Automata based Access Control Privacy Preserving in DIS

Jyothsna Pamala

M.Tech Student, Department of Computer Science and Engineering
AITS Tirupathi, Chittor, Andhra Pradesh, India

C. Usha Rani

Assistant Professor, Department of Computer Science Engineering
AITS Tirupathi, Chittor, Andhra Pradesh, India

Abstract- A Distributed Information Brokering System (DIBS) is a peer-to-peer overlay network that comprises various data servers and brokering components helping client queries locate the data servers. Many existing information brokering systems adopt server side access control deployment and honest assumptions on brokers. However, little attention has been drawn on privacy of data and metadata stored and exchanged within DIBS. In this paper, addresses privacy preserving information sharing via on-demand information access with a focus on two attacks: attribute-correlation attack and inference attack. It has been propose a flexible and scalable system using a broker-coordinator overlay network. Through an innovative automaton segmentation scheme, distributed access control enforcement, and query segment encryption, our system integrates security enforcement and query forwarding while preserving system-wide privacy. It presents the automaton segmentation approach, analyze privacy preservation in details, and finally examine the end-to-end performance and scalability through experiments and analysis.

Keywords: – Access control, information sharing, privacy, DIS (Distributed Information Sharing).

I. Introduction

In recent years, we have observed an explosion of information shared among organizations in many realms ranging from business to government agencies. To facilitate efficient large scale information sharing, many efforts have been devoted to reconcile data heterogeneity and provide interoperability across geographically distributed data sources. Meanwhile, peer autonomy and system coalition becomes a major tradeoff in designing such distributed information sharing systems.

Most of the existing systems work on two extremes of the spectrum: (1) in the query-answering model for on-demand information access, peers are fully autonomous but there is no system-wide coordination; so that participants create pair wise client-server connections for information sharing; (2) in the traditional distributed database systems, all the participates lost autonomy and are managed by a unified DBMS. Unfortunately, neither of them is suitable for many newly emerged applications, such as information sharing for healthcare or law enforcement, in which organizations share information in a conservative and controlled manner, not only from business considerations but also due to legal reasons.

The rest of the paper is organized as follows. Proposed IBS (Information Brokering System), proposed system discus in II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. Proposed Algorithm

2.1 INFORMATION BROKERING SYSTEM (IBS):

As shown in Figure 1, applications atop IBS always involve some sort of consortium (e.g. RHIO) among a set of organizations. Databases of different organizations are connected through a set of brokers, and metadata (e.g. data summary, server locations) are “pushed” to the local brokers, which further “advertise” (some of) the metadata to other brokers. Each query is sent to the local broker, and routed according to the metadata until reaching the right database(s). In this way, a large number of information sources in different organizations are loosely federated to provide an unified, transparent, and on-demand data access.

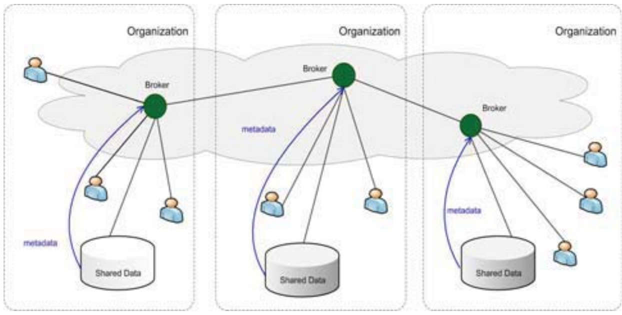

Fig1: An overview of the IBS infrastructure.

2.2 PRIVACY-PRESERVING QUERY BROKERING SCHEME:

While Q-Broker seamlessly integrates the content-based indexing function into the NFA-based access control mechanism, it heavily relies on the Q-Broker for the enforcement and shifts all the data (i.e., the ACR, index rules, and user queries) to it. However, if the Q-Broker is compromised or no longer assumed fully trusted (e.g. under the honest-but-curious assumption as in our study), the privacy of both the requestor and the data owner is under risk. To tackle the problem, we present a privacy-preserving information brokering (PPIB) infrastructure with two core schemes. The automata segmentation scheme divides the Q-Broker into multiple logically independent components so that each component only needs to process a piece of an user query but still can fulfill the original brokering functions via collaboration. The query segment encryption scheme allows to encrypt query pieces with different keys so that one automaton component can decrypt the responsible piece(s) for further processing, while not hurdling the original distributed indexing function.

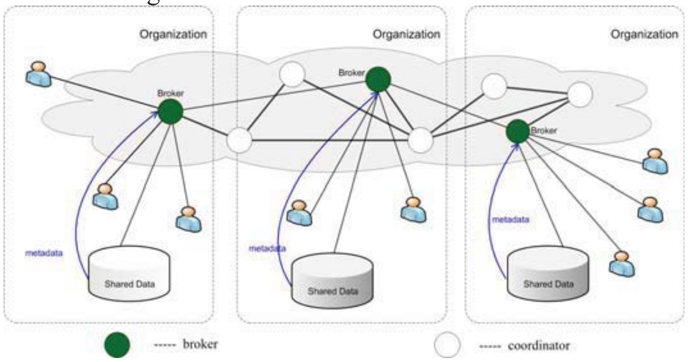

Fig2: The architecture of PPIB.

To address the privacy vulnerabilities in current information brokering infrastructure, we propose a new model, namely Privacy Preserving Information Brokering (PPIB). PPIB has three types of brokering components: brokers, coordinators, and a central authority (CA). The key to preserve privacy is to divide the work among multiple components in such a way that no single node can make a meaningful inference from the information disclosed to it.

While the IBS approach provides scalability and server autonomy, privacy concerns arise, as brokers are no longer assumed fully trustable – they may be abused by insiders or compromised by outsiders. In this article, we present a general solution to the privacy-preserving information sharing problem. First, to address the need for privacy protection, we propose a novel IBS, named Privacy Preserving Information Brokering (PPIB). It is an overlay infrastructure consisting of two types of brokering components: brokers and coordinators.

The brokers, acting as mix anonymizers, are mainly 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 curious or corrupted coordinators from inferring private information, we design two novel schemes: (a) to segment the query brokering automata, and (b) to encrypt corresponding query segments. While providing full capability to enforce in-network access control and to route queries to the right data sources, these two schemes ensure that a curious or corrupted coordinator is not capable to collect enough information to infer privacy, such as "which data is being queried", "where certain data is located", or "what are the access control policies", etc. We show that PPIB provides comprehensive privacy protection for on-demand information brokering, with insignificant overhead and very good scalability.

2.3 Algorithm : The automata based segmentation algorithm:
deploy Segment()

```
Input: Automaton State S
Output: Segment Address: addr
for each symbol k in S:StateT ransT able do
    addr=deploySegment(S:StateT ransT able(k):nextState)
    DS=createDummyAcceptState()
    DS:nextState   addr
    S:StateT ransT able(k):nextState   DS
end for
Seg = createSegment()
Seg:addSegment(S)
Coordinator = getCoordinator()
Coordinator:assignSegment(Seg)
return Coordinator:address
```

III. PRELIMINARIES

1) XML Data Model and Access Control: The extensible Markup Language (XML) has emerged as the de facto standard for information sharing due to its rich semantics and extensive expressiveness. We assume that all the data sources in PPIB exchange information in XML format, i.e. take XPath [30] queries and return XML data. Note that the more powerful XML query language, XQuery, still uses XPath to access XML nodes. In XPath, predicates are used to eliminate unwanted nodes, and test conditions are contained within square brackets "[ ]". In our study, we mainly focus on value-based predicates.

To specify the authorization at the node level, fine-grained access control models are desired. We adopt the 5-tuple access control policies that are used in the literature. The policy consists of a set of access control rules (ACR) = f subject, object, action, sign, type g, where (1) subject is the role to whom the authorization is granted; (2) object is a set of XML nodes specified by an XPath expression; (3) action is operations as "read", "write", or "update"; (4) sign 2 f+; fig refers to access "granted" or "denied", respectively; and (5) type 2 fLC;RCg denotes "local check" (i.e., applying authorization only to the attributes or textual data of the context nodes) or "recursive check" (i.e., applying authorization to all the descendants of the context node). A set of example rules are shown below:

Example 2. Example ACRs:

R1:frole1,/site//person/name, read, +, RCg
R2:frole1,/site/regions/asia/item, read, +, RCg
R3:frole2,/site/regions/asia/item, read, +, RCg
R4:frole2,/site/regions/*/item/name, read, +,
RCg
R5:frole2,/site/regions/*/item[location="USA"]
/description,read,+,RCg

Example 3. Example index rules:

I1:f/site/people/person/name, 130.203.189.2g
I2:f/site/regions//item[@id>"100"],
135.176.4.56g
I3:f/site/regions/samerica/item[@id>"200"],
195.228.155.9g
I4:f/site//namerica/item/name, 135.207.5.126g
I5:f/site/regions/namerica/item/location, 74.128.5.91g

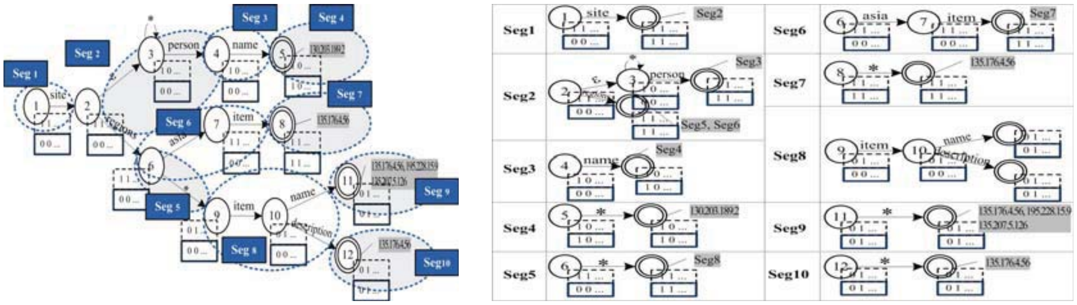

Fig 3: An example to illustrate the automaton segmentation scheme: (a) divide the global automaton with granularity level of 1; (b) the segments are linked to form a tree structure.

QBroker is an NFA constructed in a similar way as QFilter. Fig. 3 shows the data structure of each NFA state in QBroker, where the state transition table stores the child nodes specified by the XPath expression as the child states in eSymbol. The binary flag DSState indicates that the state is a “double-slash” state, which means it will recursively, accepts any input symbols, and the child state will be the transition state that directly transits to the next state without consuming any input symbol. Fig. 4 shows a QBroker constructed from Examples 2 and 3.

Different from QFilter, QBroker can capture ACRs for different roles by adding two binary arrays to each state: the Access List determines the roles that are allowed to access the state and the Accept List indicates for which roles the state is an accept state. For instance, in Fig. 4, the accept list of state 5 is [1 0], indicating the state is an accept state for role1 but not for role2, and the access list of state 6 is [1 1], indicating this state is accessible by both roles. A LocationList is attached to each accept state. In the brokering process, the QBroker first checks if a query is allowed to access the requested nodes according to the role type. If the query can access (a subset of) the requested data,it will be rewritten into a “safe” query and sent to the data servers according to the location list; otherwise, the query will be rejected.

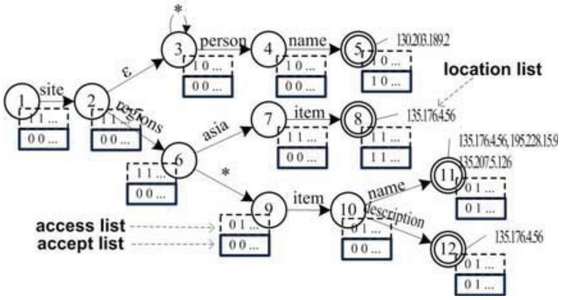

Fig. 4. The state transition graph of the QBroker that integrates index rules with ACRs.

IV. PERFORMANCE ANALYSIS

4.1 Average network transmission latency: We adopt average Internet traffic latency 100 ms as a reasonable estimation of TN 1, instead of using data collected from our gigabyte Ethernet.

4.2 Average number of hops: We consider the case in which a query Q is accepted or rewritten by n ACRs fR1……Rng into the union of n safe sub-queries fQ0 1…..Q0 ng. When an accepted/rewritten sub-query Q0 i is processed by the rule Ri, the number of hops it experiences is determined by the number of segments of Ri. In the experiment, we generate a set of 200 synthetic access control rules and 1000 synthetic XPath queries. It is obvious to see that the more segments the global automaton is divided into, the more coordinators are needed and the less scalable the system is, due to the increased query processing cost. However, higher granularity leads to better privacy preserving performance. We choose the finest granularity automaton segmentation (each XPath step of an ACR is partitioned as one segment and kept at one coordinator) for maximum privacy preserving. Our experiment result shows that NHOP is 5.7, and the maximum number of hops of all queries is 8.

Privacy Type	Local eaves dropper	Global eavesdropper	Malicious broker	Collusive Coordinator
User Location	Exposed	Exposed	Exposed	Protected
Query Content	Protected	Exposed	Exposed	Exposed only with compromised root coordinator
AC Policy	Protected	Protected	Protected	Exposed if path coordinators collude
Index Rules	Protected	Protected	Protected	Exposed if path coordinators collude
Data Distribution	Protected	Protected	Protected	Exposed if path coordinators collude
Data Location	Protected	Beyond Suspicion	Protected	Exposed with malicious leaf Coordinator

TABLE I - THE POSSIBLE PRIVACY EXPOSURE CAUSED BY FOUR TYPES OF ATTACKERS: LOCAL EAVESDROPPER (LE), GLOBAL EAVESDROPPER (GE), MALICIOUS BROKER (MB), AND COLLUSIVE COORDINATORS (CC).

4.3 Data size: When data volume increases (e.g. adding more data items into the online auction database), the number of indexing rules also increases. This results in increasing of the number of leaf-coordinators. However, in PPIB, query indexing is implemented through hash tables, which is scalable. Thus, the system is scalable when data size increases.

V.CONCLUSION

With little attention drawn on privacy of user, data, and metadata during the design stage, existing information brokering systems suffer from a spectrum of vulnerabilities associated with user privacy, data privacy, and metadata privacy. In this paper, we propose PPIB, a new approach to preserve privacy in XML information brokering. Through an innovative automaton segmentation scheme, in-network access control, and query segment encryption, PPIB integrates security enforcement and query forwarding while providing comprehensive privacy protection. Our analysis shows that it is very resistant to privacy attacks. End-to-end query processing performance and system scalability are also evaluated and the results show that PPIB is efficient and scalable.

Many directions are ahead for future research. First, at present, site distribution and load balancing in PPIB are conducted in an ad-hoc manner. Our next step of research is to design an automatic scheme that does dynamic site distribution. Several factors can be considered in the scheme such as the workload at each peer, trust level of each peer, and privacy conflicts between automaton segments. Designing a scheme that can strike a balance among these factors is a challenge. Second, we would like to quantify the level of privacy protection achieved by PPIB. Finally, we plan to minimize (or even eliminate) the participation of the administrator node, who decides such issues as automaton segmentation granularity. A main goal is to make PPIB self-reconfigurable.

REFERENCESS

[1] F. Li, B. Luo, P. Liu, D. Lee, P. Mitra, W. Lee, and C. Chu, “In-broker access control: Towards efficient end-to-end performance of information brokerage systems,” in Proc. IEEE SUTC, 2006.

[2] F. Li, B. Luo, P. Liu, D. Lee, and C.-H. Chu, “Automaton segmentation: A new approach to preserve privacy in XML information brokering,” in ACM CCS ’07, pp. 508–518, 2007.

[3] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” Communications of the ACM, vol. 24, no. 2, 1981.

[4] R. Agrawal, A. Evfimivski, and R. Srikant, “Information sharing across private databases,” in Proceedings of the 2003 ACM SIGMOD, 2003.

[5] M. Genesereth, A. Keller, and O. Duschka, “Informaster: An information integration system,” in SIGMOD, (Tucson), 1997.

[6] I. Manolescu, D. Florescu, and D. Kossmann, “Answering XML queries on heterogeneous data sources,” in VLDB, pp. 241–250, 2001.

[7] J. Kang and J. F. Naughton, “On schema matching with opaque column names and data values,” in SIGMOD, pp. 205–216, 2003.

[8] S. Durkin, “Surveying the RHIO landscape: A description of current RHIO models, with a focus on patient identification,” Journal of AHIMA 77, pp. 64A–D, January 2006.

[9] A. P. Sheth and J. A. Larson, “Federated database systems for managing distributed, heterogeneous, and autonomous databases,” ACM Computing Surveys (CSUR), vol. 22, no. 3, pp. 183–236, 1990.

[10] L. M. Haas, E. T. Lin, and M. A. Roth, “Data integration through database federation,” IBM Syst. J., vol. 41, no. 4, pp. 578–596, 2002.

[11] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, “CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming,” in Proceedings of IEEE INFOCOM, 2005.

[12] A. C. Snoeren, K. Conley, and D. K. Gifford, “Mesh-based content routing using XML,” in SOSP, pp. 160–173, 2001.

[13] N. Koudas, M. Rabinovich, D. Srivastava, and T. Yu, “Routing XML queries,” in ICDE ’04, p. 844, 2004.

[14] G. Koloniari and E. Pitoura, “Peer-to-peer management of XML data: issues and research challenges,” SIGMOD Rec., vol. 34, no. 2, 2005.