Secure Information Brokering System with Privacy Preserving Capabilities

[1]Patrick Raj Ezhil F, [2]Mano Paul P
[1]PG Student, Loyola Institute of Technology and Science, Thovalai, TN, India
[2]Assistant Professor, Loyola Institute of Technology and Science, Thovalai, TN, India

***Abstract* - Information brokers are individuals or business entities that researches information for their clients. They are usually used for patent searches, business forecasts and market information research. Information brokering systems are used to retrieve information from various data sources available across the global internet. Existing information brokering systems operate assuming that the brokers are fully trusted and adopt mechanisms only for data confidentiality. But corrupted brokers can violate the privacy of data owners and data requesters by leaking sensitive private information. In this paper we propose schemes to preserve the privacy of data requesters and data owners. The privacy is ensured through two counter measure schemes secure metadata share, and secure data retrieval.**

1. INTRODUCTION

The rise in the amount of information collected by different organizations through their operations in areas like business information, governmental information, healthcare information etc, contributes for sharing these data in order to facilitate collaborative computing. Client server model and distributed data base model are the traditional models widely used for information sharing. But these models may not suit for sharing information in emerging areas such as healthcare and law enforcement where the information sharing must be done in a controlled manner.

A more data centric query answering model called Information Brokering System (IBS) came into existence in order to solve the issues with the two traditional models mentioned earlier. In this model a set of independent entities known as brokers make the routing decision of queries based on the contents of queries towards the actual data owners. Semantic mechanisms are built to support the content based query routing. In the IBS model heterogeneous data owners are interconnected through the brokers. A metadata consisting of the summary of the data and server locations along with the access control information is shared to the local broker of the particular data owner to whom the data owner is subscribed to. The local broker advertises the metadata in full or part to other brokers to indicate about what information he could provide.

The data requester on its necessity places the query to its local broker to whom it is subscribed to. The query will be routed based on the metadata until it reaches the right data server. Thus the IBS provides a unified, transparent and on-demand information sharing functionality.

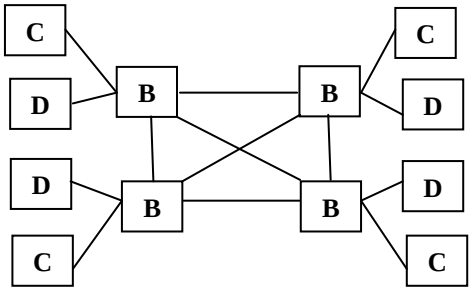

Figure 1 Architecture of Information Brokering System

Figure 1 represents the architecture of the existing Information Brokering Systems. The aim of the data requester [C] is to obtain information about some topic. The requester role initiates the process transaction by requesting data, and receives the results of the transaction. The requester may deal with actors playing either of the other two roles: the broker [B] or the data owner [D]. These actors may themselves play the role of the requester while requesting further services from other brokers.

The broker provides brokerage services to the requester and the owner. It responds to requests from the requester to provide information. The information that the broker supplies to the requester may originate from one or more sources. The broker's primary role is to act as a collector and collator of information from a number of different sources, and to supply this information to the requester, thus obviating the need for the requester to deal with a variety of sources. A broker can also be considered to act on behalf of a source, to distribute information.

The owner is the source of the data supplied to the requester. The owner provides the broker with information that it can supply.

IBS provides required scalability and server autonomy. But on the other hand privacy issues may arise. The existing IBS operates with fully trusted broker principle. But with broker functionality outsourced to third party providers the brokers are no longer trustable. Corrupted brokers can violate privacy of data requesters and data owners by inferring private information of data requesters and data owners through the query and metadata respectively and leaking the information.

To address these privacy vulnerabilities in existing IBS, we propose a new model in this paper. The proposed system has three types of brokering components: local proxy nodes, a distributed routing system and an admin network. The complete functionality is divided among various brokering components and no meaningful inference can be done by any of the participating components.

2. PRIVACY VULNERABILITIES IN EXISTING IBS

In a typical IBS three types of components are involved; the data requester, the broker and the data owner. Each component has its own privacy which should be preserved. The data owner's privacy is the sensitive personal information which it is going to share. For example in a healthcare IBS scenario the data owner may be a patient who shares his medical records. The data owner signs strict privacy agreements with its local broker. The data requester may reveal private personal interests in its query. For example in the healthcare IBS scenario the query by a requester may reveal his disease.

Privacy issues arise when private information is disseminated without control in disclosure. Through the query content and the metadata the identity of both the data requestor and the source may be revealed with their private information. An attacker can be a corrupted broker or an eavesdropper who intercepts data flowing across the network. The attacker could infer sensitive private information and reveal the same in public. This will violate the privacy of the victim. The victim could be either the data owner or the data requester. As we seen in the healthcare IBS, if no disclosure control is present the disease of either the requester or the owner may be made public.

In summary, from the query content and metadata an attacker can infer both: who is interested in what and which server has what data.

3. SOLUTION OVERVIEW

To address the privacy vulnerabilities as discussed above in existing IBS systems, we propose a new model in this paper. The proposed system has three types of brokering components: proxy nodes, routing network and an admin network. The complete functionality is divided among various brokering components and no meaningful inference can be done by any of the participating components.

Figure 2 shows the architecture of the proposed system. Data owners [D] and requesters [C] from different organizations or individuals will be connected to local proxies [P]. Proxies are interconnected through distributed routing system i.e. the network formed by routing nodes [R]. During the data retrieval process, local proxy authenticates the requester and hides its identity from other brokering components. Similarly a proxy authenticates the owner and hides its identity to other components during the metadata share process.

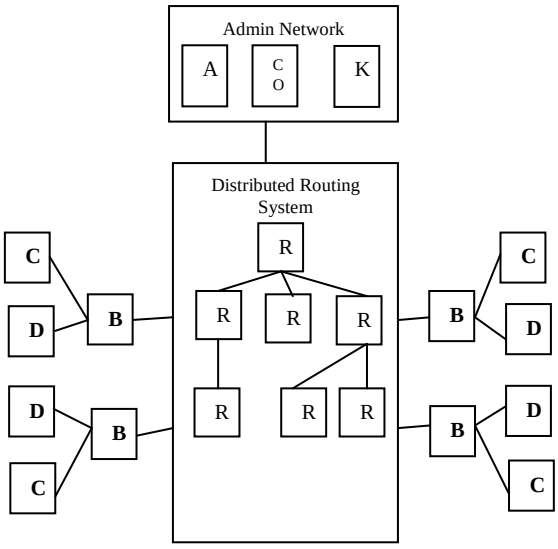

Figure 2 Architecture of the Proposed System

The distributed routing system is a distributed index server system which is made up of routing nodes. The routing nodes are connected in an m-way search tree topology. The routing nodes are responsible for content based query routing and access control enforcement. The entire index information is fragmented and each fragment is assigned to a routing node. This is to eliminate possibility of single point failure bottleneck and to ensure survivability even at the failure of a routing node. The routing of query through this network is known as distributed query routing.

The admin network is used for managing the entire system. Following are its components. The admin node [A] controls the whole network. The kerberos system [K] performs cryptographic key management and the coordinator node [CO] performs metadata maintenance and maintenance of the routing network.

A secure data retrieval scheme is proposed to prevent brokering components from inferring sensitive private information during the actual data retrieval process. Similarly a secure metadata share scheme is proposed to prevent brokering components from inferring sensitive private information during the metadata sharing process.

4. RELATED WORK

Database federation [1] employs a database engine to create a virtual database from several, possibly many, heterogeneous and distributed data stores. In all of the database federation techniques, the database engine is the key driver, but the method by which data or functions are included in the federation differs. It is feasible to construct a toolkit [2] of privacy preserving computations that can be used to build data mining techniques. There are still many subtleties involved, simply performing one secure computation, then using those results to perform another, reveals intermediate information that is not part of the final results. The resulting data mining technique no longer meets the definition of a secure multiparty computation.

Regarding indexing [3], the main challenge is handling both value and path indexes. In structured p2p systems, the central problem is deriving appropriate mappings of documents to peers, while in unstructured p2p systems, the main issue is constructing space and update efficient routing indexes. But centralized repository of indexes may create central sever bottle neck which is a disadvantage, which is needed to be addressed.

Replication [4] is commonly used in peer-to-peer systems to improve response time of queries. The focus is on replicating XML documents. A distinctive feature of XML replication is determining an appropriate granularity that would allow different levels of replication for fragments of the same XML document. But with no global knowledge on fragments is a disadvantage of this system.

Privacy concerns arise in inter-organizational information brokering since one can no longer assume brokers controlled by other organizations are fully trustable. As the major source that may cause privacy leak is the metadata, secure index based search schemes [5] may be adopted to outsource metadata in encrypted form to un-trusted brokers.

A DHT framework [6] is used for efficiently locating XML data in a peer to peer environment. Each XML document is mapped into an algebraic signature that captures the structural summary of the document.

Brokers are assumed to enforce security check and make routing decision without knowing the content of both query and metadata rules. Various protocols have been proposed for searchable encryption [7] however all the schemes presented so far only support keyword search based on exact matching. While there are approaches proposed for multidimensional keyword search and range queries, supporting queries with complex predicates or structures is still a difficult open problem. Research on anonymous communication [8] provides a way to protect information from unauthorized parties.

Finally, research on distributed access control [9-11] gives a good overview on access control in collaborative systems. In summary, earlier approaches implement access control mechanisms at the nodes of XML trees and filter out data nodes that users do not have authorization to access. These approaches rely much on the XML engines.

5. DISTRIBUTED ROUTING SYSTEM

When we consider IBS, many organizations join a consortium and agree to share the data within the consortium. Different organizations may have different schemas; we assume a global schema which all organizations need to follow. All index and access control metadata are captured into a global automation. The key idea of distributed routing system is to divide the global automation into independent logical fragments and assign each fragment to a participating routing node in order to avoid central server bottleneck. The routing network is created, maintained and managed by the coordinator node. Following are various processes of distributed routing system.

1) Fragmentation: The unit of fragmentation is the metadata formed by index and access control information. Each fragment may have one or more metadata states. We further define the granularity level of fragmentation with the parameter called partition size. With higher granularity the query processing overhead will be reduced. But a coarse partition may increase security risk. The trade off between processing complexity and degree of privacy need to be considered in deciding the granularity level.

2) Deployment: We employ routing nodes to store logical fragments. To reduce the need for number of needed routing nodes, several fragments into same routing nodes but at different port numbers. Therefore a tuple <routing node, port no> can identify a fragment. After deployment, the routing nodes can be linked together according to their relative position in the m-way search tree structure based on the fragments they hold. The root node first intercepts the query and the query is traversed on the tree based on m-way tree search mechanism till it reaches the metadata which matches the query.

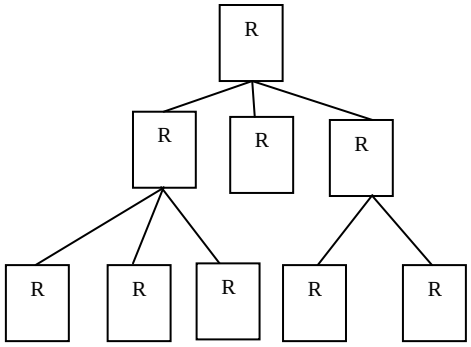

Figure 3 Distributed Routing System

3) Replication: Since the root node processes all queries, the root node may become a bottleneck and a single point of failure. Hence it should be replicated. Such replicas are maintained by the coordinator node. The passive path replication strategy is adopted to create the replicas for creating such replicas. The coordinator node maintains a set of replicas and creates or revokes the replicas as when needed. Figure 3 represents an example of how the distributed routing system looks like. The nodes are connected in an m-way search tree topology.

4) Handling Queries: The query when intercepted by the root node, it checks its metadata directory to locate the necessary data. If it does not find the necessary metadata in its directory, it forwards the query to one of its children based on m-way tree search strategy on the query. The same procedure is followed at each of the nodes until it reaches the node with the required information. At any point if it is found that the role which requests data is denied access to the particular data, based on the access control information in the metadata, the access denied message is returned to the requestor. When a routing node finds that it finds the required metadata in its directory, it simply requests for the data to the corresponding owner proxy with the corresponding anonymity code.

6. SECURE METADATA SHARE

When a data owner joins the system in the existing IBS, it simply forwards the metadata to its local broker. Since the local broker knows the identity of the owner and the detail about the data available with the owner, he could infer sensitive information and violate the privacy of the data owner. To avoid this possibility the secure metadata share scheme is proposed.

In this scheme when an owner joins the system, it prepares the metadata and encrypts it with the public key of the coordinator node. Then it forwards it to its local owner proxy. The owner proxy hides the identity of the owner to itself. Then it generates an anonymity code for the particular metadata which may be a random number. Now it forwards the encrypted metadata along with the anonymity code through the network to the coordinator node. Simultaneously it responds to the owner with the anonymity code for further reference.

Figure 4 represents the process of metadata registration. The coordinator decrypts and extracts the metadata and places it in the appropriate routing node in the routing network, which is an m-way search tree network, according to its position along with the anonymity code.

Now in this scheme only the local owner proxy knows the identity of the actual owner. But it does not know what data the owner is ready to share. The coordinator node and the routing nodes do not know who the actual owner of the metadata is but only know the corresponding proxy and the anonymity code.

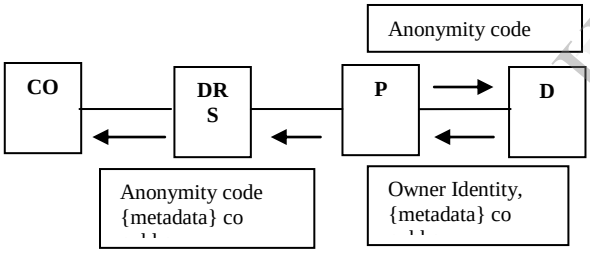

Figure 4 Secure Metadata Share

7. SECURE DATA RETRIEVAL

This is during the actual data retrieval process. 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 to do that since brokering servers need to view query content to fulfill access control and query routing. This is a three stage process: secure query forwarding, secure query routing and secure data forwarding.

1) Secure Query Forwarding: When a data requester wants to make a request for a data it contacts it local proxy. The local proxy forms an entrance to the system. The requester submits its client information i.e. user account and password details along with the query which is encrypted with the public key of the root of the DRS and a symmetric session key which is generated and meant only for the particular data retrieval session encrypted with the public key of the data owner which is assumed to be common for all data owners. When the proxy receives this request it hides the identity of the requester within itself. It then generates a request code for this request and forwards the request the root of DRS with the request code, encrypted query and the encrypted session key. By this way the query content is hidden to local proxy and all intermediate brokering components until the query reaches the root of the DRS. This stage is as represented in Figure 5.

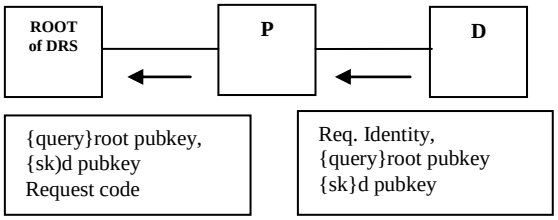

Figure 5 Secure Query Forwarding

2) Secure Query Routing: Once the query reaches the root node, it decrypts the encrypted query and the distributed query routing gets underway. It forwards the request code, the query and the encrypted session key through the DRS to find the appropriate proxy to whom the actual data owner is subscribed to. The routing is based on an m-way tree search strategy. Once the appropriate routing node is reached the request with the corresponding anonymity code and the encrypted session key is forwarded to the corresponding proxy. The proxy simply forwards the request to the actual data owner. This stage is as represented in Figure 6.

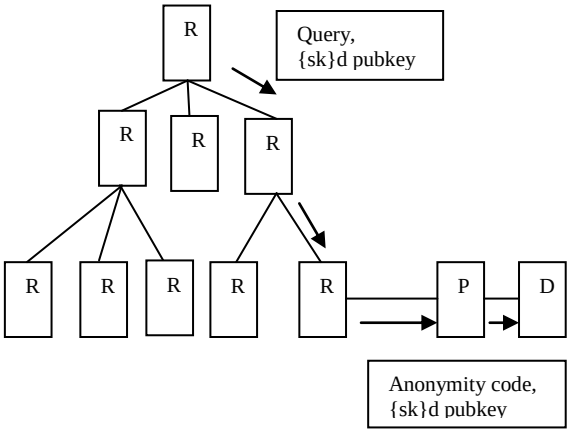

Figure 6 Secure Query Routing

3) Secure Data Forwarding: The data owner when returning data to the requester encrypts the data with the session key received from the requestor and forwards it along with request code. Once the encrypted data reaches the proxy of the requester, based on the request code it forwards the encrypted data to the actual requester. This is in order to provide data confidentiality and blocking any attempt to infer any private information based on the data content. The requester decrypts and extracts the actual data. This stage is as represented in Figure 7.

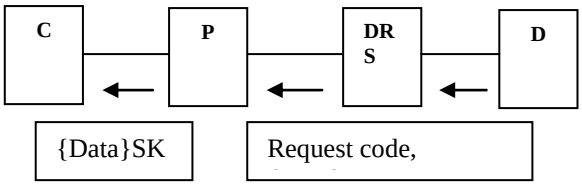

Figure 7 Secure Data Forwarding

8. ADMIN NETWORK

The admin node is assumed for initiation and maintenance. This is the only node operating with a higher level of trust. This is the overall controller of the system.

The kerberos node plays an important role in cryptographic key management. There are three types of keys used in the brokering process: the session key used for data encryption, public-private key pair of nodes at the routing network used for pre-routing encryption and public-private key pair which is shared by all data owners used for encrypting session key. Except the session key which is generated by the requestor all other keys are generated by the kerberos node.

Along with construction of distributed routing network, adding new data owners to the network is taken care by the coordinator node. It also creates replica of the root of the routing network and maintains it for failure recovery purpose. The admin functionality is distributed among these three nodes in order to avoid effects of single point failure. The replicas of the kerberos node and the coordinator node are maintained by the admin node in order to recover from failures of either of these nodes.

The admin node is self reconfigurable with low data overheads and hence can recover itself from any failures.

9. MAINTENANCE

1) Brokering Components Join and Leave: Proxies and routing nodes contributed by different organizations are allowed to join and leave the system dynamically. When a proxy joins the system, it contacts the coordinator to get the address list of the root of the routing network and its replicas. It also broadcasts its address to the local requesters. When it leaves the system it needs to send a disconnect message to its clients.

When a routing node joins the system it sends a request to the coordinator node. The coordinator node authenticates the node and assigns a fragment of global automation considering the balance among routing network and the trust of the routing node, and attaches it to the routing network and re-organizes the routing network accordingly. When a routing node leaves the system, the coordinator node extracts the fragments associated with it and decides whether a new routing node is kept as replacement or the fragment is employed at another node based on the load conditions.

2) Metadata Update: The metadata may change with changes in access control policy or the data distribution policy in a participating organization. The organization needs to send an update message through its proxy to the coordinator node, and the coordinator node updates the metadata at the appropriate routing node.

10. PRIVACY AND SECURITY ANALYSIS

The modules such as requester, owner, proxy and the routing nodes are coded in Java and except requester other modules are implemented as multithreaded servers. During every stage like metadata share or query brokering all data flow are displayed as status messages at each component and verified for compliance to privacy requirements.

There are various types of attackers in the information brokering process. In this section we consider three types of attackers: eavesdroppers, malicious brokers and malicious routing nodes. We analyze the possible privacy breakages by each one of these.

1) Eavesdroppers: An eaves dropper is an attacker who can observe all communication between two ends of a communication link. With the proposed schemes an eavesdropper cannot infer any meaningful information along with the identity.

2) Malicious proxies: A corrupted proxy node can expose the query content or metadata content if the encryption is not done on them by the requester or owner respectively. This would violate the privacy of the concerned data owner or the data requester respectively. But with the metadata content as well as the query content are in encrypted form when passing through the proxies as with the secure metadata share and secure data retrieval schemes this possibility is removed.

3) Malicious routing nodes: With the requester and owner identity are kept hidden by the local proxies malicious routing nodes have very little possibility of compromising the privacy of the requester and owner.

Table 1 presents the comparison between the traditional information brokering systems and the proposed system.

11. PERFORMANCE ANALYSIS

In this section we analyze the performance of the proposed system using the attributes end-to-end query processing time and system scalability. In our experiments all the components are coded using java and the results are collected from the components running at Windows desktops.

Table 1 Comparison with existing system

EXISTING SYSTEM	PROPOSED SYSTEM
Little attention on data requester and owner privacy	Ensures privacy of user and data
Adopts trusted broker model	Adopts semi honest broker model
Inference attack by corrupted insiders possible	Reduces inference attack
Eavesdropping could violate privacy of requester and source	Eavesdropping could not make any meaningful inference

1) End-to-End query processing time: We define the end-to-end query processing time as the time elapsed for the entire process during a requester transaction. It is a combination of average query processing time at each intermediate component during distributed query routing, average network transmission latency and number of hops passed through. Lesser the number of nodes traversed through in distributed routing network the time required is also reduced. The experiments were conducted by gradually increasing the number of data owners in the network and thereby increasing the number of proxies and routing nodes. The increased number of proxies does not have any impact on the performance of the system. But with the increase in the height of the routing tree network definitely has impact.

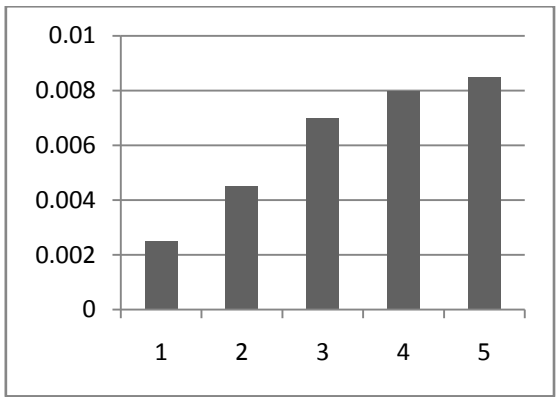

Figure 8 Overall Query Processing Times.

Figure 8 represents a graph which shows the variation in the overall query processing time with the increase in the number levels in routing network during simulation. The X axis represents the no of levels or the height of the routing tree and the Y axis represents time is seconds.

2) Scalability: The system scalability can be evaluated against complicity to global schema, number of queries submitted at unit time and data size. With data volume increases, the indexing rules also increases. This results in increase in number of the routing nodes. However there is possibility to add routing nodes and proxies in the proposed system, the system scalability can be achievable.

Similarly due to the flexibility of the secure metadata registration, adding more data owners is highly possible.

Similarly when we consider figure 8 when the number of routing nodes increases the overall processing gradually grows but it does not impact much because of the fact that, height of the tree does not increase every time with the increase in the number of routing nodes.

12. CONCLUSION

The existing IBS systems have very little attention on privacy of user, data and metadata and suffer from vulnerabilities associated with privacy. The proposed system provides approaches to preserve privacy in information brokering process. It provides comprehensive privacy protection for both data requesters and data owners.

13. REFERENCES

[1]. L. M. Haas, E. T. Lin, and M. A. Rot, "Data integration through database federation", IBM Syst. J., vol. 41, no. 4, pp. 578–596, 2002
[2]. C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu, "Tools for privacy preserving distributed data mining", ACM SIGKDD Explorations Newsletter, vol. 4, no. 2, pp. 28–34, 2003
[3]. 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
[4]. P. Skyvalidas, E. Pitoura, and V. Dimakopoulo, "Replication routing indexes for XML documents", Proc. DBISP2P Workshop, Vienna, Austria, 2007
[5]. D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data", in Proc. TCC'07, Amsterdam, The Netherlands, pp 535–554
[6]. P. Rao and B. Moon, "Locating XML documents in a peer-to-peer network using distributed hash tables", IEEE Trans. Knowl. Data Eng.,vol. 21, no. 12, pp. 1737–1752, Dec. 2009
[7]. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data", in Proc. ICDCS'10, Genoa, Italy, pp. 253–262
[8]. Larry A. Dunning and Ray Kresman, "Privacy Preserving Data Sharing With Anonymous ID Assignment", IEEE Transactions On Information Forensics And Security, Vol. 8, No. 2, February 2013
[9]. Fengjun Li, Bo Luo, Peng Liu, Dongwon Lee and Chao-Hsien Chu, " Enforcing Secure and Privacy-Preserving Information Brokering in Distributed Information Sharing", IEEE Transactions On Information Forensics And Security, Vol. 8, No. 6, June 2013
[10]. 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
[11]. 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.