A Privacy Protection Model for Patient Data with Multiple Sensitive Attributes

Tamas S. Gal, Univeristy of Maryland, Baltimore, USA

Zhiyuan Chen, University of Maryland, Baltimore, USA

Aryya Gangopadhyay, University of Maryland, Baltimore, USA

ABSTRACT

The identity of patients must be protected when patient data are shared. The two most commonly used models to protect identity of patients are L-diversity and K-anonymity. However, existing work mainly considers data sets with a single sensitive attribute, while patient data often contain multiple sensitive attributes (e.g., diagnosis and treatment). This article shows that although the K-anonymity model can be trivially extended to multiple sensitive attributes, the L-diversity model cannot. The reason is that achieving L-diversity for each individual sensitive attribute does not guarantee L-diversity over all sensitive attributes. We propose a new model that extends L-diversity and K-anonymity to multiple sensitive attributes and propose a practical method to implement this model. Experimental results demonstrate the effectiveness of our approach.

Keywords: *data security; healthcare privacy issues; privacy protection*

INTRODUCTION

Patient data are often shared for research and disease control purposes. For example, the Center for Disease Control and Prevention has a National Program of Cancer Registries which collects data on cancer patients. Such data are made available to public health professionals and researchers to understand and address the cancer burden more effectively.

Privacy is one of the biggest concerns in sharing patient data because without appropriate protection, personal information is vulnerable to misuse. For example, identity theft remains the top concern among customers contacting the Federal Trade Commission (Federal Trade Commission, 2007). According to a Gartner study (Gartner Inc., 2007), there were 15 million victims of identity theft in 2006. Another study showed that identity theft cost U.S. businesses and customers $56.6 billion in 2005 (MacVittie, 2007). Therefore, legislation such as the Health Insurance Portability and Accountability Act

(HIPAA) requires that health care agencies protect the privacy of patient data. This article focuses on models that protect identity of patients and at the same time still allow analysis to be conducted on the sanitized data.

K-Anonymity and L-Diversity Privacy Protection Model: The two most commonly used privacy protection models for identity protection are K-anonymity (Sweeney, 2002b) and L-diversity (Machanavajjhala, Gehrke, Kifer, & Venkitasubramaniam, 2006). K-anonymity prevents *linking attack*, which recovers private information by linking attributes such as race, birth date, gender, and ZIP code with publicly available data sets such as voter's records. Such attributes that appear in both public and private data sets are called *quasi-identifiers*. The K-anonymity model divides records into groups with sizes ≥ K such that each group has identical value or range on quasi-identifier attributes.

Example 1: Figure 1 shows some patient records, where age is the quasi-identifier and disease type and treatment are sensitive attributes (i.e., attributes with privacy sensitive information). Figure 2 shows the anonymized data where the first four rows belong to the same group and have the same range of age. Linking attack cannot discover the identity of a patient using the age attribute because there are at least K (K = 4) patients with the same age range.

L-diversity further enhances K-anonymity by preventing another type of privacy attack called *elimination attack* (which was used by Sherlock Holmes to solve mysteries by excluding the impossible). We use an example to illustrate elimination attack. In Figure 2, if K=3, then the first three patients satisfy 3-anonymity. However, they have only two different disease type values: heart disease and flu. If someone knows that the patient with ID 3 is unlikely to have heart disease, then he can infer that the patient most likely has flu.

L-diversity prevents elimination attack by requiring that the values of privacy sensitive attributes (e.g., the attribute disease type) in a group have enough degree of diversity. Several

Figure 1. Original patient data

Patient ID	Age	Disease Type	Treatment
1	42	Heart disease	Medicine
2	41	Heart disease	Surgery
3	49	Flu	Intravenous therapy
4	43	Stomach disease	Intravenous therapy
…	…	…	…

Figure 2. Anonymized patient data with K=4

Patient ID	Age	Disease Type	Treatment
1	41-50	Heart disease	Medicine
2	41-50	Heart disease	Surgery
3	41-50	Flu	Intravenous therapy
4	41-50	Stomach disease	Intravenous therapy
…	…	…	…

different definitions of diversity were proposed in Machanavajjhala et al. (2006). In this article, we use the definition of L-diversity where a sensitive attribute in a group contains at least *L* different values. For example, in Figure 2, the group of first four patients has three different disease types and is thus 3-diverse. If the attacker knows that a patient does not have heart disease, he cannot decide which type of disease the patient has because he may have either stomach disease or flu. In general, L-diversity can protect privacy against attackers who are able to eliminate up to *L-2* values.

Problem of data with multiple sensitive attributes: There has been a rich body of work on implementation and application of K-anonymity and L-diversity (Bayardo & Agrawal, 2005; LeFevre, DeWitt, & Ramakrishnan, 2005; Menon & Sarkar, 2006; Samarati, 2001; Xiao & Tao, 2006). However, existing research considers data sets with only one sensitive attribute, while many data sets such as patient data have multiple sensitive attributes (e.g., disease and treatment). Since K-anonymity does not have any condition on sensitive attributes, it can be applied to data sets with multiple sensitive attributes. However, we will next show that L-diversity cannot be directly applied to data sets with multiple sensitive attributes.

Suppose in Figure 2 both disease type and treatment are sensitive attributes. The group of the first four patients has three distinct values on both attributes. However, if the attacker knows that a patient does not have heart disease, he can decide that this patient has IV (intravenous therapy) as treatment because only patients with heart diseases received the other two types of treatment. Thus this group is not 3-diverse. The root cause of this problem is that the elimination of rows containing one sensitive attribute value may eliminate multiple values of other sensitive attributes. In this case, the elimination of rows containing the value heart disease (i.e., the first two rows) also eliminates values medicine and surgery. Therefore, preserving L-diversity on each individual sensitive attribute will not preserve L-diversity for multiple sensitive attributes.

Our contributions: This article has made the following contributions.

- We propose a privacy model that extends K-anonymity and L-diversity to data with multiple sensitive attributes.
- We provide a method to implement our model. Experimental results show that our method also introduces little distortion to data, which will help subsequent data analyses.

The rest of the article is organized as follows. The next section describes related work. We then describe the proposed privacy model for multiple sensitive attributes and describe how to implement the model. Finally we present the experimental results and conclude the article.

Related Work

The existing work on privacy protection techniques can be divided into two categories: those that protect personal identity (called data anonymization) and those that protect sensitive attribute values or sensitive patterns. We first review literature in both categories and then describe the relationship of this articles to the literature.

Data anonymization: The research in this field is based on two privacy protection models: K-anonymity and L-diversity. The K-anonymity model was first proposed by Sweeney (2002b) and it protects the data against linking attacks. The L-diversity model was proposed as a complement to K-anonymity in Machanavajjhala et al. (2006). L-diversity further protects the data against elimination attack. It is a general consensus in the field that both models need to be implemented to protect privacy.

There are two approaches to implement K-anonymity and L-diversity. The first approach is generalization, that is, replacing values of quasi-identifiers with more general values (Samarati, 2001; Sweeney, 2002a). The second approach is called anatomy and it divides data into multiple tables, one storing the quasi-identifier attributes and group ID of

each record, and the others storing the count of sensitive attribute values in each group (Xiao et al., 2006). This approach introduces less distortion to data (Xiao et al., 2006). LeFevre et al. (2005) proposed a method based on full domain generalization, which maps the domain (or range) of attribute values to a more general domain. For example, all five-digit zip codes can be generalized to a domain that contains only the first four digits (e.g., 21250 will become 2125*). A method that uses k-d tree to divide data into groups was proposed in LeFevre, DeWitt, and Ramakrishnan (2006). K-d tree puts data with similar values into the same group, leading to less data distortions.

Hiding sensitive values and patterns: Research in this field is also called privacy-preserving data mining because the goal is to preserve privacy and at the same time allow data mining on the modified data. A survey can be found in Verykios, Bertino, Fovino, Provenza, Saygin, and Theodoridis (2004a). The most well known method in this field is random perturbation, which adds some random noise to sensitive attribute values (Agrawal & Aggarwal, 2001; Agrawal & Srikant, 2000; Zhu & Liu, 2004). However, Kargupta, Datta, Wang, and Sivakumar (2003) showed that random perturbation method is subjected to attacks using correlations of data. A similar method for association rule mining was proposed in Evfimevski, Gehrke, and Srikant (2003) and Evfimevski, Srikant, Agrawal, and Gehrke (2002). A tree-based approach (Li & Sarkar, 2006) was also proposed. The basic idea is to divide data into groups using k-d tree, and replace values of sensitive attributes with the average of their values within a group.

There has also been work to hide sensitive patterns such as association rules in the data (Hintoglu, Inan, Saygin, & Keskinoz, 2005; Menon & Sarkar, 2006; Menon, Sarkar, & Mukherjee, 2005; Oliveira & Zaiane, 2002; Saygin, Verykios, & Clifton, 2001; Verykios, Elmagarmid, Elisa, Saygin, & Elena, 2004).

Relationship of this article to existing research: This article focuses on data anonymization (i.e., the first category) for two reasons. First, it is important to hide the identity of patients. Second, many privacy protection methods proposed in the second category of research only work for numerical attribute values, while patient data contain many categorical attributes. For example, for the tree-based approach (Li et al., 2006), it is unclear how to compute the average of a categorical attribute such as disease type.

As mentioned in the Introduction, existing work on data anonymization mainly considers the data set with only one sensitive attribute, while patient data often contains multiple sensitive attributes. We have also shown in the Introduction that the L-diversity model cannot be directly extended to multiple sensitive attributes. This article proposes a privacy protection model that works for multiple sensitive attributes.

Privacy Model over Multiple Sensitive Attributes

This section proposes our privacy model. The first subsection reviews the existing K-anonymity and L-diversity models over a single sensitive attribute. The second subsection proposes a novel privacy model over multiple sensitive attributes. The third subsection proposes a variant of our model that deals with data sets with very few distinct values on some sensitive attributes.

Existing Model

Let T be the data to be shared. T contains d quasi-identifiers attributes A_{Q1}, ..., A_{Qd}, and m sensitive attributes A_{S1}, ..., A_{Sm}. Based on the literature (Sweeney, 2002b), we assume quasi-identifier attributes and sensitive attributes have no overlap because sensitive attributes normally do not appear in public available data sets. Next we give a few definitions.

Definition 1: (Partition/Group) *A* **partition** *consists of several subsets of T, such that each record in T belongs to exactly one subset. We refer to these subsets as* **groups** *and denote them as* G_1, ..., G_p.

Next we describe two approaches to implement K-anonymity and L-diversity: generalization (Samarati, 2001; Sweeney, 2002a) and anatomy (Xiao & Tao, 2006).

Definition 2: (Generalization) *Given a partition of T, generalization of T makes all records in group G_i to have the same values on quasi-identifier attributes A_{Q1}, ..., A_{Qd}. For example, Figure 2 in the Introduction shows a generalization of T, where the first four records belong to one group. Numerical values such as ages can be generalized to ranges, and categorical values can be generalized according to a predefined hierarchy.*

Definition 3: (Anatomy) *Anatomy of T produces a quasi-identifier table (QIT) and m sensitive tables (ST) as follows. The QIT stores all quasi-identifier attributes A_{Q1}, ..., A_{Qd} and the group ID of each record. Each ST stores the group ID, distinct values of a sensitive attribute A_{Sj} ($1 \le j \le m$), and the count of the value of A_{Sj} in each group.*

For example, Figure 3 shows anatomy of T where the table on the left is the QIT table, and the other two tables are ST for sensitive attribute "disease type" and "treatment." It has been shown in Xiao and Tao (2006) that anatomy and generalization are equivalent in terms of privacy protection. For example, suppose an attacker knows that a patient's age is 49 (the third patient). All four patients in the generalized table (Figure 2) have age in the range of 41–50. Thus, the attacker cannot decide which record belongs to this patient. He can do a random guess and the probability of getting the correct disease type (flu) is 25% because only one of four patients has the flu. Similarly, when the attacker looks at the anatomized table in Figure 3, the attacker can learn that the patient with age 49 belongs to the first group using the QIT table. However, he cannot figure out the exact disease type of the patient because the ST table only stores counts of disease types for the first group. Thus the attacker can only do a random guess of the patient's disease type and the probability of a correct guess is again 25%. This article uses the anatomy approach to implement our privacy model because as shown in the work of Xiao and Tao (2006), anatomy often leads to a smaller degree of data distortion and benefits subsequent analysis of anonymized data.

Definition 4: (K-anonymity). *A data set T satisfies K-anonymity if it is divided into a partition and each group G_i ($1 \le i \le p$) in the partition contains at least K records, and T is either generalized or anatomized.*

Definition 5: (L-diversity for a single sensitive attribute). *A data set T satisfies L-diversity if it is divided into a partition and the sensitive attribute A_{S1} contains at least L different val-*

Figure 3. Anatomized patient data

Patient ID	Age	Group ID
1	42	1
2	41	1
3	49	1
4	43	1
…	…	…

Group ID	Disease Type	Count
1	Heart disease	2
1	Flu	1
1	Stomach disease	1
…	…	…

Group ID	Treatment	Count
1	Medicine	1
1	Surgery	1
1	IV	2
…	…	…

ues in each group G_i. T is also generalized or anatomized.

The data in Figure 2 satisfies 4-anonymity because each group contains at least four records. If disease type is the only sensitive attribute, this data set is also 3-diverse. It is clear that K-anonymity has no condition on sensitive attributes, thus K-anonymity model applies to data sets with any number of sensitive attributes. However, as shown in the Introduction, L-diversity cannot be directly extended to multiple sensitive attributes because making each sensitive attribute L-diverse does not guarantee that all sensitive attributes are L-diverse.

Our Model

This section presents our privacy model. We assume that when a distinct sensitive attribute value is deleted from a group, all rows containing that value will be deleted. For example, if the value "heart disease" is deleted in Figure 1, the first two rows are both deleted.

Definition 6. (K-anonymity and L-diversity for multiple sensitive attributes). *T satisfies both K-anonymity and L-diversity if T is divided into a partition and each group G_i ($1 \le i \le p$) in the partition satisfies the group containing at least K records, and (2) at least L distinct values (possibly from multiple sensitive attributes) need to be deleted to delete all rows in the group. T is also anatomized or generalized.*

The first four patients in Figure 2 satisfy 4-anonymity. They also satisfy 2-diversity because we need to delete at least two values (e.g., "heart disease" and "IV") to delete all the rows in the group. The new model survives linking attack because it ensures K-anonymity. The next theorem shows that it also survives elimination attacks.

Theorem 1: *Our privacy model survives elimination attacks with up to L-2 values, the same as the L-diversity model over a single sensitive attribute.*

Proof: *If the attacker eliminates L-2 sensitive attribute values, the remaining rows must have at least two distinct values for each sensitive attribute. This is because if the remaining rows have only one distinct value, all rows in the group will get deleted after deleting L-1 values, and this violates the second requirement of L-diversity.*

Machanavajjhala et al. (2006) proposed a definition of L-diversity over multiple sensitive attributes where each sensitive attribute must have at least L distinct values for records that have the same values on all other attributes. For example, if there are two sensitive attributes, disease type and treatment, then each disease type must have L corresponding treatment values and vice versa. For a data set with m sensitive attributes, the group must have at least L^m rows. Note that rows in the same group will be generalized or anatomized. Thus having such large group sizes will introduce too much data distortion, making the data not useful for subsequent analysis. We will compare our model with this model experimentally in experimental evaluation.

Variant of Our Model with Column-wise Constraints

The model proposed so far treats all sensitive attributes uniformly. However, in practice some sensitive attributes may have very few distinct values while other attributes may have many. Thus sometimes it makes sense to allow a different degree of diversity on different attributes. We propose a variant of our model that adds this flexibility.

Definition 7. (Privacy model with column-wise *constraints*). *T satisfies both K-anonymity and L-diversity with column-wise constraints if T is divided into a partition and each group G_j ($1 \le j \le p$) in the partition satisfies that: (1) the group contains at least K records, and (2) to delete all rows in the group, at least L distinct values need to be deleted, and at most L_i ($0 \le L_i \le L$) of them are from sensitive attribute A_{Si}. T is also anatomized or generalized.*

Figure 4. Example for column-wise constraints

Disease Type	Treatment
D1	T1
D2	T1
D3	T2
D4	T2
D5	T3
D6	T3

The definition is the same as the general model in Definition 6 except that at most L_i values will be deleted from attribute A_{Si}. For attributes with few distinct values, the user can set a $L_i < L$ such that fewer values can be deleted from this attribute. For attributes with many distinct values, the user can set $L_i = L$ (i.e., still allow L values to be deleted from that attribute).

Consider the data set in Figure 4 (quasi-identifier attributes are not shown). Let L_1 and L_2 be the constraints on disease type and treatment, respectively. Disease type attribute has six distinct values and treatment attribute only has three distinct values. Thus, we can set $L_2 = 2$ for treatment such that at most two treatment values can be deleted. After the deletion of two treatment values (e.g., T1 and T2), there are always two disease types left (e.g., D5 and D6). Hence, at least four deletions (e.g., deleting T1, T2, and D5, D6) are needed to delete all rows and $L = 4$. We set $L_1 = 4$ because disease type has many values. Without column-wise constraints, L can be 3 at most because there are only 3 distinct treatment values. The next theorem shows the relationship of these two models.

Theorem 2: *If a group G satisfies L-diversity without column-wise constraints (the general model), then G also satisfies L-diversity with constraints where* $L_i \leq L$.

Proof: *Suppose G satisfies the L-diversity model without column-wise constraints. If G is not L-diverse with constraints, then there exists L' < L such that G satisfies L'-diversity with constraints. Thus all rows in G will be deleted after deleting L' values (with at most* L_i *of them from attribute* A_{Si}*). This conflicts with the condition that G satisfies L-diversity without constraints (i.e., at least L (L>L') values need to be deleted to delete all rows in G). Thus G satisfies L-diversity with constraints. The reverse of Theorem 2 is not true. For example, the data in Figure 4 satisfies the column-wise model with* $L=L_1=4$, $L_2=2$, *but not the general model with L=4.*

Method to Implement the Model

This section describes how to implement our privacy model. The first subsection shows an overview. The second subsection proposes a method to check L-diversity. The third subsection shows how to check L-diversity with column-wise constraints.

Overview

Figure 5 shows the algorithm to anonymize the data. It has three input parameters: the data set *T,* and the parameter *K* and *L* in our privacy model. It contains two steps. In the first step, data are divided into a partition such that each partition contains at least *K* records and satisfies L-diversity. In the second step data are anatomized. Next we describe these two steps.

Figure 5 also shows the algorithm (Split) to partition the data. The algorithm follows the K-d tree approach to generate the partition (LeFevre et al., 2006). The benefits of K-d tree is that records with similar values will be put in the same group, thus there will be less data distortion. The algorithm works top down, that is, starting with the whole data set as a single group *G* and then splitting the existing groups into smaller groups. The algorithm stops when further splits will violate K-anonymity or L-diversity conditions.

At line 1 the algorithm selects a quasi-identifier attribute to split. Following the literature (LeFevre et al., 2006), we select the splitting attribute as follows. We first normalize each quasi-identifier attribute by subtracting the mean and then dividing the difference by the standard

Figure 5. Anonymize algorithm

```
Anonymize(data set T, K, L)
    1)  P = empty set
    2)  Split(T, K, L, P)
    3)  Anatomize(T, P)

Split(Current group G, K, L, partition P)
    1)  A_Qj = choose_dimension()
    2)  splitVal = find_median(G, A_Qj)
    3)  G_L = {records in G and with value on A_Qj ≤ splitVal}
    4)  G_R = {records in G and with value on A_Qj > splitVal}
    5)  if Satisfy-Model(G_L, K, L)
    6)    Split(G_L, K, L, P)
    7)  if Satisfy-Model(G_R, K, L)
    8)    Split(G_R, K, L, P)
    9)  if neither G_L nor G_R satisfies our model
    10)   Add T to P
Satisfy-Model(G, K, L)
    1)  if |G| ≥ K and L-diverse(G)
    2)   return true
    3)  else return false
```

deviation of that attribute (i.e., computing the Z-score). We then select the attribute with the largest range. For categorical attributes, we first represent them using integer values such as 1, 2, 3 (this is often already done in many patient data sets because categorical attributes are often represented by integer code), and then use the above method. Note that we only need to apply this conversion to quasi-identifier attributes. Sensitive attributes are not touched.

At line 2 to line 4, the algorithm splits the current data into two groups G_L and G_R by the median of the selected attribute. At line 5 to 8, the algorithm checks whether G_L and G_R satisfy our privacy model. The check for K-anonymity is straightforward: the group is K-anonymous if the group contains at least K records. The check for L-diversity is more complicated and will be discussed in the following subsections.

If G_L or G_R satisfies our model, it will be split further by recursively calling the algorithm. If neither of them satisfies our model, no further split is possible. Thus at line 10 the algorithm adds the current group G to the partition. Once the partition is generated, the data set T will be anatomized. Based on Xiao et al. (2006), we use the QIT table to store the quasi-identifier attributes and group ID for each record. The count of values of each sensitive attribute is stored in a separate ST table.

Complexity of our algorithm: let n be the number of records, m be the number of sensitive attributes, d be the number of quasi-identifier attributes, and $|G|$ be number of rows in G. The cost of building the k-d tree is $O(d\ n \log n)$ (LeFevre et al., 2006), excluding the cost of checking L-diversity. The following subsection will show that the cost of checking L-diversity for a group G_i is $O(|G_i|\ m + |G_i|\ log\ |G_i|)$. Note that in each level of the k-d tree, the union of all groups equals the complete data set T. Thus the cost of checking L-diversity for one level of k-d tree is $O(n\ m + n \log n)$. There can be at most $O(\log n)$ levels of the tree. Thus the total cost of checking L-diversity is $O(\ m\ n \log n + n\ (\log n)^2)$. The total cost of partitioning is thus $O((m+d)\ n \log n + n\ (\log n)^2)$. The cost of generating the anatomized data is $O((m+d)\ n)$ because the data only needs to be scanned once to generate the ST and QIT tables. Therefore, the total cost of the algorithm is $O((m + d)\ n \log n + n\ (\log n)^2)$. The last subsection will

show that the cost for checking column-wise constraints is $O(|G_i|\ m \log m + |G_i| \log |G_i|)$ and the total cost is $O((m \log m + d)\ n \log n + n (\log n)^2)$. Since $\log m$ and $\log n$ are quite small, both costs are almost linear with the data size $(m+d)\ n$.

Checking L-Diversity

This section describes how to check L-diversity for a group.

Theorem 3: *Checking for L-diversity for multiple sensitive attribute is NP hard.*

We can prove this theorem by reducing the minimal set cover problem to the problem of checking L-diversity. The detail of the proof is omitted due to space constraints. Here we just give some intuition about how we link these two problems. For each sensitive attribute value *v*, we create a set *RID(v)* that records the IDs of rows that contain that value. For example, consider the group of first four records in Figure 1; we have RID(heart disease)={1,2}, RID(flu) = {3}, RID(stomach disease)={4}, RID(medicine)={1}, RID(surgery)={2}, and RID(IV) = {3,4}.

The definition of L-diversity means that at least *L* values need to be deleted to delete all rows in the group. This is equivalent to state that at least *L* RID sets are needed to cover all rows in the group. In the above example, at least two RID sets are needed to cover the group of four records. For example, we can choose RID(heart disease)={1,2} and RID(IV)={3,4}.

However, finding the minimal number of RID sets to cover all row IDs is the minimal set cover problem which is NP hard. Thus we use a heuristic algorithm to check L-diversity in polynomial time. The next two theorems give the basis of our algorithm.

Theorem 4: *If there exists a set C of at least L rows, and no two rows have the same value on any sensitive attribute, then C is L-diverse.*
Proof: *If no two rows have the same value on any sensitive attribute, then deleting one distinct sensitive value can delete at most one row in C. Thus at least L such deletions are needed and C is L-diverse.*

Theorem 5: *If a subset C of a group G is L-diverse, G is at least L-diverse.*
Proof: *If at least L distinct sensitive attribute values need to be deleted to delete a subset*

Figure 6. Algorithm to check L-diversity

L-diverse(*G*)
1) compute for each sensitive attribute value *v* the number of rows in *G* contains that value, call it *f(v)*
2) for each row r_i, compute a total frequency $f(r_i) = \Sigma\ f(v)$ for *v* in r_i
3) sort rows in ascending order of total frequency, store them in *G'*
4) *C* = empty set
5) while *G'* is not empty
6) pick the row r_i with the minimal $f(r_i)$, delete it from *G'*
7) add r_i to *C* if it does not share any common value with existing rows in *C*
8) return yes if C contains *L* rows
9) end while
10) return no

of a group, at least that many deletions are needed to delete the whole group. The heuristic algorithm tries to find a subset C that has no common values on sensitive attributes. Figure 6 shows the algorithm. It starts with an empty set C, and then repeatedly adds rows to this set when no two rows in the set have common values on any sensitive attribute. Line 1 to 3 also compute the total frequencies of sensitive attribute values in a row and sort rows in ascending order of the frequency. This allows us to add rows that have values that are less frequent in the group first. The intuition is that such rows have smaller chances of sharing common values on sensitive attributes.

For the group of the first four records in Figure 1, suppose $L = 2$. Line 1 computes the frequencies of all sensitive attribute values. Thus, we have f(heart disease) = f(IV) = 2, and f(flu) = f(stomach disease) = f(medicine) = f(surgery)=1. Line 2 computes the total frequency of each row. The total frequencies of all rows are 3. Suppose row 1 is added to *C*. Next we try row 2, but it shares the disease type value with row 1. Thus we try row 3 and it is added to *C*. Now *C* contains 2 rows and the algorithm returns yes because *L=2*.

Let *m* be the number of sensitive attributes and $|G|$ be the group size. The frequency of values and rows can be computed in $O(m\ |G|)$ time. The sort at line 3 takes O(|G| log |G|) time. We can use a hash table to keep track of the values of sensitive attributes in *C*. Thus checking whether a row contains values already in *C* (line 7) takes O(m) time. Since at most |G| rows can be added to *C*, the total complexity of the algorithm is thus O(m |G| + |G| log |G|).

The algorithm is sound in the sense that for any group that the algorithm returns yes, the group is indeed L-diverse. Thus using this algorithm will not affect privacy protection. On the other hand, the algorithm does not check all possible subsets of the group (doing so requires exponential time). Thus some of the groups may be L-diverse but the algorithm may return no. In consequence, the algorithm may generate groups larger than the optimal case because these groups may be split further. This is the price we pay for not spending exponential time.

Checking L-Diversity with Column-wise Constraints

This section presents the algorithm to check L-diversity with column wise constraints.

Figure 7. Algorithm to check L-diversity with column-wise constraints

```
L-diverse-Column(G)
    1)  find a subset C with no common sensitive attribute values using the algorithm in Figure 6.
    2)  If |C| ≥ L, return yes.
    3)  sort rows not in C in ascending order of total frequency and store them in G'
    4)  x=|C|
    5)  while G' is not empty
    6)    pick row r_j in G' with the minimal f(r_j), delete it from G', add r_j to C
    7)    select values v_1, …, v_{x+1} in C with the highest frequencies in C and with at most L_i values
          from sensitive attribute i.
          Let v_i's frequency in C be f_C(v_i)
    8)    if Σ f_C(v_i)> |C|, 1 ≤ i ≤ x
    9)      delete r_j from C // C is not x-diverse
    10)   else
    11)     if Σ f_C(v_i)≤|C|, 1 ≤ i ≤ x+1
    12)       x = x+1 // C is x+1-diverse
    13)       if x = L return yes
    14)   end while
    15)   return no
```

Figure 8. Example of check L-diversity with column-wise constraints

Step 1: initial C

D1	T1
D3	T2
D5	T3

|C| = 3, C is 3-diverse

Step 2: add row (D2, T1) to C

D1	T1
D3	T2
D5	T3
D2	T1

3 most frequent values: T1(2), T2(1), D1(1)
Total frequency = 4, C is still 3-diverse

Step 3: add row (D4, T2) to C

D1	T1
D3	T2
D5	T3
D2	T1
D4	T2

3 most frequent values: T1(2), T2(2), D1(1)
Total frequency = 5, C is 3-diverse

Step 4: add row (D6, T3) to C

D1	T1
D3	T2
D5	T3
D2	T1
D4	T2
D6	T3

4 most frequent values: T1(2), T2(2), D1(1), D2(1)
Total frequency = 6, C is 4-diverse

Figure 7 shows the algorithm. Figure 8 shows an example of how the algorithm works. The algorithm consists of two steps. In the first step (line 1), it finds a subset C without common sensitive attribute values as in the algorithm in Figure 6. For example, for the data set in Figure 4, C will contain 3 records as shown in Step 1 in Figure 8.

Now there are two possible cases. In the first case, C contains at least L rows. By Theorem 4, C is L-diverse without column-wise constraints. By Theorem 2, C is also L-diverse with column-wise constraints. Using Theorem 5, the group G is also L-diverse and the algorithm returns yes.

In the second case, C contains less than L rows. The example in Figure 8 is in this case because C contains 3 rows and L=4. The algorithm repeatedly adds rows to C and check whether C satisfies L-diversity. Each round the algorithm adds a remaining row with the minimal total frequency to C (line 6) and checks C's diversity. A variable x is used to keep track of C's diversity and $x = |C|$ initially.

The algorithm uses the following lemma and theorem to check C's diversity.

Lemma 1: *Let $v_1, v_2, \ldots, v_x$ be the values of sensitive attributes in a C, and $f_C(v_i)$ be the frequency of v_i in C. Let V be a set of values to be deleted. Let SumF(V) = $\Sigma f_C(v_i)$, where v_i is in V. Then the maximal number of rows to be deleted is SumF(V).*

Proof: *Let RID(v_i) be the IDs of rows containing value v_i. Size of RID(v_i) = $f_C(v_i)$. When all values in V are deleted, the set of rows that gets deleted is the union of all RID(v_i) for v_i in V. The size of union is at most the sum of sizes of each RID sets, which equals SumF(V).*

Theorem 6: *Suppose frequencies of values are sorted in descending order, i.e., $f_C(v_1) \geq f_C(v_2) \geq \ldots \geq f_C(v_x)$. Suppose set V contains the L most frequent values, with at most L_i values from*

attribute A_{Si} (if A_{Si} has more than L_i values, we select the L_i most frequent ones). If SumF(V) ≤ |C|, then C is L diverse with column-wise constraints.

The proof is straightforward. Consider any set *U* containing *L* values. By Lemma 1, deleting *U* will delete at most *SumF(U)* rows. Since set *V* contains the most frequent values, *SumF(U) ≤ SumF(V)*. Since *SumF(V) ≤ |C|*, *SumF(U) ≤ |C|*. Thus at least *L* deletions (with at most L_i from attribute A_{Si}) are needed to delete all rows in *C* and *C* is L-diverse.

Based on Theorem 6, the algorithm checks *C*'s diversity as follows. At line 7, the algorithm selects the *x* most frequent values from *C*, with at most L_i values selected from attribute *i*. At line 8, the algorithm checks whether the total frequency of these values is greater than size of *C*. If so, the newly added row is rejected because we cannot prove that *C* is x-diverse using Theorem 6. Otherwise, we can prove that *C* is x-diverse and the algorithm keeps the newly added row in *C*. Next, the algorithm checks whether *C* is *x+1* diverse by computing the total frequency for the *x+1* most frequent values. If the total frequency is less or equal to the size of *C*, *C* is x+1 diverse and *x* is increased by 1 at line 12. Finally, if the value of *x* reaches *L*, *C* is L-diverse. By Theorem 5, *G* is also L-diverse and the algorithm returns yes.

For example, consider the data in Figure 4. Figure 8 shows the process of the algorithm where the numbers in parenthesis are frequencies. At step 2, the row with value *(D2, T1)* is added to *C*, and the total frequency of the 3 most frequent values *(T1, T2, D1)* equals the size of *C*. Thus *C* remains 3-diverse. At step 3, the row with value *(D4, T2)* is added and C is still 3-diverse. At step 4, the row with values *(D6, T3)* is added. Now the 4 most frequent values are *T1, T2, D1*, and *D2*. Note that T3 is not counted because at most 2 treatment values can be selected according to the column-wise constraint $L_2=2$. The total frequency is 2+2+1+1=6, which equals size of *C*. Thus *C* is 4-diverse. Since *L=4*, the algorithm returns yes.

Complexity: This algorithm calls the algorithm in Figure 6 first. Thus it takes at least *O(m |G| + |G| log |G|)* time. Line 7 is the most expensive step because it needs to find the *x* most frequent values in *C*. Each time only the *m* values in the new row r_j will get their frequency increased by one (the frequency of other values stay the same). Thus the *x* most frequent values must come from the list of *m* values in the new row and the list of previously *x* most frequent values. If the previously most frequent values are already sorted, we just need to sort the *m* new values on their frequency and merge them with the previous list. Thus line 7 can be done in *O(m log m +x)* time. Since x ≤ L, the time is *O(m log m + L)*. Line 7 can be executed at most |G| times, thus the complexity of the algorithm is *O(|G| log |G| + |G| (m log m+L))*. The value of L is typically quite small. If *L* and *m* are in the same order, the complexity becomes *O(|G| log |G|+ |G| m log m)*.

Experimental Evaluation

The first subsection describes the setup of experiments. The second subsection reports the results of our privacy model. The third subsection reports the results of our model with column-wise constraints.

Setup

Machine: Experiments were run on a Dell PowerEdge Server with 3 GHz CPU and 2 GB memory, running Windows Server 2003.

Data: We used a patient data set obtained from the Kentucky Cancer Registry. It contains information about 72,194 patients. We used 4 quasi-identifier attributes: birth date, gender, race, and zip. We used 7 sensitive attributes: tumor topography, histology, survival years, diagnose date, age at diagnosis, tumor size, and tumor site. This is a real data set and the data distribution is skewed for many attributes. Our method was implemented in Perl.
Metrics: A successful privacy protection method protects privacy and introduces little data distortion. *K* and *L* indicate the degree of privacy protection. We use two metrics to measure

distortion. The first is *discernability* (Bayardo & Agrawal, 2005; LeFevre et al., 2006), which is the average group size of all records. Since each group needs to have at least K rows, the best possible value of discernability is K. Smaller discernability means less distortion.

The second metric for distortion is the average relative error for a large number of randomly generated structured query language (SQL) queries. Ideally, if we know the details of the subsequent data analysis, we can measure data distortion by its impact on all types of data analysis that can be performed. However, it is difficult to know beforehand all types of data analysis, thus we use random SQL queries. These queries return the number of patients satisfying several randomly generated conditions. The number of conditions was randomly selected from 1 to 4. At least one condition was on a randomly selected sensitive attribute. The other conditions were on randomly selected quasi-identifier attributes. The conditions were randomly generated equality or range conditions. 8800 different queries were tested. The error is computed to compare the counts over the sanitized data to the counts over the original data.

Algorithms: We compare the following three algorithms:

- Anonymize: this is our algorithm proposed in Figure 6.
- Anonymize-Column: this is our algorithm implementing column-wise constraints in Figure 8.
- *Exponential-L*: this algorithm implements the L-diversity model proposed by Machanavajjhala et al. (2006), which requires that each attribute has at least L different values for all records with the same values on the other attributes. This algorithm is the same as our method except the way it checks L-diversity.

Results for Our Model

Discernability results: There are three important parameters: K, L, and the number of sensitive attributes (m) included in the data set. We fixed two of them and varied the third. Figure 9 reports the discernability for various L values when K=50 and m=3 (the first three sensitive attributes are used). Figure 10 reports the discernability for various K values when L=10 and m=3. Figure 11 reports the discernability when various number of sensitive attributes are included and K=50 and L=10.

The discernability of Exponential-L is very high (meaning very high degree of data distortion). Exponential-L generates a single group that contains the whole data set for all cases except when there is only one sensitive attribute (in Figure 11). Exponential-L requires each attribute have at least L different values for all records with the same values on the other

Figure 9. Varying L, K=50, m=3

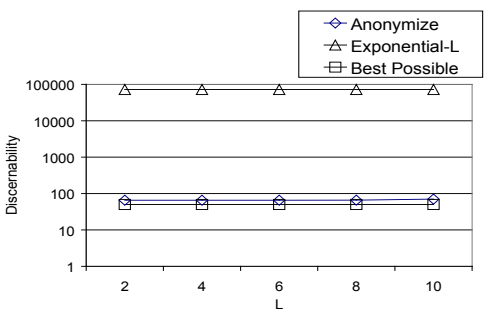


Figure 10. Varying K, L=10, m=3

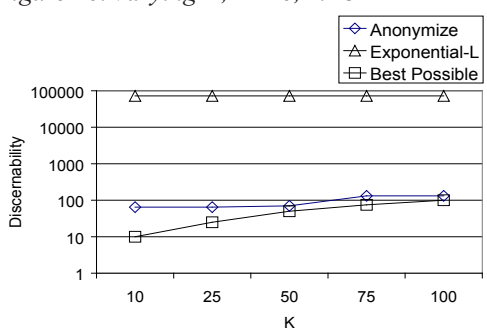


Figure 11. Vary number of sensitive attributes, K=50, L=10

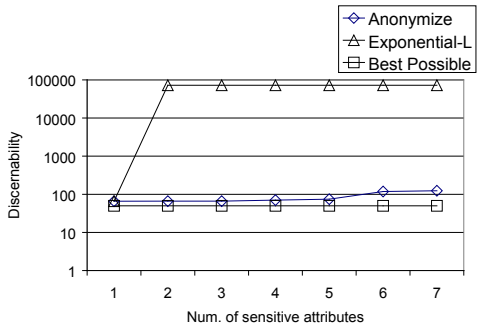
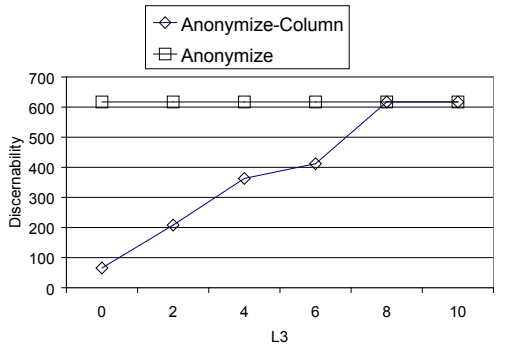


Figure 12. Vary L_3, $L=L_1=L_2=10$, $K=50$, $m = 3$



attributes. This leads to very large group sizes and large distortions to data.

The discernability of our method (Anonymize) is lower than that of Exponential-L by several orders of magnitude (note that the scale for y-axis is logarithmic), and is not much higher than the best possible discernability (which is *K)*. The discernability increases with *K* and *L*, because more data distortion is introduced by providing a higher degree of privacy. The increase of discernability is quite small when *L* increases because most groups cannot be split further due to violation of K-anonymity. Thus, the increase of *L* has a smaller impact on group size. The discernability of Anonymize also increases with the number of sensitive attributes. As the number of sensitive attributes increases, two rows are more likely to have the same values on a sensitive attribute, and thus larger groups are needed to achieve L-diversity. However, the discernability of our approach is still not much higher than the best possible case (less than a factor of two in most cases).

Error of random queries: Table 1 shows the average relative error for random queries when K=50, L=10, and m=3. The error of Anonymize algorithm for all queries is 11.3%. The error for queries returning less than 1% of the total number of patients is 14.6% and the error for queries returning 1% or more patients (there are 1117 such queries) is only 1.6%. The error for the first subset is higher because data distortion has a larger impact on these queries. For example, if a query returns just one patient and the distortion makes it return two, the error is 100%. The results show that our method introduces small error, especially when a medium to large number of records are returned. This property is suitable for data analysis because it is not very meaningful to study a very small fraction (e.g., less than 1%) of a data set. The relative error for Exponential-L is about twice of the error for Anonymize on all queries, and is about five times of the error for Anonymize on queries returning more than 1% of patients.

Table 1. Average relative error of random queries when K=50, L=10, and m=3

Algorithm	All queries	Queries returning < 1% of patients	Queries returning ≥ 1% of patients
Anonymize	11.3%	14.6%	1.6%
Exponential-L	23.2%	28.6%	8.5%

Figure 13. Execution time when varying number of sensitive attributes

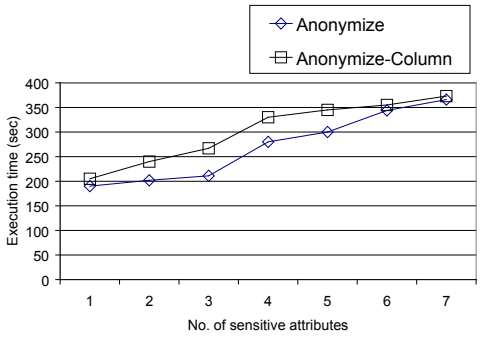
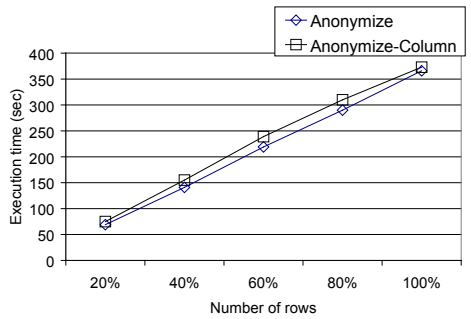


Figure 14. Execution time when varying number of rows (as fraction as total data set)



This is expected because Exponential-L generates very large groups and introduces larger data distortion. Since Exponential-L has very bad performance, we only show the results of Anonymize and Anonymize-Column in the remaining sections.

Results for L-Diversity with Column-wise Constraints

Figure 12 shows the results for L-diversity with column-wise constraints. We use three sensitive attributes: tumor topography, histology, and recur status. The attribute "recur status" has very few distinct values (31). Let L_1, L_2, and L_3 represent the column wise constraints for each attribute, we set $L = L_1 = L_2 = 10$, and varies L_3 from 0 to 10. Note that L_3=0 means none of the values of "recur status" attribute will be deleted and thus there is no L-diversity requirement on that attribute. $L_3 = 10$ means it has the same requirement as other attributes. The results show that the discernability increases for larger L_3 values, and becomes the same as the discernability of the case without column-wise constraints as $L_3 = 8$. This is expected because a smaller L_3 means fewer values are deleted from "recur status" attribute, making it easier to achieve L-diversity because this attribute has very few distinct values. The average relative error of random queries for $L_3 = 2$ is 10.4%. The average relative error for $L_3 = 10$ is 11.7%. Thus using column-wise constraints also leads to lower average relative error. The difference is not as big as the difference for discernability, because in both settings many groups are generated and thus the estimation of query results is quite accurate.

Execution Time

Figures 13 and 14 report the execution time of our algorithms when the number of rows and number of attributes are varied. We vary the number of sensitive attributes in the same way as in the previous subsection and use all rows in the data set. We vary the number of rows by selecting a fraction of rows in the data set, and use all seven sensitive attributes. $K=50$ and L=10 in all cases, and L_i of the last attribute is set to 2 for the Anonymize-Column. The results show that the execution time of both Anonymize and Anonymize-Column scale almost linearly with the number of rows and number of attributes. The execution time also increases at a slower pace when the number of sensitive attributes increase because it is more difficult to satisfy L-diversity for more sensitive attributes, and thus fewer groups are generated. Anonymize-Column also takes slightly more time than Anonymize because it calls the Anonymize algorithm first. However, the difference is not big because many groups also satisfy L-diversity without constraints, and Anonymize-Column does not need to do extra work for these groups.

CONCLUSION

This article proposes a privacy model that protects identity of patients for data with multiple sensitive attributes. A variant of this model is also proposed, which allows the user to specify a lower degree of diversity for attributes with very few distinct values. This article also proposes efficient algorithms to implement the model. Experiments show that the proposed approach introduces distortion orders of lower magnitude than the distortions introduced by the existing approach in the literature, and introduces small relative error for random SQL queries. As future work, we will study how to extend other formats of L-diversity to multiple sensitive attributes.

REFERENCES

Agrawal, D., & Aggarwal, C. C. (2001). *On the design and quantification of privacy preserving data mining algorithms.* Paper presented at the 20th ACM SIGMOD SIGACT-SIGART Symposium on Principles of Database Systems, Santa Barbara, CA.

Agrawal, R., & Srikant, R. (May 2000). *Privacy preserving data mining.* Paper presented at the 2000 ACM SIGMOD Conference on Management of Data, Dallas, TX.

Bayardo, R. J., & Agrawal, R. (2005). *Data privacy through optimal k-anonymization.* Paper presented at the IEEE International Conference on Data Engineering.

Evfimevski, A., Gehrke, J., & Srikant, R. (2003, June). *Limiting privacy breaches in privacy preserving data mining.* Paper presented at the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, San Diego, CA.

Evfimevski, A., Srikant, R., Agrawal, R., & Gehrke, J. (2002, July). *Privacy preserving mining of association rules.* Paper presented at the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), Edmonton, Alberta, Canada.

Federal Trade Commission. (2007). *Identity Theft Resource Center: Facts and statistics.* Author.

Gartner Inc. (2007). *Gartner says number of identity theft victims has increased more than 50 percent since 2003.* Author.

Hintoglu, A. A., Inan, A., Saygin, Y., & Keskinoz, M. (2005). *Suppressing data sets to prevent discovery of association rules.* Paper presented at the IEEE International Conference on Data Mining.

Kargupta, H., Datta, S., Wang, Q., & Sivakumar, K. (2003). *On the privacy preserving properties of random data perturbation techniques.* Paper presented at the IEEE International Conference on Data Mining.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2005). *Incognito: Efficient full-domain K-anonymity.* Paper presented at the ACM SIGMOD Conference on Management of Data.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2006). *Mondrian multidimensional k-anonymity.* Paper presented at the IEEE International Conference on Data Engineering.

Li, X.-B., & Sarkar, S. (2006). A tree-based data perturbation approach for privacy-preserving data mining. *IEEE Transactions on Knowledge and Data Engineering, 18*(9), 1278--1283.

MacVittie, D. (2007, August 31). Javelin 2006 identity fraud report. *Network Computing*.

Machanavajjhala, A., Gehrke, J., Kifer, D., & Venkitasubramaniam, M. (2006, April). *L-diversity: Privacy beyond k-anonymity.* Paper presented at the 22nd IEEE International Conference on Data Engineering (ICDE 2006), Atlanta, Georgia.

Menon, S., & Sarkar, S. (2006). *Exploiting problem structure to efficiently sanitize very large transactional databases.* Paper presented at the WITS.

Menon, S., Sarkar, S., & Mukherjee, S. (September 2005). Maximizing accuracy of shared databases when concealing sensitive patterns. *Information Systems Research, 16*(3), 256–270.

Oliveira, S., & Zaiane, O. R. (2002). *Privacy preserving frequent itemset mining.* Paper presented at the IEEE International Conference on Privacy, Security and Data Mining, Maebashi City, Japan.

Samarati, P. (2001). Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering, 13*(6), 1010–1027.

Saygin, Y., Verykios, V. S., & Clifton, C. (2001). Using unknowns to prevent discovery of association rules. *SIGMOD Record, 30*(4), 45–54.

Sweeney, L. (2002a). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10*(5), 571–588.

Sweeney, L. (2002b). K-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10*(5), 557–570.

Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., & Theodoridis, Y. (2004, March). State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record, 33*(1), 50–57.

Verykios, V. S., Elmagarmid, A. K., Elisa, B., Saygin, Y., & Elena, D. (2004, April). Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering, 16*(4), 434–447.

Xiao, X., & Tao, Y. (2006). *Anatomy: Simple and effective privacy preservation.* Paper presented at the International Conference on Very Large Data Bases.

Zhu, Y., & Liu, L. (2004). *Optimal randomization for privacy preserving data mining.* Paper presented at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Tamas S. Gal is a PhD. student at the Department of Information Systems at the University of Maryland, Baltimore County. He holds a masters degree from the same department. He works for the Kentucky Cancer Registry as a database analyst. His research specialization is in the areas of privacy preserving data mining in the medical field, healthcare information systems and bioinformatics.

Zhiyuan Chen is an assistant professor at information systems department, UMBC. He has a PhD in computer science from Cornell University. His research interests include privacy preserving data mining, data navigation and visualization, XML, automatic database tuning and database compression.

Aryya Gangopadhyay is an associate professor of information systems at the University of Maryland, Baltimore County (UMBC). He has a PhD in computer information systems from Rutgers University. His research interests include privacy preserving data mining, OLAP data cube navigation and core and applied research on data mining. He has co-authored and edited three books, many book chapters and numerous papers in peer-reviewed journals.