Towards a Privacy Diagnosis Centre: Measuring *k*-anonymity

Mohammad Reza Zare Mirakabad[1], Aman Jantan[1] and Stéphane Bressan[2]
[1]*School of Computer Sciences, USM, Malaysia and* [2]*School of Computing, NUS, Singapore*
[1]*(reza.aman)@cs.usm.my and* [2]*steph@nus.edu.sg*

Abstract

Most of the recent efforts addressing the issue of privacy have focused on devising algorithms for the anonymization and diversification of data. Our objective is upstream of these works: we are concerned with privacy diagnosis. In this paper, we start by investigating the issue of k-anonymity. We propose algorithms to explore various questions about k-anonymity of data. Such questions are, for instance, "is my data sufficiently anonymous?", "which information, if available from an outside source, threatens the anonymity of my data?" In this paper we focus on anonymity and, in particular, k-anonymity. The algorithms that we propose leverage two properties of k-anonymity that we express in the form of two lemmas. The first lemma is a monotonicity property that enables us to adapt the a-priori algorithm for k-anonymity. The second lemma is a determinism property that enables us to devise an efficient algorithm for δ-suppression. We illustrate and empirically analyze the performance of the proposed algorithms.

1. Introduction

Privacy preservation ought to become a major concern for organizations and individuals publishing data. Organizations and professionals publish operational data to ensure business visibility and effective presence on the World Wide Web. Individuals publish personal data in the hope of becoming socially visible and attractive in the new electronic communication forums. As a result, large amounts of data, high level of details and the numerous sources are publically available.

Consequently, even though data may locally seem to respect privacy, cross referencing with external data and statistical inferences can disclose more than intended. For instance, while data publishers generally remove direct identifiers such as names, social security numbers, full addresses (a process referred to as de-identification) cross referencing and quasi-identifiers allow identification of individuals. In her seminal work [1], Latanya Sweeney strikingly illustrated her thesis by showing that she could retrieve the medical record of the governor of Massachusetts from publicly available and supposedly anonymous data.

Privacy preservation involves controlling anonymity and diversity of published data to prevent cross-referencing and inferences while maintaining sufficient usefulness. While anonymity prevents identity of individuals from being revealed in published data, diversity prevents unwanted disclosure of sensitive information. Anonymity and diversity are quantified by such notions as k-anonymity [1], l-diversity [2], (α,k)-anonymity [3] and t-closeness [4], for instance.

Processes transforming data (by generalization, suppression or fragmentation, for instance) to achieve required levels of anonymity and diversity are called anonymization and diversification, respectively. Most of the recent efforts addressing the issue of privacy have focused on anonymization and diversification.

Fewer efforts have been made to devise techniques, tools and methodologies that assist data publishers, managers and analysts in their investigation and evaluation of privacy risks. We propose the idea of a one-stop privacy diagnosis centre that offers the necessary algorithms for the exploratory analysis of the data and of various publication scenarios. Such a diagnosis centre should assist answering questions such as "is my data anonymous?", "is my data sufficiently diverse?" and "which information, if available from an outside source, threatens the anonymity of my data?"

In this paper, as a first step towards a privacy diagnosis centre, we focus on anonymity diagnosis. More specifically, we propose algorithms for the diagnosis of k-anonymity for relational data. We also consider the diagnosis of k-anonymity with δ-suppression. We first prove a monotonicity property of k-anonymity and leverage it to devise algorithms as a variant of the a-priori algorithm of [5]. We prove a determinism property of δ-suppression to devise efficient algorithms when considering tuple suppression.

978-0-7695-3428-2/08 $25.00 © 2008 IEEE
DOI 10.1109/CSA.2008.44

The rest of this paper is organized as follows. In section 2 we survey some state of the art related works with a focus on k-anonymity. In section 3, we give our working definitions for k-anonymity and δ-suppression. In section 4, we present the prototypical questions that our diagnosis centre can answer and propose the corresponding algorithms. In section 5, we illustrate and empirically analyze the performance of the proposed algorithms. Finally, we conclude our discussion in section 6 with some directions for future work.

2. Literature review

k-anonymization was first introduced and proposed by Samarati and Sweeney [1, 6] as a model for protecting privacy. They noticed the existence of quasi-identifiers, i.e. sets of attributes that can be cross-referenced in other sources and reveal identity. In their approach, data privacy is guaranteed by ensuring that any record in the released data is indistinguishable from at least (k-1) other records with respect to the quasi-identifier. That is each equivalence class (the set of tuples with the same values for the attribute in the quasi identifier) has at least k tuples. An individual is hidden in a crowd of size k, thus the name is k-anonymity.

Most of the works on k-anonymity concerns k-anonymization [1, 6-15]. Sweeney [9] proposes generalization and suppression. With generalization the value of an attribute is changed to a "less specific but semantically consistent value" [9]. For instance age is changed to an age range. Suppression can be applied to values or instances. With value suppression a value is not released, for instance replaced by a special value (e.g. '*'). With instance suppression instances are removed. For instance, selected tuples of a table are not published. We use a notion of δ-suppressed subset adapted from [6, 10, 14]. Usefulness or quality of the resulting data is measured by information loss metrics which we do not discuss here.

Lefevre et al. [14] use full-domain generalization, one specific version of global recoding proposed in [9], as the recoding model for generalization. In addition to generalization they use tuple suppression by removing a certain number of outliers to improve the quality. They prove that if Q is a subset of attributes in table T and T is k-anonymous with respect to Q then T is k-anonymous with respect any subset of Q (notice that we will adapt a slightly different definition of k-anonymity and reformulate this property in Lemma 1). They use this property to prune their search space when generating the generalization graph.

While k-anonymity is concerned solely with identity, *l*-diversity aims at protecting sensitive information. It ensures this protection by guaranteeing that one can not associate an identifier with sensitive information with a probability larger than 1/*l*. Many different notions contributing to privacy, together with the corresponding transformation processes, have been introduced that refine anonymity and diversity. For instance Anatomy [16], (α,k)-anonymity[3], [2], [4] and [7] are some of the proposed methods address this problem in different perspectives.

Recently authors of [17] have also proposed anonymity diagnosis algorithms. They are, however, concerned with fuzzy functional dependencies and fuzzy quasi-identifiers (although they don't explicitly use the term "fuzzy"), while we look at the conventional [1, 6, 18] notion of quasi identifiers for k-anonymity.

The monotonicity property of k-anonymity that we enounce and prove is related and similar to, but different from the monotonicity/anti-monotonicity property of anonymization given and proved in [2, 14, 19]. These latter properties are concerned with generalization/specialization of attribute values, while our proposal is concerned with adding and removing attributes in the candidate quasi identifiers.

3. Definitions

Let us present working definitions and the results that motivate our algorithms.

Definition 1 (Equivalence class with respect to a set of attributes). Given a multiset instance r of a relation R[1] and a set of attributes $S \subset R$; $t \subset r$ is an equivalence class with respect to S if and only if t is the multiset of tuples in r that agree on the values of their attributes in S.

These equivalence classes are the equivalence classes of the relation on tuples "have the same values for the attributes in S". The notion suggests the partitioning of the instances. The notion was introduced and the name was given in [3, 7].

Definition 2 (k-anonymity). Given an integer k, an instance r of a relation is k-anonymous with respect to $S \subset R$ if and only if the cardinality of every equivalence class with respect to S is greater or equal to k and r is not k+1 anonymous.

This definition of k-anonymity is compatible with but not identical to definitions given in other papers such as [1, 7, 9, 14, 15]. This is a recursive definition that chooses k to be exactly the minimum cardinality of an equivalence class with respect to S. Without this recursion ("not k+1 anonymous") an instance which is

[1] R is both the name of a relation and its schema (i.e. a set of attributes).

k-anonymous would also be k-1-anonymous.With the recursive definition it is not the case.

Lemma 1. (Monotonicity). If an instance r of R is k-anonymous with respect to S, then for any S' such that S⊂S', r is k'-anonymous with respect to S' with k' ≤ k.

Proof. *If r is k-anonymous with respect to S, then the minimum cardinality of every equivalence class with respect to S is k. For a superset S' of S, every equivalence class with respect to S' is included in an equivalence class with respect to S. This is because the tuples of an equivalence class with respect to S' agree on the values of their attributers in S' and therefore also agree on the values of their attribute in S. Therefore the minimum cardinality of the equivalence classes with respect to S' is less or equal to k.*

A consequence of Lemma 1 is that if an instance r of R is k-anonymous with respect to S, then for any S' such that S' ⊂ S then r is k'-anonymous with respect to S' with k'•k.

Definition 3 (δ-suppression). Given δ between 0 and 1 and an instance r of R, r' ⊂ r is an δ-suppressed subset of r if and only if the cardinality of r' is the ceiling of δ times the cardinality of r. δ is called the suppression threshold.

This notion of δ-suppressed subset is adapted from [6, 10, 14].

Lemma 2. (Deterministic suppression). Given an instance r of R, instance r' ⊂ r that is obtained by removing all and entire smallest equivalence classes is the minimum possible δ-suppressed (δ≠0) with the maximum k-anonymity.

Proof. *Constructing a δ-suppressed subset consists in removing some tuple. Removing tuples from each equivalence class decreases the cardinality of it and accordingly k value. However the value of k will be increased, only if all equivalence classes with smallest size entirely be removed.*

(Notice that this lemma can be used iteratively to approach the desired k-anonymity with minimum suppression.)

4. Algorithms

Armed with these results, we can now devise algorithms for privacy diagnosis. Lemma 1, the monotonicity lemma, allows us to adapt the a-priori algorithm. Lemma 2 allows us to devise a deterministic strategy for the suppression of tuples.

4.1 What questions can be asked?

A user who wants to publish data from a relation instance r of R needs to decide of the subset of attributes S of R that may constitute a quasi-identifier and needs to evaluate the risk for r. The user wants to know for which k r is k-anonymous with respect to S. If the user has no a priori idea of the quasi-identifiers, he can investigate which subsets S yield dangerous k values. The user might also be ready to accept some suppression to protect her data.

In this paper three parameters are considered for diagnosing anonymity of an instance: k, S, and δ. There are twelve possible questions. The list of questions is given below.

1. Is r k-anonymous with respect to S?
2. For which k is r k-anonymous with respect to S?
3. For which S is r k-anonymous with respect to S?
4. For which S and k is r k-anonymous with respect to S?
5. Is δ-suppressed r k-anonymous with respect to S?
6. For which k is δ-suppressed r k-anonymous with respect to S?
7. For which S is δ-suppressed r k-anonymous with respect to S?
8. For which k and S is δ-suppressed r k-anonymous with respect to S?
9. For which δ is δ-suppressed r k-anonymous with respect to S?
10. For which δ and S is δ-suppressed r k-anonymous with respect to S?
11. For which δ and k is δ-suppressed r k-anonymous with respect to S?
12. For which δ, k and S is δ-suppressed r k-anonymous with respect to S?

The questions that consider a given k may be asked as "is at least k-anonymous" or "is at most k-anonymous". For δ-suppression, we only consider the question "is at most δ-suppressed".

In this paper, we propose algorithms that answer questions 2, 3 and 6. These algorithms can be adapted to answer the other questions above.

The reader notices that every algorithm involving the suppression threshold can also output the actual tuples to be suppressed. We do not include this straightforward step in the algorithms presented.

4.2 Measuring *k*-anonymity given a quasi identifier S (Question 2)

Given a relation instance r of R and a set of attributes S ⊂ R, we can easily compute *k* such that r is *k*-anonymous with respect to S. This is the minimum cardinality of equivalence classes with respect to S. Equivalence classes are obtained by grouping tuples that agree on the values of S. This trivial algorithm is the elementary algorithm that is used for some other questions; hence we give it here for the sake of clarity. Algorithm 1 outlines this algorithm.

```
Input : r an instance of R and S ⊂ R
Output: k with respect to S
1.  rs = projection of r on S
2.  min = ∞
3.  for each distinct value of rs
4.    count number of it's similar tuple in c
5.    if (c < min) then
6.       min = c
7.       if (c = 1) then break end if
8.    end if
9.  end for
10. output min
```

Algorithm 1. Measuring k-anonymity given S

4.3 Finding the candidate quasi identifiers S that respect a given minimum k-anonymity (Question 3)

We wish now to find the sets of attributes of R that satisfy a given minimum k-anonymity for the instance r of R that we are studying. (An algorithm for the maximum k-anonymity can easily be derived from the algorithm that we present in this section.)

An obvious brute force algorithm enumerates the 2^n combinations of attributes of R and computes k for each of them. However Lemma 1 tells us that if r with respect to a set of attributes is not k-anonymous then it is not k-anonymous with respect to any super set of that set of attributes either. Consequently we consider subsets only if all their subsets are at least k-anonymous. Thanks to Lemma 1 we devise a level-wise algorithm which is similar to the a-priori algorithm [5] and can achieve significant pruning. The algorithm output the largest sets of attributes that are at least k-anonymous. Algorithm 2 outlines details of this algorithm.

We start from the first level and compute k for each subset of one attribute. A set is added to the level only if its K is greater than or equal to the given k (lines 2 to 6). Subsequent levels are processed in the nested loops of lines 7 to line 25. The nested loops create all sets of the next level by combining sets of the current level that differ symmetrically of exactly one element (line 11) (this generation is rather naïve for the sake of clarity and can be easily improved). At line 13 we check whether all subsets of a candidate exist in the previous level. If they not all exist, the set can be pruned. Then we compute K for this candidate set using Algorithm 1 (line 14) and add it to next level (N) only if it satisfies k-anonymity (line 15). Since all results are not necessarily in the last level, we add immediate ancestors (super sets in next level) of each candidate in NN. At line 22 if NN is empty, we add the current set to final result set (F). When this algorithm terminates, we have maximum subsets of attributes such that r is k-anonymous or k'-anonymous where k' > k with respect to them in F and given as output.

```
Input : r an instance of R and k
Output: all largest S ⊂ R for which r is k-
anonymous or k'-anonymous with k' > k
1.  P , N , F = ∅
2.  for Ai ∈ R do
3.    S = {Ai}
4.    K = (compute K for S using Algorithm 1)
5.    if K >= k then  P = P ∪ {S} end if
6.  end for
7.  while (P <> ∅ )
8.  for Si ∈ P do
9.   NN = ∅
10.  for Sj ∈ P do
11.   if Si-Sj and  Sj-Si are singleton then
12.    S = Si ∪ Sj
13.    if all subsets of S of cardinality
                        |S|-1 exist in P then
14.     K=(compute K for S using Algorithm 1)
15.     if K >= k then
16.      N = N ∪ {S}
17.      NN = NN ∪ {S}
18.     end if
19.    end if
20.   end if
21.  end for
22.  if ( NN == ∅ ) then F = F ∪ {Si} end if
23. end for
24. P = N , N = ∅
25.end while
26.output F
```

Algorithm 2. Finding S that respect a minimum (or maximum) k-anonymity

Using a similar idea we can also measure *k*-anonymity for all candidate quasi identifiers (Question 4). Again we exploit Lemma 1 to prune unnecessary computing. If r is 1-anonymous with respect to S then it is 1-anonymous with respect to any superset of S, then S will be pruned. Because of space limitation we do not give the details of this algorithm which is similar to Algorithm 2 above.

4.4 Measuring *k*-anonymity given a quasi identifier S and a suppression threshold δ (Question 6)

We wish to compute the value of maximum k that can be obtained by suppressing δ (rather δ times the cardinality of the instance) tuples or less. Using Lemma 2 we infer that in order to achieve maximum k-anonymity we need only remove entire equivalence classes of minimum cardinality. Algorithm 3 outlines the algorithm.

We compute the equivalence classes with respect to S and their cardinality from line 1 to 12. We suppress equivalence classes in ascending order of their cardinality (line 13 to 18) while the number of suppressed tuple is less than δ (test at line 16).

```
Input : r an instance of R, S ⊂ R and δ
Output: k for δ-suppressed r with respect to S
1.  rs = projection of r on S
2.  sort rs according to S
3.  define countArray as list of integers
4.  i = 0
5.  while ( rs )
6.     aTuple = rs.next()
7.     count = 1
8.     while (rs.next() == aTuple)
9.        count++
10.    end while
11.    countArray[i++] = count
12. end while
13. sort countArray in ascending order
14. partialSum = 0
15. j = 1
16. while partialSum < δ * |r|
17.    partialSum += countArray[j++]
18. end while
19. output countArray[j]
```

Algorithm 3. Measuring k-anonymity given S and δ

For answering to Question 7 (maximum subset of attributes satisfy k-anonymity after suppression) we use algorithm same as Algorithm 2. Only the difference is that we call Algorithm 3 instead of Algorithm 1 in lines 4, 14. We shall refer to this algorithm as Algorithm of Question 7.

5. Performance evaluation

We now evaluate the performance of our algorithms with the publicly available Adult data set from the UC Irvine Machine Learning Repository [20]. It has become a de facto benchmark for k-anonymization algorithms. We remove records with missing values as described and used in [7, 11, 15, 21]. The cleaned data set contains 30165 records. For the sake of simplicity we keep the following 8 attributes: {age, work class, education, status, occupation, race, sex, country}.

A Pentium V computer Intel(R) 2.4 GHZ with 1GB RAM was used to conduct our experiments. Operating system on the machine was Microsoft Windows XP. The algorithms were implemented, run and built by Java, Standard Edition 5.

We focus on the performance evaluation of Algorithm 2 and its variant incorporating δ-suppression and finding k-anonymity for all subsets. We evaluate the efficiency of the algorithms by counting the numbers of calls to the procedure "Measuring k-anonymity given S" (Algorithm 1) and "Measuring k-anonymity given S and δ" (Algorithm 3). We have verified that the curves below are commensurate to the run time. Instead of showing the curves for run time, for each curve we give the proportion of time for each procedure call. We use the adult data set described above for the following experiments.

We now evaluate the economy that Algorithm 2 can achieve by pruning some of the 256 subsets a naïve algorithm would visit.

Figure 1 shows the number of calls to the procedure "Measuring k-anonymity for given S" (Algorithm 1) for varying values of k from 1 to 50. We see that even for k as low as 2 we have 23 calls which is less than 10% of the numbers of calls needed for a naïve version. For k=50 there are 9 calls: an economy of more than 96%. From run time point of view each procedure call needs about 90 ms in our run.

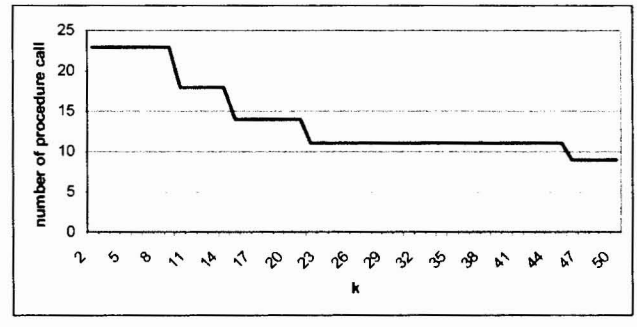

Figure 1. Number of calls to Algorithm 1 in Algorithm 2

We now look at the question of finding all maximum subsets of attributes that respect k-anonymity with 0.1-suppression (Question 7). Figure 2 shows the number of calls to the procedure "Measuring k-anonymity for given S with δ=0.1" (Algorithm 3) for k varying between 1 and 50 to find maximum subsets satisfy k-anonymity.

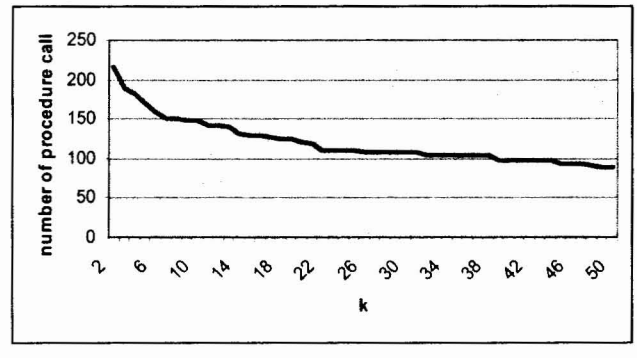

Figure 2. Number of calls to algorithm 3 for finding maximum subsets satisfy k-anonymity (δ=0.1)

Suppression removes equivalence classes with small size in order to increase k-anonymity exploiting the determinism of Algorithm 3 as provided by Lemma 2. Again lemma 1 helps improve on a naïve algorithm that needs to make 256 calls but as one expects the improvement is less than previous case since by suppression further subsets satisfy k-anonymity and aren't pruned. Then number of procedure calls is increased. The proportion of time related to each call is 300ms in this example

Now we look at cost for computing k for all subsets of attributes and different suppression thresholds. Figure 3 shows number of calls to the procedure "Measuring k-anonymity for given S with δ" (Algorithm 3) for varying suppression threshold δ. As the suppression threshold increases, we need to consider more combinations of attributes. Too high suppression create too many candidates and the performance of the algorithm converges quickly towards the worst case (naïve algorithm) performance. For this algorithm proportion of time for each procedure call is 490ms by average.

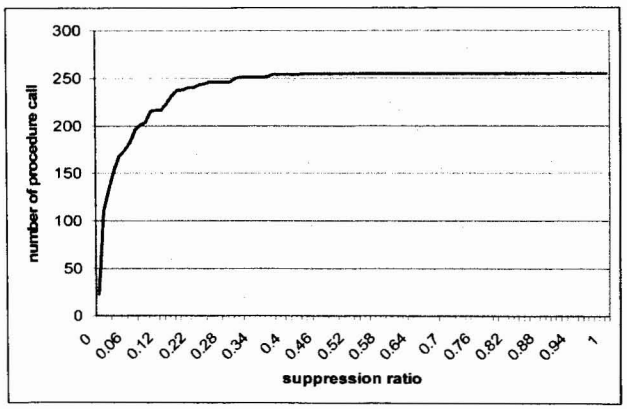

Figure 3. Number of calls to Algorithm 3 for finding k for all subsets for different δ thresholds

6. Conclusion and future work

In this paper, upstream of anonymization and diversification algorithms proposed in the recent literature, we propose the idea of a privacy diagnosis centre as a library of algorithms for measuring various notion of privacy such as k-anonymity and l-diversity. We make a concrete step towards this idea by presenting and evaluating several algorithms for measuring k-anonymity and k-anonymity with δ-suppression. We show that efficient algorithms can be devised thanks to a monotonicity property and a determinism property.

We are now exploring other metrics, such as l-diversity, for evaluating the privacy risk as well as information loss metrics for evaluating the consequences of anonymization and diversification.

7. References

[1] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy", *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems,* vol. 10, pp. 557-570, 2002.

[2] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-Diversity: Privacy beyond k-Anonymity", in *ICDE'06*, 2006.

[3] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, "(alpha, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy Preserving Data Publishing", in *KDD 2006*, 2006.

[4] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and *l*-Diversity", in *ICDE'07*, 2007, pp. 106-115.

[5] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", in *SIGMOD* Washington DC, USA, 1993.

[6] P. Samarati and L. Sweeney, "Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression", Technical Report SRI-CSL-98-04, 1998.

[7] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k-Anonymity Using Clustering Technique", Center for Education and Research in Information Assurance and Security, Purdue University 2006.

[8] C. C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality", in *VLDB'05,* 2005, pp. 901–909.

[9] L. Sweeney, "Achieving k-Anonymity Privacy Protection Using Generalizing and Suppression", *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems,* vol. 10, pp. 571-588, 2002.

[10] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering", in *PODS'06*, 2006.

[11] V. Iyengar, "Transforming Data to Satisfy Privacy Constraints", in *SIGKDD*, 2002, pp. 279–288.

[12] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based Anonymization Using Local Recoding", in *KDD'06*, 2006.

[13] B. C. M. Fung, K. Wang, and P. S. Yu., "Top-down Specialization for Information and Privacy Preservation", in *ICDE*, 2005, pp. 205–216.

[14] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-domain k-Anonymity", in *SIGMOD*, 2005, pp. 49-60.

[15] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity", in *ICDE*, 2006.

[16] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation", in *VLDB'06,* 2006, pp. 139-150.

[17] R. Motwani and Y. Xu, "Efficient Algorithms for Masking and Finding Quasi-Identifiers", in *VLDB '07*, 2007.

[18] P. Samarati, "Protecting Respondents' Identities in Microdata Release", *IEEE Transactions on Knowledge and Data Engineering,* vol. 13, pp. 1010-1027, 2001.

[19] K. Wang and B. C. M. Fung, "Anonymizing Sequential Releases", in *KDD'06,* 2006, pp. 414-423.

[20] C. Blake and C. Merz, "UCI Repository of Machine Learning Databases", 1998.

[21] R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization", in *ICDE'05*, 2005.