Privacy Preserving Data Mining in Distributed System using RDT Framework

Kripa Joy
Department of Computer Science & Engineering
Mar Athanasius College of Engineering Kothamangalam, Kerala

Aparnasree R
Department of Computer Science & Engineering
Mar Athanasius College of Engineering Kothamangalam, Kerala

Linda Sara Mathew
Assistant Professor
Department of Computer Science & Engineering
Mar Athanasius College of Engineering Kothamangalam, Kerala

Abstract

Distributed data is very important in modern information driven applications. Most of the applications are using distributed databases because of the availability of data in different databases. Use of distributed data is very challenging because of the difficulty of merging the data which are more private. Without losing the privacy of data each application need to maximize the utility of the collected information. Using only local data will not give an optimal utility of the data. In this case techniques for privacy-preserving knowledge discovery is very important. Existing techniques for privacy-preserving data mining are cryptography based techniques and perturbation based technique. Cryptographic techniques are too slow for large scale data sets. Perturbation based technique doesn't give a much privacy for the data which are distributed. Random Decision Tree framework can used for privacy preserving data mining. Random decision trees (RDT) shows that it is possible to generate equivalent and accurate models with much smaller cost and it is very suitable for parallel and fully distributed architecture.

Keywords: Random Decision Tree (RDT), Distributed System, classification

I. INTRODUCTION

Building a data mining model will be an easy task for locally placed data. But in the case of distributed systems it is not that much easy to create a model for data mining since all the data is not present in the same system. Privacy concerns restrict the data distributed in different systems are merged together. Privacy preserving data mining is the best method to solve this problem. For privacy preserving data mining two techniques are there perturbation based technique and cryptographic technique. In perturbation based technique data is added with some noise from global distribution. And send that perturbed data to the data miner. Data mining retrieve the value and done the data mining process. Here the problem is, it doesn't give much better security to the private data. Cryptographic techniques use the encryption algorithm for the data security. If the data set is large it is very difficult to encrypt the complete data

Proposed solution uses both randomization and cryptographic technique. The technique will provide improved efficiency and security for several decision tree based learning task. Proposed system is based on the Random Decision Tree framework (RDT). Multiple trees will have created in RDTs and these trees will be very useful for various learning tasks like classification. RDT is very good frame work for classification model in distributed system, because of the randomness in structure compared to the traditional perturbation based method. For privacy preserving the property of the RDT to generate trees that are random in structure are exploited. In classification method RDT creates different decision trees and structure and the leaf nodes are alone encrypted in order to share the information. Cost of the creation of tree is very less.

Proposed technique is used for the classification of the distributed data using RDT. In Distributed systems data is distributed as horizontally partitioned and vertically partitioned. In both cases RDT create random decision trees based on the available information in each sites. Structure of the random decision tree is completely independent to the training data set. Creation of the tree uses only the attributes name and the constraints of the attributes. Therefore, the dataset in any site will not be disclosed to the other sites.

II. RELATED STUDIES

There are lot many works done in the fields of privacy preserving data mining, Random Decision Tree, partitioning od database and in the fields of Encryption. In the field of privacy preserving data mining, the major contribution was done in [1], [10], and [14]. In [10], talks about the tree building from perturbed training data records, were original data values appear different. To estimate the original value, they find a novel reconstruction procedure. In [1] computational models are introduced to perform privacy preserving data mining.

While talking about privacy preserving data mining in the horizontally and vertically partitioned data, in [5], [21], [22], and [28], have the earlier works to crack the issue of improving security. In horizontally partitioned data the method used was distorting the data values. They proposed efficient method to generate approximation to original data distribution, by not revealing original data

In [5], the vertically partitioned data classification is detailed. A two party algorithm is introduced which efficiently discovers frequent patterns without disclosing much information about the transaction values. Privacy preserving association rule mining algorithm is introduced, which have a privacy preserving scalar product protocol.

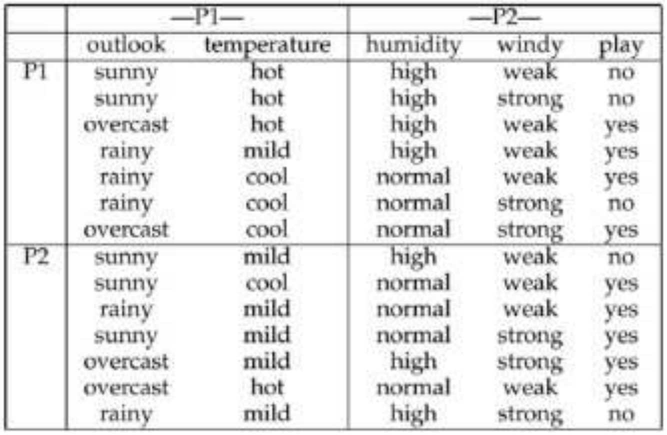
In the area of random decision tree the major works were done in [2], [3], [4], [12] and [13]. In [2], the detailed description about the benefits of Random Decision tree frame work. The system proposes that multiple random iso-depth decision trees can effectively classify the tuples.

III. PROPOSED METHODOLOGY

In this section the data set used, as well as the methods for the construction of the random decision tree for vertically and horizontally partitioned data and the classification method is introduced.

A. *Data Set:*

We can use any data set in our Random Decision Tree Framework. The famous weather data set is used for the proposed system here. Weather data set have 5 attributes and 14 transactions for those attributes. Data set is partitioned vertically and horizontally. Both the vertically partitioned data and horizontally partitioned data stored in different data bases in different systems.



	—P1—		—P2—		
	outlook	temperature	humidity	windy	play
P1	sunny	hot	high	weak	no
	sunny	hot	high	strong	no
	overcast	hot	high	weak	yes
	rainy	mild	high	weak	yes
	rainy	cool	normal	weak	yes
	rainy	cool	normal	strong	no
	overcast	cool	normal	strong	yes
P2	sunny	mild	high	weak	no
	sunny	cool	normal	weak	yes
	rainy	mild	normal	weak	yes
	sunny	mild	normal	strong	yes
	overcast	mild	high	strong	yes
	overcast	hot	normal	weak	yes
	rainy	mild	high	strong	no

Fig. 4.1: The Distributed Weather Data Set

B. *Construction of Horizontally Partitioned Tree:*

Here data is horizontally partitioned therefore each sites have the schema of the data set. So each site can independently create the random decision trees. For random tree creation it needs to select the random attribute from the given set of data and set it as root of the tree. Again select the random attribute for the construction of the intermediate node. Once the depth of the tree is become n/2 tree creation will stop and update the structure of the tree by populating the data in

The training data set. Only the information of the leaf node is sharing between the sites.

Architecture of RDT construction in Horizontal dataset

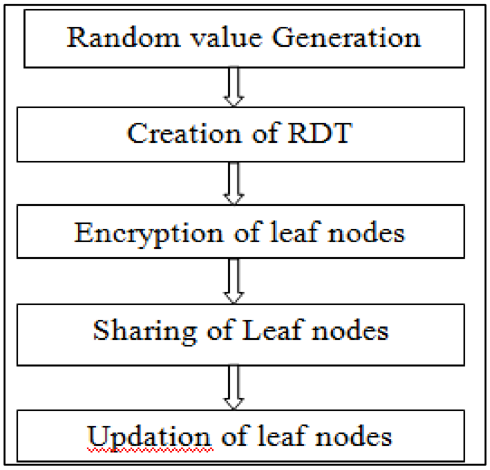


Fig. 4.2: Proposed Architecture Horizontal

1) Algorithm 1BuildTreeHorizontal

- Input: Transaction set T is partitioned horizontally between sites p1 to pkNumber of attributes and the attribute's name and it's constraints
- Output: number of RDTs created by each participant.

```
Begin
 1) Random value generation
 2) For every attributes {
     1) level = 1;
     2) depth = total number of attributes / 2;
     3) Number of trees= number of root nodes +1;
     4) createNode(level, depth);
     5) updateHorizStats(rootNode);
 3) end for
 4) Each party Pi locally computes the class distribution vectors for each node.
 5) Each party encrypts the class distribution vector for all leaf node using homomorphic encryption.
 6) Stop
Return the m number of trees
```

C. *Encryption:*

Once the trees generated we need to share the information of the leaf node alone. In order to share the leaf nodes cryptographic method is used for privacy preserving. Here Homomorphic encryption is used. Homomorphic encryption is a form of encryption that allows computations to be carried out on ciphertext, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext.

This is sometimes a desirable feature in modern communication system architectures. Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services. Since we are using the homomorphic encryption no need to decrypt the value of data. We just need to add the encrypted values. Addition of encrypted value give the same result for comparison as the result of the count of original data.

For encryption we have some key values like secret key and public key. For generating the secret key we need to set one random value which will be a long integer it is denoted by p. Get the system time and take the modulo of that ,system time %1000 denoted by tm. For getting the secret key

$$Skey=p+tm$$

Once the secret key is generated corresponding public key should be generated. For that we need some keys like Skey, q, x, r.For generating the Public key,

$$PKey= (SKey * q) + (x * r) \text{ where } q>x>r$$

Public key is used for the encryption of the plain text. Secret key is used to decrypt the encrypted value. Here no need of decryption since we are using only the sum of count of class attributes.

1) Algorithm 2 Encryption

```
Begin
    1) Get the system time as tm
    2) tm = t % 1000;
    3) Generate secret key p = 985745000
    4) Skey = p + tm;
    5) Assign some values q = 825212, m = 1000;
    6) Calculating the public key, No. of public key= 1
    7) PKey = (p * q) + (x * r);
    8) returnSkey+"#"+PKey;
    9) End
```

D. *Sharing of Leaf Nodes:*

Leaf values are encrypted and the encrypted leaf values will broadcast to all the sites which are up in the network. While sharing the leaf values the entire structure of the tree and the class (leaf nodes have class) value count will be shared. Using that structure and the value each sites will update their leaf node.

E. *Updating of Leaf Nodes*

Encrypted values are shared with all other sites and using those values each sites will update their RDTs . Each sites have different trees depend upon the structure of tree. But all the structure will be present in all sites. Sometimes the number of attributes is larger than we will create only a certain number of tress and it will be less than 10. If any of the structure is not acceptable by any other parties that tree structure will be discarded. Whenever a new site send the structure and class distribution vector all other sites will update their leaf nodes by taking the count of the class constraints. For example, in whether data set class attribute is play and value of that class attribute is YES and NO.

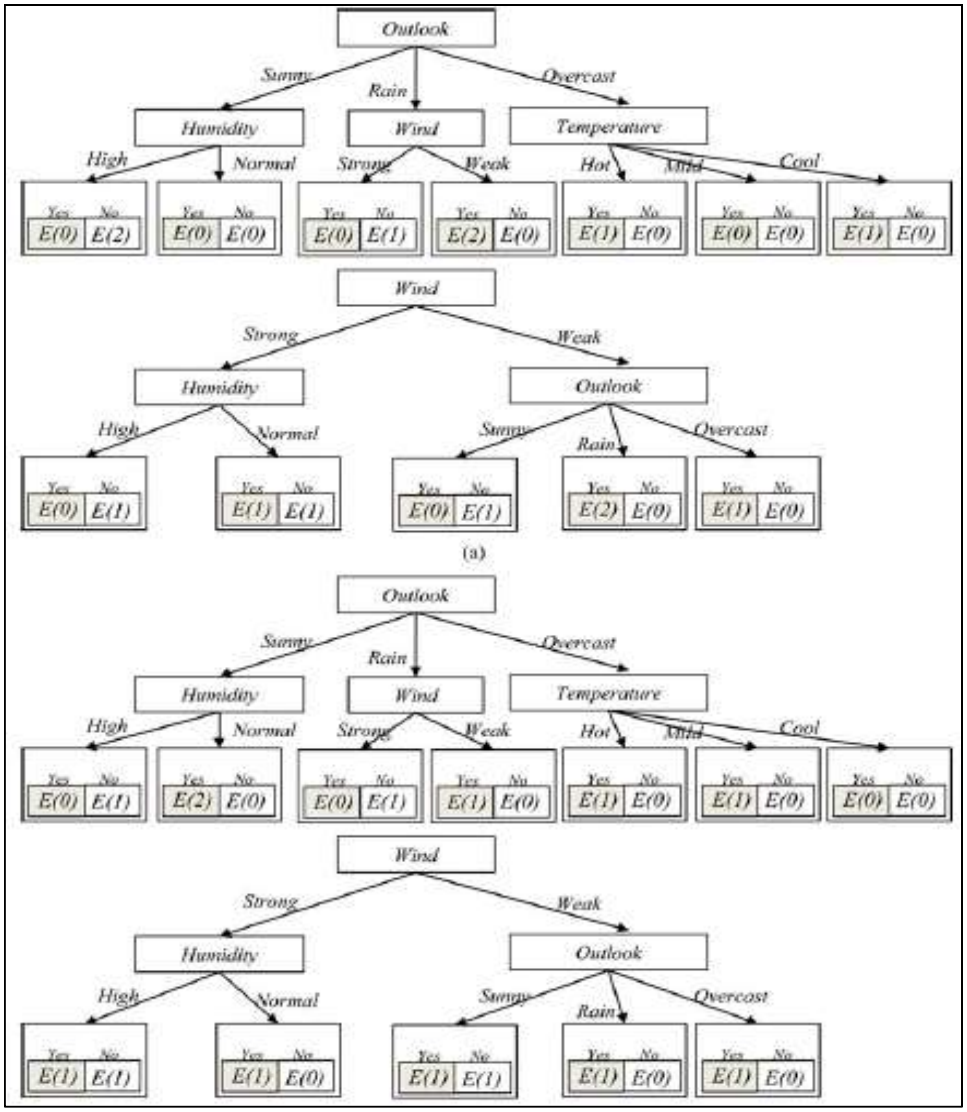


Fig. 3: Random Decision tree for Horizontally Partitioned Data

F. *RDT for Vertically Partitioned Data*

For vertically partitioned data since the attributes present in different sites are different. For creating the RDT, primary site need to select one of the participating sites and that site will select one of its attributes and set that node as root node. Generate one key value for that attribute by using cryptography and send it to any other randomly selected site. Same process will do until the level of the tree is equal to n/2.

For Vertically partitioned data creation of the tree is more complicated because of the attributes are distributed in different sites. Therefore, two algorithms are used for the construction of RDT

1) Algorithm 3: Create RDT for vertically partitioned data

```
Input: Ni attributes are holding Pi parties, p class value c1… cp with Pk holding the class   attributes,
Output: m, number of RDTs
Begin
 1)  All parties together compute the total number of attribute, n.
 2)  Select one primary node.
 3)  depth=n/2{depth of the random decision tree}
 4)  fori=1..m {build the ith tree}do
      1)  level=1
      2)  nodeId= Buildtree(level , depth)
 5)  end for
End
```

2) *Algorithm 4: BuildTree(Level, Depth)*

```
Begin
 1)  if level<=depth then
     1)  Primary node will randomly choose one site Pi
     2)  Randomly choose one attribute r from Pi
     3)  At Pi create interior node Nd with attribute Nd.A<-Ar(the rth attribute)
     4)  For each attribute value ai  do
     –   update local constraints t0uple
     –   nodeId= BuildTree(level, depth)
     –   add appropriate branch to interior node
     5)  end for
     6)  For each randomly chosen value generate an encrypted key
     7)  Pass the encrypted key inorder to find the parent of the next attribute
 2)  Else if
     1)  Update the leaf node with class attribute
End
```

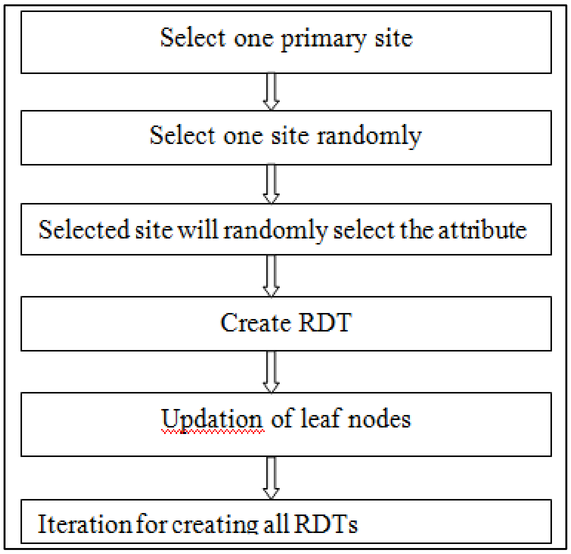

Select one primary site
Select one site randomly
Selected site will randomly select the attribute
Create RDT
Updation of leaf nodes
Iteration for creating all RDTs

Fig. 4.6: Proposed Architecture

For Updating leaf node, Once Tree structure is created leaf node will update by the class attribute present one of its site. Each branch will have updated by the transaction present in the site by traversing from the root to the leaf. One tree is updated we need to again create the next vertical tree and update the leaf node of that too.

G. *Classification:*

1) *Algorithm 5ClassificationHorizontal*

```
Input: An instance like {sunny,mild,normal,weak}
Output: Class value
Begin
     1)  For each value in the instance
     2)  Traverse from the root node of each tree present in the site
     3)  Take the count of each value in the leaf node(class value)
     4)  Add the count of class value with other trees in the same site.
     5)  Return the class with larger count as class
End
```

IV. EXPERIMENTAL RESULTS

Random Decision tree is useful for the distributed data systems. Here we have used data set whether dataset. Performance evaluation is done between Random Decision Tree and Cryptographic methods. We have also done the comparative study between RDT and the ID3 methods which is based on the information gain for decision tree creation.

When compared with the cryptographic techniques we count the number of operations in the cryptographic methods like encryption, decryption and the transfer of complete encrypted data set. Encryption of complete data set and the transfer of that take much time compared with the RDT. In RDT we are not encrypting the complete data set, encrypting only the leaf nodes. Decryption is not done at the receiver side because we are taking only the total count of the class value and comparing only that. Since we are using the homomorphic algorithm comparison of count will be same for both encrypted value and decrypted value.

When we are comparing with the decision tree and the RDT, RDT take more time for the creation of horizontally partitioned data because it is partitioned in different data bases. But when we give an instance for classification both decision tree and RDT take same time. Vertically Partitioned data RDT creation take more time compared to RDT creation in Horizontally Partitioned data and Decision tree based on information gain

A. *Results:*

The table below shows the execution time comparison as the number of sites increases.

Table - 8.1
Comparison of Performance

No of sites	*Sites- 1*	*Sites- 2*	*Sites- 3*
Time for ID3Decision Tree Creation	*22ms*		
Time for RDT creation for horizontally partitioned data		*676ms*	*1076ms*
Time for RDT creation for vertically partitioned data		*11578ms*	*17465ms*
Time for ID3 Decision making	*1ms*		
Time for Decision making horizontally partitioned data		*1ms*	*1ms*
Time for Decision making vertically partitioned		*3647*	*9735*

ID3 is having lesser execution time than other methods. But the above observations about ID3 were done after transferring the data from all the sites' training data to a single host. This will increase the network traffic. Also when the complete data is transferred, there is a higher chance of information loss. Even if we encrypt the data, encrypting the large training data set is an overhead.

V. Conclusion

Random Decision tree frame work is used to improve the efficiency of privacy preserving data mining, in distributed environments. Both horizontally and vertically partitioned data were considered. The security breach and increase in network traffic that can be caused by using other methods of decision tree construction is removed. The system proved that it is feasible to perform knowledge discovery in distributed privacy preserving environments. When dealing with distributed data that is partitioned across multiple sites, the system has considered the security and privacy implications. The proposed method overcomes some of the challenges of performing data mining tasks on such data. RDTs can be used to generate decision trees with comparable accuracy, that also in much lesser cost. Our approach leverages the fact that using lesser computation, randomness in structure can provide strong privacy. When compared with other cryptographic methods, RDT requires significantly lesser time. Experimental results also show that the privacy preserving RDT algorithm scales linearly with data set size. Also the methods like ID3 which executes the classification in lesser time, are found to be not suitable for the scenario under consideration, ie, distributed privacy preserving systems.

References

[1] J. Vaidya, C. Clifton, and M. Zhu, Privacy-Preserving Data Mining Advances in Information Security first ed., vol. 19, Springer-Verlag, 2005.
[2] W. Fan, H. Wang, P.S. Yu, and S. Ma, "Is Random Model Better? On Its Accuracy and Efficiency," Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03), pp. 51-58, 2003.
[3] W. Fan, J. McCloskey, and P. S. Yu, "A General Framework for Accurate and Fast Regression by Data Summarization in Random Decision Trees," Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06), pp. 136-146, 2006.
[4] G. Jagannathan, K. Pillaipakkamnatt, and R.N. Wright, "A Practical Differentially Private Random Decision Tree Classifier," Proc. IEEE Int'l Conf. Data Mining Workshops (ICDMW '09), pp. 114-121, 2009.
[5] J. Vaidya, C. Clifton, M. Kantarcioglu, and A.S. Patterson, "Privacy-Preserving Decision Trees over Vertically Partitioned Data," ACM Trans. Knowledge Discovery from Data, vol. 2, no. 3, pp. 1-27, 2008.
[6] O. Goldreich, "General Cryptographic Protocols," The Foundations of Cryptography, vol. 2, pp. 599-764, Cambridge Univ. Press, 2004.
[7] D. Gritzalis, Secure Electronic Voting.ser. Advances in Information Security first ed., vol. 7, Springer-Verlag, 2003.
[8] R. Cramer, I. Damgard, and J.B. Nielsen, "Multiparty Computation from Threshold Homomorphic Encryption," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT '01), pp. 280-299, May 2001.
[9] P. Paillier, "Public Key Cryptosystems Based on Composite Degree Residuosity Classes," Proc. 17th Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT '99), pp. 223-238, 1999.
[10] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," Proc. ACM SIGMOD Conf. Management of Data, pp. 439-450, May 2000.
[11] D. Agrawal and C.C. Aggarwal, "On the Design and Quantification of Privacy reserving Data Mining Algorithms," Proc. 20th ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems, 0pp. 247-255, May 2001.
[12] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the Privacy Preserving Properties of Random Data Perturbation Techniques," Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03), Nov. 2003.
[13] Z. Huang, W. Du, and B. Chen, "Deriving Private Information from Randomized Data," Proc. ACM SIGMOD Int'l Conf. Management of Data, June 2005.
[14] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," J. Cryptology, vol. 15, no. 3, pp. 177-206, 2002.
[15] K. Wang, Y. Xu, R. She, and P.S. Yu, "Classification Spanning Private Databases," Proc. 21st Nat'l Conf. Artificial Intelligence, pp. 293-298, 2006.
[16] W. Du and Z. Zhan, "Building Decision Tree Classifier on Private Data," Proc. IEEE Int'l Conf. Data Mining Workshop on Privacy, Security, and Data Mining, pp. 1-8, Dec. 2002.

[17] J. Vaidya, M. Kantarcioglu, and C. Clifton, "Privacy Preserving Naive Bayes Classification," Int'l J. Very Large Data Bases, vol. 17, no. 4, pp. 879-898, July 2008.

[18] R. Wright and Z. Yang, "Privacy-Preserving Bayesian Network Structure Computation on Distributed Heterogeneous," Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, Aug. 2004.

[19] M. Kantarcio^glu and C. Clifton, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, pp. 1026-1037, Sept. 2004.

[20] M. Kantarcioglu and C. Clifton, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data," Proc. ACM SIGMOD Workshop Research Issues on Data Mining and Knowledge Discovery (DMKD '02), pp. 24-31, June 2002.

[21] X. Lin, C. Clifton, and M. Zhu, "Privacy Preserving Clustering with Distributed EM Mixture Modeling," J. Knowledge and Information Systems, vol. 8, no. 1, pp. 68-81, July 2005.

[22] J. Vaidya and C. Clifton, "Privacy-Preserving k-Means Clustering over Vertically Partitioned Data," Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 206-215, Aug. 2003.

[23] G. Jagannathan and R.N. Wright, "Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 593-599, Aug. 2005.

[24] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 639-644, July 2002.

[25] J. Vaidya and C. Clifton, "Secure Set Intersection Cardinality with Application to Association Rule Mining," J. Computer Security, vol. 13, no. 4, pp. 593-622, Nov. 2005.

[26] M. Kantarcioglu and J. Vaidya, "An Architecture for Privacy- Preserving Mining of Client Information," Proc. IEEE Int'l Conf. Data Mining Workshop on Privacy, Security, and Data Mining, pp. 37-42, Dec. 2002.