AN ACCESS CONTROL MECHANISM WITH PRIVACY PROTECTION FOR RELATIONAL DATA

Anusha Hiremath[1], Shrikanth Athanikar[2]

[1]PG Scholar, [2]Assistant Professor, Dept of CSE,

KLE DR MS Sheshagiri College of Engineering and Technology, Belgaum (India)

ABSTRACT

Privacy Preserving Data mining (PPDM) is the new territory of examination that studies the reactions of learning mining systems on people and associations security. With the advancement of data mining innovation, an expanding number of information can be mined out to uncover some potential data about client. While this will prompt an extreme issue, which is clients' protection may be damaged effectively. The objective of privacy preserving is to mine the potential significant information without spillage of sensitive records. An Access Control Mechanism with privacy protection has been proposed in this paper. The access control mechanism characterizes the authorization to get to the asked for traits while the privacy protection mechanism is to conceal the sensitive records. The idea of imprecision bound has been utilized to fulfill the privacy prerequisites.

Keywords: Sensitive Attributes, Privacy, Access Control

I. INTRODUCTİON

Associations gather and dissect purchaser information to enhance their administrations. Access Control Mechanisms (ACM) are utilized to guarantee that just approved data is accessible to clients. Nonetheless, sensitive data can even now be abused by approved clients to bargain the security of buyers. The idea of privacy protection for sensitive information can require the authorization of security arrangements or the assurance against personality divulgence by fulfilling some security prerequisites. Privacy-preservation from the anonymity aspect has been investigated. The sensitive data, even after the removal of identifying attributes, is still helpless to connecting assaults by the approved clients. This issue has been concentrated on broadly in the territory of small scale information distributed and security definitions, e.g., k-anonymity, l-diversity, and variance diversity. Anonymization calculations use concealment and speculation of records to fulfill security necessities with negligible contortion of smaller scale information. The The anonymity techniques can be utilized with an access control mechanism to guarantee both security and protection of the sensitive data. The security is attained to at the expense of exactness and imprecision is presented in the approved data under an access control policy.

The idea of imprecision bound is utilized for every authorization to characterize a limit on the measure of imprecision that can be endured. Existing workload aware anonymization systems minimize the imprecision total for all inquiries and the imprecision added to every consent/question in the anonymized miniaturized scale information is not known. Making the security prerequisite more stringent (e.g., expanding the estimation of k or l) brings about extra imprecision for inquiries. The issue of fulfilling precision imperatives for individual

authorizations in an arrangement/workload has not been mulled over anytime recently. The heuristics proposed for precision compelled protection saving access control are additionally important in the setting of workload-mindful anonymization. The anonymization for persistent information distributed has been mulled over in writing. The emphasis is on a static social table that is anonymized just once. To epitomize the methodology, part based access control is accepted.

The paper is organized as follows: section 2 contains the related work, section 3 contains the background and section 4 contains the proposed work.

II. RELATED WORK

The first and most closely related work is the oracle's virtual private database model (VPD) [1]. VPD is a collection of fine grained access control enforced by the server along with the secure application context in the oracle9i database server. It enables to create security policies to control database access at row and column level. It also provides the users a flexible mechanism to build the applications that enforce the security policies they want enforced i.e. only where such control is necessary. VPD enforces the security at a fine level of granularity directly on database tables, views or synonyms. There is no way to bypass the security since the policies are attached directly to the database objects and these policies are automatically applied when the user access the data. Whenever the user accesses a table, a view or a synonym which is protected with an oracle virtual private database policy, oracle database dynamically modifies the SQL statement of the user. This modification contains a WHERE condition (called as a predicate) returned by the function implementing the security policy. Oracle Database modifies the statement dynamically, transparently to the user, using any condition that can be expressed in or returned by a function.VPD offers benefits such as lower cost of ownership, elimination of application security problem, application transparency, and new business opportunities. On the other hand it has several disadvantages: limited scope control, limited predicate size, and repetitive execution.

The limitations of VPD are addressed in [2] which specifies the access control using authorization views. An authorization view is a traditional relational view or parameterized view. A parameterized authorization view is an SQL view which makes use of parameters such as user-id, user-location etc. they proposed two models: Truman model and non-Truman model. Truman model provides each user with a personal and restricted view of complete database. Queries submitted by users will be modified transparently so that the user does not see anything more than his view and the returned answer is correct with respect to the view. A parameterized authorization view is defined for each relation of the database by the DBA. This view defines that all users can have access from this database relation. Transparent modification of user query is carried out by substituting each relation by corresponding parameterized view.

Limitations of virtual private database and Truman model motivated the authors to develop non Truman model. In this model a query is undergone into a validity test, which if it fails the query is rejected and a notification is sent to the user. Otherwise the query is allowed to execute without modification. Under this model a user query is said to be valid if it can be answered only using the information contained in the authorization view. User can write queries against the database relations. The DBA can create several authorization views and any of those views can testify for the validity of the user's query. The non-Truman model guarantees correctness, but it requires powerful query inferencing mechanism. In general such inferencing mechanisms are not decidable and

the query accepted by one database can be rejected by another. Such a kind of unpredictability is undesirable for applications.

Further, the predicate based fine grained access control has been proposed in [3] by S. Chaudhuri et.al. The authors proposed a model for fine grained authorization which is based on adding predicates to authorization grants. This model supports predicated authorization to specific columns, cell level authorization with nullification, authorization for function/procedure execution and grants with grant option. The authorization model proposed by the authors extends the authorization models of the SQL: 2003 standard and add the several new components such as a user context, authorization predicates and query defined user groups.

Access control with privacy mechanisms have been studied by S. chaudhuri in [4]. They proposed a hybrid architecture which combines (a) a set of authorization predicates which restricts each user to only a subset of data (b) a set of noisy views that exhibits the deranged collective information over data which are not accessible through the authorization predicates. Noisy views are the abstraction to integrate privacy mechanisms in traditional database. They use the definition of differential privacy proposed in [5]. Privacy requirements in terms of k-anonymity has been defined in [6]. In [6] they analyzed the k-anonymity and explored its limitations latter they proposed a safe k-anonymity algorithm.

Workload aware anonymization was studied in [7]. They used a simple language for describing a family of target workloads and a suite of algorithms for incorporating these workloads into anonymization process. The authors discussed three techniques: classification and regression, selection, and aggregation and summary statistics.

Existing literature on workload aware anonymization focuses to minimize the overall imprecision for a given set of queries. However anonymization with imprecision constraints for individual queries has not been studied been studied before. The imprecision definition of [12] has been followed and introduced the constraint of imprecision bound for each query in a given query workload.

III. BACKGROUND

Privacy definitions are briefly explained in this section. For a given relation R= {a1, a2… an} where ai is an attribute, R* is the anonymized table of the relation R. Relation R is assumed to be a static table which includes the following types of attributes.

Identifier Attributes: These are the attributes which individually identifies the user. E.g. name, social security number etc.

Quasi Identifier (QI) Attributes: Attributes, e.g., gender, zip code, birth date, that can potentially identify an individual based on other information available to an adversary.

Sensitive Attributes**:** Attributes, e.g., disease or salary, that if associated to a unique individual will cause a privacy breach.

IV. PROPOSED WORK

This section is divided to three sub sections. Architecture Design, Access control mechanism: which includes brief explanation about various access control mechanisms, the mechanism which has been adopted in our work and a key generation algorithm has been proposed. The last section is privacy protection: this includes brief about how data will be protected from leakage and also a matrix encryption algorithm has been proposed.

4.1 Architecture Design

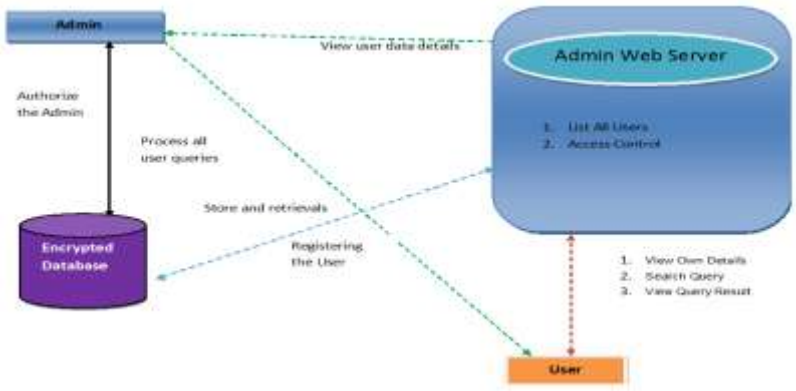

Fig 1. Architecture Design

Above fig 1 shows proposed architecture where each user registers with the system then searches for query by providing bound (attribute based on which query needs to be searched). The administrator receives the query and checks for the bound. If the bound is not an identifier attribute then he provides access control for that attribute. Otherwise the query will be denied and the user is intimated about the denial. Privacy is maintained by encrypting the entire database.

4.2 Access Control Mechanism

Access control is the process of mediating every request to resources and data maintained by a system and determining whether the request should be granted or denied. An imperative necessity of any data administration framework is to secure information and assets against unapproved revelation (secrecy) and unapproved or ill-advised alterations (integrity), while in the meantime guaranteeing their accessibility to authentic clients (no denials-of-service). Authorizing security accordingly obliges that each access to a framework and its assets be controlled and that all and just approved gets to can occur.

The various types of access control mechanisms [8] are discussed below:

4.3 Discretionary Access Control

This is the traditional access control in which user has the complete control over all the programs. DAC is based on giving access to the user on the basis of user identity and authorization which is defined for open policies. DAC owns and executes and also it determines permissions to the particular user to the object. DAC policies considers the access of users to the object which is based on the user's identity and authorization that specifies for each user's access method and object that is requested by user. Each individual request to access an object that has been checked. In DAC access method flexibility will be good. In this method most of the authorization is specified explicitly and also authorizations of individual user is closed. And also when authorizations are open then it is said to be open policies. DAC consists of access rules and access attributes .The access attributes allows the system to define several distinct level of authorization, and the access rules provide the mechanism for the cloud to prevent unauthorized access of sensitive information. DAC provides controlled sharing of objects among various subjects. DAC is said to be the mechanism of "who can access what". In DAC the owner of an object can choose to grant access permissions to other users. Access control list is associated with each object's file system. A simple form of Discretionary access control can be file passwords and giving access to

the authorized users. DAC mainly deals with the following that are Inheritance of permissions, User-Based Authorizations, Auditing of System Events, and Administrative Privileges.

4.4 Mandatory Access Control

Mandatory access control is based on the access of objects to number of subjects. Mandatory access control is mainly based on the security level. In this individual cannot change the access. Traditional MAC mechanism is mainly coupled with some security consideration. This follows the following two principles. Those are, read down (users current security level must dominate the access of the object being read) and write up (users current security level must dominate the access of the object being write) MAC based on the classification of objects and subjects present in the cloud environment. Access to a particular object is allowed only if some relationship is satisfied. Each object and subject present in cloud environment assigned some security level. This security level helps to identify the current access state of the object. Security level associated with user also called clearance. MAC used to protect network and file system, block users from accessing without appropriate authorization. In MAC the users will not be permitted to change the access control and its security level. MAC label is said to be security attribute which may be applied to subjects and objects throughout the system.

4.5 Role-Based Access Control

In role based access control access decisions are based on the individual's roles and responsibilities within the cloud environment. It formulates the user's access to the system based on the activities that the user has been executed in the cloud. It requires the identification of roles of users on the system. Role can be set of objects or actions associated with the subject. Role may vary depends on the user's priority. RBAC provides the web based application security. Roles are assigned based on the particular cloud organizational structure with their security policies. Each role in the organization's profile includes all authorized users, commands, transaction and allowable information access. Roles can be assigned based on the least privilege. These identified roles can be transferred and used based on the appropriate procedures and security policies. Roles can be managed centrally. RBAC implemented in three ways based on the design constraints that are, RABC0, RBAC1, RBAC2, RBAC3. RBAC0 is based on the least privileges and separation of roles. It does not contain hierarchy and permissions to the particular object is assigned directly. RBAC1 is based on the use of hierarchies and RBAC2 is based on the hierarchy within the RBAC1. RBAC3 is based on the both constraints and hierarchy. RBAC allows users to execute multiple roles at the same time and roles are the useful approach to organizations such as cloud, grid and peer to peer environment. In some cases the only one role can be assigned to one user and it recognize the same roles to other users jointly. After the DAC and MAC Mechanism RBAC has been proven as the efficient access control mechanisms. So securing information on the cloud is similar to securing data on the web. RBAC on the web is user pull architecture. RBAC can be used to provide service and assigns roles to each user's based on the user identity and its role based on the execution environment in cloud. RBAC on the web is implemented with server pull architecture. RBAC permissions are associated with roles and users are assigned to appropriate roles. System administrators only can be able to create roles and granting permissions to those roles. Without RBAC it is difficult to determine what permission has been assigned to which user.

4.7 ABAC (Attribute Based Access Control)

ABAC is attribute based access control normally considers identification, authentication, authorization and accountability. In attribute based access control the attributes are considered based on the user's request and the

type of access user wish to access and the needed resources of user. ABAC is more secure and flexible and scalable and it provides hierarchical structure. Set of user attributes will be maintained individually.

4.7.1Methodology Used

In our work ABAC Access control mechanism has been used wherein user requests admin to access the attribute of his choice. Administrator will be having the right to decline the request. He will do so if the requested attribute is identifier attribute. Here access is provided by a key through which user can access the attribute. One time key will be provided if the user wishes to access a particular attribute more than once. Key generation algorithm [9] has been used which is as below:

```
Function token generation
Output: key
Begin
  Arr= generate ();
  Print (arr);
 End function
Function generate
Begin
  For i= 0 to 4 step by 1
    C= random character;
     D= random number;
     Key= concatenate (C, D);
Return st;
End function
```

Once the user is provided with the key he can search for a query by providing the bounds and the disease which he want to search. Bounds are the limit for which the result is given. In base paper administrator will be providing bounds which is not convenient for the user. In our work user is allowed to provide the bounds. After getting the bounds and the disease suffix array algorithm has been used for searching [10] which is given below:

```
Function suffixArray
Input: string to be found
Begin
 Initialize suffixarray;
 For i=0 to s.length () step by 1
 Find index;
 Find substring using index
 Assert s.substring (index).equals (suffix.select (i));
 Find rank;
 If (i==0)
   Print (I, index, rank, Substring);
Else
  Find lcp;
  Print (I, index, lcp, rank substring);
End function
```

```
Function rank
Input: string
Begin
  Lo= 0, hi= suffixarray.length-1;
  While (lo < = hi)
     Mid= lo+ (hi-lo)/2;
     Cmp= compare (query, suffixarray [mid]);
     If (Cmp<0) hi= mid-1;
     Else if (Cmp >0) lo= mid+1;
     Else return mid;
End function
Function compare
Input: query, suffix
N= Math.min (query, length);
For i= 0 to N step by 1;
   If (query. (CharAt (i) < suffix.charAt (i)) return -1;
   If (query. (CharAt (i) > suffix.charAt (i)) return +1;
Return query.length () – suffix.length ();
End function
Function lcp
Input: i
Begin
  If (i< 1|| i>= suffixarray.length ())
   Return lcp (suffixarray [i], suffixarray [i-1]);
End function
Function lcp
Input: suffixs, suffixt
N= Math.min (s.length (), t.length ());
For i=0 to N step by 1
  If (s.charAt (i)! = t.charAt (i))
 Return i;
Return N;
End function;
```

Suffix arrays are essentially a representation of the lexicographic order of all suffixes of a string. They can be constructed very efficiently in terms of time and space. The use of suffix arrays makes string mining scalable to large databases.

Input to the algorithm is the string to be matched. It first initializes the suffixarray with the string and finds the substring. Then finds the index computes rank and lcp and then prints the rank, lcp and substrings.

4.8 Privacy Protection

Information becomes sensitive when they are specific to a small number of individuals. Data mining, on the other hand, typically makes use of information shared by some minimum number of individuals to ensure a required statistical significance of patterns. As such, sensitive information are to be protected from any leakage. In our work the sensitive information will be encrypted and stored in the database so that even if the database is hacked no one can decrypt the data.

Matrix encryption and decryption algorithm [11] is used for encryption.

4.8.1 Matrix Encryption and Decryption Algorithm

```
Function encrypt
Input: msg, key
Output: cipher text
Begin
   len= key.length ();
   x= 80/len;
  Ckt= 0;
  For j=0 to x step by1
      For i=0 to len step by 1
       Matrix[i] [j] = (char) (46+ckt);
       Ckt++;
  For i= 0 to msg.length () step by 1
     For k= 0 to x step by 1
        For j= 0 to len step by 1
           If (msg.charAt (i) == matrix[j] [k])
              Out+= key.charAt (i) +k+j;
   Return out;
End function
```

Input to the algorithm is the message to be encrypted and globally accepted key. The algorithm first constructs the matrix of numbers, alphabets and symbols (starting from ASCII character 46). It then matches each character of message against the characters in the matrix and the corresponding row and column numbers are concatenated with the corresponding character of the key. This forms the cipher text.

Below is the algorithm used for decryption. Input to the algorithm is the cipher text to be decrypted and the same key which was used for encryption. First it will construct the matrix in the same way as in encryption algorithm. It then extracts the row and column number from the cipher text which is used to search the corresponding character from the matrix. The result will be concatenated with previously computed result and finally the original message will be constructed.

```
Function decrypt
Input: cipher text, key
Output: msg
Begin
   Len= key.length ();
   x= 80/len;
```

```
Ckt= 0;
For j=0 to x step by1
    For i=0 to len step by 1
     Matrix[i] [j] = (char) (46+ckt);
     Ckt++;
 a=1;
 b=3;
 loop_len= cipher.length/3;
 While (count<loop_len)
    Asd [count] = cipher.substring (a, b);
    t1= Integer.parseInt (asd [count].charAt (0));
    t0= Integer.parseInt (asd [count].charAt (1));
    Out+=matrix [t0] [t1];
    a= a+3;
    b= b+3;
    Count++;
  Return out;
End function
```

V. CONCLUSION

An access control mechanism with privacy protection framework has been proposed. Access control mechanism allows only authorized queries. In privacy protection the data is encrypted and stored in the database. In this framework user is allowed to provide the bounds for searching the query which is efficient.

REFERENCES

[1] K. Browder and M. Davidson, "The Virtual Private Database in oracle9ir2," Oracle Technical White Paper, vol. 500, 2002.

[2] S. Rizvi, A. Mendelzon, S. Sudarshan, and P. Roy, "Extending Query Rewriting Techniques for Fine-Grained Access Control,"Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 551-562, 2004.

[3] S. Chaudhuri, T. Dutta, and S. Sudarshan, "Fine Grained Authorization through Predicated Grants," Proc. IEEE 23rd Int'l Conf. Data Eng., pp. 1174-1183, 2007.

[4] S. Chaudhuri, R. Kaushik, and R. Ramamurthy, "Database Access Control & Privacy: Is There a Common Ground?" Proc. Fifth Biennial Conf. Innovative Data Systems Research (CIDR), pp. 96-103, 2011.

[5] S. Chaudhuri, R. Kaushik, and R. Ramamurthy, "Database Access Control & Privacy: Is There a Common Ground?" Proc. Fifth Biennial Conf. Innovative Data Systems Research (CIDR), pp. 96-103, 2011.

[6] N. Li, W. Qardaji, and D. Su, "Provably Private Data Anonymization: Or, k-Anonymity Meets Differential Privacy," Arxiv preprint arXiv:1101.2604, 2011.

[7] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Workload-Aware Anonymization Techniques for Large-Scale Datasets," ACM Trans. Database Systems, vol. 33, no. 3, pp. 1-47, 2008.

[8] Punithasurya K, Jeba Priya S, "Analysis of Different Access Control Mechanism in Cloud", International Journal of Applied Information Systems, Vol 4- No 2, September 2012.

[9] N Ajlouni, A El-Sheikh and A Abdali, "A New Approach in Key Generation and Expansion in Rijndael Algorithm", The International Arab Journal of Information Technology, Vol 3, No 1, January 2006.

[10] M. I. Abouelhoda, E. Ohlebusch, and S. Kurtz. "Optimal exact string matching based on suffix arrays", In Proc. 9th Symposium on String Processing and Information Retrieval, volume 2476 of LNCS, pages 31-43. Springer, 2002.

[11] B Acharya, G Rath and Sarat K P, "Novel Modified Hill Cipher Algorithm", Proceedings of ICETAETS 2008 pages 126-130.

[12] Zahid P, Walid G. Aref, Senior Member, IEEE, Arif G, Fellow, IEEE, and Nagabhushana Prabhu, "Accuracy-Constrained Privacy Preserving Access Control Mechanism for Relational Data", IEEE transactions on Knowledge and data Engineering, VOL. 26, No. 4, April 2014.