A Privacy Preserving Algorithm for Mining Rare Association Rules by Homomorphic Encryption

Weimin Ouyang, Qinhua Huang
Department of Computer Teaching, Shanghai University of Political Science and Law
Shanghai, China
{oywm, hqh}@shupl.edu.cn

Abstract—**Privacy-preserving data mining have greate significance in the era of big data. The Privacy-preserving condition on rare association rules mining is about the sensitive information regarding participants. Each side have a private dataset, aims to collaboratively find rare association rules on data set like a logically unified frame, but actually composed of distributed private data set. We proposed a new efficient algorithm to discover privacy-preserving rare association rule mining technique. The main principle idea is that with the secure two-party computation theory we employ homomorphic encryption to hide the private information.**

Keywords-privacy preserving; homomorphic encryption; secure two-party computation; rare association rule mining

I. INTRODUCTION

Data mining and knowledge discovery is a hot topic in the research of artificial intelligence and big data It aims to find previously unknown, but potentially useful rules, knowledge from traditionally big data [1]. The research of knowledge discovery data mining has a implicit assumption that the database is open to be used. However, this assumption is often untrue in the real world. Some databases may involve sensitive or private personal information that do not be wanted or should not be public. Therefore, the real world needs data mining algorithms that ensure private information kept. There are many researches developed to keep the mining process in the condition of privacy preserving for this purpose [2]. Knowledge discovery and data mining, also known as KDD, encompasses many typical meaningful research directions, such as mining association rule, discover of sequential pattern, classifying and clustering problem. For these typical problems, the research community has proposed the corresponding algorithms for privacy preserving KDD. However, there are few privacy preserving mining algorithms for many new problems derived from these typical problems. For example, rare association rule [3,4] has no corresponding privacy preserving mining algorithm. The problem of mining on vertically distributed data to get rare association rules in it under the condition of the privacy preserving is addressed in this paper.

Suppose we have two agents, A and B, which have private database, P1 and P2, respectively, where P1 and P2 are vertically distributed database, that is to say, different agent collect different attribute values for the same entities. These two agents will carry out a certain kind of privacy preserving algorithm on P1 ∪ P2 with no third party to mine rare association rule. We need a mining algorithm to ensure that the two agencies will not disclose each participant's privacy. Based on the principle of homomorphic encryption, we put forward a protocol to deal with the situation of secure two-party computation, during the process of privacy preserving rare association rules mining.

We organize our paper as follows: section 1 gives the related work review; we describe the rare association rule mining algorithm in section 2 and 3. The secure two-party protocols employing scalar product operations are carefully designed to find rare rules in section 4. To have a performance comparison, we implemented our proposed algorithm. We report the experiment result in section 5. We make our conclusion and describe the possible future works in section 6.

II. RELATED WORK

A. Secure Two-party Computation

Yao [5] firstly put forward the problem of secure two-party computation, and O. Goldreich [6], which extended the problem of secure two-party computation to problem of secure multiparty computation. Based on the cryptography secure model, the computation protocol of secure multiparty is able to compute any operation in distributed computational environment where each side of participants has his inputs. Meanwhile all of the participant roles do not need to have trust between each side, nor the channels the participants communicate by. Generally, all of the sides should expect to get the correct result of computation starting from their local inputs, while guarantee not to disclose their local private data. This is the so-called secure multiparty computing problem. Secure multiparty computing allows each participant to know its own data, computation results, as long as with any information that might be inferred from its own data and the computation results. If we can have a trustful third party existed, the secure multiparty computation should be very simple. We can directly transfer the data of all parties to this trusted third party, perform any required computation, and release the result to each party respectively. However, the problem is that such trusted third parties are hard to find in the real world. Therefore, there is an urgent and practical need for the research of security two-party and multiparty computation.

978-1-7281-5255-1

The standard for secure multiparty computation can reflect the level of secrecy that could be get by performing independent calculations with a trusted third party. The trusted third party is a data storage and a processing node which role is to guarantee the secrets of their private data not disclosed. The trustful third party executes the computation. Only the results will be released at the end. In such a computation environment, no participant can infer the private data information of the other participants based on its own data and the public computation outputs. The goal of secure multiparty computing algorithms is to achieve the equivalent performance of privacy preservation with that of the trustful third party.

B. Privacy Preserving KDD

Since R. Agrawal. proposed the problem of privacy preserving data mining in 2000, data mining with privacy preserving has attracted more and more attention from knowledge discovery data mining communities. Researchers have proposed a number of privacy preserving data mining techniques such as data perturbation, encryption, secure two-party computation, secure multiparty computation, and so on. Privacy preserving data mining issues includes association rules mining, classification, etc. [7]. Serval researchers give solutions from different perspectives [8][9], using the property of uncertainty of equations solutions while having excessive unknown variables and Bayesian networks structures to execute homo encryption, respectively. Unfortunately, the computing protocol is fragile and not secure enough. According to his computation protocols, the party A is able to find privacy in case b_i is 1 or 0, by just comparing e(1) and $e(b_i)$ when the other part B sends his encrypted vector $(e(b_1),\ldots,e(b_n))$ to Part A, for Part A knows public key e. We put forward a computation protocol by securely computing scalar product in the scenario of two parties for this problem.

III. RARE ASSOCIATION RULE MINING

For the importance of Association rules mining job in data mining community, a huge variety of algorithms have been proposed to low the running time cost, which were aim to generate frequent itemsets candidate for further result of association rules. And R. Agrawal firstly raised this problem in 1993.

Recent research showed some infrequent itemsets can also have important insight view [3] as an supplemental perspective to find the knowledge in big data. Thus a new problem was proposed as looking for rare association rule. While the association rules can be mined from frequent itemsets, rare association rules can be discovered from rare itemsets. Frequent itemsets reveal the information about the items occurs frequently, and rare itemsets reveal the information about the items, which occurs infrequently.

Rare association rules mining is an association rule which has low support and high confidence. In recent years, the problem of mining rare association rules has gained quite a lot of attention, which has become a hot topic in KDD research. However, the current research on mining rare association rules do not consider any problem of privacy preserving mining, and all data are open and transparent, and they do not care about whether there are sensitive, secret and other privacy data.

Suppose we have two agents, Alice and Bob, which have their private database, P1 and P2, respectively, where P1 and P2 are database vertically distributed,

A. Problem Representation

We think about such a circumstances:agents, Alice and Bob, each agent has a private dataset vertically partitioned (marked as P1 and P2 respectively). The two agents wish to discover rare association rule on $P1 \cup P2$. To keep the security of data, each agent cannot obtain any information about the other agent, except for the consequence of mining algorithm. There is no third agent, whether it is trusted or non-trusted, during this mining. For simplicity and generality, hypotheses about P1 and P2 are set in the following:

(1) P1 and P2 have the same transaction records number L;
(2) The ID number of i-th transaction in P1 is the same as the ID number of i-th transaction in P2. But the other attributes except for ID number are secret to different agent.

Alice and Bob, each has their private database P1 and P2. These two agents manage to find out rare association rule which has low support and high confidence with no private information leaked out. We say a rule $X \rightarrow Y$ is a rare association rule, if and only if $sup(X) < maxs$, $sup(X) \geq mins$ and $Conf(X, Y) \geq c$, if given a predefined maximum threshold of support maxs, a specified minimum threshold of support mins, and a specified mini-confidence c.

B. Rare Association Rules Mining Algorithm

The algorithm of mining rare association rules on $P1 \cup P2$ is presented in [2]:

Algorithm *MRAR*

Input: *P* (a vertically distributed transaction database), maximum support threshold: maxs; minimum support threshold: mins; minimum confidence threshold: c.

Output: Set of rare association rules: RAR；
Begin
  /* transforming each item in transactions * /
  Repeat:
    For all transaction in P do
        transform each item X in T_i into bit-sequence formation;
    For all Bit(*X*) in P do
      Remove *X* , if sup(*X*) = 0;
  EndRepeat
/* Generating the rare itemsets*/
  R= ∅;
  R_1 = {rare 1-itemsets};
  For (k=2; $R_{k-1} \neq$ Null; k++) do {
      C_k = Candidate_Gen(R_{k-1});
      Counting the supports of C_k by bit-operator AND;
      R_k = { $c_k \in C_k$ | $sup(c_k) < maxs \wedge sup(c_k) \geq mins$ };

```
    R = R ∪ Rk;
    }
/* Generating the rare association rules */
  RAR = ∅;
  For each itemset i in R Do {
    For any X∪Y=i and X ∩Y = ∅ Do {
      If Conf(X→Y)≧c
      Then PAR = PAR ∪ { X→Y };
    }
    }
End
```

IV. METHOD TO THE MINING OF PRIVACY PRESERVING RARE ASSOCIATION RULE

In our algorithm, there are two steps need data access operations: (1) On the counting of rare 1-itemset. (2) On the support counting of candidate itemset C_k. Since there is just one property invloved (after mapping), the rare 1-itemset can be found out within a agent, data accessing operations over two agents can be saved. If candidate itemset C have 2 or more properties, which may be located in both agents, data accesing on both would have to be executed. We proposed the secure computation method on the support counting of candidate itemset C_k while keeping participants' private information.

If all the properties of the candidate set C belong to one agent, we can simplify the problem by counting candidate support immediately. The problem is solved by calculating the scalar product of the corresponding attributes in the candidate set C.

Otherwise, if the attributes of candidate itemset C are distributed over both two agents, the computation of the scalar product to find support of itemset C must ensure that their private information will not be disclosed.

A. Homomorphic Property of the Encryption

We use isomorphic encryption algorithms to secure privacy protection slots in two-party scalar product agreements to find support for candidate project sets. If the encryption algorithm meets the following properties, it is homogeneous:

(x) * e(y) = e(x + y).

Suppose Z to be a k-item set and agent Alice has m attributes $X_1,X_2,\ldots,X_m$ and agent Bob have n attributes $Y_1,Y_2,\ldots,Y_n$, i.e., $Z=(X_1,X_2,\ldots,X_m, Y_1,Y_2,\ldots,Y_n)$, where m>=1，n>=1，m+n=k. We denotes the j-th attributes value of X_i as X_{ij}, and j-th attribute value of Y_i as Y_{ij}. For simplicity, let vector A= $(a_1,a_2,\ldots,a_L)$, vector B=$(b_1,b_2,\ldots,b_L)$, where $a_j = \prod_{i=1}^{m} X_{ij}$ (j=1,2,3,…,L), $b_j = \prod_{i=1}^{n} Y_{ij}$ (j=1,2,3,…,L).

In order to get the support count of candidate itemset Z, agent X and agent Y can compute scalar product $X_1*X_2*\ldots*X_m * Y_1*Y_2*\ldots*Y_n = A*B$. We have: Z.Count=A*B.

Next we need to figure out how to compute the scalar product A*B for Alice and Bob, while keeping the privacy of these two agents.

By the property of scalar product, it holds P(A)*P(B) = A*B, here P is an permutation operation on Alice and Bob. If agent B sends vector P(B) to agent Alice, in this transfer process, Bob will keep his secrecy of P and P(B). As you can see, the probability of him guessing the order of an element for agent A is 1/n, and the probability of guessing the order of all elements is 1/n!, very similar to NP-complete problem. In this case, agent Alice knows A and result of P(A) operation, thus there are risks for agent Alice deducing out the permutation P. To deal with this situation, instead we can let Alice know P(A+R) rather than P(A), here R is introduced with a random vector, given by agent Bob. Bob need to keep the secrecy of it. So the probability of agent Alice successfully guessing permutation P is decrease down to 1/n! by this vector A+R operation From Bob's perspective, to securely fetch scalar product A*B agent Bob need to sends P(A+R) and P(B) to agent A. As the protocol goes, Alice should compute P(A+R)*P(B) and sends the result in return. The agent B receives Alice's data then computes P(A+R)*P(B)-R*B=P(A)*P(B)=A*B.

Also the probability of agent Alice guessing out a certain elements in agent Bob from P(B) can be lowered by a certain kind of cutting operation. For example, we can randomly partition vector B into m parts, $U_1,\ldots,U_m$, here $B= U_1+\ldots+U_m$. Then agent B defined a permutation P_i with some randomness and send P_i (U_i) (i=1,2,…, m) to agent A. Therefore, the probability that Agent A will guess the order of an element will be greatly reduced to only $1/n^m$.

Based on above discussion, there is a key problem need to be solved. How do we make agent Alice get the result of P(A+R) while keeping P and R's secrecy? We propose the following secure vector permutation protocols to address this problem. To keep security, we designed the following vector permutation protocol to resolve this issue.

Protocols 1: Vector Permutation Operation Protocol for Two Party

Input: The agent Alice has her private vector A. Permutation P and random vector R is with Bob.

Output: The agent Alice receives P(A+R).

Begin:

(1) For agent Alice, she need to generates public key and keep a private key pair (e,d) by the method of homomorphic encryption, here *e*, *d* are encryption and decryption operation, respectively. The e should let agent Bob know.

(2) The agent Alice will encrypt vector X, using public key e, e(A) = $(e(a_1),\ldots, e(a_L))$ and sends e(A) to the agent Bob.

(3) The agent Bob encrypts vector R with public key, e(R) = $(e(r_1),\ldots, e(r_L))$ and computes e(X)*e(R) = e(A+R). Then the agent B performs random permutations on e(A+R) and gets P(e(A+R)). Send P(e(A+R)) to the agent Alice.

(4) The agent Alice decrypts the P(e(A+R)) with private

key d: d(P(e(A+R))) = P(d(e(A+R))) = P(A+R).

End

On the basis of protocol 1, we propose a secure two-party scale product calculation protocol.

Protocol 2: Securely Calculating Two-party Scalar Product

Input: Agent Alice and Bob own private vector A, B respectively.

Output: Alice receives A*B.

Begin

(1) For Bob:
   (a) Vector B will be divided into m parts, denotes by $\{U_1,\ldots U_m\}$.
   (b) Generates m random vectors $\{R_1, \ldots, R_m\}$. Denotes $w = \sum_{i=1}^{m} U_i R_i$.
   (c) Generates m random permutations, denotes by $\{P_1,\ldots, P_m\}$.

(2) For Alice:
   Do partitions, dividing A into m parts, denoted by $\{V_1,\ldots,V_m\}$.

(3) For Alice and Bob, when i=1, …, m:
   (a) Alice receives $P(V_i+R_i)$, through protocol 1.
   (b) Agent Bob sends the permutations vector..
   (c) The agent Alice computes a small part of S: $s_i = P_i (U_i) * P(V_i+R_i) = V_i * U_i + U_i * R_i$.

(4) The agent Alice computes $S = \sum_{i=1}^{m} s_i = \sum_{i=1}^{m} U_i * V_i + \sum_{i=1}^{m} U_i * R_i = X * Y + w$ .. S will be send back to Bob.

(5) The agent Bob computes S = S-w = A*B+w-w =A*B. The agent Bob sends S to the agent Alice.

End

B. Security Analysis

By performing above protocols, for Alice, all information of agent Alice get is $P_i (U_i)$ and $P(V_i+R_i)$ (i =1, …, m), in addition to the final result of scalar product. Agent Alice have no information of Bob's randomly generated vector. Alice also knows nothing about the permutation P_i (i =1, …, m), neither U_i nor B.

It is very hard for Alice to guess out an element in B, theoretically with chance of 1/nm. To break all $P_i (U_i)$ (i =1, …, m) by brute-force, Alice only can traverse all elements in B for about n!m steps. In addition to the results of the scalar product, all the information Bob gets from agent Alice is e(X). Because agent Bob doesn't know the private key, Bob can't reveal A.

Therefore, it is hold that agent Alice and agent Bob cannot reveal the opponent's private vector information but the scalar product value.

Because permutation is just to change the order rather than the value of element, by receiving all $P_i (U_i)$ (i =1, … , m) in the protocols, Alice could disclose the sum of elements in Y. This might generate a weak point of the protocols.

V. EXPERIMENTAL RESULTS

In our experiments, we compare MRAR against our privacy-preserving mining. The algorithms is implemented in Java. The experiments' environment is CPU 2.33 GHz with operating system Ubuntu GNU/Linux and RAM of 8 GB. The data set used is T20I6D100K [10]. This data is a typical weakly correlated sparse data set, constructed from the market basket data.

TABLE I. MRAR ALGORITHM VS. OUR PRIVACY PRESERVING VERSION

Min_supp	MRAR (sec)	Privacy_preserving (sec)
10%	2.8	15.6
0.5%	39	367
0.25%	90	1041

From the Table 1, it can be conclude our privacy-preserving version have a rather highly time cost compare to original MRRA, from 4 to 10 times, depend on value of the min_supp. So the security and the performance is the two blades of a sword. While in situations where the privacy preserving prevails, this computation cost is often considered to be necessary and affordable. And we should point out this is a very preliminary work of our exploration, and it still has spaces to optimize in the future.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a two-party secure computing protocol under the condition of protecting privacy in the mining of rare association rules. Depend on the based on homomorphic cryptography theory, we designed a protocol to address this problem. Through the application protocol, each party does not necessarily transfer its data to a neutral third party. The homomorphic cryptographic computation this secure two party case can successfully to handle the computation problem, thus each party's data privacy can be guaranteed.

As a major future study, the evaluation method for quantitative analysis is urgently in need in privacy preserving rare association rule mining scenarios. Since secure multiparty actually is a computation-intensive job, the optimization of the protocols model is of most importance. Security both clustering and classification models will also be continuously developed in accordance with the protocols prepared in this article.

REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, Privacy-Preserving Data Mining, ACM SIGMOD, 2000:439-450

[2] Weimin Ouyang, Discovery of Fuzzy Rare Association Rules from Large Transaction Databases,In: Proceedings of International Conference on Education, Management, Computer and Medicine (EMEC2016), pp.160-165. ALTANTIS PRESS.

[3] Torino, L., Sibelius, G., Barolo, C.: A fast algorithm for mining rare itemsets. In: Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications, ISDA 2009, pp. 1149–1155. IEEE Computer Society, Washington, DC (2009).

[4] A.C.Yao, Protocols for secure computations. In Proc. Of the 23rd Annual IEEE Symposium on Foundations of computer Science,1982.

[5] O. Goldreich. Secure multi-party computation (working draft). http://www.wisdom.weizmann.ac.il/~oded/pp.html, 1998.

[6] V.S.Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, Y. Theodoridis State-of-the-art in Privacy Preserving Data Mining.,In SIGMOD Record, 2004 http://www.sigmod.org/sigmod/record/issues/0403/B1.bertion-sigmod-recorP2.pdf

[7] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine 17(3),pp37-54 Fall 1996.

[8] J.Vaidya and C.W.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, July 23-26, 2002, Edmonton,Alberta, Canada.

[9] R.Wright and Z. Yang. Privacy-preserving bayesian network structure computation on distributed heterogeneous data. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2004.:713-718.

[10] Hacene, M.R., Toussaint, Y., Valtchev, P.: Mining safety signals in spontaneous reports database using concept analysis. In: Proc. 12th Conf. on AI in Medicine, AIME 2009. Volume 5651 of Lecture Notes in Computer Science. (2009) 285–294