PRIVACY-PRESERVING-OUTSOURCED ASSOCIATION RULE MINING ON HORIZONTALLY PARTITIONED DATABASES

[1]V.Sravan Kumar, [2] Dr. Rashmi Agarwal
[1]Research Scholar, [2]Professor
[1,2]Computer Science and Engineering,
[1,2]Madhav university, Rajastan, India

Abstract : In the current technological years, privacy is a major issue in numerous data mining applications. As the information sharing becomes a general demand, instinctively, the information security also turns out to be a serious concern. The advances of data mining techniques played an important role in various applications. In the context of privacy and security issues, the problems caused by association rule mining technique are recently investigated. The misuse of this technique may disclose the database owner's sensitive information to others. Hence, the privacy of individuals is not protected. Many of the researchers have recently made an effort to preserve privacy of sensitive knowledge or information in a real database. The paper deals with privacy preserving on anonymous database and on devising private update techniques to database systems. We propose a protocol for solving this problem on suppression-based and generalization-based k-anonymous and confidential databases. We employed horizontal partitioning method in heterogeneous database system to obtain privacy preserved data mining using the attributed Homomorphic encryption scheme.

IndexTerms - Privacy preserving data mining, Homomorphic encryption, mediated certificate less algorithm and horizontal partitioning.

I. INTRODUCTION

Frequent item set mining and association rule mining are the extensively used techniques in many files as market basket analysis, health care, bioinformatics and web usage mining. The first technique is employed to notice frequently co-occurring data elements in large databases and the second technique is employed to note the association relationships between data items in large transaction databases. A transaction database contains an array of transactions and each transaction contains a group of data elements with a separate Transaction ID (TID). An item set I is viewed as a frequent item set if and only if $Supp(I) \geq T_s$. T_s is a threshold fixed by the data miner. $Supp\ (I)$ denotes the support of item set I. It is also is stated as number of occurrences of item set I in the database. If A and B are the two disjoint item sets, an association rule is given as $A \Rightarrow B$. This association rule denotes that occurrence of set A implies occurrence of set B in the same data transaction with a guaranteed confidence. Similarly $A \Rightarrow B$ is considered as an association rule if and only if $Supp(A \cup B) \geq T_s$ and $Conf\ (A \Rightarrow B) \geq T_c$. $Conf\ (A \Rightarrow B)$ is the confidence of $A \Rightarrow B$. T_c represents the threshold assigned by the data miner. We also remark that the values of T_s and T_c are formed according to the kind of transactions, the manipulation of the mining result, the volume of database. Many association rule mining algorithms are developed from the frequent item set mining algorithms.

Standard frequent itemset mining and association rule mining algorithms, like Apriori [1], Eclat [2],[3] and FP-growth [4], were constructed for a centralized database setting. In this centralized database, the raw data is kept in the central site for mining. However privacy concerns are not accounted in this setting. Vaidya and Clifton and Kantarcioglu and Clifton [5] first identified and explained the privacy concerns in horizontally/vertically partitioned databases. After getting awareness of the consequence of data privacy, many privacy-preserving data mining algorithms have been presented recently.

Due to an increased understanding of the importance of data privacy, a number of privacy-preserving mining solutions have been proposed in recent times. In their proposals, there are multiple data owners desiring to study association rules or frequent itemsets from the joint data. Still, the data owners do not wish to transmit their original data to a central site in order to protect their privacy. If there are one or more rows for each data owner in the joint database, then the database is known as horizontally partitioned. If there are one or more columns for each data owner in the joint database, then the database is known as vertically partitioned. Only some of the conventional solutions employ a third-party server to server for data mining. Some solutions employ asymmetric encryption to process the supports of item sets, whereas other solutions employ a protected scalar product protocol, a set intersection cardinality protocol or a secret sharing protocol to accomplish these computations. The common disadvantage with the conventional data mining systems is that mining process employ third party server services.

The proposed system manages certificate less approach based data partitioning in distributed network with using Mediated Certificate less Algorithm to keep secure and sensitive of partitioned data. An efficient attributed homomorphic encryption scheme is being proposed. The advantage of the proposed system is that the valid user can extract with key issue in partition data in automated approach. The feature of the proposed system is managing data in horizontal partitioning. The partition data is converted sensitive format by using Mediated Certificate less Algorithm. If any valid user wants to review their original sources, they must submit valid attribute to extract heterogeneous databases.

II. LITERATURE SURVEY

The authors in [6] presented a framework for developing hierarchical categorical clustering algorithm on horizontal and vertical partitioned dataset. It is anticipated that data is dispersed between two events, such that for common profits, both are eager to sense the clusters on the entire dataset. However they decline to distribute the actual datasets because of privacy concerns. Hence, they presented algorithms by using secure weighted average protocol and secure number comparison protocol in order to securely determine the required criteria in constructing clusters' scheme. The authors in [7] considered the concern of association rule mining from distributed vertically partitioned data with the intention of keeping the confidentiality of each database. Each site contains some attributes of each transaction, and the sites would like to work together to discover globally valid association rules without exposing separate transaction data. The limitation of

paper is the privacy/performance tradeoffs both analytically and experimentally and test these tradeoffs with different support values. In particular, to analyze the general case and arrive at privacy loss measure under general assumptions on the databases. The authors in [8] the author proposed secure data mining of association rules in horizontally partitioned data. They include cryptographic techniques to decrease the distributed information, during the insertion of little overhead to the mining operation. The concern is to mine association rules through two databases, where the columns are at different sites, separating each row. One database is labelled the master and the initiator of the protocol whereas another database is the slave. A join key exist in these two databases and the remaining attributes exist in one database or the other. However, the attributes do not exist in both databases. The goal is to find association rules involving attributes other than the join key. The authors in [9] presented a novel classification technique known as Classification based on Predictive Association Rules (CPAR). This technique merges the merits of associative classification technique and conventional rule-based classification technique. Associative classification technique develops numerous candidate rules. However, CPAR applies a greedy algorithm to obtain rules directly from the training data. In order to test all the significant rules, CPAR creates and tests many rules comparing with the conventional rule-based classifiers. The authors in [10] proposed efficient algorithms to find out frequent itemsets developing the compute intensive phase of the task. The algorithms exploit the structural properties of frequent itemsets to enable fast discovery. In [11] the authors present frequent pattern tree (FP-tree), for keeping compressed, crucial data about frequent patterns, and presented a pattern growth method, FP-growth, for effective mining of frequent patterns in large databases. Frequent pattern tree (FP-tree) structure, which is an extended prefix-tree structure for keeping compressed, crucial information about frequent patterns, and progress an efficient FP-tree-based mining method, FP-growth, for mining the entire set of frequent patterns by pattern fragment growth. In [12] the author proposed Apriori and AprioriTid for determining all important association rules between items in a large database of transactions. Furthermore, the execution time reduces a small as the number of items in the database increases. The average transaction size rises (while keeping the database size constant) the execution time increases only progressively. The authors [13**]** consider the problems of sensing new, unexpected, and interesting designs in many fields such as hospital infection control and public health surveillance data. They presented a new data analysis method and system based on association rules to address this issue. The new process and system are efficient and effective in identifying new, unexpected, and interesting patterns in surveillance data. The clinical relevance and utility of this process await the results of prospective studies currently in progress. The author presented a safe protocol for numerous parties to manage the anticipated computation. The solution is dispersed, there is no central, trusted party having retreive to all the data.

III. METHODOLOGY

The partitioned data in the distributed network are reserved securely through the Mediated Certificate less Algorithm. In this paper, a dynamic Homomorphic encryption is proposed. In many existing works, asymmetric encryption, secure scalar product protocol, and a secret sharing scheme were employed for computation. The present system employs third party server to server mining methods for computation and vertical partitioning technique for data partitioning and mining is done by vertical partitioning and thus it becomes a major drawback. In the proposed system, this drawback is neglected as the valid user can extract with key issue in partitioning data through an automatic approach. In this paper, horizontal partitioning is used since the data has the ability to partition itself.

The features of the proposed system have many advantages over the existing system. The important features are listed below:

1) Management of data in horizontal partitioning.
2) Conversion of partitioned data to sensitive data.
3) Submission of valid attribute to extract heterogeneous database.

Here, the homomorphic encryption is proposed and thus a secured outsource comparison scheme is constructed. This is done on the basis of our privacy preserving mining solutions. As discussed earlier, in the existing system, the homomorphic encryption schemes are asymmetric in general. In this paper, by using only the modular additions and multiplications – a symmetric homomorphic encryption scheme is implemented.

Homogeneous database systems can be designed easily. This method offers incremental growth and the insertion of a new site to the Distributed Database Management System (DDBMS) is easy, and the ability of parallel processing of multiple sites improves its performance. When the individual data owners have employed their own database, Heterogeneous system will be obtained. In this system, translations are needed to establish communication between Database Management System (DBMS). To obtain transparency in the heterogeneous system, the end users should rise query using the language used at the local site. Then the system performs the required translation. This system is complex but it provides various advantages.

- Large amount of data are stored in one global center.
- Remote access is done using the global scheme
- Different DBMSs can be employed at each node
- It is not always expected that all the data owners have same kind of databases

Horizontal partitioning versus vertical partitioning

In horizontal partitioning, different rows are created as different tables. In a certain DB, the customers having ZIP codes below 50000 are stored in table named CustomersEast, whereas the customers having ZIP codes greater than 50000 are stored in table named CustomersWest. These two partition tables can provide complete information of all customers if a union is created using both tables. Horizontal partitioning splits a table into multiple tables which consists of the same number of columns, but fewer rows.

In vertical partitioning tables with fewer columns are generated and additional tables are employed to store the remaining columns. Normalization is a process of splitting the columns across tables, but vertical partitioning is the process of splitting the columns even they are already normalized.

3.1 System Design

The proposed system architecture consists of two or more data owners and a cloud. Each data owner owns a private database, and the data owners encrypt their private databases before outsourcing the encrypted databases to the cloud. Data owners can also ask for the cloud to mine association rules or frequent itemsets from the joint database on their behalf. The honest but curious cloud compile and store

the databases obtained from various data owners, the mining of association rules or frequent itemsets for data owners, and the transfer of the mining report to relevant data owners.

The cloud is reflected as honest but curious in this paper. Primarily, the cloud honestly keeps and mines data for data owners. Data owners reimburse for the cloud's services, and they will take a cloud thought to be honest (e.g. a cloud provider with a trusted reputation). Many techniques were presented to notice dishonest clouds and dishonest clouds are identified by just associating the results obtained from mining the different clouds. Then, the cloud is provoked to acquire the data of data owners for financial benefits. The cloud know the private database of the owners as it contains the background knowledge of some elements. This allows the attackers to attack easily and the cloud is able to detect the cipher text of an item easily. The data owners like to know the mining result, and are ready to share the information with each other. Thus each data owner know some details of the private databases of other data owners. Thus each data owners get benefit from the collaborative mining.

3.2 Mediated Certificateless Cryptosystem

To maintain the privacy of sensitive data stored in cloud, the data encryption is usually employed to before uploading the data in the cloud. The cloud is not aware of the keys employed to encrypt the data. Thus the privacy of the data in the cloud is ensured. However, every company necessitates fine-grained encryption techniques. A general approach used for encryption based access is symmetric key encryption. In this scheme, a private key is used for encryption and decryption. The key based encryption schemes decrease the number of keys to be managed. But symmetric key based encryption schemes require more costs for key management. Public key cryptography is employed as an alternative solution to decrease the cost of key management. In this system, a public key and private key are employed.

The existing public key cryptosystem entails a trusted Certificate Authority to deliver digital certificates to make binding between the user and their public keys. The certificate management requires more cost and has more complexity. For these drawbacks, identity based public key cryptosystem was presented, but key escrow problem was happened which is described that the key generation center may know about the private keys of all users. Then, attribute based encryption is presented. But in this scheme along with the key escrow problem certificate, revocation problem was occurred. Hence, certificateless encryption has been proposed to overcome these drawbacks.

Among the many available symmetric key encryption schemes, The Advanced Encryption Standard (AES) is the secure method. One of this Standard is Rijndael algorithm. There are many differences between AES and Data Encryption Standard (DES). In DES, key size and block size are fixed whereas in AES key size and block size can be varied with the use of the Rijndael algorithm. Though the symmetric key encryption can be applied for large data encryption and decryption, it has the shortcoming that the keys can be shared. When a malicious person comes to know the secret key he can decrypt all the data encrypted using this key. Another method is the public key cryptography in which public key and a private key are available and either of them can be used for encryption. If public key is used for encryption then the data must be encrypted with the help of the public key. Hence the public key of the receivers must be known by all. Then the message is decrypted at the receiver side by using the private key. But this method invloves a trusted third party called certificate authority. When a malicious person comes to know the private key used for encryption, he can decrypt the message of the owner of the private key. It is not similar to the symmetric key encryption. The disadvantage of public key approach is that overall certificate management requires more cost, more time and also has certificate revocation problem. For reducing these kind of shortcomings identity based encryption has been proposed. Here any user can create its public key from a known identity which may be ASCII string. A private key generator is used for this purpose. Here the authorized user can create public key of any other user with the help of identity and master key. The advantage of this kind of encryption is that certificates are not required for this scheme. The receivers public key is obtained mathematically from its identity and master key is obtained from private key generator. The disadvantages of this scheme is that it is centralized approach and key generation center knows about the private key. It is referred to as key escrow problem. Hence to remove the key escrow problem certificateless encryption is incorporated with identity based encryption where the private key generation procedure is distributed to the user and the server. It is not identity based approach since the public key will not be created by the identity only. In addition, the private key is not revealed to key generation center. Hence the key escrow problem is removed. The drawback of identity based and security mediated cryptosystems is that they require a trusted third party to create keys of all entities. This is known an escrow problem. To avoid key escrow problem completely certificateless cryptosystem has been proposed. Each entity have public key but it does not have certificate. The identity string is employed to guarantee that only the correct entity has the private key corresponding to the public key. But this method fails to provide how the instant revocation can be obtained when required. This problem is solved in this work. The shortcoming of these methods is that it uses bilinear pairing which requires more cost. When this scheme is applied into cloud computing, if many users are access same data, the encryption cost becomes high. Hence the data owner has to encrypt the same data encryption key many times.

In this work a mediated certificate-less encryption is applied without pairing operations for obtaining privacy in transmitting data in public clouds. Mediated certificate-less public key encryption (mCL PKE) offers the solution for the key escrow problem presented in identity based encryption and certificate revocation problem presented in public key cryptography. However, the conventional mCL- PKE encryption schemes are inefficient as it needs expensive pairing operations. In this paper, we presented a mCL PKE method without using pairing operations. This mCL PKE method is employed to obtain a solution to the problems of distribution of critical and confidential information in clouds. In our system, the data owner encrypts his data by using the cloud generated users' public keys. These keys are generated based on access monitoring policies. Then the cloud partially decrypts the encrypted data for the authorized users. Then the user fully decrypt the data with the help of the private key given to it.

IV. PROPOSED METHOD

4.1 Symmetric Homomorphic encryption scheme

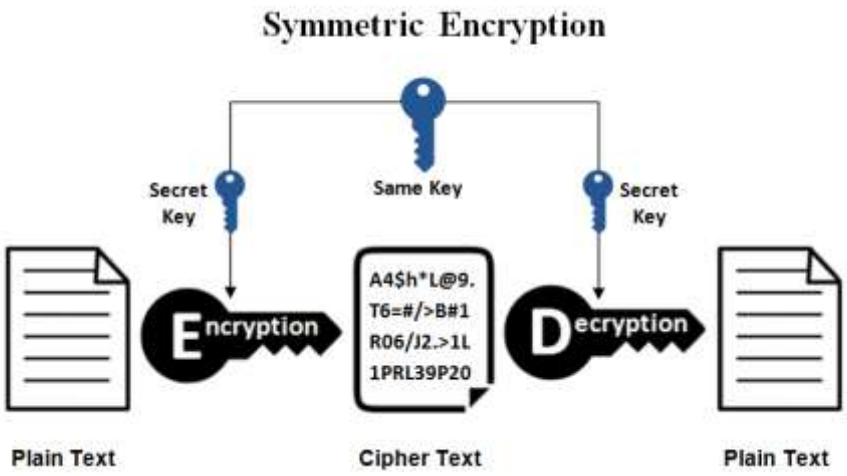

Fig. 1 Symmetric key based data confidentiality

Symmetric key encryption is the simpler method which requires only one secret key to encrypt and decrypt information. It has a secret key which can be a number, a word or a string of random letters. It is merged with the plain text of a message to modify the content in a certain way. The secret key must be known by the sender and the receiver so that the messages can be encrypted and decrypted. Blowfish, AES, RC4, DES, RC5, and RC6 are the kind of symmetric encryption. The generally used symmetric algorithm is AES-128, AES-192, and AES-256. The major disadvantage of the symmetric key encryption scheme is that all sectors involved in the cloud must exchange the key employed for encryption so that the decryption can be performed at the other end.

The symmetric method is considered as more efficient than the earlier techniques, and it comprises of three main algorithms. They are:

1) Key generation algorithm
2) Encryption algorithm
3) Decryption algorithm

4.1.1 Key generation algorithm KeyGen( )

$$(s,q,p) \leftarrow \text{KeyGen}(\lambda)$$

The key algorithm is denoted as **"KeyGen()"** and it comprises of a security parameter **"(λ)"** which is taken as input. The output is a secret key **"SK = (s,q)"** and **p** is taken as a public parameter. Here, both **"p"** and **"q"** are big primes, where **"p >> q"**. The bit length of **"q"** depends on the input security parameter, and **"s"** is a random number from **"Z^*_p"**. Key algorithm is basically a probabilistic algorithm.

4.1.2 Encryption algorithm E( )

$$E(SK, m, d) = s^d (rq + m) \bmod p$$

The encryption algorithm is also an probabilistic algorithm. Here the inputs are a secret key **"SK"**, a plaintext **"$m \in F_q$"** and also with a parameter **"d".** The algorithm output is a ciphertext **$c \leftarrow E(SK, m, d)$**. The parameter **"d"** is basically called as d-degree ciphertext as it is a small integer. Let **"r"** be a big random positive integer and where the bit length satisfies $|r| + |q| < |p|$. The encryption of a plaintext **"m"** is denoted by **"E(m)"** in short.

4.1.3 Decryption algorithm D( )

$$D(SK, c, d) = (c \times s^{-d} \bmod p) \bmod q$$

The decryption algorithm is a deterministic algorithm. It takes a secret key **"SK"**, a plaintext **"$c \in F_q$"** and also with a parameter **"d"** as inputs. The algorithm output is a ciphertext **$m \leftarrow D(SK, c, d)$**. Let s^{-d} represent the multiplicative inverse of s^d in the field F_p. the correctness proof of the algorithm is given below.

$$\begin{aligned} D(SK, c, d) &= (c \times s^{-d} \bmod p) \bmod q \\ &= ((s^d (rq + m) \bmod p) \times s^{-d} \bmod p) \bmod q \\ &= (rq + m) \bmod q \\ &= m \end{aligned}$$

4.2 Proposed- asymmetric key based data confidentiality improves data security

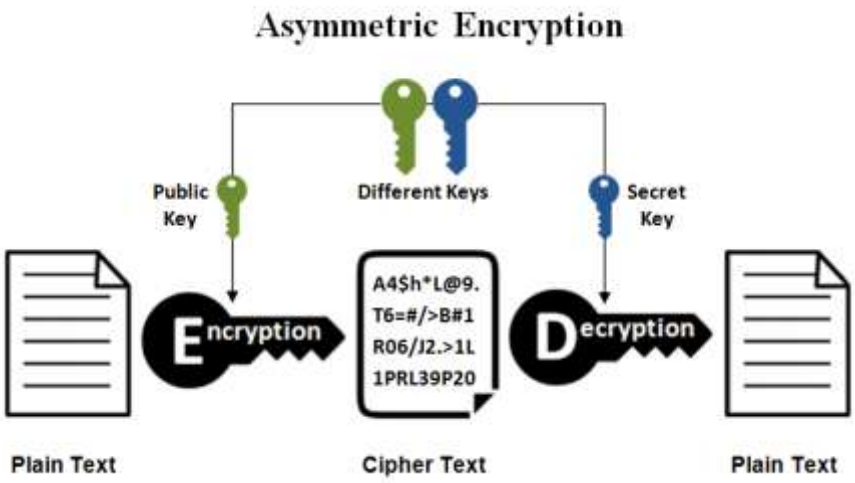

Fig.2 Asymmetric encryption scheme

Asymmetrical encryption is a different method, which involves two keys to encrypt a plain text. It guarantees that mischievous persons do not mishandle the keys. Asymmetrical encryption uses two related keys for improving the security. A public key is made available to everyone who wants to store data in the cloud. The private key is kept as a secret key, which cannot be known by others.

In this scheme, a message encrypted using a public key can be decrypted using a private key only and vice versa. Security is not required for the public key since it is available publicly and can be distributed over the cloud. Asymmetric encryption is regularly used in day-to-day communication, specifically over the Internet. The general asymmetric key encryption algorithm includes EIGamal, RSA, DSA, Elliptic curve techniques, PKCS. In asymmetric encryption, there is no need to share the key as in the symmetrical encryption model since pair of keys are used; public and private keys. Asymmetric encryption consumes comparatively more time than the symmetric encryption.

4.2.1 Property of the proposed Homomorphic encryption:

The proposed Homomorphic encryption comprises of three properties. They are

1) Homomorphic multiplication
2) Homomorphic addition
3) Homomorphic subtraction

4.2.1.1 Homomorphic multiplication

Consider two ciphertexts as c1 and c2, of two plain texts m1 and m2. For some random ingredients r1 and r2, we have c1 = sd1 (r1q+m1) mod p ; and c2 = sd2 (r2q+m2) mod p. As shown below, given d1 degree ciphertext c1 and d2 degree ciphertext c2, the d1+d2 degree ciphertext of m1 x m2 can be computed with a modular multiplication. To correctly decrypt m1 x m2 from its ciphertext, (r1r2q + r1m2 + m1r2)q + m1 x m2 < p must be satisfied where (r1r2q + r1m2 + m1r2) is the random ingredient. Therefore, we choose the bit lengths which satisfy the condition and ensure the correctness of decryption. It is not hard to do so, as |q| > |m1| and |q| > | m2| and we have |r1r2q| > |r1m2| + |m1r2|. We only need to ensure |r1| + |r2| + 2|q| + 1 < |p|.

$(c1\times c2) \bmod p = sd1\ (r1q + m1) \bmod p \times sd2\ (r2q + m2) \bmod p$

$= sd1+d2\ (r1r2q2 + r1qm2 + m1r2q + m1 \times m2) \bmod p$

$= sd1+d2\ ((r1r2q + r1m2 + m1r2)\ q + m1 \times m2) \bmod p$

4.2.1.2 Homomorphic addition

The ciphertext of m1 + m2 mod q can be computed by a modular addition of c1 and c2 if d1 = d2. To correctly decrypt m1 + m2 from its ciphertext, (r1 + r2) q + m1 + m2 < p must be satisfied. Therefore, we choose the bit lengths of p,q and random ingredients to ensure that all ciphertexts in our privacy – preserving mining solutions can be decrypted correctly.

$c1 + c2 \bmod p = sd1\ (r1q + m1) \bmod p + sd2\ (r2q + m2) \bmod p$

$= sd1\ ((r1r2)\ q + m1 + m2) \bmod p$ if d1 = d2

4.2.1.3 Homomorphic subtraction

Similarly, homomorphic subtraction can also be achieved with a modular subtraction. To correctly decrypt m1 – m2 from its ciphertext, r1 – r2 must be satisfied.

$c1 - c2 \bmod p = (sd1\ (r1q + m1) - sd2\ (r2q + m2)) \bmod p$

$= sd1\ ((r1r2)\ q + m1 - m2) \bmod p$ if d1 = d2

4.2.1.4 Degree alignment for homomorphic addition/subtraction

Ciphertexts sharing the same degree is required for homomorphic addition and subtraction. If c1 and c2 have different ciphertext degrees, then after upgrading the lower degree cipertext to higher degree ciphertext – the homomorphic addition and subtraction can be performed. If suppose, c2's degree d2 is lower. A d1 degree cipertect of m2, c2 can be computed by doing a homomorphic multiplication of c2 and a (d1 – d2) degree ciphertext of 1. Then the homomorphic addition/subtraction can be performed.

4.2.1.5 Homomorphic scalar multiplication

Consider m1's ciphertext c1 and a plaintext m2, the ciphertext of m1 x m2 can be computed with a modular multiplication. To correctly decrypt m1 x m2 from its ciphertext, r1m2q + m1 x m2 < p must be satisfied. Therefore, we choose the bit lengths that the condition is satisfied which will ensure the correctness of decryption.

$(c1 - m2) \bmod p = sd1\ (r1q + m1) \bmod p \times m2 \bmod p$

= sd1 ((r1m2q + m1 x m2) mod p

4.3 Privacy preserving outsourced association rule mining

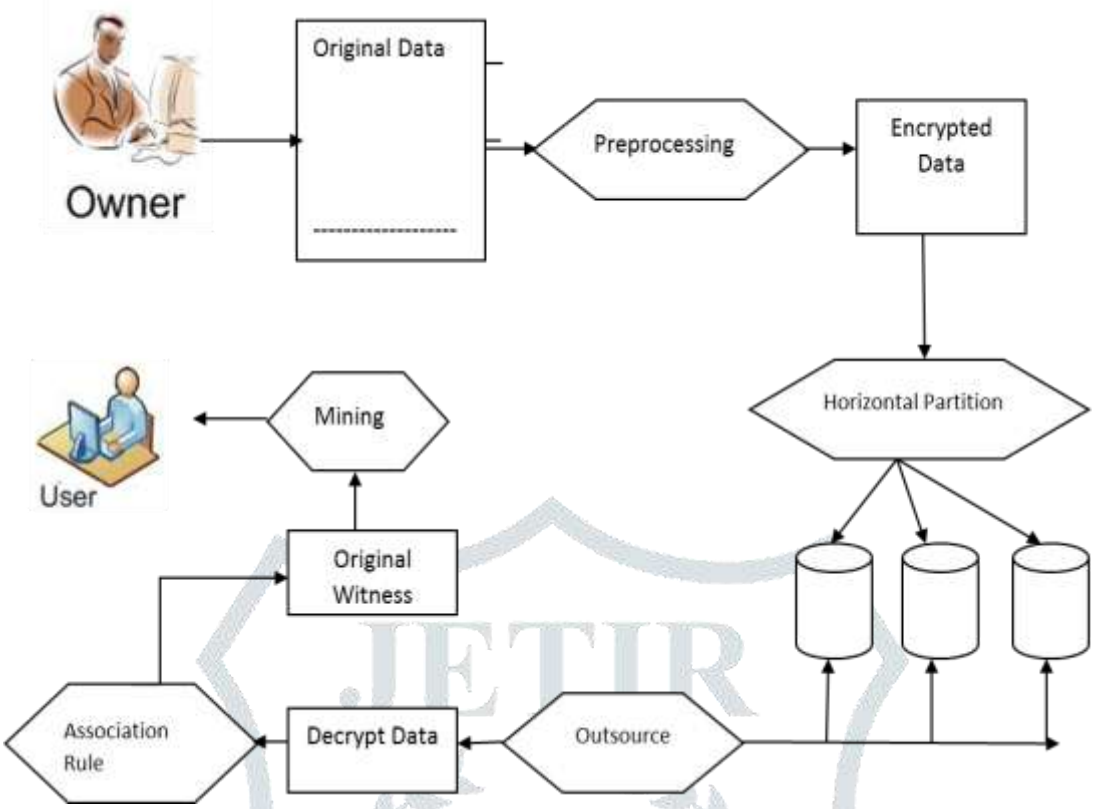

Fig.3 Privacy Preserving Association Rule Mining Architecture

The above figure shows that association rule mining approach for partition databases. The figures shows the user attribute is used to convert data into sensitive. The convert data will be stored into partition dbs. By using the homomorphic encryption scheme and secure comparison scheme – the privacy preserving association rule mining and frequent itemset mining solution can be presented.

Each data owner owns a private database and with the help of the cloud, the data owners cumulatively mine their joint database's association rules. Basically, the association rule mining solution includes two levels, they are – Preprocessing and Mining.

Initially, in the preprocessing stage – data owners and the cloud combines to produce an encrypted joint database at the cloud's end and some auxiliary data for privacy preserving mining. Each data owner injects duplicate/imaginary transactions to his private database, and encrypts items in the database with a replaced cipher. Due to the intrinsic weakness of the replaced cipher the duplicate transaction is done, which reduces the frequency analysis attacks. When the encryption of database is done, they are outsourced as a part of joint database to the cloud, which is maintained by the cloud. To permit the cloud to mine the database accurately (which actually has the duplicate transactions), the data owners – by using the customized homomorphic encryption scheme, name each transactions with an encrypted realness value (ERV) in their outsourced database and joint database. To identify the real transaction from duplicate transaction, the Realness value (RV) is used which indicates with 0's and 1's. Even after sending all the ERV's to the cloud, the cloud will be unable to predict the real transactions from the duplicate.

Secondly, in the mining stage – the cloud mines associates for the data owners in a privacy preserving manner. From the encrypted joint database, the cloud mines association candidates. Few candidates will be "False positives", due to the existence of the duplicate transactions. The cloud checks the candidates in the privacy preserving manner in order to permit the data owners to identify the "false positives". By utilizing the homomorphic encryption and secure comparison schemes, each candidate's encrypted verifying result from the ERVs are computed by the cloud. All candidates and their encrypted verifying results are returned back by the cloud to the data owners. Finally, to recover the real association rules the data owners decrypt the encrypted verifying results and the association rule candidates.

The concept of the frequency itemset differs only in the mining stage, but it is similar in the preprocessing stage. Instead of the association rule candidates, the cloud mines frequent itemset candidates, in the mining stage. To recover the real frequent itemsets, the data owners decrypt the encrypted verifying results and frequent itemset candidates.

4.4 Frequent itemset mining solution

The frequency itemset mining solutions in the t-data owner setting is described and are shown below with the tables 1, 2 and 3 as examples.

4.4.1 Preprocessing stage

4.4.1.1 Initialization for homomorphic encryption

Consider D_1, D_2, D_3, ….. D_t be the data owners. Let data owner D_1 runs KeyGen(λ) to produce a secret key SK and a public parameter p of the proposed homomorphic encryption scheme. p is shared with other data owners and the cloud, while SK is shared only with data owners.

The bit lengths of keys and the parameters are carefully selected depending not only on the security parameter λ, but also on the estimated maximum ciphertext degree and joint database size. In this process, it is done to utilize the proposed homomorphic encryption and

outsourced secure comparison schemes. D1 will select the bits and the lengths of the bits are verified by the other data owners to satisfy the selection rules.

4.4.1.2 Initialization for secure threshold comparison

Each data owner computes a 1 – degree ciphertext of "1", and sends it to the cloud, to enable outsourced secure comparison. Let μ_i be the ciphertext produced by i-th data owner.

Consider data owner (D_1) computes c_s = E(SK, - T_s mod q, 1) and c_e = E(SK, 1, 1). The owner sends c_s, c_e along with T_s to the cloud. The cloud also sends the received c_s, c_e and T_s to other data owners for verifying correctness. This is to prevent D_1 from cooperative mining protocol.

4.4.1.3 Insertion of fictitious transactions

The data owner hides the frequency of data item with the use of fictitious transactions. The insertion of fictitious transactions makes each items to share the equal frequency with other items in the same database. The number of other item is k-1. The value of k should be high so that the frequency analysis attack in the cloud can be reduced.

4.4.1.4 Homomorphic encryption

This scheme encrypt RV and generate ERV. The value of RV is obtained as below.

$$RV = \begin{cases} 0 & if\ transaction\ is\ fictiitous \\ 1 & if\ transaction\ is\ not\ fictitious \end{cases}$$

In this paper, user attribute property is chosen, hence the cloud is not able to detect if the two ERV share the same text.

4.4.1.5 Database outsourcing

Each data owner broadcasts his encrypted database with ERV to the cloud, and the cloud merges received data by TIDs to generate a joint database. Then the association rule is employed to get the mined results.

V. PERFORMANCE COMPARISON

The performance of the proposed Homomorphic encryption based horizontal partitioning approach are analysed and compared with the existing vertical partitioning approach. In order to evaluate the performance metrics the proposed method is implemented for heterogeneous databases. We considered three databases, namely, SQL server, MySQL server and MS Access. Three parameters are measured in this work for performance comparison. They are partitioning latency, mining latency, performance factor. Partitioning latency is defined as the time taken for partitioning. The table shows the average value obtained on three databases. Mining latency is defined as the time taken for obtaining result by arising query. Performance factor is defined as the time required for taking the overall trend of a particular attribute. Thus the mining latency is defined as the time taken for retrieving certain data of an attribute whereas the performance factor is defined as the time taken for retrieving the overall information (showing the history) of an attribute. From the table it is observed that the partitioning latency is reduced in the proposed method by 10.8%. The mining latency is reduced by 14.7%. The performance factor is reduced by 27%. Thus, the proposed horizontal partitioning approach is proved as a better solution for heterogeneous data mining comparing with the vertical partitioning approach. In [14], the vertical partitioning approach is tested on homogenous databases. In this work, we have implemented the approaches on heterogeneous databases. The data mining approach should perform well for the homogenous databases as well as heterogeneous databases since the data owners do not have same kind of databases. By horizontal partitioning, the security is enhanced as the data irrelevant to a group are separated from the data relevant to it. Since the encryption based approach is employed the process of insert, delete and update at the other end is not possible.

Table 1 Performance analysis

Criteria	Vertical partitioning [14]	Horizontal partitioning
Partition latency	1020	910
Mining latency	340	290
Performance Ratio	850	620

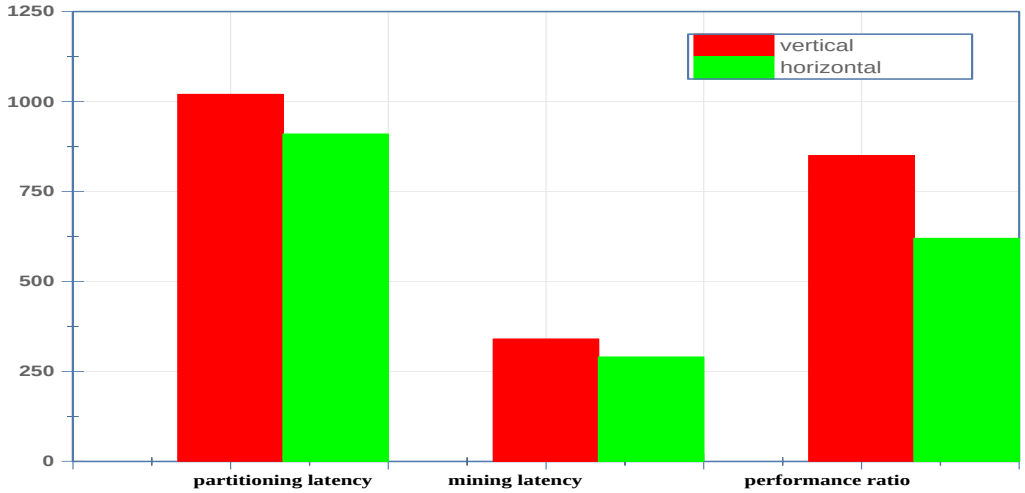

Fig.4 Performance analysis

VI. CONCLUSION

We proposed a privacy-preserving outsourced frequent itemset mining solution for horizontal partitioned databases. This allows the data owners to outsource mining task on their joint data in a privacy-preserving manner. Based on this solution, we built a privacy-preserving outsourced association rule mining solution for horizontal partitioned databases. Our solutions also ensure the privacy of the mining results from the cloud. Compared with most existing solutions, our solutions leak less information about the data owners' raw data. Our evaluation has also demonstrated that our solutions are very efficient; therefore, our solutions are suitable to be used by data owners wishing to outsource their databases to the cloud but require a high level of privacy without compromising on performance.

To realize our solutions, an efficient homomorphic encryption scheme and a secure outsourced comparison scheme were presented in this paper. Both schemes have potential usage in other secure computation applications, such as secure data aggregation, beyond the data mining solutions described in this paper. Demonstrating the utility of the proposed homomorphic encryption scheme and outsourced comparison scheme in other settings will be the focus of future research.

REFERENCES

[1] Agrawal, R. and Srikant. R. 1994. Fast Algorithms for Mining Association Rules. Morgan Kaufmann, San Mateo, CA, USA Proceedings of the 20th international conference on very Large Databases (VLDB, Santiago de Chile), 1215: 487–499.

[2] Agrawal, R. Mannila, H. Srikant, R. Toivonen, H. and Verkamom, A. 1996. Fast discovery of association rules. Advances in knowledge discovery and data mining, 12(1): 307-328.

[3] Zaki, M. Parthasarathy, S. Ogihara, M. and Li, W. 1997. New Algorithms for Fast Discovery of Association Rules. Menlo Park, CA, USA, Proceedings of the 3rd international conference on Knowledge Discovery and Data Mining, 97: 283-286.

[4] Han, J. Pei, H. and Yin, Y. 2000. Mining frequent patterns without candidate generation. New York, USA In ACM sigmod record, 29(2): 1-12.

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

[6] Sheikhalishahi, M. and Martinelli, F. 2017. Privacy preserving clustering over horizontal and vertical partitioned data. IEEE Symposium on Computers and Communications (ISCC), 1237-1244.

[7] Gudes, E. and Rozenberg, B. 2004. Collaborative Privacy Preserving Frequent Item Set Mining in Vertically Partitioned Databases. Data and Applications Security XVII, 91-104.

[8] Kantarcioglu, M. and Clifton, C. 2004. Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE transactions on knowledge and data engineering, 16(9): 1026-1037.

[9] Yin, X. and Han, J. 2003. CPAR: Classification based on predictive association rules. Proceedings of the 2003 Society for Industrial and Applied Mathematics International Conference on Data Mining, 331-335.

[10] Zaki, M.J. Parthasarathy, S. Ogihara, M. and Li, W. 1997. New Algorithms for Fast Discovery of Association Rules. In KDD, 97: 283-286.

[11] Han, J. Pei, J. Yin, Y. and Mao, R. 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data mining and knowledge discovery, 8(1): 53-87.

[12] Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. Proceedings of the 20th international conference on very large data bases, VLDB, 1215: 487-499.

[13] Brossette, S.E. Sprague, A.P. Hardin, J.M. Waites, K.B. Jones, W.T. and Moser, S.A. 1998. Association rules and data mining in hospital infection control and public health surveillance. Journal of the American medical informatics association, 5(4): 373-381.

[14] 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.