Australian Journal of Basic and Applied Sciences, 9(11) May 2015, Pages: 755-758

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Privacy and Security in Distributed Information Sharing Through Brokering System

Veeramalai Sankaradass, D.Daya Florance, M. Gowthami, J. Briskilal

Department of Computer Science and Engineering Vel Tech High Tech Dr. Rengarajan Dr. Sagundhala Engineering College, Chennai, Tamil Nadu, India

ARTICLE INFO

Article history:
Received 25 March 2015
Accepted 28 May 2015
Available online 14 June 2015

Keywords:
Information brokering, Automaton Segmentation, Commutative encryption, Privacy preserving Component.

ABSTRACT

Privacy Preserving is used to protect delicate information in the network. Security is the importance of being secure from any obliteration. Allocating the particulars between administrations becomes universal to expand the extensive spread association. Information Brokering will try to find to construct steering decisions to direct client demands to the invited data servers. Securing the persons reserved facts through brokers is fewer in the information brokering system. At this time the privacy problem will arise. To maximize the privacy, in this paper I concentrate on the privacy preserving information brokering approach for exchanging multiple shareholders information without dripping a data. Then we use the automaton segmentation and the commutative encryption scheme for refining the level-based encryption in flexible method. It is also used to decrease the communication cost and computation cost.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: Veeramalai Sankaradass, D.Daya Florance, M. Gowthami, J. Briskilal, Privacy and Security in Distributed Information Sharing Through Brokering System. ***Aust. J. Basic & Appl. Sci.,*** *9(11): 755-758, 2015*

INTRODUCTION

Organizations desire to provide data access to their collaborators while preserving full Control over the data and comprehensive privacy of their users. A amount of information systems have been proposed to recommend efficient and secure information sharing (Li, 2013). The difficulty of balancing peer autonomy and system coalition is silent tricky. The enlargement of dispersed information management (DIM) applications has followed the rise of the Internet. In these applications, users store information on a site for the purpose of sharing it with recipients (Li, 2007).

On the other hand, distribution of more information is dangerous because the recipients of the information or the scheme itself (imagine it is not prohibited by the information owner), may have encouragement to distribute sensitive data through eaves droppers with the intention of a user has not approved to assessment of his data, but would get from significant it.

In particular, recent extensible Markup Language (XML) has become a promising solution by integrating incompatible data while preserving semantics. An XML-oriented information brokerage system has data sources and brokers hold XML documents and document distribution Information (Carzaniga, 2004). In such systems, databases know how to be queried all the way during brokers with no schema-relevant or geographical difference being noticed.

In recent trends, organizations raise an increasing need of information sharing to facilitate extensive collaboration among business to administration agencies. In long-ago days Information brokering system(IBS) which acts as intermediate brokers which make decision routing between the query requester and data server by trusting third party. To prevail over the trusted third party IBS, extended into privacy preserving for exchanging multiple stakeholders information.

A reference architecture for distributed database management systems from system and schema viewpoints and show how various FDBS architectures can be developed. (Back, 2001). The basics of database federation, introduce several styles of database federation, and outline the conditions under which each style of federation should be used (Blum, 2004). Our PC-based XML router prototype can route an 18 Mbit per second XML stream (Chaum, 1981; Dingledine, 2000).

Challenges **t**aken-for-granted access control strategies can have a significant impact on the "whole" system's end-to-end deployment methodology, and argue that query brokering and access control are not two orthogonal issues because access control deployment performance. We propose the first in-broker access control deployment strategy where access control is "pushed" from the boundary

Corresponding Author: Veeramalai Sankaradass, Department of Computer Science and Engineering Vel Tech High Tech Dr. Rengarajan Dr. Sagundhala Engineering College, Chennai, Tamil Nadu, India
E-mail: veera2000uk@gmail.com

into the "heart" of the information brokerage system (Donoho, 2002; Eclipse, 2010).

Information Brokering System:

An IBS system is a peer-to-peer overlay network consisting of data owners, brokers and end users. This kind of architecture is engaged in an inter association scenario, where different organizations have prevailing needs of cross-organizational information sharing.

Information Brokering System (IBS) is an approach which includes a particular set of brokers. This broker will acts as an agent between the query requestor and the data server. Many brokers will be caught up in forwarding the queries but all the brokers cannot be a trusted one. The broker's functionality may possibly be outsourced to the intermediary providers. At this position the confidentiality concern can happen. Not only that, the hacker will hack the data by using two techniques. They are *attribute-correlation hits* and *inference hits*

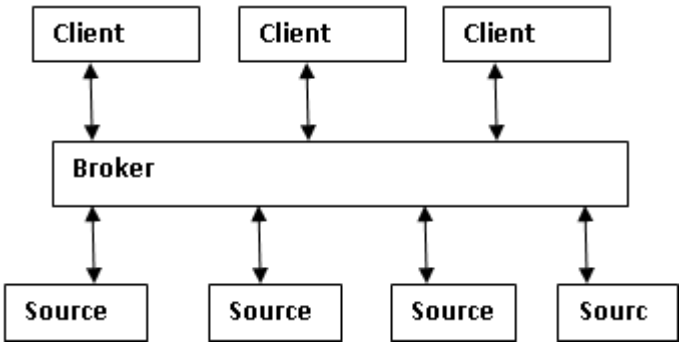

Fig. 1: Overview of the IBS infrastructure.

Attribute Correlation Attack:

The information will be send to brokers will be in the form of XML query. The attacker can "evaluate" the attributes in the predicates to gather sensitive information with reference to data owner. This is recognized as the *attribute correlation hits*.

For example: A patient Bob is asked to sent and consult Hospital. Doctor posted queries of her medical records with the help of a Medicare IBS. Because Bob has the indication of blood cancer. The query contains two predicates: [pName="Bob"] and[symptom=" cancer"]. Several malicious broker can guess that "Bob have a cancer" by correlating the two predicates in the query.

Inference Attack:

This attack is taking place by pronouncement the location of the data server or the data owner by using the IP address.

For example: if an attacker came to know that a data server is placed at a cancer research center, he know how to tag the queries as "cancer-related".

To conquer these attacks, Privacy Preserving Information Brokering is used. Beneath this two algorithms are used. They are Commutative encryption scheme and Automaton Segmentation

Privacy Preserving Information Brokering:
System:

It is an overlay infrastructure consisting of two types of brokering components, *brokers* and *coordinators*. The brokers, performing as mix anonymizer, are mostly responsible for user authentication and query forwarding. The coordinators, concatenated in a tree structure, enforce access control and query routing based on the embedded nondeterministic finite automata the *query brokering automata*.

To prevent snooping or dishonored coordinators from inferring confidential information, we suggest two new schemes to fragment the query brokering automata and encrypt related query segments so that routing judgment making is decoupled into multiple correlated tasks for a particular set of collaborative coordinators. While on condition that integrated in-network access control and content-based query routing.

System Architecture:

The patient will register his details in the hospital. The hospital will create a digital certificate from the certificate authority and provide to the patient through mail. The hospital will send the send the information to the server along with the digital certificate. The data will be encrypted using Rijndael algorithm and stored in the coordinators of the server. By encrypting and decrypting the information are transferred from one hospital to another. The other hospital retrieves the information easily from the server by decrypting the data using Rijndael algorithm. In Pharmacy, before issuing the medicine the details of the patient are checked in the server.

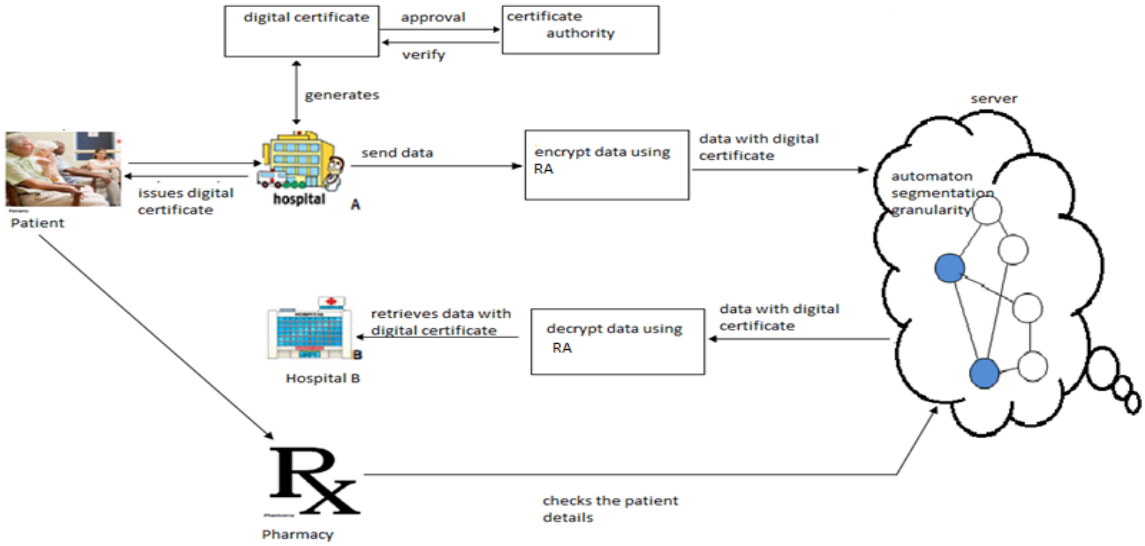

Fig. 2: System Architecture.

Automaton Segmentation:

In PPIB, we adopt the view-free automaton-based access control mechanism, and extend it in a decentralized manner with our Automaton Segmentation scheme. The idea of automaton segmentation comes from the concept of multilateral security: split sensitive information to largely meaningless shares held by multiple parties who cooperate to share the privacy-preserving responsibility.

Then we assign each segment to one independent site.

1. Segmentation: The atomic unit in the segmentation is an NFA state of the original automaton. Each segment is allowed to hold one or several NFA states.

2. Deployment: We employ physical brokering servers, called *coordinators*, to store the logical segments. To reduce the number of needed coordinators, several segments can be deployed on the same coordinator using different port numbers. Therefore, the tuple uniquely identifies a segment.

3. Replication: Since all the queries are supposed to be processed first by the root coordinator, it becomes a single point of failure and a performance bottleneck. For robustness, we need to replicate the root coordinator as well as the coordinators at higher levels of the coordinator tree. Replication has been extensively studied in distributed systems.

Algorithm 1: The automaton segmentation algorithm:

deploySegment()
Input: Automaton State S
Output: Segment Address:addr
1: for each symbol in S.StateTransTabledo
2: addr= deploySegment
(S.StateTransTable(k).nextState)
3: DS=createDummyAcceptState()
4: DS.nextState ←addr
5: S.StateTransTable(k).nextState ← DS
6: endfor
7: Seg=createSegment()
8: Seg.addSegment(S)
9: Coordinator=getCoordinator()
10:Coordinator.assignSegment(Seg)
11: return Coordinator.address

Commutative Encryption Scheme:

Commutative encryption is a collection of algorithms that have the property of being commutative. In short, an encryption algorithm E(:) is commutative if for any two keys e1 ande2, Ee1 [Ee2 [m]] = Ee2 [Ee1 [m]], where m is the message to be encrypted. We adopt Pohlig-Hellman exponentiation cipher with modulus p as our commutative encryption function. We employ the commutative encryption algorithm in order to make flexible switching of decryption sequence possible.

The commutative-based encryption scheme introduces a commutative symmetric key for each level, namely commutative level key Ci. Besides being issued to brokers of level i, Ci is also issued to all the brokers at level Ci+2. The public and private level keys P1 and P2, as defined in level-based encryption scheme, are also defined and assumed to be commutative. Moreover, a pointer p is introduced to indicate the x path step to be processed by the current broker. A broker will always decrypt the x path step marked by the pointer with its private level key, and move the pointer to the next x path step. Commutative encryption scheme is an improver for the level-based encryption scheme. In general, the brokers process x path steps of a query in the same way as they do following the level-based scheme.

When encountering a step with wildcard "//", and the token in wildcard step does not match with the one in its routing segment, the broker will launch the special commutative encryption process, which is

summarized as follows: first, the broker starts the set wildcard processing stage by setting a flag f = 1. Then, it encrypts all the following unprocessed x path steps with its commutative level key. The x path step "/name" is encrypted with key C2 at the node "regions" of B2.

The following brokers compare the wildcard token with the one in its routing segment. If they do not match, the broker, says Bj, will apply two symmetric commutative keys onto all the x path steps in the query. This additional encryption guarantees every unprocessed x path segments are protected by the commutative keys of the current level and its upper level.

If they match with each other, Bj will set f = 2 to indicate post-wildcard processing. In this stage, a broker encrypts all the unprocessed steps with Cj and decrypts them with Cj2. B2 wraps unprocessed query step/name with the commutative level key C2 and C3. Then, when B8 _nds "//item" matches with the token in its routing segment, it decrypts /name with C2 andC3, accordingly.

These commutative encryption scheme reduces the two types of cost. They are as follows:

Computational cost:

Assume the cost of asymmetric encryption and decryption as C ae and C ad , respectively, and the cost of commutative encryption and decryption as C ce and C cd , respectively. Since Pohlig-Hellman method adopt C e and C d to denote the cost of encryption and decryption. Let C e at the user-side, asymmetric decryption cost of C d at each intermediate broker, and additional commutative encryption and decryption cost C e + C d at each broker when encountering query segment with wildcard. At each broker, the query segment is matched against the routing schema carried by the broker. This way the computation cost is reduced.

Communication cost:

Since the computational cost at each intermediate broker is at millisecond level, the overhead in end-to-end query brokering is mainly caused by the direct IP latency in overlay routing. The latency is measured using round trip time (RTT), where data from Internet traffic report shows average IP latency is about 200ms. The result is based on simulations with synthetical XML queries and XPath routing rules that are generated atop of the XML benchmark. The number of hops is exactly the depth of the query, providing that the query has no wildcard. When there exists a wildcard, the number of hops cannot be measured accurately.

Conclusion:

We have described an application of content-based information brokering to defend against attribute-correlation attacks. A commutative encryption based scheme is further designed to protect query content from irrelevant brokers. The main purpose of this work is to protect the privacy of the data owners while authorized organizations collect the data from them and share with other collaborators. More specifically, we protect the content of the query from the malicious or compromised intermediate servers during information brokering process. The privacy of query content is enhanced with the PPIB scheme.

REFERENCES

Back, A., I. Goldberg and A. Shostack, 2001. "Freedom 2.1 SecurityIssues and Analysis," Zero Knowledge Systems, Inc., White Paper.

Blum, A., D. Song and S. Venkataraman, 2004. "Detection of InteractiveStepping Stones: Algorithms and Confidence bounds," Proc.SeventhSymp. Recent Advances in Intrusion Detection (RAID).

Carzaniga, A., M.J. Rutherford and A.L. Wolf, 2004. "Arouting scheme forcontent-based networking," in *Proc. INFOCOM*, Hong Kong, pp: 918–928.

Chaum, D., 1981. "Untraceable Electronic Mail, Return Addresses, andDigital Pseudonyms," Comm. ACM, 24(2): 84-88.

Damiani, E., S. di Vimercati, S. Paraboschi and P. Samarati, 2000. "Securing{XML} documents," in *Proc. EDBT*, pp: 121–135.

Dingledine, R., N. Mathewson and P. Syverson, 2000. "Tor: The SecondGeneration Onion Router," Proc. 13th USENIX Security Symp.

Donoho, D.L., A.G. Flesia, U. Shankar, V. Paxson, J. Coit and S. Staniford, 2002. "Multiscale Stepping Stone Detection: Detecting Pairsof Jittered Interactive Streams by Exploiting Maximum TolerableDelay," Proc. Fifth Symp. Recent Advances in Intrusion Detection (RAID)

Eclipse 2010. Aspectj Compiler," http://eclipse.org/aspectj, 2010.

Li, F., B. Luo, P. Liu, D. Lee and C.H. Chu, 2007. "Automaton segmentation:A new approach to preserve privacy in XML information brokering,"in *Proc. ACM CCS'07*, pp: 508–518.

Li, F., B. Luo, P. Liu, D. Lee and C.H. Chu, 2013. " Enforcing Secure and Privacy-Preserving Information Brokering in Distributed Information Sharing," IEEE Transaction.

TelegeographyResearch, 2010. "http://www.telegeography.com.

Xiao, Y., B. Luo and D. Lee, 2007. "Security-conscious XML indexing,"Adv. Databases: Concepts, Syst., Applicat., 4443: 949–954.