An Improved Approach of Data Integration Based on Differential Privacy

Qihong Yu, Ruonan Rao
School of Software, Shanghai Jiaotong University, Shanghai, China
qihong891009@gmail.com, rnrao@sjtu.edu.cn

***Abstract*—Multiset operation and data transmission are the key operations for privacy preserving data integration because they involve the interaction of participants. This paper proposes an approach which contains anonymous multiset operation and distributed noise generation based on the existing researches and we apply it in data integration. Analysis shows that the improved approach provides security for data integration and has lower overhead than the existing researches.**

Keywords—differential privacy; data integration; multiset operation; noise generation

I. INTRODUCTION

With the rapid development of computer technology and the lower cost of storage, more and more data is stored in a variety of systems. Data sharing is becoming more and more important. For data mining, in order to dig out more useful and accurate information, data integration from various data sources is required. However, integration is unacceptable for the systems which has the obligation to protect the data from privacy disclosure. To eliminate the concern about this problem, privacy preserving of data integration is strongly required.

During data integration, data sources and data warehouse may not be creditable for each other. The multiset operation and data transmission between two parties can disclose the privacy to the other. Some of the existing researches put forward the methods of multiset operation and distributed noise addition. However, those methods have some defects should be modified in the situation of data integration. So this paper designs an approach to solve the problem in those method and uses differential privacy in distributed noise generation to improve the quality of data release.

II. RELATED WORK

A. Existing Researches

The existing technologies of privacy preserving can be divided into three parts [2]: data distortion, data anonymization and data encryption. These three types of privacy preserving technologies have their own applicable occasions. Data encryption method such as MPC [1] is always used to meet the requirement of distributed data mining. The distributed sites can operate the mining tasks without knowing the data of other sites [7], which does not involve data integration issues. Compared with data encryption, though data anonymization such as k-anonymous and data distortion such as differential privacy have a lower computational overhead [4], they are originally proposed to solve the privacy problem of public data dissemination, which assumes that the applied environment is in the centralized database. To apply them to the situation of data integration, more problem should be taken into consideration.

To generate the proper noise in distributed environment, reference [3] develops two algorithms to generate Binomial and Poisson noise in shares. This paper solves the problem of losing control during the distributed noise generation, which provides the idea of noise generation for our paper.

Reference [5] presents a protocol for multiset operation, and shows how this can be used to generate association rules where multiple parties have different (and private) information about the same set of individuals. This approach can be used to make the multiset operation anonymous. However, it involves the data sets transformation among different data sources, which increases the overhead of data sources.

B. Differential Privacy

In 2006, Dwork C. proposed a data distortion approach called differential privacy [6], which aims at ensuring adequate accuracy of records read from the database while minimizing the probability of privacy disclosed. This approach uses noise addition method to distort sensitive data to preserve the privacy [8]. While the processed data should be required to maintain some statistical properties, i.e. some statistical work such as data mining should not be affected. The detail definition is shown in Definition 1.

Definition 1: Pr stands for the risk of data being disclosed, K stands for a random function execute in the data set. K gives ε-differential privacy if for all value of DB, DB’ differing in a single element, and all S in Range(K)

$$\frac{\Pr(K(DB)\ in\ S)}{\Pr(K(DB')\ in\ S)} \le e^{\varepsilon} \sim (1+\varepsilon)$$

III. PROCESS OF PRIVATE DATA INTEGRATION

As shown in Fig. 1, a traditional data integration process includes the following steps: (1) Complete mapping operation among parties (1) Access to distributed data sources and execute data transmission (3) Complete data conversion, clean and integration during transmission (4) Data dissemination executed by data warehouse.

978-1-4799-2030-3 /14/$31.00 ©2014 IEEE

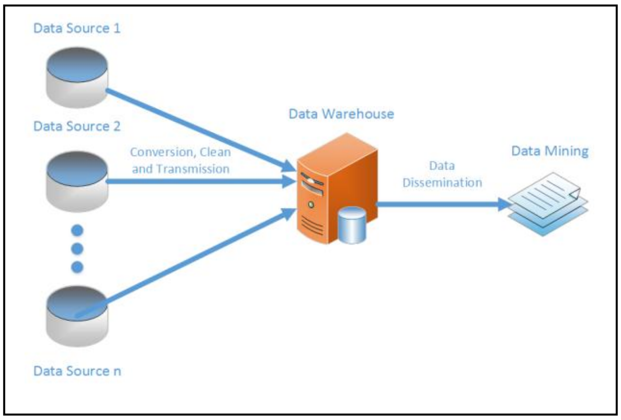

Fig. 1. Process of traditional data integration

According to the problem and process, we can conclude the following rules of privacy preserving data integration:

1) Relationship between data source and data warehouse is mutually untrusted. Noise addition should be executed before the data transmission to ensure privacy is not been disclosed.
2) Relationships among data sources are untrusted, Interaction among data sources should not involve real data. Data sets is isolated between data sources.
3) Because data sources are just the providers of data, they should not bear too much computing and transmission overhead.

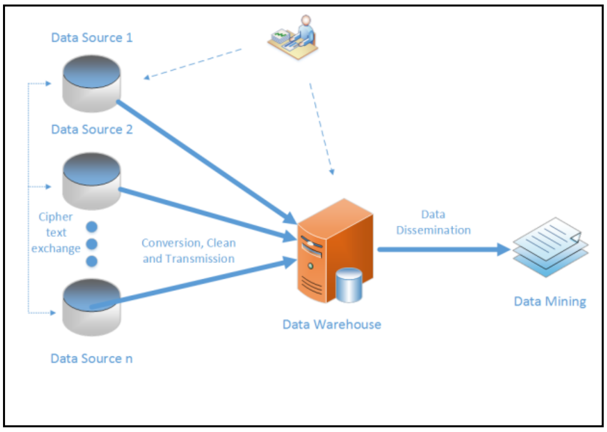

Fig. 2. Process of private data integration

To meet the above requirements, we add a director party to direct the multiset operation and noise generation among data sources. The architecture diagram can be change to Fig. 2. The director is responsible for the direction of cipher text exchange among data sources and private multiset operation in data warehouse. The differential privacy noise should be generated by its support.

IV. Anonymous Multiset Operation

Multiset operation plays an important role in data integration. Data sets from different data sources execute the operation through primary keys or a collection of fields as the uniquely identifier of entities. Traditional multiset operation will disclose the important information of entities to the data warehouse. So conversion or encryption for fields used for multiset operation is needed.

A. Design

Reference [5] proposed an approach for anonymous multiset operation. However, this approach is proposed to support the mining tasks in different data sources. So the whole data sets of each data sources are transformed between participants, which violates the fifth rule mentioned in Section III.

So we change the design of the approach. By adding the participant of director and change the transformed data structure. The new approach can be more suitable for data integration.

As shown in Fig. 2, the director makes responsible for the following two tasks: (1) Direction of encryption and communication among data sources; (2) Join guide for data warehouse after the data transmission.

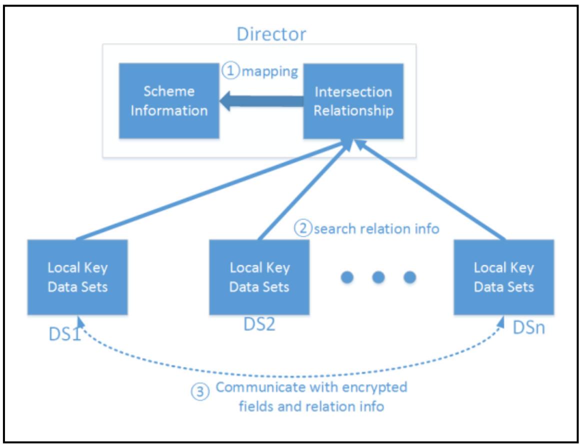

Fig. 3. Process of anonymous multiset operation

At first, director should obtain the schemes of data sources and data warehouse and design the join tasks. The design is mainly about drawing up the intersection relationship among the tables should be joined and mapping the data fields from data sources and data warehouse. To complete the join task without disclosing the sensitive data. We design an approach to complete the multi anonymous encryption among data sources. The algorithm of this approach will be detail described in part B.

B. Algorithm

As shown in Fig. 3, Director will maintain two kinds of information. They are the complete information of schemes of each data sources and information of intersection relationship IR_i among data sources.

The intersection relationship should tell the data sources the message about the data sources they should communicate with and the tables and fields they should use to execute the join tasks.

So each IR can be described as a set of three tuple $\{Set_{DS}, Set_{table}, Set_{field}\}$.

Besides the data set should be contained, each data source should keep a local key LK_i to help complete the encryption task.

To complete the private join task, we design the following anonymous encryption algorithm.

1) Each data source encrypts their own join fields with their own keys as EK_i.
2) Data source cleans up the sensitive fields used to complete the join task and fakes a unique key correspond to each record FKV_i as $\langle FK_i, record_i \rangle$.
3) Data source mapping the encrypted result with the fake key EKV_i as $\langle EK_i, FK_i \rangle$ to help the last join task.
4) Data source searches the relationship table in director and send the key-value data with the relationship record to the next party due to the record. To avoid causing falsify during the transmission, the data should be encrypted.
5) Next party gets the message and encrypts the data using its key and send to the next party. A flag should be assigned to true to represent the operation of the party has completed successfully. At last the encrypted data will send back to the original party.
6) This party will send the last key-value data and the data sets to the data warehouse.
7) Data warehouse completes the last join step due to the key-value and joins the data sets, then transforms to its own database based on the mapping information.

In the above algorithm, the choice of the encryption method is important. We follow the encryption rule in [5]:

1) For the different input data of specific fields, the encrypted output must be different.
2) Irreversible cryptographic operation .i.e. the encrypted data should be unrecoverable.
3) For two different cryptographic operations E_1 and E_2, $E_1(E_2(data))= E_2(E_1(data))$.

V. Distributed Noise Generation

If we simply aggregate the noise generated in each data source, the noise of each data source will compound and it is likely to affect the result of the operation using the integrated data. So we have to design a method to make the data sources cooperatively generate the noise to keep the noise in a controllable scale. One method is to remove the noise each data source generated and generate unified noise bases on the integrated data sets.

We define generated local noise as LN_i, the unified generated noise as UN, the noise-added data sets as DS_i. So the final data set can be described as follow:

$$DS = \bigcup_{i}^{n} DS_i - \sum_{i}^{n} LN_i + UN$$

While this method has the following problem:

1) If each data source simply tells the data warehouse the noise it generates. It may disclose the privacy of the data source.
2) The integrated data sets should not be disclosed to the data warehouse without noise distort.

One of the method can be design to solve the problem. The steps can be described as follow,

1) Each data source generates the noise and adds it to the local data sets. The noise should be keep by the data sources.
2) By using the algorithm proposed in section IV, data sources make the multiset operation with the noise-added data sets. The data warehouse will get an integrated data sets with compounded local noise.
3) Director command data sources operate by using a collaborate method to compute the unified noise. Then send the noise to the first data source.

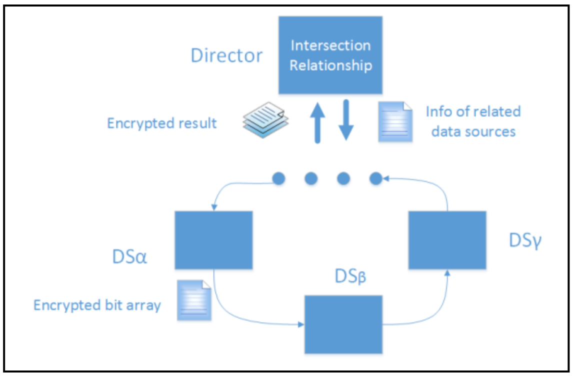

Fig. 4. the process of distributed noise generation

4) The director command first data source to minus the unified noise by the local noise. Then exchange the result to the next data source which has the multiset operation with it. Then the received data source does the same operation as last data source did and sends it to the next. Because the data sources have no the detail data information of the other. So none of them can get the real data sets message.
5) After the noise exchanging operation complete. The result will be sent to the data warehouse. Data

warehouse will add the data sets with the noise it received.

From the above steps we can see that the third step is the most important. How to compute the collaborative noise without letting each data source know the data of others is critical.

We do some change of the approach proposed in [3] to formulate the unified noise generation algorithm. As shown in Fig.4, the new algorithm should meet the following requirements:

1) The algorithm should clean up all the bias of each data source, i.e. the final noise we generate should not be deduced by the complicity data sources. Even the n-1 data sources are the attackers, the safety will be guaranteed.

2) The amount of the noise generated by the algorithm should be controllable, i.e. the noise should not destroy the availability and exactness even the number of data

3) Each data source and data warehouse cannot deduce the real version of data according to the unified noise.

By implement the approach proposed in reference [3], we do some change and design the following method to generate the unified noise:

1) Generate a random bit array for each data source, which is keep by the owner.

2) The director inquiry the intersection relationship information and direct the data sources in relation to prepare the array computing task.

3) The specific data source does homomorphic encryption for the local bit array then sends the cipertext to the next related data source.

4) The data source does the XOR bit array operation with the transformed and local cipertext, then transforms the result to the next. After one round cipertext anonymous exchange is complete, the result will be sent to the director.

5) The director decrypts the cipertext according to the homorphic encryption algorithm.

6) Director changes the distribution of XOR operation result to the specific noise using the method of fundamental transformation law of probabilities mentioned in Definition 2.

Definition 2: The Fundamental transformation law of probabilities is a good transformation function that converts a random variable of one distribution to another distribution. If we have a random variable x from a known distribution p(x), and a function y=y(x), the probability distribution of y, f(y), is determined through

$$\left|p(x)dx\right| = \left|f(y)dy\right| \Rightarrow f(y) = p(x)\left|\frac{dx}{dy}\right|$$

To get the transformation law y=y(x), we will integrate the above equation:

$$\int_{a_1}^{x} p(x')dx' = \int_{a_2}^{y} f(y')dy' \Rightarrow y = P^{-1}[F(x)]$$

By using the law y, the distributions can be transform to each other.

In Section II we describe the notion of differential privacy. To achieve ε-differential privacy, we use scaled symmetric noise $[\text{Lap}(R)]^d$ with R=Δf/ε. Lap(R) defines a scaled symmetric noise base on the Laplace distribution function.

$$Lap(R) = \frac{1}{2} R * \exp(-\frac{|x|}{R})$$

Δf defines the sensitivity of data query function.

To transform the distribution of XOR operation result, we put the Lap(R) into the formula of Definition 2.We can get the following result:

$$\int_{a_1}^{x} p(x')dx' = \int_{a_2}^{y} \frac{1}{2} R * \exp(-\frac{|y'|}{R})dy'$$

By this formula we can transform the distribution. After choosing the best Δf and ε based on the specific situation, The noise generation will complete.

VI. Security and Performance Evaluation

This section mainly analysis the security and performance of the data integrator. The security is mainly about the ability of avoiding the collusion attack and malicious attack. And the performance is mainly about the time complexity of algorithm and the transmission efficiency.

A. Security Analysis

Collusion attack is attackers who are the participant of the integrating work but want to get the real information of other participants by sharing the information of each attacker [9].

In our approach we need to consider the following two situations.

1) During the multiset operation, each data source holds one local key to be used to encrypt the data transformed by the upper data source, i.e. one round encryption should use all the keys to complete the task. Even when n-1 attackers cooperate to decrypt the data, they still cannot get the real information of the last participant.

2) During the noise generation situation, each participant uses their own local bit array to complete the generation of non-bias bit array. The result of the bit array can be described as follow:

$$B = b_1 \wedge b_2 \wedge \cdots \wedge b_n$$

Form the formula we can know that the result array B is the XOR operation of b_i. So even the n-1 bit array is known to the attackers. Without the array of the last participant, the result cannot be predicted.

Malicious attack is the attack one participant or external eavesdropper launches by breaking the protocol the approach draft. There is two situations we need to consider:

1) Malicious eavesdropping and tampering: The attacker gets the message by listening the transmission link. Because of the two main operations of the approach simply encrypted before the transmission. The listening is of no use. If the attacker wants to tamper with the cipertext, the signature verification will find the operation.

2) Missing Operation: Because the multiset operation and noise generation both use the one-round transmission to ensure the elimination of bias. So one attacker may change the order of the operation to let some of the data sources miss the operation. Because of the route markers we make for each time of transmission. If we check that the markers are in an error situation, The transmission will be abandoned.

B. *Performance Analysis*

The performance of data integration contains two parts: The local computing spending of the participants and communication overhead among participants. To make sure data sources take less computing task, we need acceptable time complexity of computing. To reach an acceptable time delay, a good communication overhead is preferred.

We assumes the number of data records in data source i is n_i. The number of participants is m. The main components of the local computing spending contains (1) the encryption of join fields for each data source (2) the noise generation in director (3) the simply data integration in data warehouse (4) the decryption and re-encryption when transmission.

The time complexity of 1, 2, 3 can be the linear complexity of $\sum_i n_i$. The complexity of 4 depends on the package size s of data being transformed. The complexity can be ms. In data integration, data sources should bear less computing tasks. So the complexity is acceptable.

For transmission overhead, we can know that the main cost parts are as follow:

1) The transmission of encrypted join fields between related data sources

2) The transmission of random bit arrays of each data sources

3) The transmission of noised data sets for data sources to data warehouse.

So the transmission overhead of our approach can be described as follow:

$$O = ml_{field}\sum n_i + ml_{bit} + l_{record}\sum n_i$$

The transmission overhead of previous research [5] is

$$O = kml_{record}\sum n_i + ml_{bit}$$

We can see that with the help of director, we can reduce the transformation time and the length of data need to be transformed. Our approach has a better transmission performance than the previous research [5].

VII. Conclusions

In this paper, we proposed an improved approach based on the existing researches to solve the problem of privacy preserving of data integration. The anonymous multiset operation and distributed noise generation method is the core of the approach which makes sure that data sources and data warehouse can be separated and keep their privacy without hinder the use of data. Analysis shows that the approach can ensure the safety and the approach can still be improved somehow. The fault tolerance of transmission can be in-depth studied to make the approach better.

Acknowledgment

We thank the partners of cloud platform of Fujian transport and logistics who proposes the requirement of privacy preserving data integration and give the good environment to make the research. We would like to acknowledge the financial support by the Research Funds of Shanghai Municipal Science and Technology Commission (No. 12511502902) and the National Natural Science Foundation of China (No. 61375053).

References

[1] W. Du and M. J. Atallah. Secure multi-party computational geometry. In Proceedings of the Seventh International Workshop on Algorithms and Data Structures, Providence, Rhode Island, Aug. 8-10 2001.

[2] AGGARWAL, C. C. AND YU, P. S. 2008c. Privacy-Preserving Data Mining: Models and Algorithms. Springer,Berlin.

[3] Dwork, Cynthia, et al. "Our data, ourselves: Privacy via distributed noise generation." Advances in Cryptology-EUROCRYPT 2006. Springer Berlin Heidelberg, 2006. 486-503.

[4] R. Agrawal and R. Srikant. Privacy preserving data mining. In Proceedings of the 19th ACM SIGMOD Conference on Management of Data, Dallas, Texas, USA, May 2000.

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

[6] DWORK C. Differential Privacy[C] //33rd International Colloquium on Automata, Language and Programming, Part II (ICALP 2006). Venice, Italy, Springer Verlag, July 2006

[7] Meyerson A, Williams R. On the complexity of optimal k-anonymity. In: Deutsch A, ed. Proc. of the 23rd ACM SIGACTSIGMOD-SIGART Symp. on Principles of Database Systems (PODS 2004). New York: ACM, 2004. 223−228.

[8] McSherry, Frank, and Ilya Mironov. "Differentially private recommender systems: building privacy into the net." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009.

[9] Nissim, Kobbi, Sofya Raskhodnikova, and Adam Smith. "Smooth sensitivity and sampling in private data analysis." Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. ACM, 2007.