Differentially Private Distributed Data Release for Data Mining

Noman Mohammed and Wenbo He
School of Computer Science
McGill University
Montreal, QC H3A 0G4
Email:{noman, wenbohe}@cs.mcgill.ca

***Abstract*—In this paper, we study the privacy threats caused by distributed data sharing and present an algorithm to securely integrate person-specific sensitive data from multiple data owners, whereby the integrated data still retains the essential information for supporting general data exploration or a specific data mining task, such as classification analysis.**

I. Introduction

Numerous organizations such as governmental agencies, hospitals, and financial companies collect and share various person-specific data for research and business purposes. Often, data from different sources need to be integrated to gain better insights and deliver highly customizable services to their customers. For example, Shared Pathology Informatics Network (SPIN) initiated by the National Cancer Institute of the United States aims to provide an interface to cancer researchers to access pathology specimens' data stored across multiple healthcare institutes. While data sharing can help their clients obtain the required information or explore new knowledge, it can also be misused by adversaries to reveal sensitive information.

Motivating Example. Consider the raw patient data in Table I, where three hospitals want to integrate their data and use the integrated data to build a classifier on the $Class$ attribute. Each row in the table represents the information of an individual, where records 1 to 3 are from Party A, records 4 to 7 are from Party B, and records 8 to 11 are from Party C. The attribute $Class$ contains the class label Y or N, representing whether or not the patient has cancer. 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.

To prevent such linking attacks, Jurczyk and Xiong [1] and Mohammed *et al.* [2] have proposed algorithms to securely integrate horizontally-partitioned data from multiple data owners. Their methods [1], [2] adopt k-anonymity [3] or its extensions [4] as the underlying privacy principle. However, recent research has indicated that these privacy models are vulnerable to various privacy attacks [5], [6] and provide insufficient privacy protection.

In this paper, we adopt differential privacy [7], a recently proposed privacy model that provides a provable privacy guarantee. Differential privacy is a rigorous privacy model that makes no assumption about an adversary's background knowledge. A differentially-private mechanism ensures that the probability of any output (released data) is equally likely from all nearly identical input data sets and thus guarantees that all outputs are insensitive to any individual's data. In other words, an individual's privacy is not at risk because of the participation in the data set.

TABLE I. Raw patient data

	ID	Job	Sex	Age	Surgery	Class
Party A	1	Janitor	M	34	Transgender	Y
	2	Lawyer	F	58	Plastic	N
	3	Mover	M	58	Urology	N
Party B	4	Lawyer	M	24	Vascular	N
	5	Mover	M	34	Transgender	Y
	6	Janitor	M	44	Plastic	Y
	7	Doctor	F	44	Vascular	N
Party C	8	Doctor	M	58	Plastic	N
	9	Doctor	M	24	Urology	N
	10	Janitor	F	63	Vascular	Y
	11	Mover	F	63	Plastic	Y

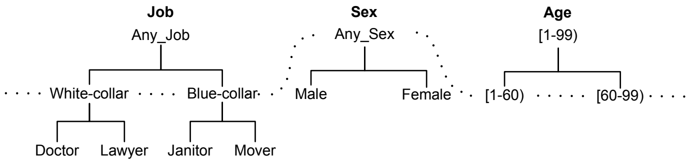

Fig. 1. Taxonomy tree for the attribute, Job, Sex, and Age.

Current Techniques. There are two obvious, yet incorrect approaches. The first one is *integrate-then-anonymize*: first integrate the local tables and then anonymize the integrated table using some single table anonymization methods [8]. Unfortunately, this approach does not preserve privacy in the studied scenario because any party holding the integrated table will immediately know all private information of all parties. The second approach is *anonymize-then-integrate*: first anonymize each table locally and then integrate the anonymous tables. However, such a distributed anonymize-then-integrate approach suffers significant utility loss compared to the centralized integrate-then-anonymize approach due to the extra noise added by each party to satisfy differential privacy.

Contributions. We present a distributed algorithm for differentially-private data release for horizontally-partitioned data among several parties. The proposed algorithm also satisfies the security definition of the semi-honest adversary model. In this model, parties follow the algorithm but may try to deduce additional information from the received messages.

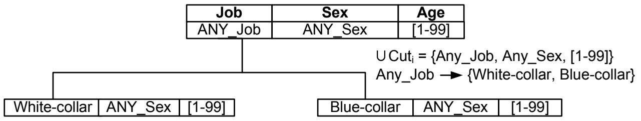

Fig. 2. Generalized Data Table (D_g)

Therefore, at any time during the execution of the algorithm, no party should learn more information about the other party's data than what is found in the final integrated table, which is differentially private. Section II provides an overview of the proposed distributed anonymization algorithm.

II. Distributed Anonymization Algorithm

Preliminaries. Consider an untrusted data aggregator, S and n data owners $\{Party1, \ldots, Partyn\}$, where each Party i owns a private table $D_i(A_1^{pr}, \ldots, A_d^{pr}, A^{cls})$ over the same set of attributes. Each party 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 release an integrated anonymous data table $\hat{D}(A_1^{pr}, \ldots, A_d^{pr}, A^{cls})$ to the public for classification analysis. The untrusted data aggregator facilitates the anonymization process. However, it learns no more information than the final integrated anonymous data table. The attributes in D_i are classified into three categories: (1) An explicit identifier attribute A^i that explicitly identifies an individual, such as *SSN* and *Name*. These attributes are removed before releasing the data. (2) A class attribute A^{cls} that contains the class value, and the goal of the data miner is to build a classifier to accurately predict the value of this attribute. (3) A set of predictor attributes $\mathcal{A}^{pr} = \{A_1^{pr}, \ldots, A_d^{pr}\}$, whose values are used to predict the class attribute.

Given an untrusted data aggregator S, data tables D_i owned by P_i, where $i \in (1, \ldots, n)$ and a privacy parameter ϵ, our objective is to generate an integrated anonymized data table $\hat{D}$ such that (1) $\hat{D}$ satisfies ϵ-differential privacy and (2) the algorithm to generate $\hat{D}$ satisfies the security definition of the semi-honest adversary model. We require the class attribute to be categorical. However, the values of the predictor attribute can be either numerical v_n or categorical v_c. Further, we require that for each predictor attribute A^{pr}, which is either numerical or categorical, a taxonomy tree is provided.

Overview. The data aggregator first generalizes the raw data and then adds noise to achieve ϵ-differential privacy. The general idea is to anonymize the raw data by a sequence of specializations starting from the topmost general state. A specialization, written $v \rightarrow child(v)$ replaces the parent value v with its set of child values $child(v)$. The specialization process can be viewed as pushing the "cut" of each taxonomy tree downwards. A *cut* of the taxonomy tree for an attribute A_i^{pr}, denoted by Cut_i, contains exactly one value on each root-to-leaf path. The data aggregator keeps a copy of the current $\cup Cut_i$ and a generalized table D_g. Initially, all values in $\mathcal{A}^{pr}$ are generalized to the topmost value in their taxonomy trees, and Cut_i contains the topmost value for each attribute A_i^{pr}.

Example 1: Consider Table I and the taxonomy trees presented in Fig. 1. We do not show the attributes *Class* and *Surgery* in Fig. 2 due to space limitation. Initially, D_g contains one root node representing all the records that are generalized to $\langle$*Any_Job,Any_Sex, [1-99)*$\rangle$. $\cup Cut_i$ is represented as $\{$*Any_Job, Any_Sex, [1-99)*$\}$ and includes the initial candidates.

The anonymization process involves the following three steps:

1) Score Calculation: The aggregator calculates the score of the candidates in $\cup Cut_i$ and selects a candidate for specialization. Cryptographic primitives such as homomorphic encryption [9] and garbled circuits [10] are used to calculate the scores securely without leaking any information to the parties.

2) Candidate Selection: Next, the aggregator selects a candidate using exponential mechanism to satisfy differential privacy. Once a candidate is determined, the aggregator specializes the winner candidate w on D_g according to the provided taxonomy trees. Then, the aggregator updates the local copy of $\cup Cut_i$. This process is repeated for a given number of specializations h.

3) Generating Noisy Count: Finally, the aggregator determines the noisy counts of each leaf node of the generalized data table D_g using a distributed noise additionl technique and releases these noisy counts for data analysis.

Example 2: Suppose that the winner candidate is *Any_Job* $\rightarrow$ $\{$*White-collar,Blue-collar*$\}$. The aggregator creates two child nodes under the root node as shown in Fig. 2 and updates $\cup Cut_i$ to $\{$*White-collar*, *Blue-collar*, *Any_Sex*, *[1-99)*$\}$. Suppose that the next winner candidate is *[1-99)* $\rightarrow$ $\{$*[1-60), [60-99)*$\}$. Similarly, the aggregator creates further specialized partitions according to the taxonomy tree and releases each leaf partition along with its noisy count.

III. Concliusion

We proposed an algorithm for anonymizing horizontally-partitioned data among multiple data owners. The proposed algorithm provides differential privacy guarantee and satisfies the security definition of semi-honest adversary model.

References

[1] P. Jurczyk and L. Xiong, "Distributed anonymization: Achieving privacy for both data subjects and data providers," in *DBSec*, 2009.

[2] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C. Lee, "Centralized and distributed anonymization for high-dimensional healthcare data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2010.

[3] L. Sweeney, "k-anonymity: A model for protecting privacy," in *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[4] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys*, 2010.

[5] R. C. W. Wong, A. W. C. Fu, K. Wang, and J. Pei, "Minimality attack in privacy preserving data publishing," in *PVLDB*, 2007.

[6] S. R. Ganta, S. Kasiviswanathan, and A. Smith, "Composition attacks and auxiliary information in data privacy," in *SIGKDD*, 2008.

[7] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006.

[8] N. Mohammed, R. Chen, B. C. M. Fung, and P. S. Yu, "Differentially private data release for data mining," in *SIGKDD*, 2011.

[9] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999.

[10] A. C. Yao, "How to generate and exchange secrets," in *STOC*, 1986.