Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 464731, 14 pages
http://dx.doi.org/10.1155/2015/464731

Research Article

Privacy Protection Method for Multiple Sensitive Attributes Based on Strong Rule

Tong Yi and Minyong Shi

College of Computer Science, Communication University of China, Beijing 100024, China

Correspondence should be addressed to Tong Yi; yitongt@cuc.edu.cn

Received 14 April 2015; Revised 10 July 2015; Accepted 15 July 2015

Academic Editor: Nazrul Islam

At present, most studies on data publishing only considered single sensitive attribute, and the works on multiple sensitive attributes are still few. And almost all the existing studies on multiple sensitive attributes had not taken the inherent relationship between sensitive attributes into account, so that adversary can use the background knowledge about this relationship to attack the privacy of users. This paper presents an attack model with the association rules between the sensitive attributes and, accordingly, presents a data publication for multiple sensitive attributes. Through proof and analysis, the new model can prevent adversary from using the background knowledge about association rules to attack privacy, and it is able to get high-quality released information. At last, this paper verifies the above conclusion with experiments.

1. Introduction

Data publishing is widely used in the field of information sharing and scientific research, and how to ensure the availability of data and the security of user's privacy is the core content of studies. The data tables usually contain three types of attribute: identifier, which can identify the individual uniquely, for example, the social security number (SSN); quasi-identifier (QI), which cannot identify the individual uniquely but can provide individual information, for example, the country and age attributes; sensitive attribute (S), which is usually related to the privacy of users, for example, the disease attribute. The sensitive attribute needs to be protected in the published table. A series of data publishing methods [1–11] are presented, in order to prevent adversary from linking quasi-identifying attributes with public available dataset to reveal personal identity, and paper [1, 2] presents k-anonymity method, which partitions the table into equivalence groups (EG). Each equivalence group consists of at least k different records, and k-anonymity generalizes the quasi-identified attributes of records in the same equivalence groups. But the k-anonymity is faced with the risk of sensitive attribute disclosure due to lack of diversity. In order to solve this problem, [4] proposals L-diversity, which not only can satisfy the k-anonymity but also requires that there are at least L different sensitive attribute values in each equivalence group. In addition, privacy protection methods in [5–11] also improve the k-anonymity from different angles, respectively. But most of them only consider the situation of single sensitive attribute, so some privacy protection methods for multiple sensitive attributes are presented. Papers [12, 13] attempt to directly use L-diversity for multiple sensitive attributes, which result in a lot of information loss. Paper [14] protects users' privacy through disturbing the order of sensitive attributes values in the same equivalence groups. But this method needs to add fake sensitive attribute values to EG, and it breaks the relationship between sensitive attributes, so useful relationships cannot be provided. The publication method in paper [15] can prevent adversary using nonmembership knowledge to attack data table, but its strict grouping condition will result in excessive information loss. According to the theory of paper [16], we can know that the publication methods of [12, 13, 15] cannot ensure good diversity and are vulnerable to background-join attack, so paper [16] divides the raw data table into several projected tables, puts the sensitive attributes which have strong dependency into the same projected table, and makes each projected table satisfy t-closeness at last. But this method ignores

the association rules with high confidence, and the adversary can use the knowledge of these rules to get the privacy of users. In order to avoid the suppression of records, paper [17] presents a new publication method, which chooses to generalize each sensitive attribute, respectively. But like other privacy protection methods for multiple sensitive attributes, paper [17] ignores the inherent relationship between sensitive attributes, so the adversary can use the related background knowledge to attack privacy, and it is difficult for users to find valuable relationships from its released tables. In order to resolve this problem, this paper introduces the association rule into the design of privacy protection method and presents an improved data publishing model for multiple sensitive attributes based on the work of [17].

2. The Main Work of This Paper

Most existing researches on privacy preserving technology for multiple sensitive attributes have not taken the inherent relationship between sensitive attributes into account, so adversary sometimes can use the related background knowledge to attack the privacy of users, and some valuable relationships cannot be provided by released tables. Faced with this situation, we introduce the association rules into the research on data publishing. The main works of this paper are as follows.

(1) It analyses the data publishing model-Rating in paper [17], points out its weakness, and presents an attack method with strong rule (Section 3).

(2) It takes relationship between different sensitive attributes into account, presents a mixed data publishing model based on Rating, then improves the algorithm of Rating, makes it more effective, and at last analyses and proves the correctness of the algorithm and the security of the mixed data publication model (Section 4).

(3) It proves that the new model has better quality of released information than Rating in theory (Section 5).

(4) Through the experiments, it verifies that the new data publishing model can provide better privacy, and it is able to preserve valuable relationships between sensitive attributes in released tables (Section 6).

3. The Analysis of Rating

This section will introduce Rating [17] model briefly and present an attack method which can use strong rules to get users' privacy from Rating.

3.1. Description of Symbol. Table T contains n records, record $t = (\mathrm{QI}_1, \mathrm{QI}_2, \ldots, \mathrm{QI}_i, S_1, S_2, \ldots, S_j)$, $\mathrm{QI}_1, \mathrm{QI}_2, \ldots, \mathrm{QI}_i$ are i quasi-identifiers of T, and $S_1, S_2, \ldots, S_j$ are j sensitive attributes of T. $t.\mathrm{QI}_m$ $(1 \le m \le i)$ represents the value of t in quasi-attribute QI_m. Similarly, $t.S_m$ $(1 \le m \le j)$ represents the value of t in sensitive attribute S_m.

Table 1: Original table.

ID	Age	Country	S_1	S_2	S_3
1	23	China	a_1	b_1	c_2
2	35	India	a_4	b_2	c_1
3	29	Mexico	a_5	b_2	c_2
4	31	Japan	a_6	b_2	c_2
5	38	Mexico	a_1	b_1	c_2
6	23	Japan	a_1	b_1	c_1
7	36	America	a_1	b_2	c_1
8	38	America	a_2	b_5	c_2
9	21	Australia	a_2	b_2	c_1
10	36	Britain	a_2	b_3	c_1
11	22	China	a_2	b_4	c_2
12	25	India	a_4	b_3	c_2
13	23	Korea	a_5	b_4	c_1
14	33	Canada	a_6	b_1	c_1
15	36	Australia	a_3	b_2	c_2
16	38	Britain	a_3	b_5	c_1

Table 2: Rating published IDT for Table 1.

ID_j	S_1	S_2	S_3
ID_1	(a_1, a_2)	(b_1, b_2)	(c_2, c_1)
ID_2	(a_1, a_2)	(b_1, b_2)	(c_2, c_1)
ID_3	(a_1, a_2)	(b_1, b_2)	(c_2, c_1)
ID_4	(a_4, a_5)	(b_2, b_3)	(c_2, c_1)
ID_5	(a_3, a_6)	(b_2, b_5)	(c_2, c_1)
ID_6	(a_6, a_5)	(b_4, b_3)	(c_2, c_1)
ID_7	(a_3, a_2)	(b_4, b_1)	(c_2, c_1)
ID_8	(a_1, a_4)	(b_5, b_2)	(c_2, c_1)

Definition 1 (generalization). Assume S is a sensitive attribute of table T, $t.S \in Q$, Q is a subset of S in table T, and generalization means using abstract value Q to replace specific value $t.S$. For example, in Table 1, $t_1.S_1 = a_1$, we can use (a_1, a_2) to replace $t_1.S_1$ in Table 3.

Definition 2 (L-diversity). L is the parameter input by users. After generalization, for each sensitive S, if $\forall v \in t.S$ satisfies $\mathrm{num}(v)/|t.S| \le 1/L$, record t satisfies L-diversity. Here $\mathrm{num}(v)$ represents the number of v in $t.S$, and $|t.S|$ represents the number of values in set $t.S$. If all records of model satisfy L-diversity, the model satisfies L-diversity.

3.2. Review of Rating. Rating generalizes each sensitive attribute, respectively, and can improve the quality of released information. Assume Table 1 is the original data table, S_1, S_2, S_3 are sensitive attributes, and age and country are quasi-identifiers. $L = 2$, Rating generates ID table (IDT) first, then uses the values of ID table to generalize the original data table, gets attribute table (AT), and at last releases IDT and AT. Tables 2 and 3 are IDT and AT, respectively, and both of them are released tables of Table 1.

Table 3: Rating published AT for Table 1.

ID	Age	Country	S_1	S_2	S_3
1	23	China	(a_1, a_2)	(b_1, b_2)	(c_2, c_1)
2	35	India	(a_4, a_5)	(b_1, b_2)	(c_2, c_1)
3	29	Mexico	(a_4, a_5)	(b_1, b_2)	(c_2, c_1)
4	31	Japan	(a_3, a_6)	(b_1, b_2)	(c_2, c_1)
5	38	Mexico	(a_1, a_2)	(b_1, b_2)	(c_2, c_1)
6	23	Japan	(a_1, a_2)	(b_1, b_2)	(c_2, c_1)
7	36	America	(a_1, a_4)	(b_2, b_3)	(c_2, c_1)
8	38	America	(a_1, a_2)	(b_2, b_5)	(c_2, c_1)
9	21	Australia	(a_1, a_2)	(b_2, b_5)	(c_2, c_1)
10	36	Britain	(a_1, a_2)	(b_2, b_3)	(c_2, c_1)
11	22	China	(a_3, a_2)	(b_4, b_3)	(c_2, c_1)
12	25	India	(a_1, a_4)	(b_4, b_3)	(c_2, c_1)
13	23	Korea	(a_6, a_5)	(b_4, b_1)	(c_2, c_1)
14	33	Canada	(a_6, a_5)	(b_4, b_1)	(c_2, c_1)
15	36	Australia	(a_3, a_6)	(b_5, b_2)	(c_2, c_1)
16	38	Britain	(a_3, a_2)	(b_5, b_2)	(c_2, c_1)

$S_i\mathrm{ID}_j$ $(1 \le i \le 3, 1 \le j \le 8)$ is a subset of sensitive attribute S_i, it satisfies that $\bigcup_{j=1}^{8} S_i\mathrm{ID}_j = \mathrm{domain}(S_i)$ $(1 \le i \le 3)$, $\mathrm{domain}(S_i)$ refers to the set of all the S_i values in original table. After getting the IDT, we use the values of ID table to generalize original data table. If $t.S_i = v$ and $v \in S_i\mathrm{ID}_j$ $(1 \le i \le 3, 1 \le j \le 8)$, use $S_i\mathrm{ID}_j$ to replace v in original data table. For example, $t_1.S_1 = a_1$ in Table 1, because $a_1 \in S_1\mathrm{ID}_1$ $(S_1\mathrm{ID}_1 = (a_1, a_2))$, and use (a_1, a_2) to replace $t_1.S_1$ in original table, so $t_1.S_1 = (a_1, a_2)$ in Table 3. AT displays the name of $S_i\mathrm{ID}_j$ in [17], and users can use the $S_i\mathrm{ID}_j$ name to get the corresponding $S_i\mathrm{ID}_j$ value in IDT. For convenient description, AT displays the $S_i\mathrm{ID}_j$ value directly in this paper.

3.3. The Weakness of Rating. Rating takes the generalization strategy which is suitable for multiple sensitive attributes and can improve the quality of released information. But Rating ignores the relationship between different sensitive attributes, and sometimes the users' privacy disclosure may happen.

Compared with data publishing for single sensitive attribute, multiple sensitive attributes not only mean the increase of the sensitive attributes' number but also need more effective method to decrease the information loss, and adversary may use the relationship between sensitive attributes to attack privacy of user. So when designing the data publishing model for multiple sensitive attributes, the association rules should be taken into account. An attack method will be presented as follows.

Assuming the original table can basically reflect the real world, if Bob knows his neighbor Alice is in Table 3 and Alice is 23 years old and Chinese, Bob is sure that t_1 is Alice according to the quasi-identifier. Through Bob's common sense of life or previous investigations, he knows that if event a_1 happens to someone, the probability of occurrence of b_1 is usually not less than 75%; namely, $P(b_1 \mid a_1) \ge 75\%$. Then through the ID table Bob knows in this table that there are four people whose attribute S_1 values are a_1, so in these four people, there are at least three whose attribute S_2 values must be b_1. That is to say, there are at least three people whose S_1 values are a_1 while their S_2 values are b_1.

So Bob begins to analyze attribute table, and he finds out that there are 8 records whose S_1 value may be a_1, and these 8 records are $t_1, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{12}$, but in these records, only 3 records' S_2 values may be b_1, and these three records are t_1, t_5, t_6, respectively. Then Bob can be sure that $t_1.S_1 = t_5.S_1 = t_6.S_1 = a_1$, $t_1.S_2 = t_5.S_2 = t_6.S_2 = b_1$. And t_1 is Alice, so the privacy of Alice is disclosed.

The above is an example of attack with association rules. Because Rating has not taken the correlation between sensitive attributes into account, if adversary masters corresponding background knowledge, the privacy of user may be disclosed. So in the next section, an improved model will be presented.

4. The Data Publishing for Multiple Sensitive Attributes Based on Strong Rules

In this section, we introduce the association rules and present a new data publishing model which can avoid attacking with strong rules between sensitive attributes. The records are divided into two categories; each category will be processed by different data publishing models, respectively. So this new data publishing model for multiple sensitive attributes is actually a mixed model.

4.1. Data Publishing Method Based on Sensitive Attributes Clustering (SAC). The original data table T will be divided into two tables: table SAC and table IR, process the two tables, respectively. This part introduces the division of table T and the processing of table SAC. We will introduce some definitions and parameters first.

Definition 3 (association rules). Assume that when event A happens, the probability of occurrence of event B is s $(0 \le s \le 1)$; namely, $A \Rightarrow B : s$. $A \Rightarrow B$ is an association rule.

Definition 4 (support degree). It represents the number of occurrences of an event. For example, in Table 1, the number of a_1 is 4, so the support degree of a_1 is 4. Namely, $\mathrm{support}(a_1) = 4$. $\mathrm{support}(a_1, b_1)$ especially represents the simultaneity number of a_1 and b_1. In Table 1, there are 3 records satisfying $t.S_1 = a_1, t.S_2 = b_1$, so $\mathrm{support}(a_1, b_1) = 3$.

Definition 5 (confidence). Assume when A happens, the probability of occurrence of event B is s $(0 \le s \le 1)$, so the confidence of $A \Rightarrow B$ is s. For example, in Table 1, there are 4 records whose S_1 values are a_1, and three of them have S_2 value b_1. So when a_1 appears, the probability of occurrence of b_1 is 3/4. Namely, $\mathrm{confidence}(a_1 \Rightarrow b_1) = 3/4$. It is not difficult to prove that $\mathrm{confidence}(a_1 \Rightarrow b_1) = \mathrm{support}(a_1, b_1)/\mathrm{support}(a_1)$.

Usually we set minimum support degree threshold (min_support) and the minimum confidence threshold (min_confidence); if an association rule satisfies both these two thresholds, the rule is meaningful. In this paper, as long

as one sensitive attribute value v appears once, adversary may use it to attack the privacy of users, so the min_support is set to 1. And the min_confidence is set by users.

Definition 6 (strong rule). Assuming association rule $A \Rightarrow B$ [confidence = s $(0 \le s \le 1)$], $s \ge$ min_confidence, so we call $A \Rightarrow B$ strong rule.

Usually the strong rule's confidence is relatively higher, and adversary may use the strong rule to attack users' privacy if adversary has the related background knowledge. On the other hand, we hope to preserve information of strong rule in the released data, because it is valuable. So we need to put the records containing strong rules into table SAC and process table SAC with consideration for strong rule. Record containing strong rules means that, for record t, if $\exists t.S_x, t.S_y$ $(1 \le x, y \le j)$, $t.S_x \Rightarrow t.S_y$ is a strong rule, so record t contains strong rule. And for a value v, if there is a strong rule that $A \Rightarrow B$ satisfies $v = A$ or $v = B$, we call v strong value.

4.1.1. Partition Table. Assuming the association rules in original data table are close to the situation of the real world, first use classic association rule mining algorithm-Apriori [18] to find out all the strong rules in table, according to min_confidence set by user (line 1). Then find a sensitive attribute S that has the largest number of strong values, put all other sensitive attributes into S_set˜, and if a record contains strong value in S_set˜, add it to table SAC (line 2). At last delete the records contained by SAC in original table, and get table IR (line 3), so the original table T is divided into two tables: SAC and IR. Obviously, the records which all contain strong association rules are in SAC, and all the strong values in IR belong to the same sensitive attribute, so there are no probability tilts caused by strong association rules in IR (please see Section 4.2.2).

Algorithm 7 (partition table). Input: original data table T, min_confidence

Output: Table SAC, Table IR

(1) Accoding to min_confidence find out all strong rules with Apriori.

(2) Find the records which all contain strong values in S_set˜, and put them into table SAC.

(3) Table IR = T-Table SAC.

4.1.2. Partition Sensitive Attributes. After partition table, begin to process the table SAC. In order to preserve the information of strong rules, cluster the sensitive attributes. First, we need to define the distance between sensitive attributes.

Definition 8 (distance between sensitive attributes). Given two sensitive attributes S_x, S_y $(1 \le x, y \le j)$, the distance between the two sensitive attributes can be defined as

$$\text{distance}\left(S_x, S_y\right) = \begin{cases} 1, & \text{there are no strong rules between } S_x \text{ and } S_y, \\ 0, & \text{there are strong rules between } S_x \text{ and } S_y. \end{cases} \quad (1)$$

Here, if $\exists v_x \in \text{domain}(S_x)$, $\exists v_y \in \text{domain}(S_y)$, $v_x \Rightarrow v_y$, or $v_y \Rightarrow v_x$ is strong rule, we say there are strong rules between S_x and S_y; else, there are no strong rules between S_x and S_y.

Definition 9 (distance between sensitive attribute and cluster). Assuming C is a cluster, S_x $(1 \le x \le j)$ is a sensitive attribute, and the distance between C and S_x can be defined as

$$\text{distance}\left(S_x, C\right) = \begin{cases} 1, & \forall S_y \in C\left(1 \le y \le j\right), \ \text{distacne}\left(S_x, S_y\right) = 1, \\ 0, & \exists S_y \in C\left(1 \le y \le j\right), \ \text{distacne}\left(S_x, S_y\right) = 0. \end{cases} \quad (2)$$

In this method, put the similar sensitive attributes into a same cluster, as long as the set of sensitive attributes is not empty, and generate new cluster constantly (line 1). For each new empty cluster C, pick a sensitive attribute S_x $(1 \le x \le j)$ from sensitive attribute set S_set orderly, put S_x into C, and S_x is the first attribute of cluster C (line 2). Find out all such sensitive attributes $S_y \in S$_set $(1 \le y \le j)$, and S_y satisfies that distance$(S_y, C) = 0$. Add S_y to C, and delete S_y in S_set (line 3 to line 4). Similarly, generate other clusters.

After Algorithm 10, we get the set of clusters: Cluster_Set = $\{C_1, C_2, \ldots, C_m\}$, $\forall C \in$ Cluster_Set, and C is a subset of sensitive attributes set S_set = $\{S_1, S_2, \ldots, S_j\}$.

Assuming $C \in$ Cluster_Set, $C = \{S_x, S_y, S_z\}$ $(1 \le x, y, z \le j)$, record $t \in$ SAC, and the C value of t is $t.C = (t.S_x, t.S_y, t.S_z)$.

Algorithm 10 (partition sensitive attributes). Input: Table SAC

Output: set of sensitive attributes' cluster (Cluster_Set)

S_set is the set of sensitive attributes, S_set = $\{S_1, S_2, \ldots, S_j\}$.

(1) While the S_set is not empty, repeat (2) to (5).

(2) Generate cluster C, add the first sensitive attribute S_x $(1 \le x \le j)$ to C, S_set - S_x.

(3) For each $S_y \in S$_set $(1 \le y \le j)$, do (4).

(4) If distance$(C, S_y) = 0$, add S_y to C, S_set -S_y.

(5) Cluster_set $\cup$ C.

4.1.3. Partition Records. This part will divide the Table SAC into several groups and anonymize the records in the same group.

Algorithm 11 (partition records). Input: Table SAC, Table IR, and Cluster_Set, L

Output: released Table SAC˜

(1) While Table SAC is not empty, repeat (2) to (7).

(2) Generate a new group G.

(3) Choose a record t from Table SAC orderly, $G \cup t$, SAC-t.

(4) While $|G| < L$, repeat (5) to (6).

(5) If $\exists t^{\sim} \in$ SAC, this satisfies $\forall t \in G$, $1 \le \forall x \le j$, $t^{\sim}.S_x \ne t.S_x$, $G \cup t^{\sim}$, and SAC-$t^{\sim}$.

(6) Else choose record $t^{\sim}$ from table IR orderly, $t^{\sim}$ satisfies $\forall t \in G$, $1 \le \forall x \le j, t^{\sim}.S_x \ne t.S_x$, $G \cup t^{\sim}$, and IR-$t^{\sim}$.

(7) SAC˜ $\cup$ G

(8) Permutate cluster values in each group randomly.

Table 4: The SAC table.

ID	Age	Country	S_1	S_2	S_3
1	23	China	a_1	b_1	c_2
5	38	Mexico	a_1	b_1	c_2
6	23	Japan	a_1	b_1	c_1
7	36	America	a_1	b_2	c_1

In the algorithm of partition records, while table SAC is not empty, generate group constantly (lines 1-2). For each empty group G, choose a record t from table SAC as G's first record (line 3). Choosing $t^{\sim} \in$ SAC, or $t^{\sim} \in$ IR, $t^{\sim}$ does not have the same sensitive attributes values with G, and add $t^{\sim}$ to G, until the number of records in G is not less than L (line 4–6). Within each group, sensitive attributes values are permutated randomly in each cluster to break the linking between different clusters (line 8). That is to say, adjust the position of cluster values randomly. Finally, release SAC$^{\sim}$.

Now take Table 1 as an example, there are three steps on this process. Here, assume that $L = 2$, min_confidence = 0.75.

(1) *Partition table*: both S_1 and S_2 have four strong values, and S_3 has no values, so $S\_set^{\sim} = \{S_1, S_3\}$. We first find out that there are 4 records containing strong values in $S\_set^{\sim}$, and they are t_1, t_5, t_6, and t_7, respectively. These 4 records make up table SAC (Table 4) and meanwhile, delete these 4 records in Table 1.

(2) *Partition sensitive attributes*: generate a new cluster C_1, add S_1 to C_1, and now there remain two sensitive attributes in sensitive attributes set. Because there is strong rule $a_1 \Rightarrow b_1$ between S_1 and S_2 and distance$(C_1, S_2) = 0$, add S_2 to C_1. But both distance(S_1, S_3) and distance(S_2, S_3) are 1, distance$(C_1, S_3) = 1$, S_3 cannot be added to C_1. And the only one attribute S_3 in sensitive attributes set makes up a cluster C_2 alone. So clustering is over, and we get two clusters $C_1 = \{S_1, S_2\}, C_2 = \{S_3\}$.

(3) *Partition records*: according to the grouping condition, it cannot have the same sensitive attribute values in a group, t_1 and t_2 make up a group, similarly, and t_5 and t_9, t_6 and t_4, t_7 and t_8 make up groups, respectively (Table 5). After grouping, randomly permutate the cluster values in the same groups and release the table SAC$^{\sim}$ (Table 6). For example, in group 1, permutate C_1 value (a_1, b_1), (a_4, b_2) randomly. Here, each group has two records, according to the random principle; after disturbing order, (a_1, b_1) may swap position with (a_4, b_2), or both (a_1, b_1) and (a_4, b_2) remain in the original positions. Similarly, for C_2, permutate C_2 value, c_1, c_2, randomly. Although through anonymity, the relationship between S_1 and S_2 is still preserved. On the other hand, linking between C_1 and C_2 has been broken. So this method preserves the links between sensitive attributes in the same clusters and breaks the links between sensitive attributes from different clusters.

Table 5: Partition records for Table 4.

ID	Age	Country	S_1	S_2	S_3	Group ID
1	23	China	a_1	b_1	c_2	1
2	35	India	a_4	b_2	c_1	1
5	38	Mexico	a_1	b_1	c_2	2
9	21	Australia	a_2	b_2	c_1	2
6	23	Japan	a_1	b_1	c_1	3
4	31	Japan	a_6	b_2	c_2	3
7	36	America	a_1	b_2	c_1	4
8	38	America	a_2	b_5	c_2	4

Table 6: Mixed model published SAC$^{\sim}$ for Table 1.

ID	Age	Country	S_1	S_2	S_3	Group ID
1	23	China	a_4	b_2	c_2	1
2	35	India	a_1	b_1	c_1	1
5	38	Mexico	a_1	b_1	c_1	2
9	21	Australia	a_2	b_2	c_2	2
6	23	Japan	a_6	b_2	c_2	3
4	31	Japan	a_1	b_1	c_1	3
7	36	America	a_2	b_5	c_1	4
8	38	America	a_1	b_2	c_2	4

Lemma 12. *Assuming adversary has the background knowledge about strong rules, for* $\forall t \in T$, $\forall r \in Strong\_Rule\_Set(T)$, $r : A \Rightarrow B$, *adversary is sure that* t *contains* r *with the probability:*

$$P(t : A \Longrightarrow B) \leq \frac{1}{L}. \tag{3}$$

Here, $Strong\_Rule\_Set(T)$ *represents the set of strong rules in table* T.

Proof. If $t \in$ IR, there are no records containing strong rules in table IR, so $P(t : A \Rightarrow B) = 0$. If $t \in$ SAC$^{\sim}$, for each group G in SAC$^{\sim}$, G has at least L records; if there is record containing $A \Rightarrow B$ in G, according to the nature of random permutation, each record in G contains $A \Rightarrow B$ with equality probability $1/L$, so we can know that even though adversary can be clear of how many records contain $A \Rightarrow B$ through related background knowledge, $P(t : A \Rightarrow B)$ is not more than $1/L$. □

In the example of Section 3.3, for some records, adversary can make sure that they contain strong rules with probability 100%. SAC$^{\sim}$ table can prevent adversary from using strong rules to attack users' privacy.

4.2. Improved Rating (IR). The table IR will be processed by improved Rating. For each sensitive attribute S, Rating [17] hashes $t.S$ in S by their values (each bucket corresponds to each value), and if S has m different values $(v_1, v_2, \ldots, v_m)$, it can get sequence = $\{\text{bucket}_1, \text{bucket}_2, \ldots, \text{bucket}_m\}$, and assuming there are n v_i $(1 \leq i \leq m)$ in S, so the corresponding bucket_i contains n v_i in it. Every time Rating chooses the L buckets that have the largest size, gets a value

from every one of these L buckets to make up a SID, uses SID to replace corresponding sensitive attribute values in original table, and gets attribute table, and the SID makes up ID table. Every time Rating generates an SID, it needs to reselect L buckets, because after the last generation of SID, the sizes of buckets have been updated. Paper [17] has not presented the algorithm of choosing L largest buckets, so this paper will present a heuristic algorithm of choosing buckets.

4.2.1. Heuristic Algorithm for Choosing Buckets. This heuristic algorithm (Algorithm 13) is actually a stable sorting algorithm, sort sequence in descending order, and choose the first L buckets from sequence. This algorithm will be called after updating the size of buckets. Let us introduce some parameters, and sequence[i] refers to the i ($1 \le i \le m$) bucket in the sequence:

sequence[i].value: the attribute value in sequence[i],

sequence[i].size: the size of sequence[i],

sequence[i].size˜: after updating size, the size of sequence[i], if $1 \le i \le L$, sequence[i].size˜ = sequence[i].size-1; if $L < i \le m$, sequence[i].size˜ = sequence[i].size,

sequence[i].position: the position of sequence[i], sequence[i].position = i,

sequence[i].position˜: after sorting, the new position of sequence[i].

Algorithm 13 (heuristic choosing L largest buckets). Input: sequence, L,

Output: sequence satisfies stable descending order,

(1) If sequence[L].size˜ < sequence[$L+1$].size˜, do (2) to (5).

(2) For $i = 1; i < L + 1; i++$, repeat (3) to (5)

(3) If sequence[i].size˜ < sequence[$L + 1$].size˜, do (4) to (5).

(4) Find sequence[j] and sequence[$j + 1$] which satisfy sequence[j].size˜ > sequence[i].size˜ ≥ sequence[$j+1$]. size˜

(5) remove Q = {sequence[i], sequence[$i + 1$], ..., sequence[L]} to the position between sequence[j] and sequence[$j + 1$]; meanwhile keep the relative position among Q, end loop.

If the sequence always satisfies the stable descending order, there are Lemmas 14 and 15 and Corollaries 16 and 17, and we will prove the correctness of algorithm.

Lemma 14. *After updating size of buckets, if sequence*[L].*size*˜ ≥ *sequence*[$L + 1$].*size*˜, *the position of all buckets in sequence does not need to be adjusted.*

Proof. Using proof by contradiction, only the sizes of sequence[1], sequence[2], ..., sequence[L] have been changed, so these buckets may need to adjust their position in the sequence. If sequence[i] ($1 \le i < L$) needs to be removed to the back of sequence[j] ($L < j \le m$), then sequence[j].size˜ > sequence[i].size˜ and sequence[$L + 1$].size˜ ≥ sequence[j].size˜ > sequence[i].size˜. Because sequence[i].size ≥ sequence[L].size, and after generation, in both the sizes of sequence[i] and sequence[L] minus 1, there is sequence[i].size˜ ≥ sequence[L].size˜, we can get the contradictory conclusion:

$$\text{sequence}[L+1].\text{size}^{\sim} > \text{sequence}[L].\text{size}^{\sim}. \quad (4)$$

□

Lemma 15. *If sequence*[j].*size*˜ > *sequence*[i].*size*˜ ≥ *sequence*[$j + 1$].*size*˜ ($1 \le i, i + 1 \le L, L < j, j + 1 \le m$), *in order to preserve stable descending order, there are sequence*[j].*position*˜ < *sequence*[i].*position*˜ < *sequence*[$j + 1$].*position*˜ *and sequence*[$i + 1$]. *position*˜ = *sequence*[i]. *position*˜ + 1.

Proof. Obviously, according to the nature of stable descending sorting, the new position of sequence[i] is between sequence[j] and sequence[$j + 1$]. Here we will discuss the new position of sequence[$i + 1$]. In addition to the situation in Lemma 15, the new position of sequence[$i + 1$] has two possibilities, using the contradiction to proof, respectively.

(1) After adjusting, the position of sequence[$i + 1$] will be sequence[$i + 1$].position˜ and sequence[$i + 1$].position˜ < sequence[i].position˜. According to the nature of descending order, before generating SID, sequence[i].size ≥ sequence[$i + 1$].size and sequence[i].position < sequence[$i + 1$].position, after generating SID, in both the sizes of sequence[i] and sequence[$i + 1$] minus 1, one has sequence[i].size˜ ≥ sequence[$i + 1$].size˜. So sequence[$i + 1$].position˜ < sequence[i].position˜ contradicts the nature of stable descending order.

(2) After adjusting, the position of sequence[$i+1$] will be sequence[$i+1$].position˜, sequence[$i+1$].position˜ > sequence[i].position˜, and existing set Q = {sequence[k] | $1 \le k \le m$}, and each sequence[k] ∈ Q satisfies that sequence[i].position˜ < sequence[k]. position˜ < sequence[$i + 1$].position˜. If $1 \le k \le L$, according to the nature of stable descending order, sequence[i].size˜ ≥ sequence[k].size˜ ≥ sequence[$i + 1$].size˜, sequence[i].size ≥ sequence[k].size ≥ sequence[$i + 1$].size, sequence[i].position < sequence[k].position < sequence[$i + 1$].position, and we can get the contradictory conclusions: $i < k < i+1$. In another situation, if $L < k \le m$, one must have sequence[i].size˜ ≥ sequence[k].size˜ > sequence[$i + 1$].size˜ and sequence[k].size = sequence[$i + 1$].size ≥ sequence[j].size; because the sizes of sequence[j] and sequence[k] have not changed, there is sequence[k].size˜ ≥ sequence[j]. size˜, so we can get the contradictory conclusions: sequence[i].size˜ ≥ sequence[j].size˜.

□

Corollary 16. *If sequence*[i].*size*˜ < *sequence*[$L+1$].*size*˜, *the new position of sequence*[i] *will be sequence*[i].*position*˜; *for each sequence*[k] ($i < k \le L$), *its new position is as follows: sequence*[k].*position*˜ = *sequence*[i].*position*˜ + ($k - i$).

Proof. According to Lemma 15, sequence[i + 1].position$^{\sim}$ = sequence[i].position$^{\sim}$ + 1; for each sequence[k] ($i < k \leq L$), there exists sequence[k].position$^{\sim}$ = sequence[$k-1$].position$^{\sim}$ + 1, so for each sequence[k] ($i < k \leq L$), it satisfies

$$\text{sequence}[k].\text{position}^{\sim} = \text{sequence}[i].\text{position}^{\sim} + (k-i). \tag{5}$$

□

Corollary 17. *After updating size of bucket, if sequence[j].size$^{\sim}$ > sequence[i].size$^{\sim}$ ≥ sequence[$j+1$].size$^{\sim}$ ($L < j$, $j+1 \leq m$), here $i = 1$ or ($1 < i \leq L$, sequence[$i-1$].size$^{\sim}$ ≥ sequence[$L+1$].size$^{\sim}$); to make sequence satisfy the stable descending order, one only needs to remove Q = {sequence[i], sequence[$i+1$], ..., sequence[L]} to the position between sequence[j] and sequence[$j+1$].*

Proof. After updating size, only sizes of {sequence[1], sequence[2], ..., sequence[L]} have changed, so only the positions of {sequence[1], sequence[2], ..., sequence[L]} may need to be adjusted. According to the nature of stable descending order sequence[i].position$^{\sim}$ = sequence[j].position$^{\sim}$ + 1; according to Corollary 16, sequence[k].position$^{\sim}$ = sequence[i].position$^{\sim}$ + ($k-i$), ($i < k \leq L$), and sequence[$j+1$].position$^{\sim}$ = sequence[L].position$^{\sim}$ + 1. So removing {sequence[i], sequence[$i+1$], ..., sequence[L]} to the position between sequence[j] and sequence[$j+1$] can make sequence satisfy stable descending order.

Here one analyzes the efficiency of this sorting algorithm. In the best situation, sequence[L].size$^{\sim}$ ≥ sequence[$L+1$].size$^{\sim}$, it only needs to compare sequence[L].size$^{\sim}$ with sequence[$L+1$].size$^{\sim}$ in this algorithm, and the time complexity is $O(1)$. The worst situation is that sequence[1].size$^{\sim}$ < sequence[m].size$^{\sim}$, the algorithm needs to compare $m-L+1$ times, and the time complexity is $O(m-L+1)$. The efficiency of this sorting algorithm is much better than most of existing sorting algorithms. □

4.2.2. SID Creation. This part will introduce the algorithm of creating SID. The definition of dangerous bucket will be introduced as follows.

Definition 18 (dangerous bucket). Assuming sequence is in sensitive attribute S, for each bucket $B \in$ sequence, if B.size/$n \geq 1/L$, B is a dangerous bucket in S. Here, n is size of the domain (S).

The SID creation can be seen in Algorithm 19. For each sensitive attribute S_i, one generates its sequence and removes the dangerous buckets from sequence to SDB. Every time one generates a new $S_i\text{ID}_j$, for each bucket in SDB, removes one value from it to $S_i\text{ID}_j$ (line 6), for each bucket of sequence[1], sequence[2], ..., sequence[L-|SDB|] which have largest size, removes a value from it to $S_i\text{ID}_j$ (line 7). When a $S_i\text{ID}_j$ is completed, call for Algorithm 13 to sort the sequence. In the step of processing residual values, for each value v in a nonempty bucket, remove v to a $S_i\text{ID}_j$ which does not contain value v (line 9).

Algorithm 19 (SID creation). Input: L, IR table
Output: $S_i\text{ID}_j$
(1) For each sensitive attribute S_i, do (2) to (9).
(2) Generate sequence.
(3) Find the dangerous buckets, and put them in SDB.
(4) Remove the dangerous buckets in sequence;
(5) When there are at least L-|SDB| buckets in sequence which are not empty, generate a new $S_i\text{ID}_j$ repeat (6) to (8).
(6) For each bucket in SDB, remove a value to $S_i\text{ID}_j$.
(7) For each bucket from {sequence[1], sequence[2], ..., sequence[L-|SDB|]}, remove a value to $S_i\text{ID}_j$;
(8) Call for Algorithm 13 and use sequence and L-|SDB| as input; //processes the residual attribute values
(9) For each value v in nonempty buckets, find a $S_i\text{ID}_j$ which contains no value v, and remove v to $S_i\text{ID}_j$. If one cannot find this $S_i\text{ID}_j$, value v cannot be grouped.

Lemma 20. *If bucket B is a dangerous bucket in sensitive attribute S_i, after completing a new S_iID_j, B is still a dangerous bucket.*

Proof. After completing a new $S_i\text{ID}_j$, the frequency of B.value is (B.size − 1)/($n - L$), and n is the size of domain(S_i). Before generating the new $S_i\text{ID}_j$, the frequency of B.value was B.size/n. If there is (B.size − 1)/($n-L$) ≥ B.size/n, Lemma 20 can be proved.

The left side of the equation is equal to (B.size−1) ∗ n/(($n-L$) ∗ n), and the right side of the equation is equal to B.size ∗ ($n-L$)/(n ∗ ($n-L$)); one only needs to prove that

$$(B.\text{size} - 1) * n \geq B.\text{size} * (n - L). \tag{6}$$

Because before generating the new $S_i\text{ID}_j$, B was dangerous bucket, and satisfied B.size/$n \geq 1/L$, has B.size ∗ $L \geq n$, B.size ∗ $n - n \geq B$.size ∗ $n - B$.size ∗ L so we can get

$$(B.\text{size} - 1) * n \geq B.\text{size} * (n - L). \tag{7}$$

□

Corollary 21. *If bucket B is a dangerous bucket, after completing a new S_iID_j, B is still one of the L buckets which have the largest size.*

Proof. Using proof by contradiction, if there is Set = {bucket$_1$, bucket$_2$, ..., bucket$_L$}, for each bucket ∈ Set, it has larger size than B. According to Lemma 20, after generating a new $S_i\text{ID}_j$, the B.size$^{\sim}$/$n \geq 1/L$, so bucket satisfies bucket.size$^{\sim}$/$n \geq 1/L$, so there has

$$\frac{\text{bucket}_1.\text{size}^{\sim}}{n} + \frac{\text{bucket}_2.\text{size}^{\sim}}{n} + \cdots + \frac{\text{bucket}_L.\text{size}^{\sim}}{n} + \frac{B.\text{size}^{\sim}}{n} > 1. \tag{8}$$

Get the contradictory conclusions:

$$\text{bucket}_1.\text{size}^{\sim} + \text{bucket}_2.\text{size}^{\sim} + \cdots + \text{bucket}_L.\text{size}^{\sim} + B.\text{size}^{\sim} > n. \tag{9}$$

From the above certification, we can find out that the dangerous bucket is always one of the L largest buckets, so each new $S_i\mathrm{ID}_j$ should take one value of dangerous bucket, and it does not need to consider dangerous bucket for sorting sequence. This method further improves the efficiency of the algorithm.

Besides, in this improved Rating, the algorithm for AT&IDT Creation is the same as in Rating, uses $S_i\mathrm{ID}_j$ to generalize corresponding value in IR table, after processing IR, gets AT, and then uses the set of $S_i\mathrm{ID}_j$ to make up IDT. At last release AT and IDT with the previous SAC˜ table.

Here we assume adversary masters strong rules and summarize the security of mixed model. For released table SAC˜, because each group contains at least L records without the same values and refers to Lemma 12, it is easy to know that SAC˜ satisfy L-diversity. For released AT, we will discuss a problem of probability tilts first. For record t, after generalization if $\exists v_1 \in t.S_x$, $\exists v_2 \in t.S_y$ $(1 \le x, y \le j)$ satisfy $v_1 \Rightarrow v_2$ which is a strong rule, there will be probability tilt between $t.S_x$ and $t.S_y$ obviously. And according to the method of partition table, all the strong values in AT belong to the same sensitive attribute, so the probability tilts will not happen in AT. Besides, each $S_i\mathrm{ID}_j$ consists of at least L different values, so AT also satisfies L-diversity. And through the above analysis, we know that mixed model is able to satisfy L-diversity. □

5. Analysis and Proof of Information Availability

This section analyzes the information loss of the new model from availability of association rules and the quality of published data table.

5.1. The Availability of Association Rules. In Rating model, all the relationships between different sensitive attributes are broken, the new model presented by this paper makes improvement, and all the strong rules can be preserved in released table.

Lemma 22. *If the association rule is as follows: $A \Rightarrow B$ [confidence = s $(0 \le s \le 1)$], $s \ge$ min_confidence in the original data table, in the released data tables, the confidence of $A \Rightarrow B$ is still s.*

Proof. The released data tables contain SAC˜ table, ID table, and the attribute table, and user can get the support degree of A from SAC˜ table and ID table, namely, support(A), because in attribute table there is support(A, B) = 0, one only needs to get support(A, B) from the SAC˜ table. So the confidence s of $A \Rightarrow B$ is support(A, B)/support(A).

Here, we can see that the mixed model preserves all the strong association rules, and user can get the confidence of strong association rules from the released tables. And the Rating model breaks all the relationships between sensitive attributes and generates unnecessary information loss. □

5.2. The Quality of Published Data Table. This part uses the reconstruction error (RCE) [9, 17] to measure the quality of the published tables. Assume original table $T = (\mathrm{QI}_1, \mathrm{QI}_2, \ldots, \mathrm{QI}_i, S_1, S_2, \ldots, S_j)$ gets a $i + j$ dimensional space DS_{i+j}; for record t in table T, the probability density function (pdf) of t is

$$F_t(x) = \begin{cases} 1, & \text{if } x.\mathrm{QI}_p = t.\mathrm{QI}_p,\ x.S_q = t.S_q,\ 1 \le p \le i,\ 1 \le q \le j, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Here, the x is a $i + j$ dimensional variable in DS_{i+j}.

For record t in the released table of Rating, the pdf of t is

$$F_t^{\mathrm{rating}}(x) = \begin{cases} \prod_{q=1}^{j} \frac{1}{|t.S_q|}, & \text{if } x.\mathrm{QI}_p = t.\mathrm{QI}_p,\ x.S_q \in t.S_q,\ 1 \le p \le i,\ 1 \le q \le j, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Assume the Cluster_Set = $\{C_1, C_2, \ldots, C_m\}$ in the mixed model, the $(\mathrm{QI}_1, \mathrm{QI}_2, \ldots, \mathrm{QI}_i, C_1, C_2, \ldots, C_m)$ defines a $i + m$ dimensional space DS_{i+m}. In the released tables of the mixed model, if $t \in$ SAC˜, the pdf of t is

$$F_t^{\mathrm{SAC}^{\sim}}(x) = \begin{cases} \prod_{q=1}^{m} \frac{1}{|t.C_q^{\sim}|}, & \text{if } x.\mathrm{QI}_p = t.\mathrm{QI}_p,\ x.C_q \in t.C_q^{\sim},\ 1 \le p \le i,\ 1 \le q \le m, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Here, x is a $i + m$ dimensional variable in DS_{i+m}, $t.C_q^{\sim}$ represents the set of the possible values of $t.C_q$, and $|t.C_q^{\sim}|$represents the number of the possible values. For example, in Table 6, a user wants to reconstruct the pdf of t_1; in his view, the $t_1.C_1$ can be (a_4, b_2) or (a_1, b_1) with equality probability 1/2, and $t_1.C_2$ can be c_2 or c_1 with equality probability 1/2, so the pdf of t_1 is

$$F_{t_1}^{\mathrm{SAC}^{\sim}}(x) = \begin{cases} \dfrac{1}{4}, & \text{if } \left(x.\mathrm{QI}_p = t_1.\mathrm{QI}_p,\ x.C_1 = (a_4b_2),\ x.C_2 = (c_2)\right) \text{ or } \left(x.\mathrm{QI}_p = t_1.\mathrm{QI}_p,\ x.C_1 = (a_1b_1),\ x.C_2 = (c_2)\right), \\ & \text{or } \left(x.\mathrm{QI}_p = t_1.\mathrm{QI}_p,\ x.C_1 = (a_4b_2),\ x.C_2 = (c_1)\right) \text{ or } \left(x.QI_p = t_1.\mathrm{QI}_p,\ x.C_1 = (a_1b_1),\ x.C_2 = (c_1)\right)\ 1 \le p \le i, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

If $t \in \mathrm{AT}$, pdf of t is $F_t^{\mathrm{rating}}(x)$. So in the released tables of mixed model, the pdf of t is

$$F_t^{\mathrm{mm}}(x) = \begin{cases} F_t^{\mathrm{SAC}^{\sim}}(x), & t \in \mathrm{SAC}^{\sim}, \\ F_t^{\mathrm{rating}}(x), & t \in \mathrm{AT}. \end{cases} \tag{14}$$

We can measure the distance between released tables of mixed model and original table as follows:

$$\sum_{x \in \mathrm{SD}_{i+m}} \left(F_{t^{\mathrm{mm}}}^{\mathrm{mm}}(x) - F_t(x)\right)^2. \tag{15}$$

Here, assume t is a record in original table and t^{mm} is the form of t in released tables of mixed model. Similarly, t^{rating} is the form of t in released table of Rating, and the distance between released table of Rating model and original table is as follows:

$$\sum_{x \in \mathrm{SD}_{i+j}} \left(F_{t^{\mathrm{rating}}}^{\mathrm{rating}}(x) - F_t(x)\right)^2. \tag{16}$$

The released table would have higher quality with the smaller distance. Take all the records $\{t_1, t_2, \ldots, t_n\}$ into account, and the reconstruction error (RCE) of mixed model and Rating, respectively, are

$$\int_{k=1}^{n} \sum_{x \in \mathrm{SD}_{i+m}} \left(F_{t^{\mathrm{mm}}}^{\mathrm{mm}}(x) - F_t(x)\right)^2 dk, \qquad \int_{k=1}^{n} \sum_{x \in \mathrm{SD}_{i+j}} \left(F_{t^{\mathrm{rating}}}^{\mathrm{rating}}(x) - F_t(x)\right)^2 dk. \tag{17}$$

Lemma 23. *If the original table contains strong rules, the RCE of mixed model is smaller than Rating model:*

$$\int_{k=1}^{n} \sum_{x \in SD_{i+m}} \left(F_{t^{mm}}^{mm}(x) - F_t(x)\right)^2 dk < \int_{k=1}^{n} \sum_{x \in SD_{i+j}} \left(F_{t^{rating}}^{rating}(x) - F_t(x)\right)^2 dk. \tag{18}$$

Proof. To simplify the proof, assume both two models have no remaining attribute value to process, which means $t^{\mathrm{mm}}.C^{\sim}$ and $t^{\mathrm{rating}}.S$ satisfy $|t^{\mathrm{mm}}.C^{\sim}| = L$, $|t^{\mathrm{rating}}.S| = L$. In order to prove the conclusion, we only need to prove

$$\sum_{x \in \mathrm{SD}_{i+m}} \left(F_{t^{\mathrm{mm}}}^{\mathrm{mm}}(x) - F_t(x)\right)^2 < \sum_{x \in \mathrm{SD}_{i+j}} \left(F_{t^{\mathrm{rating}}}^{\mathrm{rating}}(x) - F_t(x)\right)^2. \tag{19}$$

For each t in AT table of mixed model, because the AT table satisfies the requirement of the Rating model, there exists $\sum_{x \in \mathrm{SD}_{i+m}} (F_{t^{\mathrm{mm}}}^{\mathrm{mm}}(x) - F_t(x))^2 = \sum_{x \in \mathrm{SD}_{i+j}} (F_{t^{\mathrm{rating}}}^{\mathrm{rating}}(x) - F_t(x))^2$. For each t in $\mathrm{SAC}^{\sim}$ table, one has

$$\begin{aligned} &\sum_{x \in \mathrm{SD}_{i+m}} \left(F_{t^{\mathrm{mm}}}^{\mathrm{SAC}^{\sim}}(x) - F_t(x)\right)^2 \\ &= \sum_{\substack{x.\mathrm{QI}_p = t.\mathrm{QI}_p, \\ x.C_q = t.C_q \\ 1 \le p \le i,\ 1 \le q \le m}} \left(\prod_{q=1}^{m} \frac{1}{\left|t^{\mathrm{mm}}.C_q^{\sim}\right|} - 1\right)^2 \\ &\quad + \sum_{\substack{x.\mathrm{QI}_p = t.\mathrm{QI}_p, \\ \exists x.C_q \ne t.C_q, \\ x.C_q \in t^{\mathrm{mm}}.C_q^{\sim} \\ 1 \le p \le i,\ 1 \le q \le m}} \left(\prod_{q=1}^{m} \frac{1}{\left|t^{\mathrm{mm}}.C_q^{\sim}\right|} - 0\right)^2 \\ &= \left(\prod_{q=1}^{m} \frac{1}{\left|t^{\mathrm{mm}}.C_q^{\sim}\right|} - 1\right)^2 + \left(\prod_{q=1}^{m} \left|t^{\mathrm{mm}}.C_q^{\sim}\right| - 1\right) * \left(\prod_{q=1}^{m} \frac{1}{\left|t^{\mathrm{mm}}.C_q^{\sim}\right|}\right)^2 = 1 - \prod_{q=1}^{m} \frac{1}{\left|t^{\mathrm{mm}}.C_q^{\sim}\right|}. \end{aligned} \tag{20}$$

Similarly, for t in released table of Rating, one has

$$\sum_{x \in \mathrm{SD}_{i+j}} \left(F_{t^{\mathrm{rating}}}^{\mathrm{rating}}(x) - F_t(x)\right)^2 = 1 - \prod_{q=1}^{j} \frac{1}{\left|t^{\mathrm{rating}}.S_q\right|}. \tag{21}$$

Because the original table has strong rules, $m < j$, $|t^{\mathrm{mm}}.C_q^{\sim}| = |t^{\mathrm{rating}}.S_q| = L$,

$$\prod_{q=1}^{j} \frac{1}{\left|t^{\mathrm{rating}}.S_q\right|} < \prod_{q=1}^{m} \frac{1}{\left|t^{\mathrm{mm}}.C_q^{\sim}\right|}, \quad \text{so } 1 - \prod_{q=1}^{m} \frac{1}{\left|t^{\mathrm{mm}}.C_q^{\sim}\right|} < 1 - \prod_{q=1}^{j} \frac{1}{\left|t^{\mathrm{rating}}.S_q\right|}. \tag{22}$$

The mixed model has lower RCE than Rating, which means the released tables of mixed model are closer to the original table. The linking between sensitive attributes in the same cluster is preserved, so the mixed model has higher quality than the Rating. □

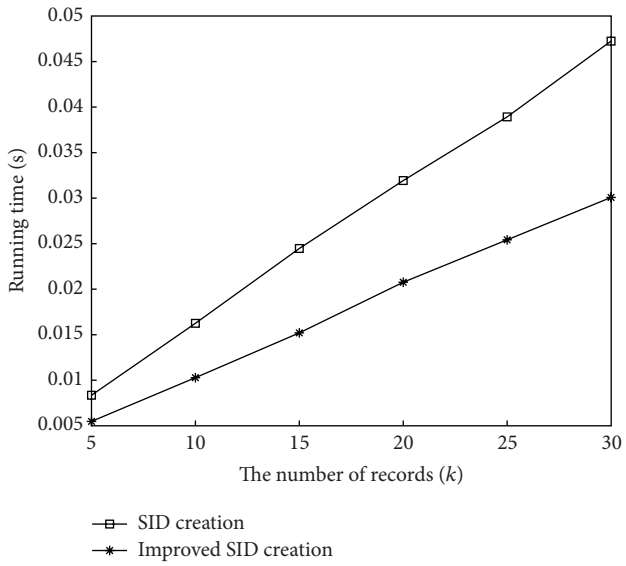

Figure 1: The comparison of execution time, $L = 3, 2S$.

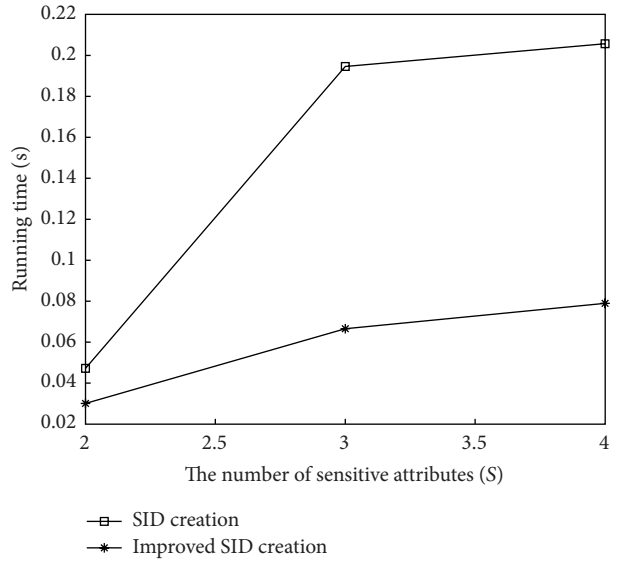

Figure 2: The effect of sensitive attributes number on execution time, $L = 3, 30k$.

6. Experiment

The experiment uses the real dataset Adult (http://archive.ics.uci.edu/ml/datasets/Adult), we get 30718 records after deleting the incomplete ones, and the experiment consists of four parts: (1) execution time, (2) additional information loss, (3) accuracy rate of mining strong association rules, and (4) probability of privacy disclosure. We choose {education, occupation, age, relationship} as sensitive attributes, $2S$ {education, occupation}, $3S$ {education, occupation, age}, and $4S$ {education, occupation, age, relationship}. If there are no special statements, the experiments use the default parameters: the number of records is 30718, and the min_confidence is 80%.

6.1. Execution Time. This paper presents an improved algorithm of Rating and mainly improves the algorithm of the SID creation, and the AT and IDT creation is the same as Rating. So this part will compare the improved algorithm of SID creation with the old one. Here, the old SID creation uses the stable bubble sort algorithm when choosing the L largest buckets. We set parameters $\{L = 3, 2S\}$ and choose a certain number of records randomly, and Figure 1 shows that the execution time of improved SID creation is much better than the old one, because of the heuristic search. And the improved SID creation is more suitable for the large dataset.

Figure 2 shows the effect of sensitive attributes number on execution time. Because age has much more different values than other 3 sensitive attributes, Bubble sort needs more time to compare. After adding age attribute, the running time of SID creation increases rapidly. So the running time of SID creation is influenced seriously by the number of different values of sensitive attribute.

6.2. Additional Information Loss. This part compares the additional information loss (AIL) [12] of mixed mode with the Rating. In order to make AIL more suitable for mixed model and Rating, we slightly change its definition. Assume sensitive S has m SID in IDT, the additional information loss of S is

$$\mathrm{AIL}(S) = \sum_{i=1}^{m} \frac{(|\mathrm{SID}_i| - L)}{m * L}. \tag{23}$$

Here, $|\mathrm{SID}_i|$ represents the number of values in SID_i. And the additional information loss of table T is:

$$\mathrm{AIL}(T) = \sum_{i=1}^{j} \frac{\mathrm{AIL}(S_i)}{j}. \tag{24}$$

Here, T has sensitive attributes $\{S_1, S_2, \ldots, S_j\}$.

Figure 3 shows the AIL of the mixed model and Rating; when L increases, both the additional information losses of the two models increase basically. And the additional information loss of the mixed model is slightly more than the one of Rating but is always less than 0.03%, and the security of mixed model is enhanced. Here, Rating uses the stable bubble sort algorithm in SID creation. This part of experiment also finds an interesting phenomenon: if the sort algorithm in SID creation is unstable, the additional information loss will be much more than the stable sorting algorithm in SID. This phenomenon needs to be further studied.

Figure 4 shows the effect of the minimum confidence on additional information loss. When the minimum confidence decreases, the additional information loss increases. Because more records are put in SAC table, the available sensitive attribute values in the IR table will be less, and additional information loss increases.

Figure 5 shows the effect of number of sensitive attributes on additional information loss. Because all the sensitive attributes in Rating are processed independently, the AIL of Rating is almost not influenced by the number of sensitive

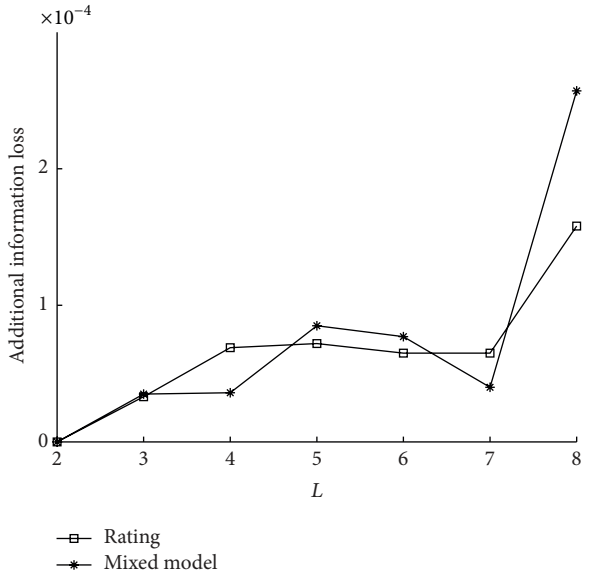

Figure 3: The comparison of additional information loss, 2*S*.

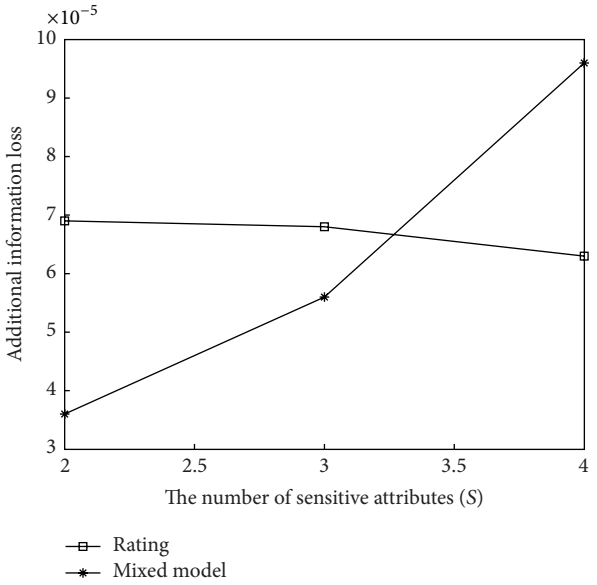

Figure 5: The effect of sensitive attribute number on additional information loss, $L = 4$.

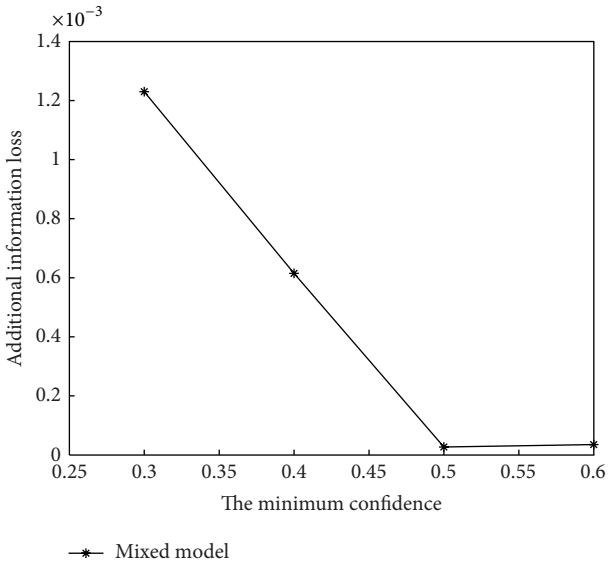

Figure 4: The effect of minimum confidence on additional information loss, $L = 3, 2S$.

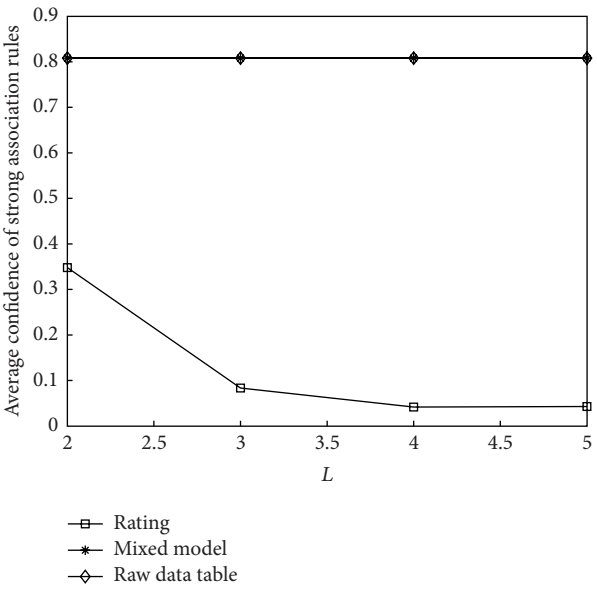

Figure 6: The average confidence of strong rules with varying *L*, 2*S*.

attributes. For mixed model, when the number of sensitive attributes increases, more records are removed to SAC table, and values for grouping are less in IR table, so the AIL of mixed model grows with the sensitive attribute number but is still in the realm of acceptable.

6.3. The Accuracy of Mining Strong Association Rules. Strong rules tend to be valuable in practice, so the ability to provide strong rules will be analyzed for publication models by this experiment. We use the method of Lemma 22 to excavate strong association rules from the released tables of mixed model, and in the released tables of Rating and the raw data table, we use Apriori to calculate confidence of strong association rules.

Figures 6 and 7 show the average confidence of strong rules in the three tables. We can see that if all the records in SAC and all values in IR can be grouped, user can accurately calculate confidence of strong rules from the released tables of mixed model, and the results also verify the conclusion of Lemma 22. When $L = 5$ in Figure 7, because the sensitive attribute relationship only has 6 different values and *L* is very close to 6, some records cannot be grouped in SAC, and they have to be deleted, and then the average confidence in

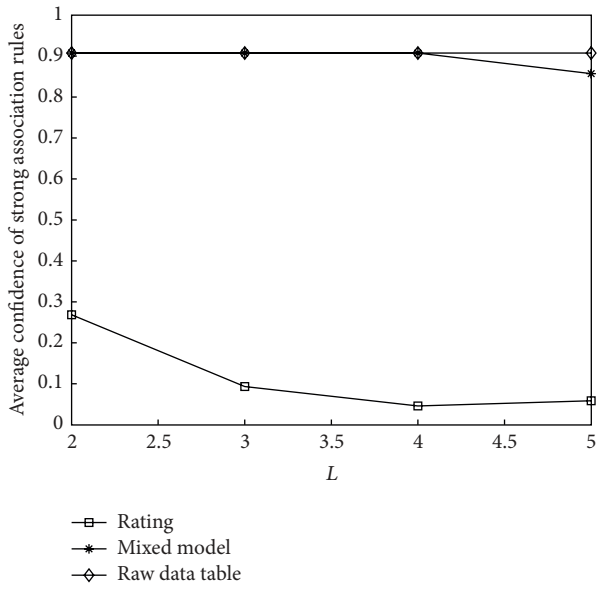

Figure 7: The average confidence of strong rules with varying L, 4S.

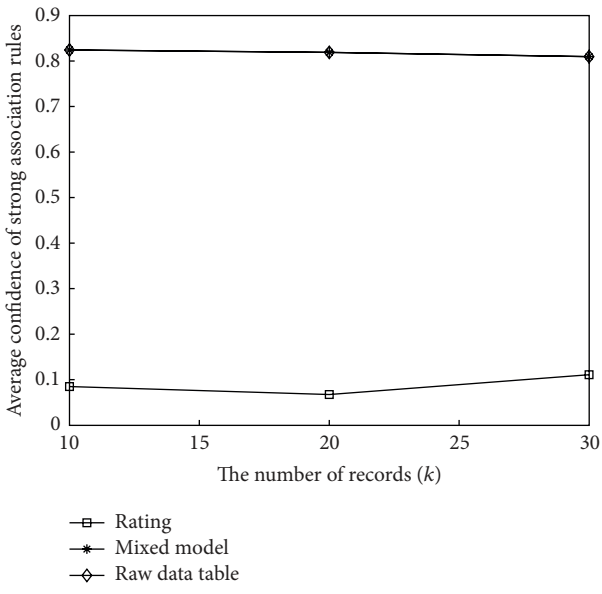

Figure 8: The average confidence of strong rules with varying the number of records $L = 3$, 2S.

mixed model deviates from the one in raw data table, but the difference is small. And the average confidence of strong rules in Rating greatly deviates from the one in raw data table; because Rating breaks all the relationships between sensitive attributes, it is difficult for users to calculate the confidence of strong rules from the released tables of Rating. Figures 8 and 9 show the similar results. Because mixed model has considered association rules between sensitive attributes, it can provide more valuable relationships than Rating in released tables.

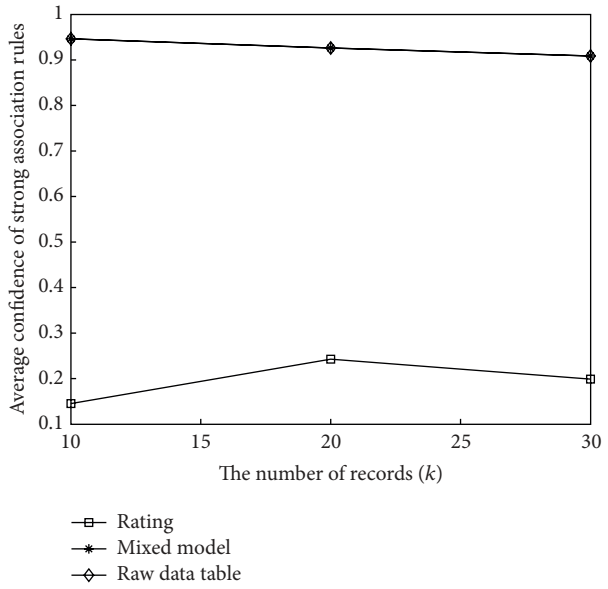

Figure 9: The average confidence of strong rules with varying the number of records, $L = 3$, 4S.

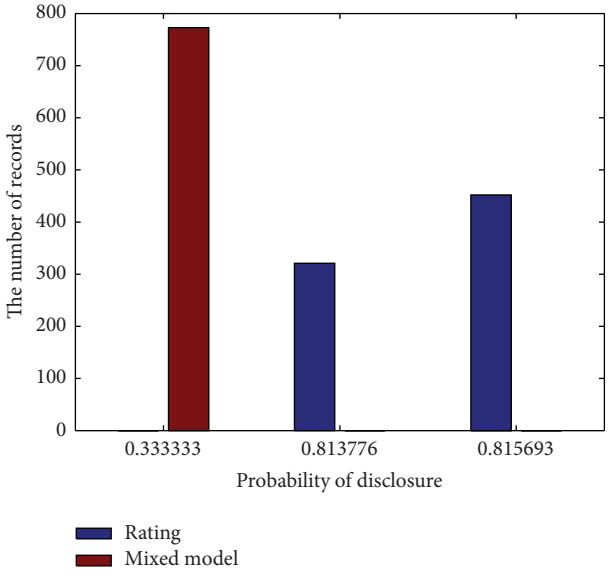

Figure 10: The probability of disclosure, $L = 3$, 2S.

6.4. The Probability of Privacy Disclosure. We refer to the method of paper [15] and analyze the probability of disclosure in this experiment. Assume adversary has background knowledge about strong rules. Because the records containing no strong rules satisfy L-diversity according to [17] and previous analysis, we study the safety of records that contain strong rules in mixed model and Rating and analyze the probability that these records contain known strong rules from released tables. In Figure 10, x-dimension represents the probability of disclosure, and y-dimension represents the number of records. We can see that the probability records contain strong rules is 1/3 in the released tables

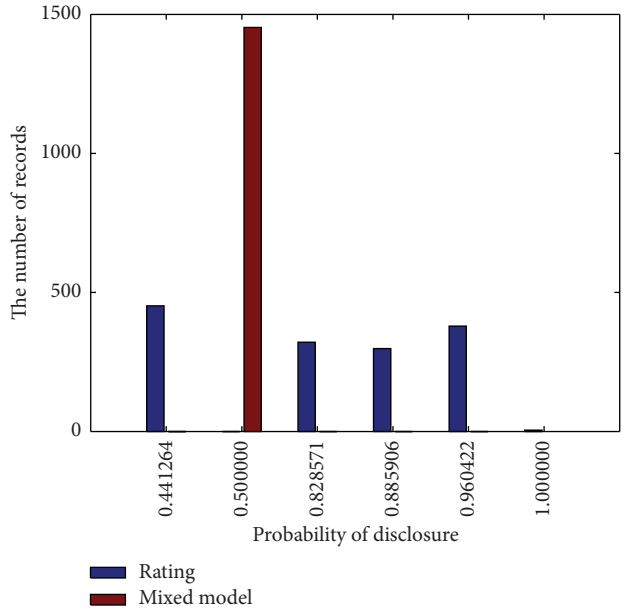

Figure 11: The probability of disclosure, $L = 2, 4S$.

of mixed model, mixed model can ensure a maximum of $1/L$ disclosure probability for records, and the conclusion of Lemma 12 is verified. On the other hand, because Rating has not considered the relationship between sensitive attributes, the disclosures probabilities for records in the released tables of Rating are more than 80%.

Figure 11 shows the similar result, the disclosures probabilities for many records are more than $1/L$ ($L = 2$) in Rating, while mixed model ensures a maximum of 1/2 probability for records. Here we will discuss an extreme situation. Assume $A \Rightarrow B$ is a strong rule, and in released tables of Rating, the number of records that actually contain $A \Rightarrow B$ is m, and the number of records that may contain $A \Rightarrow B$ is n; obviously, we have $n \geq m$. But if A or B has very low frequency in raw data table, the n may be very small or even equals m. When $n = m$, adversary can be sure which records contain $A \Rightarrow B$ in released tables of Rating. So we can see that the disclosures probabilities for several records of Rating are 100% in Figure 11. From these experiments, we can know that the mixed model can prevent adversary from attacking data table with related background knowledge, and it is able to provide better protection for privacy.

6.5. Summary of Experiment. Section 6.1 verifies the efficiency of the improved SID creation. And we know the additional information loss of mixed model is acceptable from the results of Section 6.2. Through the analysis of Sections 6.3 and 6.4, Rating cannot preserve strong rules in released tables, and as long as adversary masters background knowledge about these strong rules, Rating is unsafe. On the other hand, mixed model can provide strong rules for users forwardly and is able to ensure the security of privacy at the same time.

7. Summary

In view of the situation that most of existing privacy protection methods for multiple sensitive attributes have not taken the inherent relationship between different sensitive attributes into account, this paper presents that an attack method uses the association rules to get the users' privacy and accordingly presents a protection model. Through theoretical and experimental analysis, we prove that the new protection model can provide better protection for privacy, and it is able to preserve useful relationships in released tables. Besides, in order to improve the efficiency of algorithm, we present an improved SID creation method, and prove it is more effective with experiment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This paper is supported by "Guangzhou Research Institute of Communication University of China Common Construction Project, Sunflower – the Aging Intelligent Community."

References

[1] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[2] L. Sweeney, "Achieving *k*-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[3] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain K-anonymity," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '05)*, pp. 49–60, June 2005.

[4] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 24–26, April 2006.

[5] X. Xiao and Y. Tao, "Personalized privacy preservation," in *Prceedings of the 25th ACM SIGMOD International Conference on ACM Management of Data (SIGMOD '06)*, pp. 229–240, ACM, Chicago, Ill, USA, 2006.

[6] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, "Distribution based microdata anonymization," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 958–969, 2009.

[7] L. Ninghui, L. Tiancheng, and S. Venkatasubramanian, "t-Closeness: privacy beyond k-anonymity and l-diversity," in *Proceedings of the 23rd International Conference on Data Engineering (ICDE '07)*, pp. 106–115, IEEE, Istanbul, Turkey, April 2007.

[8] X.-C. Yang, X.-Y. Liu, B. Wang, and G. Yu, "K-anonymization approaches for supporting multiple constraints," *Journal of Software*, vol. 17, no. 5, pp. 1222–1231, 2006.

[9] X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06)*, pp. 139–150, Seoul, Republic of Korea, September 2006.

[10] N. Li, W. Qardaji, and D. Su, "Provably private data anonymization: or, *k*-anonymity meets differential privacy," CERIAS TR 2010-24, Center for Education and Research Information Assurance and Security, Purdue University, West Lafayette, India, 2010.

[11] G. Aggarwal, T. Feder, K. Kenthapadi et al., "Achieving anonymity via clustering," in *Proc. of the 25th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems*, S. Vansummeren, Ed., pp. 153–162, ACM, New York, NY, USA, 2006.

[12] X.-C. Yang, Y.-Z. Wang, B. Wang, and G. Yu, "Privacy preserving approaches for multiple sensitive attributes in data publishing," *Chinese Journal of Computers*, vol. 31, no. 4, pp. 574–587, 2008.

[13] Y. Jing and W. Bo, "Personalized l-diversity algorithm for multiple sensitive attributes based on minimum selected degree first," *Journal of Computer Research and Development (China)*, vol. 49, no. 12, pp. 2603–2610, 2012.

[14] Y. Ye, Y. Liu, D. Lv, and J. Feng, "Decomposition: privacy preservation for multiple sensitive attributes," in *Database Systems for Advanced Applications*, vol. 5463 of *Lecture Notes in Computer Science*, pp. 486–490, Springer, Berlin, Germany, 2009.

[15] A. Abdalaal, M. E. Nergiz, and Y. Saygin, "Privacy-preserving publishing of opinion polls," *Computers & Security*, vol. 37, pp. 143–154, 2013.

[16] Y. Fang, M. Zaman Ashrafi, and S. Kiong Ng, "Privacy beyond single sensitive attribute," in *Database and Expert Systems Applications*, pp. 187–201, Springer, 2011.

[17] J. Liu, J. Luo, and J. Z. Huang, "Rating: privacy preservation for multiple attributes with different sensitivity requirements," in *Proceedings of the 11th IEEE International Conference on Data Mining Workshops (ICDMW '11)*, pp. 666–673, Vancouver, Canad, December 2011.

[18] J. W. Han and M. Kamber, *Data Mining Concepts and Techniques*, China Machine Press, Beijing, China, 2011.