Secure Outsourced Association Rule Mining using Homomorphic Encryption

Sandeep Varma[1], LijiP I[2]

Department of CSE, College of Engineering, Trivandrum

Abstract—*Several techniques are used in data analysis, where frequent itemset mining and association rule mining are very popular among them. The motivation for 'Data Mining as a Service' (DMaaS) paradigm is that when the data owners are not capable of doing mining tasks internally they have to outsource the mining work to a trusted third party. Multiple data owners can also collaboratively mine by combining their databases. In such cases the privacy of outsourced data is a major issue. Here the context includes necessity of 'corporate privacy' which means other than the data, the result of mining should also preserve privacy requirements. The system proposed uses Advanced Encryption Standard (AES) to encrypt the data items before outsourcing in order to prevent the vulnerability of 'Known Plaintext' attack in the existing system. Fictitious transactions are inserted to the databases using k-anonymity method to counter the frequency analysis attack. A symmetric homomorphic encryption scheme is applied in the databases for performing the mining securely. Based on the experiments and findings, though the running time of proposed solution is slightly greater than the existing system, it provides better security to the data items. Since the computations tasks are performed by the third party server, consumption of resources at the data owners' side is very less.*

Keywords—privacy preserving, association rule mining, data mining, vertically partitioned databases, homomorphic encryption.

I. INTRODUCTION

The method of evaluating data from different angles and summarizing it into favorable information is the essence of data mining [14]. Reviewing the data or mining information can be very useful to a business. The extracted information can be used to raise revenue, reduce costs, or both. The method is also called data or knowledge discovery. The term data mining refers to extracting or mining information from massive quantity of data. There are several analytical tools for analyzing data. Data mining software is one among them. The data analyzed from different angles or dimensions are categorized and the identified relationships are summarized. Identifying the inter relations or patterns from large relational databases is carried out in data mining. Data mining is particularly vulnerable to misuse. So, the requirement of protecting privacy is a major concern in data mining.

In data mining the term privacy [6] is referred for finding valuable information. There is an enormous collection of large amounts of personal data such as criminal records, purchase details, health records etc. Each individual has the right to control their personal information. When the control in privacy is lost, the major issues which can be occurred are misuse of private information, handling misinformation and granulated access to personal information.

The datasets can be analyzed by techniques called frequent itemsets mining and association rule mining [3]. The former method is used to find data items or itemsets that co-occur frequently and the latter method is used for identifying exciting association coherence between data items in heavy transaction databases. The consistency between the products in large transaction databases are found out using association rules and an event which involves one or more products (items) in the trade or domain is referred as a transaction. For example, purchasing of items by a customer in a supermarket is a transaction. A set of items is called an "itemset" and an itemset with "k" number of items is called "k-itemset".

The context dealt here includes multiple data owners collaboratively mining from their joint data to learn frequent itemsets and association rules. The major issue of privacy pull back the data owners from sending their unprocessed data to a central station. In a horizontally partitioned database, one or more tuples (i.e., transactions) in the combined database are possessed by every data owner while in a vertically partitioned database the data owners possess one or more attributes in the joint database. This work focus on vertically partitioned databases which are beneficial in market basket analysis [3].

II. BACKGROUND

2.1 Pattern Mining Task

The formal definition of problem of mining association rules can be stated as follows [1]: Let$L = \{l_1, l_2, \ldots, l_n\}$be a set of literals called items. Suppose a set of transactions is represented by T, where each transaction t is a set of items such that$t \subseteq L$. A unique identifier called Transaction ID (TID) is integrated with each transaction. It is said that transaction t contains P, a set of some items in L, if$P \subseteq t$. An association rule is an implication like$P \Longrightarrow Q$, where$P \subseteq L, Q \subseteq L$, and$P \cap Q = \phi$. The rule$P \Longrightarrow Q$exists in the transaction set T with confidence value c, if c% of the transactions in T that contain P also contain Q. The rule$P \Longrightarrow Q$has support value s in the transaction set T, if s% of the transactions in T contains$P \cup Q$. If the support value of an itemset P is greater than or equal to the *minimum support* threshold, then it is called *large* or *frequent* itemset. Our aim is to discover all frequent itemsets and generate association rules from them.

2.2 Cryptographic Hash Function

A cryptographic hash function $H()$ is a one-way function. If we have a hash value $\hat{h}$, it is arithmetically not feasible to discover another plaintext msuch that $H(m) = \hat{h}$. The hash function has two properties called weak and strong collision resistance. Collision resistance is the property in which the collision of two hash values doesn't occur i.e., it is arithmetically not feasible to discover two different plaintexts m1 and m2 such that $H(m_1) = H(m_2)$. SHA-1 and SHA-2 are two widely used cryptographic hash functions.

2.3 Homomorphic Encryption

In homomorphic encryption schemes, the operations we perform on plaintexts can be done on ciphertexts without any error. The main operations used are addition and multiplications. In additive homomorphic encryption schemes, only addition operation is allowed. Similarly, multiplicative homomorphic encryption only allows multiplication operation. When only one of the operation is allowed it is partially homomorphic and if both operations are allowed it is fully homomorphic encryption scheme.

In an homomorphic encryption which is additive, the ciphertext of the total of two messages, $m_1 + m_2$, can be calculated using some operation "•" on the encrypted version of m_1 and m_2, without initially deciphering m_1 and m_2 or carrying the key for decryption.

In a multiplicative homomorphic encryption, the ciphertext of the product of two messages, $m_1 \times m_2$, can be calculated with an operation "$\otimes$" on the encrypted version of m_1 and m_2, without initially deciphering m_1 and m_2 or carrying the key for decryption.

III. RELATED WORKS

Association rules are statements of the form 'if … then'[14]. They are used to find relationships among large data in a relational database or some other data depository even though the data may seem unrelated. "If a customer buys shoes, he is 70% likely to purchase socks also" is an example for an association rule as statement. An antecedent (if) and a consequent (then) are two parts of an association rule. An antecedent is an element encountered in the data. A consequent is an element that happens in association with the antecedent. Association rules are generated on the basis of support and confidence by evaluating data for frequent if/then patterns to determine the utmost relevant relationships. Support value shows how frequently the data items appear in the database. Confidence value point out how many times the if/then statements become true. In data mining techniques, association rules are widely used for analyzing and anticipating customer behavior. They have an important role in shopping basket data analysis, product clustering, catalog design and store layout.

3.1 Vaidya and Clifton Algorithm (VDC)

The first work which recognized and addressed the privacy concerns in vertically partitioned databases is done by Vaidya and Clifton [8]. They proposed a protocol for finding scalar product securely and a solution for mining frequent itemsets preserving owner's privacy which is built based on that protocol. Association rules can be generated from the result of frequent itemset mining and their support values.

The VDC algorithm is designed for only two parties to find out the frequent itemsets having minimum support values. The individual transaction values are not revealed by both the parties. The algorithm is based on the classic data mining algorithm called Apriori [2]. The function apriori-gen is used to generate all candidate itemsets. For each candidate itemset, the algorithm counts the support value by secure computation of the scalar product.

Secure computation of scalar product is the crucial part of the protocol. They proposed an algebraic solution that hides genuine values by putting them in equations masked with arbitrary values.

***Drawbacks*:** First, to find the support of each itemset the parties has to communicate every time. Both parties get the true support of each itemset. This may lead to exposure of information in some cases. The method can only be used to mine boolean association rules. Since the input values are restrained to 0 or 1, it also creates exposure risk.

3.2 Vaidya and Clifton N-Party Protocols (VDCN)

There are two protocols defined in [9]. The first one addresses the scenario where several parties are involved. It refers to the problem of secure computation of the size of intersection. The parties in the protocol have sets of items from a common domain. The target is to compute the cardinality of the intersection of those sets securely. In formal, k parties$P_1, \ldots, P_k$ having local sets $S_1, \ldots, S_k$are given. The aim is to compute$|S_1 \cap \ldots \cap S_k|$ securely.

The basic idea is to encrypt all items using a parametric commutative encryption function. RSA public key encryption is used for that. Each party encrypts their items with their own key and the sets are passed on to another party. The received parties also encrypt the items and pass them to next party. This continues until all parties have encrypted all items. If the values are same the ciphertext generated by different sets will be equal because the commutative property of encryption. Finally any party can calculate the total number of values present in all of the encrypted itemsets. The parties would not know which of the items are present in a particular site.

The other protocol in [9] is called VDCN protocol. It is an extension of the VDC algorithm. VDCN is for N-parties whereas VDC is for two parties. This algorithm uses the first protocol to find the frequent itemsets having minimum support values. The apriori-gen function is used in the algorithm for generating candidate itemsets. For each candidate itemset, the parties run the protocol to compute support value. These support values are shared among all the parties.

***Drawbacks*:** The drawbacks of the two-party algorithm are present here also because the support values and frequent itemsets are known by everyone. A site can learn the exact support of an itemset from the result of secure computation. With that knowledge, the site sayA can find the probability that an item in the set supported by A has a property in another site say B. It can be calculated as the ratio of the actual support to A's support.

3.3 Association rules mining in vertically partitioned databases

The work done by B. Rozenberg and E. Gudes [7] is the most relevant one among different privacy-preserving mining solutions. Two algorithms are proposed by them. The first algorithm is for two parties and the parties are symmetric. One party is named Master and the other party is named Slave. The protocol initiation is done my master. Slave only contributes its information. Only master do global computations. The master finds all frequent itemsets from its own real transactions. Then master check whether those itemsets are present in the slave's real transactions or not. This process is done either by a trusted third party or using secure computation. Then the two parties swap their roles and repeat the method. Second algorithm is for N parties. It is an extension to the first algorithm. It contains N-1 slaves and one master. Master do the global computations and inform the result to slaves.

The mining process is done by master. The fictitious transactions are injected by slaves to their individual database and transmit them to the master. A group of IDs of legitimate transactions are sent to a semi-trusted third party by all slaves. Master discovers association rules from the joined database. Since that database contains fictitious data, master transmits the ID lists of the transactions containing$P \cap Q$for all association rules$P \Longrightarrow Q$to the third party for verification.

***Drawbacks*:** This method cannot be called as an outsourced mining solution since a data owner (named master) performs the computations. The information about the raw data of data owners is available to the master even if it contains fictitious transactions.

3.4 Privacy Preserving Algorithms for Distributed Mining of Frequent Itemsets

In [11], there are two algorithms proposed for vertically partitioned databases and they have two levels of privacy. Both the algorithms are two party algorithms. One is weakly privacy-preserving and the other is strongly privacy-preserving. The transaction database is represented as a boolean matrix. The overview of the context considered is: The private inputs of A and B are$(p_1, \dots, p_n)$ and$(q_1, \dots, q_n)$. The goal is to decide whether$\sum_{i=1}^{n} p_i q_i > s$,where s is a public input.

Probabilistic public-key encryption is the basis of algorithm having weak privacy-level. The algorithm has 3 steps. In Step 1, A encrypts$(p_1, \dots, p_n)$using her *own* public key (so that B cannot decrypt them) and sends the ciphers to B. In Step 2, the encryption of$(p_1 q_1, \dots, p_n q_n)$ is calculated by B from the received ciphertexts. Then the newly generated ciphertexts are re-randomized and re-permuted by B and sent to A. In the last step, the cipher obtained from B is decrypted by A and the numbers of 1s are counted to compare with the threshold.

Anhomomorphic encryption scheme is the basis of the second algorithm having strong privacy level. This algorithm also has 3 steps. In Step 1, A encrypts$(p_1, \dots, q_n)$ using her *own* public key (so that B cannot decrypt them) and sends the ciphers to B. In Step 2, the encryption of$(k_1(T - s - 1), \dots, k_n(T - s - n))$is calculated by B where T is the support count of candidate itemset and k_i is random string. Then the newly generated ciphertexts are re-randomized and re-permuted by B and sent to A. In the final step, Achecks whether there is a ciphertext available which can be decrypted to 0.

***Drawbacks*:** The first solution unravels the true supports, which is not acceptable. The next solution does not reveal the true supports but association rules cannot be discovered based on the result of this method since confidence values cannot be calculated without the true supports. Since computing confidence value securely is very complicated than computing support, this method cannot be used to generate association rules.

3.5 Privacy-preserving mining of association rules from outsourced transaction databases

The work done in [4] deals with mining of frequent itemsets and generation of association rules in privacy-preserving manner when the data is outsourced in the scenario of only one data owner. The data owner encrypts the data items using a substitution cipher in order to maintain the unprocessed data confidential. Since the substitution cipher is vulnerable to frequency analysis attack, a solution made from the concept called ‘k-anonymity’ is used for hiding the true frequencies. K-anonymity is a method used in data anonymization.

Fictitious transactions are added to the database before encryption to resist the attack by frequency analysis. The frequency of each data item will be same as at least k-1 other data items. After the encryption, the data owner exports the encrypted data to the server for performing mining process. The server is supposed to do the mining task using a standard algorithm for frequent itemset mining. Then the results are send to data owner. Since the outsourced database contains fictitious transactions, data owner has to subtract the support of fictitious transactions from the mining results to get the real support values of itemsets. At last, the itemsets which has support higher than a threshold are decrypted.

***Drawbacks*:** To conserve privacy of data in the vertically partitioned databases, using these techniques alone, is not sufficient. To remove the effect of inserted fictitious transactions, the data owner needs to count support of itemsets in fictitious transactions. In the scenario of vertically partitioned databases, it is not feasible to perform the computations using the methods described in [4] by the data owners.

IV. EXISTING SYSTEM

4.1 Overview

The work done in [5] is studied. The system consists of at least two data owners and a third party server or cloud. The private database owned by data owners are encrypted before they export the databases to the server. The data owners assign the responsibility of discovering frequent itemsets and association rules to the server. The server receives the databases from data owners and joins them before starting mining. Then the server run a classic data mining algorithm and sends the result to the data owners.

To encipher the Transaction Database (TDB), all items in the database are encrypted using substitution cipher. A method called frequency analysis where the frequency of ciphertext units are analyzed, has been used to crack some encryption methods such as substitution cipher. This attack is possible here also to find the ciphertext of each data item if adversary knows the real frequency of items. For example, suppose shirt and shoes are the topmost frequent items in a transaction database. If an adversary knows this, he can understand that the top two cipher items having maximum frequencies in the encrypted database are ciphers of shirt and shoes, respectively. To prevent the frequency analysis attack, fictitious transactions are inserted into the database before encryption using k-anonymity method [4]. A Transaction ID (TID) is used to uniquely identify a transaction in a TDB. A customer identification number and date of purchase is used as TID. For the confidentiality of TID, the hash value of TID is stored in the encrypted database instead of storing the original TID.

4.2 Working

There is a private database owned by every data owner. The data owners are willing to collaboratively mine association rules from their databases. A third party server is used for the mining tasks. There are two stages in the solution, namely: preprocessing and mining.

In the preprocessing stage, the data owners encrypt their private databases and the encrypted databases are outsourced to the server. Before encrypting the databases, the data owners insert fictitious transactions to the private databases in order to prevent frequency analysis attack. The k-anonymity method explained in [4] is used for generating fictitious transactions. Substitution cipher is used for encrypting the data items in the database. To detect the genuine and fictitious transactions, data owners mark each transaction with a tag called Realness Value (RV). The RV is either 0 or 1, which indicates whether the transaction is fake or genuine, respectively. The realness value is encrypted using the proposed symmetric homomorphic encryption scheme. Because of the fact that the homomorphic encryption used is probabilistic, the encrypted realness value (ERV) will be different even for the same RV.

In the mining stage, the server runs the classic data mining algorithm called Eclat[10] to discover the association rules. The association rule candidates are generated from the encrypted joint database. There may be candidates which are “false positives” because of the presence of added fictitious transactions. In order to detect them, the server verifies the candidates in a privacy-preserving manner. Using homomorphic addition, the Encrypted Support Verifying Result (ESVR) is calculated from the ERVs by the server. Then the results which include candidates and their encrypted support are sending from the server to data owners. The data owners then verify the encrypted support since it contains support of the fictitious items also. The Encrypted Confidence Verifying Result (ECVR) is also computed in similar manner. At last, the encrypted support and the corresponding association rules are decrypted by the data owners.

4.3 Problem Definition

- The added fictitious transactions increase the size of database to more than double
- Hence the memory consumption and time consumption becomes huge compared to that required for plain data

4.3.1 Known Plaintext Attack

In a known plaintext attack, normally the attacker possesses plaintext-ciphertext pairs. In this system, since customer identification number and date of purchase is used as TID which is known to every customer, a customer can be an attacker. He could purchase an item and locate the corresponding transaction in the encrypted database by comparing the hash value of his ID and date with TID. By repeating the same, he can get several plaintext-ciphertext pairs and break the substitution cipher in a worst case of $O(n)$ steps where n is the total number of items in the database.

V. PROPOSED MODIFICATION

To encrypt the data items, Advanced Encryption Standard (AES) algorithm can be used as a replacement for the vulnerable substitution cipher. The key sizes of AES are 128, 192 or 256 bits. Even if a 128 bit key is used, it provides very strong encryption compared to substitution cipher. For an attacker, a known plaintext attack on AES requires 2^{128} steps for finding the key of encryption.

VI. ANALYSIS

6.1 Confidentiality of TID

There may be sensitive information in the original TIDs of databases owned by data owners. For the confidentiality of such data, the TIDs in the outsourced databases are replaced by their hash values. Because of the property of pre-image resistance of hash function, the server or an attacker cannot get the original TIDs.

6.2 Computational Complexity

To analyze the computational complexity of the system, graphs are plotted as size of data against time taken for different executions. Fig.1 shows the time taken for encrypting the data when AES and substitution ciphers are used. It is clear from the graph that time taken for encryption while using AES is slightly greater than while using substitution cipher.

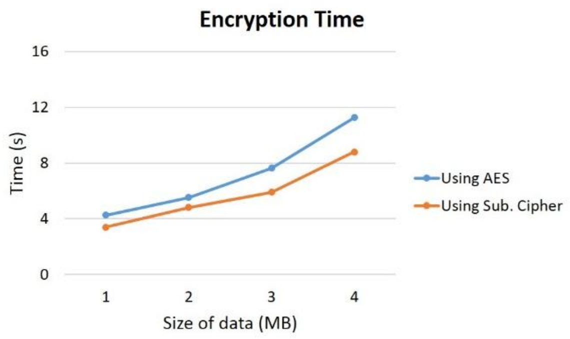

FIGURE 1. Encryption time when AES and substitution ciphers are used

Fig. 2 shows the time taken for decrypting the data when AES and substitution ciphers are used. Since only the data items in the frequent itemsets and association rules have to be decrypted, decryption time is very small compared to encryption time.

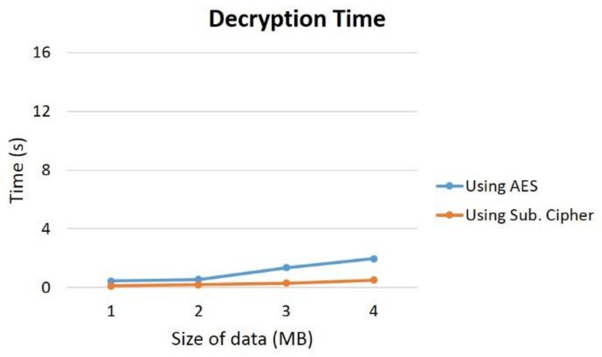

FIGURE 2. Decryption time when AES and substitution ciphers are used

Fig.3 shows the variation in running time of the system when the value of k used for achieving k-anonymity in database is increased. The k-anonymity method is used for inserting fictitious transactions and thus the size of database increases according to the value of k. Since the database size increases as value of k increases the running time will also increases. The data extracted from the datasets retail, chainstore and foodmart[13] are used for the experiment.

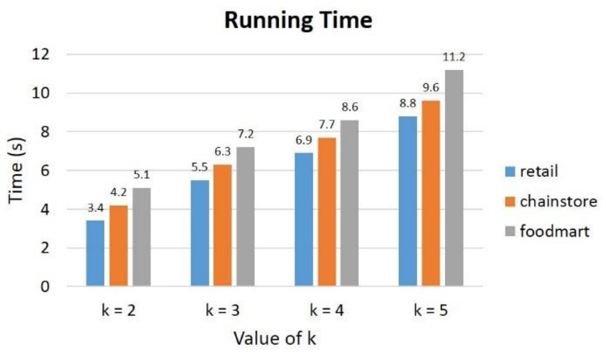

FIGURE 3. Total running time for different datasets

6.3 Communication Overhead and Storage Cost

The data owners communicate with the server for the exchange of encryption parameters as well as outsourcing the data. Other than that there is a mutual understanding among the multiple data owners regarding the selection of keys and parameters. After the mining process also, the server has to send the results back to the data owners. The quantity of communication rounds is not going to change with any other entities.

The private databases of the data owners are stored at their end and the outsourced database is larger than their private databases because of the added attributes like ERV and fictitious transactions. The server who receives multiple databases from different sources combines them and store the joined database for mining. Thus the storage cost at server side is very large compared to data owners. Other than that, the outsourced database size also depends on the chosen value of k by the data owners.

VII. CONCLUSION

The scenario dealt here is the problem of corporate privacy in the cases where the database for mining tasks are outsourced to a third party. The reason for the need of outsourcing is that sometimes the data owners may not have the expertise or resources for performing the mining tasks internally. In this system, multiple data owners can do the mining tasks collaboratively. The paradigm of data mining as a service helps them to perform such tasks. In the stage before outsourcing the database called pre-processing, the database goes through data item encryption using AES and insertion of fictitious transactions with the help of k-anonymity method to counter the frequency analysis attack. A ciphertext tag approach is used to identify genuine and fictitious transactions which is encrypted using homomorphic encryption. The server who receives the outsourced databases from different data owners combines them and perform the mining task using Eclat algorithm. Homomorphic operations are the key to compute the support values of itemsets. The frequent itemsets are found securely and association rules can be generated from those itemsets. Based on the experiments and findings, though the time taken for encryption shows some increase compared to the existing system, the strength of encryption has increased from $O(n)$ to 2^{128}.

REFERENCES

[1] Agrawal R., Imielinski T. and Swami A. M. (1993), 'Mining association rules between sets of items in large databases', *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington*, 207-216.

[2] Agrawal R. and Srikant R. (1994), "Fast algorithms for mining association rules", In Proc. 20th int. conf. very large data bases, VLDB, Vol. 1215, 487-499.

[3] Brijs T., Swinnen G., Vanhoof, K. and Wets G. (1999),"Using association rules for product assortment decisions: A case study", In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 254-260.

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

[5] Li L., Lu R., Choo K. K. R., Datta A. and Shao J. (2016),"Privacy-preserving-outsourced association rule mining on vertically partitioned databases", IEEE Transactions on Information Forensics and Security, 11(8), 1847-1861.

[6] Malik M. B., Ghazi M. A. and Ali R. (2012), "Privacy preserving data mining techniques: current scenario and future prospects", In Computer and Communication Technology (ICCCT), 2012 Third International Conference on IEEE, 26-32.

[7] Rozenberg B. and Gudes E. (2006), "Association rules mining in vertically partitioned databases", Data & Knowledge Engineering, 59(2), 378-396.

[8] Vaidya J. and Clifton C. (2002), "Privacy preserving association rule mining in vertically partitioned data", In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 639-644.

[9] Vaidya J. and Clifton C. (2005), "Secure set intersection cardinality with application to association rule mining", Journal of Computer Security, 13(4), 593-622.

[10] Zaki M. J. (2000), "Scalable algorithms for association mining", IEEE Transactions on Knowledge and Data Engineering, 12(3), 372-390.

[11] Zhong S. (2007),"Privacy-preserving algorithms for distributed mining of frequent itemsets", Information Sciences, 177(2), 490-503.

[12] Han J. and Kamber M. (2006), "Data Mining: Concepts and Techniques", 2nd ed., The Morgan Kaufmann Series in Data Management Systems.

[13] Fournier-Viger P. (2016), "Real-life Datasets in SPMF Format", [Online]. Available: http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

[14] http://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-mining.