ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 239 – 244 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

IMPLEMENTATION OF PRIVACY PRESERVED BROKERING IN DISTRIBUTED INFORMATION SHARING

N.Nagendra[1], MS.G.Sushma[2]

[1]POST GRADUATE STUDENT, DEPT OF C.S.E MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDIA, 534nagendra@gmail.com

[2]ASSISTANT PROFESOR, DEPT OF C.S.E MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDIA, sushmagudivada05@gmail.com

***Abstract*—Today's organizations raise an increasing need for information sharing via on-demand access. Information brokering systems (IBSs) have been proposed to connect large-scale loosely federated data sources via a brokering overlay, in which the brokers make routing decisions to direct client queries to the requested data servers. Many existing IBSs assume that brokers are trusted and thus only adopt server-side access control for data confidentiality. However, privacy of data location and data consumer can still be inferred from metadata (such as query and access control rules) exchanged within the IBS, but little attention has been put on its protection. In this paper, we propose a novel approach to preserve privacy of multiple stakeholders involved in the information brokering process. We are among the first to formally define two privacy attacks, namely *attribute-correlation attack* and *inference attack*, and propose two countermeasure schemes *automaton segmentation* and *query segment encryption* to securely share the routing decision-making responsibility among a selected set of brokering servers. With comprehensive security analysis and experimental results, we show that our approach seamlessly integrates security enforcement with query routing to provide system-wide security with insignificant overhead.**

***Index Terms*—Access control, information sharing, privacy**

1.INTRODUCTION

It consists of diverse data servers and brokering components, which help client queries to locate the data servers. However, many existing IBSs adopt server side access control deployment and honest assumptions on brokers, and shed little attention on privacy of data and metadata stored and exchanged within the IBS. We implement a novel approach to preserve privacy of multiple stakeholders involved in the information brokering process and propose two countermeasure schemes automaton segmentation and query segment encryption to securely share the routing decision-making responsibility among a selected set of brokering servers. With comprehensive security analysis and experimental results, we show that our approach seamlessly integrates security enforcement with query routing to provide system-wide security with insignificant overhead.

ALONG 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 pair wise 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. Take healthcare information systems as example. 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 to retain full control over the *data* and the *access to the data*. Meanwhile, as a consumer, a healthcare provider requesting data from other providers expects to preserve her privacy (e.g., identity or interests) in the querying process.

In such a scenario, sharing a complete copy of the data with others or "pouring" data into a centralized repository becomes impractical. To address the

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 239 – 244 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

need for autonomy, federated database technology has been proposed 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 sharing.

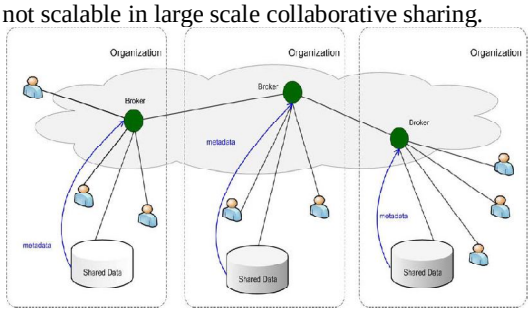


Fig. 1. Overview of the IBS infrastructure.

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 bead used 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, 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 embedded nondeterministic finite automata—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.

2.METHODS

2.1 Vulnerabilities and the Threat Model

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 (e.g., a patient in RHIO) is the identifiable data and sensitive or personal information carried by this data (e.g., medical records). 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 (e.g., information specifying her interests) in the querying content. For example, a query about AIDS treatment reveals the (possible) disease of the requestor. We adopt the *semi-honest* [12] assumption for the brokers, and assume two types of adversaries, *external attackers* and *curious or corrupted brokering components*. External attackers passively eavesdrop communication channels. Curious or corrupted brokering components, while following the protocols properly to full fill brokering functions, try their best to infer sensitive or private information from the querying process. Privacy concerns arise when identifiable information is disseminated with no or poor disclosure control. For example, when data provider pushes routing and access control metadata to the local broker a curious or corrupted broker learns *query content* and *query location* by intercepting a local query, *routing metadata* and *access control metadata* of local data servers and from other brokers, and *data location* from routing metadata it holds. 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 *query location* and *data location* from eavesdropping. Combining types of unintentionally disclosed information, the attacker could further infer the privacy of different stakeholders through *attribute-correlation attacks* and *inference attacks*.

2.1.1 Attribute-correlation attack. Predicates of an XML query describe 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

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 239 – 244 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

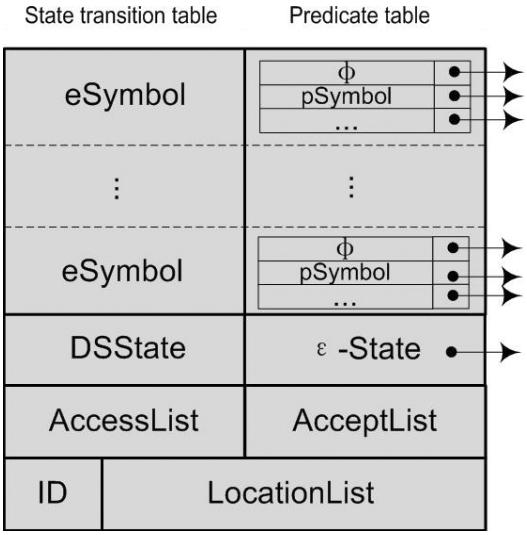
expressions, the attacker can "correlate" the attributes in the predicates to infer sensitive information about data owner. This is known as the *attribute correlation attack. Example 1:* A tourist Anne is sent to ER at California Hospital. Doctor Bob queries for her medical records through a medicare IBS. Since Anne has the symptom of leukemia, the query contains two predicates: [pName="Anne"], and [symptom="leukemia"]. Any malicious broker that has helped routing the query could guess "Anne has a blood cancer" by correlating the two predicates in the query. Unfortunately, query content including sensitive predicates cannot be simply encrypted since such information is necessary for content-based query routing.

2.1.2 Inference 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 (e.g., IP address). 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. 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"

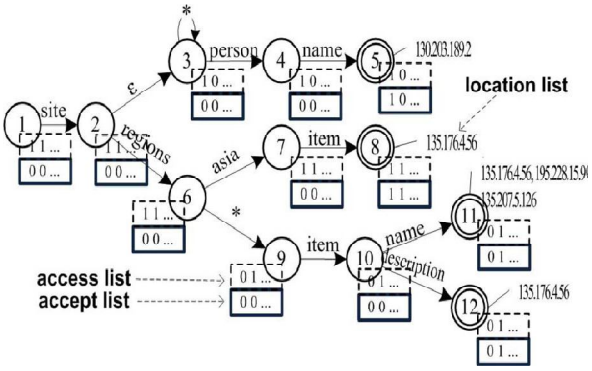
Solution Overview
To address the privacy vulnerabilities in current information brokering infrastructure, we propose a new model, namely *Privacy Preserving Information Brokering* (PPIB). PPIB has three types of brokering components: *brokers*, *coordinators*, and a *central authority* (CA). The key to 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.

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 assign each segment to a coordinator. Coordinators operate collaboratively to enforce secure query routing. A *query segment encryption scheme* is further proposed to prevent coordinators from 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 secure routing are revealed. Last but not least, we assume a separate *central authority* handles key management and metadata maintenance.

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. 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.



. Data structure of an NFA state.



State transition graph of the QBroker that integrates index rules with ACRs.

2.2 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

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 239 – 244 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

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.

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

Algorithm 1 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: $DS.nextState \leftarrow addr$
5: $S.StateTransTable(k).nextState \leftarrow DS$
6: **end for**
7: $Seg = createSegment()$
8: $Seg.addSegment(S)$
9: $Coordinator = getCoordinator()$
10: $Coordinator.assignSegment(Seg)$
11: **return** $Coordinator.address$

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.

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. 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 C Amaintains 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.

Handling the Predicates: In the original construction of NFA predicate table is attached to every child state of an NFA state as shown in Fig. 3. The predicate table stores predicate symbols (i.e., pSymbol), if any, in the corresponding query XPath step. An empty symbol means no predicate.

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

Level-Based Pre-encryption: 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

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 239 – 244 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

Post encryption: 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, $\{pk_{DS}, sk_{DS}\}$, 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 so that only the data servers can view it.

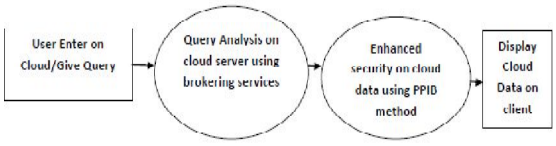
3. PROPOSED SYSTEM

Here we propose the problem of privacy protection in information brokering process. We first give aformal presentation of the threat models with a focus on two attacks: attribute correlation attack and inference attack. Then, we propose a broker-coordinator overlay, a well as two schemes ,automaton segmentation scheme and query segment encryption scheme, to share the secure query routing function among a set of brokering servers. With comprehensive analysis on privacy, end to-end performance, and scalability, we show that the proposed system can integrate security enforcement and query routing while preserving system-wide privacy with reasonable overhead. We propose a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. In particular, we propose an object-cantered approach that enables enclosing our logging mechanism together with users' data and policies. To strengthen user's control, we also provide distributed auditing mechanisms. We provide extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches.

3.1 ADVANTAGES

The Data kept confidential and hence an organisation can fully share his data on to the cloud. The user is provided with the integrated data by health care provider which is kept away from the brokering services. The brokering services can't be able to take or modify a decision of an healthcare data.

3.2 SYSTEM ARCHITECTURE

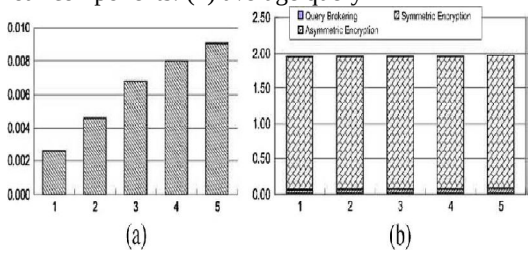


3.3 PERFORMANCE ANALYSIS

In this section, we analyze the performance of proposed PPIB system using end-to-end query processing time and system scalability. In our experiments, coordinators are coded in Java (JDK 5.0) and results are collected from coordinators running on a Windows desktop (3.4 G CPU). We use the XMark XML document and DTD, which is wildly used in the research community. As a good imitation of real world applications, the XMark simulates an online auction scenario.

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



Estimate the overall processing time at each coordinator. (a) Average query brokering time at a coordinator. X: Number of keywords at a query broker. Y: Time (s). (b) Average symmetric and asymmetric encryption time. X: Number of keywords at a query broker. Y: Time (ms).

3.3.1 Average Query Processing Time at the Coordinator:

Query processing time at each broker/coordinator consists of: (1) access control enforcement and locating next coordinator (Query brokering); (2) generating a key and encrypting the processed query segment (Symmetric encryption); and (3) encrypting the symmetric key with the public key created by super node (Asymmetric encryption).

Average Network Transmission Latency: We adopt average Internet traffic latency 100 ms as a reasonable estimation of (from Internet traffic report) instead of using data collected T_N from our gigabyte Ethernet.

3.3.2 System Scalability

We evaluate the scalability of the PPIB system against complicity of ACR, the number of user queries, and data size (number of data objects and data servers).

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 239 – 244 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

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.

Number of Queries: Considering queries submitted into the system in a unit time, we use the total number of query segments being processed in the system tomeasure the system load. When a query is accepted as multiple sub queries, all sub queries are counted towards system load. rejected after segments, the processed segments are counted.

4. 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 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.

5. 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.

[9] 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*, Taichung, Taiwan, 2006, pp. 252–259.

[10] 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 *Proc. ACM CCS’07*, 2007, pp. 508–518.

[11] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[12] R. Agrawal, A. Evfimivski, and R. Srikant, “Information sharing across private databases,” in *Proc. 2003 ACM SIGMOD*, San Diego, CA, USA, 2003, pp. 86–97.

[13] M. Genesereth, A. Keller, and O.Duschka, “Informaster: An information integration system,” in *Proc. SIGMOD*, Tucson, AZ, USA, 1997.

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

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

[16] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup protocol for Internet applications,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, Feb. 2003.