1, ..., T_{n-1} have to also satisfy non-centralized distinct l-diversity with $l \geq l'$. Otherwise, it is possible that T_n cannot satisfy non-centralized distinct l-diversity with $l = l'$ no matter how T_n is generalized.

Adopting distinct l-diversity [2] to the problem of non-centralized data publishing allow us to propose an efficient algorithm to generate a published table that satisfies non-centralized distinct l-diversity. It is explained as follows. Firstly, as proved in Theorems 1 and 2, we only need to ensure that the worst case will not cause privacy breaches instead of checking all possible cases. Secondly, non-centralized distinct l-diversity satisfies the generalization property and the subset property, which are necessary to apply Incognito [13] to generate published tables correctly and efficiently (see Theorems 3 and 4).

5.2. Algorithm

Intuitively, a data holder, say DH_n, can use the following steps (Steps 1 to 3) to generate a published table T_n that satisfies non-centralized distinct l-diversity (later we will show that Step 2 can be skipped).

- Step 1: Collect T_1, ..., and T_{n-1}.
- Step 2: Construct a probabilistic table $PT_{1\sim(n-1)}$ from T_1, ..., and T_{n-1}.
- Step 3: Generate T_n from Pri_n such that $PT_{1\sim n}$ satisfies non-centralized distinct l-diversity.

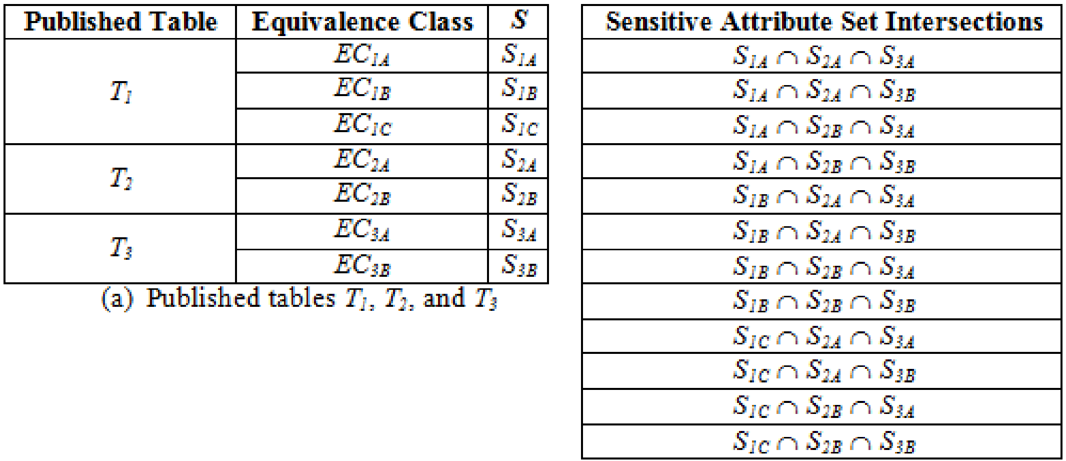
Step 1 is to collect the related published tables that have been already released by other data holders. Step 2 is to construct a probabilistic table $PT_{1\sim(n-1)}$ from the collected published tables. Step 3 is to generate T_n by the modified version of Incognito [13], so that $PT_{1\sim n}$, which is constructed by combining T_n and $PT_{1\sim(n-1)}$, satisfies non-centralized distinct l-diversity.

Nevertheless, we figure out how to verify whether $PT_{1\sim n}$ satisfies non-centralized distinct l-diversity without actually constructing $PT_{1\sim n}$. Hence, we do not need Step 2 to construct $PT_{1\sim(n-1)}$ neither. To explain how to do so, we first describe the steps (Steps A to C) to construct $PT_{1\sim n}$ from $PT_{1\sim(n-1)}$ and T_n.

- Step A: The first step is to generate all equivalence classes of $PT_{1\sim n}$. Recall that an equivalence class is a collection of tuples with the same quasi-identifier attribute values. Hence, we can use a set of quasi-identifier attribute values to be the name of an equivalence class. The name of an equivalence class in $PT_{1\sim n}$ can be obtained by taking the union of the name of an equivalence class in $PT_{1\sim(n-1)}$ and that in T_n. For example, let X, Y, and Z be $\{A_1 \geq v_1, ..., A_{m'} \geq v_{m'}\}$ in $PT_{1\sim(n-1)}$, $\{A_{m'+1} \geq v_{m'+1}, ..., A_m \geq v_m\}$ in T_n, and $\{A_1 \geq v_1, ..., A_m \geq v_m\}$ in $PT_{1\sim n}$ respectively. Z is obtained by taking the union of X and Y. The names of all equivalence classes in $PT_{1\sim n}$ are all possible combinations of the equivalence class names in $PT_{1\sim(n-1)}$ and those in T_n. In other words, the number of equivalence classes in $PT_{1\sim n}$ equals to the number of equivalence classes in $PT_{1\sim(n-1)}$ multiplies the number of equivalence classes in T_n.
- Step B: The second step is to generate sensitive attribute values with non-zero conditional probabilities for all equivalence classes in $PT_{1\sim n}$. Let S_X, S_Y, and S_Z be the set of S values with non-zero conditional probabilities in X, Y, and Z respectively. In Equation (5), if all the terms on the right hand side are non-zero, the resulting conditional probability on the left hand side is non-zero. If any term on the right hand side is zero, the resulting conditional probability on the left hand side is zero. Hence, if Z is obtained by taking the union of X and Y, we have $S_Z = S_X \cap S_Y$. For example, if S_X = {“High”, “Middle”} and S_Y = {“Middle”, “Low”}, we have S_Z = {“Middle”}.
- Step C: The third step is to apply Equation (5) to calculate the non-zero conditional probabilities for each sensitive attribute value generated in the Step B.

For non-centralized distinct *l*-diversity, we are only interested in the number of different sensitive attribute values with non-zero probabilities in each equivalence classes. We do not care about the names of equivalence classes (obtained by Step A) and the actual values for the non-zero conditional probabilities (obtained by Step C). Hence, to verify whether $PT_{1\sim n}$ satisfies non-centralized distinct *l*-diversity, we can skip Step A and Step C and only perform Step B, which becomes the requirement checking step of the modified version of Incognito in Step 3. The modified checking step counts the number of elements in each intersection of every possible combinations among the set of S values with non-zero conditional probabilities for every equivalence class in T_n and that in $PT_{1\sim(n-1)}$, and checks if all such numbers are not less than l. To know the set of S values in the equivalence classes in $PT_{1\sim(n-1)}$, we only need to refer to T_1, ..., T_{n-1}. In other words, there is no need to explicitly construct any probabilistic tables, including $PT_{1\sim(n-1)}$ and $PT_{1\sim n}$. We only need to count the number of elements in each intersection of every possible combinations among the set of S values with non-zero conditional probabilities for every equivalence class in T_1, ... , T_n, and deduce that $PT_{1\sim n}$ satisfy non-centralized *l*-diversity if and only if all such numbers are not less than l. For example, suppose two published tables T_1 and T_2 have been published by two data holders and the third data holder is now generating T_3. Assume that T_1 has three equivalence classes, each of which has the sensitive attribute sets S_{1A}, S_{1B}, and S_{1C}; T_2 has two equivalence classes, each of which has the sensitive attribute sets S_{2A} and S_{2B}. If T_3 has two equivalence classes, each of which has the sensitive attribute sets S_{3A} and S_{3B}. An example is shown in Table 8. Suppose two published tables, namely T_1 and T_2, are published by two data holders. Both published tables have the same sensitive attribute S with the same domain. Further suppose that the third data holder wants to publish a published table, say T_3, with the same S sensitive value with the same domain. The third data holder has to consider 12 sensitive attribute set intersections as shown in Table 8(b). If the size of every intersection is not less than l, releasing T_3 satisfies non-centralized *l*-diversity.

Table 8. The sensitive attribute set intersections that are considered for releasing T_3 after T_1 and T_2 are published.



Published Table	Equivalence Class	S
T_1	EC_{1A}	S_{1A}
	EC_{1B}	S_{1B}
	EC_{1C}	S_{1C}
T_2	EC_{2A}	S_{2A}
	EC_{2B}	S_{2B}
T_3	EC_{3A}	S_{3A}
	EC_{3B}	S_{3B}

(a) Published tables T_1, T_2, and T_3

Sensitive Attribute Set Intersections
$S_{1A} \cap S_{2A} \cap S_{3A}$
$S_{1A} \cap S_{2A} \cap S_{3B}$
$S_{1A} \cap S_{2B} \cap S_{3A}$
$S_{1A} \cap S_{2B} \cap S_{3B}$
$S_{1B} \cap S_{2A} \cap S_{3A}$
$S_{1B} \cap S_{2A} \cap S_{3B}$
$S_{1B} \cap S_{2B} \cap S_{3A}$
$S_{1B} \cap S_{2B} \cap S_{3B}$
$S_{1C} \cap S_{2A} \cap S_{3A}$
$S_{1C} \cap S_{2A} \cap S_{3B}$
$S_{1C} \cap S_{2B} \cap S_{3A}$
$S_{1C} \cap S_{2B} \cap S_{3B}$

(b) The sensitive attribute set intersections that are considered

Incognito is still efficient with the aforesaid modification. It is because the modified requirement checking step, which is the only step that we modify on Incognito, is still working on equivalence classes and is not computationally intensive.

5.3. Theorems

In this subsection, we prove the correctness of applying the modified version of Incognito to the non-centralized data publishing problem.

To apply Incognito, the privacy preserving requirement has to satisfy the generalization property and the subset property [13]. Generalization property means that suppose T_A and T_B are generalized from a private table *PriT* and T_B is more general than T_A; if T_A satisfies a privacy preserving requirement, T_B also satisfies the same requirement. Subset property means that suppose T_C is generalized from a private table *PriT* and T_D is obtained by removing some quasi-identifier attributes of T_C; if T_C satisfies a privacy preserving requirement, T_D will also satisfy the same requirement. Theorems 3 and 4 show that non-centralized distinct *l*-diversity satisfies the generalization property and the subset property.

Theorem 3. Non-centralized distinct *l*-diversity satisfies the generalization property. *(Proof skipped due to page limit)*

Theorem 4. Non-centralized distinct *l*-diversity satisfies the subset property. *(Proof skipped due to page limit)*

Theorems 3 and 4 imply that we can apply the modified version of Incognito to generate published table T_n such that when T_n is considered with $PT_{1\sim(n-1)}$ together to construct $PT_{1\sim n}$, $PT_{1\sim n}$ satisfies distinct *l*-diversity. By Definition 4, we say that T_n satisfies non-centralized distinct *l*-diversity and releasing T_n will not cause privacy breach even if it is considered with other published tables T_1, ..., and T_{n-1}.

6. EXPERIMENTS

In this section, we present the experiment results. The objective of our experiment is to evaluate whether our non-centralized algorithm is feasible to real applications. The algorithm is feasible if it can generate a published table in a reasonable time from real databases, where each of them contains more quasi-identifier (QI) attributes and a large number of tuples. Section 6.1 presents the settings of the experiment. Section 6.2 presents the metrics measured in the experiment. Section 6.3 presents and discusses the experiment results.

Table 9. Schema of *SAL*.

Attribute Type	Attribute	Range	Number of Possible Values
QI Attributes	Age	16 – 94	77
	Education	1 – 17	14
	Birthplace	1 – 710	133
	Occupation	1 – 983	509
	Race	1 – 9	9
	Work class	2 – 10	7
	Marital status	1 – 6	6
Sensitive Attribute	Income	0 – 49	50

6.1. Settings

All experiments were performed on a computer with Intel Core 2 Duo 2.80GHz CPU and 3.2GB RAM running Windows XP and IBM DB2 Express-C. We modify Incognito [13] to become our non-centralized algorithm. To simulate a non-centralized data publishing environment, we divide a real (centralized) database *SAL* (from *http://ipums.org*) into a number of (non-centralized) private tables. *SAL* contains 700k tuples, each of which stores information about an American adult. Table 9 shows the schema of *SAL* that we use, which has seven QI attributes: age, education, birthplace, occupation, race, work class, and marital status; and a sensitive attribute: income. Each possible value of an attribute is represented by an integer. Table 9 shows the range and the number of possible values for each attribute.

Similar to Incognito, our non-centralized algorithm requires users to define generalization levels for each QI attribute to indicate how to generalize the QI attribute in a published table. In our experiments, we define four generalization levels for each QI attribute: level-0 represents no generalization, while level-1, level-2, and level-3 represent the generalization that results in four, two, and one equivalence classes respectively.

The four parameters for our experiments are l, $|QI|$, l_{pre}, and $|QI_{pre}|$. They are explained as follows.

- l is the l value for the distinct l-diversity requirement, i.e. the number of distinct sensitive values for each equivalence class has to be larger than or equal to l.
- $|QI|$ is the number of QI attributes for the current published table.
- l_{pre} is the distinct l-diversity requirement for the previous published table (if there are multiple previous published tables, here refer to the latest one). Note that it is not necessary for l_{pre} to be equal to the minimum number of distinct sensitive values for all equivalence classes in the (latest) previous published table.
- $|QI_{pre}|$ is the number of QI attributes in the published tables that have been released by other data holders already (i.e. in the previous published tables) before generating the current published table. If $|QI_{pre}| = 0$, it means that the published table is generated when there is no previous published tables. In this case, we can use distinct l-diversity [2] directly. If $|QI_{pre}| > 0$, it means that the published table is generated when there are some previous published tables. In this case, our non-centralized algorithm is used.

Note that we do not need to set the number of previous published tables as a parameter in our experiments. It is because such a number does not directly affect the probabilistic table. Instead, we can tune the $|QI_{pre}|$ parameter to control the effect of the previous published tables in the experiments.

6.2. Metrics

We measure the following three metrics in our experiment.

1. Elapsed time for generating the published table given a probabilistic table and a private table.

2. The number of published tables that satisfy the requirement. We call these published tables published table candidates. The published table candidates differ in the generalization levels for each QI attribute.

3. The minimum average generalization level. For each published table, there are a number of QI attributes, and each QI attributes are generalized according to the generalization level. Hence, we can calculate the average generalization level for each published table. Similar to Incognito, our non-centralized algorithm outputs a number of published tables. A user only needs to select one published table among all the published table candidates to release. Usually the one that is least generalized (with the lowest average generalization level) is chosen in order to have the highest data quality to the data miner. Therefore, we are interested in the minimum average generalization level among all these published tables.

Table 10. Parameters for the four experiment sets.

Experiment Set	Parameter	Values
A	l_{pre}	30
	$\lvert QI_{pre}\rvert$	2
B	l_{pre}	50
	$\lvert QI_{pre}\rvert$	2
C	l_{pre}	30
	$\lvert QI_{pre}\rvert$	4
D	l_{pre}	50
	$\lvert QI_{pre}\rvert$	4

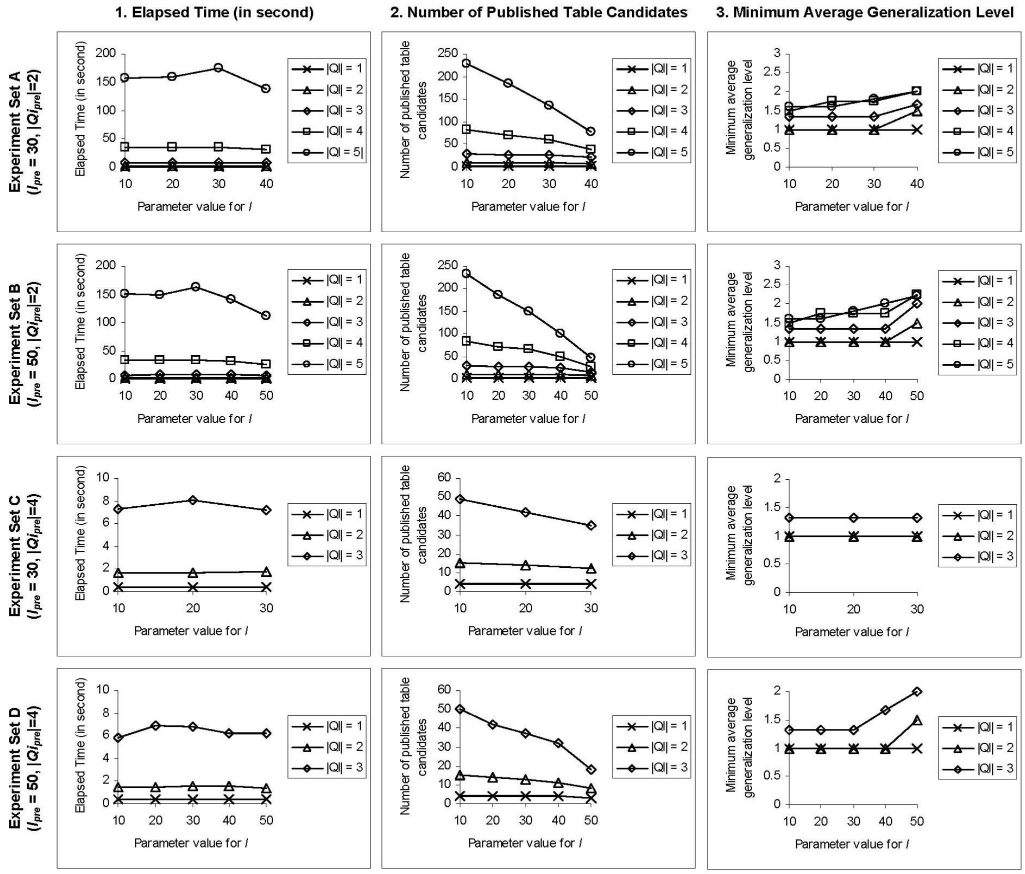


Figure 1. Experimental results.

6.3. Results and Discussions

We performed four sets of experiment. Their parameters are shown in Table 10. Figure 1 shows the experimental results. There are three observations about elapsed time based on the experimental results.

1. The first observation is that our non-centralized algorithm takes less than 180s to generate a published table with five QI attributes.
2. The second observation is that increasing the $|QI_{pre}|$ value does not significantly increase the time to generate a published table. It is because the non-centralized algorithm only reads the sensitive attribute values in each equivalence class in the probabilistic table but

not the QI attributes. Although the number equivalence classes increases when the $|QI_{pre}|$ value increases, our algorithm only needs to consider one equivalence class among the equivalence classes with the same set of sensitive attribute values. In addition, the time to read the probabilistic table is negligible when comparing with the core algorithm that generates the published table by building and searching for the published table lattice. Hence, the $|QI_{pre}|$ value does not cause much impact to the total elapsed time to generate a published table.

3. The third observation is that our algorithm takes more time to generate a published table with more QI attributes (i.e. a higher $|QI|$ value), which is reasonable since the lattice size is increased when there are more QI attributes. Therefore, the time needed to build and search the lattice is increased.

The first two observations about elapsed time indicate that our algorithm is feasible for real applications.

There are three observations about the number of published table candidates that satisfy the requirement based on the experimental results. Note that only one of the published table candidates will be chosen as the published table.

1. The first observation is that if the l value increases (i.e. the requirement is more restrictive), fewer published tables will satisfy the requirement. That means the number of published table candidates decreases.
2. The second observation is that if the $|QI|$ value increases, the number of published table candidates increases. It is because there are more combinations for different generalization levels of QI attributes and hence there are more published table candidates.
3. The third observation is that if the l_{pre} value increases, the number of published table candidates increases. It is because releasing a new published table can only make the requirement enforced by previous published tables less restrictive. A more restrictive old requirement allows more rooms for it to be lessen and hence more published table candidates can satisfy the new requirement.

There are two observations about the minimum average generalization level based on the experimental results.

1. The first observation is that if the l value increases, then the minimum average generalization level increases. It is because a more generalized published table is needed for a more restrictive requirement.
2. The second observation is that if the $|QI|$ value increases, then the minimum average generalization level increases. It is because more QI attributes will cause the equivalence classes to have less number of tuples for the same generalization levels. Hence the generalization levels have to be increased in order to satisfy the same requirement.

7. CONCLUSIONS

This paper studied the problem of data publishing on multiple non-centralized datasets, which store information about the same group of individuals and share a common sensitive attribute with the same domain. We have shown that such a problem is different from data publishing on a centralized dataset and serial data publishing on a dynamic/static dataset in the sense that there is no dataset that stores all information for non-centralized data publishing. To solve the problem, we first propose a statistical approach, which does not rely on any communications between the data holders, to estimate the information that can be deduced by an adversary who can obtain access to all non-centralized published tables. Then, we propose how to adopt distinct l-diversity [2] to non-centralized data publishing. After that, we propose an algorithm to generate published tables that satisfy non-centralized distinct l-diversity. Our algorithm is modified from Incognito [13], which is an efficient and widely-used algorithm for (centralized) data publishing. Finally, theorems and experiments are presented to show that the proposed algorithm is correct and feasible to real applications.

REFERENCES

[1] L. Sweeney, (2002) "Achieving *k*-anonymity privacy protection using generalization and suppression", International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, Vol.10, No.5, pp.571-588.

[2] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, (2006) "*l*-diversity: privacy beyond *k*-anonymity", in proceedings of *ICDE*.

[3] N. Li, T. Li, and S. Venkatasubramanian, (2007) "*t*-closeness: privacy beyond *k*-anonymity and *l*-diversity", in proceedings of *ICDE*, pp.106-115.

[4] J. Byun, Y. Sohn, E. Bertino, and N. Li, (2006) "Secure anonymization for incremental datasets", in proceedings of *SDM*.

[5] B.C.M. Fung, K. Wang, A. Fu, and J. Pei, (2008) "Anonymity of continuous data publishing", in proceedings of *EDBT*.

[6] X. Xiao and Y. Tao, (2007) "*m*-invariance: towards privacy preserving re-publication of dynamic datasets", in proceedings of *SIGMOD*.

[7] Y. Bu, A. Fu, R. Wong, L. Chen, and J. Li, (2008) "Privacy preserving serial data publishing by role composition", in proceedings of *VLDB*.

[8] R.C.W. Wong, A. Fu, J. Liu, K. Wang, and Y. Xu, (2010) "Global Privacy Guarantee in Serial Data Publishing", in proceedings of *ICDE*.

[9] K. Wang and B.C.M. Fung, (2006) "Anonymizing sequential releases", in proceedings of *KDD*.

[10] P. Andritsos, A. Fuxman, and R. J. Miller, (2006) "Clean answers over dirty databases: A probabilistic approach", in proceedings of *ICDE*.

[11] N. Dalvi and D. Suciu, (2004) "Efficient query evaluation on probabilistic databases", in proceedings of *VLDB*.

[12] R. Cheng, S. Singh, and S. Prabhakar, (2005) "U-DBMS: A database system for managing constantly-evolving data", in proceedings of *VLDB*.

[13] K. LeFevre, D. Dewitt, and R. Ramakrishnan, (2005) "Incognito: efficient fulldomain k-anonymity", in proceedings of *SIGMOD*.

[14] R. Agrawal, R. Srikant, and D. Thomas, (2005) "Privacy preserving OLAP", in proceedings of *SIGMOD*.

[15] S. Zhong, Z. Yang, and R. N. Wright, (2005) "Privacy-enhancing *k*-anonymization of customer data", in proceedings of PODS.

[16] R.J. Bayardo and R. Agrawal, (2005) "Data privacy through optimal *k*-anonymization", in proceedings of *ICDE*.

[17] C. Wang, S. Parthasarathy, and R. Jin, (2006) "A decomposition-based probabilistic framework for estimating the selectivity of XML twig queries", in proceedings of *EDBT*.

[18] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum, (2004) "Probabilistic ranking of database query results", in proceedings of *VLDB*.

[19] V. Markl, N. Megiddo, M. Kutsch, T.M. Tran, P. Haas, and U. Srivastava, (2005) "Consistently estimating the selectivity of conjuncts of prediciates", in proceedings of *VLDB*.

[20] A. Darwiche, (1997) "A logical notion of conditional independence: properties and applications", *Artificial Intelligence*, vol. 97, pp. 45-82.

[21] M.G. Walker and G. Wiederhold, (1990) "Acquisition and validation of knowledge from data", *Intelligent Systems*, chapter 17, pp 415-428.

[22] G. Shafer, (1998) "Advances in the understanding and use of conditional independence", *Annals of Mathematics and Artificial Intelligence,* 21.

[23] B.P. Carlin and T.A. Louis, (2008) *Bayesian Methods for Data Analysis*, Chapman and Hall/CRC.

[24] R. Agrawal and R. Srikant, (2000) "Privacy preserving data mining", in proceedings of *SIGMOD*.

[25] S. Vijayarani and A. Tamilarasi, (2010) "Bit Transformation Perturbative Masking Technique for Protecting Sensitive Information In Privacy Preserving Data Mining", *International Journal of Database Management Systems (IJDMS).*

[26] A. Sharma[1] and V. Ojha, (2010) "Implementation of Cryptography for Privacy Preserving Data Mining", *International Journal of Database Management Systems (IJDMS).*