Vertically Partitioning of Database for Secured Data Release

Kanchan Kauthale
Department of Computer Science
Dr. D Y Patil SOE
Pune, India

Sunil. D. Rathod
Department of Computer Science
Dr. D Y Patil SOE
Pune, India

ABSTRACT
There is a huge database which contains some private and public information. When we are mining the useful information from the enterprise data, there can be issues regarding disclosure of the private data. Many times, the sensitive data can be directly or indirectly derived from the answered queries, to overcome these issues we extend the differential privacy model. In this privacy model, source database table is divided into some parties which hold different attributes for the same set of individuals. We have addressed the problem of private data exposure, which can be prevented by forming vertically partitioned databases. This partitioning is by an exponential mechanism algorithm which guarantees that the other party can't derive extra information from the answered query.

The proposed algorithm also provides the security for the data which is release from the scatter pattern. To improve query response time of the system some schemes are used like Vertical Partitioning Scheme (VPS), Statistics Collector, and Partitioning Generator.

General Terms
Vertical Partitioning Algorithm; Differential Privacy.

Keywords
Secure Data Integration; VPS; Statistics Collector; Partitioning Generator**.**

1. INTRODUCTION
As there are so many organizations, who stored their huge data in the database, that data may be the private data or the public data. Giving security to this data along with satisfaction of the customer's requirements are the important aspects in day to day life. Customer's data contain some private information and some of the public information. Public information can be accessed by anyone, but the data which is private cannot be accessed by an unauthorized person.

So to provide the security for the sensitive data and also to provide better services to the customers, there are different organizations that satisfy the customer's requirement. In our scenario we are taking bank as one organization which wants to provide different services to the customers or the normal user. For this service bank approaches to those organizations which work for the bank products. In this scenario, the bank system provides the customer's data to the organizations like Credit Card Company or loan companies. These organizations then directly interact with the corresponding customers and offer their services to customers for their betterment. The bank provide a partial set of information to these companies, i.e. only the information required by the companies are forwarded to them and all the private data of the customers are kept hidden. For example, exposer of details like name of the customer, contact details etc. won't harm, hence these details can be forwarded to the companies and details like transaction details and other sensitive data are made hidden from the companies. By this, banks will provide only the data that is required by the companies, the companies won't be able to derive any extra or sensitive data of the customer.

We generalize the problem, for example suppose a bank X hires a Loan company Y. Both of them have different sets of attributes for the same set of individual which are identified by the customer's ID, like bank X owns database K having fields like ID, Job, Account_Balance where company Y owns database J having fields like ID, gender, income. For better decision about the credit card or the loan sanctions, both the entities have to merge their data. Suppose there is one more company Z like credit card which also needs the information from both X and Y, which will be a combination of databases of X and Y. Hence, Z will be accessing the entire database and can access the sensitive data of the customer.

For example, when Party X and Party Y joined their database K and J, then Z can easily discover the sensitive data of the other entity as Z is accessing the whole database which is formed by combining the databases of X and Y i.e. databases K and J. This way sensitive data can be exposed through linking attack and can be misused. X and Y do not contain any sensitive data of the customer but still this integration of the databases can enhance the chances of recognizing the customer's profile. This will be a threat to the customer's private data

Hence to avoid such linking attacks we vertically partitioned the database into two parts, where the first part contain the public information and the second part contain the private information. This vertical partitioned technique will increase the security for the private data. The main goal is to provide the security for the sensitive data which is achieve by using the exponential mechanism and also improve the query response time of the system.

In this paper we are using exponential mechanism for the partitioning purpose which ensures that no extra information is access by the other party. This algorithm also satisfies the differential privacy model. To improve the query response time of the system we are using Vertical Partitioning Scheme (VPS), Statistic Collector, and Partitioning Generator.

2. RELATED WORK

To provide the security for the sensitive data in the database, a lot of research work is being done in the past, on the respective field.

B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, survey on Privacy Preserving Data Publishing (PPDP) [3]. In this, PPDP provides methods and tools for publishing useful information while preserving data privacy. In existing system there are different scenarios used like, Interactive mechanism in which data requester fetches the queries through some private mechanism and administrator replies the answer queries in response. Limitation of this is that it can only answers the liner number of queries. Otherwise the attacker will try to construct the original data.

In case of non-interactive mechanism the owner of the data first analyzes the data and then publishes. Once the data is published the data can't be change by the owner. This mechanism is known PPDP [3]. Some limitations of the non-interactive mechanism is that sometime this mechanism does not answers the queries appropriately because the data receiver can't construct a query for data mining in short period of time. For this an attacker can develop unlimited queries so this model cannot achieve privacy as that of interactive.

In case of distributed mechanism the data may be gain by one party or the many party but the owners of the data want to gain the same task as the unique one party without sharing their data with other parties. [3]

In existing system anonymization refers to the PPDP approach to hide the identity and/or sensitive data of the record owner. This operation hides the detailed information so that several records become indistinguishable with respect to the other. Some anonymization algorithms are proposed like Optimal Anonymization Algorithms: In this, algorithm finds some optimal anonymization for given data. But there is one limitation that this algorithm can't find the optimal solution for the huge data sets. To overcome this problem new algorithm is introduced [3]. Minimal Anonymization Algorithms: This algorithm finds the minimal solution but does not given an appropriate solution when more than 3 attributes are taken. But some of them fulfill the goal of the classification analysis. [3]

Differential privacy [1][2] model proposed the alternative for the partitioned-based privacy model for PPDP. Many of the research related to the differential privacy [2] focuses on the goal of reducing the added noise which is received during the data mining results on interactive setting [4].In case of the non-interactive mechanism it only holds the single-party mechanism. Therefore the proposed techniques do not satisfy the requirements of the privacy model. [5][6]. Privacy Preserving Distributed Data Mining (PPDDM) [7] is one of the proposed approach in which many data owners calculate their inputs without sharing data with other parties. The function used in this scenario is like a clustering, classification etc. Many organizations like bank want the customer's finance data for purpose of analysis. For this, different techniques have been proposed for data mining including association rule [8], classification [9], clustering [10]. But all these proposed techniques do not provide any privacy guarantees on calculated output. Dwork et al. and Narayan [11] [12] proposed the differentially private queries for horizontal and vertical partitioned data. . N. Mohammed, R. Chen, B.C.M. Fung, and P.S. Yu [14] the non-interactive approach is more flexible than the interactive approach because informer the data receiver can analyze and exploration their data. Clifton and Jiang [13] have suggested the techniques for distributed k-anonymity framework which securely divide the database into two parts and also fulfill the k-anonymization requirements.

3. PROPOSED SYSTEM

To overcome the problems of the existing systems we propose exponential mechanism which satisfies the definition of the differential privacy model. Also to improve the query response time of the system some algorithms are used like Vertical Partitioning Scheme, Statistics Collector and Partitioning Generator which improve the query response time of the system. The data can be dynamically access by the user whenever required.Fig.1 shows the system architecture of our proposed approach.

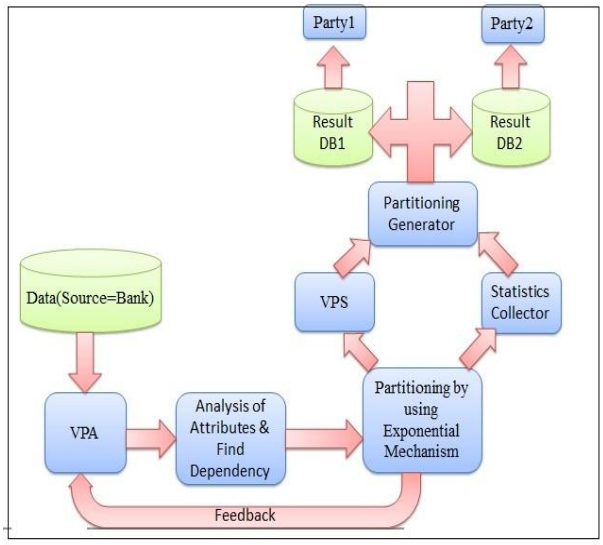

Fig 1. System Architecture

In this proposed architecture we are using bank database as a source database. This database is send for the partitioning to VPA (Vertical Partitioning Algorithm) in this VPA first analyzes the attributes which are important in the database and which we don't want to share with other entity. After that VPA, find the dependencies between the attributes. Then by using the exponential mechanism the partitioning process is completed. In this partitioning two parties are generated, the first party contains the entire regular information whereas the second party contains private information.

The information of that two party database is send to the statistics collector. This Statistics Collector contains all the

information about the query and the attributes. After the partitioning, the Statistics Collector generates table for all the entities like attribute Table, master query Table, attribute use table, attribute affinity table, cluster affinity table and also set some active rules for the table. Vertical Partitioning Scheme (VPS) is used to improve the query result. VPS is calculated based on the Statistics Collector and database. After that, data result is send to the Partitioning Generator. Partitioning Generator materializes the new VPS and deletes the old fragments index. Finally we get two partitioned database which contain the partitioned data and can be access by two different entities.

There are basically four modules:

3.1 OVPA (Optimal Vertical Partitioning Algorithm)

In this module we are analyzing the attributes which attributes are more important and which are less important. Also we are finding the dependencies between the attributes.

3.2 Exponential Mechanism

In this, we are trying to partitioning the database into two parties and check the partitioning done correctly or not. Here we using some privacy definition and some exponential equation for partitioning like $\sum \exp(\epsilon s_i/2\Delta s)$

Where s is the score means the access level of each entity.

The responsibility of exponential mechanism is to create virtual vertical partitioning, which will be generated by the reports of attributes and dependencies. Now it is little complicated to insert, delete data from views so to do that operation we are creating a mechanism to update or insert or delete data from view from actual database.

3.3 VPS (Vertical Partitioning Scheme)

VPS is used to improve the result of the query. VPS is calculated based on the Statistics Collector and database. VPS has some active rules, based on that rules actions are taken in database. Working of VPS

Step1: Generating Attribute Use Table
get AUT (MasterQT, AUT)
{generate the AUT form MasterQT}
Step2: Getting OVPS
getVPS (AUT, VPS)
End
Statistics Collector contains all the information about the query and the attributes. After the partitioning, the statistics collector generates table for all the entities like attribute Table, master query Table, attribute use table, attribute affinity table, cluster affinity table and also set some active rules for the table. For example when query Q is executed, statistics collector will get to know this query is already in the table, if the query is not present then statistics collector generate ID for that query and stores its description and time of execution in table. Working of Statistics Collector

Step1: Query Q fired
Step2: Check for Q exist or not
Step3: If Q exist then just its frequency increment
By 1
Else assign the id for query
And store the description and time of execution of the query in table

3.4 Partitioning Generator

This is used to materialize the new VPS, this generator delete's the old fragments index and generate new one. This module decides the actual partitioning based on the access level, which party should access which database.

4. MATHEMATICAL MODEL

The proposed system is defined by a set of tuples, as follows:

- X = {U, Ad, SD, P, C, A, PA, F, DB,SDB, DDB}
- U = {U1,U2_ _ __Un }
  U is set of Users,
- C = {C1, C2_ _ __Cn}
  C is set of clients
- A= {A1, A2_ _ __An }
  A is set of attributes
- P = {P1, P2 }
  Where:
  P1 is Set of Regular Profile Information
  P2 is Set of Sensitive data of customer
- PA= {PA_1, PA_2}
  PA is set of partitioning attributes
  Where:
  PA_1 subset of A
  PA_2 subset of A
  P1, P2 subsets of C
- Ad is admin which is unique set
- SD = {SD1,SD2_ _ __SDn }
  SD is set of Source data
- F={F1,F2_ _ __Fn }
  F is set of Find the Dependencies which is required for partitioning
- DB= {SDB, DDB}
  DB is database
  SDB is the source database
- DDB= {DDB1, DDB2 }
  DDB is the destination database

All users i.e. the customer of the bank sending the source data to the source database

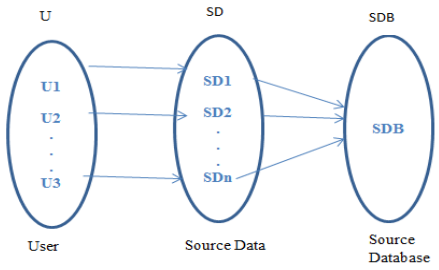

Fig 1.User interaction with the database

Fig.3 shows the Venn diagram for the saving the Partitioning data in the distributed databases.

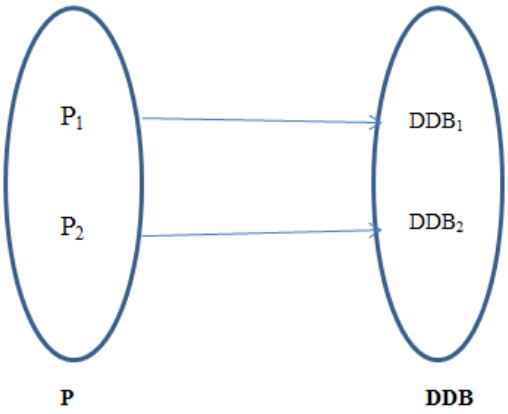

Fig 2. Venn diagram of partitioning data in distributed database

5. RESULT ANAYLISYS

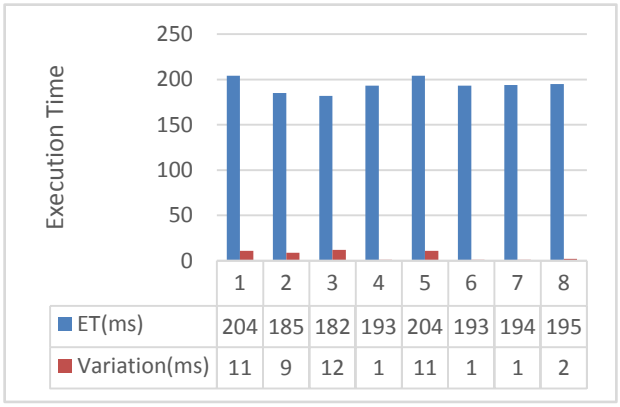

Fig 4. Execution Time and Variance

Here we have eight different queries and the ET column contain the execution time for each query. Variation is the difference between the current execution time and the average execution time. The last column is the percent variation which is calculated by using the following formula.

% variation= (variation/ET)*100

As shown in the table the approximate percent variation is 6.5 which is very less. Hence, proposed system will give the better query response time.

We executed a set of queries to check the query response time of the system. The responses are taken in milliseconds. We found that earlier the queries took slightly longer time to get the execution response as compared to now. Also there are variations in responses with each query fired. The fig. 4 shows these variations in response time. When first query fired, the response time was 215 milliseconds but when the same query was fired under our proposed system the response time was 204 milliseconds with variance of 11 milliseconds.

6. CONCLUSION

In this paper, we have proposed the vertical partitioning algorithm using the exponential mechanism to ensure safety of database while sharing the partial information of the database to the third parties. This mechanism is designed with the differential privacy model and also secures the data in the distributed framework. The query response time of the system is also improved by using the VPS, Statistics Collector, and Partitioning Generator.

7. ACKNOWLEDGMENT

We would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. Also, my family and friends for constantly encouraging and supporting me.

8. REFERENCES

[1] N. Mohammed, D.Alhadidi, B.C.M.Fung,"Secure Two-Party Differentially Private Data Release for Vertically Partitioned Data",Proc. IEEE TRANSACTION ON DEPENDABLE AND SECURE COMPUTING Volume. 11, No.1,Jan/Feb 2014,pp. 59-70

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith,"Calibrating Noise to Sensitivity in Private Data Analysis", Proc. Theory of Cryptography Conf. (TCC 06), 2006..

[3] B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," ACM Computing Surveys, vol. 42, no. 4, pp. 1-53, June 2010.

[4] N. Mohammed, R. Chen, B.C.M. Fung, and P.S. Yu, "Differentially Private Data Release for Data Mining," Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '11), 2011.

[5] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar, "Privacy Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release," Proc. ACM Symp. Principles of Database Systems (PODS '07), 2007.

[6] C. Dwork, "A Firm Foundation for Private Data Analysis," Comm. ACM, vol. 54, no. 1, pp. 86-95, 2011.

[7] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M.Y. Zhu, "Tools for Privacy Preserving Distributed Data Mining," ACM SIGKDD Explorations Newsletter, vol. 4, no. 2, pp. 28-34, Dec. 2002.

[8] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '02), 2002.

[9] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," J. Cryptology, vol. 15, no. 3, pp. 177-206, 2002

[10] J. Vaidya and C. Clifton, "Privacy-Preserving k-Means Clustering over Vertically Partitioned Data," Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '03), 2003.

[11] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our Data Ourselves: Privacy via Distributed Noise Generation," Proc. 25th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '06), 2006.

[12] A. Narayan and A. Haeberlen, "DJoin: Differentially Private Join Queries over Distributed Databases," Proc. 10th USENIX Conf. Operating Systems Design and Implementation (OSDI '12), 2012.

[13] W. Jiang and C. Clifton, "A Secure Distributed Framework for Achieving k-Anonymity," Very Large Data Bases J., vol. 15, no. 4, pp. 316-333, Nov. 2006.

[14] N. Mohammed, B.C.M. Fung, and M. Debbabi, "Anonymity Meets Game Theory: Secure Data Integration with Malicious Participants," Very Large Data Bases J., vol. 20, no. 4, pp. 567-588, Aug. 2011