Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 2, February 2015, pg.255 – 259

RESEARCH ARTICLE

Securely Sharing Data Using AIDA

Nihalahmad R.Shikalgar[1], Pravin B. More[2]

[1,2]Assistant Professor, Adarsh Institute of Technology and Research Center, Shivaji University, India

***Abstract:** In this paper, we study the strategies for efficiently achieving data sharing among networks. Unlike the traditional research, we do not intend to identify the access patterns to facilitate the future requests. Instead, with such a kind of information presumably known in advance, our goal is to efficiently share the data items. Existing work is consisting of secure sum data mining operation using Newton's identities and Sturm's theorem. For achieving secure data sharing in distributed system, our algorithm is built for accurately work on such environment. It is the extended work over to traditional algorithm. In the proposed algorithm developed for anonymous sharing of private data among parties. Secret Identity assignment is consisting of hiding original identity, so that Anonymous ID assignment (AIDA) is used. This assignment of secret numbers allows more complex data to be shared and has applications to other problems in privacy preserving data mining. The required computations are distributed without using a trusted central authority.*

***Keywords:** Cloud Computing, distributed systems, privacy preservation, privacy protection*

1. Introduction

WITH increasing data accessibility demands on clouds, data availability maximization seems to be an important problem to consider to maintain high-fidelity and time-bounded service expectations in clouds.The increasing use of internet as a communication medium whether for individual or commercial use depends in part on its support for anonymous communication. Industries also have appropriate reasons to participate in anonymous statement and avoid the significances of identity revelation. For example, to allow distribution of summary data without revealing the identity of the thing the underlying data is associated with, or to protect whistle-blower's right to be unsigned and free from party-political [1]or economic vengeances. Cloud-based website management tools provide capabilities [2] for a server to anonymously capture the visitor's web actions. The problem of sharing privately held data so that the individuals who are the subjects of the data cannot be identified has been researched extensively [10]. Investigators have also inspected the relevance of anonymity and/or privacy in various application domains: The associate editor coordinating the review of this patient medical records [4], electronic voting [5], e-mail [6], social networking [7], etc. Another form of anonymity, as used in secure multiparty computation, allows multiple parties on a network to jointly carry out a global computation that depends on data from each party while the data held by each party remains unknown to the other parties. A secure totaling function widely used in the literature is secure sum that allows parties to compute the sum of their individual inputs without disclosing the inputs to one another [10, 11]. This function is popular in data mining applications and also helps characterize the difficulties of the secure multiparty computation.

This work deals with efficient algorithms for transmission identifiers (IDs) to the nodes of a network in such a way that the IDs are anonymous using a distributed computation with no central authority. Given nodes, this assignment is essentially a variation of the integers with each ID being known only to the node to which it is assigned. Our main algorithm is based on a method for anonymously sharing simple data and results in methods for efficient sharing of complex data. There are many applications that require dynamic unique IDs [12] for network nodes . Such IDs can be used as part of schemes for sharing/dividing communications bandwidth, data storage, and other possessions anonymously and without conflict. The IDs are needed in sensor networks for security or for administrative tasks requiring reliability [3], such as conformation and monitoring of individual nodes, and download of binary code or data aggregation descriptions to these nodes. An application where IDs need to be anonymous is grid calculating where one may seek services without revealing the identity of the service requestor.

2. System Analysis

2.1 Existing System

Existing and new algorithms for assigning anonymous IDs are examined with respect to trade-offs between communication and computational requirements.. Also, suppose that access to the database is strictly controlled, because data are used for certain experiments that need to be maintained confidential. Clearly, allowing Alice to directly read the contents of the tuple breaks the privacy of Bob; 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.

Disadvantage:

1. The database with the tuple data does not be maintained confidentially.
2. The existing systems another person to easily access database.

2.2 Proposed System

An algorithm for anonymous sharing of private data among parties is developed. This technique is used iteratively to assign these nodes ID numbers ranging from 1 to N. This assignment is anonymous in that the identities received are unknown to the other members of the group. Resistance to collusion among other members is verified in an information theoretic sense when private communication channels are used. This assignment of serial numbers allows more complex data to be shared and has applications to other problems in privacy preserving data mining, collision avoidance in communications and distributed database access. The required computations are distributed without using a trustedcentral authority.

Advantage:

1. The anonymity of DB is not affected by inserting the records.
2. We provide security proofs and experimental results for both protocols.

3. Algorithmic Strategies

3.1 Homomorphic encryption Module:

This module to use 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.

3.2 Generalization Module:

In this module, the second protocol is aimed at generalization-based anonymous databases, and it relies on a secure set intersection protocol, such as the one found in, to support privacy-preserving updates on a generalization based k-anonymous DB.

3.3 Cryptography Module:

In this module, the process of converting ordinary information called plaintext into unintelligible gibberish called cipher text. Decryption is the reverse, in other words, moving from the unintelligible cipher text back to plaintext. A *cipher* (or) *cypher* is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a *key*. This is a secret parameter (ideally known only to the communicants) for a specific message exchange context.

3.4 User and Admin Module:

In this module, to arrange the database based on the patient and doctor details and records. The admin to encrypt the patient reports using encryption techniques using suppression and generalization protocols.

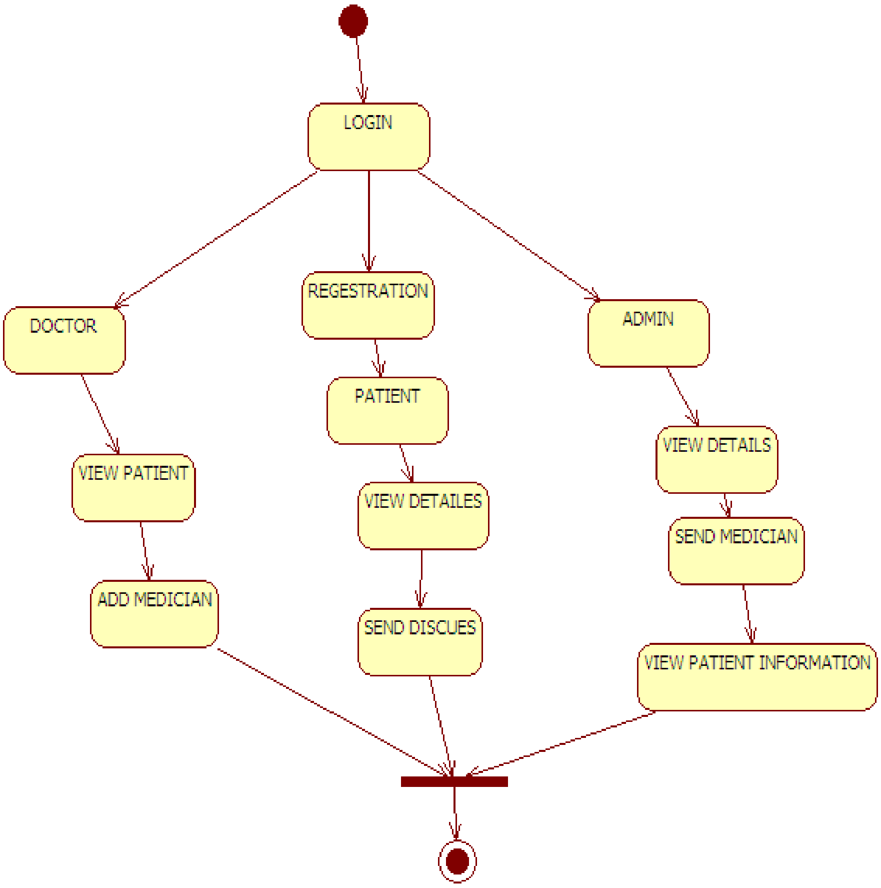

Figure 1: State diagram of proposed system

4. Practical Impact of Proposed System

4.1 Functional Requirements:

Inputs:

Browsing and uploading of files..

Processing:

Cluster server: There are 3 cluster servers Cluster server1 stores files of server1.Cluster server2 stores files of server2.cluster server3 stores files of server3.

Load server: Stores all files

Slip server cluster:

Browses the file
Selects the path
Download the file

Output: SIP user agent clients select file and location to download the file. To download the selected file server will send file to the SIP user agent.

4.2 Non Functional Requirements

Performance is measured in terms of the output provided by the application.

Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of users of the existing system to give the requirement specifications because they are the people who finally use the system.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system.
- The system should be accurate.
- Te system should be better than existing system.

Portability: It should run on specified platforms successfully. To achieve this we should test the product on all platforms before launching the product. If our project runs successfully on different platforms then our system is portable in nature.

Reliability: The system should perform its intended functions under specified conditions. If our system satisfies all the specified conditions then it is Reliable in nature.

Reusability: The system should be extremely reusable as a whole or part. Make the system modularize and make sure that modules are loosely coupled. This project is having reusability nature because we can reuse whole or part of this project on other systems.

Robustness: The system on the whole should be robust enough to perform well under different circumstances without any inconsistencies.

Testability: The product of a given development phase should satisfy the conditions imposed at the start of that phase.

Usability: It should be perfect and comfortable for users to work.

Security: The system is completely based on the security. This system will provide security base on the password.

5. Conclusion

Proposed algorithm greatly decreases communication overhead. It can work efficiently for secure sharing of data by using anonymous key assignment techniques. This can reduce number of round required to execute. This solution can complete within polynomial time. It gives exact completion time. In private communication channels, our algorithms are secure in an information theoretic sense.

In future there is consideration on communication requirements. It is depends on implementation of different kind of algorithm. It may useful merging of different methods to reduce communication overhead.

References

[1] Sarbanes–Oxley Act of 2002, Title 29, Code of Federal Regulations, Part 1980, 2003.

[2] White Paper—The Essential Guide to Web Analytics Vendor Selection, IBM [Online]. Available: http://measure.coremetrics.com/corem/getform/ reg/wp-evaluation-guide.

[3] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[4] A. Friedman, R. Wolff, and A. Schuster, "Providing k-anonymity in data mining," *VLDB Journal*, vol. 17, no. 4, pp. 789–804, Jul. 2008.

[5] F. Baiardi, A. Falleni, R. Granchi, F. Martinelli, M. Petrocchi, and A. Vaccarelli, "Seas, a secure e-voting protocol: Design and implementation,"*Comput. Security*, vol. 24, no. 8, pp. 642–652, Nov. 2005.

[6] D. Chaum, "Untraceable electronic mail, return address and digital pseudonyms," *Commun. ACM,* vol. 24, no. 2, pp. 84–88, Feb. 1981.

[7] Q. Xie and U. Hengartner, "Privacy-preserving matchmaking for mobile social networking secure against malicious users," in *Proc. 9th Ann. IEEE Conf. Privacy, Security and Trust*, Jul. 2011, pp. 252–259.

[8] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proc. 19th Ann. ACM Conf. Theory of Computing*, Jan. 1987.

[9] A. Yao, "Protocols for secure computations," in *Proc. 23rd Ann. IEEESymp. Foundations of Computer Science*, 1982, pp. 160–164, IEEE Computer Society.

[10] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, “Tools for privacy preserving distributed data mining,” *ACM SIGKDD ExplorationsNewsletter*, vol. 4, no. 2, pp. 28–34, Dec. 2002.

[11] J. Wang, T. Fukasama, S. Urabe, and T. Takata, “A collusion-resistanta pproach to privacy-preserving distributed data mining,” *IEICE Trans. Inf. Syst. (Inst. Electron. Inf. Commun. Eng.)*, vol. E89-D, no. 11, pp. 2739–2747, 2006.

[12] J. Smith, “Distributing identity [symmetry breaking distributed access protocols],” *IEEE Robot. Autom. Mag.*, vol. 6, no. 1, pp. 49–56, Mar.1999.