International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 8 Issue 1 –APRIL 2014.

PRIVACY PRESERVING UPDATES USING GENERALIZATION-BASED AND SUPPRESSION-BASED K-ANONYMITY

K.Dhivya and L.Prabhu
M.Tech IT,JJ College of Engineering and Technology,Trichy,India
M.Tech IT,JJ College of Engineering and Technology,Trichy,India

***Abstract*— One of the emerging concept in micro data protection is k-anonymity. It permits to assess the risk of disclosure for a data set protected with micro aggregation. Suppose if John owns a k-anonymous database and Kevin wants to insert his own tuple. After insertion if Kevin check the whole database to find out whether anonymity is maintained or not it will violate confidentiality maintained by John. On the other hand if John checks Kevin's data it will violate privacy. The problem is checking k-anonymity of the database without letting John and Kevin know content of tuple and database. In this paper we propose two protocols namely generalization-based and suppression based k-anonymous and confidential databases. These protocols rely on cryptographic assumptions.**

***Index Terms*— privacy, anonymity, generalization, Suppression, confidentiality, disclosure**.

I.INTRODUCTION

Today's globally networked society places great demand on the collection and sharing of person-specific data for many new uses . This happens at a time when more and more historically public information is also electronically available. Database is an important asset for many applications and their security is crucial. Data confidentiality is particularly relevant because of the value, often not only monetary, that data have. For example, medical data collected by following the history of patients over several years may represent an invaluable asset that needs to be adequately protected. Such a requirement has motivated a large variety of approaches aiming at better protecting data confidentiality and data ownership. Relevant approaches include query processing techniques for encrypted data and data watermarking techniques. Data confidentiality is not, however, the only requirement that needs to be addressed.

Today there is an increased concern for privacy. The availability of huge numbers of databases recording a large variety of information about individuals makes it possible to discover information about specific individuals by simply correlating all the available databases. Although confidentiality and privacy are often used as synonyms, they are different concepts: data confidentiality is about the difficulty by an unauthorized user to learn anything about data stored in the database. Usually, confidentiality is achieved by enforcing an access policy, or possibly by using some cryptographic tools. Privacy relates to what data can be safely disclosed without leaking sensitive information.

A release of data is said to adhere to *k*-anonymity if each released record has at least (*k*-1) other records also visible in the release whose values are indistinct over a special set of fields called the quasi-identifier. The quasi-identifier contains those fields that are likely to appear in other known data sets. Therefore, *k*-anonymity provides privacy protection by guaranteeing that each record relates to at least *k* individuals even if the released records are directly linked (or matched) to external information.

The operation of updating such a database, e.g., by inserting a tuple containing information about a given individual, introduces two problems concerning both the anonymity and confidentiality of the data stored in the database and the privacy of the individual to whom the data to be inserted are related: 1) Is the updated database still privacy preserving? and 2) Does the database owner need to know the data to be inserted? Clearly, the two problems are related in the sense that they can be combined into the following problem: can the database owner decide if the updated database still preserves privacy of individuals without directly knowing the new data to be inserted? The answer we give in this work is affirmative.

In this paper Section II describes existing system. Section III describes the proposed model. Section IV describes the primitives and notations used. Section V describes cryptographic primitives. Section VI describes the architecture of the proposed system and experimental results. Section VII describes an overall description of proposed system, various issues and future enhancement.

II. EXISITNG SYSTEM

The first research direction deals with algorithms for database anonymization. The idea of protecting databases through data suppression or data perturbation has been extensively investigated in the area of statistical databases [1]. Relevant work has been carried out by Sweeney [32], who initially proposed the notion of k-anonymity for databases in the context of medical data, and by Aggarwal et al. [2], who have developed complexity results concerning algorithms for k-anonymization. The problem of computing a k-anonymization of a set of tuples while maintaining the confidentiality of their content is addressed by Zhong et al. [35]. However, these proposals do not deal with the problem of private updates to k-anonymous databases. The problem of protecting the privacy of time varying data have recently spurred an intense research activity which can be roughly divided into two broad groups depending on whether data are continuously released in a stream and anonymized in an online fashion, or data are produced in different releases and subsequently anonymized in order to prevent correlations among different releases. Relevant work in this directions include [9], [14], [18], [21], and [34].

The second research direction is related to Secure Multiparty Computation (SMC) techniques. SMC represents an important class of techniques widely investigated in the area of cryptography. General techniques for performing secure computations are today available [16]. However, these techniques generally are not efficient. Such shortcomings have motivated further research in order to devise more efficient protocols for particular problems. Of particular relevance for data management are the techniques presented in [3], [13], in which the authors address the problems of efficiently and privately computing set intersection and database oriented operations, such as joins.

The third research direction is related to the area of private information retrieval, which can be seen as an application of the secure multiparty computation techniques to 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 [10], [22]. Again, the problem of privately updating a database has not been addressed in that these techniques only deal with data retrieval.

III. PROPOSED SYSTEM

In this paper in order to preserve privacy and confidentiality two protocols namely generalization and suppression is used. Fig. 1 captures the main participating parties in our application domain. We assume that the information concerning a single patient (or data provider) is stored in a single tuple, and DB is kept confidentially at the server. The users in Fig. 1 can be treated as medical researchers who have the access to DB. Since DB is anonymous, the data provider's privacy is protected from these researchers. As mentioned before, since DB contains privacy-sensitive data, one main concern is to protect the privacy of patients. Such task is guaranteed through the use of anonymization. Intuitively, if the database DB is anonymous, it is not possible to infer the patients' identities from the information contained in DB. This is achieved by blending information about patients. See Section 3 for a precise definition. Suppose now that a new patient has to be treated. Obviously, this means that the database has to be updated in order to store the tuple t containing the medical data of this patient.

Fig 1. Anonymous database system

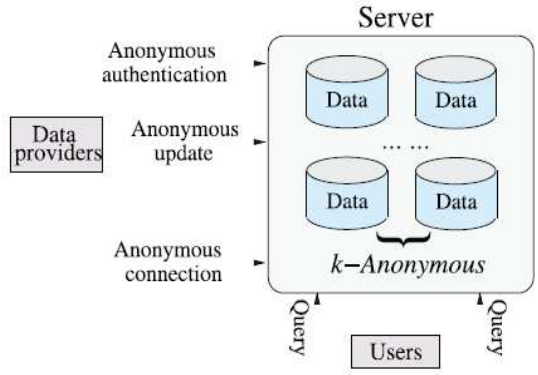


Fig. 1. Anonymous Database System.

Table 1: Anonymous Database System Requirements

Requirement	*Objective*	*Protocol*
Anonymous connection	Protect IP address and sensitive info	Crowds [27], Onior Routing [26]
Anonymous authentication	Protect sensitive authentication info	Policy-hiding access control [20]
Anonymous update	Protect non-anonymous data	**Proposed in this paper**

Fig. 1 summarizes the various phases of comprehensive approach to the problem of anonymous updates to confidential databases, while Table 1 summarizes the required techniques and identifies the role of our techniques in such approach.

IV. BASIC PRIMITIVES

We consider a table T {t1; . . . ; tn} over the attribute set A. The idea is to form subsets of indistinguishable tuples by masking the values of some well-chosen attributes. In particular, when using a suppression-based anonymization method, we mask with the special value, the value deployed by Kevin for the anonymization. When using a generalization-based anonymization method, original values are replaced by more general ones, according to a priori established VGH.

1. Quasi-Identifier (QI): A set of attributes that can be used with certain external information to identify a specific individual.
2. T[QI]: T[QI] is the projection of T to the set of attributes contained in QI.

TABLE 2
Original Data Set

AREA	POSITION	SALARY
Data Mining	Associate Professor	$90,000
Intrusion Detection	Assistant Professor	$78,000
Handheld Systems	Research Assistant	$17,000
Handheld Systems	Research Assistant	$15,500
Query Processing	Associate Professor	$100,000
Digital Forensics	Assistant Professor	$78,000

TABLE 3
Suppressed Data with k =2

AREA	POSITION	SALARY
*	Associate Professor	*
*	Assistant Professor	*
Handheld Systems	Research Assistant	*
Handheld Systems	Research Assistant	*
*	Associate Professor	*
*	Assistant Professor	*

TABLE 4
Generalized Data with k = 2

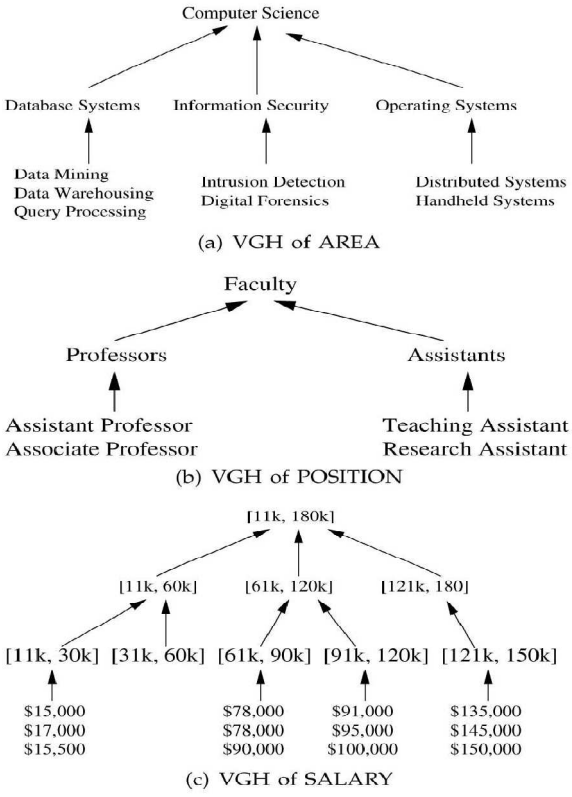
AREA	POSITION	SALARY
Database Systems	Associate Professor	[61k, 120k]
Information Security	Assistant Professor	[61k, 120k]
Operating Systems	Research Assistant	[11k, 30k]
Operation Systems	Research Assistant	[11k, 30k]
Database Systems	Associate Professor	[61k, 120k]
Information Security	Assistant Professor	[61k, 120k]

TABLE 5
The Witness Set

AREA	POSITION	SALARY
Database Systems	Associate Professor	[61k, 120k]
Information Security	Assistant Professor	[61k, 120k]
Operating Systems	Research Assistant	[11k, 30k]

With respect to suppression-based anonymization [23], [32] QI can be classified into two subsets: suppressed attributes and nonsuppressed attributes Suppose QI {AREA, POSITION, SALARY}, Table 3 shows a suppression based k-anonymization with k = 2.Choosing the suppressed attributes for every tuple of T is referred as the anonymization problem, and finding the anonymization that minimizes the number of masked values is an NP-hard problem.

Fig. 2. Value Generalization Hierarchies



For generalization-based anonymization [32], we assume that each attribute value can be mapped to a more general value. The main step in most generalization based k-anonymity protocols is to replace a specific value with a more general value. For instance, Fig. 2 contains VGHs for attributes AREA, POSITION, and SALARY. According to the VGH of AREA, we say that the value "Data Mining "can be generalized to "Database Systems". (Suppression can be viewed as an extreme form of generalization, in which the generalized attributes cannot be further generalized.)Let T refer to Table 4 and QI {AREA, POSITION,SALARY}. Then T (T[QI]) satisfies 2-anonymity. According to the three VGHs, it is easy to verify that the original data represented by Table 2 can be generalized to T. When T is k-anonymous, we can delete duplicate tuples, and we call the resulting set the witness set of T. Table 5 represents a witness set of Table 4.

V.CRYPTOGRAPHIC PRIMITIVES

The protocol in Section 4 uses a commutative, product homomorphic encryption scheme E. Loosely speaking, a commutative, product-homomorphic encryption scheme ensures that the order in which encryptions are performed is irrelevant (commutativity) and it allows to consistently perform arithmetic operations over encrypted data (homomorphic property). Further, for the security proofs we require that the encryption scheme E satisfies the indistinguishability property. We extend the definition of commutative, indistinguishable encryption scheme presented in [3], in order to obtain an encryption scheme which also product-homomorphic. Given a finite set K of keys and a finite domain D, commutative, product homomorphic encryption scheme E is a polynomial time computable function E : K*D->D satisfying the following properties:

A.Commutativity

For all key pairs K1,K2 £ K and value d £ D, the following equality holds

Ek1(Ek2(d))=Ek2(Ek1(d)) (1)

B. Product-homomorphism

For every K £ K and every value pairs d1, d2 £ D, the following equality holds:

Ek(d1).Ek(d2)=Ek(d1,d2) (2)

C. Indistinguishability

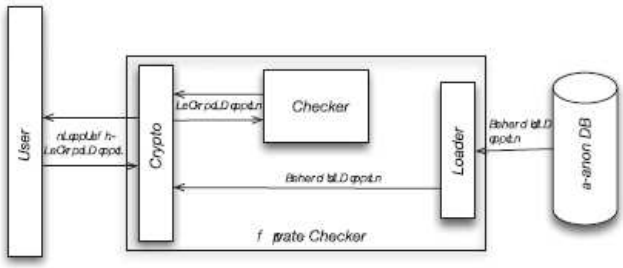
It is infeasible to distinguish an encryption from a randomly chosen value in the same domain and having the same length. In other words, it is infeasible for an adversary, with finite computational capability, to extract information about a plain text from the cipher text.

VI.ARCHITECURE AND EXPERIMENTAL RESULT

Our prototype of a Private Checker is composed by the following modules: a crypto module that is in charge of encrypting all the tuples exchanged between an user and the Private Updater, using the techniques exposed .The checker module that performs all the controls, as prescribed by generalization and suppression. The loader module that reads chunks of anonymized tuples from the k-anonymous DB. The chunk size is fixed in order to minimize the network overload.

Fig 3:Prototype Architecture overview

.



In Fig. 3 such modules are represented along with labeled arrows denoting what information are exchanged among them. Note that the functionality provided by the Private Checker prototype regards the check on whether the tuple insertion into the k-anonymous DB is possible. We do not address the issue of actually inserting a properly anonymized version of the tuple. The information flow across the above mentioned modules is as follows: after an initial setup phase in which the user and the Private Checker prototype exchange public values for correctly performing the subsequent cryptographic operations. If none of the tuples in the chunk matches the User tuple, then the loader reads another chunk of tuples from the k-anonymous DB. Note the communication between the prototype and User is mediated by an anonymizer (like Crowds, not shown in figure) and that all the tuples are encrypted.

Fig 4: Experimental result

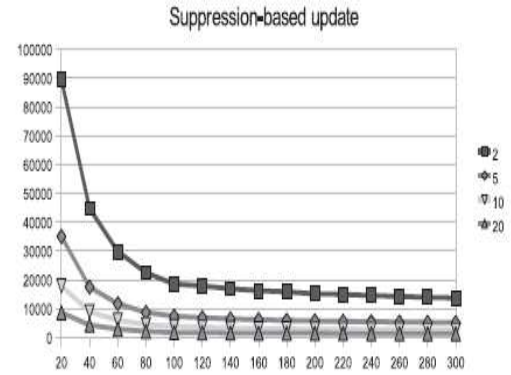
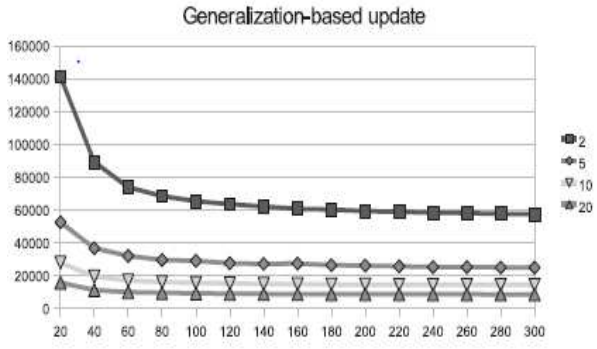


Fig 5: Experimental result



We report the average execution times (expressed in milliseconds) of suppression and generalization, respectively, in figures 4 and 5. The experiments confirm the fact that the time spent by both protocols in testing whether the tuple can be safely inserted in the anonymized database decreases as the value of k increases. Intuitively, this is due to the fact that the larger the k is, the smaller the witness set. Fewer are the partitions in which table T is divided.

Consequently, fewer protocol runs are needed to check whether the update can be made. Further, we report that the experiments confirm the fact that the execution times of of Protocols grow as dataset size=k. That is, each protocol has to check the anonymized tuple to be inserted against every witness in the worst case, and the larger the parameter k is, the fewer the witnesses are.

VII.CONCLUSION

In this paper, we have presented two secure protocols for privately checking whether a k-anonymous database retains its anonymity once a new tuple is being inserted to it. Since the proposed protocols ensure the updated database remains k-anonymous, the results returned from a user's (or a medical researcher's) query are also k-anonymous. Thus, the patient or the data provider's privacy cannot be violated from any query. As long as the database is updated properly using the proposed protocols, the user queries under our application domain are always privacy-preserving.

In order for a database system to effectively perform privacy preserving updates to a k-anonymous table, generalization and suppression are necessary but clearly not sufficient. Concerning the actual execution of the database update, once the system has verified that the user's tuple can be safely inserted to the database without compromising k-anonymity, the user is required to send to the Private Updater the non anonymous attributes' values to be stored in the k-anonymous database as well. The deployment of an anonymity system ensures that the system cannot associate the sender of the tuple with the subject who made the corresponding insertion's request. In case of malicious environment k-anonymous system will not work properly. Implelmenting a real-world anonymous database system is difficult. we believe that all these issues are very important and worthwhile to be pursued in the future.

REFERENCES

[1] N.R. Adam and J.C. Wortmann, "Security-Control Methods for Statistical Databases: *A Comparative Study," ACM Computing Surveys*, vol. 21, no. 4, pp. 515-556, 1989.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy,D. Thomas, and A. Zhu, "Anonymizing Tables," *Proc. Int'l Conf.Database Theory (ICDT)*, 2005.

[3] R. Agrawal, A. Evfimievski, and R. Srikant, "Information Sharing across Private Databases," *Proc. ACM SIGMOD Int'l Conf.Management of Data*, 2003.

[4] C. Blake and C. Merz, "UCI Repository of Machine Learning Databases," http://www.ics.uci.edu/mlearn/MLRepository. html, 1998.

[5] E. Bertino and R. Sandhu, "Database Security—Concepts, Approaches and Challenges," *IEEE Trans. Dependable and Secure Computing*, vol. 2, no. 1, pp. 2-19, Jan.-Mar. 2005.

[6] D. Boneh, "The Decision Diffie-Hellman Problem," *Proc. Int'l Algorithmic Number Theory Symp.*, pp. 48-63, 1998.

[7] D. Boneh, G. di Crescenzo, R. Ostrowsky, and G. Persiano, "Public Key Encryption with Keyword Search," *Proc. Eurocrypt Conf.*, 2004.

[8] S. Brands, "Untraceable Offline Cash in Wallets with Observers,"*Proc. CRYPTO Int'l Conf., pp. 302-318*, 1994.

[9] J.W. Byun, T. Li, E. Bertino, N. Li, and Y. Sohn, "Privacy-Preserving Incremental Data Dissemination," J. Computer Security,vol. 17, no. 1, pp. 43-68, 2009.
[10] R. Canetti, Y. Ishai, R. Kumar, M.K. Reiter, R. Rubinfeld, and R.N.Wright, "Selective Private Function Evaluation with Application to Private Statistics," *Proc. ACM Symp. Principles of Distributed Computing (PODC)*, 2001.
[11] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee,"Towards Privacy in Public Databases,"
[12] U. Feige, J. Kilian, and M. Naor, "A Minimal Model for Secure Computation," *Proc. ACM Symp. Theory of Computing (STOC)*,1994.
[13] M.J. Freedman, M. Naor, and B. Pinkas, "Efficient Private Matching and Set Intersection," *Proc. Eurocrypt Conf.*, 2004.
[14] B.C.M. Fung, K. Wang, A.W.C. Fu, and J. Pei, "Anonymity for Continuous Data Publishing," *Proc. Extending Database Technology Conf. (EDBT*), 2008.
[15] O. Goldreich, Foundations of Cryptography: Basic Tools, vol. 1.Cambridge Univ. Press, 2001.
[16] O. Goldreich, Foundations of Cryptography: Basic Applications, vol. 2.Cambridge Univ. Press, 2004.
[17] H. Hacigu¨mu¨ s¸, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over Encrypted Data in the Database-Service-Provider Model,"Proc. ACM SIGMOD Int'l Conf. Management of Data, 2002.
[18] Y. Han, J. Pei, B. Jiang, Y. Tao, and Y. Jia, "Continuous Privacy Preserving Publishing of Data Streams," *Proc. Extending Database Technology Conf. (EDBT)*, 2008.
[19] US Department of Health & Human Services, Office for Civil Rights, Summary of the HIPAA Privacy Rule, 2003.
[20] J. Li, N. Li, and W. Winsborough, "Policy-Hiding Access Control in Open Environment," *Proc. ACM Conf. Computer and Comm.Security (CCS)*, 2005.
[21] J. Li, B.C. Ooi, and W. Wang, "Anonymizing Streaming Data for Privacy Protection," *Proc. IEEE Int'l Conf. Database Eng. (ICDE)*,2008.
[22] U. Maurer, "The Role of Cryptography in Database Security,"*Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2004.
[23] A. Meyerson and R. Williams, "On the Complexity of Optimal KAnonymity,"*Proc. ACM Symp. Principles of Database Systems (PODS)*, 2004.
[24] S. Micali, M. Rabin, and J. Kilian, "Zero-Knowledge Sets," *Proc.44th Symp. Foundations of Computer Science*, 2003.
[25] T. Pedersen, "Noninteractive and Information-Theoretic Secure Verifiable Secret Sharing," *Lecture Notes in Computer Science*,vol. 576, pp. 129-140, 1991.
[26] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous Connections and Onion Routing," *IEEE J. Selected Areas in Comm.*, vol. 16,no. 4, pp. 482-494, May 1998.
[27] M.K. Reiter and A. Rubin, "Crowds: Anonymity with Web Transactions," *ACM Trans. Information and System Security (TISSEC)*, vol. 1, no. 1, pp. 66-92, 1998.
[28] P. Samarati, "Protecting Respondent's Privacy in Microdata Release," *IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6,pp. 1010-1027*, Nov./Dec. 2001.
[29] V. Shoup, "Lower Bounds for Discrete Logarithms and Related Problems," *Proc. Eurocrypt Conf.*, 1997.
[30] D.X. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," *Proc. IEEE Symp. Security and Privacy*, 2000.
[31] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," *Proc. ACM Conf. Computer and Comm. Security*, 1996.
[32] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5,pp. 557-570, 2002.
[33] A. Trombetta and E. Bertino, "Private Updates to Anonymous Databases," *Proc. Int'l Conf. Data Eng. (ICDE)*, 2006.
[34] K. Wang and B. Fung, "Anonymizing Sequential Releases," *Proc.ACM Knowledge Discovery and Data Mining Conf. (KDD)*, 2006.
[35] S. Zhong, Z. Yang, and R.N. Wright, "Privacy-Enhancing k-Anonymization of Customer Data," *Proc. ACM Symp. Principles of Database Systems (PODS)*, 2005.

Dhivya.K received Bachelor degree from Saranathan college of engineering under Anna University Chennai in 2013.Currently she is pursuing her Master Degree in J.J college of Engineering and technology,Trichy.

Prabhu.L received Bachelor degree from Dhanalakshmi srinivasan Engineering College under Anna university Trichy in 2012. Currently he is pursuing his Master Degree in JJ College of Engineering and technology,Trichy.