ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

Automata Based Access Control Privacy Preserving in PPIB

[1]Jyothsna Pamala, [2]C. Usha Rani
[1]M.Tech Student, Department of CSE, JNTUA, Anantapur/ Annamacharya Institute of Technology and Sciences, Tirupati, India
[2]Assistant Professor, Department of CSE, JNTUA, Anantapur/Annamacharya Institute of Technology and Sciences, Tirupati, India

ABSTRACT: At present many outsourcing organizations need sharing of the information via on-demand access. Information Brokering System (IBS) have been already proposed to connect large-scale loosely-federated data sources via brokering overlay. But there are some attacks with these traditional IBS, namely attribute-correlation attack and inference attack. To overcome these attacks here we are proposing two countermeasure schemes they are automaton segmentation, query segment encryption to safely share the routing among the selected set of brokering servers. Existing system work on two extremes of the spectrum by adopting either query-answering model or distributed database model. Our proposed novel IBS, Privacy Preserving Information Brokering (PPIB), is an overlay infrastructure consisting of two types. Firstly Brokers, acting as mix anonymizer, is mainly responsible for user authentication and query forwarding. Next the coordinators, concatenated in a tree structure enforce access control and query routing based on the embedded non-deterministic finite automata – the query brokering automata. IBS also ensures curious or corrupted coordinator is not capable to collect enough information to interfere the privacy.

KEYWORDS*:* Access control, information sharing, privacy, DIS (distribute Information Sharing), PPIB (Privacy Preserving Information Brokering).

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. 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.

Today's fast and continuous growth of large business organizations, often deriving from mergers of smaller enterprises, enforces an increasing need in integrating and sharing large amounts of data, coming from a number of heterogeneous and distributed data sources. Nonetheless, such needs are also shown by others application fields, like information systems for administrative organizations, life sciences research and many others. More, is not un frequent that different parts of the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

same organization, whatever it is, adopt different systems to produce and maintain its critical data. All these kinds of situations are concerned with the problem of information integration.

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.

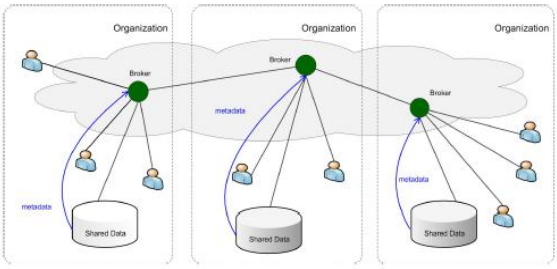

Fig. 1: An overview of the IBS infrastructure.

II. RELATED WORK

Research areas such as information integration, peer-to-peer file sharing systems and publish-subscribe systems provide partial solutions to the problem of large scale data sharing. Information integration approaches focus on providing an integrated view over large numbers of heterogeneous data sources by exploiting the semantic relationship between schemas of different sources. The PPIB study assumes that a global schema exists within the consortium, therefore, information integration is out of our scope.

Peer-to-peer systems are designed to share files and data sets (e.g. in collaborative science applications). Distributed hash table technology is adopted to locate replicas based on keyword queries. However, although such technology has recently been extended to support range queries, the coarse granularity (e.g. files and documents) still makes them short of our expressiveness needs. Further, P2P systems may not provide complete set of answers to a request while we need to locate all relevant data.

III. PROBLEM DEFINITION

Vulnerabilities and the Threat model:

In the information brokering infrastructure, privacy concerns arise when identifiable information is disseminated with no or poor disclosure control. Data providers push routing and access control metadata to brokers , which also handles queries from requestors. Therefore, a curious or corrupted brokering server could: (1) learn query content and query location by intercepting a local query; (2) learn routing metadata and access control metadata from local data servers and other brokers; (3) learn data location from routing metadata it holds. On the other hand, although eavesdroppers may not obtain plaintext data over encrypted communication, they can still learn query location and data location from eavesdropping. Moreover, with one or several types of such information, the attacker could further infer the privacy of different stakeholders. We classify the attacks into two major classes: the attribute-correlation attack and the inference attack.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

Attribute-correlation attack: An attacker intercepts a query (in plaintext), which typically contains several predicates. Each predicate describes a condition, which sometimes involves sensitive and private data (e.g. name, SSN or credit card number, etc.). If a query has multiple predicates or composite predicate expressions, the attacker can "correlate" the corresponding attributes to infer sensitive information about the data owner. This attack is known as the attribute correlation attack:

Inference attack: More severe privacy leak occurs when an attacker obtains more than one type of sensitive information, and further associates them to learn explicit and implicit knowledge about the stakeholders. By "implicit", we mean the attacker infers the fact by "guessing". For example, an attacker can guess the identity of a requestor from her query location.

IV. PROPOSED METHOD

The Overall PPIB Architecture:

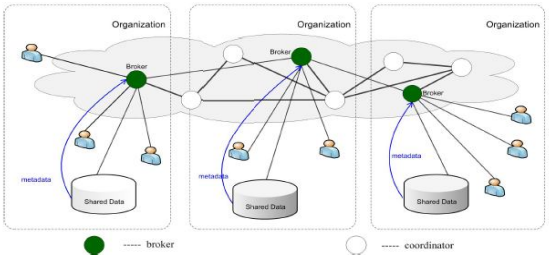

Fig.2: The architecture of PPIB.

The architecture of the privacy preserving information brokering system is shown in figure where users and data servers of multiple organizations are connected via a broker, coordinator overlay. User requests for remote data by sending a query to the local broker, which further forwards the query to the root of the coordinator tree. The query is processed along a path of the coordinator tree, until it is denied by any coordinator or accepted by a leaf coordinator. The accepted query is or rewritten into a safe query, and thus sent to the relevant data server(s). In particular, the brokering process consists of four phases:

- **Phase 1:** To join the system, a user needs to authenticate himself to the local broker. After that, the user submits an XML query with each segment encrypted by the corresponding public level keys, and a unique session key K_Q, encrypted with the public key of the data servers, for the data server to return data.

- **Phase 2:** Beside authentication, the major task of the broker is metadata preparation: (1) it extracts the role of the authenticated user and attaches it to the encrypted query; (2) it creates a unique ID for each query, and attaches QID with its own address (as well as $< K_Q >_{pkDS}$) to the query so that the data server can directly return the data.

- **Phase 3:** When the root of the coordinator tree receives the query and its metadata from a local broker, it follows the automata segmentation scheme and the query segment encryption scheme to perform access control and indexing to route the query within the coordinator tree, until it reaches a leaf coordinator, which further forwards the query to the related data servers. For any query that is denied access based on the ACRs, a failure message will be returned to the broker with QID. At the leaf coordinator, all the query segments should be processed and encrypted with the public key of the data server.

- **Phase 4:** In the final phase, the data server receives a safe query in an encrypted form. After decryption, the data server evaluates the query and returns the data, encrypted by KQ, to the broker of the query.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

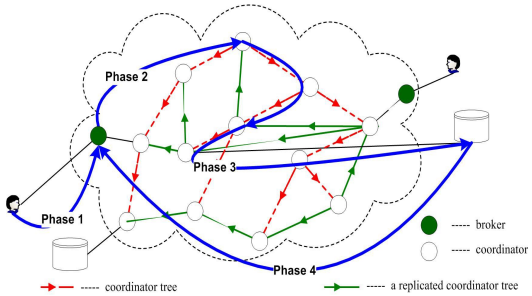

Fig.3. Query brokering process in four phases.

Existing Algorithm:

Algorithm1: The automaton segmentation algorithm:

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

Input: Automaton State S
Output: Segment Address: addr

Proposed Algorithm:

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits fewer than a multiple of 512. The remaining bits are filled up with 64 bits representing the length of the original message, modulo 2.

F (B, C, D) = (B ΛC) V (¬B ΛD)
G (B, C, D) = (B Λ D) V (¬D Λ C)
H (B, C, D) = B⊕C⊕D
I (B, C, D) = C⊕ (B V ¬D)
Here mentioned ¬, Λ, V, ⊕ Denote the NOT, AND, OR and XOR operations respectively.
.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

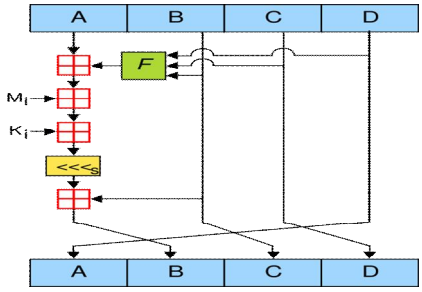

Fig 4. One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. *F* is a nonlinear function; one function is used in each round. M_i denotes a 32-bit block of the message input, and K_i denotes a 32-bit constant, different for each operation. $<<<_s$ denotes a left bit rotation by *s* places; *s* varies for each operation. ⊞ denotes addition modulo 2^{32}.

The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted *A*, *B*, *C* and *D*. These are initialized to certain fixed constants. The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of four similar stages, termed *rounds*; each round is composed of 16 similar operations based on a non-linear function *F*, modular addition, and left rotation. Figure 1 illustrates one operation within a round. There are four possible functions *F*; a different one is used in each round:

V.PERFOMANCE ANALYSIS:

Here, we are analyzing the performance of proposed PPIB system using end-to-end query processing time and system scalability. In our proceedings, coordinators are coded in Java (JDK 5.0) and results are collected from coordinators running on a Windows desktop (3.4G CPU). XMark is being used , XML document and DTD, which is widely used in the research community.

A. End-to-End Query Processing Time:

End-to-end query processing time is defined as the time elapsed from the point when query arrives at the broker until to the point when safe answers are returned to the user. We consider the following four components: (1) average query brokering time at each broker/coordinator (TC); (2) average network transmission latency between broker/coordinators (TN); (3) average query evaluation time at data server(s) (TE); and (4) average backward data transmission latency (T backward).

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
Data location	Protected	Beyond suspicion	Protected	Exposed with malicious leaf coordinator

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

Data Distribution	Protected	Protected	Protected	Exposed if path coordinators collude
Index Rules	Protected	Protected	Protected	Exposed if path coordinators collude

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).

Query evaluation time highly depends on XML databases system, size of XML documents, and types of XML queries.

B. System Scalability:

PPIB scalability is being evaluated against complicity of ACR, the number of user queries, and data size (number of data objects and data servers).1) Complicity of XML schema and ACR: When the segmentation scheme is determined, the demand of coordinators is determined by the number of ACR segments, which is linear with the number of access control rules.

Assume finest granularity automaton segmentation is adopted, we can see that the increase of demanded number of coordinators is linear or even better, as shown in Figure. This is because similar access control rules with same prefix may share XPath steps, and save the number of coordinators. Moreover, different ACR segments (or, logical coordinators) may accommodate at the same physical site, thus reducing the actual demand of physical sites. In this framework, the number of coordinators, m, and the height of the coordinator tree, h, is highly dependent on access control policies segmentation. 2) Number of queries: Considering n queries submitted into the system in a unit time, the total numbers of query segments being processed in the system to measure the system load are used. When a query is accepted as multiple sub-queries, all sub-queries are counted towards system load. For a query that is rejected after i segments, the processed i segments are counted.

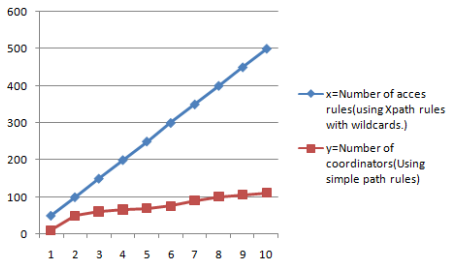

Fig 5: System scalability: number of coordinators.

VI.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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 4, September 2014

results show that PPIB is efficient and scalable. Processing performance and system scalability are also evaluated and the results show that PPIB is efficient and scalable.

REFERENCES

1. A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," ACM ComputingSurveys (CSUR), vol. 22, no. 3, pp. 183–236, 1990.
[2] 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.
[3] 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.
[4] A. C. Snoeren, K. Conley, and D. K. Gifford, "Mesh-based content routing using XML," in SOSP, pp. 160–173, 2001.
[5] N. Koudas, M. Rabinovich, D. Srivastava, and T. Yu, "Routing XML queries," in ICDE '04, p. 844, 2004.
[6] G. Koloniari and E. Pitoura, "Peer-to-peer management of XML data: issues and research challenges," SIGMOD Rec., vol. 34, no. 2, 2005.
[7] M. Franklin, A. Halevy, and D. Maier, "From databases to dataspaces: a new abstraction for information management," SIGMOD Rec., vol. 34, no. 4, pp. 27–33, 2005.