Data Attribute Security for Distributed Database

Sapana Patil

Abstract— Privacy-preserving publishing of micro data plays important in recent year because it is important to secure the sensitive information from data. We have used several micro data anonymization techniques such as Generalization, Bucketization and Slicing. A Slicing is a technique which divides data into two partitions such as horizontal and vertical. Slicing provides better data effectiveness than generalization and it also protect from membership disclosure. Data Provider aware algorithm is used for anonymization of the dataset which result into the reduction of the waiting time. Problem of attacks are removed with SQL Injection Prevention Technique. Experiments confirms that our approach achieve better utility and efficiency satisfying the security of the database.

Index Terms— Bucketization, Data Anonymization, Generalization, Slicing, SQL Injection Prevention

1 INTRODUCTION

IN recent years, separation takes a vital role for securing the data from possible attackers. Data need to be shared for public advantage as there is requirement for Health Care and researches, privacy of individual is major concerned with regard to sensitive information. So publishing of data should be done in such a way that privacy of data is maintained. Two types of problem occur while publishing aggregated data. First is insider attack and second is outsider attack. Outsider attack is caused by people who are not data provider and insider attack is by data providers. The paper focuses on distributed data security with various approaches. Distributed database increasingly share data that contain personal information. Previously the researcher avoided the problem of attacks with the help of Secure Multiparty Computation protocol. In this paper we are using the SQL Injection Prevention Technique to overcome problem of attack. For example, in the healthcare field, a national program to establish a network which share Health information nationally between hospitals, provider etc. Privacy preserving data publishing have received importance in current years due to its approaches for sharing data with preserving individual privacy. Purpose is to publish an anonymized view of aggregated data T which will be protected from attack as shown in (Fig. 1). Attacks are run by attackers; it can be a single or a group of external or internal entities that want to break privacy of data using background knowledge [2].

In Fig. 1, T_1, T_2,T_3 , T_4, T_5 are databases, the data for them is provided by provider. For example P_1 provides data for databaseT_1. The distributed data coming from different providers get aggregated first and then anonymize using anonymization technique. P_0 is the authenticate user and P_1 trying to breach privacy of data which is provided by other users with the help of BK (Background knowledge). This type of attack it can call as insider attack. Our system protect database from such attacks. Section 2 describes the related work, Section 3 gives the Basic Preliminaries required, Section 4 describes the implementation Details, Section 5 explains Results, and Section 6 describes the Conclusion.

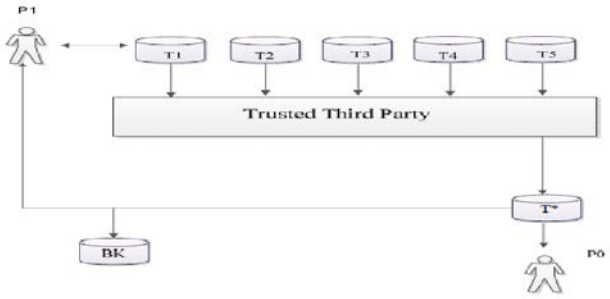

Fig 1. Collaborative Data Publishing

1.1. Motivation: As there is increasing need of sharing personal information from distributed database, the special care should be taken to protect it from attacker. Attacker can be single entity or group of entities. Attacker can breach privacy with the use of background knowledge. Multiple providers wish to compute an anonymized view of their data without disclosing any private and sensitive information. A data recipient for example an attacker, e.g., P0 , is trying to find extra information about records with the help of available data, T* , and background knowledge, BK. For example, k-anonymity protects against identity disclosure attacks by requiring each quasi identifier equivalence group (QI group) to contain at least k records.L-Diversity requires each QI group to contain at least l well represented sensitive values. Our system provides the double security. Double security because it uses the SQL injection technique to prevent malicious attack and uses the slicing technique to make database more secures. As it is very important to secure the data from attack of malicious user we are using SQL Injection Prevention technique which in turns uses Aho-CoroSick algorithm to avoid attacks. Whenever any malicious user trying to access the data immediately the notification is given showing malicious user is trying to access data.

2 LITERATURE SURVEY

This section focuses on the different approaches of Privacy Preserving Data Publishing. It converses advantages and limi-

tation of these approaches. Privacy preserving data analysis and collaborative data publishing has received considerable attention in current years as capable approaches for sharing data with preserving individual privacy.

Goryczka, B. C. M. Fung [1] in their research used the Secure Multiparty Computation protocol and Trusted Third Party Protocol to avoid the attack of the insider and outsider. The limitation of the SMC protocol is that their security reduces with the primitives used

C. Dwork in his survey result [2] of differential privacy evaluates and summarizes different approaches to privacy preserving data publishing (PPDP), study of different challenges in practically publishing of data, clarify the other related problems which are different from PPDP and requirements that make PPDP different from others and proposed future research directions. They identify the research direction in PPDP like privacy preserving tools for individuals, privacy protection in emerging technology and incorporation of privacy protection and engineering process.

N. Mohammed, B. C. M. Fung, and C. Lee, developed LKC privacy model [3] for healthcare system with high dimensional relational data. This LKC model provides good result as compare to traditional k-anonymization model. LKC model is general privacy model which stops the attribute linkage and record linkage to anonymize data.

Two party protocol DPP2GA [4] is presented by authors W.Jiang and C. Clifton. This protocol is developed for preserving privacy because only k-anonymity is not adequate. One of disadvantage of this protocol is it cannot produce a precise data when data is divided. It is only privacy preserving protocol because it introduces certain inference problem.

A. Machanavajjhala, J. Gehrke, developed a system [6] with l-diversity. This system developed satisfying the kanonymity which can be protected from attack. Attacker can cause attack on protected system using BK (background knowledge). L diversity prevents the problem of attack.

Yufei Tao has proposed ANGEL [7], a new anonymization technique that is that is effective in generalization. ANGEL (Anatomy and Generalization on Multiple Sensitive) is applicable to any monotonic principles. Yufei shows that ANGEL provides elegantly to hard problem of contiguous publication. Issue of generalization is that it is restricted with margin. So, new method ANGELM used to distribute at any margin with strong privacy guarantees.

Alberto Trombetta in his research work [8] develop a system without informing Yogesh and Mangesh, the contents of tuple and database, tuple inserted is patterned for K-anonymity. Alberto has proposed two protocols on the basis of suppression and generalization. Alberto focused on data anonymization techniques to address privacy.

Xiaolin Zhang in his research work [9] created a new generalization principle which limits the risk of Multiple Sensitive Attributes Privacy disclosure in re-publication. Respective algorithm has higher degree of privacy protection and lower hiding rate.

Tiacheng Li in his research [10] introduced a new technique for privacy preservation known as Slicing. Slicing overcome all drawbacks of generalization and bucketization. Slicing is the most suitable method for high dimensional data.

S.Kiruthika developed suppression slicing [11] by suppressing any one of attribute value in the tuples and then performs the slicing. This utility is maintained with the suppression on few value and random permutation. This paper uses slicing, data publication, bucketization and generalization in the database.

Dr. Amutha Prabhakar has developed pattern matching algorithm [12] to prevent the SQL Injection Attack. He has proposed the static and dynamic pattern matching algorithm. As it is very important to avoid the malicious attack, since it can cause major changes in the system. To avoid those changes author has proposed the SQL Injection Prevention technique approach.

3 DIFFERENT METHOD OF PRIVACY PRESERVATION

There are many different privacy preserving techniques for preserving privacy of the database as follows.

3.1. Generalization

In this process, generalizations of attributes are done separately. Disadvantage of generalization is that correlation between attribute is lost.

3.2. Supression

In Suppression technique, the attribute values are replaced by * to preserve privacy of data.

3.3. Bucketization

Bucketization is the process of separating Sensitive attribute from Quasi Identifier with the help of random permutation. Collection of data forms the anonymized data. Drawback of bucketization is that it does not prevent membership disclosure.

3.4. Slicing

It is the process of horizontal and vertical partitioning of data. Vertical partitioning is done on the basis of correlation of attributes. Horizontal partitioning is done by grouping tuples to form bucket.

4 SYSTEM ANALYSIS

4.1. Problem Definition-

To secure the sensitive information of the record because it

consists of personal information is the need of today. A lot of hurdle can occur while publishing data. Major problem is the attacks. Attacks are of two types, insider attack and outsider attack. In order to secure the data, system should be immune to attack. Main goal is to distribute an anonymized view of combined data, P^* which is protected from the attacks. Our system increases the security and privacy of system with the help of slicing methods. SQL Injection prevention technique prevent malicious user from accessing database. System will also track the malicious user when it is trying to access data.

4.2. System Structure

The provider provides the data to the database. Whenever any user fire some query first it will check whether it is authorize user or not if he or she is authorize user then it will check the user query with the SQL query pattern written in the SQL pattern database. If the anomaly scores of the query greater than the threshold then that particular query pattern gets added in the SQL pattern database. Means one SQL pattern gets added into the pattern list. If anomaly score is not greater than threshold then query undergoes execution. If the person is not authorizing user then it undergoes the slicing. It selects the point for selection to slicing and after that it performs the partitions and permutation. Next step is to check C constraint (K-Anonymity and l-diversity property) if it is satisfied then slicing is performed. If C constraint is not verified then it is send in bucket. In Bucket a secure view of database is present. After the processing the anonymized view of the data T* is obtained.

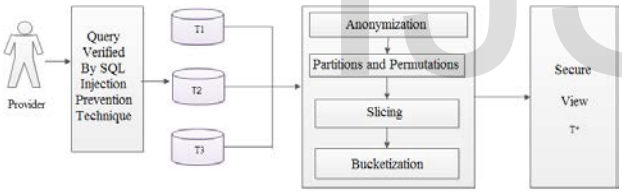

Fig. 2. System Architecture

4.3. Implementation Methodology

4.3.1. Anonymization By Slicing:

Slicing includes Attribute and tuple partitioning. In Attribute partitioning (vertical partition) we divides data into column depending on the column correlations age-zip, name and Diseases and tuple partitioning divides into the tuples or in to the buckets (horizontal partition) as t_1 , t_2, t_3, t_4, t_5, t_6. In attribute partitioning as age and zip are highly correlated they are separated together. They are known as Quasi Identifiers (QI). On the other hand tuple partitioning verifies the l-diversity of sensitive attribute (SA). Steps for implementation are as follows.

Algorithms are as follows.

Initialize bucket b=m, int j= rowcount, column count=C, Q={T}, // T= data into database, Arraylist= a[j];
While Q is not empty
If j<=m
Check L diversity;
Else
j++;
Return T*;
Q=Q-{T*+a[j]};
Repeat step 2 and 3 with next tuple in Q
T*=T*U A[T] // next anonymized view of data D

First initialize b = limit of data anonymization bucket size, number of rows, number of columns, arraylist and database in the queue (step 1). Process will be proceeds if and only if queue is not empty i.e. data should be present in the data in database. Verify data for L diversity if row count = b = m. Initially Q= Queue of data. Bucket data satisfies the conditions of the l-diversity and k-anonymity then return T*(Step 2). The data which cannot satisfies the conditions of the l-diversity and k-anonymity it is stored in a[j] arraylist. After this data in database Q=Q-{T*+a[j]} (step 3). Repeat step 2 and step 3. A [T] is anonymized data in database. Apply above all steps to remaining data to form an anonymized view T^* which is combination of original view and new one i.e. $T^* = T^*$ U A [T].

4.3.2. K-Anonymity: Record satisfies K-Anonymity property if information for each record present in the release cannot be distinguishable from at least k-1 records whose information also occurs in the release.

4.3.3. L diversity: L diversity is the phenomenon to maintain uniqueness in data. This system uses Sensitive Attribute concept on disease. Our secure bucket size is 6 and we maintain L=4 i.e. from 6 disease record 4 must be unique.

Coding for l-diversity as follows.

Initialize L=n, int i ;
If j= t-n+1;
Then a[0]…..a[1], insert these values as they are in Q;
j++;
Else
Check privacy constraint for every incremented value in Q
If
L=t then
Fscore=1
Insert value in the row
j++;
else
Add element to arraylist a[j];
4. Exit

First initialize L=n and rowcount j. If j=t-n+1 i.e. if b=t=6 and L=m=4 then j=3, up to third row Fscore is not checked because it is not needed. This data is added to the Q (Step 1 and Step 2). After that privacy constraint of data from Q is checked. If the l-diversity is satisfied then FScore=1, if not then that element is added in arraylist a [j] (step 3).

4.3.4. Permutation: Reordering of records of data is known as Permutation. Our proposed system we have used permutation on Quasi identifier (QI) attribute.

4.3.5. Fscore: The level of satisfaction of privacy constraint C is known as Fscore. If Fscore=1 then C (D*) = true.

4.3.6. Constraint C: D* which fulfill slicing conditions with l-diversity is known as Privacy Constraint C. Consider value of L diversity is 4. Fscore should be 1 when system fulfills L diversity condition.

4.4. Mathematical Model

Let, S={s, e, I, O, A}

Where S is a system of secure collaborative data publishing consist of database with some attributes related to patient data for hospital management system. S consist of

s = distinct start of collaborative system

e = distinct end of collaborative system

I = Input from users

O = output of model

A = algorithms or functions

Let,

s = {R_e} // Request from users

= {R_{ed}, R_{ea}} // R_{ed}=request from doctor

//R_{ea}=request from admin

X= {D_{D1},D_{D2}…….D_{Dn}}

// database i.e. data provided by providers

A = {Slicing Algorithm, L-Diversity, Provider Aware Algorithm }

Y = {D_1^*, D_2^*, T}

D_1^*= {R_{ed} ^ D_{Dn}}

// Data obtained by doctor request and database provided.

// Privacy and Security is provided by Slicing and L diversity.

D_2^*, = {R_{ed} ^ D_{Dn}}

// Using Provider Aware algorithm on database on user request

T1= {R_{ea} ^ D_{Dn}} // Original data

e = output table format

Success conditions are,

R_e ≠ NULL, D_{Dn}≠ NULL

Failure conditions are,

R_e= = NULL, D_{Dn} ≠ NULL

4.5 Algorithms

Aho-CoroSick pattern matching algorithm is used

4.5.1. Pattern Matching Algorithm

1: Procedure SPMA (Query, SPL[ ])
INPUT: Query=User Generated Query
SPL[ ]=Static Pattern List with m AnomalyPattern
2: For j = 1 to m do
3: If (AC (Query, String.Length(Query), SPL[j][0]) = =0)then
4: Calc anomaly score
5: If ( ) Score Value Anomaly = Threshold
6: then
7: Return Alarm ... Administrator
8: Else
9: Return Query... Accepted
10: End If
11: Else
12: Return Query… Rejected
13: End If
14: End For
End Procedure

4.5.2 Aho–Corasick multiple key word matching algorithm

1: Procedure AC(y, n, q0)
INPUT:
y= array of m bytes representing the text input (SQL Query Statement)
n= integer representing the text length (SQL Query Length)
q0=initial state (first character in pattern)
2: State: q0
3: For i = 1 to n do
4: While g (State, y[i] = = fail) do
5: State ← f (State)
6: End While
7: State ← g(State,.y[i])
8: If o (State) == NULL then
9: Output i
10: Else
11: Output
12: End If
13: End for
14: End Procedure

Pattern matching is a method used for identification and detection of anomaly pattern. In the Static Pattern List contain a list of known Anomaly Pattern. In Static Phase, verify the user generated SQL Queries with Static Pattern Matching Algorithm. In Dynamic Phase, when new anomaly is occur then Alarm will indicate and new Anomaly Pattern will be generated. The new anomaly pattern will be updated to the Static Pattern List.

Static pattern list consist of certain number of fixed pattern .Threshold is set initially. Whenever any user entered a query then verified with each pattern present in the pattern list .While comparing the query with the each pattern in pattern list the anomaly score is find out. If anomaly score of the query exactly match with the threshold then new anomaly pattern is created and it is stored in pattern list. Procedure AC does the byte by byte input of string.

5 EXPERIMENTAL EVALUATION

5.1. Experiment Setting: My experiment runs on Intel Processor dual core with 4.00 GB RAM with 32 bit operating system. Our project requires at least 2 computer systems for the execution of the proposed system. Microsoft Visual Studio 2012 is the front end of the project and MySQL Server 5.1 is the backend of the project. Here we have made a distributed database system for the execution of the system. Means with the help of this system the data attribute security is provided to the distributed databases. Distributed database is created with the help of LAN connection with the help of connecting cables and a switch. Systems are connected in LAN with the help of the connecting cables and IP addresses are set among them.

5.2 Results

In the proposed system we are using the Slicing approach for the security of the data. The existing system is using the cryptographic technique in which the encryption and decryption takes places. But it increases the time complexity of the system. Following table shows the time comparison of the existing system and proposed system.

Table 9.1: Time comparison between Existing System and Proposed System

Packet Size	AES(Existing) (Sec)	DES(Existing) (Sec)	RSA(Existing) (Sec)	Slicing(Proposed) (Sec)
200 KB	34.03	48.87	173.23	13.23
400 KB	71.67	93.09	338.09	27.34
600 KB	100.4	144.34	519.78	40.90
800 KB	135.4	183.32	677.56	52.23
1000 KB	163.23	224.15	809.67	63.78

6 CONCLUSION

Database security plays very important role in recent year. Experiment results shows that slicing technique can make database more secure. With the help of the Aho-Corosick algorithm the attacks of the malicious user can be restricted. Main aim of the distributed system is to reduce the overhead on the single system. Experimental graph shows that system using slicing technique along with SQL Injection Prevention technique increases the efficiency and utility of the model providing the security to the database.

7 FUTURE ENHANCEMENT

In future, this system can be considered for data which are distributed in ad hoc grid computing. The research work can be follow in the direction that proposed system will be implemented on wireless network with a large scope.

ACKNOWLEDGMENT

I would like to thanks my guide Dr.A.N.Banubakode and PGCo-ordinator Dr.P.K.Deshmukh for their guidance throughout this project, without them research would not have been possible.

REFERENCES

[1] Goryczka, B. C. M. Fung and L. Xiong,, "m-Privacy for collaborative data publishing," in Proc. of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2011.

[2] C. Dwork, "Differential privacy: a survey of results," in Proc. of the 5th International Conference on Theory and Applications of Models of Computation, 2008, pp. 1–19.

[3] N. Mohammed, B. C. M. Fung, C. Lee and P. C. K. Hung, "Centralized and distributed anonymization for high-dimensional healthcare data," ACM Transaction on Knowedge Discovery from Data, vol. 4, no. 4, pp. 18:1–18:33, October 2010.

[4] W. Jiang and C. Clifton, "Privacy-preserving distributed k-anonymity," in DBSec, vol. 3654, 2005, pp. 924–924.

[5] W. Jiang and C. Clifton, "A secure distributed framework for achieving k-anonymity," VLDB journal, vol. 15, no. 4, pp. 316–333, 2006.

[6] Machanavajjhala, M. Venkitasubramaniam, J. Gehrke, and D. Kifer, "l-Diversity: Privacy beyond k- anonymity," in ICDE, 2006, p. 24.

[7] R. Sheikh, D. K. Mishra and B. Kumar, "A distributed k-secure sum protocol for secure multi-party computations," Journal of Computing, volume. 2, pp. 68–72, March 2010.

[8] C. Dwork, "A firm foundation for private data analysis," Commununication ACM, vol. 54, pp. 86–95, January 2011.

[9] P. Jurczyk and L. Xiong, "Distributed anonymization: Achieving privacy for both data subjects and data providers," in DBSec, 2009, pp. 191–207.

[10] C. M. Fung , P. S. Yu, K. Wang, and R. Chen, "Privacy-preserving data publishing: A survey of recent developments" , ACM Computuing Survey., vol. 42, pp. 14:1–14:53, June 2010.

[11] T.S.Gal,Z.Chen and A. Gangopadhyay," a Privacy protection model for patient data with multiple sensitive attribute IJISP, Vol.2 no.3, PP.28-44, 2008

[12] S.Kiruthika, Dr.M.Mohamed Raseen "Enhanced Slicing Models For Preserving Privacy In Data Publication" in International Conference on Current Trends in Engineering and Technology, ICCTET'13

[13] DrM.Amuthaprabhakar,M.Kathikeyan,Prof.K.Marimuthu"An Efficient Technique for preventing the SQL Injection Attack using Pattern Matching Algorithm"IEEE International conference 2013(ICECCN)