Top$k$ Queries across Multiple Private Databases

Li Xiong, Subramanyam Chitti, Ling Liu
College of Computing
Georgia Institute of Technology
{lxiong,chittis,lingliu}@cc.gatech.edu

Abstract

Advances in distributed service-oriented computing and global communications have formed a strong technology push for large scale data integration among organizations and enterprises. It is widely observed that multiple organizations in the same market sectors are actively competing as well as collaborating with constantly evolving alliances. Many such organizations want to find out the aggregation statistics about sales in the sector without disclosing sales data in their private databases. Privacy-preserving data sharing is becoming increasingly important for large scale mission-critical data integration applications.

In this paper we present a decentralized peer-to-peer protocol for supporting statistics queries over multiple private databases while respecting privacy constraints of participants. Ideally, given a database query spanning multiple private databases, we wish to compute the answer to the query without revealing any additional information of each individual database apart from the query result. In practice, a popular approach is to relax this constraint to allow efficient information integration while minimizing the information disclosure. The paper has a number of unique contributions. First, we formalize the notion of loss of privacy in terms of information revealed and propose a data privacy metric. Second, we propose a novel probabilistic decentralized protocol for privacy preserving top$k$ selection. Third, we perform a formal analysis of the protocol and also experimentally evaluate the protocol in terms of its correctness, efficiency and privacy characteristics.

1 Introduction

Information integration has long been an important area of research as there is great benefit for organizations and individuals in sharing their data. Traditionally, information integration research has assumed that information in each database can be freely shared. Recently, it has been recognized that concerns about data privacy increasingly becomes an important aspects of the data integration because organizations or individuals do not want to reveal their private databases for various legal and commercial reasons.

Application Scenarios. The increasing need for privacy preserving data integration is driven by several trends [3]. In the business world, with the push of end-to-end integration between organizations and their suppliers, service providers, and trade partners, information sharing may occur across multiple autonomous enterprises. Full disclosure of each database is undesirable. It is also becoming common for enterprises to collaborate in certain areas and compete in others. This in turn requires selective information sharing.

Another important application scenario is driven by security. Government agencies realize the importance of sharing information for devising effective security measures. For example, multiple agencies may need to share their criminal record databases in identifying certain suspects under the circumstance of a terrorist attack. However, they cannot indiscriminately open up their databases to all other agencies.

Such concerns of data privacy place limits on the information integration. We are faced with the challenge of data integration while respecting privacy constraints. Ideally, given a database query spanning multiple private databases, we wish to compute the answer to the query without revealing any additional information of each individual database apart from the query result.

Current Techniques and Research Challenges. There are two main existing techniques that one might use for building the privacy preserving data integration applications and we discuss below why they are inadequate.

One technique is to use a trusted third party and have the participating parties report the data to the trusted third party, which performs the data integration task and reports back the result to each party. However, finding such a trusted third party is not always feasible. The level of trust required for the third party with respect to intent and competence against security breaches is too high. Compromise of the server by hackers could lead to a complete privacy loss for all participating parties should the data be revealed publicly.

The other is the secure multi-party computation approach [9, 8] that developed theoretical methods for securely computing functions over private information such that parties only know the result of the function and nothing else. However, the methods require substantial computation and communication costs and are impractical for multi-party large database problems.

Agrawal et al [3] recently proposed a new paradigm of information integration with minimal necessary sharing across private database. As a tradeoff for efficiency and practicability, the constraint of not revealing any additional information apart from the query result can be relaxed sometimes to allow

minimal additional information to be revealed. As an example, they developed protocols for computing intersection and equijoin between two parties that is still based on cryptographic primitives but more efficient with minimal information disclosure.

Given this paradigm, research opportunities arise for developing efficient specialized protocols for different operations. One important operation is statistics queries over multiple private databases, such as top$k$ data values of a sensitive attribute. In particular, when $k = 1$, it becomes the max(min) query. For example, a group of competing retail companies in the same market sector may wish to find out statistics about their sales, such as the top sales revenue among them, but to keep the sales data private at the same time. The design goal for such protocols is two fold. First, it should be efficient in terms of both computation and communication costs. In order to minimize the computation cost, expensive cryptographic operations should be limited or avoided. Second, it should minimize the information disclosure apart from the query results for each participant.

Contributions and Organizations. Bearing these design goals in mind, we propose a protocol for selecting top$k$ data values of a sensitive attribute across multiple ($n > 2$) private databases. The paper has a number of unique contributions. First, we formalize the data privacy goal and the notion of loss of privacy in terms of information revealed by proposing a data privacy metric (Section 2). Second, we propose a novel probabilistic decentralized protocol for privacy preserving top$k$ selection (Section 3). Third, We perform a formal analysis of the protocol in terms of its correctness, efficiency and privacy characteristics (Section 4) and evaluate the protocol experimentally (Section 5). We provide a brief overview of the related work (Section 6) and conclude the paper with a summary, and a brief discussion of future work (Section 7).

2 Privacy Model

In this section we define the problem of top$k$ queries across private databases. We present the privacy goal that we focus in the paper, followed by privacy metrics for characterizing and evaluating how the privacy goal is achieved.

Problem Statement. The input of the problem is a set of private databases, $D_1, D_2, \cdots, D_n (n > 2)$. A top$k$ query is to find out the top$k$ values of a common attribute of all the individual databases. We assume all data values of the attribute belong to a publicly known data domain. Now the problem is to select the top$k$ values with minimal disclosure of the data values each database has besides the final result.

2.1 Adversary Model

We adopt the *semi-honest* model [8] that is commonly used in multi-party secure computation research for privacy adversaries. A semi-honest party follows the rules of the protocol, but it can later use what it sees during execution of the protocol to compromise other parties' data privacy. Such kind of behavior is referred to as honest-but-curious behavior [8] and also referred to as passive logging [18] in research on anonymous communication protocols.

The *semi-honest* model is realistic for our context based on the following observation. Today multiple organizations in the same market sectors are actively competing as well as collaborating with constantly evolving alliances. These parties often wish to find out aggregation statistics of their sales, such as the total sales or the top$k$ sales among them in a given category or time period, while keeping their own sales data private. As a result, each participating party will want to follow the agreed protocol to get the correct result for their mutual benefits and at the same time reduce the probability and the amount of information leak (disclosure) about their private data during the protocol execution due to competition or other purposes.

Other adversary models include *malicious* model where an adversary can misbehave in arbitrary ways. In particular, it can change its input before entering the protocol or even terminates it arbitrarily. Possible attacks under this model include spoofing attack and hiding attack where an adversary sends a spoofed dataset or deliberately hides all or part of its dataset and leads to a polluted query result. We plan to study the malicious model in our future work.

2.2 Privacy Goal

We focus on the data privacy goal for top$k$ queries in this paper. Ideally, besides the final top$k$ results that are public to all the databases, nodes should not gain any more information about each others data. As we have discussed earlier, with a centralized third party approach, all participating organizations will have to trust this third party and disclose their private data to the third party, which is not only costly in terms of legal and administration procedure but also undesirable by many. We propose a decentralized approach without any third trusted party. Our goal is to minimize data exposure among the multi-parties apart from the final result of the top$k$ query.

We describe the different types of data exposure we consider and discuss our ultimate privacy goal in terms of such exposures. Given a node i and a data value v_i it holds, we identify the following data exposures in terms of the level of knowledge an adversary can deduce about v_i: (1) *Data value exposure*: an adversary can prove the exact value of v_i ($v_i = a$), (2) *Data range exposure*: an adversary can prove the range of v_i ($a \leq v_i \leq b$) even though it may not prove its exact value, and (3) *Data probability distribution exposure*: an adversary can prove the probability distribution of v_i ($pdf(v_i) = f$) even though it may prove neither its range nor exact value.

Both data value and data range exposures can be expressed by data probability distribution exposure, in other words, they are special cases of probability distribution exposure. Data value exposure is again a special case of data range exposure. Intuitively, data value exposure is the most detrimental privacy

breach. Due to the space restriction, we will focus our privacy analysis on the data value exposures in the rest of this paper.

Similar to the exposures at individual node, we can consider data exposures from the perspective of a group of nodes by treating this subset of nodes as an entity. Note that even if a groups privacy is breached, an individual node may still maintain its privacy to some extent. For example, an adversary may be able to prove that a group of nodes has a certain value but it is not certain which exact node has the value. In other words, the m-anonymity [15] is preserved given the size m of the group.

The privacy goal we aim at achieving is to minimize the degree of data value exposures for each individual node. This includes the principle that we are treating all the nodes in the system equally and no extra considerations will be given to the nodes who contribute to the final top$k$ values (e.g., the node who owns the global maximum value). In addition to protecting the data exposure of each node, a related goal could be protecting the anonymity of the nodes who contribute to the final results, though it is not the focus of this paper due to the space limitation.

2.3 Privacy Metrics

Given the data privacy goal, we need to characterize the degree with which the privacy is attained. The key question is how to measure the amount of disclosure during the computation and what privacy metrics are effective for such measurement. Concretely, we need to quantify the degree of data exposure for a single data item v_i that node i holds. Let us first consider an existing metric and discuss why it is inadequate. We then propose a general and improved metric for data privacy.

The metric that one might use is the probabilistic privacy spectrum [14] proposed for web transactions anonymity and was also adopted for document ownership privacy later [5]. Now we need to evaluate whether we can adapt it for our data privacy purpose. Assuming an adversary is able to make a claim C about the data value v_i (e.g., $v_i = a$), based on the intermediate result it sees during the execution. The privacy spectrum can be defined based on the probability that the claim is true. On one extreme is *provably exposed* where an adversary can prove that $v_i = a$ (with probability of 1). On the other extreme is *absolute privacy* where an adversary cannot determine the exact value of v_i (with probability of 0). In between, there are *possible innocence* where the claim is more likely to be true, and *probable innocence* where the claim is less likely to be true. A particularly interesting notion is *beyond suspicion* where a node is no more likely to have a value that satisfies the claim than any other nodes in the system. This is also known as m-anonymity as we have mentioned earlier.

A closer look at the spectrum shows that it does not capture the important differences among different claims for our data privacy concerns. Consider an adversary that makes a claim $v_i = a$ after executing a max query ($k = 1$) and the probability of the claim being true is $1/n$, i.e. there are some node(s) in the system that have the value but none is more likely than others to have it. By the privacy spectrum, the degree of data value exposure for the node is *beyond suspicion*. However, if $a = v_{max}$, where v_{max} denotes the final maximum value, it should not be considered as a privacy breach at all. This is because v_{max} is public information to all the nodes after the protocol and every node has a probability $1/n$ holding v_{max}. On the other hand, if $a \neq v_{max}$, then it is indeed a privacy breach because other nodes would not haven known anything about the value a by just knowing v_{max}.

In fact, such differences among different claims are more obvious and important for data range privacy. Intuitively, a data range exposure with a very precise (small) range is much more severe than those with a large range. For example, consider the case where an adversary is able to prove $v_i \leq a$. By the privacy spectrum, node i has *provable exposure* regarding its data range. However, the severity of the privacy breach actually varies (decreases as a increases). At the extreme, if $a = v_{max}$, it should not be considered as a privacy breach at all because $v_i \leq v_{max}$ is known to all the nodes after the final result of v_{max} is returned.

We propose a general metric - loss of privacy - to characterize how severe a data exposure is by measuring the relative loss in the degree of exposure. Let R denote the final result set after the execution and IR denote the intermediate result set during the execution. Let $P(C|R, IR)$ denote the probability of C being true given the final result and the intermediate results, and similarly, $P(C|R)$ the probability given only the final result. We define Loss of Privacy (LoP) in Equation 1. Intuitively, this gives us a measure of the additional information an adversary may obtain given the knowledge of the intermediate result besides the final query result.

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

We illustrate the metric for data value privacy in the context of top$k$ queries, where C is in the form of $v_i = a$ and R is the final top$k$ values denoted as $TopK$. If $a \in TopK$, every node has the same probability to hold a so we have $P(v_i = a|TopK) = 1/n$. Otherwise ($a \notin TopK$), it is close to impossible for an adversary to guess the exact value of a node given only the final result. This is especially true when the data domain is large enough, because a node can take any of the values in the data domain. So we approximate $P(v_i = a|TopK)$ with 0. Thus the LoP is slightly smaller when $a \in TopK$. In the cases where all an adversary knows is that some node in the system has a value equal to $a (a \in TopK)$, we have $P(v_i = a|TopK, IR) = 1/n$ and $LoP = 0$.

Given the definition of LoP for a single data item at a single node, we define LoP for a node as the average LoP for all the data items used by a node in participating the protocol. Intuitively, when nodes participate the protocol with their local

top$k$ values, the more values that get disclosed, the larger the LoP for the node. We measure the privacy characteristics for the system using the average LoP of all the nodes.

3 The Decentralized Protocol

In this section we describe a decentralized computation protocol for multiple organizations to compute top$k$ queries over n private databases (nodes) with minimum information disclosure from each organization.

Bearing the privacy goal in mind, we identify two important principles for our protocol design. First, the output of the computation at each node should prevent an adversary from being able to determine the nodes data value or data range with any certainty. Second, the protocol should be able to produce the correct final output of a top$k$ query (effectiveness) in a small and bounded number of rounds of communication among the n nodes (efficiency). Using these principles as the design guidelines, we propose a probabilistic protocol with a randomized local algorithm for top$k$ queries across n private databases ($n \geq 3$). 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 decentralized probabilistic protocol.

3.1 A Naive Protocol

Consider a group of n databases who wish to select the max value ($k = 1$) of a common attribute. A straightforward way to compute the result without a central server is to have the nodes arranged in a ring in which a global value is passed from node to node along the ring. The first node sends its value to its successor. The next node computes the current max value between the value it gets from its predecessor and its own value and then passes the current max value to its successor. At the end of the round, the output will be the global max value.

Clearly, the scheme does not provide good data privacy. First, the starting node has *provable exposure* to its successor regarding its value. Second, the nodes that are close to the starting node in the ring have a fairly high probability disclosing their values. A randomized starting scheme can be used to protect the starting node and avoid the worst case but it would not help with the average data value disclosure of all the nodes on the ring. In addition, every node i ($1 \leq i \leq n$) suffers *provable exposure* to its successor regarding its data range, i.e. the successor knows for sure that node i has a value smaller than the value it passes on. This leads us to consider alternative protocols for better privacy preservation.

In the rest of this section, we present our probabilistic protocol. We first give a brief overview of the key components of the protocol and then use the max (min) queries (the top$k$ queries with $k = 1$) to illustrate how the two design principles are implemented in the computation logic used at each node (private database) to achieve the necessary minimum disclosure of private information (our privacy goal).

3.2 Protocol Structure

The protocol is designed to run over a decentralized network with a ring topology, and consists of the node communication scheme, the local computation module and initialization module at each node.

Ring Topology. Nodes are mapped into a ring randomly. Each node has a predecessor and successor. It is important to have the random mapping to reduce the cases where two colluding adversaries are the predecessor and successor of an innocent node. We will discuss more on this in Section 4.

Communication protocol. The communication among the nodes is from a node to its successor. Encryption techniques can be used so that data are protected on the communication channel. In case there is a node failure on the ring, the ring can be reconstructed from scratch or simply by connecting the predecessor and successor of the failed node.

Local computation module. The local algorithm is a standalone component that each node executes independently. Nodes follow the *semi-honest* model and executes the algorithm correctly.

Initialization module. The initialization module is designed to select the starting node among the n participating nodes and then initialize a set of parameters used in the local computation algorithms.

In this paper we do not handle the data schema heterogeneity issues. We assume that the database schemas and attribute names are known and are well matched across n nodes. Readers who are interested in this issue may refer to [7] for some approaches to the problem of schema heterogeneity.

3.3 Privacy Preserving Max Selection

Before going into details of the protocol, we first present the local computation component of the protocol for max(min) queries (the special case of top$k$ with $k = 1$) over n private databases. This will help readers understand the key ideas and techniques used in our protocol design. We describe the general protocol for top$k$ queries in next subsection.

The intuitive idea of using a probabilistic protocol is to inject some randomization into the local computation at each node, such that the chance of data value disclosure at each node is minimized and at the same time the eventual result of the protocol is guaranteed to be correct. Concretely, the protocol performs multiple rounds in which a global value is passed from node to node along the ring. A randomization probability is associated with each round and decreased in the next round to ensure that the final result will be produced in a bounded number of rounds. During each round, nodes inject certain randomization in their local computation with the given probability. The randomization probability is eventually decreased to 0 so that the protocol outputs the correct result.

Randomization Probability. We first define the randomization probability. It starts with an initial probability denoted as

p_0 in the first round and decreases exponentially with a dampening factor denoted as d, so that it tends to 0 with sufficient number of rounds. Formally, the randomization probability for round r denoted as $P_r(r)$ is defined as follows:

$$P_r(r) = p_0 * d^{r-1} \quad (2)$$

Randomized Algorithm. Each node, upon receiving the global value from its predecessor, performs the local randomized algorithm, and passes the output to its successor. The core idea of this algorithm is to determine when (the right time) to inject randomization and how much (the right amount of randomization) in order to implement the two design principles of the protocol: namely, the output of the algorithm should prevent an adversary from inferring the value or range of the data that the node holds with any certainty; and the randomized output should not generate potential errors that lead to incorrect final output of the protocol.

Algorithm 1 Local Algorithm for Max Protocol (executed by node i at round r)

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

A sketch of the randomized algorithm is given in Algorithm 1 for node i at round r. The algorithm takes two inputs: (1) the global value node i receives from its predecessor $i-1$ in round r, denoted as $g_{i-1}(r)$, and (2) its own value, denoted as v_i. The algorithm compares these two input values and determines the output value, denoted as $g_i(r)$, in the following two cases. First, if the global value $g_{i-1}(r)$ is greater than or equal to its own value v_i, node i simply returns the current local maximum value ($g_{i-1}(r)$ in this case). There is no need to inject any randomization because the node does not expose its own value in this case. Second, if $g_{i-1}(r)$ is smaller than v_i, instead of always returning the current local maximum value (v_i in this case), node i returns a random value with probability $P_r(r)$, and only returns v_i with probability $1 - P_r(r)$. The random value is generated uniformly from the range $[g_{i-1}(r), v_i)$. Note that the range is open ended at v_i to warrant that the node will not return v_i when we want to return a constrained random value instead of the actual v_i.

Such randomization has a number of important properties. First, it successfully prevents an adversary from deducing the value or range of v_i with any certainty. This is because the output of node i can be either a random value, or the global value passed by the predecessor of node i, or its own value v_i. Second, the global value monotonically increases as it is passed along the ring, even in the randomization case. Recall the case when randomization is injected, the random value output $g_i(r)$ can be smaller than v_i but has to be greater than or equal to $g_{i-1}(r)$, which ensures that the global value keeps increasing. This monotonic increasing property further minimizes the need for other nodes after node i to have to disclose their own values because they can simply pass on the global value if it is greater than their own values. Finally, the randomized value will not generate any potential errors for the protocol because it is always smaller than v_i and thus smaller than the global maximum value. It will be replaced by the value that is held either by the node i itself or any other node that holds a greater value in a later round as the randomization probability decreases. We will analyze the correctness and data value privacy of the protocol formally in Section 4.

Protocol Details. We now walk through the protocol by describing the initiation process, the communication scheme, combined with the local computation logic.

At the initiation state, every node in the network sorts their values and takes the local max value to participate the global max selection. The protocol randomly chooses a node from the n participating nodes, say indexed by i with $i = 1$. In addition, the initialization module will set the default global value $g_0(1)$ to the lowest possible value in the corresponding data domain, and initialize the randomization probability with p_0, the dampening factor d (recall Section 3.2), and the round counter r. The key idea of using a randomized selection scheme for starting node is to preserve the anonymity of the starting node so an adversary does not know where the protocol start from and hence protecting the starting node.

Upon the completion of the initiation process, the local computation module is invoked at node i. Each node i, upon receiving the global value $g_{i-1}(r)$ from its predecessor at round r, executes the local computation algorithm, and passes the output $g_i(r)$ to its successor. At the end of each round r ($r \geq 1$), the last node n passes the current global value $g_n(r)$ to the first node, which serves as the input $g_0(r+1)$ at the first node in round $r+1$. The protocol terminates at the starting node after a sufficient number of rounds. We will discuss how to determine the number of rounds needed and what we mean by sufficient in Section 4. It is interesting to note that if we set the initial randomization probability to be 0 ($p_0 = 0$), the protocol is reduced to the naive deterministic protocol.

Figure 1 shows an example walk-through of the protocol over a network of 4 nodes, initialized with $p_0 = 1$ and $d = 1/2$. Assume the protocol starts from node 1 with the initial global value $g_0(1) = 0$. In the first round ($r = 1$), the randomization probability $P_r(1)$ is initialized to 1, so if a node receives a value smaller than its own value, it will always return a random value between the received value and its own value. As a result, node 1 returns a random value between [0,30), say 16. Node 2 passes 16 to node 3 because it is greater than its own value 10. Node 3 returns a random value between [16,40), say 25, since value 16 is smaller than its own value 40. Node passes value 25 to

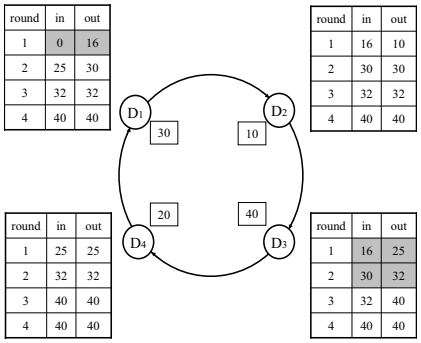


Figure 1: Illustration for Max Protocol Walk-through

the first node because it is greater than its own value 20. In the second round ($r = 2$), the randomization probability $P_r(2)$ decreases to $1/4$ according to equation 2. As a result, node 1 returns its own value 30. Node 2 passes on value 30. Node 3 returns a random value between [30,40), say 32. Node 4 passes on value 32. In the third round ($r = 3$), the randomization probability $P_r(3)$ decreases to 1/4. Node 1 and Node 2 both pass on the value 32. Node 3 finally returns its own value 40 and node 4 passes on the value 40. In the termination round all nodes simply passes on the final result.

This example illustrates how our probabilistic protocol works and why our protocol ensures that each node retains good privacy about the exact value and the range of their data. It is important to note that the random selection scheme for the starting node plays an important role for preserving good privacy of the starting node. For instance, in the above example, if node 1 was known as the starting node, then upon receiving 16 from node 1 in the first round, node 2 knows for sure that node 1 has a value greater than 16, leading to the data range exposure for node 1.

We provide an analytical model to formally study the correctness and data value privacy of the protocol in Section 4 and report the result of our experimental evaluation in Section 5.

3.4 Privacy Preserving Top-k Selection

Now we describe the general protocol for top$k$ selection. It works similarly as the max selection protocol ($k = 1$) in terms of the probabilistic scheme. The complexity of extending the protocol from max to general top$k$ lies in the design of the randomized algorithm.

At the protocol initial step, each node first sorts its values and takes the local set of top$k$ values as its local top$k$ vector to participate in the protocol, since it will have at most k values that contribute to the final top$k$ result. Similar to the max selection protocol, the initialization module randomly picks a node from the n participating nodes as the starting node, initializes the global top$k$ vector to the lowest possible values in the corresponding data domain, sets the round counter r, and initializes the randomization probability p_0 and the dampening factor d (recall Equation 2 in Section 3.3).

The protocol performs multiple rounds in which a current global top$k$ vector is passed from node to node along the ring network. Each node i, upon receiving the global vector from its predecessor at round r, performs a randomized algorithm and passes its output to its successor node. The starting node terminates the protocol after a sufficient number of rounds.

Randomized Algorithm. The randomized algorithm is the key component of the probabilistic top$k$ selection protocol. We want the algorithm to have the same properties as those of the max selection algorithm (Algorithm 1) when deciding the right time and the right amount of randomization to inject, namely, to guarantee the correctness on one hand and minimize the data value disclosure on the other hand. For example, we can use the same idea of generating random values and inject them into the output of the global top$k$ vector at node i ($1 \leq i \leq n$) in order to hide the nodes own values. However, with k values in the local top$k$ vector, we need to make sure that the randomly generated values will eventually be shifted out from the final global top$k$ vector. In other words, it is not as straightforward as in the max selection algorithm where a random value less than a nodes value will be replaced eventually.

Algorithm 2 Local Algorithm for Top$k$ Protocol (executed by node i at round r)

INPUT: $G_{i-1}(r)$, V_i, OUTPUT: $G_i(r)$
$P_r(r) \leftarrow p_0 * d^{r-1}$
$G'_i(r) = \text{topK}(G_{i-1}(r) \cup V_i)$
$V'_i \leftarrow G'_i(r) - G_{i-1}(r)$
$m \leftarrow |V'_i|$
if $m = 0$ **then**
  $G_i(r) \leftarrow G_{i-1}(r)$
else
  with probability $1 - P_r(r)$: $G_i(r) \leftarrow G'_i(r)$
  with probability $P_r(r)$:
  $G_i(r)[1 : k - m] \leftarrow G_{i-1}(r)[1 : k - m]$
  $G_i(r)[k - m + 1 : k] \leftarrow$ sorted list of m random values from $[min(G'_i(r)[k] - \delta, G_{i-1}(r)[k - m + 1]), G'_i(r)[k])$
end if

Algorithm 2 gives a sketch of a randomized algorithm for general top$k$ selection with respect to node i executing at round r. The input of the algorithm is (1) the global vector node i receives from its predecessor $i - 1$ in round r, denoted as $G_{i-1}(r)$, and (2) its local top$k$ vector, denoted as V_i. The output of the algorithm is the global vector denoted as $G_i(r)$. Note that the global vector is an ordered multiset that may include duplicate values.

The algorithm first computes the real current top$k$ vector, denoted as $G'_i(r)$, over the union of the set of values in $G_{i-1}(r)$ and V_i, say, using a merge sort algorithm. It then computes a sub-vector of V_i, denoted as V'_i, which contains only the values of V_i that contribute to the current top$k$ vector $G'_i(r)$ by taking a set difference of the set of values in $G'_i(r)$ and $G_{i-1}(r)$. Note

that the union and set difference here are all multiset operations. The algorithm then works under two cases.

Case 1: The number of elements in V_i', m, is 0, i.e. node i does not have any values to contribute to the current top$k$. In this case, node i simply passes on the global top$k$ vector $G_{i-1}(r)$ as its output. There is no randomization needed because the node does not expose its own values.

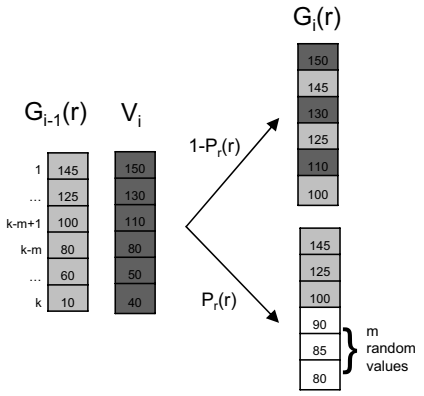


Figure 2: Illustration for Top$k$ Local Algorithm

Case 2: Node i contributes $m (0 < m \leq k)$ values in the current top$k$. Figure 2 gives an illustrative example where $m = 3$ and $k = 6$. In this case, node i only returns the real current top$k$ ($G_i'(r)$) with probability $1 - P_r(r)$. Note that a node only does this once, i.e. if it inserts its values in a certain round, it will simply pass on the global vector in the rest of the rounds. With probability $P_r(r)$, it copies the first $k - m$ values from $G_{i-1}(r)$ and generate last m values randomly and independently from $[min(G_i'(r)[k] - \delta, G_{i-1}(r)[k - m + 1]), G_i'(r)[k])$, where $G_i'(r)[k]$ denotes the kth (last) item in $G_i'(r)$, $G_{i-1}(r)[k-m+1]$ denotes the $k-m+1$th item in $G_{i-1}(r)$, and δ denotes a minimum range for generating the random values. The reason for generating m random values is because only the last m values in the output are guaranteed to be shifted out in a later round when the node inserts its real values if the global vector has not been changed by other nodes. The range is designed is such a way that it increases the values in the global vector as much as possible while guaranteeing the random values do not exceed the smallest value in the current top$k$ so they will be eventually replaced. In an extreme case when $m = k$, the current top$k$ vector is equal to V_i, it will replace all k values in the global vector with k random values, each randomly picked from the range between the first item of $G_{i-1}(r)$ and the kth (last) item of V_i.

It is worth noting that when $k = 1$ the local top$k$ selection algorithm becomes the same as the local algorithm for max protocol. We report our experimental evaluation on the correctness and privacy characteristics of the general protocol in Section 5.

4 Analysis

We conducted a formal analysis on the max protocol in terms of its correctness, efficiency, and privacy characteristics.

4.1 Correctness

Let $g(r)$ denote the global value at the end of round r and $P(g(r) = v_{max})$ denote the probability that $g(r)$ is equal to the global max value v_{max}. At round $j (1 \leq j \leq r)$, if the global value has not reached v_{max}, the nodes who own v_{max} have a probability $1 - P_r(j)$ to replace the global value with v_{max}. Once the global value reaches v_{max}, all nodes simply pass it on. So after round r, the global value will be equal to v_{max} as long as one of the nodes that owns v_{max} has replaced the global value with v_{max} in any of the previous rounds. Thus the probability of the protocol returning the global maximum value after round r can be computed as $P(g(r) = v_{max}) \geq 1 - \prod_{j=1}^{r} P_r(j)$. If we substitute $P_r(j)$ with $p_0 * d^{j-1}$ by Equation 2, we can derive the following equation:

$$P(g(r) = v_{max}) \geq 1 - p_0^r * d^{\frac{r(r-1)}{2}} \quad (3)$$

Equation 3 shows that, for any $0 < p_0 \leq 1$ and $0 < d < 1$, the precision bound increases monotonically with increasing r. For any given number of nodes, we can make the computed global value equal to the global max value with a probability very close to 1 by increasing the number of rounds.

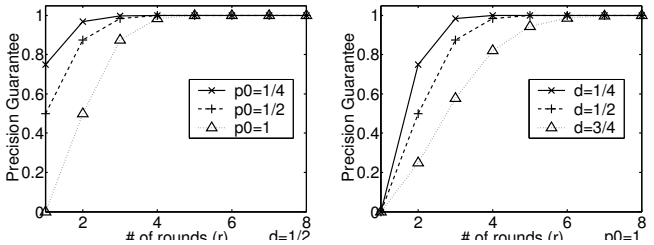


Figure 3: Precision Guarantee with Number of Rounds

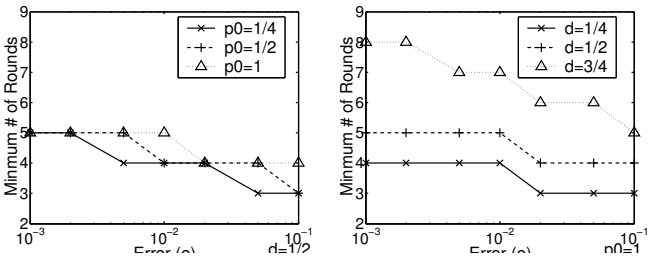
Figure 3(a) and (b) plot the precision bound in equation 3 with increasing number of rounds (r) for varying initial randomization probability (p_0) and dampening factor (d) respectively. We can see that the precision increases with the number of rounds. A smaller p_0 with a fixed d results in a higher precision in the earlier round and reaches the near-perfect precision of 100% faster. A smaller d with a fixed P_0 makes the protocol reach the near-perfect precision of 100% even faster.

4.2 Efficiency

Now we analyze the efficiency of the protocol in terms of the computation and communication cost. The computation at each node in the protocol does not involve any cryptographic operations and should be negligible compared to the communication cost over a network of n nodes. The communication cost

is determined by two factors. The first is the cost for a single round which is proportional to the number of nodes on the ring. The second is the number of rounds that is required for a desired precision. For any $\epsilon(0 < \epsilon < 1)$, we can determine a minimum number of rounds, denoted by r_{min}, such that the result is equal to the global max value with the probability greater than or equal to $1 - \epsilon$. Since we have $P(g(r) = v_{max}) \geq 1 - p_0^r * d^{\frac{r(r-1)}{2}} \geq 1 - p_0 * d^{\frac{r(r-1)}{2}}$ from Equation 3, we can ensure $P(g(r) = v_{max}) \geq 1 - \epsilon$ by requiring $1 - p_0 * d^{\frac{r(r-1)}{2}} \geq 1 - \epsilon$. We solve this equation and derive a minimum number of rounds for the desired precision $(1 - \epsilon)$ as follows:

$$r_{min} = \lceil \frac{1}{2} * (1 + \sqrt{8 * \frac{\log(\epsilon/p_0)}{\log d} - 1}) \rceil \quad (4)$$



(a) (b)

Figure 4: Required Number of Rounds with Precision Guarantee

We can see that the minimum number of rounds r_{min} scales well with the desired precision $(1-\epsilon)$ in the order of $O(\sqrt{log\epsilon})$. Figure 4(a) and (b) plot the minimum number of rounds in Equation 4 for varying error bound (ϵ) with varying initial randomization probability (p_0) and dampening factor (d) respectively. Note that the X axis is of logarithmic scale. We can see that the protocol scales well with increasing desired precision (decreasing ϵ). In addition, a smaller p_0 and a smaller d are desired for better efficiency with d having a larger effect on the reduction of the required number of rounds.

It is important to note that the minimum number of rounds is independent of the number of nodes. Hence, the overall communication cost is proportional to the number of nodes. One possible way to improve the efficiency for a system with a larger number of nodes is to break the set of n nodes into a number of small groups and have each group compute their group maximum value in parallel and then compute the global maximum value at designated nodes, which could be randomly selected from each small group.

4.3 Data Value Privacy

In addition to ensuring correct output and increasing efficiency of the protocol, another important goal of the protocol is preserving the data privacy of individual participating nodes in the network. The communication between nodes consists of sending the current global value from one node to its successor. An adversary may utilize the value it receives from its predecessor to try to gain information about the data its predecessor and other nodes hold. We dedicate this section to analyzing the loss of data value privacy in the protocol using the metric we proposed in Section 2.

Without loss of generality, we assume the protocol starts from node 1. We now analyze the loss of privacy for node i with value v_i. Since node i passes its current global value $g_i(r)$ to its successor $i + 1$ in round r, all the successor can do with respect to the exact data value node i holds is to guess that $v_i = g_i(r)$. Recall Equation 1 in Section 2, the loss of privacy is defined as the relative probability of a node holding a particular value with and without the intermediate result. Let $P(v_i = g_i(r)|g_i(r), v_{max})$ denote the probability that node i holds the value $g_i(r)$ with the knowledge of the intermediate result $g_i(r)$ and $P(v_i = g_i(r)|v_{max})$ denote the probability without it. If $g_i(r) = v_{max}$, we have $P(v_i = g_i(r)|v_{max}) = 1/n$ as all nodes have the same probability holding the global maximum value. Otherwise, we approximate $P(v_i = g_i(r)|v_{max})$ with 0 as we have discussed earlier in Section 2. Now let us look at $P(v_i = g_i(r)|g_i(r), v_{max})$ for both naive protocol and probabilistic protocol and derive the Loss of Privacy (LoP).

Naive Protocol. In the naive protocol where only one round is needed, the global value g_i that node i passes on is the current maximum value of all its previous nodes and itself. Since all of them have the same probability to hold the current maximum value, so we have $P(v_i = g_i(r)|g_i(r), v_{max}) = 1/i$. Thus the data value loss of privacy for node i in naive protocol is $1/i1/n$ if $g_i = v_{max}$ and $1/i$ otherwise.

It is clear that the loss of privacy for node i in the naive protocol depends on its position. Nodes closer to the starting node suffer a larger loss of privacy. In the worst case, the starting node ($i = 1$) has *provable exposure* of its value to its successor. By average, the loss of privacy is greater than $\sum_{i=1}^{n}(1/i - 1/n)/n$. Using the inequality bound for $\sum_{i=1}^{n} 1/i$ (the nth Harmonic number [13]), we can derive the average LoP bound for the naive protocol in Equation 5. We can see that it is fairly high especially when n is small.

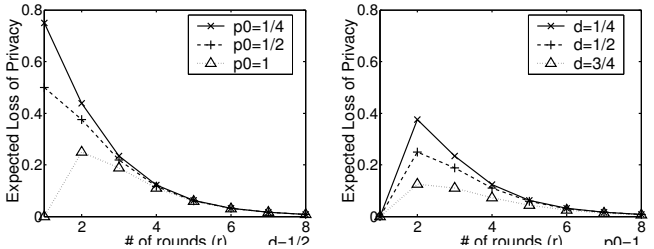
$$LoP_{Naive} > \frac{\ln n}{n} \quad (5)$$

Probabilistic Protocol. The probabilistic max selection protocol requires multiple rounds. Since aggregating the values a node passes on in different rounds does not help with determining its exact data value, though it may help with determining the probability distribution of the value, we first compute the loss of privacy for node i at each round r and then take the highest result in all the rounds as final loss of privacy for node i.

Recall the probabilistic computation in Algorithm 1, a node only replaces the global value with its own value with probability $1 - P_r(r)$ when the value it receives is smaller than its own value. Thus we have $P(v_i = g_i(r)|g_i(r), v_{max}) = P(v_i >$

$g_{i-1}(r)) * (1 - P_r(r)) + P(v_i = g_{i-1}(r))$. We performed an analysis on the expected value for $g_i(r)$ and derived an approximate lower bound of expected LoP for node i in round r. By taking the highest (maximum) LoP of all rounds for each individual node, we derive the average expected LoP for all the nodes in Equation 6.

$$E(LoP_{probabilistic}) \leq max_r(\frac{1}{2^{r-1}} * (1 - p_0 * d^{r-1})) \quad (6)$$



(a) (b)

Figure 5: Expected Loss of Privacy in Different Rounds

From Equation 6, we can see that the expected loss of privacy for the probabilistic protocol depends on how we choose the randomization parameters. Also intuitively, the highest (peak) loss of privacy may happen at different rounds with different randomization parameters. Figure 5 plots the behavior of the bound inside the max function with varying randomization parameters. Figure 5(a) shows the effect of p_0 with d set to 1/2. It is interesting to see that p_0 plays an important role in the loss of privacy. A larger p_0 results in a lower loss of privacy in the first round. The reason is quite intuitive since a larger p_0 implies that more nodes have injected randomized values instead of returning the real current max value in the computation. With a smaller p_0, the loss of privacy gradually decreases from the peak of loss as the protocol converges. With a larger p_0, such as $p_0 = 1$ as shown, the loss of privacy starts with 0 in the first round and increases in the second round to the peak of loss, and then gradually decreases. If we compare the peak loss of privacy in different rounds, we conclude that a larger p_0 provides a better privacy. Figure 5(b) shows the effect of d with p_0 set to 1. We see that a larger d corresponds to a lower loss of privacy, starting from the second round, though with a small margin. Overall, by tuning the parameters p_0 and d, we can keep the loss of privacy very low. Our experimental evaluation in Section 5 confirms with our analytical results regarding the loss of privacy.

Now we briefly discuss the loss of privacy under the scenario where the predecessor and the successor of node i happen to collude with each other. Assuming that an adversary has intermediate results of both $g_{i-1}(r)$ and $g_i(r)$, we have $P(v_i = g_i(r)|g_{i-1}(r), g_i(r), v_{max}) = 1 - P_r(r)$ when $g_{i-1}(r) < g_i(r)$. Knowing this still does not give the adversary any certainty in determining the data value of node i, especially in the beginning rounds when $P_r(r)$ is large. Intuitively, when $P_r(r)$ gets close to 0, $g_{i-1}(r)$ should be already getting close to v_{max} so there is very small chance for the data at node i to be disclosed. It is interesting to note though, if node i happens to hold v_{max} then it will be susceptible to *provable exposure* if it has two colluding neighbors. One technique to minimize the effect of collusion is for a node to ensure that at least one of its neighbors is trustworthy. This can be achieved in practice by having nodes arrange themselves along the network ring(s) according to certain trust relationships such as digital certificate based [6] combined with reputation-based [20]. Further, we can extend the probabilistic protocol by performing the random ring mapping at each round so that each node will have different neighbors at each round.

5 Experimental Evaluations

This section presents a set of initial results from our experimental evaluation of the protocols in terms of correctness and privacy characteristics.

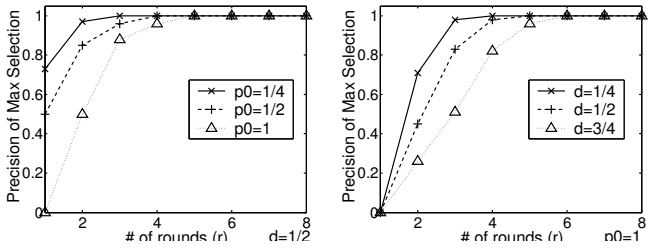
5.1 Experiment Setup

Param.	Description
n	# of nodes in the system
k	parameter in top$k$
p_0	initial randomization prob.
d	dampening factor for randomization prob.

Table 1: Experiment Parameters

The system consists of n nodes. The attribute values at each node are randomly generated over the integer domain $[1, 10000]$. We experimented with various distributions of data, such as uniform distribution, normal distribution, and zipf distribution. The results are similar so we only report the results for the uniform distribution. The experiment proceeds by having the nodes compute top$k$ values using the probabilistic protocol. We evaluate the accuracy and privacy properties. Each plot is averaged over 100 experiments. Table 1 lists the main parameters for the experiments.

5.2 Precision of Max Selection



(a) (b)

Figure 6: Precision of Max Selection with Number of Rounds

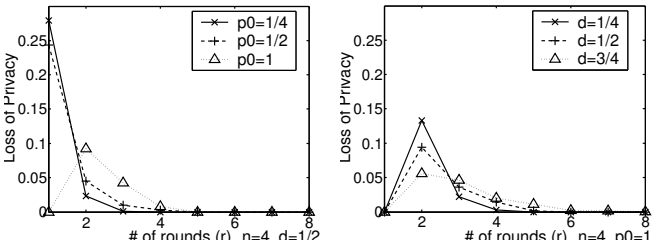
We first verify the correctness of the probabilistic max protocol ($k = 1$). Figure 6(a) and (b) show the precision with increasing number of rounds (r) for different initial randomization prob-

ability (P_0 and dampening factor (d) respectively. We see that the experimental results match the analytical bounds in Figure 3. The precision reaches to 100% as the number of rounds increases. A smaller P_0 results in a higher precision in the first round and makes the precision go up to 100% faster as the number of rounds increases, though with a small margin. A smaller d reaches the perfect precision of 100% much faster.

5.3 Privacy Characteristics of Max Selection

We evaluate the privacy characteristics of the protocol in terms of their data value loss of privacy. In particular, we want to answer a number of questions. What is the loss of privacy during the execution of the algorithm? How does the number of nodes affect the privacy characteristics? How do the randomization parameters affect the privacy characteristics and how to select them? How does the protocol compare to the naive protocol?

Loss of Privacy in Different Rounds. We first study the loss of privacy of the protocol in each round during the execution with different randomization parameters. We experimented with different number of nodes and the trends in different rounds are similar but most pronounced with a small number of nodes. So we only report the results for $n = 4$ to show the different loss of privacy in different rounds and will present another set of experiments later to show the effect of varying number of nodes.



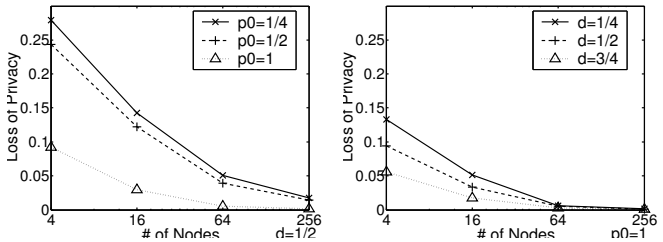
(a) (b)

Figure 7: Loss of Privacy for Max Selection in Different Rounds

Figure 7(a) and (b) show the average data value loss of privacy for all nodes in different rounds with varying initial randomization probability (p_0) and dampening factor (d) respectively. The result matches our analysis in Section 4. With a smaller P_0, the highest loss of privacy happens in the first round and it gradually decreases as the protocol converges. With a larger P_0 (e.g., $P_0 = 1$), the loss of privacy is zero in the first round and reaches the peak in the second round and then gradually decreases. If we look at the peak loss of privacy, a larger p_0 provides a better privacy. In Figure 7(b), all three cases ($d = 1$, $d = 1/2$, $d = 1/4$) start with zero loss of privacy in the first round, and increase to the highest (peak loss) in the second round, and decreases as the protocol converges. A smaller d results in a higher peak loss of privacy.

In this set of experiments, we have shown the loss of privacy in different rounds during the execution. For the rest of the experiments we will take the highest (peak) loss of privacy among all the rounds for a given node to measure its overall loss of privacy, because that gives us a measure of the highest level of knowledge an adversary can obtain regarding the nodes data value.

Effect of Number of Nodes. We now report the experiments showing how the number of nodes affects the loss of privacy of the protocol.



(a) (b)

Figure 8: Loss of Privacy for Max Selection with Different Number of Nodes

Figure 8(a) and (b) show the average data value loss of privacy for all nodes with varing initial randomization probability (P_0) and dampening factor (d) respectively. We can that the loss of privacy decreases with increasing number of nodes. This is very intuitive because the larger the number of nodes, the faster the global value increases and thus the less probability the nodes have to disclose their own values. Again, the result shows that a smaller P_0 and d provide a better privacy.

Selection of Randomization Parameters. This set of experiments is dedicated to study the effect of randomization parameters on both privacy characteristics and efficiency of the protocol. Recall the experiments described so far and our analysis in Section 4, a smaller p_0 and d provide better privacy but requires more cost in terms of number of rounds required. Our design goal is to increase the efficiency while minimizing the loss of privacy. Put differently, we want to select a pair of p_0 and d parameters that gives us the best tradeoff between privacy and efficiency.

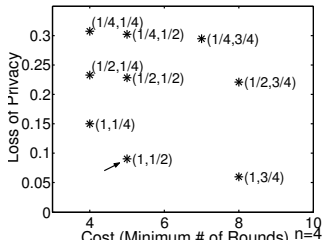
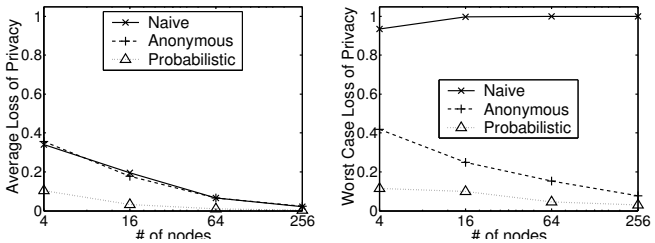


Figure 9: Tradeoff between Privacy and Efficiency with Randomization Parameters

Figure 9 shows the loss of privacy on X axis and the cost in terms of number of rounds for a given precision guarantee ($\epsilon = 0.001$) on Y axis for varying randomization parameter pairs (p_0, d). We can see that p_0 has a dominating effect on the loss of privacy while d has a dominating effect on efficiency. The (p_0, d) pair of $(1, 1/2)$ in the lower left corner gives a nice tradeoff of privacy and efficiency. Therefore, we

will use $p_0 = 1$ and $d = 1/2$ as our default parameters for the rest of the experiments.

Comparison of Different Protocols. We have discussed the naive protocol with fixed starting node, and our probabilistic protocol. The experiments reported below compare the probabilistic protocol with the naive protocol. For comparison purposes, we include the anonymous naive protocol which uses a randomized starting scheme, instead of fixed starting node, to provide the anonymity of the starting node. We show both average and worst case loss of privacy for different nodes in the system. By worst case, we mean highest loss of privacy among all the nodes and it typically happens at the starting node in the fixed starting scheme. Our goal is to show the effectives of our probabilistic protocol over the naive ones and the benefit of randomly selecting the starting node.



(a) (b)

Figure 10: Comparison of Loss of Privacy with Number of Nodes

Figure 10(a) and (b) show the average and worst case data value loss of privacy for all nodes with different number of nodes (n) respectively. We make a few interesting observations. First, the anonymous starting scheme has the same average LoP as the naive protocol but avoids the worst case scenario. This can be seen in Figure 10(b) where the naive protocol suffers a loss of privacy close to 100% (at the starting node) while the anonymous protocol does not change significantly from the average case in Figure 10(a). Second, the probabilistic protocol achieves significantly better privacy than the naive protocols. It is close to 0 in most cases. Finally, all of the protocols have a decreasing loss of privacy as the number of nodes increases. Interestingly, when the number of nodes is sufficiently large, the anonymous naive protocol performs reasonably well compared to the probabilistic protocol. However, most of the privacy preserving data integration will be among tens or hundreds of nodes with membership controls. A network of size on the order of thousands seldom happens.

5.4 Precision of Top-k Selection

We have presented results for max protocol so far. Now we evaluate the general top$k$ protocol in terms of its correctness and privacy characteristics. In addition to running the same set of experiments for max protocol, we also run a set of experiments with varying k. Since most of the results we obtained are similar to those for max protocol, we only report in these two subsections the results for general top$k$ protocol with varying k to show the effect of k.

We first verify the correctness of top$k$ protocol. In order to evaluate the precision of top-k selection, we first define the precision metric we use. Assume $TopK$ is the real set of top-k values and R is the set of top-k values returned. We define the precision as $|R \cap TopK|/K$.

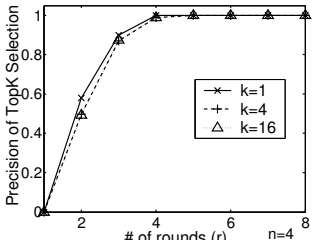
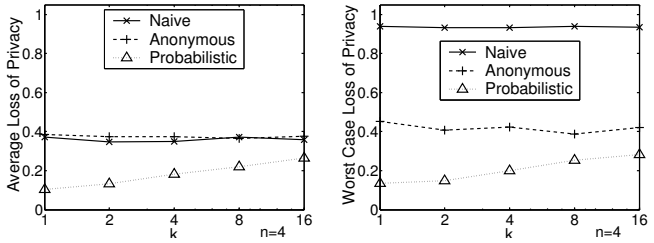


Figure 11: Precision of Top$k$ Selection with Number of Rounds

Figure 11 shows the precision of the top$k$ protocol with increasing number of rounds (r) for varying k. The precision reaches to 100% in all lines after a sufficient number of rounds. The effect of k on the convergence is not significant. We also ran experiments for varying n with $k > 1$ and the result did not show any significant effect.

5.5 Privacy Characteristics of Top-k Selection

Now we report the loss of privacy for the general top$k$ protocol with varying k and its comparison to the naive protocol.



(a) (b)

Figure 12: Comparison of Loss of Privacy with k

Figure 12(a) and (b) show the average and worst case data value loss of privacy for all nodes with varying k. We can make a few interesting observations. We see that the probabilistic protocol achieves significantly better privacy than the naive protocols. Interestingly, the probabilistic protocol has an increasing loss of privacy as k increases. An intuitive explanation is that the larger the k, the more information a node exposes to its successor and hence the larger the loss of privacy.

6 Related Work

Privacy related problems in databases have been an active and important research area. Research in secure databases, Hippocratic databases and privacy policy driven systems [12, 4, 2] has been focused on enabling access of sensitive information through centralized role-based access control. More recently research has been done in areas such as privacy preserving data mining, privacy preserving query processing on

outsourced databases, and privacy preserving information integration.

In privacy preserving data mining [17], the main approach is to use data perturbation techniques to hide precise information in individual data records, as the primary task in data mining is the development of models and patterns about aggregated data. In database outsourcing scenarios, the main technique is to use data partitioning to evaluate obfuscated range queries with minimal information leakage [10, 11]. These techniques may not apply to information integration tasks where precise results are desired.

In the information integration domain, Agrawal et al. [3] introduced the paradigm of minimal information sharing for privacy preserving information integration. Under this paradigm, a few specialized protocols have been proposed, typically in a two party setting, e.g., for finding intersections [3], and kth ranked element [1]. Though still based on cryptographic primitives, they achieve better efficiency than traditional multi-party secure computation methods by allowing minimal information disclosure. As a contrast, our protocol does not require any cryptographic operations. It leverages the multi-party network and utilizes a probabilistic scheme to achieve minimal information disclosure and minimal overhead.

Another related area is the anonymous network where the requirement is that the identity of a user be masked from an adversary. There have been a number of application specific protocols proposed for anonymous communication, including anonymous messaging (Onion Routing [16]), anonymous web transactions (Crowds [14]), anonymous indexing (Privacy Preserving Indexes [5]) and anonymous peer-to-peer systems (Mutual anonymity protocol [19]). Some of these techniques may be applicable for data integration tasks where parties opt to share their information anonymously. However, anonymity is a less strong requirement than data privacy.

7 Conclusion

We have presented a probabilistic protocol for top-k selections across multiple private databases. We formalized the notion of loss of privacy in terms of information revealed and developed an efficient decentralized probabilistic protocol, which aims at selecting top$k$ data items across multiple private databases with minimal information disclosure. We evaluated the correctness and privacy characteristics of the proposed protocol through both formal analysis and experimental evaluations.

Our work on privacy preserving data integration continues along several directions. First, we are extending and generalizing the privacy analysis on the probability distribution of the data using aggregated information from multiple rounds. Second, given the probabilistic scheme, it is possible to design other forms of randomization probability and randomized algorithms. We are interested in conducting a theoretical analysis for discovering the optimal randomized algorithm. Third, we plan to relax the semi-honest model assumption and address the situations where adversaries may not follow the protocol correctly. Finally, we are developing a privacy preserving kNN classifier on top of the top$k$ protocol.

References

[1] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth ranked element. In *IACR Conference on Eurocrypt*, 2004.

[2] R. Agrawal, P. Bird, T. Grandison, J. Kieman, S. Logan, and W. Rjaibi. Extending relational database systems to automatically enforce privacy policies. In *21st ICDE*, 2005.

[3] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *SIGMOD*, 2003.

[4] R. Agrawal, J. Kieman, R. Srikant, and Y. Xu. Hippocratic databases. In *VLDB*, 2002.

[5] M. Bawa, R. J. Bayardo, and R. Agrawal. Privacy-preserving indexing of documents on the network. In *29th VLDB*, 2003.

[6] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *IEEE Conference on Privacy and Security*, 1996.

[7] A. Elmagarmid, M. Rusinkiewicz, and A. Sheth, editors. *Management of Heterogeneous and autonomous database systems*. Morgan Kaufmann; 1st edition, 1999.

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

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

[10] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra. Executing sql over encrypted data in the database service provider model. In *SIGMOD*, 2002.

[11] B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In *ACM Symposium on Principles of Distributed Computing*, 1997.

[12] S. Jajodia and R. Sandhu. Toward a multilevel secure relational data model. In *ACM SIGMOD*, 1991.

[13] MathWorld. Harmonic number.

[14] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1), 1998.

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

[16] S. Syverson, D. M. Coldsehlag, and M. C. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, 1997.

[17] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 33(1), 2004.

[18] M. Wright, M. Adler, B. N. Levine, and C. Shields. Defending anonymous communications against passive logging attacks. In *IEEE Symposium on Security and Privacy*, 2003.

[19] L. Xiao, Z. Xu, and X. Zhang. Mutual anonymity protocols for hybrid peer-to-peer systems. In *ICDCS*, 2003.

[20] L. Xiong and L. Liu. Peertrust: supporting reputation-based trust in peer-to-peer communities. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(7), 2004.