Decomposition: Privacy Preservation for Multiple Sensitive Attributes

Yang Ye, Yu Liu, Dapeng Lv, and Jianhua Feng

Department of Computer Science, Tsinghua University
Beijing, 100084, China
{yey05,liuyu-05,lvdp05}@mails.tsinghua.edu.cn
fengjh@tsinghua.edu.cn

Abstract. Aiming at ensuring privacy preservation in personal data publishing, the topic of anonymization has been intensively studied in recent years. However, existing anonymization techniques all assume each tuple in the microdata table contains one single sensitive attribute (the *SSA* case), while none paid attention to the case of multiple sensitive attributes in a tuple (the *MSA* case).
In this paper, we conduct the pioneering study on the MSA case, observe new privacy risks, and reason why generalization, the most common approach for anonymization, is impractical in this case. Instead, we propose a new framework, decomposition, to tackle privacy preservation in the MSA case. We elaborate decompose by extending it naturally from the SSA case and introducing the $(l_1, l_2, \ldots, l_d)$-diversity principle. Experiments with real data verify the effectiveness of decomposition.

Key words: Decomposition, $(l_1, l_2, \ldots, l_d)$-diversity, MSA

1 Introduction

With a host of organizations increasingly publishing personal data for scientific and business uses, the issue of privacy preservation in personal data publication has drawn broad attention. *Anonymization*[1, 2] is the most popularly adopted approach for this objective.

Typically, attributes in the *microdata*, like Table 1, can be categorized into three types: *identifying* attributes are the attributes that can be used to explicitly identify an individual; *quasi-identifying* (QI) attributes are the set of attributes that can be linked with public available datasets to reveal personal identity; *sensitive* attributes contain personal privacy information like *disease*, *salary*. Therefore, the exact value for an individual's sensitive attributes should not be directly or indirectly revealed. In the running example of Table 1, *Name* is the identifying attribute; {*Gender*,*ZipCode*,*Birthday*} is the set of QI attributes; *Occupation* and *Salary* are the sensitive attributes.

To fulfill the privacy goals, removing identifying attributes is necessary but insufficient, because the set of QI attributes can be linked with public available datasets, to reveal personal identity[2]. In the running example, Table 1 may be

linked with the voter register of Table 2 on the QI attributes. To counter such "*link attack*", anonymization techniques typically perform *generalization*[1–3] on QI attributes. Generalization transforms original QI values into a "less specific but semantically consistent form"[2] and partitions the table into *QI-groups*, each composed of tuples with identical and generalized QI values. The generalized table (Table 3 in the running example) is finally published.

Anonymization principles such as k-anonymity[2], l-diversity[5], put constraints on each QI-group. The pioneering principle, k-anonymity, requires each QI-group with size at least k. The subsequent and widely-adopted principle, l-diversity, further requires each group contains at least l "*well-represented*" sensitive values. In this way it reduces the risk of *sensitive attribute disclosure* to no higher than *1/l*.

1.1 Motivation

Current researches on anonymization all assume that there is one single sensitive attributes (the SSA case) in the microdata table. This assumption is arbitrary and inapplicable to practical use. For instance, in the "Adult" dataset we adopted in the experiment, multiple attributes, like "*Work-class*", "*Education*" and "*Hours per Week*" can be treated as sensitive attributes (the MSA case).

In the running example, two attributes, *Occupation* and *Salary* are treated as sensitive attributes. In Table 3, the first group satisfies 3-diversity for both *Occupation* and *Salary* attributes. Consider an adversary who obtains the QI values $\{M, 10076, 1985/03/01\}$ of Carl. Given the published Table 3, s/he can locate Carl in the first QI-group. However, since the first two tuples of Group 1 have "*nurse*" as the occupation value and according to common sense, nurse is generally a female occupation, thereby the adversary can locate Carl in the last two tuples. S/he will be able to reveal with high confidence that Carl's monthly salary is 8000-10000 dollars, belonging to the high-end. There is another case when the adversary previously knows Carl's occupation is *cook*. Since there is only tuple 3 in Group 1 having cook as the occupation value, the adversary can also reveal Carl's salary information.

In the forgoing examples, although Table 3 satisfies 3-diversity for *Occupation* and *Salary* separately, if the adversary obtains some information about the target's occupation value through *background knowledge*, s/he would be able to reveal the salary value.

1.2 Contributions

This paper provides the first study towards privacy preservation in the MSA case. First, we observe new privacy risks and conduct both exemplary and theoretical observation to conclude that generalization is ineffective in this case.

Second, we propose a new publishing methodology, decompose, to achieve privacy preservation in the MSA case. Instead of performing generalization on

Table 1. The Microdata Table

Tuple#	**Gender**	**ZipCode**	**Birthday**	**Occupation**	**Salary** [1]
1(Alice)	F	10078	1988/04/17	nurse	1
2(Betty)	F	10077	1984/03/21	nurse	4
3(Carl)	M	10076	1985/03/01	police	8
4(Diana)	F	10075	1983/02/14	cook	9
5(Ella)	F	10085	1962/10/03	actor	2
6(Finch)	M	10085	1988/11/04	actor	7
7(Gavin)	M	20086	1958/06/06	clerk	8
8(Helen)	F	20087	1960/07/11	clerk	2

Table 2. Part of a Vote Register List

Name	**Gender**	**ZipCode**	**Birthday**
Alice	F	10078	1988/04/17
Betty	F	10077	1984/03/21
Carl	M	10076	1985/03/01
Diana	F	10075	1983/02/14
Ella	F	10085	1962/10/03
Finch	M	10085	1988/11/04
Gavin	M	20086	1958/06/06
Helen	F	20087	1960/07/11

Table 3. The Generalized Table

#	**Gender**	**ZipCode**	**Birth.**	**Occ.**	**Sal.**
1	*	1007*	1983-88	nurse	1
2	*	1007*	1983-88	nurse	4
3	*	1007*	1983-88	police	8
4	*	1007*	1983-88	cook	9
5	*	*008*	1958-88	actor	2
6	*	*008*	1958-88	actor	7
7	*	*008*	1958-88	clerk	8
8	*	*008*	1958-88	clerk	2

QI attributes and forming QI-groups, our technique decomposes the table into so-called *SA-groups.*To retain valuable information lost in the transformed sensitive attributes, the original *sensitive table* is also published without privacy leakage.

We describe decompose by first elaborating it in the SSA case then extending it to the MSA case and introducing the $(l_1, l_2, \ldots, l_d)$-diversity principle. Decomposition in the SSA case largely resembles *Anatomy*[6] and *Permutation*[7]. But we amends their defects by concretely formalizing the group forming procedure and theoretically presenting its rationale. The theoretical analysis also provides us crucial knowledge about the limit of privacy preservation.

The rest of the paper is organized as follows. Section 2 formalize the problem and Section 3 introduce the general ideas of decompose. Section 4 detailedly describes decompose in the SSA case and Section 5 extends it to the MSA case. Section 6 gives the experimental evaluations. Section 7 introduces the related work and Section 8 concludes this paper.

[1] Here and in other tables, integer i means the monthly salary is between the range of $1000i - 1000(i+1)$ dollars, This representation, instead of concrete value is adopted in real datasets.

2 Preliminary

2.1 Basic Notations

Let $T = \{t_1, t_2, \cdots, t_n\}$ be the micordata table. Each tuple t contains a set of quasi-identifying attributes $\{A^1, A^2, \ldots, A^q\}$ and multiple sensitive attributes $\{S^1, S^2, \ldots, S^d\}$. We use $t.A$ to denote t's value of attribute A. We use T^S to denote the "*sensitive table*", or the projection of T on $\{S^1, S^2, \ldots, S^d\}$.

2.2 Privacy Goal and Utility Goal

To model the power of the adversary, we assume s/he has strong background knowledge that s/he knows: (i) the existence of target individual o in T, (ii) the whole QI values of o, and (iii) arbitrary information of o's some sensitive attributes.

The adversary obtains o's QI values by a ccessing the *external database* D. We assume each distinct combination of QI values can uniquely identify a single individual. Besides linking with D, the adversary can get plentiful knowledge about o's privacy by just observing the overall distribution of the sensitive attributes. For instance, if for attribute S, there are just two values: v_1 and v_2, with appearances of 10000 and 100 times respectively, the adversary can deduce with high confidence that the S value for any target is v_1. To sum up, the privacy goal can be stated as: (i) investigating the limit of privacy preservation techniques, achieve privacy requirement if within such limit; (ii) reducing the privacy leakage of one sensitive attribute because of background knowledge on other sensitive attributes.

A legal researcher should be allowed to obtain valuable information from the published table. (i) The researcher should be able to research the overall distribution of one sensitive attribute and the relationship between different sensitive attributes. This goal can be easily achieved through publishing the sensitive table T^S. (ii) The researcher should be able to retrieve valuable correlation between sensitive attributes and QI attributes.

3 From Generalization to Decomposition

3.1 Ineffectiveness of Generalization in the MSA Case

To illustrate the ineffective of generalization in the MSA case, we attempt to protect the privacy in one sensitive attribute, say S^d, even when the adversary have arbitrary background knowledge about $S^1, S^2, \ldots, S^{d-1}$, like the most extreme case "*the adversary obtains the exact value of other sensitive attributes*".

In this situation, to reduce the privacy risk for S^d to $1/l$, for each combination of $(S^1, S^2, \ldots, S^{d-1})$ values, there should be at least l distinct associated S^d values in G. Likewise, to protect attribute S^i, for each combination of $(S^1, S^2, \ldots, S^{i-1}, S^{i+1}, \ldots, S^d)$ values, there should be at least l distinct associated S^i values in G. We term this hypothetical diversity requirement d-SA-l-diversity. We can prove:

Theorem 1. *For QI-group G to satisfy d-SA-l-diversity, it must have exactly l distinct value for each $S^i(1 \leq i \leq d)$ and G is composed of l^d tuples, each taking one of the l^d possible combinations of $(S^1, S^2, \ldots, S^d)$.*

Proof. We prove by induction on d. When $d = 1$, the result is apparent. Assume when $d = k$, the conclusion holds. There are totally l^k tuples, each taking one of the l^k possible combinations of $(S^1, S^2, \ldots, S^k)$. For the privacy guarantee to be held when $d = k + 1$, for each combination of $(S^1, S^2, \ldots, S^k)$, there must be at least l associated distinct S^{k+1} values. If there are more than l associated distinct S^{k+1} values. This leads to some combination of $(S^2, S^3, \ldots, S^{k+1})$ that requires more S^1 values to be associated with. Likewise some combination of $(S^1, S^3, \ldots, S^{k+1})$ that requires more S^2 values to be associated with, and so on. So there must be exactly l distinct S^{k+1} values, and totally l^{k+1} tuples, each taking one of the l^{k+1} possible combinations.

Theorem 1 presents the ineffectiveness of generalization in the MSA case: the contradiction between *the tough selectivity requirement on sensitive attribute distribution within a group* and *the arbitrary and unpredictable distribution of microdata's sensitive attributes*. In fact, the hypothetical d-SA-l-diversity requirement is almost unachievable even when $d = 2$ and $l = 2$, not to mention the information loss even if achieved. In sum, we can conclude:

Generalization is ineffective for privacy preservation in the MSA case

3.2 General Idea of Decomposition

The privacy risk in the MSA case arises because of the linkage between different sensitive attributes. Therefore, a better approach may be directly cutting off such linkage and publish T^S separately without revealing privacy. If the target o can be directly associated with a set of distinct values for each S^i while values of different S^i do not have one-to-one linkage, any background knowledge about some S^i could not increase the risk of other sensitive attributes.

We term our methodology "*decomposition*". Firstly, it publishes the decomposed sensitive table. Secondly, instead generalized on QI attributes, tuples are grouped properly. Their QI values remain unchanged while tuples within a group share the union of their sensitive values. We decompose the table into such so-called SA-groups.

Definition 1. *(*SA-group*) A SA-group G contains tuples with their original, non-transformed QI values and for each S^i, each tuple in G is associated with the set of $G.S^i$ values.*

In the SSA case, decomposition resembles *anatomy*[6] and *permutation*[7]. Table 4 depicts a possible result of decompose on Table 1 with single sensitive attribute *Occupation*.

Table 4. The Decomposed Table for Single Sensitive Attribute

#	Gender	Zip.	Birth.	Occ.
1	F	10078	1988/04/17	police
	F	10085	1962/10/03	nurse
	M	20086	1958/06/06	actor
	M	10076	1985/03/01	clerk
2	F	10077	1984/03/21	nurse
	M	10085	1988/11/04	actor
	F	10075	1983/02/14	cook
	F	20087	1960/07/11	clerk

Table 5. The Decomposed Table for Two Sensitive Attributes

#	Gender	Zip	Birth.	Occ.	Sal.
1	F	10078	1988/04/17	police	1
	F	10085	1962/10/03	nurse	2
	M	20086	1958/06/06	actor	8
	M	10076	1985/03/01	clerk	
2	F	10077	1984/03/21	nurse	2
	M	10085	1988/11/04	actor	4
	F	10075	1983/02/14	cook	7
	F	20087	1960/07/11	clerk	9

4 Decomposition in the SSA case

In this section, we assume there is one single sensitive attribute S and aim at achieving l-diversity, namely, each tuple is associated with l distinct S values, so as to reduce the risk of privacy leakage to $1/l$. We shall research, given a diversity parameter l, how to properly decompose the table into SA-groups so that: (i) each group had better contains exactly l distinct sensitive values. (ii) the number of such SA-groups should be maximized.

We shall use following method: first place tuples with identical sensitive values into a same "*bucket*". Let B_i denote the i^{th} largest bucket and $\mathscr{B} = \{B_1, B_2, \ldots, B_m\}$ denote the set of buckets. We have: $n_i = |B_i|,\ n_1 \geq n_2 \geq \cdots \geq n_m$ and $\Sigma_{i=1}^{m} n_i = n$.

(*Largest-l group forming Procedure*) In each iteration, one tuple is removed from each of the l largest buckets to form a new SA-group. Note that after one iteration, the size of some buckets will be changed. So in the beginning of every iteration, the buckets are sorted according to their sizes, as shown in Figure 1.

Theorem 2. *The Largest-l group forming procedure creates as many groups as possible.*[2]

Proof. We prove by induction on $m = |\mathscr{B}|$ and $n = |T|$.

Basis. $m = n = l$. This is the basis because when $m < l$ or $n < l$, no group can be created. In this case, there is exactly one tuple in each bucket, apparently, the *Largest-l* procedure creates as many groups as possible.

Induction. When $m > l,\ n > l$. Assume the way W creates maximal number of groups, which equals k. We denote $G_i = \{i_1, i_2, \ldots, i_l\}$ $(i_1 < i_2 < \cdots < i_l)$ to be the i^{th} group created by W and G_i contains one tuple from each of $B_{i_1}, B_{i_2}, \ldots, B_{i_l}$. From W, a new way W' can be constructed that satisfies: (1) W' creates k groups; (2) the first group created by W' is $G_i^{'} = \{1, 2, \ldots, l\}$. The construction takes two operations: *swap* and *alter*.

[2] Similar procedure is also stated in [6], however [6] ignores to prove its optimality.

- $swap((G_i, a), (G_j, b))(1 \leq a, b \leq m, a \in G_i, a \notin G_j, b \in G_j, b \notin G_i)$ means to exchange a in G_i with b in G_j. For example, if $G_1 = \{1, 2\}$, $G_2 = \{3, 4\}$, $swap((G_1, 1), (G_2, 3))$ leads to $G_1 = \{2, 3\}$, $G_2 = \{1, 4\}$. Since $a \notin G_j$, $b \notin G_i$, after swap, G_1 and G_2 still contain l distinct values, the grouping way after this operation is always valid.

- $alter(a, b)$ $(1 \leq a < b \leq m)$ means to replace each a in every G_i with b and replace each b with a. For the above example, $alter(2, 3)$ leads to $G_1 = \{1, 3\}$, $G_2 = \{2, 4\}$. The grouping way is valid after this operation if and only if a's total appearing times is no more than b's.

The construction from W to W' is like this: for integer i from 1 to l, assume the i^{th} element in G_1 is b. If $i = b$, we do nothing. Otherwise, b must be greater than i. We check for other $k - 1$ groups $G_2, \ldots, G_k$. There are two possible cases:

- There is a group G_j such that $i \in G_j$ and $b \notin G_j$. In this case, we perform $swap((G_1, b), (G_j, i))$ to obtain a new grouping way. Since $i \notin G_1, b \notin G_j$, it is still a valid grouping way.

- Every group that contains i also contains b. Therefore, the total number of i's is no more than that of b's. In this case, we perform $alter(i, b)$, the grouping way is still valid after this operation.

Note operation on i ensures the i^{th} element in G_1 to be i and does not change the first $i-1$ elements. So when the whole process finishes, we obtain a valid grouping way W' with $G'_1 = \{1, 2, \ldots, l\}$. Removing tuples corresponding to the elements in G'_1, we obtain a new instance of the problem with $m' \leq m, n' = n - l < n$. Due to induction hypopiesis, the Largest-l procedure generates as many groups as possible for the new instance. In the meantime, the best solution to the new instance contains at lest $k-1$ groups, because $G'_2, G'_3, \ldots, G'_k$ is such a grouping way. So for the original instance, the Largest-l procedure generates at least k groups. That is the maximal number as assumed. The proof is completed.

Theorem 2 guarantees given the diversity parameter l, maximum groups can be formed. We shall also investigate: (i) in which case there will be not tuples left after the procedure; and (ii) what is the property of residual tuples, if any.

Theorem 3. *When the Largest-l group forming procedure terminates, there will be no residual tuples if and only if the buckets formed after the bucketizing step satisfy the following properties (we term it l-Property):*
(i) $\frac{n_i}{n} \leq \frac{1}{l},\ i = 1, 2, \ldots, m$ *(Use the same notation:* n_i, m, n *as in Theorem 2);*
(ii) $n = kl$ *for some integer* k*.*

Proof. First notice that $\frac{n_i}{n} \leq \frac{1}{l}$ is equivalent with $n_1 \leq k$, because n_1 is the largest among all n_i's.

(*If*) We prove by induction on $m = |\mathscr{B}|$ and $n = |T|$.

Basis. $m = n = l$, this is the basis because m cannot be smaller than l. Now there's one tuple in each bucket. Obviously the procedure leaves none.

Induction. $m > l$ or $n > l$. Resembling the proof of Theorem 2, we assume that when the first group is created by the procedure, the remaining buckets and tuples form a new instance of the problem with parameter (m', n'). We shall prove this new instance also satisfy l-Property.

Apparently $m' \leq m, n' = n - l = (k-1)l$. To prove $\frac{n'_1}{n'} \leq \frac{1}{l}$. We need only to prove $n'_1 \leq k - 1$.

Otherwise, $n'_1 = k$. However, it is required that in iteration 1, each of the largest l buckets contributes one tuple to G_1. After iteration 1, the largest bucket contains k tuples, so it didn't contribute to G_1. This means their are at least $k + 1$ buckets before iteration 1 that contains k tuples and their are totally at least $(k + 1)l > n$ tuples. This leads to contradiction.

Therefore the new instance satisfies l-Property. With the very same idea as used in the proof of Theorem 2, the outcome of the remaining execution of the procedure equals to what we obtain by running the procedure individually on the new instance. Due to induction hypopiesis, Largest-l procedure leaves no residual tuples for the new instance. So for the original instance, the conclusion also holds. The proof of if-part is completed.

(*Only-if*) It is easy to verify that n must be multiple of l to guarantee that no tuple will be left. So there exists some integer k such that $n = kl$.

Since there's no residual tuples, for the requirement of l-diversity, each group contains at most one tuple from the first bucket. The mapping from the tuples in B_1 to the groups is *one-to-one*, but not necessarily *onto*. Therefore, we have: $n_1 \leq k = \frac{n}{l}$ or $\frac{n_1}{n} \leq \frac{1}{l}$. The proof of only-if part is completed.

For the second problem about residual tuples, when the buckets formed through bucketization satisfy the first condition while do not satisfy the second condition of l-Property, we have following conclusion:

Corollary 1. *If the buckets satisfy:* $\frac{n_i}{n} \leq \frac{1}{l}$*, then when the Largest-l group forming terminates, each non-empty bucket contains just one tuple.*[3]

Proof. Assume $n = kl + r$, $0 \leq r < l$, hypothetically change the group forming procedure like this: first subtract one tuple from each of $B_1, B_2, \ldots B_r$, then operate the Largest-l group forming procedure on this new instance. It's easy to verify the new instance satisfies l-Property (i) and (ii), so k groups will be formed. Therefore on the original instance, the Largest-l procedure creates no less than k groups. In the meantime it creates no more than k groups because $n = kl + r$.

Now we already know there are k iterations of group forming procedures in total, denote them to be $I_1, I_2 \ldots I_k$. Assume one bucket (denoted B_{bad}) contains at least 2 tuples after I_k. Note before I_k, there are at most $l - 1$ buckets with

[3] Similar conclusion is stated in [6], however we find its proof incomplete because of the unproved assumption that the number of iterations equals k.

size at least 2, otherwise there will be at least l non-empty buckets after I_k. So a tuple from B_{bad} is selected during I_k and $|B_{bad}| \geq 3$ before I_k. Similarly, before I_{k-1}, there are at most $l-1$ buckets with size at most 3. So a tuple from B_{bad} is selected during I_{k-1} and $|B_{bad}| \geq 4$ before I_{k-1}. Recursively, we obtain $|B_{bad}| \geq k+2$ before I_1, this contradicts the condition. The proof is completed.

The above result is of great merits. When the assignment of diversity parameter l is smaller than or equal to $\lfloor \frac{n}{n_1} \rfloor$, the number of residual tuples are nicely bounded. However, when l grows greater than $\lfloor \frac{n}{n_1} \rfloor$, the number of residual tuples may grow dramatically. It's difficult to guarantee privacy for these tuples. Thereby, the assignment of l should not exceed $\lfloor \frac{n}{n_1} \rfloor$. To sum up, we have:

Corollary 2. *The largest permissable assignment to the diversity parameter l is $l_{per} = \lfloor \frac{n}{n_1} \rfloor$*

We can consider Corollary 2 from another angle. By just observing the overall distribution of S, the adversary can deduce with confidence $\frac{n_1}{n}$ that any target o has the most frequent value. This gives the limit of privacy preservation power. If some privacy preserving technique reduces the adversary's deduction confidence to lower than $\frac{n_1}{n}$, s/he can obtain more information just by the overall distribution. To sum up, we have:

(*The Limit of Privacy Preservation Power*) The limit of privacy preservation power for all techniques is the distribution of original values. One cannot achieve protection better than $\frac{n_1}{n}$ for all tuples if the original distribution is published.

If n is not a multiple of l, according to Corollary 1, there will be no more than $l-1$ tuples left. For each t of them, we choose a *proper* group to merge it.

5 Extending Decomposition to the MSA case

The extension of Decomposition to the MSA case is intuitive. First, as discussed in Section 2, the sensitive table T^S is published. Next, one sensitive attribute (denoted S^{pri}), is chosen as the "*primary sensitive attribute*" and largest-l procedure is exerted on S^{pri} to form SA-groups.

Definition 2. *(*Primary Sensitive Attribute*) In the MSA case, the* primary sensitive attribute *is the sensitive attribute chosen by the publisher, according to which SA-groups are formed.*

Third, for each SA-group and each non-primary sensitive attribute, the original values are united up, as depicted in Table 5. Reduplicated values are counted once because multiple counts just increase the privacy disclosure risk.

As discussed in Section 4, the limit of privacy preservation for each S^i is bounded by the most frequent S^i value's percentage. So, we should not assign a uniform l for all S^i. Instead, each S^i should have its own l_i. We may set $l_{per}(S^i)$ to be default. In this way, we introduce a new MSA diversity principle:

Definition 3. *($(l_1, l_2, \ldots, l_d)$-*diversity*) A decomposed table is said to satisfy $(l_1, l_2, \ldots, l_d)$-diversity, if for each of its SA-group G and each $i \in \{1, 2, \ldots, d\}$, $G.S^i$ contains at least l_i distinct sensitive values.*

Table 6. The Final Publishing of Decomposition

The Sensitive Table

Occupation	Salary
nurse	1
nurse	4
police	8
cook	9
actor	2
actor	7
clerk	8
clerk	2

The Decomposed Table after Adding Noise

Group#	Gender	ZipCode	Birthday	Occupation	Salary
1	F	10078	1988/04/17	police	1
	F	10085	1962/10/03	nurse	2
	M	20086	1958/06/06	actor	4
	M	10076	1985/03/01	clerk	8
2	F	10077	1984/03/21	nurse	2
	M	10085	1988/11/04	actor	4
	F	10075	1983/02/14	cook	7
	F	20087	1960/07/11	clerk	9

As for some non-primary sensitive attribute S^i, there may be groups with less than l_i distinct S_i values. Like in Group 1 of Table 5, $l_{per}(Salary) = \frac{8}{2} = 4$, because the most frequent salary value 2 and 8 both appear twice. However, Group 1 contains just 3 distinct values for Salary. To satisfy the privacy goal, some "*noise*" should be added. These noise values cannot be arbitrarily chosen. For example of Group 1, either 4 or 7 can be chosen while 9 cannot, we shall present the choosing method and its rationale in following subsections. In sum, the final publishing of *decomposition* is shown in Table 6

5.1 The Choice of Primary Sensitive Attribute

The primary sensitive attribute S^{pri} is publisher-predefined. Its introduction is not only necessary for group forming, but also of utility merits. Generally, S^i is retained to the maximum extent, because no noise is added on it. So publisher can choose the attribute whose data quality is specially required as S^{pri}.

In the meantime, since different S^i have different l_i and the size of each SA-group equals l_{pri}. If the attribute with largest l_i is not chosen as the primary sensitive attribute, for attributes with l_i larger than l_{pri}, the procedure of adding noise is inevitable. This is undesired. Therefore, we have:

(*Suggestion on Choice of Primary Sensitive Attribute*) Without exceptional requirement on data quality of some sensitive attribute, the S^i with largest l_i can be chosen as the primary sensitive attribute.

5.2 Adding Noise

The procedure of adding noise is conducted to compensate for SA-groups G that does not satisfy l_i-diversity for non-primary sensitive attribute S^i. This "not satisfying" may arise from two cases: (i) l_{pri} is not the largest of all l_i's. (ii) For $G.S^i$, reduplicated values are counted just once. However, noise values cannot be arbitrarily chosen, because the sensitive table T^S is also published. The adversaries can link $G.S^{pri}$ with T^S to detect which values are allowable for $G.S^i$, we term them linkable sensitive values. If a non-linkable sensitive value is chosen as noise, the adversary can perform link operation to eliminate it.

Definition 4. *(*Linkable Sensitive Values*) For non-primary sensitive attribute S^i and SA-group G, their linkable sensitive values, $LSV(S^i, G)$ are the set of S^i values resulted from natural link of $G.S^{pri}$ and T_S.*

$$LSV(S^i, G) = \Pi_{S^i}(T^S \bowtie G.S^{pri}) \tag{1}$$

In Table 5, $LSV(Salary, G_1) = \{1, 2, 4, 7, 8\}$ and 4 is chosen as noise value.

Since adding noise is a great data distortion, it should be reduced as much as possible. We revisit the process of group forming and introduce a greedy approach to reduce noise.

Definition 5. *(*Diversity Penalty*) The diversity penalty for a tuple t to be merged into group G, or $\mathcal{P}(t, G)$ is defined as:*

$$\mathcal{P}(t, G) = \sum_{\text{non-primary}S^i} p(t, G, S^i) \tag{2}$$

Where $p(t, G, S^i) = l_i - |G.S^i|$ If $t.S^i \in G.S^i$ and $G.S^i$ does not satisfies l_i-diversity. $|G.S^i|$ is the number of distinct $G.S^i$ values. $p(t, G, S^i) = 0$If $t.S^i \notin G.S^i$ or $G.S^i$ already satisfies l_i-diversity.

Namely, the diversity penalty penalize t which attempts to be merged into G, if it fails to contribute to achieving l_i-diversity for $G.S^i$. Revisit the Largest-l group forming procedure, in each iteration, one tuple is randomly selected from the largest bucket B_1 and forms the original group G, subsequently, from B_2 through B_l, one tuple that minimize $\mathcal{P}(t, G)$ is chosen and merged into G. As for the residual tuples when Largest-l procedure terminates, the choice of G to merge them it also based on minimizing $\mathcal{P}(t, G)$.

5.3 Algorithm

Summing up previous discussions, we formally present the algorithm of decomposition in this section. As shown in Figure 1, line 1 through line 8 depicts the Largest-l group forming procedure. The diversity penalty is used to reduce possible noise. The diversity penalty is also used in merging the residual tuples. Line 11 through line 15 shows the adding noise procedure.

6 Experiments

In this section, we experimentally evaluate the performance of decomposition. We utilize the "Adult" database from the UCI Machine Learning Repository[4]. It leaves 30162 tuples after removing tuples with missing value. All algorithms are built in JDK 5.0 and run on a dual-processor Intel Pentium D 2.8 GHz machine with 2GB RAM and Microsoft Windows Server 2003.

[4] http://www.ics.uci.edu/mlearn/mlrepository.html.

Algorithm 1: The Algorithm for Decomposition

Input: Original table T with sensitive attributes $S^1, S^2, \ldots, S^d$, one of them is primary: S^{pri}. Diversity Parameters $l_1, l_2, \ldots, l_d$.
Data: The set of buckets formed by primary sensitive attribute $\mathscr{B} = \{B_i\}$; $\mathcal{G} = \emptyset$, $\mathcal{G}$ is the set of SA-groups.
Output: Decomposed table T^* which satisfies $(l_1, l_2, \ldots, l_d)$-diversity.

```
1  begin
     /* The Largest-l group forming procedure                          */
2    while |ℬ| ≥ l_pri do
3      sort B_i in ℬ by their sizes in descent order;
4      Randomly remove one tuple t_1 from B_1;
5      G = {t_1};
6      for i ← 2 to l_pri do
7        Remove one tuple t_i from B_i that minimize 𝒫(t_i, G);
8        G = G ⋃ t_i;
9      endfor
10     𝒢 = 𝒢 ⋃ G;
11   endwhile
     /* Dealing with residual tuples                                   */
12   foreach residual tuple t do
13     Find SA-group G that minimize 𝒫(t, G);
14     G = G ⋃ t;
15   endforeach
     /* The adding noise procedure                                     */
16   foreach non-primary sensitive attribute S^i and each SA-group G do
17     if G.S^i does not satisfy l_i-diversity then
18       LSV(G, S^i) = Π_{S^i}(T^S ⋈ G.S^pri);
19       Merge values from LSV(G, S^i) − G.S^i into G.S^i
20       until G.S^i satisfies l_i-diversity;
21     end
22   endforeach
23 end
```

Fig. 1. The Decomposition Algorithm

Table 7. Description of Attributes

Attribute	Number of distinct values	Largest permissible diversity parameter
Age	73	N/A
Final-Weight	100	N/A
Martial Status	7	N/A
Race	5	N/A
Gender	2	N/A
Work-class	14	7
Education	16	3
Hours per Week	99	2
Relationship	6	3

There are 14 attributes in Adult. We retain 9 of them: *Age*, *Final-Weight*, *Martial Status*, *Race*, *Gender*, *Work-class*, *Education*, *Hours per Week* and *Relationship*. The descriptions of the attributes are in Table 7.

We adopt the *KL-divergence* metric which is widely used in the literals to measure the data utility. Concretely, given a probability distribution $\hat{F}_1$ associated with the original data, and a probability distribution $\hat{F}_2$ associated with the released anonymized data. Let $x_1, \ldots, x_N$ be the points in the muti-dimensional domain of the data. Let $p_i^{(1)}$ be the probability of x_i according to $\hat{F}_1$ and $p_i^{(2)}$ be the probability according to $\hat{F}_2$. The Kullback-Leibler (KL)-divergence between $\hat{F}_1$ and $\hat{F}_2$ is defined as $\sum_i p_i^{(1)} \log \frac{p_i^{(1)}}{p_i^{(2)}}$. It equals 0 only when $\hat{F}_1 = \hat{F}_2$.

6.1 Decomposition V.S. Generalization in the SSA Case

We treat *Work-class* as the sensitive attribute and develop 4 tables from Adult: q-QI-Adult ($5 \leq q \leq 8$). q-QI-Adult takes the first d of other attributes as QI.

We compare decomposition against the widely-adopted multi-dimensional generalization algorithm Mondrian[8] when achieving l-diversity. Figure 2 through Figure 5 depicts the KL-divergence of the anonymized datasets created by two algorithms. We could observe, for a fixed l, the KL-divergence of decomposed table almost does not grow with the number of QI attributes while the KL-divergence of generalized table grows significantly with QI attributes number. For any q-QI-Adult and any l, decomposition largely outperforms generalization.

For lack of space, we just plot the execution time of both algorithms on 8-QI-Adult in Figure 6. Again, decomposition greatly outperforms generalization. The execution time of decomposition almost have no growth with l while that of Mondrian generalization decreases a little, from 90.0 seconds to 75.7 seconds. In fact, we can theoretically prove that the time complexity of decomposition is $O(n^2)$, irrelevant with l. While larger l imposes more strict condition on continuing recursion for Mondrian, thereby reduces its execution time.

6.2 Decomposition in the MSA Case

To measure the effectiveness and efficiency of decomposition in the MSA case. We develop 4 tables: d-SA-Adult ($1 \leq d \leq 4$). d-SA-Adult uses the first 5 attributes as QI attributes and the subsequent d attributes as sensitive attributes. *Work-Class* is treated as primary sensitive attribute because it has largest l_{per}.

Figure 7 depicts the KL-divergence of decompositiond d-SA-Adult tables where l_{pri} is set from 3 to $l_{per}(work\text{-}class) = 7$. For each non-primary sensitive attribute S^i, l_i is set to $l_{per}(S^i)$. In Figure 7, each curve grows moderately with the increase in sensitive attribute numbers. In fact, the experimental result is quite close to the theoretical estimation of $\log(\prod_i l_i)$.

Figure 8 depicts the execution time of decomposition on d-SA-Adult tables. Again, each non-primary sensitive attribute is set to its largest permissible diversity parameter while l_{pri} varies from 3 to 7. When $d = 1$, decomposition is degenerated to the SSA case, the executions on different l cost almost the same

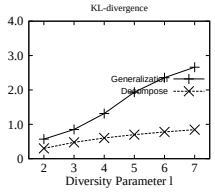

Fig. 2. 5-QI (SSA)

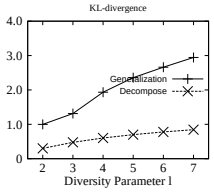

Fig. 3. 6-QI (SSA)

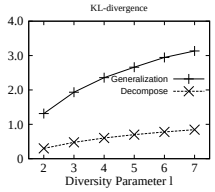

Fig. 4. 7-QI (SSA)

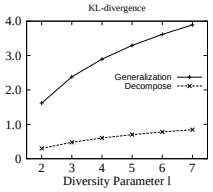

Fig. 5. 8-QI (SSA)

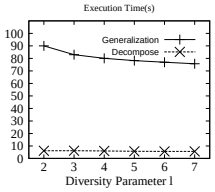

Fig. 6. time (SSA)

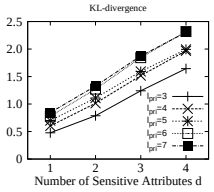

Fig. 7. MSA

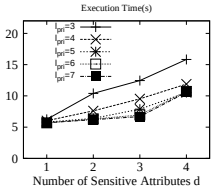

Fig. 8. time (MSA)

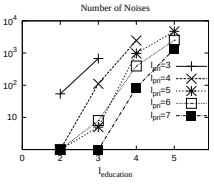

Fig. 9. Noise (MSA)

amount of time. This accords with Figure 6. When there are multiple sensitive attributes, larger l_{pri} leads to less execution time, because it reduces the necessity of adding noise.

We conduct a separate experiment to measure the number of noises in the MSA case. This experiment is on 2-SA-Adult, which takes *Work-Class* as the primary sensitive attribute and *Education* as the non-primary sensitive attribute. Figure 9 depicts the number of noises as the function of l_{pri} and $l_{Education}$. The largest number appears when $l_{pri} = 5$ and $l_{Education} = 5$, which equals 4733. In fact, this case will not appear in practise, because l_{per} for *Education* is 3 and we've reasoned that the assignment of l should not exceed l_{per}. When $l_{Education}$ is within the permissible range, the largest number of noises equals 675, a relatively quite small number comparing to the table size.

7 Related Work

Ever since Sweeney and Samarati introduce the idea of anonymization[1, 2], subsequent studies generally follow three directions: (1)developing new privacy-preservation models in face of new observed risks; (2)designing algorithms for certain privacy model; (3)and other related works.

Subsequent models include l-diversity[5], t-closeness[9], m-invariance[10], *personalization*[11], *dynamic anonymization* [12, 13] and so on.. Specially, *anatomy*[6] resort to non-generalization on QI attributes, which is extended in this paper.

Early algorithms for anonymization are *hierarchy-based*[1, 2, 4, 3], they assume a pre-defined generalization hierarchy for each QI attribute. [8] and [14] represent the *partition-based* and *clustering-based* algorithms respectively. The idea of largest-l procedure is also borrowed from [14].

There are still other related works, including privacy preservation on graphs[15], social networks[16], location information[17] and etc.

8 Conclusions

Although anonymization has received intensive studying interest in recent year, the privacy risk in the multiple sensitive attribute case has not been paid enough attention. We observe this new privacy threat and propose *decomposition* and the $(l_1, l_2, \ldots, l_d)$-diversity principle to tackle it.

This paper lays down a foundation for future works towards privacy preservation in the MSA case. Interesting directions include combining categorical and numerical sensitive attributes, working on dynamic dataset and etc..

References

1. P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *PODS*, 1998, p. 188.
2. L. Sweeney, "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, 2002.
3. K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain k-anonymity," in *SIGMOD*, 2005, pp. 49–60.
4. V. Iyengar, "Transforming data to satisfy privacy constraints," in *SIGKDD*, 2002, pp. 279–288.
5. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: privacy beyond k-anonymity," in *ICDE*, 2006, pp. 24–26.
6. X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in *VLDB*, 2006, pp. 139–150.
7. N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, "Aggregate query answering on anonymized tables," in *ICDE*, 2007, pp. 116–125.
8. K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian: multidimensional k-anonymity," in *ICDE*, 2006, p. 25.
9. N. Li and T. Li, "t-closeness: privacy beyond k-anonymity and l-diversity," in *ICDE*, 2007, pp. 106–115.
10. X. Xiao and Y. Tao, "m-invariance: towards privacy preserving re-publication of dynamic datasets," in *SIGMOD*, 2007, pp. 689–700.
11. X. Xiao and Y. Tao, "Personalized privacy preservation," in *SIGMOD*, 2006, pp. 229–240.
12. K. Wang and B. C. M. Fung, "Anonymizing sequential releases," in *SIGKDD*, 2006, pp. 414–423.
13. J. W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure anonymization for incremental datasets," in *SDM*, 2006, pp. 48–63.
14. Y. Ye, Q. Deng, C. Wang, D. Lv, Y. Liu, J. Feng, "BSGI: An Effective Algorithm towards Stronger l-Diversity,". in *DEXA*, 2008, pp. 19–32.
    G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving anonymity via clustering," in *PODS*, 2006, pp. 153–162.
15. K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *SIGMOD*, 2008.
16. B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *ICDE*, 2008.
17. G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan, "Private queries in location based services: anonymizers are not necessary," in *SIGMOD*, 2008.