International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 12 Issue 1–DECEMBER 2014.

Secure Mining of Association Rules in Distributed Database

R.Snega[*1], D.Ananthanayaki[#2] and K.K.Kavitha[#3]

[1]*PG Scholar (M.Phil-CS), Selvamm Arts and Science College (Autonomous), Namakkal, Tamilnadu, India*
[2]*Assistant Professor (CS), Selvamm Arts and Science College (Autonomous), Namakkal, Tamilnadu, India*
[3]*Assistant Professor (CS), Selvamm Arts and Science College (Autonomous), Namakkal, Tamilnadu, India*

Abstract— Association rule mining is an active data mining research area and most ARM algorithms cater to a centralized environment. Centralized data mining to discover useful patterns in distributed databases isn't always feasible because merging data sets from different sites incurs huge network communication costs. In this paper, an improved algorithm based on good performance level for data mining is being proposed. Local Site also finds a centre site to manage every message exchanged to obtain all globally frequent item sets. It also reduces the time of scan of partition database. The problem of computing efficient anonymization of partitioned databases. Given a database that is partitioned between several sites, either horizontally or vertically, we devise secure distributed algorithms that allow the different sites to obtain a k-anonymize and ℓ-diverse view of the union of their databases, without disclosing sensitive information. Without leaking any information about their inputs except that revealed by the algorithm's output. Working in the standard secure multi-party computation paradigm, we present new algorithms for privacy-preserving computation of APSD (all pairs shortest distance) and SSSD (single source shortest distance), as well as two new algorithms for privacy-preserving set union. We prove that our algorithms are secure provided the participants are "honest, but curious."

Keywords: Secure Multiparty Computation, privacy-preserving, databases partitioning.

I. INTRODUCTION

Most existing parallel and distributed ARM algorithms are based on a kernel that employs the well-known Apriori algorithm [1]. Directly adapting an Apriori algorithm will not significantly improve performance over frequent item sets generation or overall distributed ARM performance. In distributed mining, synchronization is implicit in message passing, so the goal becomes communication optimization. Data decomposition is very important for distributed memory[2]. Therefore, the main challenge for obtaining good performance on distributed mining is to find a good data decomposition among the nodes for good load balancing, and to minimize communication. Protecting the privacy of the individuals whose private data appear in those repositories is of paramount importance. Al- though identifying attributes such as names and ID numbers are always removed before releasing the table for data mining purposes, sensitive information might still leak due to linking attacks; such attacks may join the public attributes, a.k.a quasi-identifiers, of the published table with a publicly accessible table like the voters registry, and thus disclose private Information of specific individuals.

Privacy-preserving data mining [3] has been proposed as a paradigm of exercising data mining while protecting the privacy of individuals. One of the well-studied models of privacy preserving data mining is k-anonymization [4,5]. Trusted third party, each site could surrender to that third party his part of the database and trust the third party to compute an anonymization of the unified database. Without such a trusted third party, the goal is to devise distributed protocols, for the horizontal and vertical settings, that allow the data holders to simulate the operation of a trusted third party and obtain a k-anonymized and ℓ-diverse view of the union of their databases, without disclosing unnecessary information to any of the other parties, or to any eavesdropping adversary. In this paper, we construct privacy-preserving versions of classic graph algorithms for APSD (all pairs shortest distance) and SSSD (single source shortest distance). Our algorithm for APSD is new, while the SSSD algorithm is a privacy preserving transformation of the standard Dijkstra's algorithm. We also show that minimum spanning trees can be easily computed in a privacy-preserving manner.

II. RELATED WORK

This paper follows a long tradition of research on privacy-preserving algorithms in the so called secure multiparty computation (SMC) paradigm. Informally, security of a protocol in the SMC paradigm is defined as computational indistinguishability from some ideal functionality, in which a trusted third party accepts the parties' inputs and carries out the computation. The ideal functionality is thus secure by

definition. The actual protocol is secure if the adversary's view in any protocol execution can be simulated by an efficient simulator who has access only to the ideal functionality, i.e., the actual protocol does not leak any information beyond what is given out by the ideal functionality. In this paper, we aim to follow the SMC tradition and provide provable cryptographic guarantees of security for our constructions. Another line of research has focused on statistical privacy in databases, typically achieved by randomly perturbing individual data entries while preserving some global properties. A survey can be found in The proofs of security in this framework are statistical rather than cryptographic in nature, and typically permit some leakage of information, while supporting more efficient constructions. In this paradigm, Clifton et al. have also investigated various data mining problems, while Du et al. researched special-purpose constructions for problems such as privacy-preserving collaborative scientific analysis. Recent work by Chawla et al. aims to bridge the gap between the two frameworks and provide rigorous cryptographic definitions of statistical privacy in the SMC paradigm. Another line of cryptographic research on privacy focuses on private information retrieval (PIR), but the problems and techniques in PIR are substantially different from this paper.

III. Definition of Privacy

We use a simplified form of the standard definition of security in the static semi-honest model due to Goldreich (this is the same definition as used, for example, by Lindell and Pinkas).

Definition 1. Protocol π securely computes deterministic functionality f in the presence of static semi-honest adversaries if there exist probabilistic polynomial time simulators S1 and S2 such that

$\{S1(x, f(x, y))\}x,y\in\{0,1\}*\equiv c \{view\pi1(x, y)\}x,y\in\{0,1\}*$

$\{S2(y, f(x, y))\}\ x,y\in\{0,1\}* \equiv c \{view\pi2\ (x, y)\}\ x,y\in\{0,1\}*$

where $|x| = |y|$.

Informally, this definition says that each party's view of the protocol can be efficiently simulated given only its private input and the output of the algorithm that is being computed (and, therefore, the protocol leaks no information to a semi-honest adversary beyond that revealed by the output of the algorithm).

IV. Anonymization by generalization

Consider a database that holds information on individuals in some population. Each record in the database has several attributes, and we distinguish between identifiers, quasi-identifiers, and sensitive attributes. Identifiers are attributes that uniquely identify the individual, e.g. name or id. Quasi-identifiers are attributes, such as age or zip code that appear also in publicly-accessible databases and may be used in order to identify a person. The sensitive attributes are those that carry private information like a medical diagnosis or the salary of the person. k-Anonymity is a model that was proposed in order to prevent the disclosure of sensitive attributes for the purpose of protecting the privacy of individuals that are represented in the database. we view the database records as elements in $A1 \times \bullet\bullet\bullet \times Ad \times Ad+1$, where Aj is the set of possible values for the jth attribute; say, if the jth attribute is gender then Aj = {M, F}. Hereinafter, D denotes the projection of the database on the set of d quasi-identifiers and the records of D are denoted Ri, $1 \leqslant i \leqslant n$; namely, $Ri \in A1 \times \bullet\bullet\bullet \times Ad$. We denote the jth component of the record Ri by Ri(j). Also, for any set A we let P(A) denote its power set. Next, we define the notion of generalization. Definition 2.1. Let Aj , $1 \leqslant j \leqslant d$, be finite sets and let $Aj \subseteq P(Aj)$ be a collection of subsets of Aj . A mapping $g : A1 \times \bullet\bullet\bullet \times Ad \rightarrow A1 \times \bullet\bullet\bullet \times Ad$ is called a generalization if for every $(b1, \ldots, bd) \in A1 \times \bullet\bullet\bullet \times Ad$ and $(B1, \ldots, Bd) = g(b1, \ldots, bd)$, it holds that $bj \in Bj$, $1 \leqslant j \leqslant d$. As an example, consider a database D with two attributes, age (A1) and zipcode (A2). A valid generalization of the record Ri = (34, 98003) can be g(34, 98023) = ({30, . . . , 39}, {98000, . . . , 98099}) . We assume here that each of the collections Aj is a generalization hierarchy tree for Aj , $1 \leqslant j \leqslant d$. Such a tree has $|Aj|$ leaves – one for each singleton subset of Aj ; the root corresponds to the whole set; and the subset of each node is the union of the subsets that correspond to the direct descendants of that node. Definition 2.1 refers to generalizations of single records. We now define generalizations of an entire database.

4.1 Private Single Source Shortest Distance (SSSD)

The Single Source Shortest Distance (SSSD) problem is to find the shortest path instances from a source vertex s to all other vertices [11]. An algorithm to solve APSD also provides the solution to SSSD, but leaks additional information beyond that of the SSSD solution and cannot be considered a private algorithm for SSSD. Therefore, this problem warrants its own investigation. Similar to the protocol of section 5.1, the SSSD protocol on the minimum joint graph adds edges in order from smallest to largest. This protocol is very similar to Dijkstra's algorithm, but is modified to take two graphs as input.

1. Set w (0) 1 = w1 and w (0) 2 = w2. Color all edges incident on the source s blue by putting all edges esi into the set B (0). Set the iteration count k to 1.

2. Both parties privately compute the minimum length of blue edges in their graphs. $m(k) 1 = \min esi\in B(k-1)\ w(k-1)\ 1\ (esi)$, $m(k) 2 = \min esi\in B(k-1)\ w(k-1)\ 2\ (esi)$

3. Using the privacy-preserving minimum protocol, compute

m(k) = min(m(k) 1 ,m(k) 2).

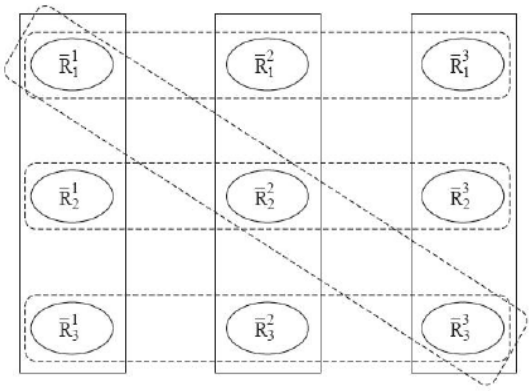
4. Each party finds the set of blue edges in its graph with length m(k). S(k) 1 = {esi|w(k−1) 1 (esi) = m(k)}, andS(k) 2 = {esi|w(k−1) 2 (esi) = m(k)}

5. Using the privacy-preserving set union protocol, compute S(k) = S(k) 1 ∪ S(k)2

4.2 The horizontal setting

The only interaction between the players in the horizontal setting is for computing the size and closure of clusters (as described in Section 4), and computing the distribution of the sensitive values in each cluster (Section 6.1). During the protocol, the players may learn information on records held by other players, which is not implied by their own input and the final output. Therefore, the protocol is not perfectly secure in the cryptographic sense. Such a compromise is widely acceptable since, as written in, "allowing innocuous information leakage allows an algorithm that is sufficiently secure with much lower cost than a fully secure approach". Indeed, many distributed protocols accept innocuous information leakage for gaining efficiency, utility and practicality. We proceed to characterize herein the excessive information that is leaked, compare it to information leakage in other protocols, and argue that such a leakage of information is benign from practical point of view. We separate our discussion to three types of information that the players may learn on the private data of other players. Assume that the different players are hospitals and the partial database of Hospital i, $1 \leqslant i \leqslant m$, holds information on the patients in that hospital. One of the participating hospitals may be interested to know whether a particular individual, Alice, was hospitalized in one of the other hospitals. Using Alice's publicly accessible quasi-identifier values, which hospital may try to examine his view of the protocol in order to deduce the answer? More generally, the hospital may wish to learn how many people from a given age range and location took part in the other databases. In Section 8.2.1 we explain why such inferences are hard and sometimes even impossible to extract from the protocol's views. Alternatively, it is possible that one hospital knows that Alice was hospitalized in another participating hospital, but it wishes to know her sensitive value. In Section 8.2.2 we explain why it is impossible to extract such information beyond what is implied by the final k-anonymized and ℓ-diversified anonymization. Finally, it is possible that hospitals will aim at learning information on the number of patients in the other hospitals. we explain how to hide also that information. (To the best of our knowledge, no other study dealt with the question of hiding the size of the partial databases.) Information on the quasi-identifiers of records of other players discuss possible inferences that the players may make on the quasi-identifier values of records of other players. In the first part of this section we show that any attempt to infer information about the inclusion of a given quasi-identifier record, R = (R(1), . . . ,R(d)), in the unified database D is useless. Then, we proceed to characterize the significantly weaker type of information leakage on the quasi-identifier values of records in D

Example 1. Some specific record R. Hence, those records are connected by an hyperedge if they could all be the generalized view of the same original record in D. Example 1. Consider the table D in Table 2 that has d = 3 quasi-identifier attributes, A1 = {a, b}, A2 = {x, y} and A3 = {1, 2}. Assume that during the distributed protocol, the players constructed p = 3 anonymized views of D as shown in Table 2. The corresponding hypergraph GD is shown in Figure 1. It has four hyperedges: The three hyperedges that correspond to the three real records in D, and a fourth artifact hyperedge. The first hyperedge is {R1,R21,R31}, since all those generalized records generalize the record R1 ∈ D. The sets {R1 2,R2 2,R3 2 } and{R13,R23,R33} are two additional hyperedges, corresponding to R2,R3 ∈ D. The fourth hyperedge is {R11,R22,R33}. All three records in that hyperedge indeed generalize the same record — (a, x, 2). However, as opposed to the first three hyperedges (which generalize a true record in D), that latter record is an artifact one that does not appear in D.



D	$\overline{D}_1$	$\overline{D}_2$	$\overline{D}_3$
$R_1=(a,x,1)$	$\overline{R}_1^1=(a,x,*)$	$\overline{R}_1^2=(*,x,1)$	$\overline{R}_1^3=(a,*,1)$
$R_2=(b,x,2)$	$\overline{R}_2^1=(b,x,*)$	$\overline{R}_2^2=(*,x,2)$	$\overline{R}_2^3=(b,*,2)$
$R_3=(a,y,2)$	$\overline{R}_3^1=(a,y,*)$	$\overline{R}_3^2=(*,y,2)$	$\overline{R}_3^3=(a,*,2)$

Table 2. A table D and three anonymized views

Figure 1. The hypergraph corresponding to the three anonymized views in Table 2

V. CONCLUSION

In this paper, we presented privacy-preserving protocols that enable two honest but curious parties to compute APSD and SSSD on their joint graph. A related problem is how to construct privacy- preserving protocols for graph comparison. Many of these problems (e.g., comparison of the graphs' respective maximum flow values) reduce to the problem of privacy-preserving comparison of two values, and thus have reasonably efficient generic solutions. For other problems, such as graph isomorphism, there are no known polynomial-time algorithms even if privacy is not a concern. Investigation of other interesting graph algorithms that can be computed in a privacy-preserving manner is a topic of future research. In conclusion, we presented a general approach to secure distributed computations of anonymized views of shared databases. The presented algorithms are highly efficient and simple, as they rely on very basic and few cryptographic primitives. Even though we focused here on distributed versions of one particular algorithm(sequential clustering) and one particular goal (anonymization), the ideas and techniques that were presented here are suitable for any other algorithm that reorganizes clusters (like simulated annealing or k-means) and could be applicable also for other distributed data mining problems.

REFERENCE

[1] T. Tassa and E. Gudes. Secure distributed computation of anonymized Views of shared databases. Transactions on Database Systems, 37,Article 11, 2012.
[2] J. Brickell and V. Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In ASIACRYPT, pages 236–252, 2005.
[3] D.W.L Cheung, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. Efficient mining of association rules in distributed databases. IEEE Trans. Knowl. Data Eng., 8(6):911–922, 1996.
[4] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth-ranked element. In EUROCRYPT, 2004.
[5] R. Agrawal and R. Srikant. Privacy-preserving data mining. In ACM-SIGMOD Conference on Management of Data, pages 439–450, May 2000.
[6] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In ICDE, 2005.
[7] D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In STOC, pages 503–513, 1990.
[8] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Towards privacy in public databases. In Proc. 2nd Theory of Cryptography Conference (TCC), volume 3378 of LNCS, pages 363–385. Springer-Verlag, 2005.
[9] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval.J. ACM, 45(6):965–981, 1998.
[10] D.W. Cheung, et al., "A Fast Distributed Algorithm for Mining Association Rules," Proc. Parallel and Distributed Information Systems, IEEE CS Press, 1996,pp. 31-42;
[11] M.J. Zaki and Y. Pin, "Introduction: Recent Developments in Parallel and Distributed Data Mining," J. Distributed and Parallel Databases, vol. 11, no. 2, 2002, pp. 123-127.

AUTHOR'S DETAILS

R.Snega, Student, Department of Computer Science, Selvamm Arts and Science College (Autonomous). sneha.sri92@gmail.com

Mrs. D.Anantha nayaki, MCA., M.Phil., works as Assistant Professor in Selvamm Arts and Science College, Namakkal, India. Her Field of interest is Data Mining, Oracle. She has 7 years of experience in teaching ananthu.sasc@gmail.com

Ms.K.K.Kavitha, MCA., M.Phil., SET, works as Assistant Professor in Selvamm Arts and Science College, Namakkal, India. Her Field of interest are Data Mining, Soft Computing. She has 12 years of experience in teaching. kavithakkcs@gmail.com