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 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

An Architecture to Enhance Privacy and Security in Collaborative Systems

Ms. G. Sathya M.E. [1], Mr. K. Loganathan M.E., [2]

Department of CSE, Maharaja Prithvi Engineering College, Avinashi, Tamilnadu, India[1]

Assistant Professor, Department of CSE, Maharaja Prithvi Engineering College, Avinashi, Tamilnadu, India[2]

ABSTRACT: Information Brokering System is a distributed system, which provides data access for the clients through a set of brokers. Brokers make routing decisions to direct client queries to the requested data servers. Many IBSs assume that brokers are trusted and provide only data confidentiality for server-side. The privacy of data location and data consumer can be inferred from metadata exchanged among the brokers. The corrupted broker may outsource the metadata to third parties. A new infrastructure called PPIB is developed to preserve the privacy of multiple stakeholders involved in brokering process. Two schemes are defined namely, Automaton Segmentation and Query Segment Encryption to securely share the routing decision among a set of brokers. The key idea for preserving privacy is to divide and allocate the functionality to multiple brokering components. The functionalities are allocated in such a way that no single component makes a meaningful inference from the disclosed information. Another key idea is to develop a scheme that does dynamic load balance for co-ordinators, which is to make PPIB self-reconfigurable. The new model integrates security enforcement with query routing to provide system wide security with insignificant overhead.

KEYWORDS: Access control, inference, information sharing, privacy, automaton segmentation

I. INTRODUCTION

With the explosion of information collected by organizations in many realms ranging from business to government agencies, there is an increasing need for inter organizational information sharing to facilitate extensive collaboration. While many efforts have been devoted to reconcile data heterogeneity and provide interoperability, the problem of balancing peer autonomy and system coalition is still challenging. Most of the existing systems work on two extremes of the spectrum adopting either the query-answering model to establish pairwise client-server connections for on-demand information access, where peers are fully autonomous but there lacks system wide coordination, or the distributed database model, where all peers with little autonomy are managed by a unified DBMS.

Unfortunately, neither model is suitable for many newly emerged applications, such as healthcare or law enforcement information sharing, in which organizations share information in a conservative and controlled manner due to business considerations or legal reasons. Regional Health Information Organization (RHIO) aims to facilitate access to and retrieval of clinical data across collaborative healthcare providers that include a number of regional hospitals, outpatient clinics, payers, etc. As a data provider, a participating organization would not assume free or complete sharing with others, since its data is legally private or commercially proprietary, or both. Instead, it requires retaining full control over the data and the access to data. Meanwhile, as a consumer, a healthcare provider requesting data from other providers expects to preserve privacy in the querying process.

Sharing a complete copy of the data with others or "pouring" data into a centralized repository becomes impractical. To address the need for autonomy, federated database technology has been used to manage locally stored data with a federated DBMS and provide unified data access. However, the centralized DBMS still introduces data heterogeneity, privacy, and trust issues. While being considered a solution between "sharing nothing" and "sharing everything", peer-to-peer information sharing framework essentially need to establish pair wise client-server relationships between each pair of peers, which is not scalable in large scale collaborative system sharing.

A more practical and adaptable solution is to construct a data-centric overlay consisting of data sources and a set of brokers that make routing decisions based on the content of the queries. Such infrastructure builds up semantic-aware index mechanisms to route the queries based on their content, which allows users to submit queries without knowing data or server location. Such a distributed system providing data access through a set of brokers is referred to as Information Brokering System (IBS). Figure 1 shows databases of different organizations are connected through a set of brokers and metadata from each server is sent to each broker. The local broker advertises metadata to other brokers. Query from user is sent to local broker and routed according to metadata until it reaches the right server.

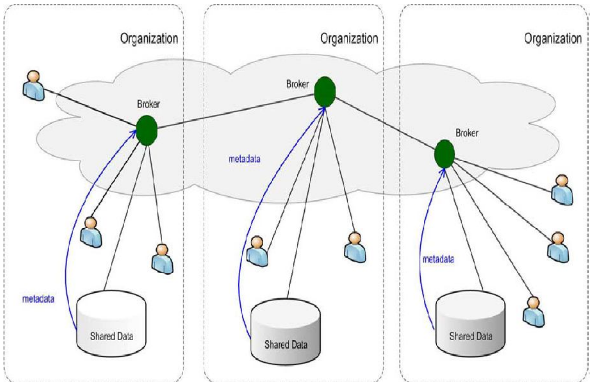

Fig 1 Overview of IBS Infrastructure

While the IBS approach provides scalability and server autonomy, privacy concerns arise, as brokers are no longer assumed fully trustable. The broker functionality may be outsourced to third-party providers and thus vulnerable to be abused by insiders or compromised by outsiders.

To address the need for privacy protection, we propose a novel IBS, namely 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 anonymizer 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 query brokering automata. To prevent curious or corrupted coordinators from inferring private information, we design two novel schemes to segment the query brokering automata and encrypt corresponding query segments so that routing decision making is decoupled into multiple correlated tasks for a set of collaborative coordinators. While providing integrated in-network access control and content-based query routing, the proposed IBS also ensures 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. Experimental results show that PPIB provides comprehensive privacy protection for on-demand information brokering, with insignificant overhead and very good scalability.

II. VULNERABILITIES & THREATS

In a typical information brokering scenario, there are three types of stakeholders, namely data owners, data providers, and data requestors. Each stakeholder has its own privacy. The privacy of a data owner is the identifiable data and sensitive or personal information carried by this data. Data owners usually sign strict privacy agreements with data providers to prevent unauthorized use or disclosure. Data providers store the collected data locally and create two types of metadata, namely routing metadata and access control metadata, for data brokering. Both types of metadata are considered privacy of a data provider. Data requestors may reveal identifiable or private information in the querying content.

ATTRIBUTE-CORRELATIN ATTACK: Predicates of an XML query describe conditions that often carry sensitive and private data. If an attacker intercepts a query with multiple predicates or composite predicate expressions, the attacker can "correlate" the attributes in the predicates to infer sensitive information about data owner.

INFERRENCE ATTACK: More severe privacy leak occurs when an attacker obtains more than one type of sensitive information and learns explicit or implicit knowledge about the stakeholders through association. 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. Meanwhile, the identity of the data owner could be explicitly learned from query content. Attackers can also obtain publicly-available information to help his inference. For example, if an attacker identifies that a data server is located at a cancer research center, he can tag the queries as "cancer-related".

Three reasonable inferences from three distinct combinations of private information: (1) from query location& data location, the attacker infers about who (i.e., a specific requestor) is interested in what (i.e., a specific type of data). (2) From query location & query content, the attacker infers about where who is, or who is interested in what (if predicates describe symptom or medicine, etc.), or *something* about the data owner (if predicate identifies name or address of a personnel) etc. (3) From query content & data location, the attacker infers which data server has which data.

III. SOLUTION

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. The key to preserving privacy is to divide and allocate the functionality to multiple brokering components in a way that no single component can make a meaningful inference from the information disclosed to it.

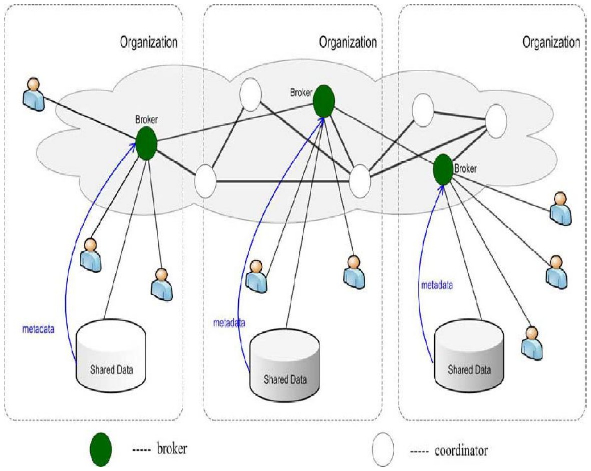

Fig 2 Architecture of PPIB

IV. AUTOMATON SEGMENTATION

In the context of distributed information brokering, multiple organizations join a consortium and agree to share the data within the consortium. While different organizations may have different schemas, we assume a global schema exists by aligning and merging the local schemas. Thus, the access control rules and index rules for all the organizations can be crafted following the same shared schema and captured by a global automaton. The key idea of automaton segmentation scheme is to logically divide the global automaton into multiple independent yet connected segments, and physically distribute the segments onto different brokering components, known as coordinators.

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. We further define the granularity level to denote the greatest distance between any two NFA states contained in one segment. Given a granularity level, k for each segmentation, the next states will be divided into one segment with a probability. Obviously, with a larger granularity level, each segment will contain more NFA states, resulting in less segments and smaller end-to-end overhead in distributed query processing. However, a coarse partition is more likely to increase the privacy risk. The trade-off between the processing complexity and the degree of privacy should be considered in deciding the granularity level. As privacy protection is of the primary concern of this work, granularity level should be less than 2. To reserve the logical connection between the segments after segmentation, we define the following heuristic segmentation rules: (1) NFA states in the same segment should be connected via parent-child links; (2) sibling without their parent state; and (3) the "accept state" of the original global automaton should be put in separate segments. To ensure the segments are logically connected, we also make the last states of each segment as "dummy" accept states, with links pointing to the segments holding the child states of the original global automaton.

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. For the ease of presentation, we assume each coordinator only holds one segment in the rest of the article. After the deployment, the coordinators can be linked together according to the relative position of the segments they store, and thus form a tree structure. The coordinator holding the root state of the

global automaton is the root of the coordinator tree and the coordinators holding the accept states are the leaf nodes. Queries are processed along the paths of the coordinator tree in a similar way as they are processed by the global automaton: starting from the root coordinator, the first XPath step (token) of the query is compared with the tokens in the root coordinator. If matched, the query will be sent to the next coordinator, and so on so forth, until it is accepted by a leaf coordinator and then forwarded to the data server specified by the outpointing link of the leaf coordinator. At any coordinator, if the input XPath step does not match the stored tokens, the query will be denied and dropped immediately.

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.

V. QUERY SEGMENT ENCRYPTION

Informative hints can be learned from query content, so it is critical to hide the query from irrelevant brokering servers. However, in traditional brokering approaches, it is difficult, if not impossible, to do that, since brokering servers need to view query content to fulfill access control and query routing. Fortunately, the automaton segmentation scheme provides new opportunities to encrypt the query in pieces and only allows a coordinator to decrypt the pieces it is supposed to process. The query segment encryption scheme proposed in this work consists of the preencryption and post encryption modules.

XML schema also forms a tree structure, in which the level of a node in the schema tree is defined as its distance to the root node. Since both the ACR and index rules are constructed following the global schema, an XPath step (token) in the XPath expression of a rule is associated with level if the corresponding node in the schema tree is at level. After automaton segmentation, the segments (and the corresponding coordinators) are assigned with the private key of level, if it contains a node of level. In preencryption, of a query are encrypted with the public level keys, respectively. Intuitively, the XPath step of a query should be processed by a segment with a node at level, and therefore, is able to be decrypted by the coordinator holding that segment. Moreover, if a coordinator has a segment that contains XML nodes of different levels, it needs to decrypt the first unprocessed XPath steps of the query.

The processed query segments should also be protected from the remaining coordinators in later processing, so post encryption is necessary. In a simple scheme, we assume all the data servers share a pair of public and private keys, where public key is known to all the coordinators. Each coordinator first decrypts the query segment(s) with its private level key, performs authorization and indexing, and then encrypts the processed segment(s) with so that only the data servers can view it.

VI. THE OVERALL PPIB ARCHITECTURE

The architecture of PPIB contains users and data servers of multiple organizations are connected via broker-coordinator overlay. 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 is encrypted with the public key of the data servers to encrypt the reply data.

PHASE 2: Besides authentication, the major task of the broker is metadata preparation: (1) it retrieves the of the authenticated user to attach to the encrypted query (2) it creates a unique for each query, and attaches and its own address to the query for data servers to return data.

PHASE 3: Upon receiving the encrypted query, the coordinators follow automata segmentation scheme and query segment encryption scheme to perform access control and query routing along the coordinator tree. At the leaf coordinator, all query segments should be processed and reencrypted by the public key of the data server. If a query is denied access, a failure message with will be returned to the broker.

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, to the broker that originates the query.

VII. MAINTENANCE OF KEY

The CA is assumed for offline initiation and maintenance. With the highest level of trust, the CA holds a global view about all the rules and plays a critical role in automaton segmentation and key management. There are four types of keys used in the brokering process, query session key, public/private level keys, commutative level keys, and public/private data server keys. Except the query session keys created by the user, the other three types of keys are generated and maintained by the CA. The data servers are treated as a unique party and share a pair of public and private keys, while each of the coordinators has its own pairs of level key and commutative level key. Along with the automaton segmentation and deployment process, the CA creates key pairs for coordinators at each level and assigns the private keys with the segments.

VIII. BROKERS JOIN AND LEAVE

Brokers and coordinators, contributed by different organizations, are allowed to dynamically join or leave the PPIB system. Besides authentication, a local broker only works as an entrance to the coordinator overly. It stores the address of the root coordinator (and its replica) for forwarding the queries. When a new broker joins the system, it registers to the CA to receive the current address list from the CA and broadcasts its own address to the local users. When leaving the system, a broker only needs to broadcast a leave message to the local users. Thing are more complicate for the coordinators. Once joining the system, a new coordinator sends a join request to the CA. The CA authenticates its identity, and assigns automaton segments to it considering both the load balance requirement and its trust level. After that, the CA issues the corresponding private level keys and sends a broadcast message to update the location list attached to the parent coordinator with the address of the newly joined coordinator. When a coordinator leaves the system, the CA decides whether to employ an existing or a new coordinator as a replacement, based on the heuristic rules for automaton deployment and the current load at each coordinator. After that, the CA broadcasts a message to replace the address of the old coordinator with the address of the new one in the location list at the dummy accept state of the parent coordinator. Finally, the CA revokes the corresponding level keys. If a failure is detected from a periodical status check by the CA or reported by a neighboring coordinator, the CA will treat the failed coordinator as a leaving server.

IX. ANALYSIS OF SECURITY

There are various types of attackers in the information brokering process. From their roles, they are classified as abused insiders and malicious outsiders; from their capabilities, they are classified as passive eavesdroppers and active attackers that can compromise any brokering server, from the cooperation mode, they are classified as and collusive attackers.

1) EAVESDROPPERS*:* A local eavesdropper is an attacker who can observe all communication to and from the user side. Once an end user initiates an inquire or receives requested data, the local eavesdropper can seize the outgoing and incoming packets.

A global eavesdropper is an attacker who observes the traffic in the entire network. It watches brokers and coordinators gossip, so it is capable to infer the locations of local brokers and root-coordinators. This is because the assurance of the connections between user and broker, and between broker and root-coordinator. Therefore, the major threat from a global eavesdropper is the disclosure of broker and root-coordinator location, which makes them targets of further Do's attack.

2) SINGLE MALICIOUS BROKER*:* A malicious broker deviates from the prescribed protocol and discloses sensitive information. It is obvious that a corrupted broker endangers user location privacy but not the privacy of query content. Moreover, since the broker knows the root-coordinator locations, the threat is the disclosure of root-coordinator location and potential Do's attacks.

3) COLLUSIVE COODINATORS: Collusive coordinators deviate from the prescribed protocol and disclose sensitive information. Consider a set of collusive (corrupted) coordinators in the coordinator tree framework. Even though each coordinator can observe traffic on a path routed through it, nothing will be exposed to a single coordinator because (1) the sender viewable to it is always a brokering component (2) the content of the query is incomplete due to query segment encryption (3) the ACR and indexing information are also incomplete due to automaton segmentation.

X. 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. 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. The proposed method is resistant to attacks. End-to-end query processing performance and system scalability are also evaluated and the results show that PPIB is efficient and scalable.

REFERENCES

[1] W. Bartschat, J. Burrington-Brown, S. Carey, J. Chen, S.Deming, and S. Durkin, "Surveying the RHIO landscape: A description of current{RHIO} models, with a focus on patient identification," *J. AHIMA*, vol. 77, pp. 64A–64D, Jan. 2006.
[2] A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM Comput. Surveys (CSUR)*, vol. 22, no. 3, pp. 183–236, 1990.
[3] L. M.Haas, E. T. Lin, andM.A. Roth, "Data integration through database federation," *IBM Syst. J.*, vol. 41, no. 4, pp. 578–596, 2002.
[4] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming," in *Proc. IEEE INFOCOM*,Miami, FL, USA, 2005, vol. 3, pp. 2102–2111.
[5] A. C. Snoeren, K. Conley, and D. K. Gifford, "Mesh-based content routing using XML," in *Proc. SOSP*, 2001, pp. 160–173.
[6] N. Koudas, M. Rabinovich, D. Srivastava, and T. Yu, "Routing XML queries," in *Proc. ICDE'04*, 2004, p. 844.
[7] G. Koloniari and E. Pitoura, "Peer-to-peer management of XML data: Issues and research challenges," *SIGMOD Rec.*, vol. 34, no. 2, pp. 6–17, 2005.
[8] 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.