International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 9433-9442

http://www.ripublication.com

An Efficient Secure Scalar Product For Privacy Preserving Data Mining Algorithms

M. Antony Sheela[1], K. Vijayalakshmi[2]
[1] *Assoc. Prof. Arunachala college of Engineering for Women*
[2] *Professor & Head, Ramco Institute of Technology*

Abstract

In distributed data mining sensitive information must be shared between parties without mutual trust. Secure Multiparty Computation is a technique that produces mining results without revealing the data used to produce them. A secure scalar product protocol is a specific type of secure multi-party computation problem underlying many privacy preserving data mining algorithms. In many cases, the efficiency and security of the privacy-preserving data mining protocol depends on the security of the underlying secure scalar product protocol. This paper proposes an efficient pseudo secure scalar product algorithm by using secret sharing in a tree structure. The proposed algorithm can be used in the decision tree induction on vertically partitioned dataset to increases the efficiency of the mining algorithm. The execution time of the proposed algorithm is lesser than the existing cryptographic methods.

Key words: scalar product, tree structure, privacy preserving data mining, communication and computation cost.

Introduction

The advancement in computing and storage technology made digital data to be easily collected and stored by organizations, industries, government, hospitals, military, etc. In most of the situations the data is distributed horizontally, vertically or both among different parties. Privacy preserving mining algorithms on the distributed data allows computation of algorithms over an entire data set without compromising the privacy of private data of the participating data sources. The different methods used in privacy preservation are data secure multiparty computation [1-3], perturbation [4,5], and k_anonymity [6]. Privacy preserving data mining on distributed data are carried out in fraud detection system, medical diagnosis, insurance companies, etc.

Secure multiparty computation (SMC) [7-10] is a technique that allows parties to carry out distributed computing tasks in a secure manner. This type of computation

among parties should have the property that the parties know the correct output without revealing the sensitive data, even if some of the parties maliciously collude to obtain more information. The protocol that satisfies this property can be used to solve privacy-preserving data mining problems using secure multiparty computation. SMC requires more cost of communication and computation. Secure Scalar product protocol is one of the prime building blocks in SMC protocol. Several approaches propose the viability of secure scalar product [11-14], but to construct a practical application for secure multiparty computation, the execution time is crucial.

This paper proposes a new pseudo secure scalar product protocol which acts as the core operation in most cases to compute efficient secure decision tree induction on vertically partitioned dataset. The proposed algorithm makes groups of the parties participating in the mining operation. Each attribute in the data set is considered as a binary vector while computing the maximum information gain. This binary vector is also being grouped as partitions instead of considering each bit at a time. For each partition, a tree is constructed to find the pseudo scalar product using Shamir's secret sharing. The sum of ones in all the partitions together gives the scalar product of the binary vector. Thus this method reduces the execution time than the cryptographic methods.

The paper describes the related works in section 2, section 3 explains the background on which the paper is being derived, section 4 describes the proposed pseudo secure scalar product protocol, section 5 gives the experimental analysis of the algorithm and section 6 concludes the work carried out.

Related Work

Several research works are being carried out in the area of privacy preserving data mining to preserve the privacy of sensitive data. The different methods to implement the privacy preservation in data mining algorithms are secure multiparty computation [1, 2, 3], perturbation [4, 5] andk-anonymity [6].

Secure multiparty computation (SMC) is a computation in which private parties that need to compute some join functions, perform it, without revealing their sensitive inputs but revealing the correct output. Few examples of SMC are secure elections, auctions, privacy preserving data mining. SMC uses either secret sharing or cryptographic methods for their computation. Yehuda Lindell et al. [10] carried a work on SMC and stated that the cryptographic protocols should use semantically secure schemes instead of deterministic encryption schemes. F.Emekci et al. [7] proposed an efficient secure sum algorithm for building decision tree over horizontally partitioned dataset. The algorithm scaled well with large number of data sources and the computation and communication costs were less than the cryptographic methods. Hong-da li et.[8] al proposed an oblivious polynomial evaluation using scale product of two vectors which can be used to obliviously obtain values and preserve anonymity in cryptographic protocols. Actively secure protocols use zero-knowledge proofs[2,12] to force parties to behave in a way consistent with the passively secure protocol.

Secret sharing distributes a secret among 'n' parties and each party is allocated a share of the secret. The secret can be reconstructed only when a sufficient number threshold 'k' of shares are combined together. Shamir's secret sharing is one such algorithm [7]. In Shamir's secret sharing algorithm each participant constructs a polynomial 'q' of degree k. The number of publically known random number for the Shamir's secret sharing corresponds to the number of parties involved in the computation. Let the random numbers be represented as $\{r_1, \ldots, r_n\}$. The shares are computed for each party as $(q(r_1) \ldots q(r_n)\}$ and given to the corresponding party. To reconstruct the secret k parties have to use their shares. The secret can be revealed only if k-1 parties collude with each other.

Scalar product is one of the basic components for many data mining algorithms. Secure scalar product in distributed data mining computation is discussed in many privacy preserving data mining papers[9, 11, 12, 13, and 14]. These protocols are implemented by using secret shares or by using cryptographic methods.

Shuguo Han et. al.[9]proposed a secure protocol to compute the Pseudo-Scalar Product for multiple parties holding arbitrarily partitions of data . The Pseudo-Scalar Product computes the scalar product of binary vectors which is a known core operation in decision tree induction and other data mining techniques for vertically partitioned data where each data partition is held privately by one party. Zhiqiang Yang et al.[11] used a relatively straightforward semantically secure homomorphic encryption for the privacy preserving scalar product. The experimental results showed that unless practical optimizations or specialized hardware are used to accelerate encryptions, computation delay is the major bottleneck of performance in their implementation. Bo Yang et al. [12] proposed a secure scalar product protocol with verifiable encryption scheme in the construction of an oblivious pseudorandom function. The proof of knowledge of a discrete logarithm and the verifiable encryption were carried out by the authors to perform secure scalar product protocol in the presence of malicious adversaries. Bart Goethals et al[13] proposed a provably private scalar product protocol that is based on homomorphic encryption which can also be used on massive datasets. Artak Amirbekyan Vladimir[14] constructed a secure scalar product protocol which was based on the Add Vectors Protocol.

Problem Definition

One of the most ubiquitous data-mining problems found in actual life is classification. Decision tree induction is a widely acknowledged solution approach for classification. The secure scalar product is a core operation in most of the decision tree induction over the vertically partitioned dataset. The column of data set held by different parties is considered as a binary vector. Scalar product is found for the binary vector which represents the column vectors of data sets from different parties. Although cryptographic techniques are very secure, the excessive computation and communication cost associated render them impractical for scenarios involving a large number of parties. The problem is to compute the secure scalar product operation in an efficient way to make the mining algorithm efficient.

Decision Tree induction

In a decision tree construction on centralized data set, the tree is built by recursively splitting the training data set based on a locally optimal criterion until all or most of the records belonging to each of the partitions bear the same class label. At each stage of splitting a tree node, the best attribute is to be determined. The concepts of entropy index and information gain are commonly used to find the best possible split. Given a data set D, 'c' be the distinct values of class attribute, the class entropy of a dataset is defined as in equation 1.

$$Entropy(D) = -\sum_{K=1}^{C}\left(\frac{\left|\sigma_{a=\propto_k}(D)\right|}{|D|}\log\frac{\left|\sigma_{a=\propto_k}(D)\right|}{|D|}\right) \quad (1)$$

The information gain when an attribute A is used to partition the dataset D is defined as in equation 2.

$$Gain(D,a) = Entropy(D) - \sum_{\propto\epsilon a}\left(\frac{|D_\alpha|}{|D|}\right) * Entropy(D_\alpha) \quad (2)$$

The dataset cannot always be centralized. In most of the cases the dataset will be distributed horizontally or vertically or both. In distributed dataset, the decision tree induction has to be performed over multiple parties without revealing the original sensitive data.

Distributed decision Tree building over vertically partitioned dataset.

Let the number of parties be 'n'. In vertically partitioned dataset, each site (say 'i') contains a collection of attributes with no party having the same attribute except the transaction identifier. Let □denote the union of attributes over all the sites and D denote the data set formed by joining the data from all the sites. Having $A_i \in$ □ the set of attributes belonging to the party i, $\|A_i\|$ represents the cardinality of the attributes of a party. $a \in A$, and α ∈ $\prod(a)$ represent the distinct values of a.

Each party *X* securely computes the information gain of its attributes using its own partition and other party partition without compromising the data privacy of other parties. Each site determines which of its transactions might lead to a particular node of the tree. With attribute ∈ A , and distinct value α ∈ $\prod(a)$, the binary column vector is represented as $A_{[a=\alpha]} = [x_1, x_2, \dots, x_m]^T$. If the value of $a = \alpha$, then the value of $x_i = 1$else $x_i = 0$, where x_i represent i[th] transaction .The secure scalar product of the binary vectors give the number of transactions that reach a particular node in the tree.

Secure cardinality of scalar product.

Let 'm' be the number of transactions and the transactions are represented in the binary form as $z_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T$, where 'i' represent the i[th] party. The value of $x_{ij} = 1$ if the transaction satisfies the conditions while finding the information gain else $x_{ij} = 0$, where j represent the bit position in the binary vector which corresponds to the position of the j[th] transaction. Let 'd' represent the number of digits in vector partition. The binary vector of a particular node will be of the form $(0/1)_2^m$. Let $m_1 = \left\lceil\frac{m}{d}\right\rceil$, where 'm_1' is the number of transaction partitions. There will be $m_1 - 1$ partitions with 'd' digits and the last partition can have 'd' or less digits. Let 'n' be the

number of parties taking part in the mining and the parties are divided into groups. Let $n_1 = \left\lceil \frac{n}{x} \right\rceil$, represent the number of party groups and 'x' represent the number of parties in each group except the last group. The last group can have 'x' or less parties.

The m_1 partitions of the 'n' parties are represented $z_1 = \left[x_{11}, x_{12}, \dots, x_{1m_1}\right]^T$, ..., $z_n = \left[x_{n1}, x_{n2}, \dots, x_{nm_1}\right]^T$, where x_{iq} represent $(0/1)_x^d$, 'i' and 'q' represent the i^{th} party and the q^{th} partition. The scalar product of all the parties is defined as in equation 3.

$$Z = \left[(x_{11} + \cdots + x_{n1}), \dots, \left(x_{1m_1} + \cdots + x_{nm_1}\right)\right]^T \quad (3)$$

Each element of Z takes the values $\sum_{i=1}^{n} x_{iq} \leq N$, where the value of N can be written as $[N_1 N_2 \dots N_d]$ and d represents the number of digits in vector partition. If $N_k = w$ then $N_k \leftarrow 1$else $N_k \leftarrow 0$, where 'k' represent the digit of N. The intersection of the set of transactions from all the parties with the transactions of the class party gives the number of transactions that reach a particular node in the tree. This is performed with the scalar product computation.

The Proposed Method To Securely Find The Pseudo Scalar Product

Pseudo scalar product computes the scalar product of binary vectors by considering binary vectors as groups of numbers. The binary vector is considered as a number with radix 'r'. The sum of the $(\text{number})_r$ of the parties of the group is computed using secret sharing scheme. That sum of the numbers is then converted to binary form by making the bit position as '1' if the digit value is equal to the number of parties of the group else it is '0'. This resultant binary vector of each group is combined till the number of groups become one. The number of 1's in the binary vector of the top node gives the pseudo scalar product.

Algorithm

Let 'm' be the number of transactions, 'd' be the maximum number of digits in the integer, $m_1 = \left\lceil \frac{m}{d} \right\rceil$ is the number of partitions of transactions and 't_j' represents each transaction partition; where j= 1 to m_1. Let 'N' be the number of parties, $r \leq 35$ be the radix , $n_1 = \left\lceil \frac{n}{r} \right\rceil$ is the number of party groups to perform the secure sum operation, and g_i represents each group; where i= 1 to n_1.

Procedure:

Phase 0: Setup phase.

Step 1: Compute $m_1 = \left\lceil \frac{m}{d} \right\rceil$, $n_1 = \left\lceil \frac{n}{r} \right\rceil$

Step 2: Each party p_i computes the following; where i= 1 to n

Step a: Create a vector by making the bit position (transaction id position) as 1 if the condition of the attribute to find the maximum information gain is satisfied else 0.

Step b: Partition the binary vector into m_1 partitions where each partition consists of d digits except the last one.

Step c: The vector is now considered as a number with radix 'r' i.e. $\left({}^{0}/_{1}\right)_{r}^{m}$

Step d: Generate a random number with base r and add it with each vector.

Step 3: Initialize the Cardinality_count to 0

Phase 1: Computing the final vector for each transaction partition.

For each transaction partition t_j where j= 1 to m_1 do

{N=n_1

While N<0

{For each party group P_I where I= 1 to N do

{One of the parties in the party group is selected as
the representative. Let the number of parties in each
group be 'f'.

Compute the secure sum of the corresponding
integers of P_I using Shamir's secure sum protocol.

All the parties in the group send the random
numbers to the representative.

The representative party finds the sum using
Shamir's reconstruction and subtracts the sum of
random numbers from it. If the value in a digit position
is 'f' then that digit position is made as 1 else it is
considered as 0.}

N is made equal to the number of representatives in this level.}

x = Count the number of 1's in the final vector

Cardinality_count = Cardinality_count + x}

In phase 0, each party forms the vector representation of the attribute. Suppose m is the number of transactions, then the transactions are divided into $m_1 = \left\lceil \frac{m}{d} \right\rceil$ partitions. Each party adds a random number with their private data partition. The number of parties is divided into $n_1 = \left\lceil \frac{n}{r} \right\rceil$ groups in each level. The maximum value of r is less than or equal to 35 because the value of single digit can be 0 to 35 (0-9, A-Z).

In phase 1, for each partition the parties form groups and the number of 1's in the partition is found out. Each of the party group, select one party in the group as representative. Using Shamir's secret sharing the sum of the secret values of all the parties in the group is added together. All the parties of a group send the corresponding random number to the representative. The representative reconstructs the sum and subtracts the added random numbers from it. It then convert the digits of the resultant sum to 1 if the digit equals the number of parties in the group else 0. The representatives of this level act as the parties for the next higher level. This proceeds till there is only one group at the top level. The number of 1's in the result is the cardinality of the partition. The sum of cardinalities of all the partitions forms the cardinality of the scalar product.

Security of the proposed algorithm against collusion

Let the number of parties in a group be $'f'$. The parties cannot learn the secret value of another party even if they exchange their shares with each other. The secret value x_i

of party p_i of a group can only be revealed if all the remaining $f-1$ parties collude and subtract their sum of values from the overall sum. Each party p_i executes Shamir's secret sharing algorithm with a random polynomial of degree $f-1$.The number of parties that are involved in secret sharing is r. The representative party knows the total number of ones. But the contribution of individual party is not revealed by the sum. So the original data is unrevealed.

Cost analysis.

The cost estimates for secure scalar product using the proposed method needs to be considered because the efficiency of the same greatly influence the execution time of the under lying decision tree induction. Let the number of parties be 'n'. The number of parties in a group is r. The maximum size of the group is r. Let the number of parties in a group be $'f'$ and the total number of partitions be $m_1 = \left\lceil \frac{m}{d} \right\rceil$. To compute the scalar product, each party selects a random polynomial and a publically known value. Then each party finds its share for each publically known value and sends the corresponding share to the respective parties. The shares can be found out by the parties. All the groups in the same level can operate in parallel. The scalar product protocol is executed $u * m_1$ times for each attribute, where $'u'$ is the possible distinct values of the attribute.Let the number of public values be z . The scalar product of each party requires z polynomial evaluation. The computation cost of finding the scalar product for an attribute is $z * d * m_1$. For $n_1 = \left\lceil \frac{n}{r} \right\rceil$ party groups the computation within a group can be performed in parallel. If there are $\log_r$ n levels in the tree then the cost is$\log_r \mathrm{n} * z * d * m_1$. The communication cost within a group to compute the scalar product of an attribute is $z * d * n_1 * \log_r \mathrm{n} * m_1$.

Experimental Results

The proposed scalar product algorithm was implemented in matlab and the execution time was measured in dual core 2GHz processor with 512M memory. The execution time of the algorithm was computed by varying the number of parties with fixed number of instances and by varying the number of instances with fixed number of parties.

Figure 6.1 shows execution time of the algorithm when the numbers of parties 'n' vary. The number of instances or the number of transactions 'm' is kept constant. The number of instances of CPUARFF dataset is 209. The value of m is set as 209. The value of n is varied from 10 to 50000. The result shows that the execution time of the algorithm shows a negligible increase when the groups are executed in parallel but when the groups are executed in sequential the execution time varies from 0 to 1200 Sec. This experiment shows that the algorithm scales well with the increase in the number of parties.

Figure 6.2 show execution time of the algorithm when the numbers of instances 'm' vary. The number of parties 'n' is kept constant. The number of parties is fixed as 20. The value of m is varied from 100 to 30000. The result shows that the execution

time of the algorithm shows gradual increase in execution time when executed in parallel.

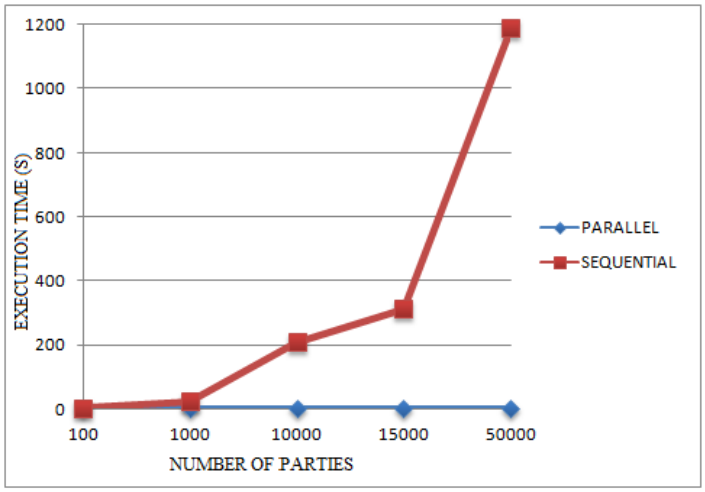

Figure 6.1: Execution time by varying the number of parties

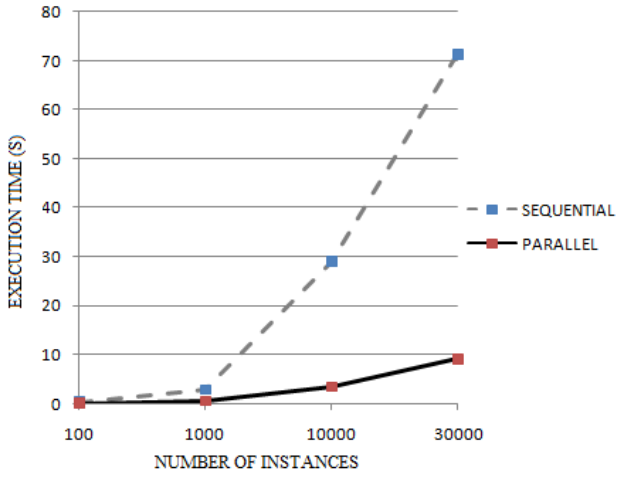

Figure 6.2: Executiontime by varying the number of instances

Conclusion

Distributed data mining algorithms are performed among multiple parties. This needs to be performed in a secure manner which needs extra overhead. In distributed decision tree induction on vertically partitioned dataset, secure scalar products may be a main operation. In algorithms where secure scalar products are used as a main operation increasing the efficiency of this will increase the efficiency of the

underlying mining algorithm. In this paper, an efficient secure pseudo scalar product has been proposed which reduces the execution overhead over cryptographic methods on the distributed decision tree induction algorithm over vertically partitioned dataset. It has used the benefits of the Shamir's secret sharing instead cryptographic methods and grouping the binary vector instead of using it individually.

References:

[1] Murat Kantarcioglu and Chris Clifton," Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data",IEEETransaction on Knowledge and Data Engineering, Vol. 16, No.9,pp 1026-1037, 2004.

[2] Kantarcioglu M,Kardes andOnur,"Privacy-Preserving Data Mining Applications in the Malicious Model ",ICDMWorkshops 2007. SeventhIEEEInternational conference on DOI: 10.1109/ICDMW. 2007.86 , pp 717 – 722, 2007.

[3] Jaideep, Vaidya, Chris Clifton, Murat KantarciogluandScott Patterson A," Privacy-Preserving Decision Trees over Vertically Partitioned Data", ACM Transactions on Knowledge Discovery from Data, Vol. 2, No. 3, pp 14:1-14:27, 2008.

[4] Geetha Jagannathan_, Krishnan Pillaipakkamnatt and Rebecca N Wright," A Practical Differentially Private Random Decision Tree Classifier", Transactionon Data Privacy ,Vol.5,pp 273–295,2009

[5] Keke Chen and Ling Liu," Geometric Data Perturbation for Privacy Preserving Outsourced Data Mining",IEEE Transactions Knowledge and Data Engineering, 2012.

[6] Wei Jiang and Chris Clifton," A secure distributed framework for achieving *k*-anonymity", The VLDB Journal, Vol. 15,pp 316–333,2006.

[7] Emekci F, Sahin O D, Agrawal D and El Abbadi A ," Privacy preserving decision tree learning over multiple parties", Science direct, Data & Knowledge Engineering, Vol. 63, pp348-361, 2007.

[8] Hong-da li ,dong-yaoji, deng-guofeng and baoli,"Oblivious polynomial evaluation", J. Comput. Sci. & Tech., Vol. 19, No. 4,pp 550-554, 2004.

[9] Shuguo Han, and Wee Keong NG," Multi-Party Privacy-Preserving Decision Trees for Arbitrarily Partitioned Data", International journalIntelligent control and systems,Vol.12, No.4,pp351-358, 2007.

[10] Yehuda Lindelland Benny Pinkas, "Secure Multiparty Computation for Privacy-Preserving Data Mining", The Journal of Privacy and Confidentiality , Vol.1, No. 1, pp 59-98, 2009.

[11] Zhiqiang Yang, Rebecca N. Wright and HiranmayeeSubramaniam," Experimental Analysis of a Privacy-Preserving Scalar Product Protocol", International Journal of Computer Systems Science and Engineering, Vol. 21, No. 1, pp 47–52, 2006.

[12] Bo Yang, Yong Yu and Chung-Huang Yang,"A Secure Scalar Product Protocol Against Malicious Adversaries", Journal of Computer Science and Technology, Vol. 28, No. 1, pp152-158, 2013.

[13] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikainen," On Private Scalar Product Computation for Privacy-Preserving Data Mining", 7th International Conference on Information Security and Cryptology (ICISC 2004), Vol. 3506, pp 104-120,2004.

[14] Artak Amirbekyan Vladimir and Estivill-Castro," A New Efficient Privacy-Preserving Scalar Product Protocol", Conferences in Research and Practice in Information Technology (CRPIT), Vol. 70, 2007.

[15] Ching-Hua Yu, Sherman S.M. Chow, Kai-Min Chung, and Feng-Hao Liu," Efficient Secure Two-Party Exponentiation" , Springer-Verlag Berlin Heidelberg 2011,CT-RSA 2011, LNCS 6558, pp. 17–32, 2011.

[16] Tony Thomas," Secure Two-party Protocols for Point Inclusion Problem", International Journal of Network Security, Vol.9, No.1, pp.1-7,2009.

[17] Jharna Chopra and SampadaSatav, "Privacy preservation techniques in data mining", IJRET, Vol.2,No.4, 2013.