Apprising in Secured Manner to Anonymous and Confidential Databases

[1]Mahendrababu P [2]Rajarajan G
[1]Assistant professor/CSE, Chettinad College of Engineering & Technology
[2]Assistant professor/CSE, Chettinad College of Engineering & Technology

ABSTRACT - If Suppose Alice owns a k-anonymous database and needs to determine whether her database, when inserted with a data owned by Bob, is still k-anonymous. To maintain confidentiality database access is restricted to bob. Clearly, allowing Alice to directly read the contents of the tuple breaks the privacy of Bob (e.g., a patient's medical record); on the other hand, the confidentiality of the database managed by Alice is violated once Bob has access to the contents of the database. Thus, the problem is to check whether the database inserted with the tuple is still k-anonymous, without letting Alice and Bob know the contents of the tuple and the database, respectively. In this paper, we propose two protocols solving this problem on suppression-based and generalization-based k-anonymous and confidential databases. The protocols rely on well-known cryptographic measures.

I.INTRODUCTION

It is today well understood that databases represent an important asset for many applications and thus 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 (or impossibility) 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 regarding the legitimate owner [5].

Thus, if one asks whether confidentiality is still required once data have been anonymized, the reply 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. (Note that under the context of this paper, the term anonymized or anonymization means identifying information is removed from the original data to protect personal or private information. There are many ways to perform data anonymization. We only focus on the k-anonymization approach.)

To better understand the difference between confidentiality and anonymity, consider the case of a medical facility connected with a research institution. Suppose that all patients treated at the facility are asked before leaving the facility to donate their personal health care records and medical histories (under the condition that each patient's privacy is protected) to the research institution, which collects the records in a research database. To guarantee the maximum privacy to each patient, the medical facility only sends to the research database an anonymized version of the patient record. Once this anonymized record is stored in the research database, the non-anonymized version of the record is removed from the system of the medical facility.

Thus, the research database used by the researchers is anonymous. Suppose that certain data concerning patients are related to the use of a drug over a period of four years and certain side effects have been observed and recorded by the researchers in the research database. It is clear that these data (even if anonymized) need to be kept confidential and accessible only to the few researchers of the institution working on this project, until further evidence is found about the drug. If these anonymous data were to be disclosed, privacy of the patients would not be at risk; however the company manufacturing the drug may be adversely affected.

Recently, techniques addressing the problem of privacy via data anonymization have been developed, thus making it more difficult to link sensitive information to specific individuals. One well-known technique is k-anonymization. Such technique protects privacy by modifying the data so that the probability of linking a given data value, for example a given disease, to a specific individual is very small. So far, the problems of data confidentiality and anonymization have been considered separately. However, a relevant problem arises when data stored in a confidential, anonymity-preserving database need to be updated. 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.It is important to note that assuring that a database maintains the privacy of individuals to whom data are referred is often of interest not only to these individuals, but also to the organization owning the database. Because of current regulations, like HIPAA [19], organizations collecting data about individuals are under the obligation of assuring individual privacy. It is thus, in their interest to check the data that are entered in their databases do not violate privacy, and to perform such verification without seeing any sensitive data of an individual

II.RELATED WORK

A preliminary approach to this problem was investigated in. However, these protocols have some serious limitations, in that they do not support generalization-based updates, which is the main strategy adopted for data anonymization. Therefore, if the database is not anonymous with respect to a tuple to be inserted, the insertion cannot be performed. In addition one of the protocols is extremely inefficient. In the current paper, we present two efficient protocols, one of which also supports the private update of a generalization-based anonymous database. We also provide security proofs and experimental results for both protocols. So far no experimental results had been reported concerning such type of protocols; our results show that both protocols perform very efficiently. In what follows, we briefly address other research directions relevant for our work.

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 data-bases [1]. Relevant work has been carried out by Sweeney, who initially proposed the notion of k-anonymity for databases in the context of medical data, and by Aggarwal [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. 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 works in this direction include [9], [14], [18], [21]. Again, none of these works address the problem of checking that.

EXISTING SYSTEM:

Number of approach has been proposed, these protocols have some serious limitations, in that they do not support generalization-based updates, which is the main strategy adopted for data anonymization Therefore, if the database is not anonymous with respect to a tuple to be inserted, the insertion cannot be performed. In addition one of the protocols is extremely inefficient.

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.

Aggarwal proposed the notion of k-anonymity for databases in the context of medical data. 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. 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.

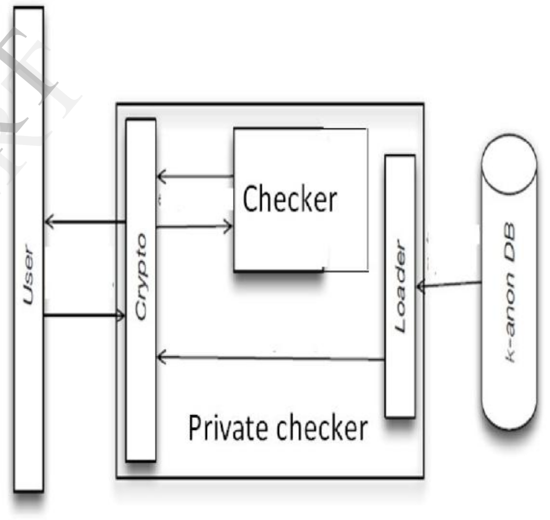
The second research direction is related to Secure Multiparty Computation (SMC) techniques. SMC represents an important class of techniques widely investigated in the area of cryptography. General techniques for performing secure computations are today available. However, these techniques generally are not efficient. Such shortcoming has motivated further research in order to devise more efficient protocols for particular problems.

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 the fourth research direction is related to query processing techniques for encrypted data. These approaches do not address the k-anonymity problem since their goal is to encrypt data, so that their management can be outsourced to external entities.

PROPOSED SYSTEM:

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.The first protocol is aimed at suppression-based anonymous databases, and it allows the owner of DB to properly anonymize the tuple t, without gaining any useful knowledge on its contents and without having to send to t's owner newly generated data. To achieve such goal, the parties secure their messages by encrypting them. In order to perform the privacy-preserving verification of the database anonymity upon the insertion, the parties use a commutative and homomorphic encryption scheme. The second protocol is aimed at generalization-based anonymous databases, and it relies on a secure set intersection protocol, to support privacy-preserving updates on a generalization-based k-anonymous DB. In particular, when using a suppression-based anonymization method, we mask with the special value *, the value deployed by Alice for the anonymization. When using a generalization-based anonymization method, original values are replaced by more general ones, according to a priori established value generalization hierarchies (VGHs).

SYSTEM ARCHITECHTURE:



Our prototype of a Private Checker (that is, User A) is composed by the following modules: a crypto module that is in charge of encrypting all the tuples exchanged between a user (that is, User B) and the Private Updater, using the techniques (suppression and generalized); a checker module that performs all the controls; a 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. 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 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, the user sends the encryption E(c(δ_i)) of her/his tuple to the Private Checker; the loader module reads from the k-anonymous DB the first chunk of tuples to be checked with E(c(δ_i)). Such tuples are then encrypted by the crypto module. The checker module performs the above mentioned check one tuple at time in collaboration with the user, according to either Protocol (in the case of suppression-based anonymization) or Protocol (in the case of generalization-based anonymization). 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 and that all the tuples are encrypted

Suppression-Based Anonymous and Confidential Databases

In this suppression based anonymous method the original values are masked with the special value *.

Working of Suppression Method:

1. User A encrypts tuple with his private key and sends to User B.
2. User B read each attribute values in table t and encrypts tuple with his key and sends to User A.
3. User A decrypt the values send by User B;
4. The encrypted values sent by User B are ordered according to the ordering of the attributes in T (assume this is a public information known to both User A and User B), User A knows which are, among the encrypted values sent by User B, the one corresponding to the suppressed and non-suppressed QI attributes.
5. User A checks the condition If true, t (properly anonymized) can be inserted to table T. Otherwise, when inserted to T, t breaks k-anonymity.

Generalized –Based Anonymous and Confidential Databases

In this generalization based anonymization method, original values are replaced by more general ones, according to a priori established value generalization hierarchies (VGHs).

Working of Generalized Method:

1. User A randomly chooses a δ ε T_w (Witness Set).
2. User A computes γ = GetSpec (δ) (bottom values of VGH(Value Generation Hierarchy)).
3. User A and User B collaboratively compute s= SSI(γ,τ) (cardinality of γ ∩τ).
4. If s=u then t's generalized form can be safely inserted to T.
5. Otherwise, Alice computes $T_w \leftarrow T_w - \{\delta\}$ and repeat the above procedures until either s=u or $T_w = \varnothing$;

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

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, Ap-proaches 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," Proc. Theory of Cryptography Conf. (TCC), 2005.

[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. Oli, and W. Wang "Anonymizing Streaming Data for Privacy Protection'', Proc. IEEE Int'1 Conf. Database Eng. (ICDE), 2008.