Decomposition+: Improving ℓ-Diversity for Multiple Sensitive Attributes

Devayon Das and Dhruba K. Bhattacharyya

Department of Computer Science and Engineering,
Tezpur University-784028, India
devayon@acm.org, dkb@tezu.ernet.in

Abstract. In this paper, we analyse existing privacy-transformation techniques in the field of PPDP that anonymize datasets with Multiple Sensitive Attributes (MSA). Of these, we present an analysis of Decomposition, an algorithm which generates a dataset with distinct ℓ-diversity over MSA using a partitioning approach. We discuss some improvements which can be made over Decomposition: in the realms of its running time, its data utility, and its applicability in the case of Multiple Release Publishing. To this effect, we describe *Decomposition+* an algorithm that implements some of these improvements and is thus more suited for use in real-life scenarios.

Keywords: Privacy Preserving Data Publishing, ℓ-diversity, Decomposition, Multiple Sensitive Attributes, Multiple Release Publishing.

1 Introduction

The rapidly growing fields of *Privacy Preserving Data Mining* (PPDM), and its newer cousin *Privacy Preserving Data Publishing*(PPDP), essentially deal with issues that can be stated in very few terms: private data should be leveraged to infer useful patterns, but not to infer private, sensitive information. However, this simple statement becomes quite difficult to model as a problem. This is because, (i) given a dataset, it is difficult to differentiate data which is sensitive from data which has legitimate purpose of utility, and (ii) as sensitive data is obscured in the dataset, its general utility for non-nefarious purposes also diminishes. Indeed, every privacy preserving data publication method will lose some information; if not, it is equivalent to disclosing the data unprotected[1]. Given the rise of the rate at which personal datasets are being published, the problem gains significance.

PPDP distinguishes itself from PPDM in the context of the usage of anonymized data. While PPDM techniques are tailor-made for the use of an anonymized dataset to a specific data mining purpose, PPDP encompasses those techniques which a data-publisher may use to secure privacy of data against a generic data-mining purpose[2]. There are a large number of approaches and techniques involved in PPDM, such as Synthetic Data Generation, Perturbation, Micro-Aggregation, Suppression and Anatomization. For a more

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 403–412, 2012.

comprehensive survey, the reader is directed to [3,2]. Many privacy models, such as k-anonymity[4] and ℓ-diversity[5] isolate some attributes in the dataset as Sensitive Attributes (SA). These are important from a data utility and mining perspective, and also pose risk if they are linked to a particular individual represented in the dataset. Most implementations of these algorithms (and many more) focus on a Single Sensitive Attribute (SSA) for simplicity and convenience, instead of Multiple Sensitive Attributes (MSA), which are more useful as an anonymization policy and more suitable to real-life datasets. As such, algorithms implementing MSA are of significant interest.

Another important scenario which modern anonymization techniques should take into account, is the case of ever improving datasets and anonymization policies. Over time, datasets are corrected, and published under different anonymization techniques. When datasets are re-published, the releases could be combined to infer sensitive information, unless precautions are built into the anonymization techniques to prevent such attacks. Thus we require a privacy-preserving framework which ensures that (i)the disclosure of sensitive information in published datasets are limited to a small and measurable quantity, (ii)Multiple Sensitive Attributes are protected against disclosure, (iii)the disclosure risk does not escalate when data is published again in the future, and (iv)the utility of the published dataset is maximized (by a measurable quantity) while enforcing these constraints.

2 Background

Celebrated privacy models, such as k-anonymity[4], ℓ-diversity[5] and closeness[6] make a preliminary set of common assumptions for the sake of simplicity: (i)the data to be protected (or anonymized) is considered to be a set of tuples in a table $\mathrm{T} = \{t_1, \ldots, t_m\}$, where $t_i, (1 \leq i \leq m)$ is a tuple, (ii)each tuple t_i, having attributes $< c_1, \ldots, c_n >$, describes one individual person, (iii)attributes of the table can be divided into three distinct, disjoint sets of attributes: (a)*Explicit Attributes*, such as {Name, Social Security Number}, which individually can link a record to a person, explicitly. These are usually removed during the process of anonymization; (b)*Quasi Identifiers* (QIDs), such as {date of birth, gender, location}, which although individually do not identify a person, but considered as a composite, can be used to link the record with a person; (c)*Sensitive Attributes* (SA), such as {Salary} or {Medical Condition}, are needed for analysis but have potentially sensitive consequences if linked to an individual with strong certainty; (d)*Non-sensitive Attributes* which don't fall into any of the above categories and can be retained as-is in the anonymized data (called microdata). Importantly, the choice of partitioning the attributes between SA and QID is crucial in determining its privacy risk and well as data-utility. This choice however is a matter of policy[4].

Definition 1: *k-anonymity:*[4] A set of data is said to be k-anonymous iff each unique sequence of QIDs appears in T with at least k occurrences. Greater the value of k (k being a positive integer), greater the protection against of record being linked with certitude to a particular person and greater the ambiguity of the published data.

k-anonymity is usually accomplished through generalization or suppression[7]. In generalization, QIDs of multiple records are replaced with one generalized value, forming groups called *Equivalence Class.* In supression values which do not conform to k-anonymity are not released at all. Newer alternates to generalization based on Partitioning such as Anatomy[1] eliminate the information-loss involved in generalization by generating two projections of the dataset, one containing QIDs and the other the sensitive attributes.

Definition 2: *ℓ-diversity principle*[5]: An equivalence class is said to be ℓ-diverse when there are at least ℓ *well-represented* values for the sensitive attribute. The ℓ-diversity privacy model overcomes a shortcoming of the k-anonymity: while k-anonymity does not specify the selection criteria of SA values in the equivalence class. *Well-represented* could be construed as distinct ℓ-diversity:

Definition 3: *Distinct ℓ-diversity*[6]: An equivalence class is said to have distinct ℓ-diversity if there are at least ℓ distinct values for the sensitive attribute.

2.1 From SSA towards the MSA Case

Real world data-sets, such as the UCI Adult Dataset[8], usually would have a large number of attributes. Since most established algorithms anonymize datasets with only a single sensitive attribute, the data publisher is left with the choice of having to identify which one attribute should be chosen as the sensitive attribute. An alternative to these is to have a model which has multiple SAs. This scenario is known to be that of *Multiple Sensitive Attribute* (MSA). MSA has been widely mentioned in literature[5,9,10] but, as Ye et al.[11] report, there are few algorithms which implement anonymization in the MSA case. This is because when algorithms such as ℓ-diversity Incognito[5] are extended to the MSA case, a large loss of utility[12] occurs. If more work were to be done in the MSA case, this choice would neither be necessary nor needed. The few works found in our survey, dealing with MSA, are outlined here:

In [12], the authors showed the difficulty in achieving ℓ-diversity in the MSA case. At the same time, achieving MSA is trivial for k-anonymity, because k-anonymity does not restrict the distribution of SAs the equivalence class. Experimental results indicate introduction of significant distortion in the resultant data and small relative error for random SQL queries. In [13] the authors describe a privacy model, Multi-Sensitive Bucketization (MSB) and three MSB-based algorithms: maximal-bucket first (MBF), maximal single-dimension-capacity first (MSDCF), and maximal multi-dimension-capacity first (MMDCF). While they achieve good data utility, an analysis of privacy guarantees is absent. Ye et al.[11] apply an existing and well-understood privacy model, ℓ-diversity, in the MSA case and use an interesting vertical partitioning technique to form ℓ-diverse groups. Their algorithm, Decomposition is discussed and analysed in the next section.

3 Decomposition

The algorithm of Decomposition[11] which satisfies ℓ-diversity in the MSA case, is of interest. This is in part because it explores an alternate to generalization: vertical partitioning in achieving ℓ-diversity. Partitioning, which has been implemented in various guises[1,14] can provide better data utility than generalization, in many cases. For a balanced analysis of partitioning, refer to [15].

Partitioning usually implies that a table T with attributes $A_1, \ldots, A_m$ is vertically partitioned into two or more sub-tables $\overline{\mathrm{T}}_1, \ldots, \overline{\mathrm{T}}_n$ such that any table $\overline{\mathrm{T}}_i, 1 > i \geq n$ has attributes $A_j, \ldots, A_k$ where $1 \geq j \geq k \geq n$. The join of two or more sub-tables forms a *lossy view* of the underlying data. Decomposition works by partitioning T vertically into sensitive attributes and non-sensitive attributes. The SA-table (see Table 3) are further partitioned horizontally into *SA-groups* of records such that each group contains at least ℓ distinct sensitive attribute instances for each sensitive attribute. Every tuple in the QID-table is associated with one SA-group (see Table 4). Apart from this, the sensitive attributes are released in a separate table which cannot be linked with the other released table. To reduce information loss, the number of SA-groups created are maximized by Ye et al. through a largest-ℓ group forming procedure, which they prove creates the maximum number of groups possible.[1]:

The largest-ℓ group forming procedure applies to creating SA-groups with respect to only one SA. To extend it to the MSA case, the authors have designated one of the many sensitive attributes as Primary Sensitive Attribute S^{pri} with corresponding diversity requirement of ℓ^{pri}, which is chosen by the publisher as a matter of policy. Once SA-groups are formed by applying the largest-ℓ group forming procedure with respect to $S^{p\,r\,i}$, the SA-table may still not satisfy $\ell_1, \ldots, \ell_d$-diversity with respect to all the non-primary sensitive attributes.To rectify this, Ye et al. introduce a noise addition step. To add noise, in every tuple in the SA-table, for every sensitive attribute S^i which does not satisfy ℓ_i-diversity, the value of S^i is replaced from a set defined as the Linkable Sensitive Value[11].

Adding noise causes distortion. However, the use of Diversity Penalty in the partitioning stage ensures that each SA-group conforms to $\ell_1, \ldots, \ell_d$-diversity as much as possible. Thus minimal amount of noise is added in this stage. Thus, Decomposition generates three tables for publishing from the original data. Table 1 shows a dataset, which has not been anonymized. Table 4 (QID-table), 3 (SA- table G) and 2 (Marginals T_S) show the published microdata when Decomposition is applied.

3.1 Discussion

Decomposition ensures distinct ℓ-diversity in the MSA case, which is a well understood privacy model and can thwart attribute-linking and record-linking attacks. It also gives better data utility than generalization. However, Decomposition has certain weaknesses: (i)using partitioning only to form ℓ-diverse data

[1] Theorem 2 in [11].

Table 1. The Microdata table

Tuple #	Gender	ZipCode	Birthday	Occupation	Salary
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. Marginals

Occupation	Salary
Nurse	1
Nurse	4
Police	8
Cook	9
Actor	2
Actor	7
Clerk	8
Clerk	2

Table 3. Sensitive attributes of Table 1 after Decomposition

Group	Occupation	Salary
1	Police	1
1	Nurse	2
1	Actor	8
1	Clerk	4
2	Nurse	2
2	Actor	4
2	Cook	7
2	Clerk	9

Table 4. QIDs and non-sensitive attributes of Table 1 after Decomposition

Group	Gender	ZipCode	Birthday
1	F	10078	1988/04/17
	F	10085	1962/10/03
	M	20086	1958/06/06
	M	10076	1985/03/01
2	F	10077	1984/03/21
	M	10085	1988/11/04
	F	10075	1983/02/14
	F	20087	1960/07/11

over the primary sensitive attribute, not other SAs, (ii)the choice of noise values could further be improved to reduce information loss, (iii)is not suitable in cases where records could be added later, which is a practical, real-life requirement.

4 Decomposition+

Based on our analysis of Decomposition in the last section, we attempt to improve upon it in the following two broad areas: (i)extending Decomposition to the continuous release scenario, (ii)Optimizing noise value selection.

(i)**Extending Decomposition to the continuous release scenario:** From a practical and long-term view, PPDP would involve the same or related data being anonymized and published multiple times. For example, a hospital may release information on a monthly basis, and may have patients who exist in multiple releases. This extended scenario could occur in the one of these three situations: (i) Multiple Release, (ii)Sequential Release and (iii)Continuous Release, also known as the Incremental Dataset Release.

In Continuous Release scenario, different anonymized releases of the same underlying data are released at different points in time, where records have been added, removed, or updated in the underlying data. The attempt is to include these changes in the published data, while reducing risk of the use of these changes in infering sensitive information. In order to enable continuous release in our proposed algorithm, if the anonymized dataset is published as a release of three tables $\hat{\mathrm{T}}_0 = \{\hat{\mathrm{T}}_0^M, \hat{\mathrm{T}}_0^Q, \hat{\mathrm{T}}_0^S\}$ where $\hat{\mathrm{T}}_0^M$ is the marginal, $\hat{\mathrm{T}}_0^Q$ is the QID-table, and $\hat{\mathrm{T}}_0^S$ is the SA-table, our concern would be that p future releases of $\hat{\mathrm{T}}_i$ $(0 \leq i \leq p)$ should not be linked to each other to leak sensitive information. Byun et al.[16] define an Inference Channel which is useful in formalizing this risk. We extend this to the $\ell_1, \ldots, \ell_d$-diversity[2] case:

Definition 4: *Inference Channel for $\ell_1, \ldots, \ell_d$-diversity*: Let $\hat{\mathrm{T}}_i$ and $\hat{\mathrm{T}}_j$ be two $\ell_1, \ldots, \ell_d$-diverse releases of T. An inference channel exists between $\hat{\mathrm{T}}_i$ and $\hat{\mathrm{T}}_j$, denoted by $\hat{\mathrm{T}}_i \rightleftharpoons \hat{\mathrm{T}}_i$ if observing $\hat{\mathrm{T}}_i$ and $\hat{\mathrm{T}}_j$ together increases the probability of attribute disclosure of an attribute S^k in either $\hat{\mathrm{T}}_i$ or $\hat{\mathrm{T}}_j$ to a probability greater than $1/\ell_k, (1 \leq k \leq d)$

Thus every new release $\hat{\mathrm{T}}_{n+1}$ must be inference-free from all the previous releases, as defined as:

Definition 5: *Inference-free data release for $\ell_1, \ldots, \ell_d$-diversity:* Let $\hat{\mathrm{T}}_0, \ldots, \hat{\mathrm{T}}_n$ be a sequence of previously releases of T, each of which is $\ell_1, \ldots, \ell_d$-diverse. A new $\ell_1, \ldots, \ell_d$-diverse release $\hat{\mathrm{T}}_{n+1}$ is said to be inference-free iff $\nexists \hat{\mathrm{T}}_i, i = 1, \ldots, n$ s.t. $\hat{\mathrm{T}}_i \rightleftharpoons \hat{\mathrm{T}}_{n+1}$.

Given the above, Byun, et al. proved that addition of a new equivalence class (or a new SA-group) to a release does not cause an inference channel to a previous release[3] as long as each SA-group is $\ell_1, \ldots, \ell_d$-diverse. If a tuple is inserted into an SA-group, the SA group must already be $\ell_1, \ldots, \ell_d$-diverse, and the tuple must remain in the same SA-group across releases.

Thus, we employ the largest-ℓ group forming procedure to the available records and unlike Decomposition we retain residual tuples for future anonymization, and do not add them to existing SA-groups. The rationale for this is to enable creation of new SA-groups when more tuples are added to the dataset. To avoid a situation where some tuples are never published at all, we assign, to each tuple t, a starvation penalty, defined as $P_s(t) = b - a$, where t is introduced into the underlying data table T after $\hat{\mathrm{T}}_0, \ldots, \hat{\mathrm{T}}_a$ releases have been made, and t first appears in a published release after another $\hat{\mathrm{T}}_{a+1} \ldots \hat{\mathrm{T}}_b$ releases.

When the number of distinct residual tuples becomes greater than ℓ^{pri}, we attempt to form an SA-group from ℓ^{pri} distinct tuples with tuples with the highest starvation penalty.

(ii)Improving the noise selection procedure: When the largest ℓ-group forming procedure is applied to the dataset, non-primary SAs may not conform to distinct $\ell_1, \ldots, \ell_d$-diversity. Noise is added to remove an offending value, which is a non-primary sensitive attribute value occurring more than once in the SA-group. Offending values can be identified during the d-SA-ℓ-diversity checking

[2] $\ell_1, \ldots, \ell_d$-diversity is defined in in [11].
[3] Section 4.3 of [16].

process described in our algorithm. Decomposition accomplishes this by adding a value from the set defined by $LSV(S^i, G) - G.S^i$ where S^i is the non primary SA, T^s is the Sensitive Table, $\bowtie$ is natural join, and S^{pri} is the primary SA. If this set contanins more than one element, Decomposition randomly chooses a value and merges it with the SA-group, assumes that all values in the set are equally distant from the original offending value and therefore any value chosen from the set is equally valid. However this may not be the case. For example, comparing Table 3 and 6, we see that the value '4' has been added as noise because tuple 5 and 8 appear have the same value 2 for salary. Now, $LSV(Salary, G_1) = \{1, 2, 4, 7, 8\}$ and $LSV(Salary, G_1) - G_1.Salary = \{4, 7\}$. Now, the offending value is 2. Clearly 4 and 7 are not equally distant from 2. Therefore it is necessary to devise a method to choose a noise value which is semantically closest to the offending value. To quantify semantic distance between sensitive attributes, we use the Hierarchical Distance[6], considering the fact that in ℓ-diversity essentially treats all attributes in the SA-group as categorical data[5]. Hierarchical Distance is defined as follows: if H be the height of the domain hierarchy tree, the distance between two attribute values v_1 and v_2 is defined to be $level(v_1, v_2)/\mathrm{H}$, where $level(v_1, v_2)$ is the height of the lowest common ancestor node of v_1 and v_2. Our algorithm, Decomposition+, accepts as input a hierarchy tree for every non-primary sensitive attribute. In light of the above discussions, our algorithm Decomposition is as follows:

Table 5. Residual tuples in each group for different values of ℓ_{pri}

Attribute No.	Distinct Values	ℓ_{per}
Age (1)	73	n.a
Final-Weight (2)	100	n.a
Marital Status (3)	7	n.a
Race (4)	5	n.a
Gender (5)	2	n.a
Work-class (6)	14	7
Education (7)	16	3
Hours per week (8)	99	2
Relationship (9)	6	3

Table 6. SAs from Table 1 without addition of noise

Group	Occupation	Salary
1	Police	1
1	Nurse	2
1	Actor	8
1	Clerk	
2	Nurse	2
2	Actor	4
2	Cook	7
2	Clerk	9

4.1 Algorithm for Decomposition+

Input: (i)Table T with sensitive attributes $S_1, S_2, S_3 \ldots Sd$, one of them being the primary: S_{pri}, (ii) Diversity parameters $\ell_1, \ell_2, \ell_3 \ldots \ell_d$, (iii) The hierarchical category tree H_i of each S_i where $i \neq pri, 1 \leq i \leq d$, (iv)Penalty Threshold $P_s^{threshold}$

Data: (i)$\mathfrak{B}$, the set of buckets formed by primary sensitive attributes. $\mathfrak{B} = (\mathrm{B}_i)$, (ii)$\mathcal{G} = \Phi$, $\mathcal{G}$ is the set of SA-groups.

Output: the decomposed table T^* which satisfies $(\ell_1, \ell_2, \ldots \ell_d)$-diversity

Algorithm:

1. Sort $\mathfrak{B}$ by decreasing size
2. *while* $|\mathfrak{B}| \geq \ell_{pri}$
2.1 Randomly remove one tuple from B_0. 2.2 set $G = \{t_1\}$;
2.3 *for* $i \leftarrow 2$ to ℓ_{pri}
2.3.1 Remove one tuple t_i from B_i, that minimizes $P(t_i, G)$;
2.3.2 $G = G \bigcup t$;
2.3.3 Mark any attribute values which repeat
2.4 $\mathcal{G} = \mathcal{G} \bigcup G$;
3 *foreach* residual tuple t
3.1 *if* $P_s(t) > P_s^{threshold}$ *then*
3.1.1 Find SA group G that minimizes $P(t, G)$;
3.1.2 $G = G \bigcup t$; mark any attribute values which repeat
4 *foreach* non-primary sensitive attribute S^i and each SA-group G
4.1 *if* $G.S^i$ does not satisfy ℓ_i-diversity *then*
4.1.1 $LSV(G, S^i) = \prod_{S^i} T^S \bowtie G.S^{pri} - G.S^i$;
4.1.2 $R_V \leftarrow$ repeated value in S^i;
4.1.3 Select value N from $LSV(G, S^i)$ such that hierarchical distance $H(v_i, R_V)$ is minimized (where v_i is a member of the set $LSV(G, S^i)$)
4.1.4 Merge N into $G.S^i$ *until* $G.S^i$ satisfies ℓ_i-diversity.

4.2 Discussion

Based on the theoretical improvements proposed, and the algorithm presented, we may conclude that (i)our algorithm builds upon Decomposition by allowing tuples to be added to the underlying dataset after it has been anonymized and published. This facilitates greater flexibility in real life scenarios where tuples may be added removed or updated and may appear in multiple releases of the same data, (ii)the addition of new tuples does not dilute the protection offered in previous releases of the data, (iii)the proposed algorithm Decomposition+ also chooses a better noise value compared to Decomposition, which chooses randomly over the allowed values, (iv)Decomposition+ chooses noise value as close to the original value. This provides better utility, especially when the space of allowed noise values are large and when Decomposition chooses a particularly distant noise value. This is done while maintaining ℓ-diversity.

5 Experiments

To experimentally evaluate our proposed algorithm, we implemented Decomposition+ and applied it on the UCI Adult Dataset[8]. Experiments were conducted on a workstation running Ubuntu 11.04 (32-bit) with 3 GB RAM. Decomposition+ and associated preprocessing tools were implemented in Python v2.7. Data was supplied to the programs in Comma Separated Values format. Some analysis was done using Microsoft Excel 2007. The dataset was preprocessed in the same manner as described in [11] for a level playing field: (i)There were

32561 tuples in the dataset and after removing tuples with missing attribute instances, 30162 records were left, (ii) out of 14 attributes of the Adult dataset Nine (9) attributes were retained: *Age, Final-Weight, Martial Status, Race, Gender, Work-class, Education, Hours per Week* and *Relationship*, (iii) *Work-class* was used as Primary Sensitive Attribute, (iv)of these, the first four attributes were deemed as QIDs and the remaining were deemed as MSA, with corresponding ℓ-diversity parameters of 7, 3, 2, 3 respectively.

Occurrence of residual tuples: In the first instance, in order to study the effect of the choice of ℓ_{pri} on the number of tuples, we published only those tuples which are grouped during the largest ℓ group forming procedure for different input values for ℓ_{pri} between 0 and the maximum permissible value, 7. The importance of this analysis is that the higher the value of ℓ_{pri} chosen, the greater will be the protection offered. However, the greater the number of tuples which remain unpublished, the more the published data will differ from the original (See Table 7(a)). In the current scenario $\ell_{pri} = 5$ would be a good tradeoff between privacy and future utility. If the data were to have a more even distribution of primary sensitive attribute instances, a higher value of ℓ_{pri} would be preferable.

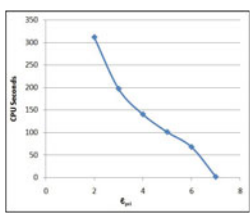
Performance: In order to measure how the choice of ℓ_{pri} affects performance, we used the Python module CProfile to measure running times for different values of ℓ_{pri}. Results are given in Figure 7(b). The results we recorded are significantly faster than those reported by Ye et al for Decomposition. However, this could be because of multiple causes such as CPU speed and implementation dependency. What is clear is that a smaller value of ℓ_{pri} causes larger number of buckets to be formed which require exponentially greater CPU seconds to distribute among SA-groups. We also noticed that the calculation of *Diversity Penalty* requires a inordinately large amount of CPU cycles (about 43.7% of total time).

Table 7. Results of our experiments

(a) Residual tuples in each group for different values of ℓ_{pri}

7	6	5	4	3	2	1
929	-	-	-	-	-	-
1060	117	-	-	-	-	-
1265	322	0	-	-	-	-
2053	1110	412	0	-	-	-
2485	1542	844	1	0	-	-
22272	21329	20631	19661	18348	14410	0

(b) ℓ_{pri} versus performance (in CPU seconds)



6 Conclusion and Further Work

Decomposition+ is an interesting and practical improvement, albeit one of many possible improvements, of Decomposition. Other improvements could be targeted to improve the efficiency of largest ℓ group forming procedure. Ye et al. do not specify the nature of how the set of all buckets in Decomposition, are formed. In our opinion, because buckets are reduced in size by one, a specific optimized data-structure to represent the collection of buckets can be useful. Further work could be extended to two interesting directions. One would be to apply decomposition over MSA to achieve (n, t)-closeness or other privacy models. The second, and more important work would be to apply Decomposition to very large datasets, which are known to suffer from the Dimensionality Curse[3].

References

1. Xiao, X., Tao, Y.: Anatomy: Simple and effective privacy preservation. In: Proceedings of the 32nd Intl. Conference on Very Large Data Bases, VLDB Endowment, pp. 139–150 (2006)
2. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A Survey of Recent Developments. ACM Computing Surveys 42, 1–53 (2010)
3. Aggarwal, C.C., Yu, P.S. (eds.): Privacy-Preserving Data Mining. Advances in Database Systems, vol. 34. Springer, US (2008)
4. Sweeney, L.: k-anonymity: A Model for Protecting Privacy. Intl. Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10, 557–570 (2002)
5. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) 1, 3 (2007)
6. Li, N., Li, T., Venkatasubramanian, S.: t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, vol. (2). IEEE (2007)
7. Sweeney, L.: Achieving k-Anonymity Privacy Protection Using Generalization And Suppression. Intl. Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10, 571–588 (2002)
8. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010)
9. Zhao, V., Wang, J., Luo, Y., Lei, J.: (α, β, k)-anonymity: An effective privacy preserving model for databases. In: 2009 Intl. Conference on Test and Measurement, pp. 412–415. IEEE (2009)
10. Venkatasubramanian, S.: Closeness: A New Privacy Measure for Data Publishing. IEEE Trans. on Knowledge and Data Engineering 22, 943–956 (2010)
11. Ye, Y., Liu, Y., Wang, C., Lv, D., Feng, J.: Decomposition: Privacy Preservation for Multiple Sensitive Attributes. In: Zhou, X., Yokota, H., Deng, K., Liu, Q. (eds.) DASFAA 2009. LNCS, vol. 5463, pp. 486–490. Springer, Heidelberg (2009)
12. Gal, T.S., Chen, Z., Gangopadhyay, A.: A Privacy Protection Model for Patient Data with Multiple Sensitive Attributes. Intl. Journal of Information Security and Privacy 2, 28–44 (2008)
13. Yang, X.C., Wang, Y.Z., Wang, B., Yu, G.: Privacy preserving approaches for multiple sensitive attributes in data publishing. Jisuanji Xuebao/Chinese Journal of Computers 31, 574–587 (2008)
14. Zhang, Q., Koudas, N., Srivastava, D., Yu, T.: Aggregate Query Answering on Anonymized Tables. In: 2007 IEEE 23rd Intl. Conference on Data Engineering, pp. 116–125. IEEE (2007)
15. Xiao, X., Tao, Y.: Anatomy: Privacy and Correlation Preserving Publication. Technical Report i, Chinese University of Hong Kong, Hong Kong (2006)
16. Byun, J., Sohn, Y., Bertino, E., Li, N.: Secure Anonymization for Incremental Datasets. In: Jonker, W., Petković, M. (eds.) SDM 2006. LNCS, vol. 4165, pp. 48–63. Springer, Heidelberg (2006)