SLOMS: A Privacy Preserving Data Publishing Method for Multiple Sensitive Attributes Microdata

Jianmin Han
Department of Computer Science and Technology
Zhejiang Normal University, Jinhua, 321004, Zhejiang, PRC
hanjm@zjnu.cn
Fangwei Luo，Jianfeng Lu and Hao Peng
Department of Computer Science and Technology
Zhejiang Normal University, Jinhua, 321004, Zhejiang, PRC

***Abstract*—Multi-dimension bucketization is a typical method to anonymize multiple sensitive attributes. However, the method leads to low data utility when microdata have more sensitive attributes. In addition, the methods do not generalize quasi-identifiers, which make the anonymous data vulnerable to suffer from linked attacks. To address the problems, the paper proposes a SLOMS method. The method vertically partitions the multiple sensitive attributes into several tables and bucketizes each sensitive attribute table to implement *l*-diversity. At the same time, it generalizes the quasi-identifiers to implement *k*-anonymity. The paper also proposes a MSB-KACA algorithm to anonymize microdata with multiple sensitive attributes by SLOMS. Experiments show that SLOMS can generate anonymous tables with less suppression ratio and less distortion compared with generalization and MSB.**

***Index Terms*—*k*-anonymity, *l*-diversity, multi-dimension bucketization method, SLOMS**

I. INTRODUCTION

Microdata play an increasingly important role in data analysis and scientific research. However, publishing and sharing microdata will threaten individuals' privacy. Therefore, some anonymity models have been proposed to protect individual's privacy for microdata publish recently. *k*-anonymity [1] is a simple and effective method to protect privacy in microdata, which requires that each tuple has at least *k* indistinguishable tuples with respect to quasi-identifier in the released data. But it cannot resist homogeneity attack and background knowledge attack, so some other enhanced anonymity models have been proposed, such as *l*-diversity [4] and *t*-closeness [5].

Several techniques have also been proposed to implement the above anonymity models. Generalization [1-3] is a typical one to implement anonymity model, whose idea is to replace real value of quasi-identifier with less specific but semantically consistent value. Generalization distorts original data, which is disadvantageous to data mining. Anatomy [6] is also a fine method to anonymize microdata, whose idea is to release all the quasi-identifier and sensitive values directly in two separate tables. However, releasing the QI-values directly may suffer from a higher breach probability than generalization. To overcome these drawbacks, Tao et al. [7] proposed ANGEL, a new anonymization method that is as effective as generalization in privacy protection, which can retain higher data utility. Leela et al. [8] applied Angelization to preserve privacy in re-publication of dynamic microdata after insertions or deletions. Li et al. [9] proposed slicing, which anonymizes microdata by partitioning microdata horizontally and vertically. Neha et al. [10] concluded that slicing preserves data utility better than generalization, in addition, it also prevents membership disclosure.

All of above works focus on microdata with single sensitive attribute. These methods will lead to much low data utility when they are directly used for microdata with multiple sensitive attributes. At present, there is only a few work concentrated on microdata with multiple sensitive attributes. Yang et al. [11] proposed a **M**ultiple **S**ensitive **B**ucketization(**MSB**) approach. But the MSB method is only suitable to deal with microdata with less sensitive attributes, e.g, 2 to 3 sensitive attributes. For microadata with more sensitive attributes, MSB would result in high suppression ratios. For example, table I is an original dataset. We assume that {*Gender*, *ZipCode*, *Age*} are quasi-identifier attributes and {*Occupation*, *Salary*, *Physician*, *Disease*} are sensitive attributes. We can achieve a 3-diversity table by MSB, seeing table II. The anonymity table only has one group with tuples {t_5, t_6, t_7} presented in table II, the rest tuples are all suppressed. The suppression ratio is 6/9, which greatly degrades the quality of data publishing.

doi:10.4304/jsw.8.12.3096-3104

TABLE I.
ORIGINAL DATA WITH MULTIPLE SENSITIVE ATTRIBUTES

Tuple	Gender	ZipCode	Age	Occupation	Salary	Physician	Disease
t_1	M	31200	23	clerk	4000+	John	Gastritis
t_2	F	32100	27	clerk	6000+	Bob	Asthma
t_3	M	31204	24	cook	4000+	Bob	Flu
t_4	M	42000	31	teacher	8000+	John	Flu
t_5	F	32100	29	teacher	6000+	Lucy	Cancer
t_6	M	42005	35	cook	10000+	John	Flu
t_7	M	42004	31	police	4000+	Tom	Asthma
t_8	F	32004	30	clerk	8000+	Tom	Cancer
t_9	F	31205	26	teacher	10000+	Tom	Asthma

TABLE II.
ANONYMIZED DATA USING MSB

Tuple	Occupation	Salary	Physicion	Diease
t_5	teacher	6000+	Lucy	Cancer
t_6	cook	10000+	John	flu
t_7	police	4000+	Tom	asthma

In this paper, we propose a method, called SLOMS (**SL**cing [9] **O**n **M**ultiple **S**ensitive). The main idea of SLOMS is to vertically partition attributes into several sensitive attribute tables and one quasi-identifier table. Tuples in each table are partitioned into some equivalence classes. The quasi-identifier values of each equivalence class are generalized to the same value under *k*-anonymity principle. Meanwhile, the sensitive values of each sensitive table are sliced and bucketized to obey the *l*-diversity requirement.

II. SLOMS

In this section, we first give an example to illustrate SLOMS, and then introduce some basic notations and definitions in Section *A*. Section *B* formalizes SLOMS, and compares it with MSB and generalization, section *C* discusses how to partition sensitive attributes, and Section *D* discusses privacy threats that SLOMS can resist.

For example, table I contains 3 quasi-identifier attributes and 4 sensitive attributes. We can vertically partition table I into three tables. The first one is a 3-diversity table with 2 sensitive attributes, seeing in table III. The second one is a 3-diversity table with another 2 sensitive attributes, seeing in table IV. The last one is a generalized table with 3 quasi-identifier attributes. We cluster quasi-identifier attributes into three batches. Each one has three tuples, seeing in table V. We then generalize the quasi-identifier attributes to make the quasi-identifier attribute table conform to 3-anonymity and insert 2 columns of sensitive values' group ID, seeing in table VI. SLOMS publishes the microdata of table III, table IV and table VI.

Observe that each tuple is anonymized into three parts: values of {*Gender*, *ZipCode*, *Age*} belong to table VI, values of {*Occupation*, *Salary*} belong to table III and values of {*Physician*, *Disease*} belong to table IV. The {*O-S*, *P-D*} columns in table VI connect the quasi-identifier attributes and sensitive attributes for the data utility. Since all tuples are published in SLOMS method, the suppression ratio is 0, which preserves higher data quality than MSB.

TABLE III.
O-S (OCCUPATION AND SALARY) SENSITIVE ATTRIBUTES

Tuple	Group	Occupation	Salary
t_1	1	clerk	4000+
t_5		teacher	6000+
t_6		cook	10000+
t_8	2	clerk	8000+
t_3		cook	4000+
t_9		teacher	10000+
t_4	3	teacher	8000+
t_7		police	4000+
t_2		clerk	6000+

TABLE IV.
P-D (PHYSICIAN AND DISEASE) SENSITIVE ATTRIBUTES

Tuple	Group	Physician	Disease
t_1	1	John	Gastritis
t_3		Bob	Flu
t_7		Tom	Asthma
t_2	2	Bob	Asthma
t_4		John	Flu
t_8		Tom	Cancer
t_5	3	Lucy	Cancer
t_6		John	Flu
t_9		Tom	Asthma

TABLE V.
QUASI-IDENTIFIER ATTRIBUTES

Tuple	Gender	ZipCode	Age
t_1	M	31200	23
t_3	M	31204	24
t_9	F	31205	26
t_2	F	32100	27
t_5	F	32100	29
t_8	F	32004	30
t_4	M	42000	31
t_6	M	42005	35
t_7	M	42004	31

TABLE VI.
GENERALIZED QUASI-IDENTIFIER ATTRIBUTES

Tuple	Gender	ZipCode	Age	O-S	P-D
t_1	*	3120*	[23,26]	1	1
t_3	*	3120*	[23,26]	2	1
t_9	*	3120*	[23,26]	2	3
t_2	F	32***	[27,30]	3	2
t_5	F	32***	[27,30]	1	3
t_8	F	32***	[27,30]	2	2
t_4	M	4200*	[31,35]	3	2
t_6	M	4200*	[31,35]	1	3
t_7	M	4200*	[31,35]	3	1

A. Notations And Definitions

Let T be a dataset with attributes $\{a_1, a_2, \cdots, a_z, s_1, s_2, \cdots, s_d\}$, where $(a_1, a_2, ..., a_z)$ are quasi-identifier attributes, $(s_1, s_2, ..., s_d)$ are sensitive attributes, d is the number of sensitive attributes. For simplicity, we use *QI* to denote quasi-identifier, and *S* to denote sensitive attributes. We assume that T has n tuples, and $t[X]$ is the value of tuple t on attribute X.

Definition 1 (Partition [6]). $G_1, G_2, \cdots\cdots, G_c$ are a partition of T, if and only if for $\forall i, j(1 \le i \ne j \le c)$,

satisfy $\bigcup_{i=1}^{c} G_i = T$ and $G_i \cap G_j = \varnothing$, where G_i is a set of tuples in T.

Definition 2 (Single sensitive attribute l-diversity). Let T be a single sensitive attribute table, $G_1,G_2,\ldots,G_c$ be a partition of T, v_i be the value of a sensitive attribute($i \le d$), $c(v_i)$ be the number of sensitive values v_i in a group G_j, $|G_j|$ be the number of tuples in G_j. if for $\forall v_i \in G_j$, $\frac{c(v_i)}{|G_j|} \le \frac{1}{l}$, then G_j satisfies single sensitive attribute l-diversity. If $G_1,G_2,\ldots\ldots,G_c$ all satisfy single sensitive attribute l-diversity, the table T satisfies single sensitive attribute l-diversity.

Definition 3 (Multiple sensitive attributes l-diversity). Given a multiple sensitive attributes table T, if all of sensitive attributes in T satisfy single sensitive attribute l-diversity, then T satisfies multiple sensitive attributes l-diversity.

B. Formalization Of SLOMS And Comparison

SLOMS splits T into one quasi-identifier table *(QIT*) and m sensitive attribute tables ($\{ST_1, ST_2, \cdots, ST_m\}$ $(1 \le m \le d)$). The *QIT* table contains generalization of all quasi-identifier attributes and m columns group ID of each sensitive attribute. The ST_i $(1 \le i \le m)$ tables contain sensitive values and their group ID. If one sensitive value is suppressed, it is replaced by "NA".

Definition 4 (SLOMS). Given a multiple sensitive attributes table T, a SLOMS of T is given by one *QIT* and m ST_i $(1 \le m \le d)$.

For example, firstly, SLOMS splits sensitive attributes of table I into two parts, i.e. {*Occupation*, *Salary*} and {*Physician*, *Disease*}. Secondly, SLOMS partitions sensitive values into groups to implement 3-*diversity*, i.e. O-S (*Occupation* and *Salary*) sensitive attributes table, seeing table III and P-D (*Physician* and *Disease*) sensitive attributes table, seeing table IV. Thirdly, SLOMS groups quasi-identifier part into clusters with no less than k tuples, seeing table V, and generalizes each clusters and inserts 2 columns of sensitive values' group ID, seeing in table VI. Finally, table III, table IV and table VI are published as result.

Now, let us compare SLOMS with MSB. First, SLOMS vertically partitions sensitive attributes in order to decrease suppression ration as we describe in our examples. Second, by releasing the quasi-identifier values directly, MSB may suffer a higher breach probability than SLOMS. Nevertheless, such probability is always bounded by 1/l, as long as the background knowledge of an adversary is not stronger than the level allowed by the l-diversity model. However, SLOMS generalizes quasi-identifier values gain another breach probability x which is the probability an adversary can sure someone is to be involved in the microdata. The value of x is always less than or equal to 1. The overall breach probability of SLOMS is $x \times \frac{1}{l} \le \frac{1}{l}$. Our goal is to sacrifice a little information loss on quasi-identifier attributes for higher privacy guarantee.

If m is 1 and k is 1, SLOMS is actually gracefully degraded into MSB, just one sensitive attribute table needed. We can say that SLOMS is a superset of MSB. When d is small, there is no denying that MSB has good publishing data with better data utility. We put emphasis on dataset which have a large number of sensitive attributes.

To compare SLOMS with generalization, we first formalize generalization obeying the same anonymization principle which is l-diversity.

Definition 5 (l-diversity generalization). Let $G_1,G_2,\cdots\cdots,G_m$ be partitions of T, we say that T is a l-diversity generalization if for all $i = 1\cdots m$, the quasi-identifier attributes in G_i are generalized into the same value with the sensitive attributes satisfying Multiple sensitive attributes l-diversity.

If T is generalized to satisfy multiple sensitive attributes l-diversity, some tuples need to be suppressed for satisfying l-diversity. For example, table I obeying the 3-diversity generalization, there are only three tuples reserved, namely t_5, t_6, t_7, others are all suppressed. The values of quasi-identifier attributes $\{t_5, t_6, t_7\}$ are all generalized into {*, *****, 29-35}. Suppression ratio and information loss will increase greatly when we anonymize multiple sensitive attributes dataset using generalization. Intuitively, generalization is not suitable for multiple sensitive attributes dataset. We would like to emphasize that our goal is not to deny generalization.There is no doubt that generalization is an important technique in single sensitive attributes dataset and it has been received much attention in lots of literatures. Instead, our intention is to combine generalization with Slicing to maximize data utility of anonymous data for handling multiple sensitive attributes dataset.

C. Sensitive Attributes Partition

Firstly, SLOMS should partition sensitive attributes into m parts. SLOMS partitions sensitive attributes according to the principle that highly correlated sensitive attributes are in the same table, because grouping highly correlated sensitive attributes preserves the correlations among those sensitive attributes. Therefore, we first compute the correlations between pairs of sensitive attributes and then cluster attributes based on their correlations.

Two widely used measures of correlations are Pearson correlation coefficient [13] and mean-square contingency coefficient [13]. We adopt the mean-square contingency coefficient because most of our sensitive attributes are categorical and the mean-square contingency coefficient is a chi-square measure of correlation between two categorical attributes. Given two attributes s_1 and s_2 with domains $\{v_{11}, v_{12}, \ldots, v_{1p}\}$ and $\{ v_{21}, v_{22}, \ldots, v_{2q} \}$, respectively. p and q are their domain size, respectively. The mean-square contingency coefficient between s_1 and s_2 is defined as (1).

$$\Phi^2(s_1,s_2)=\frac{1}{\min\{p,q\}-1}\sum_{i=1}^{p}\sum_{j=1}^{q}\frac{(f_{ij}-f_{i\cdot}f_{\cdot j})^2}{f_{i\cdot}f_{\cdot j}} \quad (1)$$

Here, $f_{i\cdot}$ and $f_{\cdot j}$ are the fraction of occurrences of v_{1i} and v_{2j} in the data, respectively. f_{ij} is the fraction of cooccurrences of v_{1i} and v_{2j} in the data. Therefore, $f_{i\cdot}$ and $f_{\cdot j}$ are the marginal totals of f_{ij}, i.e., $f_{i\cdot}=\sum_{j=1}^{q}f_{ij}$ and $f_{\cdot j}=\sum_{i=1}^{p}f_{ij}$. Obviously, $0\le\Phi^2(s_1,s_2)\le 1$.

If there are some continuous sensitive attributes, we regard them as categorical attributes after being discretized.

After computing the correlations for each pair of sensitive attributes, we use clustering to partition sensitive attributes into tables. Each sensitive attributes in our algorithm is a point in the clustering space. The distance between two attributes in the clustering space is defined as $d(s_1,s_2)=1-\Phi^2(s_1,s_2)$, which is in interval [0-1]. For two sensitive attributes, the smaller distance between the corresponding data points, the stronger correlation they are.

We use the well-known *k*-medoid algorithm PAM (Partition Around Medoids) for clustering [14]. PAM begins with *k* randomly data points as the initial medoids. In each subsequent step, PAM chooses one medoid point and one nonmedoid point and swaps them as long as the cost of clustering decreases. Here, the clustering cost is measured as the sum of the cost of each cluster, which is in turn measured by the sum of the distance from each data point in the cluster to the medoid point of the cluster. Its time complexity is $O(k(m-k)^2)$ which is a high-computational complexity for large data sets. However, the data points in our clustering space are sensitive attributes, whose number is small.

D. Privacy Preservation

There are three types of privacy disclosure threats of microdata publishing. The first one is membership disclosure, which occurs when adversaries can infer whether one's record is included in the dataset from publishing microdata. The second one is identity disclosure, which occurs when an individual is linked to some records in multiple released datasets. This type of threat was illustrated in [2], where adversaries can join a public voter registration list and the de-identified patient data of Massachusetts's state employees to determine the medical history of the state's governor. The third one is attribute disclosure, which occurs when new information about some individuals is revealed, i.e., the released data make it possible to infer the attributes value of an individual.

We consider that SLOMS can resist all these privacy disclosure threats. *k*-anonymity is an excellent model to protect membership disclosure and identity disclosure. We generalize the quasi-identifier attributes to conform the requirement of *k*-anonymity. For the attribute disclosure, *l*-diversity is used to prevent sensitive attributes from disclosure. We partition sensitive attributes to buckets, obeying the requirement of *l*-diversity.

III. Evaluation Of Anonymous Data

There are two aspects of information loss, the first one occurs in generalization, the second one occurs in anatomy.

A. Measurement For Generalization

We adapt the distortion [12] to measure the quality of generalization. We make some definitions to explain it.

Definition 6 (Weighted Hierarchical Distance). Let *h* be the height of a domain hierarchy, and let levels 1, 2, ⋯, *h*-1, *h* be the domain levels from the most general to most specific, respectively. When a cell is generalized from level *p* to level *q*, where $p > q$, the weighted hierarchical distance of this generalization is defined as (2).

$$WHD(p,q)=\frac{\sum_{j=q+1}^{p}w_{j,j-1}}{\sum_{j=2}^{h}w_{j,j-1}} \quad (2)$$

where $w_{j,j-1}=1/(j-1)\ (2\le j\le h)$.

Take ZipCode as an example. Let ZipCode hierarchy be {11323, 1132*, 113**, 11***, 1****, *****}, *WHD* from 1132* to 11*** is *WHD*(5, 3) = (1/4+1/3)/(1/5+1/4+1/3+1/2 +1) = 0.255.

Definition 7 (Distortion of generalization of tuples). Let $t = \{v_1, v_2, \ldots, v_z\}$ be a tuple with *z* quasi-identifier values and $t' = \{v'_1, v'_2, \ldots, v'_z\}$ be a generalized tuple of *t*. Let level(v_j) be the domain level of of v_j in an attribute hierarchy. The distortion of this generalization is defined as (3).

$$Distortion(t,t')=\sum_{j=1}^{z}WHD(level(v_j),level(v'_j)) \quad (3)$$

Definition 8 (Distortion of generalization of tables). Let quasi-identifier table T_q be generalized to T'_q and t'_i be generalization tuple of t_i, the distortion of T'_q is defined as (4).

$$Distortion(T_q,T'_q)=\sum_{i=1}^{n}Distortion(t_i,t'_i) \quad (4)$$

Definition 9 (Closet common generalization). All allowable values of an attribute form a hierarchical value tree. Each value is represented as a node in the tree, and a node has a number of child nodes corresponding to its more specific values. t_{12} is the closest common generalization of t_1 and t_2, and its value is defined as (5).

$$v_{12}^{i}=\begin{cases}v_1^i, & if\ (v_1^i=v_2^i)\\ \text{the value of the closet common ancestor}, & if\ (v_1^i\ne v_2^i)\end{cases} \quad (5)$$

where, v^i_1, v^i_2, and v^i_{12} are the values of the *i*-th attribute in tuples t_1, t_2 and t_{12}.

Definition 10 (Distance between two tuples). Let t_{12} be the closest common generalization of t_1 and t_2. The distance between t_1 and t_2 is defined as (6).

$$dist(t_1,t_2)=Distortion(t_1,t_{12})+Distortion(t_2,t_{12}) \quad (6)$$

Definition 11 (Distance between two equivalence partitions). Let g_1 and g_2 be two equivalence classes of quasi-identifier attributes table. They have n_1 identical tuples t_1 and n_2 identical tuples t_2, respectively. The distance between g_1 and g_2 is defined as (7).

$$dist(g_1, g_2) = n_1 \times dist(t_1, t_{12}) + n_2 \times dist(t_2, t_{12}) \quad (7)$$

The distance is equivalent to the distortions of the generalization and therefore the choice of merger should be those equivalence classes with the smallest distances.

B. Measurement For Anatomy

Now, let us discuss the metrics of sensitive attributes.

Definition 12 (Additional information loss [11]). Let G_i be a group in a l-diversity table, b be the number of groups in the l-diversity table, then Additional information loss of one sensitive attribute in group G_i can be defined as $\sum_{i=1}^{b} \frac{|G_i| - l}{b \times l}$. The Additional information loss of whole table is defined as (8).

$$AddInfoLoss = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{b_j} \frac{|G_i| - l}{b \times l} \quad (8)$$

where m is the number of sensitive attributes.

Ideally, every tuple in original table should be assigned to one group in anonymity table, however, for the restriction of l-diversity, some tuples cannot belong to any group. These tuples should be suppressed.

Definition 13 (Suppression ratio). Let n_s be the number of suppressed tuples, then the suppression ratio is defined as (9).

$$SuppRatio = \frac{n_s}{n} \quad (9)$$

Intuitively, the smaller *SuppRatio*, the higher the quality of anonymized data. Obviously, when the Suppression ratio is 0, the quality of anonymized data is optimal, for example table III and table IV. We consider a tuple as a suppressed tuple when it has at least one suppressed value. We use "NA" to replace its group ID. For instance, we have a tuple {M,4200*,[31,35],1,NA}. The first three values stand for generalized quasi-identifier values, the fourth value stands for a group ID, and the fifth value stands for a suppressed sensitive value.

IV. SLOMS ALGORITHM

In this section, we present an efficient MSB-KACA algorithm to achieve SLOMS. Given a dataset T and three parameter m, l and k, the algorithm computes the result table that consists of one generalized table satisfying the requirement of k-anonymity and m tables satisfying the requirement of multiple sensitive attributes l-diversity.

A. The MSB-KACA Algorithm

The MSB-KACA algorithm provides an implementation pattern for SLOMS. First, the algorithm vertically partitions dataset T into a quasi-identifier table and m sensitive attribute tables accordingly. Second, we use MSB method to implement l-diversity for each sensitive table and KACA algorithm [12] to implement k-anonymity for quasi-identifier table. Finally, link generalized table and ST_i according to tuple ID to form *QIT*. After these steps, *QIT* and $\{ST_1, ST_2 \cdots\cdots ST_m\}$ are achieved.

The algorithm vertically partitions the sensitive attributes into m parts according to the correlation of sensitive attributes, seeing section 2.3. For convenience, we create a sensitive attributes classification data structure Y to record how the sensitive attributes are partitioned. For example, $Y\{1, 1, 2, 2\}$ means the first two sensitive attributes belong to ST_1, and the others belong to ST_2.

The basic idea of MSB method is: (1) assign each tuple to a bucket according to each sensitive attribute values of the tuple, each bucket has the same values on all sensitive attributes, and assign a priority to each bucket according to some policies; (2) randomly choose a tuple in the highest priority bucket; (3) shield all buckets which have the same value with the selected tuple in any dimension ; (4) repeat (2)(3) until a l-diversity group is built.

There are three liner-complexity greedy algorithms to implement MSB[11], namely, the maximal-bucket first algorithm (MBF), the maximal single-dimension-capacity first algorithm (MSDCF), and the maximal multiple dimension-capacity first algorithm (MMDCF). The differences of these three algorithms are the policies of calculating bucket priority. In MBF, the maximal bucket is chosen as the criterion while MSDCF choose maximal single-dimension capacity and MMDCF choose maximal multi-dimension capacity.

The selection priority of the maximal bucket is defined as (10).

$$Seclection(buk < s_0^1, s_0^2, \cdots, s_0^d >) = size(buk < s_0^1, s_0^2, \cdots, s_0^d >) \quad (10)$$

where $size(buk < s_0^1, s_0^2, \cdots, s_0^d >)$ is the number of tuples in bucket $buk < s_0^1, s_0^2, \cdots, s_0^d >$.

The selection priority of the maximal single-dimension capacity is defined as (11).

$$Seclection(buk < s_0^1, s_0^2, \cdots, s_0^d >) = Max_{1 \le j \le d} Capa(s_0^j) + size(buk < s_0^1, s_0^2, \cdots, s_0^d >) \quad (11)$$

where $Max_{1 \le j \le d} Capa(s_0^j)$ is the maximal number of tuples in each dimension bucket, $size(buk < s_0^1, s_0^2, \cdots, s_0^d >)$ is the size of the bucket $buk < s_0^1, s_0^2, \cdots, s_0^d >$.

The selection priority of the maximal multi-dimension capacity is defined as (12).

$$Seclection(buk < s_0^1, s_0^2, \cdots, s_0^d >) = \sum_{1 \le j \le d} Capa(s_0^j) + size(buk < s_0^1, s_0^2, \cdots, s_0^d >) \quad (12)$$

where $\sum_{1 \le j \le d} Capa(s_0^j)$ is the maximal sum of the number of tuples in each dimension bucket, $size(buk < s_0^1, s_0^2, \cdots, s_0^d >)$ is the size of the bucket $buk < s_0^1, s_0^2, \cdots, s_0^d >$.

For example, *Occupation* and *Salary* are chosen as sensitive attributes in table I. In the first place, we make a 2-dimension bucket of {*Occupation* and *Salary*}, illustrated in figure 1. In MMDCF the maximal *selection* of bucket<clerk, 4000+> is 7 according to formula (12) (there are 3 tuples in 4000+ dimension, 3 tuples in clerk dimension and 1 tuple in *buk*<clerk, 4000+>, totally is 7), which is the highest priority, then we shield dimension <clerk> and <4000+>. Repeat the procedure, we can get bucket<teacher, 6000+> whose *selection* is 6 and bucket<cook, 10000+> whose *selection* is 5. Therefore, $\{t_1, t_5, t_6\}$ are chosen as a partition conforming to 3-diversity.

	Teacher	Cook	Clerk	Police
4000+		$\{t_3\}$	$\mathbf{\{t_1\}}$	$\{t_7\}$
6000+	$\mathbf{\{t_5\}}$		$\{t_2\}$	
8000+	$\{t_4\}$		$\{t_8\}$	
10000+	$\{t_9\}$	$\mathbf{\{t_6\}}$		

Figure 1. 2-dimension bucket of {*Occupation* and *Salary*}

The detail of MSB-KACA algorithm is shown in algorithm 1.

Algorithm 1: MSB-KACA Algorithm

Input:
Original dataset $T\{a_1,a_2,\cdots\cdots,a_z,s_1,s_2\cdots\cdots,s_d\}$, parameter l and k, sensitive attributes classification table $Y\{y_1,y_2,\cdots\cdots y_d\}$
Output:
QIT and $\{ST_1, ST_2\cdots\cdots ST_m\}$
Procedure:
1. begin
2. split dataset T into a quasi-identifier tables and m sensitive attribute tables according to sensitive attributes classification table Y;
3. assign each tuple an ID sequentially from 1 to n for every table;
4. for i=1 to m do
5. create ST_i which satisfies l-diversity using MSB technology;
6. end for
7. form initial equivalence classes from the quasi-identifier table;
8. while there exists an equivalence class of size < k do
9. randomly choose an equivalence class C of size < k;
10. evaluate the pairwise distance between C and all other equivalence classes;
11. find the equivalence class C' with the smallest distance to C;
12. generalize the equivalence classes C and C';
13: end while
14. link generalized table and ST_i according to tuple ID to form *QIT*;
15.end

Now we consider the time complexity of MSB-KACA algorithm. The time of partitioning dataset into m+1 tables just need liner time complexity, which is $O(n)$. In bucketization step, which is to partition sensitive attributes tables into buckets, MMDCF is a liner-complexity greedy algorithm with $O(n)$ time complexity. In generalization step, which is to generalize quasi-identifier attributes into equivalence classes, the KACA algorithm runtime is $O(n\log n+|E|^2)$ [12]. Therefore, the total time complexity of MSB-KACA algorithm is $O(n) + O(n) + O(n\log n+|E|^2) \approx O(n\log n+|E|^2)$.

V. EXPERIMENTS

A. Experimental Environment And Data

This section experimentally evaluates the effectiveness of our approach using the Adult Database from the UCI Machine Learning Repository which we can download at http://kdd.ics.uci.edu/databases/census-income/. We randomly choose 10000 records. The description of Census-Income data is shown in table VII. The experiments are conducted on a PC with CPU 3.0 GHZ and RAM 2GB. All the algorithms are implemented in Java on Windows XP with JDK version 1.6.0_23.

TABLE VII.
DESCRIPTION OF CENSUS-INCOME DATA

Number	Attribute name	Attribute type	Distinct values	Height
1	Age	Numeric	91	5
2	Sex	Categorical	2	2
3	Race	Categorical	5	3
4	Marital state	Categorical	7	3
5	Employment state	Categorical	8	3
6	Occupation	Sensitive	50	/
7	Industry	Sensitive	52	/
8	Workclass	Sensitive	10	/
9	Education	Sensitive	17	/
10	Major Occupation code	Sensitive	24	/
11	Major industry code	Sensitive	15	/

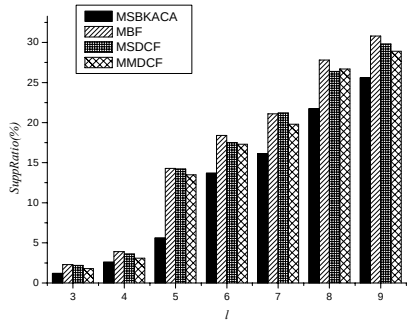
Our experiments mainly use following 4 criteria to evaluate MSB-KACA algorithm: (1) diversity parameter l (3~9); (2) sensitive attributes number d (3~6); (3) dataset size n (2000~10000); (4) quasi-identifier size $|QI|$ (2~5). The combination of sensitive attributes is described in table VIII. Taking d=5 as an example, we treat attribute numbers {6, 7, 8, 9, 10} as the sensitive attributes when implementing MBF, MSDCF, MMDCF. In the meantime, we treat attribute numbers {6, 7, 8} as the first sensitive table and attribute numbers {9, 10} as the second sensitive table when implementing MSB-KACA.

TABLE VIII.
MULTIPLE SENSITIVE ATTRIBUTES

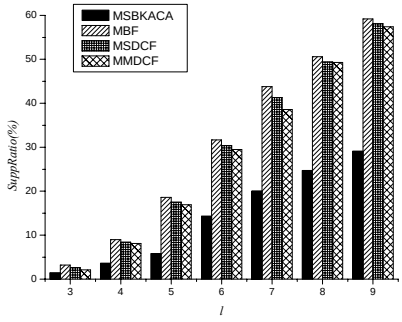
d	MBF,MSDCF,MMDCF	MSB-KACA
d=3	{6,7,8}	{{6,7},{8}}
d=4	{6,7,8,9}	{{6,7}，{8,9}}
d=5	{6,7,8,9,10}	{{6,7,8}，{9,10}}
d=6	{6,7,8,9,10,11}	{{6,7,8}，{9,10,11}}

B. Analysis Of Suppression Ratio

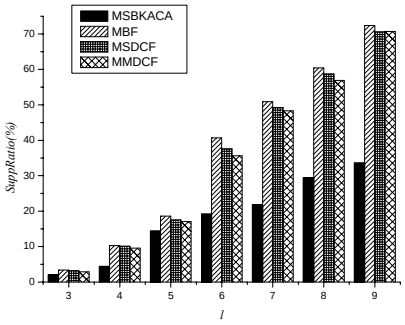
Figure 2(a), 2(b), 2(c) and 2(d) plot the variation of suppression ratio using MSB-KACA,MBF,MSDCF and MMDCF on different l when n is 10000 and d is 3-6. Apparently, MSB-KACA produces significantly lower suppression ratio than other three algorithms in all cases. This is expected, since SLOMS reduces the dimensions of buckets, which makes it easy to form l-diversity group. We can also observe that the suppression ratio grows rapidly with the increasing of parameter l. Because when l is increasing, the demands of diversity in buckets are also growing. It is harder to partition sensitive values, the higher suppression ratio generated.



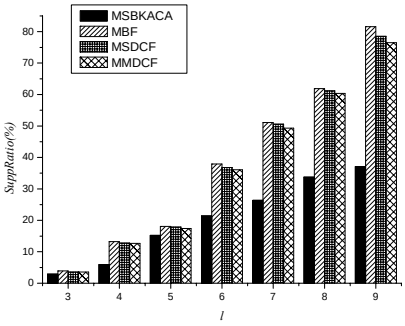
(a) Suppression ratio for various parameter l (d=3,n=10000)



(b) Suppression ratio for various parameter l (d=4,n=10000)



(c) Suppression ratio for various parameter l (d=5,n=10000)

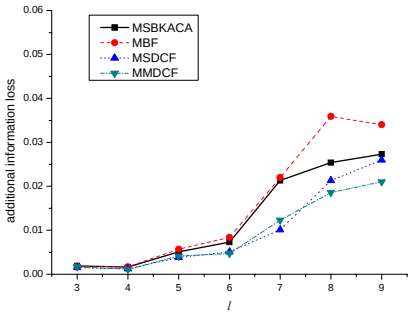


(d) Suppression ratio for various parameter l (d=6,n=10000)

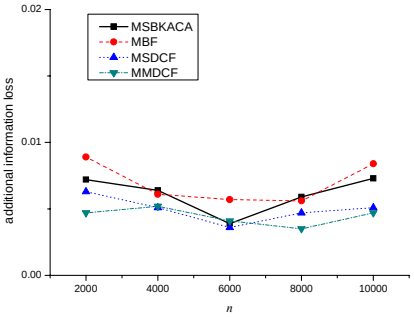
Figure 2. Comparison of Suppression ratio for the four algorithms.

C. Analysis Of Additional Information Loss

Figure 3(a) illustrates the variation of additional information loss using MSB-KACA,MBF,MSDCF and MMDCF on different l when n is 10000 and d is 4,which is calculated by formula(8). Figure 3(b) illustrates the variation of additional information loss on different n when l is 6 and d is 4. We can conclude that all the four algorithms produce low additional information loss.



(a) parameter l (d=4,n=10000)

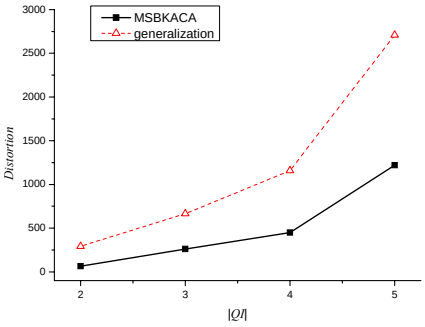


(b) tuples number n (d=4,l=6)

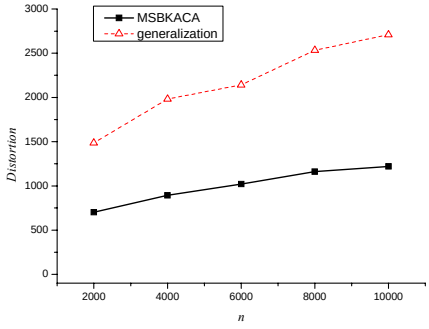
Figure 3. Comparison of Additional information loss for the four algorithms.

D. Analysis Of Distortion

Figure 4(a) shows the changes of distortion using MSB-KACA and generalization on different $|QI|$ when l is 6, k is 6 and n is 10000, which is calculated by formula(4). Figure 4(b) shows the changes of distortion on different n when l is 6, k is 6, and $|QI|$ is 5. We can conclude that MSB-KACA has low distortion since it does not need to take sensitive attributes diversity into account.



(a) $|QI|$ (l=6, k=6, n=10000)



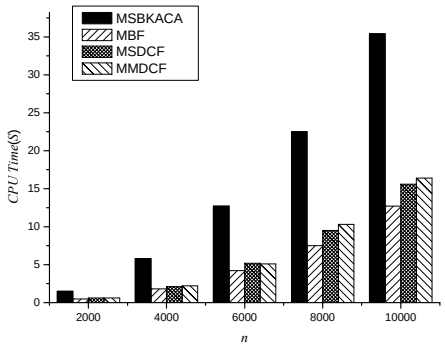
(b) n ($|QI|$=5, l=6, k=6)

Figure 4. Comparison of distortion for the two algorithms.

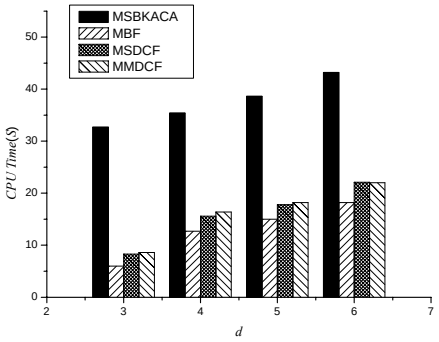
D. Analysis Of Execution Time

Figure 5(a) shows the cost of computing the publishable table by MSB-KACA,MBF,MSDCF and MMDCF on different n when d is 4 and l is 6. It is obvious that the cost increases as n increases. Since the more tuples need to be anonymized,the longer time consumed to finish the anonymization procedure. Figure 5(b) investigates the influence of d on execution time when l is 6 and n is 10000. Figure 5(c) investigates the influence of l on execution time when d is 4 and n is 10000. Apparently, MSB-KACA costs over twice time than other three algorithms. To explain this, recall that MSB-KACA combines anatomy and generalization techniques. Generalization costs more time than anatomy. That is why MSB-KACA takes more time.

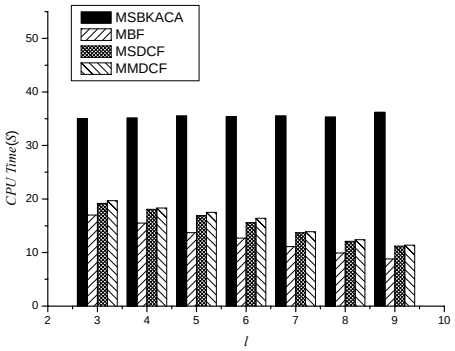
We can see that the advantage of our method in quality of anonymous data does not come for free. However, in all test cases our algorithm can finish in less than 1 minute, which is acceptable for data anonymization.



(a) n (d=4,l=6)



(b) d (l=6,n=10000)



(c) parameter l(d=4,n=10000)

Figure 5. Comparison of Execution time for the four algorithm.

VI. Conclusion And Ruture Work

The paper proposes an SLOMS method to anonymize multiple sensitive attributes microdata and analyses the privacy attacks that SLOMS can address. The paper also proposes an MSB-KACA algorithm based on SLOMS method. Experimental results show that the anonymized data by SLOMS have low additional information loss and suppression ratio compared with MSB.

Privacy preservation on multiple sensitive attributes microdata is a challenging work. There are many interesting topics in this area. For example, personalized privacy preservation [15] is an interesting work. In addition, research on efficient algorithms based on generalization and anatomy is also a significant work in the future.

Acknowledgment

This work has been supported by the National Natural Science Foundation of China under Grant No. 61170108 and No. 6110019 and the National Natural Science Foundation of Zhejiang Province under Grant No. Y1100161 and No. Q13F020007.

References

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

[2] Samarati P, Sweeney L. Generalizing data to provide anonymity when disclosing information. *In: Proc. of the 17th ACM-SIGMOD-SIGACT-SIGART Symposium on the Principles of Database Systems*. 1998, pp. 188.
[3] Samarati P, Sweeney L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. *Tech. rep., SRI International*. March, 1998.
[4] Machanavajjhala A, Gehrke J, Kifer D. *l*-Diversity: Privacy beyond K-anonymity. *In: Proc. of the 22nd International Conference on Data Engineering. Atlanta: IEEE Computer Society*, 2006, pp. 24-35.
[5] Li Ning-hui, Li Tian-cheng, Venkatasubramanian S. t-Closeness: privacy beyond *k*-anonymity and *l*-diversity. *In: Proc. of the 23rd ICDE*, 2007, pp. 106-115.
[6] Xiao X, Tao Y. Anatomy: Simple and effective privacy preservation. *In: Proc. of the 32nd International Conference on Very Large Data Bases. Seoul: VLDB Endowment*, 2006, pp. 139−150.
[7] Yufei Tao, Hekang Chen, Xiaokui Xiao, Shuigeng Zhou, Member, IEEE Computer Society, and Donghui Zhang. ANGEL: Enhancing the Utility of Generalization for Privacy Preserving Publication. *IEEE Transaction on Knowledge and Data Engineering.2009*, vol. 21.No.7. pp.1073-1087.
[8] Leela Rani. P, Revathi.N. Enhancing the Utility of Generalization for Privacy Preserving Re-publication of Dynamic Datasets. *International Journal of Computer Applications*, vol. 13, issue 6, 2011, pp. 42-49.
[9] Tiancheng Li, Ninghui Li, Jia Zhang, Ian Molloy. Slicing: A New Approach for Privacy Preserving Data Publishing. *Proc. IEEE Transactions on Knowledge and Data Engineering*, vol.24, No. 3, March 2012.
[10] Neha V. Mogre, Girish Agarwal, Pragati Patil. A Review On Data Anonymization Technique For Data Publishing[J]. *International Journal of Engineering Research & Technology*, vol. 1, issue 10, December 2012.
[11] Yang Xiao-chun, Wang Ya-zhe, Wang Bin. Privacy Preserving Approaches for Multiple Sensitive Attributes in Data Publishing [J]. *Chinese journal of computers*, 2008, 31(04), pp. 574-587.
[12] Li Jiuyong, Wong Raymond Chi-Wing, Fu Ada Wai-Chee, et al. Achieving *k*-anonymity by clustering in attribute hierarchicalstructure[C]. *DaWak. LNCS 4081, Springer-verlag, Berlin, Heidelberg*, 2006, pp. 405-416.
[13] H. Cramt'er, Mathematical Methods of Statistic. *Princeton Univ. Press*, 1948.
[14] L. Kaufman and P. Rousueeuw. Finding Groups in Data: An Introduction to Cluster Analysis. *John Wiley & Sons*, 1990.
[15] Han Jianmin, Yu Huiqun, Yu Juan and Cen Tingting. A Complete (*a*,*k*)-Anonymity Model for Sensitive Values Individuation Preservation. *International Symposium on Electronic Commerce and Security(ISECS2008)*, Guangzhou, China on 3-5 Aug., 2008, pp. 318-323

Jianmin Han was born in China. He earned his M.S. degree in 2000 and B.S. degree in 1992 both in Computer Science and Technology from DaQing Petroleum University at Anda. He received his PH.D. degree of Computer application technology from East China University of science and technology at Shanghai. He is currently a professor and M.S. supervisor in department of Computer Science and Technology at Zhengjiang Normal University. His main research interests include data mining, privacy preservation and information security. He has published over 30 papers.

Fangwei Luo was born in China. He earned his M.S. degree in 2013 and B.S. Degree in 2010 both in Zhejiang Normal University. He is currently a teacher in Yuyao High School, teaching NOIP. His main research interests include data mining, privacy preservation and information security. He has pulished 3 papers.

Jianfeng Lu was born in China. He received his B.S. degree in the School of Computer Science and Technology at Wuhan University of Science and Technology in 2005, and the PhD degree in the School of Computer Science and Technology at Huazhong University of Science and Technology in 2010.He is currently an associate professor and M.S. supervisor in department of Computer Science and Technology at Zhengjiang Normal University. His research interests include distributed system security and access control. He has published over 20 papers.

Hao Peng was born in China. He earned his M.S. degree in 2007 in Signal and Information Processing from Taiyuan University of Technology in Shanxi. He received his PH.D. degree of Communication and Information System from Shanghai Jiao Tong University at Shanghai. He is currently a lecturer in department of Computer Science and Technology at Zhengjiang Normal University. His main research interests include network and information security, distributed system security and complex system security. He has published over 10 papers.