Centralized and Distributed Anonymization for High-Dimensional Healthcare Data

NOMAN MOHAMMED and BENJAMIN C. M. FUNG
Concordia University
PATRICK C. K. HUNG
University of Ontario Institute of Technology
and
CHEUK-KWONG LEE
Hong Kong Red Cross Blood Transfusion Service

Sharing healthcare data has become a vital requirement in healthcare system management; however, inappropriate sharing and usage of healthcare data could threaten patients' privacy. In this article, we study the privacy concerns of sharing patient information between the Hong Kong Red Cross Blood Transfusion Service (BTS) and the public hospitals. We generalize their information and privacy requirements to the problems of *centralized anonymization* and *distributed anonymization*, and identify the major challenges that make traditional data anonymization methods not applicable. Furthermore, we propose a new privacy model called *LKC-privacy* to overcome the challenges and present two anonymization algorithms to achieve LKC-privacy in both the centralized and the distributed scenarios. Experiments on real-life data demonstrate that our anonymization algorithms can effectively retain the essential information in anonymous data for data analysis and is scalable for anonymizing large datasets.

Categories and Subject Descriptors: H.2.7 [**Database Management**]: Database Administration—*security, integrity, and protection*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms, Performance, Security

Additional Key Words and Phrases: Privacy, anonymity, classification, healthcare

ACM Reference Format:
Mohammed, N., Fung, B. C. M., Hung, P. C. K., and Lee, C.-K. 2010. Centralized and distributed anonymization for high-dimensional healthcare data. ACM Trans. Knowl. Discov. Data. 4, 4, Article 18 (October 2010), 33 pages. DOI = 10.1145/1857947.1857950.
http://doi.acm.org/10.1145/1857947.1857950.

The research is supported in part by Discovery Grants (356065-2008) and Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada.

Author's address: N. Mohammed (corresponding author), Concordia University, Montreal, QC, Canada H3G 1M8; email: no_moham@cse.concordia.ca.

1. INTRODUCTION

Gaining access to high-quality health data is a vital requirement to informed decision making for medical practitioners and pharmaceutical researchers. Driven by mutual benefits and regulations, there is a demand for healthcare institutes to share patient data with various parties for research purposes. However, health data in its raw form often contains sensitive information about individuals, and publishing such data will violate their privacy. The current practice in data sharing primarily relies on policies and guidelines on the types of data that can be shared and agreements on the use of shared data. This approach alone may lead to excessive data distortion or insufficient protection. In this paper, we study the challenges in a real-life information-sharing scenario with the Hong Kong Red Cross Blood Transfusion Service (BTS) and propose a new privacy model, in conjunction with data anonymization algorithms, to effectively preserve individuals' privacy and meet the information requirements specified by the BTS.

Figure 1 illustrates the data flow in the BTS. After collecting and examining the blood collected from donors, the BTS distributes the blood to different public hospitals. The hospitals collect and maintain the health records of their patients and transfuse the blood to the patients if necessary. The blood transfusion information, such as the patient data, type of surgery, names of medical practitioners in charge, and reason for transfusion, is clearly documented and is stored in the database owned by each individual hospital. Periodically, the public hospitals are required to submit the blood usage data, together with the patient-specific surgery data, to the BTS for the purpose of data analysis. Hospitals transfer their data to BTS in two ways. Sometimes, hospitals begin by transferring their data to the central government health agency. The agency then integrates the data from different hospitals and gives it to the BTS for data analysis. At other times, hospitals directly submit their data to BTS. These information sharing scenarios in BTS illustrate a typical dilemma in information sharing and privacy protection faced by many health institutes. For example, licensed hospitals in California are also required to submit specific demographic data on every discharged patient [Carlisle et al. 2007] which can provide a multitude of privacy concerns outside of the realm of health care. Our proposed solutions, designed for the BTS case, will also benefit other health institutes that face similar challenges in information sharing. We summarize the privacy concerns and the information needs of the BTS case as follows.

Privacy Concern. Giving the BTS access to blood transfusion data for data analysis is clearly legitimate. However, it raises some concerns on patients' privacy. The patients are willing to submit their data to a hospital because they consider the hospital to be a safe and trustworthy entity. Yet, the trust in the hospital may not necessarily be transitive to a third party. Many agencies and institutes consider that the released data is privacy-preserved if explicit identifying information, such as name, social security number, address, and telephone number, are removed. However, substantial research has shown that simply removing explicit identifying information is insufficient for privacy protection. Sweeney [2002] showed that an individual can be reidentified by

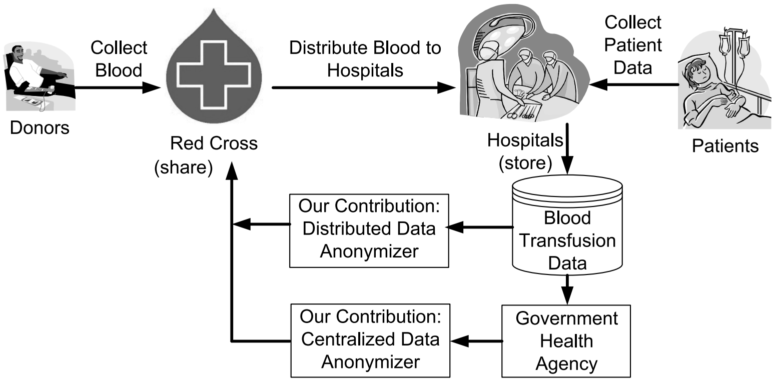


Fig. 1. Data flow in Hong Kong Red Cross Blood Transfusion Service (BTS).

simply matching other attributes, called *quasi-identifiers* (QID), such as gender, date of birth, and postal code.

Information Needs. The BTS wants to perform two types of data analysis on the blood transfusion data collected from the hospitals. First, it wants to obtain some general count statistics. Second, it wants to employ the surgery information as training data for building a classification model on blood transfusion. One frequently raised question is why the hospital does not simply release the statistical data or a classifier to the BTS in order to avoid this privacy concern. The BTS wants to have access to the blood transfusion data, not statistics, from the hospitals for several reasons. First, the practitioners in hospitals have no resources and knowledge in performing data mining. They simply want to share the patient data with the BTS, who needs the health data for legitimate reasons. Second, having access to the data, the BTS has much better flexibility to perform the required data analysis. It is impractical to continuously request practitioners in hospitals to produce different types of statistical information and fine-tune the data mining results for research purposes.

The problems with this BTS case can be generalized into two scenarios. In the first scenario, there exists a trustworthy entity such as the central government health agency to collect the raw patient data from multiple hospitals and submit the data to BTS after performing the centralized anonymization. In the second scenario, the hospitals have to directly submit the integration of their data to the BTS while protecting the patients' privacy. In the following, we explain the privacy threats and challenges of each of the scenarios by an example.

1.1 Centralized Anonymization

Example 1. Consider the integrated raw patient data in Table I (ignore Parties A, B, and C for now), where each record represents a surgery case with

Table I. Raw Patient Data

		Quasi-identifier (QID)			*Class*	*Sensitive*
	ID	**Job**	**Sex**	**Age**	**Transfuse**	**Surgery**
Party A	1	Janitor	M	34	Y	Transgender
	2	Lawyer	F	58	N	Plastic
	3	Mover	M	58	N	Urology
Party B	4	Lawyer	M	24	N	Vascular
	5	Mover	M	34	Y	Transgender
	6	Janitor	M	44	Y	Plastic
	7	Doctor	F	44	N	Vascular
Party C	8	Doctor	M	58	N	Plastic
	9	Doctor	M	24	N	Urology
	10	Carpenter	F	63	Y	Vascular
	11	Technician	F	63	Y	Plastic

the patient-specific information. *Job*, *Sex*, and *Age* are quasi-identifying attributes. Hospitals want to release Table I to the BTS for the purpose of classification analysis on the class attribute, *Transfuse*, which has two values, *Y* and *N*, indicating whether or not the patient has received blood transfusion. Without a loss of generality, we assume that the only sensitive value in *Surgery* is *Transgender*. Hospitals express concern on two types of privacy threats.

—*Identity linkage*. If a record in the table is so specific that not many patients match it, releasing the data may lead to linking the patient's record and, therefore, her received surgery. Suppose that the adversary knows that the target patient is a *Mover* and his age is 34. Hence, record #5, together with his sensitive value (*Transgender* in this case), can be uniquely identified since he is the only *Mover* who is 34 years old in the raw data.

—*Attribute linkage*. If a sensitive value occurs frequently together with some *QID* attributes, then the sensitive information can be inferred from such attributes even though the exact record of the patient cannot be identified. Suppose the adversary knows that the patient is a male of age 34. Even though there exist two such records (#1 and #5), the adversary can infer that the patient has received a *Transgender* surgery with 100% confidence since both the records contain *Transgender*.

High-Dimensionality. Many privacy models, such as *K*-anonymity [Samarati 2001; Sweeney 2002] and its extensions [Machanavajjhala et al. 2007; Wang et al. 2007], have been proposed to thwart privacy threats caused by identity and attribute linkages in the context of relational databases. The usual approach is to generalize the records into equivalence groups so that each group contains at least *K* records with respect to some *QID* attributes, and the sensitive values in each *QID* group are diversified enough to disorient confident inferences. However, Aggarwal [2005] has shown that when the number of *QID* attributes is large, that is, when the dimensionality of data is high, most of the data have to be suppressed in order to achieve *K*-anonymity, resulting in poor data quality for data analysis. Our experiments confirm this *curse of high-dimensionality on K-anonymity* [Aggarwal 2005]. In order to overcome this bottleneck, we exploit one of the limitations of an adversary.

Table II. Anonymous Data (L = 2, K = 2, C = 0.5, S = {*Transgender*})

	Quasi-identifier (QID)			*Class*	*Sensitive*
ID	**Job**	**Sex**	**Age**	**Transfuse**	**Surgery**
1	Nontechnical	M	[30 – 60)	Y	Transgender
2	Professional	F	[30 – 60)	N	Plastic
3	Nontechnical	M	[30 – 60)	N	Urology
4	Professional	M	[1 – 30)	N	Vascular
5	Nontechnical	M	[30 – 60)	Y	Transgender
6	Nontechnical	M	[30 – 60)	Y	Plastic
7	Professional	F	[30 – 60)	N	Vascular
8	Professional	M	[30 – 60)	N	Plastic
9	Professional	M	[1 – 30)	N	Urology
10	Technical	F	[60 – 99)	Y	Vascular
11	Technical	F	[60 – 99)	Y	Plastic

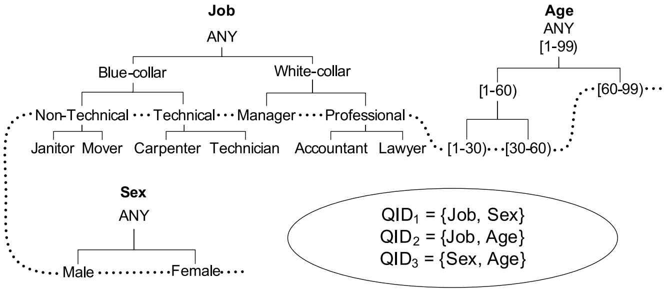


Fig. 2. Taxonomy trees and QIDs.

In real-life privacy attacks, it is very difficult for an adversary to acquire all the QID information of a target patient because it requires nontrivial effort to gather each piece of prior knowledge from so many possible values. Thus, it is reasonable to assume that the adversary's prior knowledge is bounded by at most L values of the QID attributes of the patient. Based on this assumption, we define a new privacy model called LKC*-privacy* for anonymizing high-dimensional data.

The general intuition of LKC-privacy is to ensure that every combination of values in $QID_j \subseteq QID$ with maximum length L in the data table T is shared by at least K records, and the confidence of inferring any sensitive values in S is not greater than C, where L, K, C are thresholds and S is a set of sensitive values specified by the data holder (the hospital). LKC-privacy bounds the probability of a successful identity linkage to be $\leq 1/K$ and the probability of a successful attribute linkage to be $\leq C$, provided that the adversary's prior knowledge does not exceed L. Table II shows an example of an anonymous table that satisfies $(2, 2, 50\%)$-privacy with S = {*Transgender*} by generalizing all the values from Table I according to the taxonomies in Figure 2 (Ignore the dashed curve for now). Every possible value of QID_j with maximum length 2 in Table II (namely, QID_1, QID_2, and QID_3 in Figure 2) is shared by at least 2 records, and the confidence of inferring the sensitive value *Transgender* is not greater than 50%. In contrast, enforcing traditional 2-anonymity will require

Table III. Distributed Anonymization (L = 2, K = 2, C = 0.5, S = {*Transgender*})

	Quasi-identifier (QID)			*Class*	*Sensitive*
ID	**Job**	**Sex**	**Age**	**Transfuse**	**Surgery**
1	ANY	ANY	[30 – 60)	Y	Transgender
2	ANY	ANY	[30 – 60)	N	Plastic
3	ANY	ANY	[30 – 60)	N	Urology
4	Professional	ANY	[1 – 60)	N	Vascular
5	Nontechnical	M	[30 – 60)	Y	Transgender
6	Nontechnical	M	[30 – 60)	Y	Plastic
7	Professional	ANY	[1 – 60)	N	Vascular
8	Professional	M	[1 – 60)	N	Plastic
9	Professional	M	[1 – 60)	N	Urology
10	Technical	F	[60 – 99)	Y	Vascular
11	Technical	F	[60 – 99)	Y	Plastic

further generalization. For example, in order to make $\langle Professional, M, [30 - 60)\rangle$ to satisfy traditional 2-anonymity, we may further generalize all instances of $[1 - 30)$ and $[30 - 60)$ to $[1 - 60)$, resulting in much higher utility loss.

1.2 Distributed Anonymization

The centralized anonymization method can be viewed as "integrate-then-generalize" approach, where the central government health agency first integrates the data from different hospitals then performs generalization. In real-life information sharing, a trustworthy central authority may not always exist. Sometimes, it is more flexible for the data recipient to make requests to the data holders, and the data holders directly send the requested data to the recipient. For example, in some special occasions and events, BTS has to directly collect data from the hospitals without going through the government health agency.

In this distributed scenario, each hospital owns a set of raw patient data records. The data can be viewed as horizontally partitioned among the data holders over the same set of attributes. Consider the raw patient data in Table I, where records 1–3 are from Party A, records 4–7 are from Party B, and records 8–11 are from Party C. To achieve distributed anonymization, a naïve approach is to anonymize the patient data independently by the hospitals and then integrate as shown in Table III. However, such a distributed "generalize-then-integrate" approach suffers significant utility loss compared to the centralized "integrate-then-generalize" approach as shown in Table II.

The distributed anonymization problem has two major challenges in addition to high dimensionality. First, the data utility of the anonymous integrated data should be as good as the data quality produced by the centralized anonymization algorithm. Second, in the process of anonymization, the algorithm should not reveal more specific information than the final anonymous integrated table. In Section 5, we propose a distributed anonymization algorithm that incorporates secure multiparty computation (SMC) techniques to achieve the same data utility as centralized approach while ensuring the privacy requirements.

Contributions. The contributions of this article are summarized as follows.

(1) We use the Red Cross BTS as a real-life example to present the challenges of privacy-aware information sharing for data analysis and define the problems of centralized and distributed anonymization in the context of BTS.

(2) We propose a new privacy model called *LKC-privacy* that overcomes the challenges of anonymizing high-dimensional relational data without significantly compromising the data quality (Section 3).

(3) We present two anonymization algorithms to address the problems of centralized anonymization and distributed anonymization. Both the algorithms achieve LKC-privacy with two different adaptations. The first adaptation maximizes the information preserved for classification analysis; the second one minimizes the distortion on the anonymous data for general data analysis. Minimizing distortion is useful when the particular information requirement is unknown during information sharing or the shared data is used for various kinds of data mining tasks (Sections 4 and 5).

(4) We implement the proposed algorithms and evaluate the performance. Experiments on real-life data demonstrate that our developed algorithm is flexible and scalable enough to handle large volumes of blood transfusion data that include both categorical and numerical attributes. Scalability is an important requirement in the BTS project, for example, the BTS received about 150,000 records from the public hospitals in 2008 (Section 6).

2. RELATED WORK

Data privacy has been an active area of research in statistics, database, and security communities for the last two decades [Adam and Wortman 1989; Fung et al. 2010]. In this section, we briefly present various privacy-preserving techniques for both single and multiple data holders scenarios and explain why the existing techniques are not applicable to the problem studied in this article.

2.1 Privacy-Preserving Methods for Single Data Holder

There is a large body of work on anonymizing relational data. Traditional K-anonymity [Samarati 2001; Sweeney 2002], ℓ-diversity [Machanavajjhala et al. 2007], and confidence bounding [Wang et al. 2007] are based on a predefined set of QID attributes. (α, k)-anonymity [Wong et al. 2006] further requires every QID group to satisfy both K-anonymity and confidence bounding. As discussed earlier, these single QID-based approaches suffer from the curse of high dimensionality [Aggarwal 2005] and render the high-dimensional data useless for data mining. Xiao and Tao [2006a] propose the notion of personalized privacy to allow each record owner to specify her own privacy level. This model assumes that a sensitive attribute has a taxonomy tree and that each record owner specifies a guarding node in the taxonomy tree. Dwork [2006] proposes a privacy model called *differential privacy*, which ensures that the removal or addition of a single record does not significantly affect the overall privacy of the database. Most of the works in differential privacy are based on the interactive privacy model, where the result of a query is in the form of aggregation [Blum et al. 2005; Dinur and Nissim 2003; Dwork et al. 2006].

There are some recent works on anonymizing transaction data that model the adversary's power by a maximum number of known items as prior knowledge [Ghinita et al. 2008; Terrovitis et al. 2008; Xu et al. 2008]. Although the assumption is similar, our studied problem is different from these other works. First, a transaction is a *set* of items, whereas the health data is relational with predefined taxonomy trees. Second, we have different privacy and data utility requirements. The privacy model of Terrovitis et al. [2008] is based on only K-anonymity and does not consider attribute linkages. Xu et al. [2008] aim at minimizing data distortion while we aim at preserving classification quality. Finally, Xu et al. [2008] use suppression, while we use generalization and discretization for anonymizing various types of attributes.

There are many different techniques which can be used to achieve a privacy model. Perturbation-based techniques achieve privacy by adding noises, randomizing data, and generating synthetic data. Perturbed data is useful at the aggregated level (such as average or sum), but not at the record level [Agrawal and Srikant 2000; Fuller 1993; Kim and Winkler 1995]. Data recipients can no longer interpret the semantic of individual record, which is important in some knowledge exploration tasks, such as visual data mining [Zhao et al. 2005]. Instead of perturbing the data, we generalize the data to make information less precise while preserving the "truthfulness" of information (say, generalizing *Lawyer* to *Professional*). Generalized data is meaningful at the record level and, therefore, can be utilized to guide the search or interpret the result.

Unlike generalization, Xiao and Tao [2006b] propose a very different approach, called *anatomy*, that does not modify the QID and the sensitive attribute (SA), but de-associates the relationship between the two. However, it disguises the correlation between SA and other attributes, and therefore, hinders data analysis that depends on such correlation. For example, the sensitive attribute is important for classification analysis at BTS. Disguising the correlation between the class attribute and SA defeats the purpose of releasing the SA.

Many techniques have been previously proposed to preserve privacy, but only a few have considered the goal for classification. Iyengar [2002] presents the anonymity problem for classification and proposes a genetic algorithmic solution. Bayardo and Agrawal [2005] also address the classification problem using the same classification metric as Iyengar [2002]. Unlike random genetic evolution, our approach produces a *progressive* generalization process that users can step through to determine a desired trade-off of privacy and accuracy. Recently, LeFevre et al. [2008] proposed another anonymization technique for classification using multidimensional recoding [LeFevre et al. 2006]. Multidimensional recoding allows a value v to be independently generalized into different parent values. Mining classification rules from multidimensional recoded data may result in ambiguous classification rules, for example, White-collar → Class A and Lawyer → Class B. Moreover, all the proposed models for classification analysis do not address the problems of high dimensionality and distributed anonymization, which are primary contribution of this paper.

2.2 Privacy-Preserving Methods for Multiple Data Holders

The primary goal of our study in this paper is to share data. In contrast, techniques based on secure multiparty computation (SMC) allow sharing of the computed result (e.g., a classifier), but completely prohibit sharing data. An example is the secure multiparty computation of classifiers [Du et al. 2004; Du and Zhan 2002; Yang et al. 2005]. Yang et al. [2005] propose a cryptographic approach to acquire classification rules from a large number of data holders while their sensitive attributes are protected. Vaidya and Clifton [2002; 2003] propose privacy-preserving techniques to mine association rules and to compute k-means clustering in a distributed setting without sharing data. When compared to data mining results sharing, data sharing offers more freedom to the data recipients to apply their own classifiers and parameters.

Jiang and Clifton [2005] propose an approach to integrate data by maintaining k-anonymity among the participating parties. However, this approach does not fulfill the security requirements of a semihonest adversary model. To satisfy this requirement, Jiang and Clifton [2006] further propose a cryptographic approach to securely integrate two distributed data tables to form an integrated k-anonymous table, without considering a data mining task. Mohammed et al. [2009b] propose an anonymization algorithm for vertically partitioned data from multiple data holders without disclosing data from one party to another. Unlike the distributed anonymization problem for horizontally partitioned data studied in this article, all these methods [Jiang and Clifton 2005, 2006; Mohammed et al. 2009b] generate a k-anonymous table in a distributed setting for vertically partitioned data. Recently, Jurczyk and Xiong [2009] proposed an algorithm to integrate data for horizontally partitioned data. To the best of our knowledge, this is the only work that generates a k-anonymous table in a distributed setting for horizontally partitioned data. However, their anonymization model does not take into consideration the information requirement for classification analysis, which is the ultimate purpose of data sharing in our context. Moreover, our model addresses the issue of anonymizing high-dimensional data.

This article is the extension of our previous work [Mohammed et al. 2009a] which only addressed the centralized anonymization algorithm for the BTS. In this article, we formally define the distributed anonymization problem and propose a distributed anonymization algorithm which can be used by hospitals to anonymize their data distributively. This algorithm does not require a trusted third party (the government health agency) for integrating the data from different data holders before anonymization. To perform the anonymization process distributively, we incorporate secure multiparty techniques with our anonymization algorithm.

3. PROBLEM DEFINITION

We first describe the privacy and information requirements, followed by the problem statement.

3.1 Privacy Measure

Suppose a data holder (e.g., the central government health agency) wants to publish a health data table $T(ID, D_1, \ldots, D_m, \mathit{Class}, \mathit{Sens})$ (e.g., Table I) to some recipient (e.g., the Red Cross BTS) for data analysis. ID is an explicit identifier, such as *SSN*, and it should be removed before publication. We keep the ID in our examples for discussion purpose only. Each D_i is either a categorical or a numerical attribute. *Sens* is a sensitive attribute. A record has the form $\langle v_1, \ldots, v_m, \mathit{cls}, s\rangle$, where v_i is a domain value of D_i, *cls* is a class value of *Class*, and s is a sensitive value of *Sens*. The data holder wants to protect against linking an individual to a record or some sensitive value in T through some subset of attributes called a *quasi-identifier* or QID, where $QID \subseteq \{D_1, \ldots, D_m\}$.

One recipient, who is an adversary, seeks to identify the record or sensitive values of some target victim patient V in T. As explained in Section 1, we assume that the adversary knows at most L values of QID attributes of the victim patient. We use *qid* to denote such prior known values, where $|qid| \leq L$. Based on the prior knowledge *qid*, the adversary could identify a group of records, denoted by $T[qid]$, that contains *qid*. $|T[qid]|$ denotes the number of records in $T[qid]$. For example, $T[\langle \mathit{Janitor}, M\rangle] = \{ID\#1, 6\}$ and $|T[qid]| = 2$. Then, the adversary could launch two types of privacy attacks:

(1) *Identity linkage*. Given prior knowledge *qid*, $T[qid]$ is a set of candidate records that contains the victim patient V's record. If the group size of $T[qid]$, denoted by $|T[qid]|$, is small, then the adversary may identify V's record from $T[qid]$ and, therefore, V's sensitive value. For example, if $qid = \langle \mathit{Mover}, 34\rangle$ in Table I, $T[qid] = \{ID\#5\}$. Thus, the adversary can easily infer that V has received a *Transgender* surgery.

(2) *Attribute linkage*. Given prior knowledge *qid*, the adversary can identify $T[qid]$ and infer that V has sensitive value s with confidence $P(s|qid) = \frac{|T[qid \wedge s]|}{|T[qid]|}$, where $T[qid \wedge s]$ denotes the set of records containing both *qid* and s. $P(s|qid)$ is the percentage of the records in $T[qid]$ containing s. The privacy of V is at risk if $P(s|qid)$ is high. For example, given $qid = \langle M, 34\rangle$ in Table I, $T[qid \wedge \mathit{Transgender}] = \{ID\#1, 5\}$ and $T[qid] = \{ID\#1, 5\}$, hence $P(\mathit{Transgender}|qid) = 2/2 = 100\%$.

To thwart the identity and attribute linkages on *any* patient in the table T, we require every *qid* with a maximum length L in the anonymous table to be shared by at least a certain number of records, and the ratio of sensitive value(s) in every group cannot be too high. Our privacy model, *LKC-privacy*, reflects this intuition.

Definition 3.1 *LKC-privacy*. Let L be the maximum number of values of the prior knowledge. Let $S \subseteq \mathit{Sens}$ be a set of sensitive values. A data table T satisfies *LKC-privacy* if and only if for any *qid* with $|qid| \leq L$,

(1) $|T[qid]| \geq K$, where $K > 0$ is an integer anonymity threshold, and

(2) $P(s|qid) \leq C$ for any $s \in S$, where $0 < C \leq 1$ is a real number confidence threshold. Sometimes, we write C in percentage.

The data holder specifies the thresholds L, K, and C. The maximum length L reflects the assumption of the adversary's power. LKC-privacy guarantees that the probability of a successful identity linkage to be $\leq 1/K$ and the probability of a successful attribute linkage to be $\leq C$. LKC-privacy has several nice properties that make it suitable for anonymizing high-dimensional data. First, it only requires a subset of QID attributes to be shared by at least K records. This is a major relaxation from traditional K-anonymity, based on a very reasonable assumption that the adversary has limited power. Second, LKC-privacy generalizes several traditional privacy models. K-anonymity [Samarati 2001; Sweeney 2002] is a special case of LKC-privacy with $L = |QID|$ and $C = 100\%$, where $|QID|$ is the number of QID attributes in the data table. Confidence bounding [Wang et al. 2007] is also a special case of LKC-privacy with $L = |QID|$ and $K = 1$. (α, k)-anonymity [Wong et al. 2006] is also a special case of LKC-privacy with $L = |QID|$, $K = k$, and $C = \alpha$. Thus, the data holder can still achieve the traditional models, if needed.

3.2 Utility Measure

The measure of data utility varies depending on the data analysis task to be performed on the published data. Based on the information requirements specified by the BTS, we define two utility measures. First, we aim at preserving the maximal information for classification analysis. Second, we aim at minimizing the overall data distortion when the data analysis task is unknown.

In this BTS project, we propose a top-down specialization algorithm to achieve LKC-privacy. The general idea is to anonymize a table by a sequence of specializations starting from the topmost general state in which each attribute has the topmost value of its taxonomy tree [Fung et al. 2007]. We assume that a *taxonomy tree* is specified for each categorical attribute in QID. A leaf node represents a domain value and a parent node represents a less specific value. For a numerical attribute in QID, a taxonomy tree can be grown at runtime, where each node represents an interval, and each nonleaf node has two child nodes representing some optimal binary split of the parent interval. Figure 2 shows a dynamically grown taxonomy tree for *Age*.

A *specialization*, written $v \rightarrow child(v)$, where $child(v)$ denotes the set of child values of v, replaces the parent value v with the child value that generalizes the domain value in a record. A specialization is *valid* if the specialization results in a table satisfying the anonymity requirement after the specialization. A specialization is performed only if it is valid. The specialization process can be viewed as pushing the "cut" of each taxonomy tree downwards. A *cut* of the taxonomy tree for an attribute D_i, denoted by Cut_i, contains exactly one value on each root-to-leaf path. Figure 2 shows a solution cut indicated by the dashed curve representing the anonymous Table II. Our specialization starts from the topmost cut and pushes down the cut iteratively by specializing some value in the current cut until violating the anonymity requirement. In other words, the specialization process pushes the cut downwards until no valid specialization is possible. Each specialization tends to increase data utility and decrease privacy because records are more distinguishable by specific values. We define two utility measures depending on the information requirement to evaluate the "goodness" of a specialization. Here, we assume that BTS only receives one version of the sanitized data for a given dataset anonymized by using one of the following *Score* functions.

3.2.1 *Case 1: Score for Classification Analysis.* For the requirement of classification analysis, we use information gain, denoted by $InfoGain(v)$, to measure the *goodness* of a specialization. Our selection criterion, $Score(v)$, is to favor the specialization $v \rightarrow child(v)$ that has the maximum $InfoGain(v)$:

$$Score(v) = InfoGain(v). \tag{1}$$

InfoGain(v): Let $T[x]$ denote the set of records in T generalized to the value x. Let $freq(T[x], cls)$ denote the number of records in $T[x]$ having the class cls. Note that $|T[v]| = \sum_c |T[c]|$, where $c \in child(v)$. We have

$$InfoGain(v) = E(T[v]) - \sum_c \frac{|T[c]|}{|T[v]|} E(T[c]), \tag{2}$$

where $E(T[x])$ is the *entropy* of $T[x]$ [Quinlan 1993]:

$$E(T[x]) = -\sum_{cls} \frac{freq(T[x], cls)}{|T[x]|} \times log_2 \frac{freq(T[x], cls)}{|T[x]|}, \tag{3}$$

Intuitively, $I(T[x])$ measures the mix of classes for the records in $T[x]$, and $InfoGain(v)$ is the reduction of the mix by specializing v into $c \in child(v)$.

For a numerical attribute, the specialization of an interval refers to the optimal binary split that maximizes information gain on the *Class* attribute. See [Quinlan 1993] for details.

3.2.2 *Case 2: Score for General Data Analysis.* Sometimes, the data is shared without a specific task. In this case of general data analysis, we use discernibility cost [Skowron and Rauszer 1992] to measure the data distortion in the anonymous data table. The discernibility cost charges a penalty to each record for being indistinguishable from other records. For each record in an equivalence group qid, the penalty is $|T[qid]|$. Thus, the penalty on a group is $|T[qid]|^2$. To minimize the discernibility cost, we choose the specialization $v \rightarrow child(v)$ that maximizes the value of

$$Score(v) = \sum_{qid_v} |T[qid_v]|^2 \tag{4}$$

over all qid_v containing v. Example 3 shows the computation of $Score(v)$.

3.3 Problem Statement

We generalize the problems faced by BTS to the problems of centralized anonymization and distributed anonymization. The problem of centralized anonymization models the scenario of the central government health agency that anonymizes the integrated data before transferring it to BTS. The problem of distributed anonymization results from the scenario of the hospitals that distributively anonymize the data without the need of the central government health agency.

3.3.1 *Centralized Anonymization for Data Analysis.* The problem of centralized anonymization is to transform a given dataset T into an anonymous version T' that satisfies a given LKC*-privacy* requirement and preserves as much information as possible for the intended data analysis task. Based on the information requirements specified by the BTS, we define the problems as follows.

Definition 3.2 *Centralized Anonymization for data analysis*. Given a data table T, a LKC*-privacy* requirement, and a taxonomy tree for each categorical attribute contained in QID, the problem of *centralized anonymization for data analysis* is to generalize T on the attributes QID to satisfy the LKC-privacy requirement while preserving as much information as possible for data analysis.

3.3.2 *Distributed Anonymization for Data Analysis.* Consider n hospitals {Party 1,...,Party n}, where each Party i owns a private table $T_i(ID, D_1, \ldots, D_m, Class, Sens)$ over the same set of attributes. Each hospital owns a disjoint set of records, where $record_i \cap record_j = \emptyset$ for any $1 \leq i, j \leq n$. These parties are required to form an integrated table T for conducting a joint data analysis. The anonymization process is required to ensure two different privacy requirements: privacy for data subjects (patients) and privacy for data holders (hospitals).

To protect privacy for data subjects, we require the integrated data to satisfy a given LKC-privacy requirement. However, the integrated data is less anonymous to the data holders (hospitals) because a data holder can always remove his own data records from the integrated data and make the remaining data less anonymous than the LKC-privacy requirement. To protect privacy for data holders, we require that hospitals should not share more detailed information than the final integrated data table during the distributed anonymization process.

Definition 3.3 *Distributed Anonymization for data analysis*. When given multiple private tables $T_1, \ldots, T_n$, where each T_i is owned by different Party i, a LKC*-privacy* requirement, and a taxonomy tree for each categorical attribute contained in QID, the problem of *distributed anonymization for data analysis* is to efficiently produce a generalized integrated table T such that (1) T satisfies the LKC*-privacy* requirement, (2) T contains as much information as possible for data analysis, and (3) each party learns nothing about the other party more specific than what is in the final generalized integrated table T.

The requirement (3) in Definition 3.3 requires that each party should not reveal any additional information to other parties than what is in the final integrated table. This requirement is similar to the secure multiparty computation (SMC) protocols, where no participant learns more information than the outcome of a function. In the problem of distributed anonymization, we assume that the parties are *semihonest*. In the semihonest adversary model, each

Algorithm 1 Centralized Anonymization Algorithm

```
1: Initialize every value in $T$ to the topmost value;
2: Initialize $Cut_i$ to include the topmost value;
3: while some $x \in \cup Cut_i$ is valid do
4:     Find the $Best$ specialization from $\cup Cut_i$;
5:     Perform $Best$ on $T$ and update $\cup Cut_i$;
6:     Update $Score(x)$ and validity for $x \in \cup Cut_i$;
7: end while
8: Output $T$ and $\cup Cut_i$;
```

party obeys the protocol. However, they may be curious to derive more information from the received messages in the course of the protocol execution. This assumption about participating parties is very common in the SMC problems.

4. CENTRALIZED ANONYMIZATION ALGORITHM

Algorithm 1 provides an overview of our *centralized anonymization algorithm* for achieving LKC-privacy. Initially, all values in QID are generalized to the topmost value in their taxonomy trees, and Cut_i contains the topmost value for each attribute D_i. At each iteration, the algorithm finds the *Best* specialization, which has the highest *Score* among the *candidates* that are valid specializations in $\cup Cut_i$ (Line 4). Then, apply *Best* to T and update $\cup Cut_i$ (Line 5). Finally, update the *Score* of the affected candidates due to the specialization (Line 6). The algorithm is terminated when there are no more valid candidates in $\cup Cut_i$. In other words, the algorithm is terminated if any further specialization would lead to a violation of the LKC-privacy requirement. An important property of Algorithm 1 is that the LKC-privacy is antimonotone with respect to a specialization; if a generalized table violates LKC-privacy before a specialization, it remains violated after the specialization because a specialization never increases the $|T[qid]|$ and never decreases the maximum $P(s|qid)$. This antimonotonic property guarantees that the final solution cut is a suboptimal solution. Algorithm 1 is modified from TDR [Fung et al. 2007], which is originally designed for achieving only K-anonymity, not LKC-privacy. One major difference is the validity check in Line 6, which will be discussed in detail in Section 4.3.

Example 2. Consider the integrated raw patient data in Table I with $L = 2$, $K = 2$, $C = 50\%$, and $QID = \{Job, Sex, Age\}$. Initially, all data records are generalized to $\langle ANY\_Job, ANY\_Sex, [1\text{-}99)\rangle$, and $\cup Cut_i = \{ANY\_Job, ANY\_Sex, [1\text{-}99)\}$. To find the *Best* specialization among the candidates in $\cup Cut_i$, we compute $Score\ (ANY\_Job)$, $Score(ANY\_Sex)$, and $Score([1\text{-}99))$.

A simple yet inefficient implementation of Lines 4–6 is to scan *all* data records and recompute $Score(x)$ for all candidates in $\cup Cut_i$. The key to the efficiency of our algorithm is having *direct access* to the data records to be specialized, and updating $Score(x)$ based on some statistics maintained for candidates in $\cup Cut_i$, instead of scanning all data records. In the rest of this section, we explain our scalable implementation and data structures in detail.

4.1 Find the Best Specialization

Initially, we compute *Score* for all candidates x in $\cup Cut_i$. For each subsequent iteration, information needed to calculate *Score* comes from the update of the previous iteration (Line 6). Finding the best specialization *Best* involves at most $|\cup Cut_i|$ computations of *Score* without accessing data records. The procedure for updating *Score* will be discussed in Section 4.3.

Example 3. Continue from Example 2. We show the computation of $Score(ANY\_Job)$ for the specialization

$$ANY\_Job \rightarrow \{Blue\text{-}collar, White\text{-}collar\}.$$

For classification analysis,
$E(T[ANY\_Job]) = -\frac{6}{11} \times log_2 \frac{6}{11} - \frac{5}{11} \times log_2 \frac{5}{11} = 0.994$
$E(T[Blue\text{-}collar]) = -\frac{1}{6} \times log_2 \frac{1}{6} - \frac{5}{6} \times log_2 \frac{5}{6} = 0.6499$
$E(T[White\text{-}collar]) = -\frac{5}{5} \times log_2 \frac{5}{5} - \frac{0}{5} \times log_2 \frac{0}{5} = 0.0$
$InfoGain(ANY\_Job) = E(T[ANY\_Job]) - (\frac{6}{11} \times$
$E(T[Blue\text{-}collar]) + \frac{5}{11} \times E(T[White\text{-}collar])) = 0.6396$
$Score(ANY\_Job) = InfoGain(ANY\_Job) = 0.6396.$

4.2 Perform the Best Specialization

Consider a specialization $Best \rightarrow child(Best)$, where $Best \in D_i$ and $D_i \in QID$. First, we replace *Best* with $child(Best)$ in $\cup Cut_i$. Then, we need to retrieve $T[Best]$, the set of data records generalized to *Best*, to tell the child value in $child(Best)$ for individual data records. We employ a data structure called *Taxonomy Indexed PartitionS (TIPS)* [Fung et al. 2007] to facilitate this operation. This data structure is also crucial for updating $Score(x)$ for candidates x. The general idea is to group data records according to their generalized records on QID.

Definition 4.1 *TIPS*. TIPS is a tree structure with each root-to-leaf path represents a generalized record over QID. Each leaf node stores the set of data records having the same generalized record for all the QID attributes along the path. Each path is called a *leaf partition*. For each x in $\cup Cut_i$, P_x denotes a leaf partition whose generalized record contains x, and $Link_x$ denotes the link of all P_x, with the head of $Link_x$ stored with x.

At any time, the generalized data is represented by the leaf partitions of TIPS, but the original data records remain unchanged. $Link_x$ provides a direct access to $T[x]$, the set of data records generalized to the value x. Initially, TIPS has only one leaf partition containing all data records, generalized to the topmost value on every attribute in QID. In each iteration, we perform the best specialization *Best* by refining the leaf partitions on $Link_{Best}$.

Updating TIPS. We refine each leaf partition P_{Best} found on $Link_{Best}$ as follows. For each value c in $child(Best)$, a new partition P_c is created from P_{Best}, and data records in P_{Best} are split among the new partitions: P_c contains a data record in P_{Best} if c generalizes the corresponding domain value in the record.

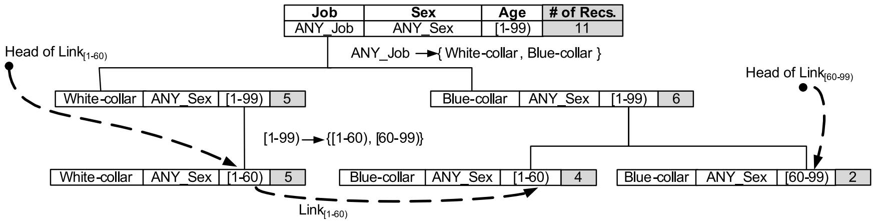


Fig. 3. The TIPS data structure.

An empty P_c is removed. $Link_c$ is created to link up all P_c's for the same c. Also, link P_c to every $Link_x$ to which P_{Best} was previously linked, except for $Link_{Best}$. We emphasize that this is the only operation in the whole algorithm that requires accessing data records. The overhead of maintaining $Link_x$ is small. For each attribute in $\cup QID_j$ and each leaf partition on $Link_{Best}$, there are at most $|child(Best)|$ "relinkings," or at most $| \cup QID_j| \times |Link_{Best}| \times |child(Best)|$ "relinkings" in total for applying *Best*.

Example 4. Initially, TIPS has only one leaf partition containing all data records and representing the generalized record $\langle$*ANY_Job, ANY_Sex, [1-99)*$\rangle$. Let the best specialization be *ANY_Job* → {*White-collar*, *Blue-collar*} on *Job*. We create two new partitions under the root partition as in Figure 3, and split data records between them. Both the leaf partitions are on $Link_{ANY\_Sex}$ and $Link_{[1\text{-}99)}$. $\cup Cut_i$ is updated into {*White-collar*, *Blue-collar*, *ANY_Sex*, *[1-99)*}. Suppose that the next best specialization is *[1-99)* → {*[1-60), [60-99)*}, which specializes the two leaf partitions on $Link_{[1\text{-}99)}$, resulting in the TIPS in Figure 3.

A scalable feature of our algorithm is maintaining some statistical information for each candidate x in $\cup Cut_i$ for updating $Score(x)$ without accessing data records. For each new value c in $child(Best)$ added to $\cup Cut_i$ in the current iteration, we collect the following *count statistics* of c while scanning data records in P_{Best} for updating TIPS: $|T[c]|$, $|T[d]|$, $freq(T[c], cls)$, and $freq(T[d], cls)$, where $d \in child(c)$ and cls is a class label. These information will be used in Section 4.3.

TIPS has several useful properties. First, all data records in the same leaf partition have the same generalized record although they may have different raw values. Second, every data record appears in exactly one leaf partition. Third, each leaf partition P_x has exactly one generalized qid on QID and contributes the count $|P_x|$ towards $|T[qid]|$. Later, we use the last property to extract $|T[qid]|$ from TIPS.

4.3 Update Score and Validity

This step updates $Score(x)$ and validity for candidates x in $\cup Cut_i$ to reflect the impact of the *Best* specialization. The key to the scalability of our algorithm is updating $Score(x)$ using the count statistics maintained in Section 4.2 without accessing raw records again.

4.3.1 *Updating Score.* The procedure for updating *Score* is different depending on the information requirement.

Case 1 Classification Analysis. An observation is that $InfoGain(x)$ is not affected by $Best \rightarrow child(Best)$, except that we need to compute $InfoGain(c)$ for each newly added value c in $child(Best)$. $InfoGain(c)$ can be computed from the count statistics for c collected in Section 4.2.

Case 2 General Data Analysis. Each leaf partition P_c keeps the count $|T[qid_c]|$. By following $Link_c$ from TIPS, we can compute $\sum_{qid_c} |T[qid_c]|^2$ for all the qid_c on $Link_c$.

4.3.2 *Validity Check.* A specialization $Best \rightarrow child(Best)$ may change the validity status of other candidates $x \in \cup Cut_i$ if $Best$ and x are contained in the same qid with size not greater than L. Thus, in order to check the validity, we need to keep track of the count of every qid with $|qid| = L$. Note, we can ignore qid with size less than L because if a table satisfies LKC-privacy, then it must satisfy $L'KC$-privacy where $L' < L$.

We present an efficient method for checking the validity of a candidate. First, given a QID in T, we identify all $QID_j \subseteq QID$ with size L. Then, for each QID_j, we use a data structure, called $QIDTree_j$, to index all qid_j on QID_j. $QIDTree_j$ is a tree, where each level represents one attribute in QID_j. Each root-to-leaf path represents an existing qid_j on QID_j in the generalized data, with $|T[qid_j]|$ and $|T[qid_j \wedge s]|$ for every $s \in S$ stored at the leaf node. A candidate $x \in \cup Cut_i$ is valid if, for every $c \in child(x)$, every qid_j containing c has $|T[qid_j]| \geq K$ and $P(s|qid_j) \leq C$ for any $s \in S$. If x is invalid, remove it from $\cup Cut_i$.

4.4 Analysis

Each iteration involves two types of work: (1) Accessing data records in $T[Best]$ for updating TIPS and count statistics (Section 4.2). (2) Updating $Score(x)$ and validity status for the affected candidates x in $\cup Cut_i$ (Section 4.3). Only the work in (1) involves accessing data records, which is in the order of $O(|T|)$; the work in (2) makes use of the count statistics without accessing data records and can be performed in constant time. This feature makes our approach scalable. We will empirically evaluate the scalability of the algorithm on a real-life dataset in Section 6. For one iteration, the computation cost is $O(|T|)$ and the total number of iterations is bounded by $O(log|T|)$; therefore, the total computation cost is $O(|T|log|T|)$.

5. DISTRIBUTED ANONYMIZATION ALGORITHM

In the section, we extend the centralized anonymization algorithm to address the problem of distributed anonymization for data analysis described in Definition 3.3. Initially the data is located among n different locations. Each Party i (hospital) owns a private database T_i. The union of the local databases constructs the complete view of the data table, $T = \bigcup T_i$, where $1 \leq i \leq n$. Note that the quasi-identifiers are uniform across all the local databases.

As discussed in Section 1.2, if the data holders perform anonymization independently before data integration, then it results in higher utility loss than the centralized approach. To prevent utility loss, parties need to know whether a locally identifiable record will or will not satisfy the privacy requirement *after* integration. Moreover, to satisfy the utility requirement, all the parties should perform the same sequence of anonymization operations. In other words, parties need to calculate the *Score* of the candidates over the integrated data table. To overcome these problems, each party keeps a copy of the current $\cup Cut_i$ and generalized T, denoted by T_g, in addition to the private T_i. The nature of the top-down approach implies that T_g is more general than the final answer, therefore, does not violate the requirement (3) in Definition 3.3. At each iteration, all the parties cooperate to determine the *Best* specialization that has the highest *Score* and perform the same specialization. The exchanged information for determining the *Score* does not violate the requirement (3) in Definition 3.3.

The proposed distributed anonymization algorithm requires one party to act as a leader. It is important to note that any hospital can act as a leader and the leader is not necessarily to be more trustworthy than others. Unlike the centralized approach, hospitals do not share their data with the leader and after the anonymization the data resides with the respective data holders. The only purpose of the leader is to synchronize the anonymization process. Algorithms 2 and 3 describe the algorithms for leader and nonleader parties. Without loss of generality, we assume that Party 1 is the leader in the explanation. The sequence of specialization operations performed by the parties in this distributed anonymization algorithm is the same as the centralized anonymization algorithm. Initially, each party initializes T_g to include one record containing the top most values and $\cup Cut_i$ to include the top most value for each attribute D_i (Lines 1–2 of Algorithms 2 and 3). First, the leader collects all the count statistics from all the parties to determine the *Best* candidate. The count statistics are collected through the propagation of the $Information$ message by using *secure sum protocol* [Schneier 1995] (Lines 3–4 of Algorithms 2 and 3). Secure sum protocol ensures that the leader only knows the global count statistics without the knowledge of the specific individuals' contribution. Once the leader determines the *Best* candidate (Line 6 of Algorithm 2), it informs the other parties through the propagation of the *Instruction* message to specialize the *Best* on T_g (Line 7 of Algorithm 2 and Lines 6–7 of Algorithm 3). Then the leader performs $Best \rightarrow child(Best)$ on its copy of $\cup Cut_i$ and T_g (Line 8 of Algorithm 2). This means specializing each record $t \in T_g$ containing the value of *Best* into more specialized records, $t'_1, \ldots, t'_z$ containing the child values of $child(Best)$. Similarly, other parties update their $\cup Cut_i$ and T_g, and partition $T_g[t]$ into $T_g[t'_1], \ldots, T_g[t'_z]$ (Line 8 of Algorithm 3). Finally, the leader again collects global count statistics from the other parties (Lines 9–10 of Algorithms 2 and 3) to update the *Score* and validity of the candidates (Line 11 of Algorithm 2). The algorithm terminates when there are no more valid candidates in $\cup Cut_i$.

Example 5. Consider Table I with $L = 2$, $K = 2$, $C = 50\%$, and $QID = \{Job, Sex, Age\}$. Initially, all data records are generalized to $\langle ANY\_Job, ANY\_Sex, [1\text{-}99)\rangle$ in T_g, and $\cup Cut_i = \{ANY\_Job, ANY\_Sex, [1\text{-}99)\}$.

Algorithm 2 Distributed Anonymization Algorithm for Leader Party

```
1: Initialize $T_g$ to include one record containing the top most values;
2: Initialize $Cut_i$ to include all the valid top most values;
3: Send Information to Party 2;
4: Read Information from Party $n$;
5: while some $x \in \cup Cut_i$ is valid do
6:     Find the Best specialization from $\cup Cut_i$;
7:     Send Instruction to Party 2 to specialize Best on $T_g$;
8:     Perform Best on $T_g$ and update $\cup Cut_i$;
9:     Send Information to Party 2;
10:    Read Information from Party $n$;
11:    Update the $Score(x)$ and validity for $\forall x \in \cup Cut_i$;
12: end while
13: Send End to Party 2 and terminate;
```

Algorithm 3 Distributed Anonymization Algorithm for Non-leader Parties

```
1: Initialize $T_g$ to include one record containing the top most values;
2: Initialize $Cut_i$ to include all the valid top most values;
3: Read Information from Party $(i - 1)$;
4: Send Information to Party $(i + 1)$ % $n$ after adding its own information;
5: while received message $\neq$ End do
6:     Read Instruction from Party $(i - 1)$;
7:     Send Instruction to Party $(i + 1)$ % $n$;
8:     Perform specialization on $T_g$ according to the received Instruction;
9:     Read Information from Party $(i - 1)$;
10:    Send Information to Party $(i + 1)$ % $n$ after adding its own counts;
11: end while
12: Send message End to Party $(i + 1)$ % $n$ and terminate;
```

To find the *Best* specialization among the candidates in $\cup Cut_i$, the leader collects the global count statistics to compute $Score(ANY\_Job)$, $Score(ANY\_Sex)$, and *Score*(*[1-99)*).

Similar to the centralized algorithm, we describe the key steps as follows: find the *Best* candidate, perform the *Best* specialization, and update the *Score* and validity of the candidates. Only the leader determines the *Best* candidate and updates the *Score* and validity of the candidates. All the other parties perform the *Best* specialization according to the instruction of the leader. Finally, all the parties integrate their local anonymous databases after anonymization.

5.1 Find the Best Candidate

Initially, the leader computes the *Score* for all candidates x in $\cup Cut_i$ to determine the *Best* candidate. For each subsequent iteration, $Score(x)$ come from the update done in the previous iteration (Line 11 of Algorithm 2). To calculate the *Score* of a candidate x, the leader needs the value of $|T[x]|$, $|T[c]|$, $freq(T[x], cls)$, and $freq(T[c], cls)$, where $c \in child(x)$ and cls is a class label. Refer to Section 3.2 for *Score* functions. These values can be obtained by summing up the individual count statistics from all the parties: $|T[x]| = \sum_i |T_i[x]|$, $|T[c]| = \sum_i |T_i[c]|$,

$freq(T[x], cls) = \sum_i freq(T_i[x], cls)$, and $freq(T[c], cls) = \sum_i freq(T_i[c], cls)$. However, disclosing these values for summation violates the privacy requirement, since a party should not know the count statistics of other parties. To overcome this problem, we use secure sum protocol.

Secure sum protocol calculates the sum of the values from different parties without disclosing the value of any individual. Suppose there are n (> 2) different parties each holding a secret number, where Party 1 is the leader. The leader first generates a random number R, adds it to its local value v_1 and sends the sum $R + v_1$ to Party 2. Thus, Party 2 does not know the value of v_1. For the remaining parties, $2 \le i \le n - 1$, each party receives $V = R + \sum_{j=1}^{i-1} v_j$, adds its own value to the sum and passes it to Party $i + 1$. Finally, Party n receives the sum, adds its value, and passes it to Party 1. Since Party 1 (leader) knows the random number, it can obtain the summation by subtracting R from V. Hence, the leader can determine the summation without knowing the secret value of the individual parties. However, secure sum protocol does not work when $n = 2$ because Party 1 can always know the value of Party 2 by subtracting its own value from the summation. We further discuss this issue in Section 5.5.

To obtain the global count statistics, the leader first creates an *Information* message by adding random numbers to its own local count statistics and passes the message to Party 2 (Line 3 of Algorithm 2). Similarly, all of the nonleader parties add their count statistics to the *Information* and pass it to the next party (Lines 3–4 of Algorithm 3). Finally, the leader gets the message from Party n and subtracts the random numbers to get the global count statistics for computing the *Score* of the candidates.

Example 6. Continue with Example 5. First, the leader (Party 1) computes the *Information* message by its local count statistics. The *Information* message has two parts: validity and score. The validity portion contains count statistics needed to determine the validity of the candidates. Specifically, it contains the number of records generalized to a particular equivalence group and the size of the new subgroups if any of the attribute is specialized. Following is the validity part of an *Information* message.

Validity = {(*ANY_Job, ANY_Sex, [1-99), 3(1)), (ANY_Job, 2(1), 1(0)),*
(ANY_Sex, 2(1), 1(0)), ([1-99), 3(1), 0(0))}

This means that Party 1 has three records in an equivalence group with *qid* = {*ANY_Job*, *ANY_Sex*, *[1-99)*}, where one of the records contains sensitive value. If *ANY_Job* is specialized, then it generates two equivalence groups, where the first group contains two records including one sensitive value and the other group contains one record with no sensitive value. Similarly, it also contains the count statistics if *ANY_Sex* and *[1-99)* are specialized. Note that, validity part also provides enough count statistics to compute the *Score* for general data analysis.

Score part contains count statistics needed to compute the *Score* for classification analysis. It contains the number of records for all the class labels for each candidate in the $\cup Cut_i$. Following is the score part of an *Information* message.

$$Score = \{(ANY\_Job, 1, 2)\ (Blue\text{-}collar, 1, 1)\ (White\text{-}collar, 0, 1),$$
$$(ANY\_Sex, 1, 2)\ (M, 1, 1)\ (F, 0, 1), ([1\text{-}99), 1, 2)\ ([1\text{-}60), 1, 2)\ ([60\text{-}99), 0, 0)\}$$

The number of records are one and two for the class labels "Yes" and "No" respectively for the *ANY_Job*. It also provides the detailed counts when *ANY_Job* is specialized into *Blue-collar* and *White-collar*. *Blue-collar* has one record containing "Yes" and one record containing "No" class label. *White-collar* has only one record with "No" class label. Similarly, it provides necessary counts for the other candidates. After computing the *Information* message, Party 1 adds a random number to each of the values and sends the message to Party 2. As mentioned earlier, all of the parties add their part into the *Information* and thus the message comes back to the leader with the global count statistics. Then the leader subtracts the random numbers to get the real global counts for computing the *Score* and validity of the candidates. Figure 4 shows the information flow among the parties.

5.2 Perform the Best Candidate

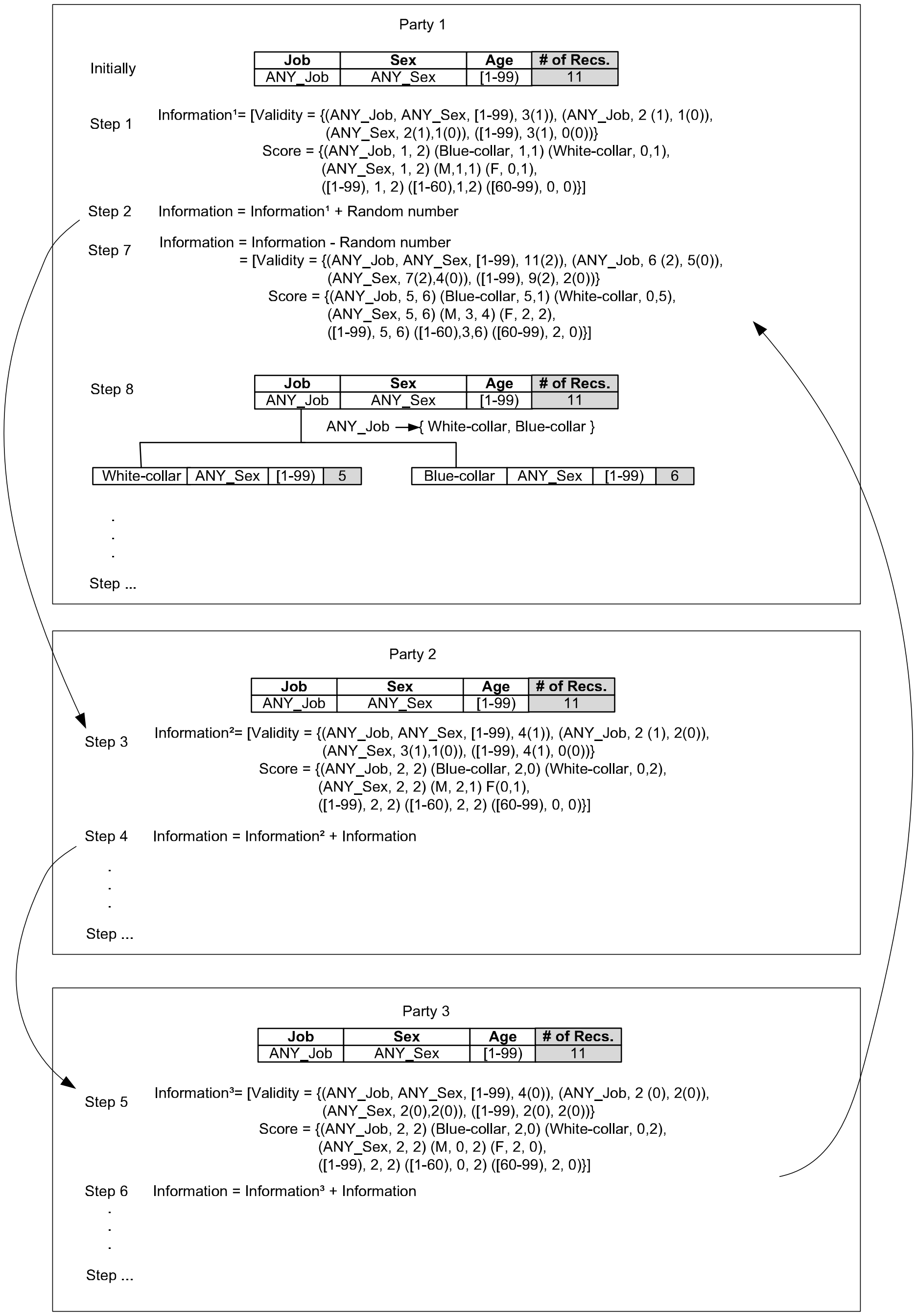
Once the *Best* candidate is determined, the leader instructs all the other parties to specialize $Best \rightarrow child(Best)$ on their local T_g (Line 7 of Algorithm 2). The *Instruction* message contains the *Best* attribute and the number of global generalized records in each new subgroups. Similar to the centralized anonymization algorithm, each party uses the Taxonomy Indexed Partitions (TIPS) data structure to facilitate the operations on T_g. The difference is that in the centralized approach, one party (central government health agency) specializes the records, but in the distributed setting, every data holder concurrently specialize its own records. If $\bigcup Cut_i$ has no valid attributes, then the leader sends the *End* message to terminate the anonymization algorithm. Thus, both centralized and distributed anonymization algorithms produce the same anonymous integrated table by performing the same sequence of operations.

Example 7. Continue with Example 6. Initially, TIPS has one partition (root) representing the most generalized record $\langle ANY\_Job, ANY\_Sex, [1\text{-}99) \rangle$. Suppose that the *Best* candidate is

$$ANY\_Job \rightarrow \{Blue\text{-}collar, White\text{-}collar\}.$$

The leader creates two child nodes under the root and partitions $T_g[root]$ between them resulting in the TIPS in Figure 4 and further instructs Party 2 to perform the same specialization. On receiving this instruction, Party 2 sends the message to the next party and similarly creates two child nodes under the root in its copy of TIPS. Thus, all the parties perform the same operation on their TIPS. This specialization process continues as long as there is a valid candidate in $\bigcup Cut_i$.

Updating TIPS. Updating TIPS is similar to the centralized approach. Each party refines its own leaf partition P_{Best} on $Link_{Best}$ into child partitions P_c.



Party 1

Initially

Job	Sex	Age	# of Recs.
ANY_Job	ANY_Sex	[1-99)	11

Step 1 Information^1= [Validity = {(ANY_Job, ANY_Sex, [1-99), 3(1)), (ANY_Job, 2 (1), 1(0)),
(ANY_Sex, 2(1),1(0)), ([1-99), 3(1), 0(0))}
Score = {(ANY_Job, 1, 2) (Blue-collar, 1,1) (White-collar, 0,1),
(ANY_Sex, 1, 2) (M,1,1) (F, 0,1),
([1-99), 1, 2) ([1-60),1,2) ([60-99), 0, 0)}]

Step 2 Information = Information^1 + Random number

Step 7 Information = Information - Random number
= [Validity = {(ANY_Job, ANY_Sex, [1-99), 11(2)), (ANY_Job, 6 (2), 5(0)),
(ANY_Sex, 7(2),4(0)), ([1-99), 9(2), 2(0))}
Score = {(ANY_Job, 5, 6) (Blue-collar, 5,1) (White-collar, 0,5),
(ANY_Sex, 5, 6) (M, 3, 4) (F, 2, 2),
([1-99), 5, 6) ([1-60),3,6) ([60-99), 2, 0)}]

Step 8

Job	Sex	Age	# of Recs.
ANY_Job	ANY_Sex	[1-99)	11

ANY_Job → { White-collar, Blue-collar }

White-collar	ANY_Sex	[1-99)	5

Blue-collar	ANY_Sex	[1-99)	6

.
.
.

Step ...

Party 2

Job	Sex	Age	# of Recs.
ANY_Job	ANY_Sex	[1-99)	11

Step 3 Information^2= [Validity = {(ANY_Job, ANY_Sex, [1-99), 4(1)), (ANY_Job, 2 (1), 2(0)),
(ANY_Sex, 3(1),1(0)), ([1-99), 4(1), 0(0))}
Score = {(ANY_Job, 2, 2) (Blue-collar, 2,0) (White-collar, 0,2),
(ANY_Sex, 2, 2) (M, 2,1) F(0,1),
([1-99), 2, 2) ([1-60), 2, 2) ([60-99), 0, 0)}]

Step 4 Information = Information^2 + Information

.
.
.

Step ...

Party 3

Job	Sex	Age	# of Recs.
ANY_Job	ANY_Sex	[1-99)	11

Step 5 Information^3= [Validity = {(ANY_Job, ANY_Sex, [1-99), 4(0)), (ANY_Job, 2 (0), 2(0)),
(ANY_Sex, 2(0),2(0)), ([1-99), 2(0), 2(0))}
Score = {(ANY_Job, 2, 2) (Blue-collar, 2,0) (White-collar, 0,2),
(ANY_Sex, 2, 2) (M, 0, 2) (F, 2, 0),
([1-99), 2, 2) ([1-60), 0, 2) ([60-99), 2, 0)}]

Step 6 Information = Information^3 + Information

.
.
.

Step ...

Fig. 4. Distributed anonymization.

$Link_c$ is created to link up the new P_c's for the same c. Add P_c to every $Link_x$ other than $Link_{Best}$ to which P_{Best} was previously linked. While scanning the records in P_{Best}, each Party i also collects the following counts for updating $Score$. (1) For each c in $child(w)$: $|T_A[c]|$, $|T_A[d]|$, $freq(T_A[c], cls)$, and $freq(T_A[d], cls)$, where $d \in child(c)$ and cls is a class label. Refer to Section 3.2 for these notations. $|T_A[c]|$ is computed by $\sum |P_c|$ for P_c on $Link_c$. (2) For each P_c on $Link_c$: $|P_d|$, where P_d is a child partition under P_c *as if* c was specialized. These count statistics are collected by the leader through $Information$ message (Lines 9–10 of Algorithms 2 and 3) to update the $Score$ and validity of the candidates. Thus, updating $Score(x)$ (in Section 5.3) makes use of the count statistics without accessing raw data records.

5.3 Update Score and Validity

This step is performed only by the leader and is similar to the centralized approach. All of the count statistics that are needed to update $Score(x)$ and validity for candidates x in $\cup Cut_i$ are collected through the $Information$ message from all the other parties that they maintained in Section 5.2.

5.4 Data Integration

After executing the distributed anonymization algorithm, each party generates a local anonymous database which by itself may not satisfy LKC-privacy, while the union of the local anonymous databases is guaranteed to satisfy the privacy requirements. The final task is to integrate these local anonymous databases before giving it to the BTS. A naïve approach could be that each data holder sends its local anonymous data to the leader for data integration. However, this approach does not provide effective enough privacy to the data holders because it reveals the ownership of the records to the leader. To provide better privacy, *secure set union protocol* can be used to integrate the data distributively without disclosing any additional information [Clifton et al. 2002]. Secure set union is a useful SMC technique that determines the union of the items securely from different parties. We can use the secure set union protocol proposed by Jurczyk and Xiong [2008] to securely integrate the local anonymous databases from different data holders.

5.5 Analysis

The distributed anonymization algorithm produces the same anonymous integrated table as the centralized anonymization algorithm. This claim follows from the fact that Algorithms 2 and 3 perform exactly the same sequence of specializations as the centralized anonymization algorithm in a distributed manner where T_i is kept locally at each party.

For the privacy requirement, the only information revealed to the leader is content found in the global count statistics of $Information$ message. The count statistics are needed for the calculation of $Score$ and validity of the candidates. The validity part of the $Information$ message determines whether a candidate can be further specialized or not. However, such information can also be determined from the final integrated table because a specialization should take place as long as it is valid. The disclosure of the score part does not breach privacy because it contains only the frequency of the class labels for the candidates. These values only indicate how good a candidate is for classification analysis, and does not provide any information for a particular record. Moreover, the *Score* is computed by the leader over the global count statistics without the knowledge of the individual local counts.

The computation cost of the distributed algorithm is similar to the centralized approach. Each party only scans its own data in every iteration. As a result, the computational cost for each party is bounded by $O(|T_i|log|T|)$. However, distributed algorithm has some additional communication overhead. In every iteration, each party sends one *Instruction* and one $Information$ message. The *Instruction* message contains the *Best* candidate that needs to be specialized. The $Information$ message contains different count statistics for every candidate in the $\cup Cut_i$. Thus, these messages are compact. Moreover, there is a synchronization delay in every iteration, which is proportional to the number of parties n since the parties form a ring topology.

Due to the limitation of the employed secure sum protocol in our proposed distributed anonymization algorithm, the present solution is applicable only if there are more than two parties. A distributed anonymization algorithm for two parties requires a different cryptographic technique, which is not as simple as the secure sum protocol [Du 2001]. The solution for the two-party case is beyond the scope of this article because in the BTS scenario, the number of hospitals is more than two.

6. EXPERIMENTAL EVALUATION

In this section, our objectives are to study the impact of enforcing various LKC-privacy requirements on the data quality in terms of classification error and discernibility cost, and to evaluate the efficiency and scalability of our proposed centralized and distributed anonymization methods by varying the thresholds of maximum adversary's knowledge L, minimum anonymity K, and maximum confidence C.

We employ two real-life datasets, *Blood* and *Adult*. *Blood* is a real-life blood transfusion dataset owned by an anonymous health institute. *Blood* has 62 attributes after removing explicit identifiers; 41 of them are QID attributes. *Blood Group* represents the *Class* attribute with 8 possible values. *Diagnosis Codes*, which has 15 possible values representing 15 categories of diagnosis, is considered to be the sensitive attribute. The remaining attributes are neither quasi-identifiers nor sensitive. *Blood* contains 10,000 blood transfusion records in 2008. Each record represents one incident of blood transfusion. The publicly available *Adult* dataset [Newman et al. 1998] is a *de facto* benchmark for testing anonymization algorithms [Bayardo and Agrawal 2005; Fung et al. 2007; Iyengar 2002; Machanavajjhala et al. 2007; Wang et al. 2007]. *Adult* has 45,222 census records on 6 numerical attributes, 8 categorical attributes, and a binary *Class* column representing two income levels, $\leq$50K or $>$50K. See Fung et al. [2007] for the description of attributes. We consider *Divorced* and *Separated* in the attribute *Marital-status* as sensitive, and the remaining 13 attributes as

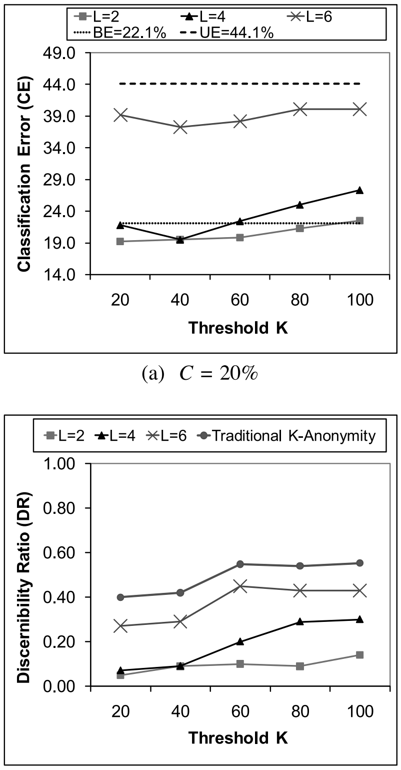
QID. All experiments were conducted on an Intel Core2 Quad Q6600 2.4GHz PC with 2GB RAM.

6.1 Data Utility

To evaluate the impact on classification quality (Case 1 in Section 3.2.1), we use all records for generalization, build a classifier on 2/3 of the generalized records as the training set, and measure the *classification error* (CE) on 1/3 of the generalized records as the testing set. For classification models, we use the well-known C4.5 classifier [Quinlan 1993]. To better visualize the cost and benefit of our approach, we measure additional errors: *Baseline Error* (BE) is the error measured on the raw data without generalization. $BE - CE$ represents the cost in terms of classification quality for achieving a given LKC-privacy requirement. A naïve method to avoid identity and attributes linkages is to simply remove all QID attributes. Thus, we also measure *upper bound error* (UE), which is the error on the raw data with all QID attributes removed. $UE - CE$ represents the benefit of our method over the naïve approach.

To evaluate the impact on general analysis quality (Case 2 in Section 3.2.2), we use all records for generalization and measure the discernibility ratio (DR) on the final anonymous data. $DR = \frac{\sum_{qid} |T[qid]|^2}{|T|^2}$. DR is the normalized discernibility cost, with $0 \leq DR \leq 1$. Lower DR means higher data quality. Sections 6.1.1 and 6.1.2 discuss the experimental results for centralized and distributed anonymization, respectively.

6.1.1 *Centralized Anonymization.* Figure 5(a) depicts the classification error CE with adversary's knowledge $L = 2, 4, 6$, anonymity threshold $20 \leq K \leq 100$, and confidence threshold $C = 20\%$ on the *Blood* dataset. This setting allows us to measure the performance of the centralized algorithm against identity linkages for a fixed C. CE *generally* increases as K or L increases. However, the increase is not monotonic. For example, the error drops slightly when K increases from 20 to 40 for $L = 4$. This is due to the fact that generalization has removed some noise from the data, resulting in a better classification structure in a more general state. For the same reason, some test cases on $L = 2$ and $L = 4$ have $CE < BE$, implying that generalization not only achieves the given LKC-privacy requirement but sometimes may also improve the classification quality. $BE = 22.1\%$ and $UE = 44.1\%$. For $L = 2$ and $L = 4$, $CE - BE$ spans from -2.9% to 5.2% and $UE - CE$ spans from 16.8% to 24.9%, suggesting that the cost for achieving LKC-privacy is small, but the benefit is large when L is not large. However, as L increases to 6, CE quickly increases to about 40%, the cost increases to about 17%, and the benefit decreases to 5%. For a greater value of L, the difference between LKC-privacy and K-anonymity is very small in terms of classification error since more generalized data does not necessarily worse classification error. This result confirms that the assumption of an adversary's prior knowledge has a significant impact on the classification quality. It also indirectly confirms the curse of high dimensionality [Aggarwal 2005].



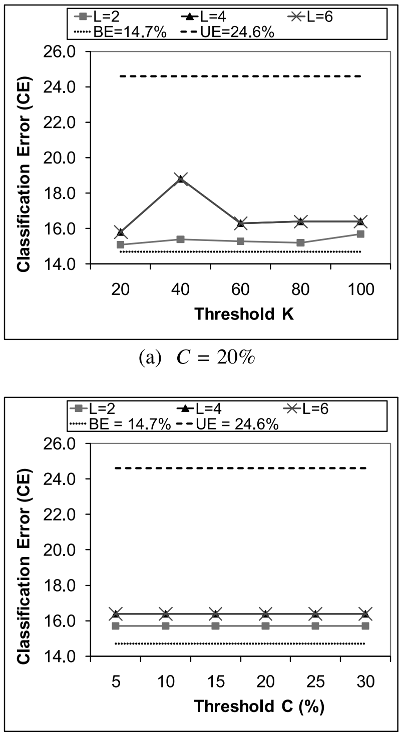
(a) C = 20%

(b) C = 20%

Fig. 5. *Blood* dataset.

Figure 5(b) depicts the discernibility ratio DR with adversary's knowledge L = 2, 4, 6, anonymity threshold $20 \leq K \leq 100$, and a fixed confidence threshold C = 20%. DR *generally* increases as K increases, so it exhibits some tradeoff between data privacy and data utility. As L increases, DR increases rapidly because more generalization is required to ensure each equivalence group has at least K records. To illustrate the benefit of our proposed LKC-privacy model over the traditional K-anonymity model, we measure the discernibility ratio, denoted DR_{TradK}, on traditional K-anonymous solutions produced by the TDR method in Fung et al. [2007]. $DR_{TradK} - DR$, representing the benefit of our model, spans from 0.1 to 0.45. This indicates a significant improvement on data quality by making a reasonable assumption on limiting the adversary's knowledge within L known values. Note, the solutions produced by TDR do not prevent attribute linkages although they have higher discernibility ratio.

Figure 6(a) depicts the classification error CE with adversary's knowledge L = 2, 4, 6, anonymity threshold $20 \leq K \leq 100$, and confidence threshold C = 20% on the *Adult* dataset. BE = 14.7% and UE = 24.5%. For L = 2, $CE - BE$ is less than 1% and $UE - CE$ spans from 8.9% to 9.5%. For L = 4 and L = 6, $CE - BE$ spans from 1.1% to 4.1%, and $UE - CE$ spans from 5.8% to 8.8%.



(a) C = 20%

(b) K = 100

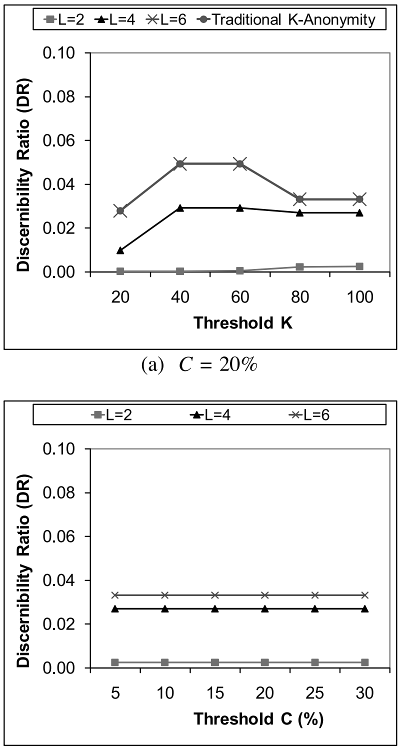
Fig. 6. *Adult* dataset.

These results suggest that the cost for achieving LKC-privacy is small, while the benefit of our method over the naïve method is large.

Figure 6(b) depicts the CE with adversary's knowledge $L = 2, 4, 6$, confidence threshold $5\% \leq C \leq 30\%$, and anonymity threshold $K = 100$. This setting allows us to measure the performance of the algorithm against attribute linkages for a fixed K. The result suggests that CE is insensitive to the change of confidence threshold C. CE slightly increases as the adversary's knowledge L increases.

Figure 7(a) depicts the discernibility ratio DR with adversary's knowledge $L = 2, 4, 6$, anonymity threshold $20 \leq K \leq 100$, and confidence threshold $C = 20\%$. DR sometimes has a drop when K increases. This is a result of the greedy algorithm only identifying the suboptimal solution. DR is insensitive to the increase of K and stays close to 0 for $L = 2$. As L increases to 4, DR increases significantly and finally equals traditional K-anonymity when $L = 6$ because the number of attributes in *Adult* is relatively smaller than in *Blood*. Yet K-anonymity does not prevent attribute linkages, while our LKC-privacy provides this additional privacy guarantee.

Figure 7(b) depicts the DR with adversary's knowledge $L = 2, 4, 6$, confidence threshold $5\% \leq C \leq 30\%$, and anonymity threshold $K = 100$. In general, DR increases as L increases due to a more restrictive privacy requirement.



(a) $C = 20\%$

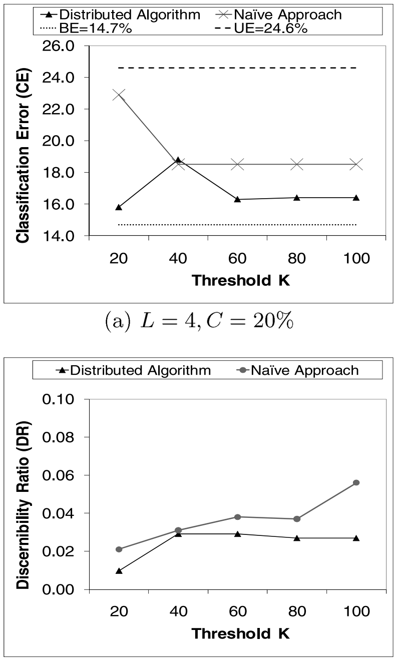
(b) $K = 100$

Fig. 7. *Adult* dataset.

Similar to Figure 6b, the DR is insensitive to the change of confidence threshold C. It implies that the primary driving forces for generalization are L and K, not C.

6.1.2 *Distributed Anonymization.* The distributed anonymization algorithm achieves same data utility as the centralized anonymization algorithm and thus all the previous results also hold for distributed anonymization algorithm. Here, we show the benefit of our distributed anonymization algorithm over the naïve "generalize-then-integrate" approach. We divide the $45,222$ records of *Adult* dataset equally among three parties. In the naïve approach, parties first generalizes their data to satisfy LKC-privacy. Classification error and discernibility ratio are then calculated on the integrated anonymous data collected from the parties.

Figure 8(a) depicts the classification error CE with adversary's knowledge $L = 4$, anonymity threshold $20 \leq K \leq 100$, and confidence threshold $C = 20\%$ on the *Adult* dataset. For the naïve approach, $CE - BE$ spans from 3.8% to 8.2%, and $UE - CE$ spans from 1.7% to 6.1%. This result confirms that the naïve approach loses a significant amount of data due to prior generalization before integration.



(a) $L = 4, C = 20\%$

(b) $L = 4, C = 20\%$

Fig. 8. *Adult* dataset.

Figure 8(b) depicts the discernibility ratio DR with adversary's knowledge L = 4, anonymity threshold $20 \leq K \leq 100$, and confidence threshold C = 20%. For both the approaches, DR generally increases as K increases. However, the naïve approach has a higher discernibility ratio for all the values of K, confirming the benefit of the distributed anonymization algorithm over the naïve approach.

6.2 Efficiency and Scalability

One major contribution of our work is the development of an efficient and scalable algorithm for achieving LKC-privacy on high-dimensional healthcare data. Every previous test case can finish the entire anonymization process within 30 seconds. We further evaluate the scalability of our algorithm with respect to data volume by blowing up the size of the *Adult* dataset. First, we combined the training and testing sets, giving 45,222 records. For each original record r in the combined set, we created $\alpha - 1$ "variations" of r, where $\alpha > 1$ is the blowup scale. Together with all original records, the enlarged dataset has $\alpha \times 45,222$ records.

Figure 9 depicts the runtime of the centralized anonymization algorithm from 200,000 to 1 million records for L = 4, K = 20, C = 100%. The total runtime for anonymizing 1 million records is 107s, where 50s are spent on reading raw data, 33s are spent on anonymizing, and 24s are spent on writing the anonymous data. Our algorithm is scalable due to the fact that we use the

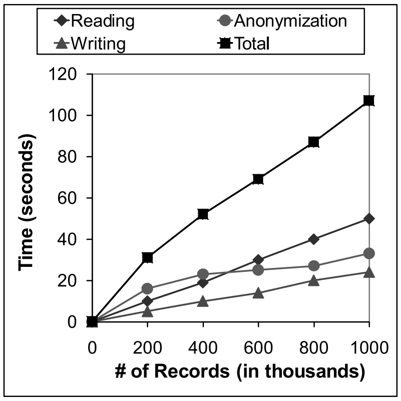


Fig. 9. Scalability (L = 4, K = 20, C = 100%).

count statistics to update the *Score*, and thus it only takes one scan of data per iteration to anonymize the data. As the number of records increases, the total runtime increases linearly.

6.3 Summary

The experimental results on the two real-life datasets can be summarized as follows: (1) Our anonymization methods can effectively preserve both privacy and data utility in the anonymous data for a wide range of LKC-privacy requirements. There is a trade-off between data privacy and data utility with respect to K and L, but the trend is less obvious on C. (2) Our proposed LKC-privacy model retains more information than the traditional K-anonymity model and provides the flexibility to adjust privacy requirements according to the assumption of adversary's background knowledge. (3) The proposed methods are highly scalable for large datasets. These characteristics make our algorithms a promising component for anonymizing healthcare data.

7. CONCLUSION AND LESSON LEARNED

We have proposed two anonymization algorithms to address the centralized and distributed anonymization problems for healthcare institutes with the objective of supporting data mining. Motivated by the BTS's privacy and information requirements, we have formulated the LKC-privacy model for high-dimensional relational data. Moreover, our developed algorithms can accommodate two different information requirements according to the BTS' information need. Our proposed solutions are different from privacy-preserving data mining (PPDM) due to the fact that we allow *data sharing* instead of *data mining result sharing*. This is an essential requirement for the BTS since they require the flexibility to perform various data analysis tasks. We believe that our proposed solutions could serve as a model for data sharing in the healthcare sector.

Finally, we would like to share our collaborative experience with the healthcare sector. Health data is complex, often a combination of relational data, transaction data, and textual data. Thus, our project focuses only on the relational data, but we notice that some recent works [Gardner and Xiong 2009; Ghinita et al. 2008; Terrovitis et al. 2008; Xu et al. 2008], are applicable to solve the privacy problem on transaction and textual data in the BTS case. Besides the technical issue, it is equally important to educate health institute management and medical practitioners about the latest privacy-preserving technology. When management encounters the problem of privacy-aware information sharing as presented in this paper, their initial response is often to set up a traditional role-based secure access control model. In fact, alternative techniques, such as privacy-preserving data mining and data publishing [Aggarwal and Yu 2008; Fung et al. 2010], are available provided that the data mining quality does not significantly degrade.

REFERENCES

ADAM, N. R. AND WORTMAN, J. C. 1989. Security control methods for statistical databases. *ACM Comput. Surv. 21*, 4, 515–556.

AGGARWAL, C. C. 2005. On *k*-anonymity and the curse of dimensionality. In *Proceedings of the International Conference on Very Large Databases*.

AGGARWAL, C. C. AND YU, P. S. 2008. *Privacy Preserving Data Mining: Models and Algorithms*. Springer.

AGRAWAL, R. AND SRIKANT, R. 2000. Privacy preserving data mining. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

BAYARDO, R. J. AND AGRAWAL, R. 2005. Data privacy through optimal *k*-anonymization. In *Proceedings of the International Conference on Data Engineering*.

BLUM, A., DWORK, C., MCSHERRY, F., AND NISSIM, K. 2005. Practical privacy: the sulq framework. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*.

CARLISLE, D. M., RODRIAN, M. L., AND DIAMOND, C. L. 2007. California inpatient data reporting manual, medical information reporting for California, 5th edition. Tech. rep., Office of Statewide Health Planning and Development.

CLIFTON, C., KANTARCIOGLU, M., VAIDYA, J., LIN, X., AND ZHU, M. Y. 2002. Tools for privacy preserving distributed data mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Explor. Newslett. 4*, 2, 28–34.

DINUR, I. AND NISSIM, K. 2003. Revealing information while preserving privacy. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*.

DU, W., HAN, Y. S., AND CHEN, S. 2004. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the SIAM International Conference on Data Mining*.

DU, W. AND ZHAN, Z. 2002. Building decision tree classifier on private data. In *Proceedings of the IEEE ICDM Workshop on Privacy, Security, and Data Mining*.

DU, W. L. 2001. A study of several specific secure two-party computation problems. PhD thesis, Purdue University, West Lafayette.

DWORK, C. 2006. Differential privacy. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*.

DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography Conference*.

FULLER, W. A. 1993. Masking procedures for microdata disclosure limitation. *Official Statistics*.

FUNG, B. C. M., WANG, K., CHEN, R., AND YU, P. S. 2010. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv. 42*, 4, 1–53.

FUNG, B. C. M., WANG, K., AND YU, P. S. 2007. Anonymizing classification data for privacy preservation. *IEEE Trans. Knowl. Data Engin. 19*, 5, 711–725.

GARDNER, J. AND XIONG, L. 2009. An integrated framework for de-identifying heterogeneous data. *Data Knowl. Engin.*

GHINITA, G., TAO, Y., AND KALNIS, P. 2008. On the anonymization of sparse high-dimensional data. In *Proceedings of the International Conference on Data Engineering*.

IYENGAR, V. S. 2002. Transforming data to satisfy privacy constraints. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

JIANG, W. AND CLIFTON, C. 2005. Privacy-preserving distributed k-anonymity. In *Proceedings of the Working Conference on Data and Applications Security*.

JIANG, W. AND CLIFTON, C. 2006. A secure distributed framework for achieving k-anonymity. *J. VLDB 15*, 4, 316–333.

JURCZYK, P. AND XIONG, L. 2008. Towards privacy-preserving integration of distributed heterogeneous data. In *Proceedings of the PhD Workshop on Information and Knowledge Management (PIKM)*.

JURCZYK, P. AND XIONG, L. 2009. Distributed anonymization: Achieving privacy for both data subjects and data providers. In *Proceedings of the Working Conference on Data and Applications Security*.

KIM, J. AND WINKLER, W. 1995. Masking microdata files. In *Proceedings of the ASA Section on Survey Research Methods*.

LEFEVRE, K., DEWITT, D. J., AND RAMAKRISHNAN, R. 2006. Mondrian multidimensional k-anonymity. In *Proceedings of the International Conference on Data Engineering*.

LEFEVRE, K., DEWITT, D. J., AND RAMAKRISHNAN, R. 2008. Workload-aware anonymization techniques for large-scale datasets. *ACM Trans. Datab. Syst.*

MACHANAVAJJHALA, A., KIFER, D., GEHRKE, J., AND VENKITASUBRAMANIAM, M. 2007. ℓ-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*.

MOHAMMED, N., FUNG, B. C. M., HUNG, P. C. K., AND LEE, C. 2009a. Anonymizing healthcare data: A case study on the blood transfusion service. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

MOHAMMED, N., FUNG, B. C. M., WANG, K., AND HUNG, P. C. K. 2009b. Privacy-preserving data mashup. In *Proceedings of the International Conference on Extending Database Technology*.

NEWMAN, D. J., HETTICH, S., BLAKE, C. L., AND MERZ, C. J. 1998. UCI Repository of Machine Learning Databases.

QUINLAN, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

SAMARATI, P. 2001. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Engin.*

SCHNEIER, B. 1995. *Applied Cryptography*. 2nd Ed. John Wiley & Sons.

SKOWRON, A. AND RAUSZER, C. 1992. The discernibility matrices and functions in information systems. In *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Set Theory*.

SWEENEY, L. 2002. k-anonymity: A model for protecting privacy. *Int. J. Uncert. Fuzz. Knowl. Based Syst.*

TERROVITIS, M., MAMOULIS, N., AND KALNIS, P. 2008. Privacy-preserving anonymization of set-valued data. In *Proceedings of the International Conference on Very Large Databases*.

VAIDYA, J. AND CLIFTON, C. 2002. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

VAIDYA, J. AND CLIFTON, C. 2003. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

WANG, K., FUNG, B. C. M., AND YU, P. S. 2007. Handicapping attacker's confidence: An alternative to k-anonymization. *Knowl. Inform. Syst. 11*, 3, 345–368.

WONG, R. C. W., LI., J., FU, A. W. C., AND WANG, K. 2006. α, k-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

XIAO, X. AND TAO, Y. 2006a. Anatomy: Simple and effective privacy preservation. In *Proceedings of the International Conference on Very Large Databases*.

XIAO, X. AND TAO, Y. 2006b. Personalized privacy preservation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

XU, Y., WANG, K., FU, A. W. C., AND YU, P. S. 2008. Anonymizing transaction databases for publication. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

YANG, Z., ZHONG, S., AND WRIGHT, R. N. 2005. Privacy-preserving classification of customer data without loss of accuracy. In *Proceedings of the SIAM International Conference on Data Mining*.

ZHAO, K., LIU, B., TIRPAK, T. M., AND XIAO, W. 2005. A visual data mining framework for convenient identification of useful knowledge. In *Proceedings of the IEEE ICDM: IEEE International Conference on Data Mining*.

Received January 2010; accepted June 2010