Technical Report

TR-2008-013

Privacy-Preserving Data Publishing for Horizontally Partitioned Databases

by

Pawel Jurczyk, Li Xiong

Mathematics and Computer Science

EMORY UNIVERSITY

Privacy-Preserving Data Publishing for Horizontally Partitioned Databases

Pawel Jurczyk
Department of Math&CS
Emory University
pjurczy@emory.edu

Li Xiong
Department of Math&CS
Emory University
lxiong@emory.edu

July 24, 2008

Abstract

There is an increasing need for sharing data repositories containing personal information across multiple distributed, possibly untrusted, and private databases. However, such data sharing is subject to constraints imposed by privacy of individuals or data subjects as well as data confidentiality of institutions or data providers. Concretely, given a query spanning multiple databases, query results should not contain individually identifiable information. In addition, institutions should not reveal their databases to each other apart from the query results. In this paper, we develop a set of decentralized protocols that enable data sharing for horizontally partitioned databases given these constraints. Our approach includes a distributed anonymization protocol that allows independent data providers to build a virtual anonymized database, and a distributed querying protocol that allows clients to query the virtual database. Together they ensure that the query results satisfy k-anonymity requirements and information disclosure between institutional databases is minimal. We present a set of formal analysis as well as experiments to evaluate the protocols in terms of their correctness, efficiency and privacy characteristics.

1 Introduction

Current information technology enables many organizations to collect, store, and use various types of information about individuals in large repositories. Government and organizations increasingly recognize the critical value and opportunities in sharing such a wealth of information across multiple distributed, private, and possibly untrusted databases. An example is the Shared Pathology Informatics Network (SPIN)[1] initiative by the National Cancer Institute that attempts to provide a system offering search interfaces for existing electronic databases at institutions across the country to locate human specimens and associated clinical and pathologic data needed for cancer research. The goal of the system is to enable investigators to query archived information in multiple institutions transparently. However, personal health information is protected under the Health Insurance Portability and Accountability Act (HIPAA)[2][3] and cannot be revealed without de-identification or anonymization. In addition, institutions may not want to reveal their private databases to each other for various reasons.

These scenarios can be generalized into the problem of privacy preserving data publishing for multiple distributed databases where multiple *data custodians* need to publish an anonymized and integrated view of the data that does not contain individually identifiable information. Such data sharing is subject to two constraints. The first constraint is the privacy of the individuals or in general the data subject (such as the patients). The second is the data

[1] Shared Pathology Informatics Network. http://www.cancerdiagnosis.nci.nih.gov/spin/

[2] Health Insurance Portability and Accountability Act (HIPAA). http://www.hhs.gov/ocr/hipaa/.

[3] State law or institutional policy may differ from the HIPAA standard and should be considered as well.

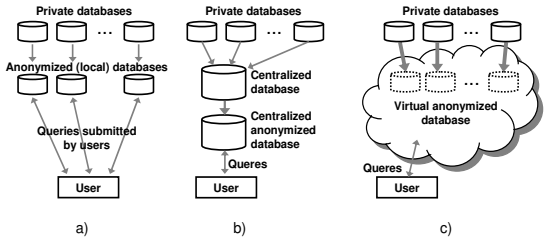

Figure 1: Architectures for privacy preserving data publishing

confidentiality of the data custodians (such as the institutions). Given a query spanning multiple databases, query results should not contain individually identifiable information. In addition, institutions should not reveal their databases to each other apart from the query results.

Existing and potential solutions. Privacy preserving data publishing for a single database has been extensively studied in recent years. A large body of work contributes to data anonymization that transforms a dataset to meet a privacy principle such as k-anonymity using techniques such as generalization, suppression (removal), permutation and swapping of certain data values so that it does not contain individually identifiable information [12, 37, 27, 5, 1, 9, 7, 45, 19, 20, 21, 36, 18, 38, 44].

There are a number of potential approaches one may apply to enable privacy preserving data publishing for distributed databases. A naive approach is for each data custodian to perform data anonymization independently as shown in Fig. 1a. Data recipients or clients can then query the individual anonymized databases or an integrated view of them. One main drawback of this approach is that data is anonymized before the integration and hence will cause the data utility to suffer. In addition, individual databases reveal their ownership of the anonymized data.

An alternative approach assumes an existence of third party that can be trusted by each of the data owners as shown in Fig. 1b. In this scenario, data owners send their data to this third party where data integration and anonymization are performed. Then, clients can query the centralized database. However, finding such a trusted third party is not always feasible. Compromise of the server by hackers could lead to a complete privacy loss for all participating parties.

In this paper, we propose a distributed data anonymization approach as illustrated in Fig. 1c. In this approach, data owners participate in distributed protocols to produce a *virtual* integrated and anonymized database which can be then queried by clients. Important to note is that the anonymized data still resides at individual databases and the integration and anonymization of the data is performed through the distributed protocols.

Contributions. We study the problem of privacy-preserving data publishing for horizontally partitioned databases and present the distributed anonymization approach for the problem. Our approach consists of two main contributions.

First, it includes two protocols, namely, a *distributed k-anonymization protocol* that allows multiple data providers with horizontally partitioned databases to build a virtual k-anonymized database based on the integration (or union) of the data, and a *distributed querying protocol* that allows clients to query the virtual database. As the output of the distributed anonymization protocol, each database produces a local anonymized dataset and their union forms a virtual database that is guaranteed to be k-anonymous. When users query the virtual database, each individual database executes the query on its local anonymized dataset, and then engage in the distributed querying protocol to assemble the results that are guaranteed to be k-anonymous. Both protocols utilize multi-party protocols for sub-operations such that information disclosure between individual databases is minimal.

Second, in addition to utilizing a set of existing multi-party computation protocols for primitive operations to build the above protocols, we also propose a novel probabilistic set union protocol that computes union of multiple data inputs in a secure manner as part of the distributed querying protocol. It can be used as a building block for various distributed privacy-preserving data mining and analysis tasks.

Finally, We present a set of formal analysis as well as experiments evaluating the protocols in terms of their correctness, complexity and privacy characteristics.

Organization. The remainder of this paper is organized as follows. Section 2 briefly reviews work related to our research. Section 3 discusses the privacy model we are using. Section 4 presents our distributed anonymization approach including the distributed anonymization protocol and distributed querying protocol. Section 5 presents the general set union protocol that is used in the distributed

querying protocol with a detailed analysis. Section 6 presents a set of experimental evaluations of our approach and Section 7 concludes the paper.

2 Related work

Our work is inspired and informed by a number of areas. We briefly review the closely related areas below.

Privacy preserving data publishing. Privacy preserving data publishing for centralized databases has been studied extensively. One thread of work aims at devising privacy principles, such as k-anonymity, l-diversity, t-closeness, and m-variance, that serve as criteria for judging whether a published dataset provides sufficient privacy protection [29, 25, 31, 2, 23, 39, 26, 28]. Another large body of work contributes to algorithms that transforms a dataset to meet one of the above privacy principles (dominantly k-anonymity) [12, 37, 27, 5, 1, 9, 7, 45, 19, 20, 21, 36, 18, 38, 44]. Our distributed anonymization protocol is built on top of the k-anonymity principle and the greedy top-down Mondrian multidimensional k-anonymization algorithm [20].

There are some works focused on data anonymization of distributed databases. [13] presented a two-party framework along with an application that generates k-anonymous data from two vertically partitioned sources without disclosing data from one site to the other. [46] proposed provably private solutions for k-anonymization in the distributed scenario by maintaining end-to-end privacy from the original customer data to the final k-anonymous results. In contrast, our work is aimed at horizontal data distribution and arbitrary number of sites. In addition, our protocols are not based on heavy cryptographic primitives. We exploit inherent anonymity of large number of participating sites and utilize probabilistic methods to achieve minimal information disclosure and minimal overhead.

Secure multi-party computation. Our approach also has its roots in the secure multi-party computation (MPC) problem [11, 10]. In MPC, a given number of participants, each having a private data, wants to compute the value of a public function. A MPC protocol is secure if no participant can learn more from the description of the public function and the result of function. While there are general secure MPC protocols, they require substantial computation and communication costs and are impractical for multi-party large database problems. By exploiting the inherent anonymity of multiple parties and probabilistic methods, our approach provides a desired tradeoff of absolute security for efficiency compared to traditional MPC approaches.

Distributed privacy preserving data mining. Another related area is distributed privacy preserving data sharing and mining that deals with specific mining tasks across multiple distributed data sources [24, 32, 15, 17, 34, 43, 33, 3, 4, 35, 16, 40, 8, 41]. The main goal is to ensure that data is not disclosed among participating parties. Two main approaches are data perturbation and specialized distributed protocols. Our research utilizes some of the primitive distributed protocols proposed in these works. It also proposes a novel and general set union protocol that can be used for various distributed privacy preserving data mining tasks.

3 Privacy Model

In this section we present the privacy goals that we focus on in the paper, followed by privacy models for characterizing how these privacy goals are achieved. As we identified in Section 1, given a query spanning multiple databases, our privacy goals are two fold. First, the query results should not contain individually identifiable information. Second, individual databases should not reveal their data to each other apart from the query results. We describe the models we use for each of these two goals.

Individual identifiability. Among the many privacy principles based on individual identifiability, k-anonymity [30] is the most widely accepted and serves as the basis for many others, and hence, will be used in our current approach. The general approach and protocol structure, however, is orthogonal to these privacy principles and it is on our research agenda to incorporate more advanced privacy principles into our protocol set.

In defining anonymization, attributes of a given relational table T, are characterized into three types. *Unique identifiers* are attributes that identify individuals. Known identifiers are typically removed entirely from released micro-data. *Quasi-identifier set* is a minimal set of attributes (X_1, ..., X_d) that can be joined with external information to re-identify individual records. *Sensitive at-*

tributes are those attributes that an adversary should not be permitted to uniquely associate their values with a unique identifier.

The k-anonymity model provides an intuitive requirement for privacy in stipulating that no individual record should be uniquely identifiable from a group of k with respect to the quasi-identifier set. The set of all tuples in T containing identical values for the quasi-identifier set X_1, ..., X_d is referred to as an *Equivalence Class*. T is *k-anonymous* with respect to X_1, ..., X_d if every tuple is in an equivalence class of size at least k. A *k-anonymization* of T is a transformation or generalization of the data T such that the transformation is k-anonymous.

Data confidentiality. Our second privacy goal follows the goal of secure MPC but is less stringent. Instead of attempting to guarantee absolute security in which individual databases reveal nothing about their data to each other apart from the query results, we wish to minimize data exposure among the multiple parties. We also adopt the *semi-honest* model commonly used in secure MPC problems. A semi-honest party follows the rules of the protocol, but it can attempt to learn additional information about other nodes by analyzing the data received during the execution of the protocol.

4 Distributed Anonymization

In this section we describe our distributed anonymization approach. We first describe the general protocol structure, then present the distributed anonymization protocol, followed by the distributed querying protocol.

We assume that the data are split horizontally among n sites ($n > 2$) and each site owns a private database d_i. In addition, the *quasi-identifier* of each local database is uniform among all the sites. The sites engage in a distributed anonymization protocol where each site produces a local anonymized dataset a_i and their union forms a virtual database that is guaranteed to be k-anonymous. Note that a_i is not required to be k-anonymous by itself. When users query the virtual database, each individual database executes the query on a_i and then engage in a distributed querying protocol to assemble the results that are guaranteed to be k-anonymous.

4.1 Protocol Structure

The proposed protocols are designed to run over a decentralized network. The protocol structure is presented in Figure 2. Nodes are mapped to a ring topology randomly. We assume that each node knows its predecessor and successor. Each node has a local computation module that executes its part of the protocol independently and passes the computation result along the ring.

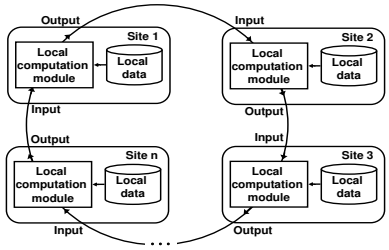

Figure 2: Protocol Structure

4.2 Distributed Anonymization Protocol

Our distributed algorithm for anonymization is based on the Mondrian algorithm that uses greedy recursive partitioning of the (multidimensional) quasi-identifer domain space. It recursively chooses the split attribute with the largest normalized range of values, and (for continuous or ordinal attributes) partitions the data around the median value of the split attribute. This process is repeated until no allowable split remains, meaning that a particular region cannot be further divided without violating the anonymity constraint, or constraints imposed by value generalization hierarchies.

The key idea for the distributed anonymization protocol is to use a set of secure atomic multi-party protocols to realize the Mondrain method for the distributed setting. We present the main protocol first, followed by a set of atomic protocols that are used in the main protocol.

We assume a leading site is selected for the protocol. The main protocol for the leading and other sites are presented in Algorithm 1 and 2 respectively. The steps performed at the leading site are similar to the centralized Mondrian method. It chooses the best attribute (with largest spread) for split. Next, it performs the split and

Algorithm 1 Distributed anonymization algorithm - leading site ($i = 0$)

function split(set c)
Compute range of values for each attribute from quasi-identifier in set $d = \bigcup d_i$ (using secure min/max protocol)
Choose best split attribute a with largest range of values
Find median value m of a in set $d = \bigcup d_i$ (using secure median protocol)
Send a and m to node 1
Split set d_0, create two sets, s_0 containing items $smaller\ than\ m$ and g_0 containing items $greater\ than\ m$. Distribute median items among s_i and g_i.
Find $size_{left} = |\bigcup s_i|$ and $size_{right} = |\bigcup g_i|$ (using secure sum protocol)
if $size_{left} > 2 * K$ **then**
    Send $split\_left = true$ to node 1
    call split(s_0)
else
    Send $split\_left = false$ to node 1
end if
if $size_{right} > 2 * K$ **then**
    Send $split\_right = true$ to node 1
    call split(g_0)
else
    Send $split\_right = false$ to node 1
end if
end function split

Algorithm 2 Distributed anonymization algorithm - non-leading node ($i > 0$)

function split(set c)
Read split attribute a and median value m from node $(i-1)$ and pass them to node $i+1$
Split set c into s_i containing items $smaller\ than\ m$ and g_i containing items $greater\ than\ m$. Distribute median items among s_i and g_i.
Read $split\_left$ from node $i-1$ and pass it to node $i+1$
if split_left **then**
    call split(s_i)
end if
Read $split\_right$ from node $i-1$
Send $split\_right$ to node $i+1$
if split_right **then**
    call split(g_i)
end if
end function split

recursively checks whether further split of new subsets is possible. When selecting the best split attribute, the leading site needs to have the knowledge of the ranges of values of each attribute with respect to data items located at all sites. So a secure min/max protocol is used to compute the minimum and maximum value of each attribute across the databases. The leading site can then compute the range of each attribute and select the best attribute with largest range. When finding the split point for partitioning, a secure median protocol is used to find the median value of the attribute with respect to the data across the databases. Finally, when determining whether a partition can be further split, a secure sum protocol is used to count the total number of tuples of the partition across the databases. We introduce each of the atomic protocols being used below.

Secure sum protocol. The secure sum protocol is a simple example of secure multi-party computation. The protocol proceeds as follows. Assume that the sum value is known to lie in the range [0..m]. First node generates a random number r and passes to the second node value $v_1 = (x_1 + r)\ mod\ m$. The second node received value v_1 from the first node, computes $v_2 = (v_1 + x_2)\ mod\ m$ and passes v_2 to the third node and so on. The last node sends value v_n to the first node. Then, the first node can subtract r from the value v_n it received and the sum is known.

Secure min/max protocol. We adopted the multi-round probabilistic protocols for computing minimum and maximum presented in [40] and we briefly describe them below for completeness. In these protocols each node runs a randomized algorithm with a given randomization probability associated with each round and the final result is produced in a bounded number of rounds. The randomization probability at round r, $Pr(r)$, is defined with an initial probability, p_0, and a dampening factor, d, and decreases in each round. At round r, node i receives a global value g_{i-1} from node $i-1$, performs the local randomized algorithm, and passes the output g_i to node $i+1$. The local algorithm is presented in Algorithm 3.

Secure median protocol. The secure median protocol is built on top of the secure sum and secure min/max protocols and uses the idea of binary search. First, the secure min/max protocol is used to find the minimum and maximum values of the attribute. A leading site initializes

Algorithm 3 Local algorithm for secure min(max) algorithm (executed at round r by node i)

1: INPUT: g_{i-1}, v_i, OUTPUT: $g_i(r)$
2: $P_r(r) \leftarrow p_0 * d^{r-1}$
3: **if** $g_{i-1}(r) \geq v_i$ **then**
4: $g_i(r) \leftarrow g_{i-1}(r)$
5: **else**
6: with probability P_r: $gi(r)$ $\leftarrow$a random value between $[g_{i-1}(r), v_i)$, with probability $1 - Pr$: $g_i(r) \leftarrow v_i$
7: **end if**

the median value as: $med = (min + max)/2$. Next, the secure sum protocol is used to count the numbers of data items less than and greater than the current median value respectively. If the numbers are equal, the median is found. If the number of items smaller than current median value is greater than number of items greater than the current median value, a new median value from the lower-half of the data is computed: $med_1 = (min + med)/2$. Otherwise, a new median value from the upper-half of the data is computed: $med_1 = (med + max)/2$. The protocol continues until the two numbers are equal, in other words, the median value splits the data in halves. A sketch of the median protocol is presented in Algorithm 4.

Algorithm 4 Secure median algorithm

1: $min, max \leftarrow$ Find minimum and maximum values of the attribute (using secure min/max protocol)
2: **repeat**
3: $median \leftarrow (min + max)/2$
4: s, $g \leftarrow$ Find number of values smaller than and greater than $median$ (using secure sum protocol)
5: **if** $s \geq g$ **then**
6: $max \leftarrow median$
7: **else if** $s \leq g$ **then**
8: $min \leftarrow median$
9: **end if**
10: **until** $s = g$

4.3 Complexity and Overhead

Having presented the distributed anonymization protocol, we briefly discuss the complexity and overhead of the protocol in this subsection. The time complexity of the original Mondrain algorithm is $O(nlogn)$ where n is the number of items in the anonymized dataset [22]. Our protocol, however, requires additional overhead due to the fact that the nodes have to use additional protocols in each step of computation. As we presented in Algorithm 1, each iteration of the distributed anonymization algorithm requires finding minimum/maximum values of attributes, median value of an attribute, and the count of tuples of a partition. The complexity of the min/max and sum protocols does not depend on the number of items in dataset. The median protocol, however, works like a binary search and therefore requires $O(logn)$ steps. As a consequence, the time complexity of our protocol can be estimated as $O(nlog^2n)$.

The communication cost of the protocol is determined by two factors. The first is the cost for a single round. This depends on the number of nodes involved in the system and the topology which is used and in our case it is proportional to the number of nodes on the ring. As the future work, we are considering alternative topologies (such as trees) in order to optimize the communication cost for each round. The second factor is the number of rounds and this is determined by all the atomic protocols used by each iteration of the anonymization protocol. For example, in the case of min/max protocol, the communication overhead depends on the stringency of the privacy requirement (which has an impact on number of rounds).

4.4 Distributed Querying Protocol

The distributed anonymization protocol enables set of nodes to produce a virtual k-anonymous database based on the union of the data horizontally split among the nodes. At the end of the protocol, the local anonymized datasets are not necessary k-anonymized. However, the union of datasets forms the virtual database and is guaranteed to satisfy k-anonymity requirement. Our approach also includes a distributed querying protocol that allows users to query this virtual database. When a query is received, each database runs the query against its local randomized dataset and the results are then unioned and included in the response. As we require that ownership of items is not revealed during the querying phase, we propose a novel and efficient secure set union protocol for this purpose. We present the set union protocol along with its analysis in the next section.

5 Secure set union protocol

In this section, we present the set union protocol used in the distributed querying protocol. It is also a general protocol that can be used as a building block for various distributed privacy preserving data mining and sharing tasks.

Given a set of nodes with private data items, the problem is to compute the union of data items while minimizing data disclosure of the nodes to each other besides the final result. Our privacy goal is to prevent an adversary from being able to determine the owner of items from the final result. Formally, given *n* sites, and each site holding a local set of data tuples or items x_i, we wish to compute $X = \bigcup x_i$ while minimizing the probability of a node revealing x_i to other nodes. Given our usage for the distributed querying protocol, we preserve duplicate items, in other words, if the same data item appears more than once in local subsets, all the same items will appear in the final result. However, the protocol can be easily modified to remove duplicates if necessary.

5.1 Protocol

The protocol follows the same protocol structure as we described in Section 4.1. To facilitate the discussion of our protocol, we first present a naive protocol as the intuitive motivation and then describe the rational and the algorithmic details of our probabilistic protocol.

Naive approach. A naive way to compute the union of data items from distributed nodes without a central server is to have the nodes pass their data items along a ring. The first node sends its items to the second node. The second node adds its own items to the intermediate result from the first node and sends it to the third node and so on. The union is found at the end of the round.

Clearly, the protocol does not offer good data privacy. First, the starting node has *provable exposure* to its successor regarding its data items. Second, the nodes that are close to the starting node in the ring have a fairly high probability disclosing their data items.

Secure set union protocol. Now we present the *secure set union protocol* utilizing randomization to minimize data exposure. There are two key ideas to the protocol. The first idea is to introduce random items by the starting node so that it will not suffer from provable exposure. The second idea is to randomly select a starting node so that nodes close to the starting node on the ring will not suffer from a high probability of data disclosure.

Algorithm 5 Secure set union protocol.

```
1: INPUT: x_i: local subset contributing to union
2: Choose random t_i
3: t_max ← max(t_1..t_n) (using secure max protocol)
4: if t_i = t_max then
5:     Generate set of random items r
6:     Send r ∪ x_i to successor
7:     Receive X from predecessor
8:     Result ← (X − r)
9: else
10:    Receive X from predecessor
11:    Send X ∪ x_i to successor
12: end if
```

The protocol works as follows. In the initialization or leader selection round, each node generates a random number t_i from predefined range. The node with highest value of t is elected to be the leader node. The secure max protocol can be used to find the highest value of t. In the main protocol round, the leader node i generates a random set r and adds its local subset x_i to this random set. Then it passes its intermediate result to the node $i+1$. Starting from this point, each node *j* adds its local subset x_j to the intermediate result and passes the result to node $j+1$. When node *i* receives the result from its predecessor, the set union can be found by removing random items r from this set. A sketch of the algorithm is presented in Algorithm 5 and Figure 3 presents the main round of the protocol.

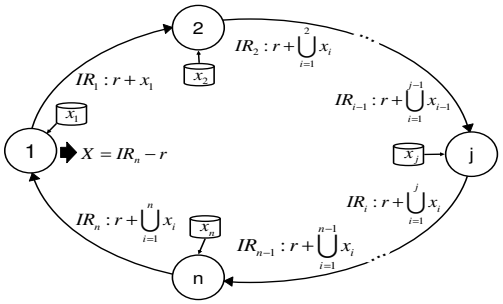

Figure 3: Main round of the secure set union protocol (assuming that node 1 chooses the highest t)

5.2 Analysis

In this subsection we analyze the set union protocol presented above. The communication overhead of the protocol is not significant. In the leader election phase, the amount of data that is transferred is constant and does not depend on the result size. In the main round of the protocol, the overhead includes passing items contributing to the final set and randomly generated items.

For the rest of the subsection, we focus our analysis on the privacy characteristics of the protocol. We first introduce the privacy metric that we use for evaluating how well we achieve our privacy goal and present a formal analysis using the metric. We will plot the analytical bounds derived in this section along with our experimental results in next section.

Privacy metric. Our privacy goal is to prevent an adversary from being able to determine the items of a node or the owner of items from the final result. Given the privacy goal, we need to quantify the degree of data exposure for each node. We adopt the privacy metric, loss of privacy (*LoP*) [42], for this purpose. Let *R* denote the final result of the algorithm and *IR* denote an intermediate result during execution of the protocol. Suppose an adversary is able to make a claim C about the data at a node, we define two probabilities. The first, $P(C|IR, R)$, is the probability of claim *C* being true when node has both *IR* and *R*. The second, $P(C|R)$, is the probability of claim *C* being true when node has only the final result *R*. The loss of privacy is defined as follows:

$$LoP = P(C|IR, R) - P(C|R) \tag{1}$$

The measure of LoP gives us a measure of the additional information about the ownership of the data an adversary may obtain given the knowledge of the intermediate result besides the final query result.

In our case, there are two kinds of data exposure with corresponding claims that can be made by an adversary, namely, *set exposure* and *item exposure*. For set exposure, an adversary is able to make a claim about the whole set of items a node contributes to the final union result (C = *node i contributed subset a_i to the final result*). For item exposure, an adversary is able to make a claim about a particular item a node contributes to the final result (e.g. C = *node i contributed item v_i to the final result*). Below we analyze the loss of privacy for these two cases respectively.

Set exposure. We first analyze loss of privacy for a node in terms of set exposure. As the worst case scenario, we consider the set exposure for the starting node (we assume node 1 is the starting node) with the second node (node 2) as the adversary. To begin with, we also assume that node 2 is aware that node 1 is the starting node. This is the worst case because node 2 only has to identify a set of real data items (not randomly generated items) from the intermediate result it receives from node 1 while any further adversary nodes will have to not only identify the real items but also the owner of items from the nodes ahead of it on the ring.

Suppose node 2 is trying to identify the whole set of items contributed by node 1 by making a claim C: $x_1 = a_1$. Recall that the loss of privacy is defined as the relative probability of the claim being true with and without the intermediate result. We compute $P(C|R)$ and $P(C|IR, R)$ below and derive the LoP.

We start by computing $P(C|R)$, the probability of the claim being true given only the final result X. Given only X, the best an adversary, node 2, can do for guessing x_1 is to select a random number of items from X which are not among his own items x_2. If we assume that X contains c distinct items, the probability the claim is true is given by:

$$P(C|X) = \frac{1}{\sum_{a=0}^{c} \binom{|X - x_2|}{a}} \approx 0 \tag{2}$$

It is important to note that we are limiting the analysis above to the case when result set contains only distinct items. As having duplicates helps an adversary, we are actually finding a lower bound for the probability $P(C|R)$ (and an upper bound for the *LoP*).

We now compute $P(C|IR, R)$, the probability of the claim being true given intermediate result IR_1 and the final result X. The intermediate result IR_1 contains the random set r generated by node 1 and the subset x_1 contributed by node 1. If it happens that no item from the random set r appears in the final result set, node 2 can determine r using $r = IR_1 - X$ and consequently determine x_1 using $x_1 = IR_1 - r$. Thus the best the adversary can do for guessing x_1 is to make a claim C:

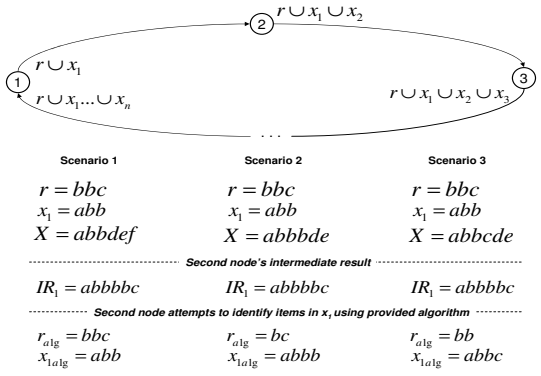

Figure 4: Example of Data Exposure of Node 1 to Node 2

$x_1 = IR_1 - (IR_1 - X)$. Figure 4 presents a few possible scenarios for the claim. The probability of this claim being true can be derived as follows:

$$\begin{aligned} P(C|X, IR_1) &= P(x_1 = r \cup x_1 - (r \cup x_1 - X)) \\ &= P(x_1 = r \cup x_1 - (r - (X - x_1))) \\ &= P(r \cap (X - x_1) = \emptyset) \end{aligned}$$

If we assume that the domain D of possible items contains m distinct items and if we again assume that the result of union algorithm contains c distinct items and that node 1 contributes c_1 distinct items to the final set, the probability that a random set r does not contain any item from the set $X - x_1$ is given by:

$$P(r \cap (X - x_1) = \emptyset) = \left(\frac{m - c + c_1}{m}\right)^{|r|} \quad (3)$$

Now we can derive the loss of privacy for the starting node (given the knowledge of the starting node by the adversary) as:

$$LoP = \left(\frac{m - c + c_1}{m}\right)^{|r|} - 0 = \left(\frac{m - c + c_1}{m}\right)^{|r|} \quad (4)$$

The above analysis assumed that node 2, the adversary node, is aware that node 1 is the starting node. As our protocol utilizes a randomized starting scheme, the probability of a node being the starting node is $\frac{1}{n-1}$ (assuming the adversary node is not the starting node). Thus we derive the bound of loss of privacy for our protocol as follows:

$$LoP = \frac{1}{n-1} * \left(\frac{m - c + c_1}{m}\right)^{|r|} \quad (5)$$

Item exposure. We now analyze loss of privacy for a node in terms of item exposure. We again consider the starting node (node 1) with the second node (node 2) as the adversary. Suppose node 2 is trying to identify a single item contributed by node 1, x_1, by making a claim C: $v_1 \in x_1$.

We first compute $P(C|R)$, the probability of the claim being true given only the final result X. Given only X, the best an adversary can do for guessing a single item in x_1 is to select an item from X. The probability this claim being true is: $P(C|R) = \frac{1}{n-1}$.

We now compute $P(C|IR, R)$, the probability of the claim being true given the intermediate result IR_1 and the final result X, and analyze how the intermediate result can help this node to find the owner of some items. Using a similar approach as in the set exposure scenario, the adversary can try to find items contributed by its predecessor. Simply put, the less items from random set r are in the algorithm's final result, the easier the second node can identify items contributed by the first node. Specifically, observing again scenarios presented in Figure 4, and assuming that node 2 is aware of node 1 being the leader, the probability $P(C|IR, R)$ is given by the following equation (we assume that each node contributes in average $\frac{c}{n}$ items):

$$\begin{aligned} P(C|IR, R) &= \frac{|x_1| + |x_1 \cap (r \cap (x - x_2))|}{|x_2| + |(r \cap (x - x_2))|} \leq \\ & \frac{2 * \frac{c}{n}}{\frac{c}{n} + |r| * \frac{c - \frac{c}{n}}{m}} \end{aligned} \quad (6)$$

The worse case LoP for the starting node can be then derived as follows:

$$LoP \leq \frac{2 * \frac{c}{n}}{\frac{c}{n} + |r| * \frac{c - \frac{c}{n}}{m}} - \frac{1}{n-1} \quad (7)$$

Considering the randomized starting scheme, the expected LoP for the item exposure is as follows:

$$\begin{aligned} LoP &= max\left(\frac{1}{n-1} * \frac{|x_1| + |x_1 \cap (r \cap (x - x_2))|}{|x_1| + |(r \cap (x - x_2))|}, \frac{1}{n-1}\right) \\ & - \frac{1}{n-1} = 0 \end{aligned} \quad (8)$$

Generating random items. One open issue remaining is how many random items a node should generate in the first phase. Now we will attempt to devise rules which

can guarantee certain LoP bounds. Our analysis is again divided into two cases, namely, set exposure and item exposure.

Set exposure. Given Equation 4 that gives the *LoP* bound for set exposure, we can obtain the following:

$$|r| = \lceil \log_{\frac{m-c+c_1}{m}} (n-1) * LoP_{expected} \rceil \quad (9)$$

Equation 9 allows one to find a minimal number of random items that should be generated in order to guarantee a given LoP bound for set exposure.

Item exposure. The *LoP* metric for item exposure was presented in equation 8. This equation shows that in average the algorithm does not reveal any additional information. Despite this fact, however, one has to deal with worst case *LoP* presented in equation 7 (the case when first node in the ring chooses the highest value of random value t). We are again interested in finding an upper bound for the LoP value. In this case, from Equation 7 we can derive the following:

$$|r| = \lceil \frac{m * \frac{c}{n}}{c - \frac{c}{n}} * \left(\frac{2}{LoP_{expected} + \frac{1}{n-1}} - 1 \right) \rceil \quad (10)$$

Equation 10 allows one to find a minimal number of random items that should be generated in order to guarantee a given LoP bound for item exposure.

6 Experimental evaluation

We have implemented the protocols in Java within the DObjects framework [14] which provides a platform for querying data across distributed and heterogeneous data sources. In this section we will present a set of experimental evaluations of the proposed protocols with respect to the two privacy constraints: individual identifiability and data confidentiality. The questions we attempt to answer are: 1) What individual privacy guarantees does the distributed anonymization approach offer and how does it compare to the naive and centralized approach? 2) What level of data confidentiality does the distributed anonymization approach offer? 3) How does the general secure set union protocol perform in various settings and how does it compare with the analytical results?

6.1 Individual Identifiability

We first present an evaluation of the distributed anonymization protocol we have proposed in section 4 in terms of the level of privacy and quality of the anonymized data.

Metrics. Based on the k-anonymity principle, the larger the k, the higher the level of individual privacy or anonymity. On the other hand, to evaluate the quality or utility of the anonymized data with different privacy levels, we use the discernability metric proposed by Agrawal in [6]. The metric assigns a penalty to each tuple based on how many tuples in the anonymized dataset are indistinguishable from it. In other words, the metric assigns to each tuple *t* a penalty, which is determined by the size of equivalence class containing t:

$$C_{DM} = \sum_{\forall t} E(t)^2 \quad (11)$$

In the equation above $E(t)$ is the size of the equivalence class to which given tuple t belongs. As a second metric for evaluating the anonymization quality, we also use the average size of equivalence classes generated by the algorithm.

Results. We used the Adults dataset from UC Irvine Machine Learning Repository. The dataset contained 30161 records and was configured as in [20]. We used 3 distributed DObjects nodes (30161 records were split among those nodes using round-robin protocol). We report results for the following scenarios: 1) the data is located in one centralized database and classical Mondrian k-anonymity algorithm was run (centralized approach), 2) data is distributed among the three nodes and Mondrian k-anonymity algorithm was run at each site independently (independent or naive approach) and 3) data distributed among the three nodes and we use the distributed anonymization approach presented in section 4. We ran each experiment for different k values.

Figures 5 and 6 show the average equivalence class size and the discernability metric for the three approaches respectively with respect to different values of k. We observe that our distributed anonymization protocol performs the same as the non-distributed version. Also as expected, the naive approach (independent anonymization of each local database) suffers in data utility because the

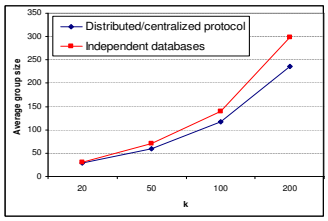

Figure 5: Average equivalence class size vs. k

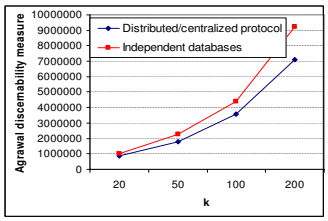

Figure 6: Discernability vs. k

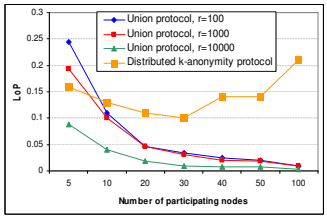

Figure 7: LoP vs. number of nodes (Q1)

anonymization is performed before the integration of the data.

6.2 Data Confidentiality

To evaluate the distributed anonymization approach in terms of data confidentiality guarantee, we ran experiments using the same dataset aiming at analyzing the loss of privacy (LoP). As we described in section 4, our approach includes the distributed anonymization protocol and the distributed querying protocol. The loss of privacy is caused by the sub-protocols used in each of these protocols, namely, the secure sum, median, min/max protocols for the anonymization protocol and the set union protocol for the querying protocol. To evaluate how these

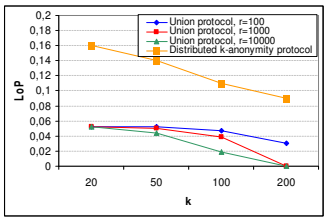

Figure 8: LoP vs. k (Q1)

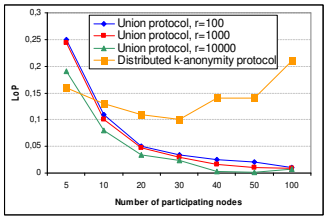

Figure 9: LoP vs. number of nodes (Q2)

sub-protocols contribute to the loss of privacy in each of these protocols, we have run a series of experiments that involved different number of participating nodes and different k values. We have also tested two types of queries: Q1 that selects records with age equal to 55 and Q2 that selects records with age equal to 55 and country equal to 26. The queries returned 386 and 2 records respectively for non-anonymized database (for k-anonymized databases such queries will return many more records, as e.g. age of individuals will be expressed as ranges).

Figures 7 and 8 present the loss of privacy for the distributed anonymization protocol and distributed querying protocol with varying number of participants and varying k respectively for Q1. Similarly, Figures 9 and 10 present

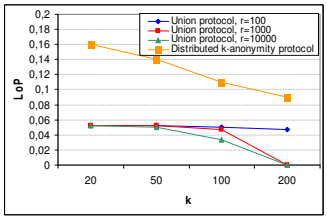

Figure 10: LoP vs. k (Q2)

Parameter name	Description	Default value
m	Size of domain	100,000
n	Number of participating nodes	20
c	Size of algorithm result	1000
r	Number of generated random items	varies

Table 1: Simulation parameters.

the results for Q2. Please also note that the set union protocol used in the querying protocol was tested for different number of generated random items (r). It can be observed that the LoP in both protocols decreases when the number of participants increase and this is very intuitive as the anonymity of the network increases. It can be also observed that LoP in both protocols decreases when k increases. The phenomenon can be explained as follows. When one increases k, the domain that is used for generating random items becomes smaller. In the consequence it is easier to generate random item that is real data (e.g. that is contributed by some node).

6.3 Secure Set Union Protocol

In this section we will present a detailed evaluation of the general secure set union protocol. We prepared a simulation and used synthetically generated data with varying parameters which allowed us to test and evaluate the protocol in multiple scenarios and settings. A summary of the set of simulation parameters we used is presented in Table 1. The default values are used unless otherwise specified. In all the experiments below we have assumed that each node contributes $\frac{c}{n}$ items to the final result. Our experimental results are again divided into two cases: set exposure and item exposure.

Set exposure. The first part of the experimental results is focused on the case of set exposure. Recall our analysis in previous section, Equation 4 gives the LoP bound and Equation 9 provides the minimal number of random items that should be generated in order to achieve an expected LoP bound. Figure 11 presents the analytical expected LoP bound (obtained from Equation 4) and the actual LoP obtained from the experiments when a given number of random items is generated. As can be noticed, given the default number of nodes being 20, even when no random items are generated, the algorithm provides quite high security (expected LoP is 0.05) by utilizing the inherent anonymity of the network of nodes. Generation of only 100 random items (which is 0.1% of the size of domain) causes actual LoP to drop below the level of 0.02. Given a smaller number of nodes, generation of random items becomes even more essential. Importantly, the analytical bound of the expected LoP is always higher than the value of the actual LoP.

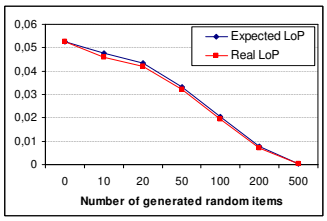

Figure 11: LoP for set exposure vs. number of generated random items

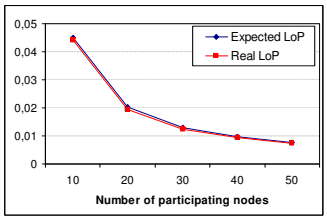

Figure 12: LoP for set exposure vs. number of participating nodes

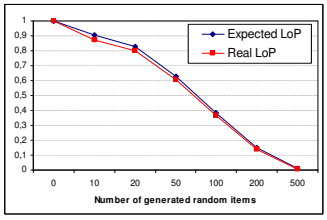

Figure 13: Worst case LoP for set exposure vs. number of generated random items

The second experiment was focused on analysis of the impact of number of nodes participating in the algorithm on LoP. Intuitively, the more nodes are involved, the higher the anonymity of the algorithm (and the lower loss

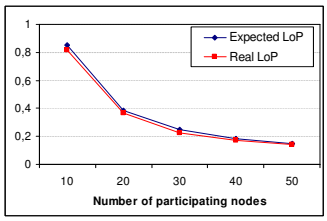

Figure 14: Worst case LoP for set exposure vs. number of participating nodes

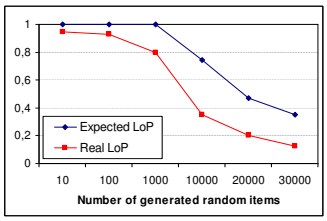

Figure 15: Worst case LoP for item exposure vs. number of generated random items

of privacy). The result of this experiment is presented in Figure 12. For each tested n we generated 100 random items. As expected, both expected and actual LoP decreases while n increases. Again, the value of actual LoP is always lower than the analytical bound.

Finally, we also tested the worst case scenario (i.e. node 2 being aware of that node 1 is starting node). We present the results in Figures 13 and 14. We observe that the LoP is higher than the average case LoP. Nevertheless, generating random items or increasing number of nodes reduces the LoP significantly.

Item exposure. The second part of the experimental evaluation was focused on item exposure. We are focusing on the worst case scenario (assuming that adversary correctly identifies the fact that his predecessor is a leader). Recall from Equation 8 that the expected LoP of the protocol is equal to 0. LoP measure for the worst case was defined in Equation 8 and the requirement on minimal number of random items was given in Equation 10. Figure 15 presents the value of expected and actual LoP metrics as a function of the number of generated random items. It is worth noting that the value of actual LoP metric is around half of the value of expected LoP. Such a phenomenon is worth of explanation. When we derived Equation 10, we have assumed worst case scenario and assumed that $|x_1 \cap (r \cap (x - x_2))| = |x_1|$. On the other hand, in most cases the value of $|x_1 \cap (r \cap (x - x_2))|$ will be much smaller (or even close to 0).

The second experiment for the case of item exposure measured the impact of number of participating nodes in the algorithm on the LoP value. For each tested value n we have generated 10,000 random items (which is 10% of the size of the domain). The result is presented in Figure 16. Similar to the previous case, increase in the number of

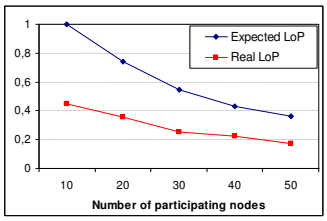

Figure 16: Worst case LoP for item exposure vs. number of participating nodes

participants leads to a reduction in the value of expected and actual LoP.

7 Conclusion

We have presented a distributed anonymization approach for privacy-preserving data publishing for horizontally partitioned databases. It includes a distributed anonymization protocol based on the Mondrian partitioning method and a distributed querying protocol for securely computing the union of data tuples partitioned among n nodes. Our protocols are not based on expensive cryptographic primitives, rather, we leverage the inherent anonymity of the multiple number of participants of the protocol and utilize randomization approaches to achieve minimal information disclosure and minimal overhead. The analysis as well as experimental results show that the protocols provide a desired level of trade-off between the cost of the protocol and the privacy requirements.

Our work continues along several directions. First, we are interested in developing a protocol toolkit incorporating more privacy principles and anonymization al-

gorithms. Second, we are exploring different network topologies and optimization techniques in order to further improve the performance of the protocols. Finally, we are interested in investigating game theoretic approaches to relax the semi-honest model assumption.

References

[1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, pages 901–909, 2005.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *PODS*, pages 153–162, 2006.

[3] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth ranked element, 2004.

[4] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases, 2003.

[5] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pages 217–228, Washington, DC, USA, 2005. IEEE Computer Society.

[6] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pages 217–228, Washington, DC, USA, 2005. IEEE Computer Society.

[7] E. Bertino, B. Ooi, Y. Yang, and R. H. Deng. Privacy and ownership preserving of outsourced medical data. In *ICDE*, 2005.

[8] S. S. Bhowmick, L. Gruenwald, M. Iwaihara, and S. Chatvichienchai. Private-iye: A framework for privacy preserving data integration. In *ICDE Workshops*, page 91, 2006.

[9] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE 2005)*, pages 205–216, Tokyo, Japan, April 2005.

[10] O. Goldreich. Secure multi-party computation, 2001. Working Draft, Version 1.3.

[11] S. Goldwasser. Multi-party computations: past and present. In *ACM Symposium on Principles of Distributed Computing*, 1997.

[12] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD*, pages 279–288, 2002.

[13] W. Jiang and C. Clifton. A secure distributed framework for achieving k-anonymity. *The VLDB Journal*, 15(4):316–333, 2006.

[14] P. Jurczyk and L. Xiong. Dobjects: Enabling distributed data services for metacomputing platforms. In *Proc. of the ICCS (to appear)*, 2008.

[15] M. Kantarcioglu and C. Clifton. Privacy preserving data mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(9), 2004.

[16] M. Kantarcioglu and C. Clifton. Privacy preserving k-nn classifier. In *ICDE*, 2005.

[17] M. Kantarcoglu and J. Vaidya. Privacy preserving naive bayes classifier for horizontally partitioned data. In *IEEE ICDM Workshop on Privacy Preserving Data Mining*, 2003.

[18] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *SIGMOD Conference*, pages 217–228, 2006.

[19] K. LeFevre, D. Dewitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACM SIGMOD International Conference on Management of Data*, 2005.

[20] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *IEEE ICDE*, 2006.

[21] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Workload-aware anonymization. In *SIGKDD*, 2006.

[22] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*

'06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), page 25, Washington, DC, USA, 2006. IEEE Computer Society.

[23] N. Li and T. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *To appear in International Conference on Data Engineering (ICDE)*, 2007.

[24] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3), 2002.

[25] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, page 24, 2006.

[26] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *ICDE*, pages 126–135, 2007.

[27] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS*, pages 223–228, 2004.

[28] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *SIGMOD Conference*, pages 665–676, 2007.

[29] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[30] L. Sweeney. k-anonymity: a model for protecting privacy. *International journal on uncertainty, fuzziness and knowledge-based systems*, 10(5), 2002.

[31] T. M. Truta and B. Vinay. Privacy protection: p-sensitive k-anonymity property. In *ICDE Workshops*, page 94, 2006.

[32] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[33] J. Vaidya and C. Clifton. Privacy preserving naive Bayes classifier for vertically partitioned data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[34] J. Vaidya and C. Clifton. Privacy preserving nave bayes classifier for vertically partitioned data. In *ACM SIGKDD*, 2003.

[35] J. Vaidya and C. Clifton. Privacy-preserving top-k queries. In *ICDE*, 2005.

[36] K. Wang and B. C. M. Fung. Anonymizing sequential releases. In *ACM SIGKDD*, 2006.

[37] K. Wang, P. S. Yu, and S. Chakraborty. Bottom-up generalization: a data mining solution to privacy protection. In *Proc. of the 4th IEEE International Conference on Data Mining (ICDM 2004)*, November 2004.

[38] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *VLDB*, pages 139–150, 2006.

[39] X. Xiao and Y. Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In *SIGMOD Conference*, pages 689–700, 2007.

[40] L. Xiong, S. Chitti, and L. Liu. Topk queries across multiple private databases. In *25th International Conference on Distributed Computing Systems (ICDCS 2005)*, 2005.

[41] L. Xiong, S. Chitti, and L. Liu. Mining multiple private databases using a knn classifier. In *ACM Symposium of Applied Computing (SAC)*, pages 435–440, 2007.

[42] L. Xiong, S. Chitti, and L. Liu. Preserving data privacy for outsourcing data aggregation services. *ACM Transactions on Internet Technology (TOIT)*, 7(3), 2007.

[43] Z. Yang, S. Zhong, and R. N. Wright. Privacy-preserving classification of customer data without loss of accuracy. In *SIAM SDM*, 2005.

[44] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *ICDE*, pages 116–125, 2007.

[45] S. Zhong, Z. Yang, and R. N. Wright. Privacy-enhancing k-anonymization of customer data. In *PODS*, 2005.

[46] S. Zhong, Z. Yang, and R. N. Wright. Privacy-enhancing k-anonymization of customer data. In *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 139–147, New York, NY, USA, 2005. ACM Press.