Suppression Slicing – using l-diversity

S. Kiruthika
Dept.CSE
Akshaya College of engineering and technology
Coimbatore, Tamilnadu

M. Mohamed Raseen M.S
Associate Professor/Dept.CSE
Akshaya college of engineering and technology
Coimbatore, Tamilnadu

ABSTRACT
An important problem in publishing the data is privately held data about individuals without revealing the sensitive information about them. Several anonymization techniques, such as suppression, bucketization and slicing have been designed for privacy preservation in microdata publishing. Suppression involves not releasing a value at all it leads to the utility loss while the anonymized table may use by the data miners. Bucketization does not prevent membership disclosure and does not apply for data that do not have a clear separation between quasi-identifying attributes and sensitive attributes. On the other hand slicing, this partitions the data both horizontally and vertically. Slicing preserves better data utility than generalization and can be used for membership disclosure protection. But in the slicing each attribute consider only single column. This releases more attribute correlations and it leads to a secrecy loss in privacy. An effective slicing is introduced in this paper to show how slicing can be performed with suppression in the attributes which have similar values in the different tuples and an efficient algorithm for computing the sliced data that obey the l-diversity requirement.

General Terms
Privacy preservation, bucketization, generalization, l-diversity, data anonymization, data publishing.

Keywords
Privacy preservation, Suppression, Slicing, Suppression Slicing.

1. INTRODUCTION
Microdata places a major role in most of organizations (medical, travel, mobile and insurance agencies) are experiencing an exponential growth in data collection. Microdata contain records each of which contains information about an individual entity, such as a person, a household, or an organization. Failure to provide privacy protection within a release of the data may create situations that harm the public. To avoid such privacy threat, the uniquely identifying information is removed from the table before publishing. Privacy preservation provides a limitation in linking the unveiled data to a particular individual. To preserve the privacy the attributes are partitioned into three categories. They are (i) identifiers (IDs) that uniquely identify an individual E.g.: name (ii) Quasi-identifiers (QIs) that which already know, when taken together can potentially identify an individual E.g.: gender, zip code and birth date and (iii) sensitive attributes (SAs) are the fields that should be considered as sensitive to avoid being associated to specific persons. E.g.: disease, salary and criminal offence. Anonymization techniques have been proposed to protect the person's private information while publishing. Most popular techniques are generalization for k-anonymity and bucketization for l-diversity. Basic Anonymization steps are to remove the identifiers from the data and then partitions tuples into buckets are common in all the techniques then further it differs for each technique.

The rest of this paper is organized as follows: section 2, groundwork definition then we discuss the existing anonymization techniques such as generalization, bucketization and slicing in section 3, 4 and 5 respectively. In section 6 the proposed techniques and their definitions are discussed and in the 7 working procedure is discussed with the sample table. Section 8 the algorithm is defined with their description and in the 9 sections the comparison the suppression slicing with the slicing and conclude the paper in section 10.

2. GROUNDWORK DEFINITION
Definition 1: Attributes
Let T (A_1, ... , A_n) be a table with a finite number of tuples. The finite set of attributes of T are {A_1, ... , A_n}
Attribute in a table can be classified as four categories according to its role:-
Explicit-identifier is the attribute which can identify individual explicitly. E.g.: Name, Social security number etc. These attributes can be deleted before publishing to protect privacy.
Quasi-identifier (QI) refers to a group of attributes that can recognize personal information by using link such as {race, birth, sex, ZIP}. These require external information for linking.
Sensitive attribute are fields that has to be hidden which includes individual sensitive information. E.g.: salary, religion, political party, physical condition, etc
Non-sensitive attributes are one which cannot be ignored while publishing. E.g.: state, country, model
Definition 2: Quasi-identifier
Given a population of entities U, an entity-specific table T (A_1 ,..., An), f_c: U → T and f_g: T → U', where U ⊆ U'. A quasi-identifier of T, written Q_T, is a set of attributes {A_i,...,A_j} ⊆ { A_1,... ,A_n }such that $f_g(f_c(p_i)[Q_T]) = P_i$.
Definition 3: k-anonymity
Let RT(A_1,... ,A_n) be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy k-anonymity if and only if each sequence of values in RT[QI_{RT}] appears with at least k occurrences in RT[QI_{RT}].
Definition 4: Single-dimensional generalization
Let D_i be the domain of an attribute Ai. A generalization, such as full-domain generalization and sub tree generalization, is defined by a function f_i : DA_i → D' for each attribute A_i in QI_{RT}.
Definition 5: Multidimensional generalization
A multidimensional generalization is defined by a single function f:DA_1x...xDA_n→D', which is used to generalize Qid = <v_1, …, v_n> to Qid' = <u_1, …, u_n>. For every v_i, either v_i = u_i or v_i is a child node of u_i in the taxonomy of A_i. This scheme flexibly allows two qid groups, each group having the same value v, to be independently generalized into different parent groups.

3. SUPPRESSION
The process of suppression is to replace the * value instead of the column value [11]. The suppression can be performed not in the fully replacement of the attribute value instead it

replace the value partially. Thus if the integer field have six digit to suppress means that the first three digits comes as the same and the * be replaced for the remaining three digits. The field want to suppress is of one digit then simply * be replaced and for character attribute also for the same procedure will be followed. Most probably the suppression can be done in the quasi identifiers. The drawback of the suppression is it reduces the utility of the dataset thus an exact value of the attribute is not present and it leads to the information or data loss for the researchers.

4. BUCKETIZATION

In bucketization SAs are separated from the QIs by doing the random permutation on the SA values in each bucket. Thus an anonymized data consist of a set of buckets with a permuted sensitive attribute values. Bucketization does not prevent membership disclosure because it publishes the QI values in their original forms thus it is easy to find out an individuals record in the published data. Bucketization requires the clear separation of SA and QI attributes and it breaks the attribute correlation between them. l-diversity is insufficient to provide attribute disclosure [10]. Two attacks on l-diversity are

Skewness Attack:

When the overall distribution is skewed, satisfying *l*-diversity does not prevent attribute disclosure [9].

Similarity Attack:

When the sensitive attribute values in an equivalence class are distinct but semantically similar, an adversary can learn important information.

This leakage of sensitive information occurs because while l-diversity requirement ensures "diversity" of sensitive values in each group, it does not take into account the semantically closeness of these values.

5. SLICING

Slicing partitions the data set both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns. The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better utility [1] than generalization and bucketization. Slicing preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Note that when the data set contains QIs and one SA, bucketization has to break their correlation; slicing, on the other hand, can group some QI attributes with the SA, preserving attribute correlations [8] with the sensitive attribute. The problem of slicing is illustrated with the sample micro table set. After the random permutation performed in the dataset one-attribute-per column slicing is done by grouping some of the attributes in the dataset. Then the fake tuples are generated from the combination of different tuples in the one-attribute-per column slicing dataset. Thus fake tuples generated by the slicing acquire more space in the dataset and the utility of the data set may also be reduced. The fake tuples are generated in the ratio 4:16. For one original tuple present in dataset that contain four tuples for each bucket with the two columns then the 4 fake tuples are generated with remaining tuples within the bucket along to the same attribute value on one column.

Table 1 The Original Microdata Table

Age	Sex	Zip code	Disease
22	M	47906	Dyspepsia
22	F	47906	Flu
33	F	47905	Flu
52	F	47905	Bronchitis
54	M	47302	Flu
60	M	47302	Dyspepsia
60	M	47304	Dyspepsia
64	F	47304	Gastritis

Table 2 One-attribute-per Column Slicing

Age	Sex	Zipcode	Disease
22	M	47906	Dyspepsia
22	F	47906	Flu
33	F	47905	Flu
52	F	47905	Bronchitis
54	M	47302	Flu
60	M	47302	Dyspepsia
60	M	47304	Dyspepsia
64	F	47304	Gastritis

Table 3 Slicing Table

{Age ,Sex }	{Zipcode, Disease}
{22,M}	{47906, Flu }
{22,F}	{47906,dyspepsia}
{33,F}	{47905, Bronchitis }
{52,F}	{47905, Flu }
{54,M}	{47304, Gastritis }
{60,M}	{47302, Flu }
{60,M}	{47304,dyspepsia}
{64,F}	{47304,dyspepsia}

The table 1 shows the original microdata table in that all identifiers are removed. From the table 2 and 3 in the bucket two 3 tuple is repeated as such from the original table. The fake tuples are generated from table 3 for the four tuples in the one bucket 16 tuples are generated in that 12 are fake tuples remaining four are original tuples. An example for the fake tuple generation for the first two tuple is shown in the table 4. When the bucket size and the number of columns are increased then the number of fake tuples becomes larger. If the adversary who knows all the QI values of tuple try to access the sensitive value of t. With the minimum probability the adversary can point out a sensitive attribute of the identity. Thus the published microdata table after the slicing may have great loss in utility of the original microdata table.

Table 3 Slicing Table

{Age ,Sex }	{ Zipcode, Disease}
{22,M}	{47906,Flu}
{22,M}	{47905, Flu }
{22,M}	{47906, dyspepsia }
{22,M}	{47905, Bronchitis }
{22,F}	{47906,Flu}
{22,F}	{47905, Flu }
{22,F}	{47906, dyspepsia }
{22,F}	{47905, Bronchitis }

6. SUPPRESSION SLICING USING l-DIVERSITY

Let T be the microdata table to be published thus the identifiers are removed. T contains d attributes $A=\{A_1,A_2,\ldots\ldots\ldots A_d\}$ and their attribute domains are $\{D[A_1],D[A_2],\ldots\ldots\ldots ,D[A_d]\}$. A tuple t ε T can be represented as $t= (t[A_1],t[A_2],\ldots\ldots\ldots ,t[A_d])$ where $t[A_i]$ is the A_i value of t.

Definitions

Attribute partition and column: An attribute partition consists of several subsets of A, such that each subset of attributes is called a column. Specifically, let there be c columns $c_1,c_2,\ldots\ldots c_c$ then $\bigcup^{c}_{i=1} c_i = A$ and for $1 \le i_1 \ne i_2 \le c$, $c_{i1} \cap c_{i2} = \Phi$

We consider only one sensitive attribute S. If the data contain multiple sensitive attributes, one can either consider them separately or consider their joint distribution [6]. Exactly one of the c columns contains S. Without loss of generality, let the column that contains S be the last column Cc. This column is also called the sensitive column. All other columns contain only QI attributes.

Tuple partition and buckets: A tuple partition consists of several subsets of T, such that each tuple belongs to exactly one subset. Each subset of tuple belongs to exactly one subset. Each subset of tuples is called a bucket. Specifically, let there be b buckets $b_1,b_2,\ldots\ldots b_c$ then $\bigcup^{b}_{i=1} b_i = T$ and for $1 \le i_1 \ne i_2 \le b$, $b_{i1} \cap b_{i2} = \Phi$

Column generalization: a column generalization for a column ci is defined as a set of non overlapping j-dimensional regions that completely cover $D[A_{i1}] \chi D[A_{i2}] \chi\ldots\ldots\chi D[A_{ij}]$. A column generalization maps each value of c_i to the region in which the value is contained.

Column generalization ensures that one column satisfies the k-anonymity requirement. It is a multidimensional encoding [3] and can be used as an additional step in slicing as well as in suppression slicing.

Matching buckets: Let c be the c columns of a sliced table. Let t be a table and $t [c_i]$ be the value of t. Let B be a bucket in the sliced table and $B[C_i]$ be the multiset of c_i values in B. we say that B is a matching bucket of t iff for all $1 \le i \le c$, $t[c_i] \varepsilon B[c_i]$.

Suppression: Suppression can be performed in the similar attribute values present in the different tuples.

7. WORKING PROCEDURE

The table 1 shows a microdata table and its suppression sliced tables using various anonymization techniques are shown in table 4 and 5. In the table-1 identifiers are removed and the QI and SA are present.

Identifiers – nil

Quasi identifiers (QI) – age, sex, zip code

Sensitive attribute (SA) – disease

Thus the attributes are partitioned into columns. Each column contains a subset of attributes. Each column in the suppression sliced table contains exactly one attribute. Then horizontal partition thus partition tuples into buckets. Each bucket contains a subset of tuples. The similarity check will be done to identify the similar QI values in the different tuples and the suppression will be performed in the tuples. The values in each column are randomly permuted with in the bucket and the suppressed quasi identifier field value in that tuple remain constant to break the linking between the different columns. Thus the table 4 shows the suppression slicing performed on one attribute per column. The table 5 gives the suppression sliced table based on the column generalization of two attributes per column.

Table 4 One-Attribute-Per-Column Suppression Slicing

Age	Sex	Zip code	Disease
22	F	47906	Flu
22	M	47905	Flu
33	F	47906	Dyspepsia
52	F	47905	Bronchitis
54	M	47302	Dyspepsia
60	F	473*	Gastritis
60	M	473*	Dyspepsia
64	M	47304	flu

The buckets values are randomly permuted with in the bucket along with the suppression thus the privacy is achieved very effectively even if some of the tuples have similar QI value.

Table 5 Suppression Sliced table

{Age ,Sex }	{ Zipcode, Disease}
{22,F}	{47906,Flu}
{22,M}	{47905, Flu }
{33,F}	{47906, Dyspepsia }
{52,F}	{47905, Bronchitis }
{54,M}	{47302, Dyspepsia }
{60,F}	{473*, Gastritis }
{60,M}	{473*, Dyspepsia }
{64,F}	{47304, Flu}

In the original table if the attackers have the background knowledge then they can easily access the individual personal record so the data publisher will go for anonymization techniques such as suppression, generalization and slicing.

8. PROPOSED ALGORITHM

8.1. Tuple partition algorithm

```
Tuple partition (T, l)
Q = {T}; MSB= Φ;
While Q is not empty
        Remove first bucket B from Q;
        Q=Q-{B}.
        Similarity check (T, d)
        Split B into two buckets B1, B2 as in Mondrian
        if diversity-check (T,Q Ụ {B1,B2} Ụ SB, l)
                Q=Q Ụ {B11, B12}
        else
                SB=SB Ụ {B11} Ụ {B21}
return SB
```

In the tuple partitioning algorithm takes two phases. In the first phase tuples are partitioned into buckets. The tuple partition algorithm is defined by modifying the Mondrian algorithm [4] and with in the bucket the similarity between the tuples are identified and suppression be implemented. In the phase two slicing [1] algorithm is modified to achieve the random permutation with in the bucket. The tuple partition algorithm uses two data structures one for queue of buckets Q in the table and another for slicing bucket SB. Initially Q contains the table and SB is null then for each iteration the first bucket (B) is removed from Q and then splits each buckets into two sub buckets. Here the splitting criteria follow [4]. Then the split satisfies the l-diversity then the two buckets are combined back to Q after the permutation done in the buckets. The tuple partitioning algorithm is as similar to the tuple partitioning algorithm for slicing [8] with some modification for similar tuple checking.

8.2. Diversity check algorithm

```
diversity-check (T, T*,l)
for each tuple t ε T, L[t] = Φ;
   for each bucket b in T*
    Record f (v) for each column value v in bucket B
        for each tuple t ε T
            Calculate p (t, B) and find D (t, B)
            L[t]=L[t] Ụ {p(t,B) and find D(t,B)}
 for each tuple t ε T
            Calculate p (t, s) for each s based on L[t]
            If p (t, s) ≥ 1/l return false
return true
```

Let p (t, s) be the probability that t takes a sensitive value s. p (t, s) is calculated using the law of total probability. Let p (s|t, B) be the probability that takes sensitive value s given that t is in bucket B, then according to the law of probability, p(t, s)is

$$P(t,s)= p(t,B)p(s|t,B)$$

The probability that t takes a sensitive value s must be less than or equal to 1/ l. thus every tuple t in the table satisfies l-diversity for any sensitive value s then the sliced table satisfies l-diversity. Let p (t, s) be the probability that t takes a sensitive value s. p (t, s) is calculated using the law of total probability. Let p (s|t, B) be the probability that takes sensitive value s given that t is in bucket B, then according to the law of probability, p (t, s) is

$$P(t,s)= p(t,B)p(s|t,B)$$

The probability that t takes a sensitive value s must be less than or equal to 1/ l. thus every tuple t in the table satisfies l-diversity for any sensitive value s then the suppression sliced table satisfies l-diversity and it is proven in [1] [6]. We can also use suppression slicing for other type of privacy measures such as t-closeness [5].

8.3. Similar tuple check algorithm

```
Similarity check (T, d)
Q ={T};
  While Q is not empty
  Remove first two tuples t1, t2 from Q
  Q=Q-{t1, t2}
  For each attribute Ai where 1≤i≤d
     if (t1[Ai]==t2[Ai] AND t1[Ai+1]==t2[Ai+1] )
        if (Ai+2 != Ad)
            t1 [Ai+2] = *;
            t2 [Ai+2] = *;
        else
            t1 [Ai+1] = *;
            t2 [Ai+1] = *;
return
```

In similarity check the similar quasi identifier values are identified between the two different tuples. Then replace the next QI value as suppressed value. Before suppressing the attribute value check that the field should not be the sensitive attribute field thus always the sensitive attribute be the last attribute in the data set.

9. COMPARISION WITH SLICING

Slicing and suppression slicing is similar in diversity check for the buckets then they differ in the way of slicing performed in the buckets. Thus slicing performs the random permutation within the buckets and in the suppression slicing performs the random permutation with in the bucket before that similarity check can be performed with in the buckets. As well as in the column generalization and in the attribute partitioning both is same. Thus in the membership disclosure protection they differ completely. In slicing membership disclosure may protect using the generation of fake tuples for the original tuples in the sliced table. The sliced table has two columns then the bucket have 4 original tuple then the fake tuples are $4^2 = 16$ tuples. In that twelve tuples are faketuples and the remaining four tuples are original tuples. Thus it reduces the utility of the dataset. But in the suppression slicing the protection of membership disclosure is obtained by the random permutation done with in the bucket along with

the suppression then if the attackers have the background knowledge they cannot obtain the identity of the member because their quasi identifier and the sensitive attributes are sliced in other buckets from their original bucket. Thus the utility of the dataset is maintained and also the membership disclosure is protected.

10. RESULTS AND DISCUSSION

The anonymization techniques play a major role in preserving privacy for publishing the data. Each anonymization techniques are differ in nature but it maintain the data utility and the time consuming for the process must be less in nature to achieve its efficiency. Previous anonymization techniques such as slicing, suppression and generalization process have a loss in utility than the proposed method suppression slicing. All the values in the table is present inside the table but it is permutated within the buckets then only some of the attribute values are suppressed this will leads the data miners with a sufficient utility of the database. The privacy is achieved here by applying the suppression only in the similar tuples within the buckets after that random permutation taken place within the buckets. The implementation results are discussed in the appendices.

The data utility and the time efficiency between the slicing and the suppression slicing are shown in the figures 3.1 and 3.2. The utility of the dataset is high in the suppression slicing when compared to the slicing by avoiding the generation of the fake tuples. The privacy is preserved by applying the suppression technique before performing the slicing process. By applying suppression in dataset requires some amount of time to find the similar tuples in the dataset. This time required is less than the time required for the generation of fake tuples.

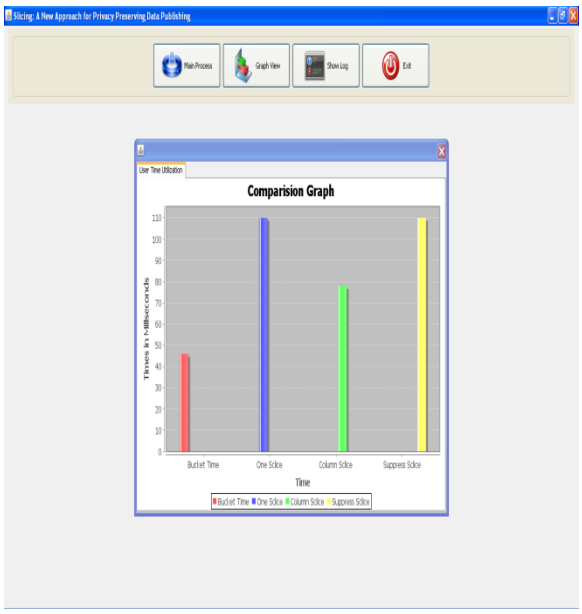

Figure 1 Time Comparison Graph

The bucket time taken for the suppression slicing is depends upon the user time required to select the quasi identifier under which the buckets are partitioned. The time required to complte the slicing (without generating fake tuples) is near to the time required to complete the suppression slicing.

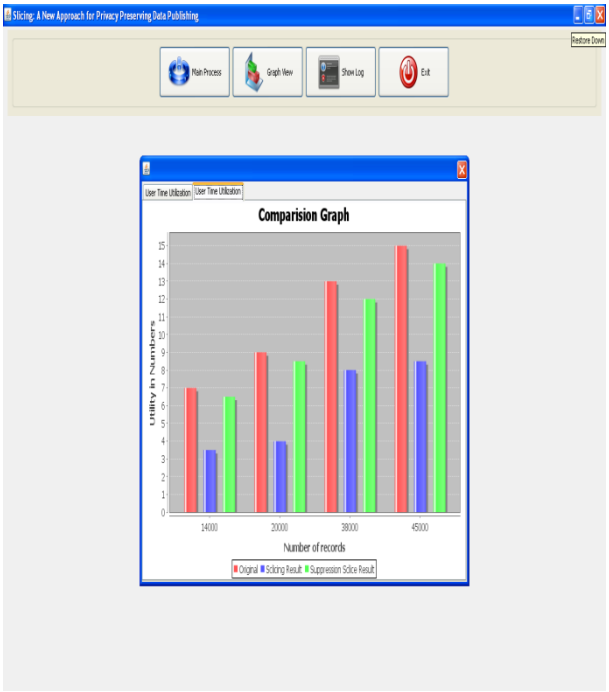

Figure 2 Utility Comparison Graph

10. CONCLUSION

This paper have the technique for preserve the privacy in data publishing and the utility of the data are also preserved for the data miners or for research scholars thus the suppression slicing gives better utility than the slicing and the membership disclosure protection is more secure than the bucketization, generalization and the slicing is discussed in the previous section. Diversity check ensures that the suppression slicing satisfies the privacy requirement of l-diversity. Suppression slicing provides better data utility, the attribute disclosure protection and the membership disclosure protection in a highly secured manner than the previous anonymization technique.

11. REFERENCES

[1] C. Aggarwal, 2005 On k-Anonymity and the Curse of Dimensionality, Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 901-909

[2] A. Inan, M. Kantarcioglu, and E. Bertino, 2009 Using Anonymized Data for Classification, Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE), pp. 429-440.

[3] Latanya Sweeney, 2002 achieving k-anonymity privacy protection using generalization and suppression1, proc International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 571-588.

[4] K. LeFevre, D. DeWitt, and R. Ramakrishnan, 2006 Mondrian Multidimensional k-Anonymity, Proc. Int'l Conf. Data Eng. (ICDE), p. 25.

[5] N. Li, T. Li, and S. Venkatasubramanian 2007 t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, Proc. IEEE 23rd Int'l Conf.Data Eng. (ICDE), pp. 106-115.

[6] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, 2006 l-Diversity: Privacy Beyond k-Anonymity, Proc. Int'l Conf. Data Eng. (ICDE), p. 24.

[7] V. Nivedhitha and S.Kiruthika, 2012 Privacy preservation using l-diversity, proc. Int'l conf. On computational intelligence and information technology, pp-DMB065-71.

[8] Tiancheng Li, Ninghui Li, Jian Zhang, and Ian Molloy 2011 slicing: A new approach for privacy preserving data publicing , IEEE transaction on knowledge and data engineering, vol.23, no.2.

[9] R.C.-W. Wong, A.W.-C. Fu, K. Wang, and J. Pei, 2007 Minimality Attack in Privacy Preserving Data Publishing, Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 543-554.

[10] R.C.-W. Wong, J. Li, A.W.-C. Fu, and K. Wang, 2006 (α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy Preserving Data Publishing, Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 754-759.

[11] X. Xiao and Y. Tao, 2006 Anatomy: Simple and Effective Privacy Preservation, Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 139-150.