International Journal of Power Control and Computation(IJPCSC)
Vol 6. No.1 – Jan-March 2014 Pp. 49-55
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-268X

A NOVEL APPROACH FOR DEFENDING AGAINST PRIVACY ATTACKS IN DISTRIBUTED FILE SHARING

V.Aiswarya , P.R.Jayanthi
Mailam Engineering College
Mailam, Villupuram District
aiswaryav90@gmail. comprjayanthi@gmail.com

ABSTRACT

Information sharing is becoming important in all organizations especially organizations that are more globalized and distributed. Information brokering systems enable information sharing through set of brokers. Most of the existing IBSs provide only server side access control and hence the privacy of data location and consumer can be compromised. A novel approach to preserve privacy of multiple stakeholders involved in the information brokering process has been defined. Two privacy attacks, namely attribute-correlation attack and inference attack, and two countermeasure schemes automaton segmentationand query segment encryptionto securely share the routing decision-making responsibility among a selected set of brokering servers has been defined. Automaton segmentation approach, analyzes privacy preservation in details, and finally examines the end-to-end performance through experiments and analysis.

IndexTerms—Privacy, Access control

1. INTRODUCTION

The explosive growth of the “Information Highway” has radically transformed the norms of information processing. Growing number of business organizations, government institutions, academics, students and a variety of users are now embracing the information highway as an apt medium for electronic, information-centered communication. In the background of such technological advances, now, it remains of interest to explore possibilities for transpiring intelligence’ to typical internet-oriented operations. One possible area that demands research attention is the extraction of information from remote databases, via the internet, by utilizing ‘intelligent’ internet-based database navigation mechanisms.

Indeed, today the need to have access to information that is both correct and complete is very real. For strategic reasons, such demands disregard geographical and time constraints information/data should be available from any database site in the world and furthermore the required data should be found and made available with the shortest possible time-delay, as and when required.

To address such demands the research proposal, then, entails dealing with two key technologies (a) the internet and (b) database technology. The explosive growth of the “Information Highway” has radically transformed the norms of information processing. Growing number of business organizations, government institutions, academics, students and a variety of users are now embracing the information highway as an apt medium for electronic, information-centered communication.

Advances in internet based technologies have generated tremendous opportunities for information-sharing by making available not only a plethora of applications, software and document archives, but also providing access to numerous informed people from various domains.

In the background of such technological advances, now, it remains of interest to explore possibilities for transpiring ‘intelligence’ to typical internet-oriented operations. One possible area that demands research attention is the extraction of information from remote databases, via the internet, by utilizing ‘intelligent’ internet-based database navigation mechanisms.

2. THE PROBLEM

A. *Problem Description*

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: (1) theprivacy of a data 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. (2) 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. (3) Data requestors may reveal identifiable or private information in the querying content.

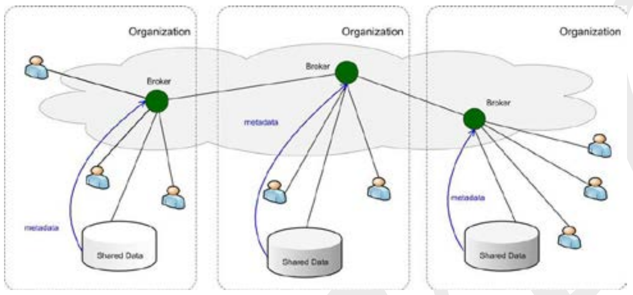

Figure. 1. Overview of the IBS infrastructure.

We adopt the *semi-honest* assumption for the brokers, and assume two types of adversaries, *external attackers* and *curious or corrupted brokering components*. External attackers passively eavesdrop communication channels.

Privacy concerns arise when identifiable information is disseminated with no or poor disclosure control. Existing security mechanisms focusing on confidentiality and integrity cannot preserve privacy effectively. For instance, while data is protected over encrypted communication, external attackers still learn *querylocation* and *data location* from eavesdropping. Combiningtypes of unintentionally disclosed information, the attacker could further infer the privacy of different stakeholders through *attribute-correlation attacks* and *inference attacks*.

Attribute-correlation attack. Predicates of an XML querydescribe conditions that often carry sensitive and private data (e.g., name, SSN, credit card number, etc.) 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. This is known as the *attribute correlation attack*.

Inference attack. More severe privacy leak occurs when anattacker 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". Meanwhile, the identity of the data owner could be explicitly learned from query content (e.g., name or SSN). Attackers can also obtain publicly-available information to help his inference. There are three reasonable inferences from three distinct combinations of private information: (1) from *query location & data location*, the attacker infers about *who* (i.e., aspecific 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. Hence, the attacker could continuously create artificial queries or monitor user queries to learn the data distribution of the system, which could be used to conduct further attacks.

B.*Solution*

To address the privacy vulnerabilities in current information brokering infrastructure, we propose a new model, namely *Privacy Preserving Information Brokering* (PPIB). PPIB has threetypes of brokering components: *brokers*, *coordinators*, and a *central authority* (CA). 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 shows the architecture of PPIB. Data servers and requestors from different organizations connect to the system through local brokers. A local broker functions as the "entrance" to the system. It authenticates the requestor and hides his identity from other PPIB components. It would also permute query sequence to defend against local traffic analysis.

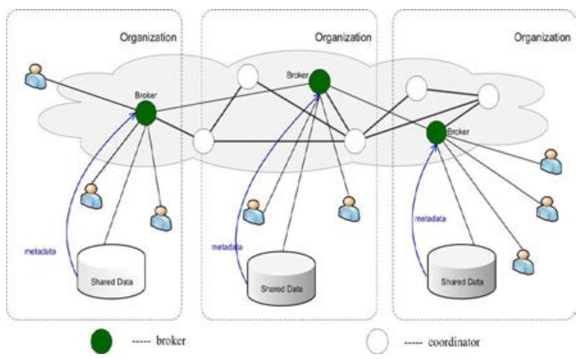

Figure. 2. Architecture of PPIB.

Coordinators are responsible for content-based query routing and access control enforcement. With privacy- preserving considerations, we cannot let a coordinator hold any rule in the complete form. Instead, we propose a novel *automaton segmentation scheme* to divide (metadata) rules into segments and as-sign each segment to a coordinator. Coordinators operate collaboratively to enforce secure query routing. A *query segmentencryption scheme* is further proposed to prevent coordinatorsfrom seeing sensitive predicates. The scheme divides a query into segments, and encrypts each segment in a way that to each coordinator enroute only the segments that are needed for se-cure routing are revealed. Last but not least, we assume a separate *central authority* handles key management and metadata maintenance.

3. PRIVACY PRESERVING QUERY BROKERING SCHEME

The QBroker approach has severe privacy vulnerability as we discussed in Section II. If the QBroker is compromised or cannot be fully trusted (e.g., under the honest-but-curious assumption as in our study), the privacy of both requestor and data owner is under risk. To tackle the problem, we present the PPIB infrastructure with two core schemes. In this section, we first explain the details of *automata segmentation* and *query segment encryption* schemes, and then describe the 4-phase query brokering process in PPIB.

A. Automaton Segmentation

In the context of distributed information brokering, multiple organizations join a consortium and agree to share the data within the consortium. 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 i(∈[1,k]) states will be divided into one segment with a probability 1/k. 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, 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 NFA states should not be put in the same segment 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.

Algorithm The automaton segmentation algorithm:

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

2) Deployment: We employ physical brokering servers, called*coordinators*, to store the logical segments. The coordinator holding the root state of the global

International Journal of Power Control and Computation(IJPCSC)
Vol 6. No.1 – Jan-March 2014 Pp. 49-55
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-268X

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. We adopt the passive path replication strategy to create the replicas for the coordinators along the paths in the coordinator tree, and let the *centralized authority* to create or revoke the replicas .The CA maintains a set of replicas for each coordinator, where the number of replicas is either a preset value or dynamically adjusted based on the average queries passing through that coordinator.

4) Handling the Predicates: In the original construction of NFA (similarly as described in QFilter and QBroker), a predicate table is attached to every child state of an NFA state. The predicate table stores predicate symbols (i.e., pSymbol), if any, in the corresponding query XPath step. An empty symbol means no predicate. To handle the predicates, either from the query or from the ACR, the original strategy is *lookup-and-attach.* The real evaluation of the predicate is left to the data servers, which inevitably causes unnecessary communication and processing overhead if the predicate conditions conflict. To address this problem, we present a new scheme to handle value-based predicates in input XML queries. We first change the data structure of the original NFA state by adding new fields as condition type and location to the predicate table:(1)pSymbol still stores the predicate token; (2)condition stores the test condition; (3) type ∈ [R,I] indicates if the predicate is introduced from an ACR or an index rule, and (4)location stores the addresses carried by the index rule.In processing, if the XPath step of a query does not have a predicate, the scheme works the same as before: it looks up the predicate table for predicates introduced by ACRs (i.e., with type = R and attaches "pSymbol || condition" to the safe query. If a predicate exists in a particular XPath step, the scheme retrieves records of the same predicate from the table, and sends them with the query predicate to a *predicate directory server*, which further examines the test conditions. Accordingly, in query processing, if an accepted query carries multiple location lists, it will be sent to the intersection of the destination data servers.

B. 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 *postencryption* modules, and a special *commutative encryption* module for processing the double-slash ("//") XPath step in the query.

1) Level-Based Preencryption: According to the automaton segmentation scheme, query segments are processed by a set of coordinators along a path in the coordinator tree. A straightforward way is to encrypt each query segment with the public key of the coordinator specified by the scheme. Hence, each coordinator only sees a small portion of the query that is not enough for inference, but collaborating together, they can still fulfill the designed function. The key challenges in this approach is that the segment-coordinator association is unknown beforehand in the distributed setting, since no party other than the CA knows how the global automaton is segmented and distributed among the coordinators. 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. We assume the nodes of the same level share a pair of public and private level keys,{pk,sk}. In preencryption, the XPath steps (between two "/"or "//") of a query are encrypted with the public level keys {pk_1, pk_2 ....} respectively. Intuitively, the Path 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.

2) Postencryption: The processed query segments should also be protected from the remaining coordinators in later processing, so postencryption is necessary. In a simple scheme, we assume all the data servers share a pair of public and private keys,{pk_{DS}, sk_{DS}} where pk_{DS} 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 pk_{DS} so that only the data servers can view it.

3) Commutative Encryption for "//" Handling: When a query has the *descendant-or-self* axis (i.e., "//" in XPath expressions), a so-called *mismatching problem* occurs at the coordinator who takes the "//" XPath step as input. This is because that the "//" XPath step may recursively accepts several tokens until it finds a match. Consequently, the coordinator with the private level key may not be the one that matches the "//" token, and vice versa.. To tackle the problem, we revise the level-based encryption scheme by adopting the commutative encryption. Commutative encryption algorithms have the property of being *commutative*, where an encryption algorithm is commutative if for any two commutative keys e_1 and e_2 and a message m,<<m> e_1> e_2=. ,<<m> e_2> e_1Therefore,we assign a new *commutative level key* to nodes at level ,and further assume nodes at level share with nodes at level i+2.

C. The Overall PPIB Architecture

The architecture of PPIB is shown in Fig. 3, where users and data servers of multiple organizations are connected via a 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 K_Q. K_Q
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 role ofthe authenticated user to attach to the encrypted query;(2) it creates a unique Q_{ID} for each query, and attaches $Q_{ID,}$< K_Q>PK_{DS} and its own address to the query for data servers to return data.

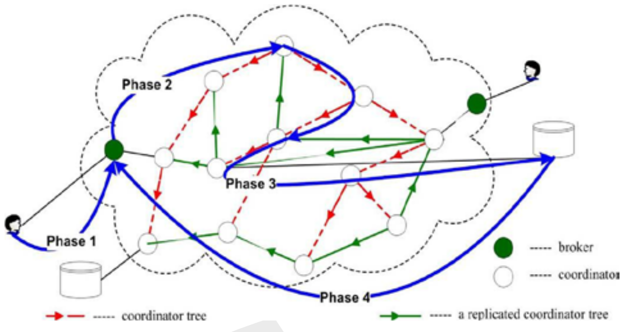

Figure. 3. The query brokering process in four phases.

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 thepublic key of the data server. If a query is denied access, a failure message with Q_{ID} 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 K_Q to the broker that originates the query.

4. MAINTENANCE

A. Key Management

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. 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.

B. Brokering Servers Join/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 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. 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.

C. Metadata Update

ACR and index rules should be updated to reflect the changes in the access control policy or the data distribution in an organization.*1) Index Rules:* To add or remove a data object, a local server need to send an update message, in the form of, to the CA, where *object* is an XPath expression to describe a set of XML nodes, *address* is the location of the data object, and *action* is either “add” or “remove”. For adding a data object, the CA sends the update message to the root coordinator, from which the message traverses the coordinator network until reaching a leaf coordinator, where the *address* will be appended to its location list. A similar process is taken for data object removal to retrieve the corresponding leaf coordinators and removes the address from the location list.
2) Access Control Rules: Any change in the access control policy can be described by positive or negative access control rules. Therefore, we construct a message to reflect the change for a particular *role* and send it to the CA. The CA forwards the message to the root coordinator, from which the XPath expression in *object* is processed by each coordinator according to its state transition table, in the same way as constructing an automaton with a new ACR. If the message is accepted by an existing leaf coordinator, new automaton segments will be created and assigned to new coordinators. The location list at the original leaf coordinator will be copied to the new leaf coordinator.

5. PRIVACY AND SECURITY ANALYSIS

There are various types of attackers in the information brokering process. From their roles, we have abused insiders and malicious outsiders; from their capabilities, we have passive eavesdroppers and active attackers that can compromise any brokering server; from their cooperation mode, we have single and collusive attackers. In this section, we consider three most common types of attackers, local and global *eavesdroppers*, malicious *brokers* and malicious *coordinators.*

1) Eavesdroppers: A local eavesdropper is an attacker whocan 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. However, it can only learn the location of local broker from the captured packets since the content is encrypted. As a conclusion, an external attacker who is not powerful enough to compromise brokering components is less harmful to system security and privacy.
2) Single Malicious Broker: A malicious broker deviatesfrom 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 DoS attacks.

3) Collusive Coordinators: Collusive coordinators deviatefrom 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; (4) the receiver viewable to it is likely to be another coordinator.

6. CONCLUSION AND FUTURE WORK

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. PPIB is a new approach to preserve privacy in 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 and also it is 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.

In future research, to achieve the maximum level of security for the distributed information sharing, sign encryption process with private and public key into the PPIB has been considered. At present, site distribution and load balancing in PPIB are conducted in an ad-hoc manner; 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. Next quantify the level of privacy protection achieved by PPIB.

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 man-aging distributed, heterogeneous, and autonomous databases," *ACMComput. Surveys (CSUR)*, vol. 22, no. 3, pp. 183–236, 1990.

[3] L. M. Haas, E. T. Lin, and M. A. Roth, "Data integration through data-base federation," *IBM Syst. J.*, vol. 41, no. 4, pp. 578–596, 2002.

AX. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: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:

Anew abstraction for information management," *SIGMOD Rec.*, vol. 34, no. 4, pp. 27–33, 2005.