Using Clustering Approach Privacy Preserving Update to Anonymous and Confidential Databases

Pravin Patil
P.G. Scholar, Department of CSE, DYPIT Bawada, Kolhapur, Maharashtra, India

Vinod Ingale
Assistant Professor, Department of CSE, AITRC Vita, Maharashtra, India

Sonali Patil
Assistant Professor, Department of IT, AITRC, Vita, Maharashtra, India

ABSTRACT
In order to update k-anonymous and confidential database, the suppression based and generalization based updating protocol technique has been proposed. These protocols check whether the database inserted with the new tuple is still k - anonymous without knowing the content of the table and database respectively. But these methods will not work if initial database is empty. Also, if the incoming tuple that fails the test of these updating protocols, there is no solution for which action to be taken. So, in this paper we propose two solutions based on pending tuple set (i.e. a collection of all tuples that fails anonymous property of database) namely the private extraction of k-anonymous part of pending tuple set or k-anonymization of pending tuple set by privately suppressing entries.

Keywords
Anonymity, Data management, Privacy, Secure computation

1. INTRODUCTION
In today's information society, provided the unprecedented ease of finding and accessing information, protection of privacy has become a very important concern. Data confidentiality is relevant because of the value. For example, medical data collected by using the history of patients over several years may represent sensitive information that needs to be protected. Such a requirement has motivated a large variety of approaches aiming at better protecting data confidentiality and data ownership. The availability of a large number of different databases which contains a large variety of information about individuals makes it possible to discover information about specific individuals by simply correlating the available databases. Basically confidentiality and privacy are different concepts. Data confidentiality is about the impossibility by an unauthorized user to access anything about data stored in the database and privacy relates to safely disclosing the data without leaking sensitive information regarding the legitimate owner. Consequently, if one asks whether confidentiality is still required if the data is anonymized then, the answer is yes, if the anonymous data have a business value for the party owning them or the unauthorized disclosure of such anonymous data may damage the party owning the data or other parties.

To understand the difference between confidentiality and anonymity, consider the example of a medical organization connected with a research institution. Assume that all patients submit their personal health care records and medical histories to medical organization under the condition that each patient's privacy is preserved against the research institution, which collects the records from medical organization database. To achieve the maximum privacy of patient data, medical organization sends data in an anonymous version to research institutes. Assume that any data of patients are related to the use of a drug over a period of some years and any side-effects have been observed and recorded by the researchers in the research database. It is clear that these data needs to be kept confidential (even if anonymized) and accessible only to the few researchers of the institution working on this project, until further research work is found about the drug.

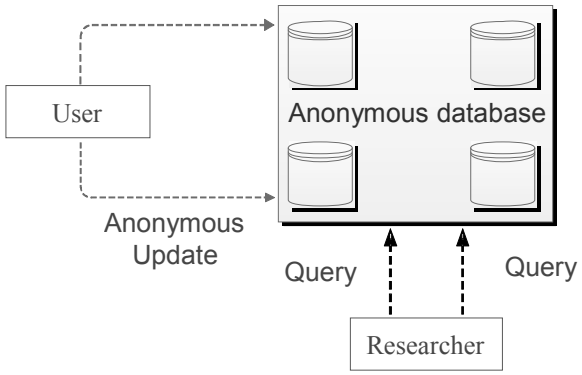

Figure 1: Anonymous System

Today there are several techniques available to address this problem of privacy via data anonymization, thus making it more difficult to link sensitive information of individuals. One well-known technique is k-anonymization [1]. This technique protects privacy by modifying the data so that the probability of linking a given data value is very small. Figure 1 captures the main participating parties in our application domain. Assume that the information concerning a single patient (or data provider) is stored in a single tuple, and DB is kept confidential on the server. The users in Figure 1 can be treated as medical researchers who have the access to the DB. Since DB is anonymous, the data provider's privacy is protected from these researchers.

Consider a table that provides health information of patients for medical studies, as shown in Table 1. Each row of the table consists of a patient's date of birth, zip code, allergy, and the past of illness. Although the identifier of each patient does not explicitly appear in this table, a dedicated adversary may be able to derive the identifiers of some patients using the combinations of date of birth and zip code. For example, he may be able to find that his roommate is the patient of the first row, who has an allergy to penicillin and a history of Pharyngitis.

In this example, the set of attributes, date of birth, zip code is called a quasi-identifier, because these attributes in combination can be used to identify an individual with a significant probability. In this paper, declare an attribute is a quasi-identifier attributes if it is in the quasi-identifier. The attributes like allergy and history of illness are called sensitive attributes. (There may be other attributes in a table besides the quasi-identifier attributes and the sensitive attributes; ignore

them in this paper since they are not relevant to our investigation.)

Table 1: A Table of Health Data

Date of Birth	Zip Code	Allergy	History of illness
03-24-79	07030	Penicillin	Pharyngitis
O8-02-57	07028	No Allergy	Stroke
11-12-39	07030	No Allergy	Polio
O8-02-57	07029	Sulfur	Diphtheria
08-01-40	07030	No Allergy	Colitis

The privacy threat we consider here is that an adversary may be able to link the sensitive attributes of some rows to the corresponding identifiers using the information provided in the quasi-identifiers. A proposed strategy to solve this problem is to make the table k-anonymous. The Table 2 shows health record after the anonymized.

Table 2: Two Anonymized Table of Health Data

Date of Birth	Zip Code	Allergy	History of Illness
*	07030	Penicillin	Pharyngitis
O8-02-57	0702*	No Allergy	Stroke
*	07030	No Allergy	Polio
O8-02-57	0702*	Sulfur	Diphtheria
*	07030	No Allergy	Colitis

The operation of updating such an anonymous database introduces problems similarly can the database owner decide if the updated database till preserves the privacy of individuals without directly knowing the new data to be inserted?

2. RELATED WORK

An approach to solve the problem of updating in the anonymous database initially discovered in [8].On the other hand, these protocols have some limitations for not supporting the generalization based updates, which is the key approach implemented for data anonymization. Thus, if the database is not anonymous analogous to the tuple to be inserted, the insertion cannot be performed. The current paper presents two proficient protocols, one of which supports the private update of a generalization-based anonymous database. All algorithms for database anonymization based on the idea of protecting databases through data suppression or data perturbation has been extensively investigated in the area of statistical databases. In the research area of statistical databases, it has been studied how to protect individual privacy while allowing information sharing [9].

Initially the concept of k-anonymous is introduced by Sweeney [1], based on medical data. The concept of k-anonymization is based on each row of table hidden in set of k tuples (similar), while making database k-anonymous by using k-anonymization by privately suppressing entries technique addressed by Zhong et al. [2]. However, this technique does not address the problem of private updates to k-anonymous databases.

Another research direction is Secure Multi-Party Computation (SMC) techniques; SMC represents a various class of techniques in the area of cryptography. The general techniques for performing protected computations are available [11].The technique of private information retrieval, which can be seen as an application of the secure multi-party Computation techniques in the area of data management. Here, the focus is to devise efficient techniques for posing expressive queries over a database without letting the database know the actual queries [12]. The previous system presents two proficient protocols, one of which also supports the private update of a generalization-based anonymous database [10]. All the algorithms for database Anonymization is based on the idea of protecting databases through data suppression or data perturbation has been extensively investigated in the area of statistical databases. In the research area of statistical databases, it has been studied how to protect individual privacy while allowing information sharing [9].k-anonymity has been verified are not specified; (ii) the specification of the actions to take in case privately updating protocol yields a negative answer; (iii) how to initially populate an empty table. Hence the goal is to sketch the solutions developed in order to address these questions and which encompass our overall methodology for the private database update.

3. OUR SOLUTIONS

3.1 Pending Tuple set

All tuples that fail the test of the private checker protocol of anonymous database updating, send to another table where all tuples are pending for getting service form database organizer so that table is called pending tuple set.

All unique tuples in the pending tuple set contain encrypted sensitive information for preserving privacy of the individual against database organizer. If the next tuple that fails the insertion and posted to the pending tuple set are having the quasi attributes values same as one of the tuple quasi-identifiers in the pending tuple set, then sensitive information of that two tuples are decrypted if they satisfy k-anonymous property.

- Each data provider encrypts her/his sensitive attributes using an encryption key that can be derived if and only if there are ≥ k rows (k=? i.e. minimum k=2) whose quasi-identifiers are equal.
- If and only if there are ≥ k data providers whose quasi-identifiers are equal, the database organizer is able to see the sensitive attributes.

3.2 Extraction of K-anonymous Part

In this problem formulation, the database organizer extracts the k-anonymous part of the table (i.e., the maximum subset of rows that is k-anonymous), but does not learn extra information about the sensitive attributes of the rows outside the k-anonymous part. Consequently, the database organizer cannot link the sensitive attributes of any row to the corresponding identifiers. Initially, our privacy requirement states that, for each party (database organizer or data provider), the view of the protocol seen by that party can be simulated by an algorithm that has no knowledge of the sensitive attributes outside the k-anonymous part. This captures the requirement that any individual party cannot learn any extra information about these sensitive attributes by virtue of engaging in the protocol.

The fundamental idea of our design is that each customer encrypts her sensitive attributes using an encryption key that can be derived if and only if there are at least k rows whose Quasi_Identifier are equal. Specifically, the key to encrypt the sensitive attributes (a(i)1 ; …….. ; a(i)n) is a function of the corresponding quasi-identifier (s(i)1 ; …….. ; s(i)m). As a

result, if and only if there are at least k customers whose quasi-identifiers are equal, the minor is able to recover a key.

3.3 K-Anonymization by Privately Suppressing Entries

In previous solution the failed tuple is maintained in pending tuple and wait until k-1 such tuple fails the insertion. After that, if subpart of the pending tuple set is anonymous than to provide service for that set of tuple the anonymized part is extracted from the pending tuple set and inserted into the original database. What action to take if all tuples in pending tuple set are unique?

This technique makes pending tuple set k-anonymized by suppressing entries _ideally suppressing as little as possible. Let Anonymized (T) denote the output (which is a k-anonymized table) of a protocol that k-anonyms the table T by suppressing entries. This technique is based on clustering algorithm using clustering algorithm make clusters of all tuple according to minimum distance. After this make all tuples in one cluster identical by suppressing minimum values of quasi-identifiers. Namely, it keeps all information about the suppressed entries private from each individual party, except revealing the distance between each pair of rows.

This protocol consists of three phases. In the first phase, the protocol allows the database organizer to compute the distance between each pair of rows. In the second phase, the database organizer uses the K-mean clustering algorithm to compute a k-partition of the table. (A k-partition is a collection of disjoint subsets of rows in which each division contains at least k rows and the union of these divisions is the entire table.) In the third phase, the protocol allows the database organizer to compute the k-anonymized table. After completing the clustering, a class-merging mechanism merges equivalence classes to make sure that all equivalence classes satisfy the k-anonymity requirement.

One problem with clustering, such as k-center [9], requires that a specific number of clusters be found in solutions. However, the k-anonymity problem does not have a restriction on the number of clusters; in its place, it requires that each cluster contains at least k records. To the best of our knowledge, this particular restriction has not been addressed in the existing clustering literature. Thus, to avoid this, k-anonymity problem as a new clustering problem referred to as k-member clustering problem has been introduced. This problem state that find a set of clusters from a given set of n records such that each cluster contains at least k ($k \leq n$) data points and that the sum of all intra-cluster distances is minimized.

4. CONCLUSIONS

The concept presents two approaches one of that provided service for every incoming tuple for insertion even if that tuple fails the test of secure protocol of the private updating anonymous database by periodically extracting k-anonymous part of pending tuple set (i.e. All tuples that fail insertion).If k-anonymous part not available in pending tuple set means all tuples in pending tuple are unique then by using another approach to populate the pending tuple set as an anonymous by using the k-mean clustering algorithm and make every tuple identical in one cluster by suppressing as few as possible. This technique used to populate the initially empty database as anonymous database instead of using dummy dataset for anonymization.

In future for a database system to effectively perform privacy preserving updates to a K-anonymous table, both approaches i.e. Extraction of k-anonymous part of pending tuple set and making pending tuple set k-anonymous are necessary, but in addition to the problem of failed insertion, there are other interesting and related issues that remain to be addressed.

(a) Devising private update techniques to a database system that supports the notion of anonymity different than k-anonymous.

(b) Dealing with case of malicious parties with the introduction of an untrusted, non-colluding third party.

(c) Devising anonymization of the set of tuple falling the insertion (i.e. pending tuple set) that supports the algorithm for anonymization of tuple set different than k-mean algorithm.

(d) Improving the efficiency of protocol in term of their required size.

(e) Improving the efficiency of protocol in term of their required time to update.

5. REFERENCES

[1] L. Sweeney. “K-anonymity: a model for protecting privacy”. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002

[2] P.Samarti.Protecting respondent’sprivacy in microdata release.IEEE Transactions on knowledge and Data Engineering,vol.13,no.6,pp.1010-1027,Nov/Dec.2001.

[3] US Department of Health & Human Services, Office for Civil Rights. Summary of the HIPAA Privacy Rule, 2003.

[4] S. Zhong, Z. Yang, R. N. Wright. “Privacy-enhancing k-anonymization of customer data”. In Proc. ACM Symposium on Principles of Database Systems (PODS), 2005.

[5] P. Samarati and L. Sweeney. Optimal anonymity using k-similar, a new clustering algorithm. Under review,2003.

[6] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati “k Anonymity”Universitµa degli Studi di Milano, 26013 Crema, Italia fciriani, decapita, foresti.

[7] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In Proc. 22nd ACMSIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Paris, France, June 2004.

[8] A. Trombetta, E. Bertino. “Private updates to anonymous databases”. In Proc. Int’l Conf. on Data Engineering (ICDE),Atlanta,Georgia,US 2006.

[9] N. R. Adam, J. C. Wortmann. “Security-control methods for statistical databases: a comparative study”, ACM Computing Surveys 1989

[10] Alberto Trombetta,Wei Jiang, Elisa Bertino and Lorenzo Bossi “Privacy-Preserving Updates to Anonymous and Confidential Databases” 2011.

[11] O. Goldreich. Foundations of Cryptography. Volume 2, Basic Applications, Cambridge University Press, 2004

[12] U.Maurer.The role of cryptography in database security.In Proc.of ACM SIGMOD Int’l Conf. on Management of Data,Paris,France,2004.

[13] R. Canetti, Y. Ishai, R. Kumar, m. K. Reiter, R. Rubinfeld, R.N. Wright. Selective private function evaluation with application to private statistics. In Proc. of ACM Symposium on Principles of Distributed Computing (PODC), Newport, Rhode Island, USA, 2001.