DARM: A Privacy-preserving Approach for Distributed Association Rules Mining on Horizontally-partitioned Data*

Omar Abdel Wahab
CIISE, Concordia University
Montréal, QC, Canada
o_abul@ciise.concordia.ca

Moulay Omar Hachami
CIISE, Concordia University
Montréal, QC, Canada
m_hacham@ciise.concordia.ca

Arslan Zaffari
CIISE, Concordia University
Montréal, QC, Canada
a_zaffa@ciise.concordia.ca

Mery Vivas
CIISE, Concordia University
Montréal, QC, Canada
m_viva@ciise.concordia.ca

Gaby G. Dagher
CSE, Concordia University
Montréal, QC, Canada
daghir@cse.concordia.ca

ABSTRACT

Extracting association rules helps data owners to unveil hidden patterns from their data for the purpose of analyzing and predicting the behavior of their clients. However, mining association rules in a distributed environment is not a trivial task due to privacy concerns. Data owners are interested in collaborating with each other to mine association rules on a global level; however, they are concerned that sensitive information related to the individuals involved in their database might get compromised during the mining process. In this paper, we formulate and address the problem of answering association rules queries in a distributed environment such that the mining process is confidential and the results are differentially private. We propose a privacy-preserving distributed association rules mining approach, named *DARM*, where global strong association rules are determined in a confidential way, and the results returned satisfy ε-differential privacy. We conduct our experiments on real-life data, and show that our approach can efficiently answer association rules queries and is scalable with increasing data records.

Categories and Subject Descriptors

H.2.7 [**Database Management**]: Database Administration—*Security, integrity, and protection*; H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.2.4 [**Database Management**]: Systems—*Distributed databases, Query processing*

General Terms

Privacy-preserving Data Mining, Database Management

Keywords

Association Rules, Differential Privacy, Data Mining

IDEAS'14, July 07-09 2014, Porto, Portugal

http://dx.doi.org/10.1145/2628194.2628206

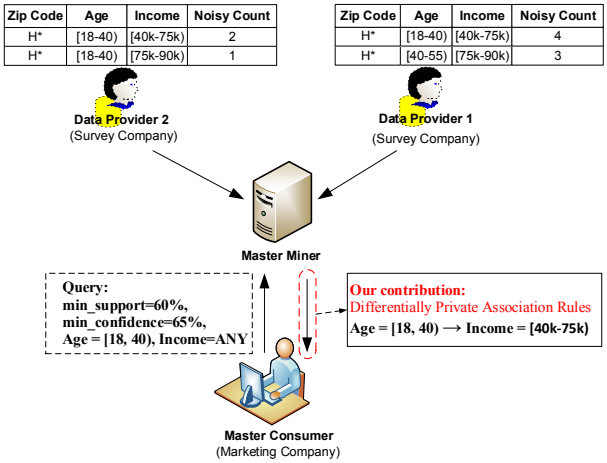

Figure 1: Marketing campaign scenario: A marketing company interested in knowing the income per age to launch its advertising campaign.

1. INTRODUCTION

Due to the rapid evolution of data collection and storage technologies, extracting knowledge and hidden patterns from stored data has become a major necessity for individuals, companies, and government agencies. However, applying data mining techniques to extract information is considered a challenge when the data is distributed over multiple owners, and each data owner is concerned about the privacy of individuals in his data. For example, companies might be interested in obtaining information concerning the fi-

nancial status of individuals from different banks. Privacy-Preserving Data Mining (PPDM) techniques has been utilized in the context of distributed computing to protect the confidentiality of the data of each provider, while still enabling the providers to perform data mining tasks, such as frequent itemsets mining and association rules mining, on the distributed data.

This paper introduces a privacy-preserving approach for distributed association rules mining. Three types of participants are assumed in the proposed model: *data providers*, *master miner*, and *data consumers*. We assume that the data being shared is in the form of a relational table that is horizontally partitioned into sub-tables, each of which is hosted by one data provider. Our framework preserve the privacy of each provider's data while also protecting the query confidentiality against the data providers. The master miner is a central web service platform that perform the mining process based on data consumers' queries and return to each user the strong association rules satisfying her request. The data consumer in our model is a user (individual or company) interested in mining the data to obtain association rules information.

Example 1. Assume that two survey companies (data providers) own information about different set of individuals in the same region. Also assume that a marketing company (data consumer) is interested in obtaining information regarding *income-per-age* in that geographic area for the purpose of launching a targeted advertising campaign, as depicted in Fig. 1. For that purpose, the marketing company sends an association rules query to the master miner that includes the following: $q = \{\gamma = 60\%,\ \alpha = 80\%,\ \mathbb{P} = (Age = [18,23], Income = *)\}$, where γ is the minimum support threshold, α is the minimum confidence threshold, and $\mathbb{P}$ is the set of attribute/value pair the marketing company is interested in. The master miner performs the mining process with the two survey companies and returns to the marketing company the strong rule: $Age = [18,23] \rightarrow Income = [20k - 35k]$.

The challenges of developing such a model are summarized as follows. The first challenge is the privacy of the data, i.e., while fetching the information, one data provider should not learn about the data of any other provider. The second challenge is the quality of the generated rules. The rules must delivered in a way that satisfies the data consumer's need and responds to his request by performing the mining task in a global manner (i.e., collecting information in a global manner from multiple data providers specialized in identical fields to enhance the quality of the generated rules). The third challenge is preventing the data consumer from inferring sensitive information about the individuals involved in the database by analyzing the generated association rules.

The contributions of this paper can be summarized as follows:

- **Contribution** #1: We propose a comprehensive privacy-preserving approach, named *DARM*, for answering association rules queries in a distributed environment, with the goal of preserving both data privacy and query confidentiality.
- **Contribution** #2: Our proposed approach protects all providers against inference attacks from data consumers by guaranteeing that the returned association rules to the data consumer satisfy ε-differential privacy. To the best of our knowledge, this is the first work that provides the strong guarantee of differentially private association rules.
- **Contribution** #3: The proposed method preserves the privacy of the mined data by preventing each data provider from learning sensitive information about other data providers during the mining process.
- **Contribution** #4: The confidentiality of the data consumer's query is protected against the data providers such that the master miner is able to mine the association rules without revealing the query to the data providers.
- **Contribution** #5: We conduct performance evaluation on real-life data to study the scalability and efficiency of our proposed model. Experimental results reveal that our approach is scalable, i.e., it grows sub-linearly with the linear increase in the number of data records. As for efficiency, we also show that our approach is efficient with regard to the number of attributes in the data consumer's query.

2. RELATED WORK

Several approaches [1][7][18][5][3][19] are proposed in the literature to study the problem of mining association rules in distributed and parallel manners, where the data is partitioned across several nodes. However, these approaches are mostly interested in increasing the efficiency of the mining process; ignoring thus the privacy concerns that may arise from building such global mining model.

On the other hand, several approaches consider the privacy concerns that may arise from mining the data globally [14][22][25][24][11]. Kantarcioglu and Clifton [14] and Vaidya and Clifton [22] propose a distributed association rules mining over horizontally partitioned data and vertically partitioned data, respectively. The authors take into account protecting the privacy of the individuals by preventing each data provider from inferring information from other providers. However, these approaches do not protect against possible inference attacks by data consumers. On the contrary, our approach prevents data providers from being able to learn any information from any data provider, as well as protects data providers from inference attacks by ensuring that the result of each query from a data consumer satisfies ε-differential privacy. In [24], the authors propose an encryption scheme based on substitution cipher techniques to preserve the privacy of the transactional data used for outsourcing association rule mining. However, they consider that the association rules mining will be centralized on a single provider, which has to receive the different databases and perform all the association rules mining tasks. In contrast, to avoid such overhead imposed on a single provider, the master miner in our model mines the strong association rules on a global level by sending count queries to the data providers while avoiding to store any part of the data locally. Giannotti et al. [11] tackle the problem of outsourcing the association rule mining task within a corporate privacy-preserving framework by proposing an encryption scheme based on substitution ciphers called *RobFrugal*. In [25], the authors propose a privacy-preserving model that merges the secure multiparty computation and differential privacy to preserve the privacy of the statistical operations (i.e., count and aggregate count). However, it is not clear how this approach can be applied to handle association rules mining given that division operations must be performed between parties in secure way in order to validate the minimum support and confidence. Note that the approaches proposed in [14][22][24][11] rely on encryption to achieve privacy between data providers. However, and besides inefficiency, a recent study

Table 1: Comparative evaluation of main features in related privacy-preserving data mining and association rules approaches.

Approach	Data			Privacy Model		Mining Model		
	Single Provider	Multiple Providers		Differential Privacy	Partition-based Privacy	PPDM		PPDP
		Horizontal Partitioning	Vertical Partitioning			Association Rules	Other	
Anitha *et al.* [3]		●				●		
Kantarcioglu and Clifton [14]		●				●		
Vaidya and Clifton [22]			●			●		
Zhang *et al.* [25]		●		●			●	
Wong *et al.* [24], Giannotti *et al.* [11]	●					●		
Arafati *et al.* [4], Gurunathan *et al.* [12]			●		●			●
Our proposed solution		●		●		●		

shows that most encryption schemes are insufficient to guarantee data privacy and confidentiality, as the protocol on which they are based, namely *precise query protocol (PQP)*, is vulnerable to attribute values inference [8].

Furthermore, several approaches [21][4][12][12] were proposed to preserve the privacy of the data in a data mashup scenario. In contrary to our model which considers the privacy-preserving data mining (PPDM) [23], these approaches are designed to support privacy-preserving data publishing (PPDP) [9] where they assume that the output data itself will be shared among the different parties.

Table 1 summarizes the features of the representative related works, including our proposed solution.

3. PROBLEM FORMULATION

This section formulates the research problem addressed in this paper. First, we give an overview on the problem of mining association rules in a distributed environment while preserving the privacy of both the data and query in Section 3.1. Next, we define the input components used by our approach in Section 3.2. Then, the trust and adversary model is described in Section 3.3. Finally, the problem statement is presented in Section 3.4.

3.1 Problem Overview

In this paper, we design a distributed association rules mining approach consisting of three party types: *data providers*, *data consumers*, and *master miner*. The data provider is a data owner interested in making its data available for data mining tasks. Each provider's data contains the same type of information (attributes) about different set of individuals. The data consumer represents the user who is interested in obtaining strong association rules concerning certain attributes from the distributed data. The *master miner* is a broker trusted by both data providers and data consumers. When the master miner receives an association rules query from the data consumer, it performs the mining process in a privacy-preserving manner with the data providers, and then delivers the strong association rules satisfying ε-differential privacy to the data consumer.

3.2 System Inputs

The proposed distributed association rules mining system takes two inputs: (1) association rules queries from data consumers, and (2) set of anonymized data, each of which is hosted by one data provider. In the following, we describe each of these inputs in details.

3.2.1 Association Rules Queries

To obtain the set of strong association rules R from the distributed data, the data consumer submits a query request q to the master miner in which he specifies the minimum support threshold γ, the minimum confidence threshold α, and a set of predicates $\mathbb{P}$. γ represents the minimum acceptable global support level for the rules, i.e., *for each* association rule $r_i \in R$, $Support(r_i) \geq \gamma$. α represents the minimum acceptable global confidence level for the rules, i.e., *for each* association rule $r_i \in R$, $Confidence(r_i) \geq \alpha$. $\mathbb{P}$ is a set of predicates $\mathbb{P} = \{\mathbb{P}_1, ..., \mathbb{P}_m\}$ where each predicate $\mathbb{P}_i = (A\ Op\ val)$ is a single criterion such that A corresponds to an attribute name from the distributed data, Op is a comparison operator such that $Op \in \{>, \geq, <, \leq, =\}$, and operand val is a value from the domain of attribute A. If A is a categorical attribute, the data consumer has to specify the exact value, i.e., $city =$ "*Canada*". However, if A is a numerical attribute, then the data consumer has the choice of either specifying the exact value (i.e., $age = 30$) or specifying a range (i.e., $age = [30, 40]$ or $age \geq 50$).

3.2.2 ε-differentially Private Data

We assume that the data in *DRAM* is *horizontally partitioned* into sub-tables each of which is hosted by one data provider. This means that each data provider's data contains the same type of attribute information on different set of individuals. Each data provider DP_i holds a relational database D^i consisting of quasi-identifier attributes A^I, predictor attributes A^P, and class attributes A^{Class} such that: $D^i = (A^I, A^P, A^{Class})$. To protect its data from inference attacks, each data provider DP_i applies ε-differential privacy scheme [16] on its data D^i data and generates an anonymized data $\hat{D}^i$. There are many privacy models with different privacy guarantees, such as k-anonymity [20], ℓ-diversity [15], and *LKC* [17]. However, we choose to the ε-differential privacy model because it provides strong assurances against adversaries with arbitrary background knowledge.

3.3 Adversary Model

We assume that all parties in *DARM* operate under the semi-honest adversary model [13]. That is, each party is expected to follow the protocol correctly; however, it is curious and might try to infer sensitive information about the other parties. We assume that there is an authenticated secure channel between the data consumer and master miner, and another one between the master miner and each data provider. We also assume that a polynomial size circuit bounds the computational power of each adversary.

3.4 Problem Statement

Given relational data D that is horizontally partitioned into n partitions: $\{D^1, ..., D^n\}$, the objective is to design a privacy-preserving model for answering association rules queries in a distributed environment. The model must achieve three objectives: (1) to prevent each data provider from learning sensitive information about other data providers during the mining process, (2) to protect all providers against inference attacks from the data consumers, and

(3) to preserve the confidentiality of each data consumer's query against the data providers.

4. SOLUTION: DARM

4.1 Solution Overview

Our solution is a *Distributed Association Rules Mining (DARM)* model that aims at generating and delivering strong and meaningful rules to the data consumer, while preserving privacy of the data and the confidentiality of the queries. Our proposed approach is composed of three steps:

- **Step 1 - Data Anonymization**: The data providers anonymize their own data using ε-differential privacy scheme to protect against table linkage and probabilistic attacks.
- **Step 2 - Frequent Itemsets Generation**: The master miner submits count queries to the data providers corresponding to the attributes specified in the data consumer's query, and generates the frequent itemsets of different length satisfying the desired *minimum support threshold* γ.
- **Step 3 - Association Rules Generation**: After generating all related frequent itemsets, the master miner submits count queries to the data providers that enable him to generates the set of strong association rules satisfying the *minimum confidence threshold* α specified by the data consumer.

4.2 Model Architecture

One of the main characteristics of Web services solution is the *Service-oriented architecture (SOA)* it offers. In fact, Web services rely on the assumption that each functionality will be exposed as a service, which makes these services loosely-coupled as they are defined, developed, and managed by different parties [2]. Fig. 2 depicts the architecture of our proposed model, which is based on *SOA*. According to this architecture, the data consumer who learns the details of the services (i.e., attributes, location, etc) offered by the master miner by means of the *WSDL* file, uses these details to construct a *SOAP* message containing the specifications of the request and sends it to the master miner via *HTTP* protocol. Thereafter, count queries are sent by the master miner to the dat providers by means of *SOAP* messages. The providers use the *Data Manager* to query their databases, and then send back the counts to the master miner also as *SOAP* messages. The master miner uses its *Computation Manager* to compute and compare the support and confidence levels. Finally, the response (strong association rules) is delivered from the master to the data consumer as *SOAP* message via *HTTP* protocol.

Table 2: Original Table

Job	Age	Class
Teacher	25	Female
Lawyer	51	Male
Painter	48	Female
Singer	20	Female
Dancer	32	Male
Lawyer	45	Male
Writer	39	Female
Doctor	58	Female

Table 3: Root Partition

Job	Age	Sex	Count
Any Job	(20-58)	3M5F	8

4.3 Data Anonymization

In this step, the data providers use the ε-differential algorithm called *DiffGen* [16] to anonymize their data and provide protection against linkage and inference attacks. By using ε-differential, the data owner makes sure that the regenerated data table provides privacy guarantee while being insensitive to any specific record. The ε-differential privacy model [14] aims at protecting against table linkage and probabilistic attacks by ensuring that the probability distribution on the published data is the same regardless of whether or not an individual record exists in the data. The main idea of the *DiffGen* [16] algorithm is to anonymize the raw data input following a sequence of specializations starting from the topmost general state. Specialization refers to creating "partitions", each of which represents an equivalence class. *DiffGen* determines the attribute to specialize on according either to "InfoGain" or to "Max" utility functions. Basically, the anonymization process can be divided into three main parts: (1) selecting a candidate attribute for specialization, (2) determining an split value parameter, and (3) publishing the noisy counts. For example, for a raw data as shown on Table 2, the algorithm creates a root partition that contains all records as shown in Table 3. Then, assume that the first specialization will be based on the *Job* attributes. Thus, *Any_Job* is splitted into *professional* and *artist* as depicted in Tables 4 and 5. For second specialization, we could decide on splitting the age, and then specialize the table based on gender. Finally, the algorithm delivers the equivalence groups of each leaf partition along with their noisy counts, as shown in Table 6.

Table 4: First Partition

Job	Age	Sex	Count
Professional	(25-58)	2M2F	4

Table 5: Second Partition

Job	Age	Sex	Count
Artist	(20-48)	1M3F	4

Table 6: Leaf Partition

Job	Age	Sex	NoisyCount
Artist	(20-48)	F	3+1=4
Artist	(20-48)	M	1-1=0

4.4 Frequent Itemsets Generation

In this step, the master miner receives the data consumer's query, requests the support counts of all the attributes the data consumer is interested in from the different slaves, and generates all the possible frequent itemsets of different lengths subject to the minimum support threshold γ specified in the query. As depicted by Algorithm 1, the length-1 frequent itemsets are initially generated after obtaining the support counts from the providers (line 8). Then, candidates are generated from this set of frequent itemsets (line 10). The master miner requests the support counts of these candidates from the different data providers (line 11). According to these support counts, the support level of each candidate is calculated (line 12). Then,

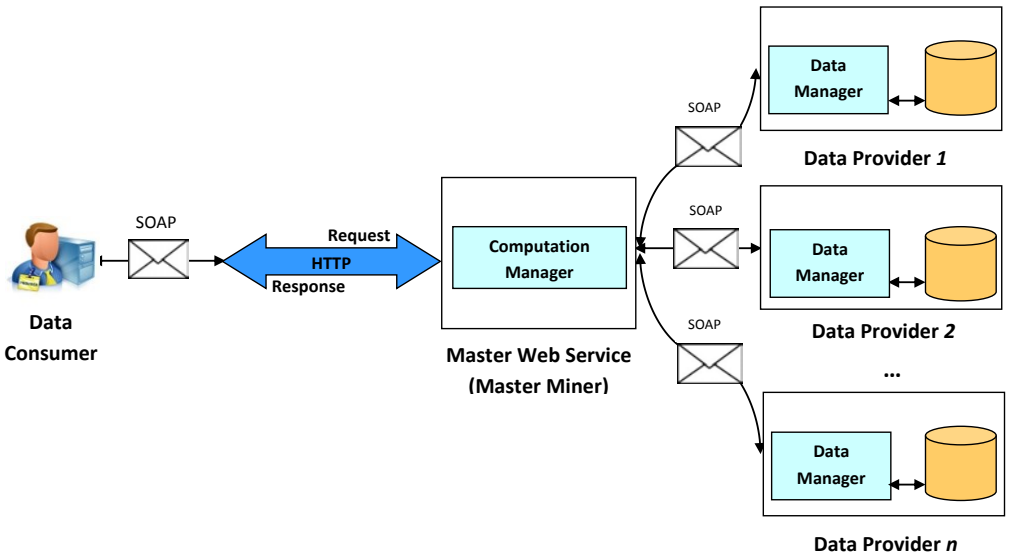

Figure 2: Model Architecture: The *Service-oriented architecture (SOA)* is used as a mean of communication among different parties

if the support level of the candidate respects the minimum support threshold γ specified by the data consumer, then this candidate is considered as frequent itemset (lines 13-15). This process is repeated until no further frequent itemsets can be found.

Algorithm 1: Frequent Itemsets Generation

```
1:  Input: minimum support threshold γ
2:  Input: set of attributes A
3:  Input: total number of rows of all the databases t
4:  Input: Candidate itemset C_k of length k
5:  Output: frequent itemsets F_k of length k
6:  procedure FREQUENTITEMSETSGENERATION
7:      C_k = ∅
8:      F_1={frequent items}
9:      for (k = 1; F_k != ∅; k++) do
10:         C_{k+1} = candidates generated from F_k
11:         request support count s(C_{k+1}) from all providers
12:         calculate Support(C_{k+1}) = s(C_{k+1})/t
13:         if Support(C_{k+1}) ≥ γ
14:             F_k = F_k ∪ C_{k+1}
15:         end if
16:     end for
17:     return ⋃_k F_k
18: end procedure
```

4.5 Association Rules Generation

Now that the frequent itemsets are known, the master miner generates all the possible combinations of the k-length ($k > 1$) frequent itemsets that may constitute association rules. It then sends these combinations to the data providers which, in their turn, calculate and send back the support counts of these combinations to the master (e.g., $\{Zip\ Code = H3M0E1,\ age \in [20,30]\} : 3$). The master miner measures the strength of the received rules subject to the *minimum confidence threshold* α specified in the data consumer query. Finally, only the strong rules are delivered to the data consumer.

As depicted in Algorithm 2, the master miner generates all the possible combinations of the length-k ($k > 1$) frequent itemsets returned by Algorithm 1 (line 7). Then, the master requests the support counts for each combination from the different providers (line 8). Based on these support counts and the support counts obtained in the previous iterations of Algorithm 1, the master computes the confidence level for each combination (line 9). Now, if the confidence level of the combination respects α, then this combination is considered as strong association rule (lines 10-13). Finally, these strong association rules are returned to the data consumer in response to his request (line 14).

Algorithm 2: Association Rules Generation

```
1:  Input: frequent itemsets F divided into L_i and R_i
2:  Input: set of left-hand-side frequent itemsets L_i
3:  Input: set of right-hand-side frequent itemsets R_i
4:  Input: minimum confidence threshold α
5:  Output: set of strong association rules R
6:  procedure ASSOCIATIONRULESGENERATION
7:      for each combination c ∈ L_i -> R_i do
8:          request support count s(c) from all providers
9:          calculate Confidence(c) = s(c)/s(L_i)
10:         if Confidence(c) ≥ α
11:             R = R ∪ c
12:         end if
13:     end for
14:     return R
15: end procedure
```

5. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed approach. First, we explain the implementation details and then we present the experimental results involving the efficiency of the approach with regard to the number of attributes specified in the data consumer query, the scalability of the overall approach with respect to the number of data records, and the sensitivity to the number of

specializations.

5.1 Implementation and Setup

We implement our approach in the 64-bit Windows 7 environment on a machine equipped with an Intel Core i7 3.80 GHz Processor and 8 GB DRAM. Oracle 11g is used as programming language to implement the different algorithms of the approach. The program is designed in a dynamic manner to support different datasets and different query inputs. The *Apriori* algorithm is used for frequent itemsets generation. As a real-life data, we use the *adult* [6] data set, which contains 45,222 census records divided into six numerical attributes, eight categorical attributes, and a class attribute with two levels: "$\leq 50K$" and "$> 50K$". More details about the attributes descriptions can be found in [10]. The number of attributes in the data consumer query can range from 2 (since no association rules can be generated from a single attribute) to 14 (the maximum number of attributes), and the average number of attributes in a query is 8. The data is horizontally partitioned over three providers in an arbitrary manner, where the first provider hosts 45% of the data, the second hosts 35%, and the third holds 20%. The idea behind making such unequal distribution is to mimic a real-life scenario, where providers host usually distinct number of data records. We generate ε-differentially private records using the *DiffGen* algorithm, where the privacy budget $\varepsilon = 1$, and the number of specializations is set to 8.

5.2 Experimental Results

5.2.1 Efficiency

To determine the efficiency of our approach, we measure the processing time of the different algorithms involved in the approach with regard to the number of attributes in the data consumer's query. The processing time is divided into two subphases: *frequent itemsets generation* and *association rules mining*. The number of specializations used is 8 and the number of attributes varies from 0 to 12. Fig. 3 reveals that the most dominant phase in our approach is generating frequent itemsets, while mining association rules has less impact on the processing time. It shows also that the total processing runtime keeps increasing linearly as the number of attributes increases. Practically, the total runtime increases from 0.1 sec to 5.1 sec when the number of attributes per query increases from 3 to 12 for frequent itemsets generation, and increases from 0.1 sec to 2.4 sec when the number of attributes per query increases from 3 to 12 for association rules mining. We observe that our proposed solution is sensitive to the number of attributes in the association rule query because adding more attributes to a query increases the possible candidates for the frequent itemsets generation algorithm and increases hence the possible number of combinations for the association rules mining algorithm.

5.2.2 Scalability

In order to study the scalability of our approach, we measure the query processing runtime with respect to the increase in the number of data records, where the number of data records linearly increases from 20,000 to 100,000 and the number of specialization is set to 8. Fig. 4 reveals that the total processing runtime increases linearly with the increase in the number of data records. Practically, the processing runtime increases from 1.8 sec for 20,000 records to 2.9 sec for 100,000 records in the frequent itemsets generation algorithm. Similarly, the processing runtime grows from 0.9 sec for 20,000 records to 2 sec for 100,000 records in the association rules mining algorithm. Therefore, we conclude that our proposed approach is scalable w.r.t. the data size.

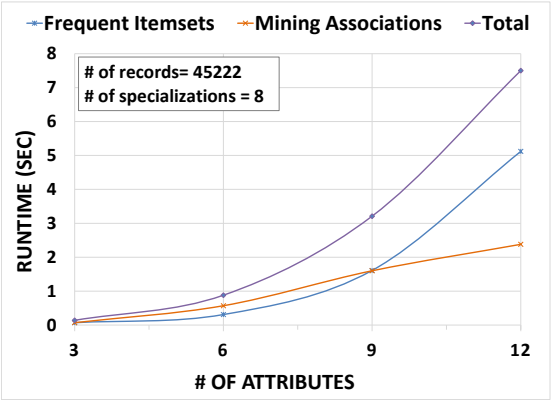

Figure 3: Efficiency w.r.t. the number of attributes per a query for frequent itemsets generation phase, and association rules mining phase.

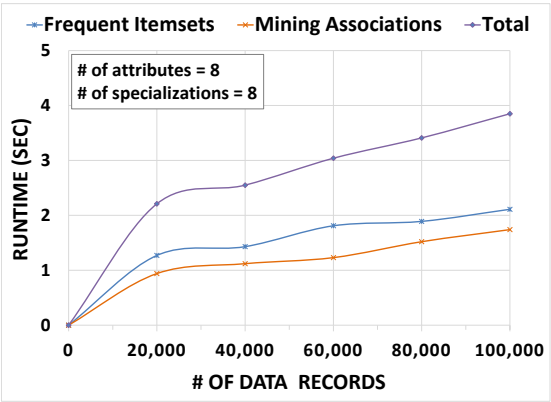

Figure 4: Scalability of query processing w.r.t. the number of data records in the database for frequent itemsets generation phase, and association rules mining phase.

5.2.3 Efficiency w.r.t. nSpecializations

In the differential privacy algorithm *DiffGen* [16], *nSpecializations* is the number of specializations parameter that determines when the algorithm should terminate. The higher the *nSpecializations* value is, the more partitions (anonymized records) are generated, which obviously has an impact on the runtime. Our goal here is to determine the efficiency of our solution with the linear increase of number of specializations. The number of raw data records used is 40,000, the *minimum support threshold* is 30%, and the *minimum confidence threshold* is 40%. Similar to the efficiency test, Fig. 5 reveals that the most dominant phase in our approach is generating frequent itemsets, while mining association rules has less impact on the processing time. It shows also that the total processing runtime slightly increases as the number of specializations increases linearly, up to 14. However, the total runtime increases from 6.4 sec to 12 sec when the number of specializations increases from

14 to 16. We therefore conclude that our proposed solution is efficient w.r.t. the number of specialization *nSpecializations*, when *nSpecializations* is less or equal to 14.

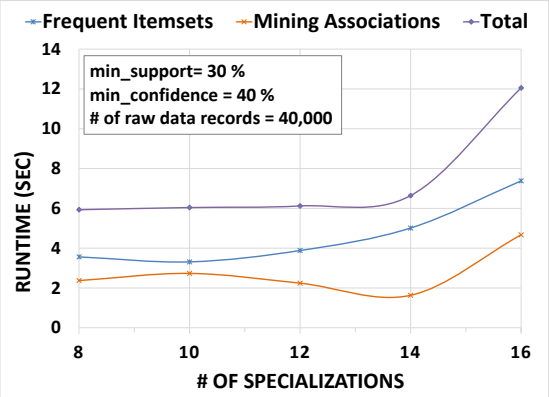

Figure 5: Efficiency w.r.t to the number of specializations during the anonymization process using *DiffGen* algorithm.

6. CONCLUSIONS

In this paper, we propose a comprehensive privacy-preserving approach for answering association rules queries in a distributed environment, with the goal of preserving both data privacy and query confidentiality. The proposed approach (1) protects all providers against inference attacks from data consumers by guaranteeing that the returned association rules to the data consumer satisfy ε-differential privacy, (2) preserves the privacy of the mined data by preventing each data provider from learning sensitive information about other data providers during the mining process, and (3) protects the confidentiality of the data consumer's query against the data providers such that the master miner is able to mine the association rules without revealing the query to the data providers. Experimental results reveal that our approach is efficient w.r.t. the increase in the number of attributes in the data consumer's query, and scalable w.r.t. the number of records in the dataset.

7. REFERENCES

[1] R. Agrawal and J. C. Shafer. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):962–969, Dec. 1996.

[2] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, 1st edition, 2010.

[3] A. Anitha, G. R. Suhanantham, and N. Krishnan. An efficient association rule mining model for distributed databases. *International Journal of Computer Science and Technology*, 3(1):794–797, 2002.

[4] M. Arafati, G. G. Dagher, B. C. M. Fung, and P. C. K. Hung. D-mash: A framework for privacy-preserving data-as-a-service mashups. In *Proceedings of the 8th IEEE International Conference on Cloud Computing (CLOUD)*, June 2014.

[5] M. Z. Ashrafi, D. Taniar, and K. Smith. Odam: an optimized distributed association rule mining algorithm. *IEEE Distributed Systems Online*, 5, 2004.

[6] K. Bache and M. Lichman. Uci machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2013.

[7] D. W. Cheung, J. Han, V. T. Ng, A. W. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In *Proceedings of the 4th International Conference on on Parallel and Distributed Information Systems (DIS)*, pages 31–43, 1996.

[8] J. L. Dautrich, Jr. and C. V. Ravishankar. Compromising privacy in precise query protocols. In *Proceedings of the 16th International Conference on Extending Database Technology (EDBT)*, pages 155–166, 2013.

[9] B. C. M. Fung, K. Wang, A. W.-C. Fu, and P. S. Yu. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Data Mining and Knowledge Discovery. August 2010.

[10] B. C. M. Fung, K. Wang, and P. S. Yu. Anonymizing classification data for privacy preservation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(5):711–725, May 2007.

[11] F. Giannotti, L. Lakshmanan, A. Monreale, D. Pedreschi, and H. Wang. Privacy-preserving mining of association rules from outsourced transaction databases. *IEEE Systems Journal*, 7(3):385–395, Sept 2013.

[12] P. Gurunathan, N. Ishwarya, V. Sridevi, C. Nandhini, and S. Deepalakshmi. High-dimensional confidential data mash up using service- oriented architecture. *International Journal of Emerging Science and Engineering (IJESE)*, 1(6), April 2013.

[13] S. Kamara, P. Mohassel, and M. Raykova. Outsourcing multi-party computation. *IACR Cryptology ePrint Archive*, 2011:272.

[14] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1026–1037, Sept. 2004.

[15] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[16] N. Mohammed, R. Chen, B. C. M. Fung, and P. S. Yu. Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 493–501, August 2011.

[17] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C. Lee. Anonymizing healthcare data: A case study on the blood transfusion service. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1285–1294, June 2009.

[18] J. S. Park, M.-S. Chen, and P. S. Yu. Efficient parallel data mining for association rules. In *Proceedings of the 4th International Conference on Information and Knowledge Management (CIKM)*, pages 31–36, 1995.

[19] J. Renjit and K. Shunmuganathan. Mining the data from distributed database using an improved mining algorithm. *International Journal of Computer Science and Information Security*, 7(3):116âĂŞ–121, 2010.

[20] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, Oct. 2002.

[21] T. Trojer, B. C. M. Fung, and P. C. K. Hung. Service-oriented architecture for privacy-preserving data mashup. In *Proceedings of the 7th IEEE International Conference on Web Services (ICWS)*, pages 767–774, July 2009.

[22] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 639–644, 2002.

[23] J. Vaidya and C. Clifton. Privacy-preserving data mining: why, how, and when. *IEEE Security Privacy*, 2(6):19–27, Nov 2004.

[24] W. K. Wong, D. W. Cheung, E. Hung, B. Kao, and N. Mamoulis. Security in outsourcing of association rule mining. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB)*, pages 111–122, 2007.

[25] N. Zhang, M. Li, and W. Lou. Distributed data mining with differential privacy. In *Proceedings of the IEEE International Conference on Communications(ICC)*, pages 1–5, 2011.