ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

Privacy Preserving and Ensuring Secure Query Brokering In Distributed System

Harshitha.A.V (PG-Student, CSE Department, RIT, Hassan)
Ravindra. A.N (Asst.Professor, ISE Department, RIT, Hassan)

Abstract*— *Information sharing is increasing in today's organization via on demand access. Information brokering system (IBS) has been proposed to support information sharing among loosely federated data sources via a brokering overlay in which the brokers make routing decision to direct the client queries to the requested data servers. Federated databases are the collection of heterogeneous databases; it allows several databases to function as a single entity. IBSs assume that brokers are trusted and to limit the control and access to host systems for the protection of transmitted data is done only in server-side. But privacy of data location and data consumer can still derived from metadata such as query and access control rules exchanged within the IBS. In this paper, a novel approach has been proposed to preserve privacy of multiple stakeholders involved in the information brokering process. The two privacy attacks, namely attribute-correlation attack and inference attack are defined, 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.

***Index Terms*— Access control, information sharing, privacy, query segment encryption, segmentation.**

I. INTRODUCTION

Organizations desire to provide data access to their collaborators while preserving full Control over the data and comprehensive privacy of their users. A number of information systems have been developed to provide efficient and secure information sharing [1]. The problem of balancing peer autonomy and system coalition is still challenging. The rise of distributed information management (DIM) applications has followed the rise of the Internet. In these applications, users store information on a site for the purpose of sharing it with recipients [2]. On the other hand, sharing too much information is dangerous because the recipients of the information, or the system itself (assuming it is not controlled by the information owner), may have incentive to share sensitive data with eavesdroppers that a user has not authorized to view his data, but would gain from knowing it.

A Distributed Information Brokering System (DIBS) is a peer-to-peer overlay network that comprises diverse data servers and brokering components helping client queries locate the data server [3]. Federated information system with diverse participants (from different organizations) such as data producers, data consumers, or both, the need of cross-organizational information sharing naturally arises.

However, different types of applications often need different forms of information sharing. Although the Internet and various virtual private networks provide good data communication links, there are major challenges in
(a) Achieving scalable, agile and secure remote access of distributed data;
(b) Handling the heterogeneity among data management systems and data formats which are not always structured and may be incompatible with each other;
(c) Handling the dynamics of modern business applications (where new schema elements may emerge everyday); and
(d) Location discovery. To tackle these challenges, mediation and federation based information brokering technologies have been proposed.

In particular, recent extensible Markup Language (XML) has become a promising solution by integrating incompatible data while preserving semantics. An XML-based information brokerage system comprises data sources and brokers respectively, hold XML documents and document distribution Information [4]. In such systems, databases can be queried through brokers with no schema-relevant or geographical difference being noticed.

In recent trends, organizations raise an increasing need of information sharing to facilitate extensive collaboration among business to government agencies. In olden days Information brokering system(IBS) which acts as intermediate brokers which make decision routing between the query requester and data server by trusting third party. To overcome the trusted third party IBS, extended into privacy preserving for exchanging multiple stakeholders information.

II. INFORMATION BROKERING SYSTEM

The Information brokering systems work on two extremes of the spectrum; 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.

A. Architecture

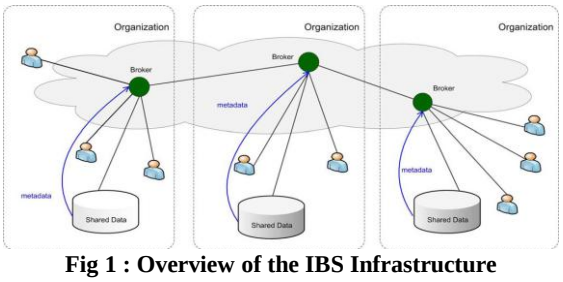

Fig 1 : Overview of the IBS Infrastructure

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

As shown in Fig. 1, applications atop IBS always involve some sort of consortium (e.g., RHIO) among a set of organizations. Databases of different organizations are connected through a set of brokers, and metadata (e.g., data summary, server locations) are "pushed" to the *local brokers*, which further "advertise" (some of) the metadata to other brokers. Queries are sent to the local broker and routed according to the metadata until reaching the right data server(s). In this way, a large number of information sources in different organizations are loosely federated to provide an unified, transparent, and on-demand data access.

B. Disadvantages

The IBS approach provides scalability and server autonomy but the 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.

III. PRIVACY PRESERVING INFORMATION BROKERING SYSTEM (PPIB)

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 are makes use routing protocols that create hard-to-trace communications by using a chain of proxy servers which untraceable and 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 non-deterministic finite automata – the query brokering automata. Two novel schemes are designed 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.

A. Architecture

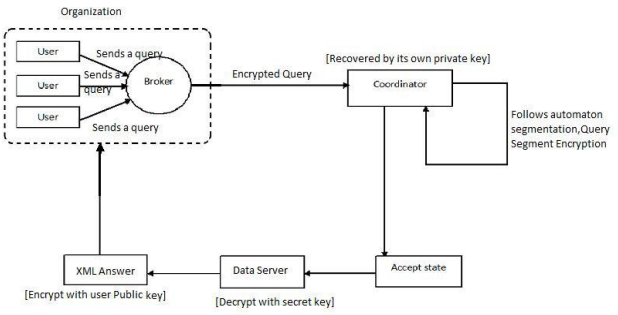

Fig 2: System Architecture of PPIB

Fig. 2 shows the architecture of PPIB. Data servers and requestors from different organizations connect to the system through local brokers (i.e., the green nodes in Fig. 2). Brokers are interconnected through coordinators (i.e., the white nodes).

A local broker is intercommunicating through coordinators and 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. Coordinators are responsible for content-based query routing and access control enforcement. Central Authority is responsible for key management and metadata maintenance.

IV. PRIVACY ATTACKS

In information brokering scenario, there are three types of entrepreneur, namely data owners, data providers, and data requestors. Each entrepreneur has its own privacy: (1) The privacy of a data owner (e.g. a patient) is identifiable data and the information keep together by this data (e.g. medical records). Data owners usually sign stiff privacy agreements with data providers to protect their privacy from unauthorized disclosure/user. (2) Data providers store collected data, and create two types of metadata, namely routing metadata and access control metadata. (3) Data requestors disclose identifiable and private information in the querying process. For example, a query process about AIDS or DNA treatment reveals the (possible) disease of the requestor.

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.

Attribute Correlation Attack: By using predicates which are used to find a specific node or a node that contains a specific value in XML query which describes conditions that 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.

Inference Attack: A procedure which combines known facts to produce ("infer") new facts, which makes use of premises, such a 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

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

association. By "implicit", we mean the attacker infers the fact by "guessing". For example, an attacker can guess the identity of a request or 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).

V. PPIB SCHEME

To tackle the privacy problem, the PPIB infrastructure is presented with different schemes. In this section, the details of access control rules, content based query routing, automata segmentation and query segment encryption scheme, the 4-phase query brokering process in PPIB are described.

A. Access Control Enforcement

The extensible Markup Language (XML) has emerged as the de facto standard for information sharing due to its rich semantics and extensive expressiveness. It is assumed that all the data sources in PPIB exchange information in XML format, i.e., taking XPath queries and returning XML data. Note that the more powerful XML query language, XQuery, still uses XPath to access XML nodes.

Access control is required in most if not all DIBS. The popular XML access control model is adopted. In this model, users are members of appropriate roles; and an access control policy consists of a set of role based 5-tuple access control rules (ACR): R = {subject, object, action, sign, type}, where
(1) Subject is a role to whom an authorization is granted;
(2) Object is a set of XML nodes specified by XPath;
(3) Action is one of "read,""write," and "update";
(4) Sign €{+, −} refers to access "granted" or "denied," respectively;
(5) Type €{LC,RC} refers to either "Local Check" (i.e., authorization is only applied to attributes or textual data of context nodes), or "Recursive Check" (i.e.,authorization is applied to context nodes and propagated to all descendants).

When an XML node does not have either explicit (via LC rules) or implicit (via RC rules) authorization, it is considered to be "access denied." It is possible for an XML node to have more than one relevant access control rule. If conflict occurs between "+" and "–" rules, "–" rules take precedence. Five example access control rules under the 5-tuple model are shown in Figure 3. In DIBS, each owner contributes a policy governing the access to her data objects, and the system-wide access control policy is simply the union of all the per-owner policies.

R1: {role 1, "/site/categories//name", read, +, RC}
R2: {role 1, "/site/regions/*/item/location", read, +, RC}
R3: {role 1, "/site/regions/*/item/quantity", read, +, RC}
R4: {role 2, "/site/regions/*/item/description", read, +, RC}
R5: {role 2, "/site /regions/*/item/name", read, +, RC}

Fig 3: Example for access control rules (ACR's)

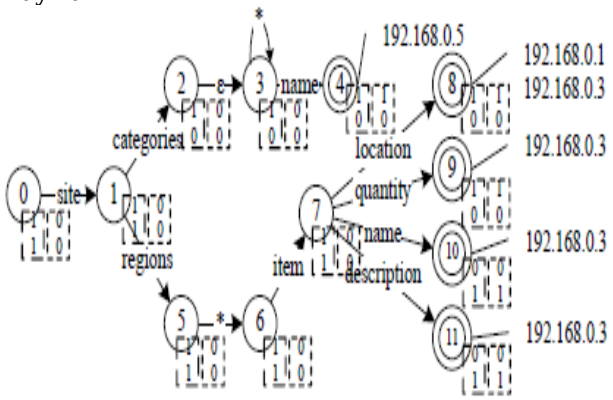

Fig 4: An example for ACR's, Index rules and routing

Here, an example used to illustrate how automaton based access control enforcement works. As shown in Figure 4, 5 access control rules assigned to two roles are used. First, an automaton is built based on the XPath expressions from object part of the rules. For example, constructing automaton with rule R1, automaton states are 0, 1, 2, 3, and 4 as shown in the figure. Especially, state 4 is an accept state, as shown in double-circle. Moreover, each state is attached with two binary arrays, namely access list (indicating which roles can access this state) and accept list (indicating this is accept state for particular role(s)), respectively. For instance, state 4 is accessible to users of role 1 only, and is an accept state for this role. At run time, user queries are checked against the automaton. Using the same example, suppose a user of role 1 asks two XPath queries.
(1) Q1: /site/categories/books/name goes through states 1, 2, 3, and reach accept state 4. As a result, this query is accepted.
(2) Q2:/site/regions/*/item/price is denied since no accept state can be reached.

B. Content based Query Brokering Scheme

In this scheme, each coordinator holds a set of indexing guidelines, and Indexing schemes have been proposed for content-based XML retrieval. The index describes the address of the data server that stores a particular data item requested by an user query. Therefore, a content based index rule should contain the content description and the address. A content-based indexing model with index rules in the form of I = { object, location } where,
(1) Object is an XPath expression that selects a set of nodes;
(2) Location is a list of IP addresses of data servers that hold the content.

It means that if a query matches the XPath expression, it will be forwarded to the IP address. Indexing guidelines are attached to accept states of the access control NFA, since only accepted/rewritten queries should be forwarded to the data servers. Comparing with the five access control rules, L1 is only relevant to R1. As a result, IP of 192.168.0.5 is attached to state 4, and all queries accepted at this state will be forwarded to data server 192.168.0.5. The indexing rules are shown in fig 5.
L1: {"/site/categories/category/name", 192.168.0.5}

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

L2: {"/site//*/item/location", 192.168.0.1}
L3: {"/site/regions", 192.168.0.3}

Fig 5: Example for Index Rules

C. *Automaton Segmentation*

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. The idea of automaton segmentation comes from the concept of multilateral security: split sensitive information to largely meaningless shares held by multiple parties who cooperate to share the privacy-preserving responsibility.

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. automaton segmentation scheme first divides the global access control automaton into several segments. Granularity of segmentation is controlled by a parameter partition size, which denotes how many XPath states in the global automaton are partitioned and put into one segment. Given a granularity level k , for each segmentation, the next i(€ [1,k]) states will be divided into one segment By and large, the granularity is a choice of the system administrator. Higher granularity leads to better privacy preserving, but also more complex query processing.

To reserve the logical connection between the segments after segmentation, the following heuristic segmentation rules are defined:
(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, the last states of each segment are made as "dummy" accept states, with links pointing to the segments holding the child states of the original global automaton.

In PPIB, a site is actually a logical unit. So a physical coordinator (i.e., a machine) can in fact hold multiple sites. For convenience, dummy accept states are added to each automaton segment. The dummy accept states do not accept queries. Instead, they are used to store the location of actual "next states," i.e. the address (es) of the coordinators who hold the next segment of the global automaton. At runtime, they are used to forward the halfway processed query to the next coordinators. On the other hand, only the sites holding original accept states accept queries and forward them to the data servers. As a result, access control and query brokering are seamlessly integrated at coordinators, and the global automaton-based query brokering mechanism is decentralized and distributed among many coordinators. The below shown are steps of the automaton segmentation algorithm.

For each granularity size k specified, the automaton is segmented recursively

- To deploy the segment, the first k size of the automaton is taken
- The dummy accept state is created
- The dummy accept state is attached to the first segment, which will hold the address of the next segment.
- As the algorithm executes recursively, the next segment will be deployed. This new segment address will be assigned to the next state of the previous dummy state.
- Again the new dummy state will be created and this is attached to the segment which is deployed previously.
- To assign segment to the coordinator, the deployed segment is created and stored in Segments field.
- The segments are added to the Segments field.
- The coordinator is taken.
- The segment is assigned to the Coordinator
- The Coordinator address is returned which will be taken as segment address.

D. *Example of Automaton Segmentation*

Algorithm demonstrates a recursive algorithm for finest-granularity automaton segmentation and deployment. As an example, the global automaton shown in Figure 6 is partitioned into 11 segments. For instance, Site 0 holds state 0 of the global automaton (symbol "site"); and a dummy accept state which holds the address of Site 1.

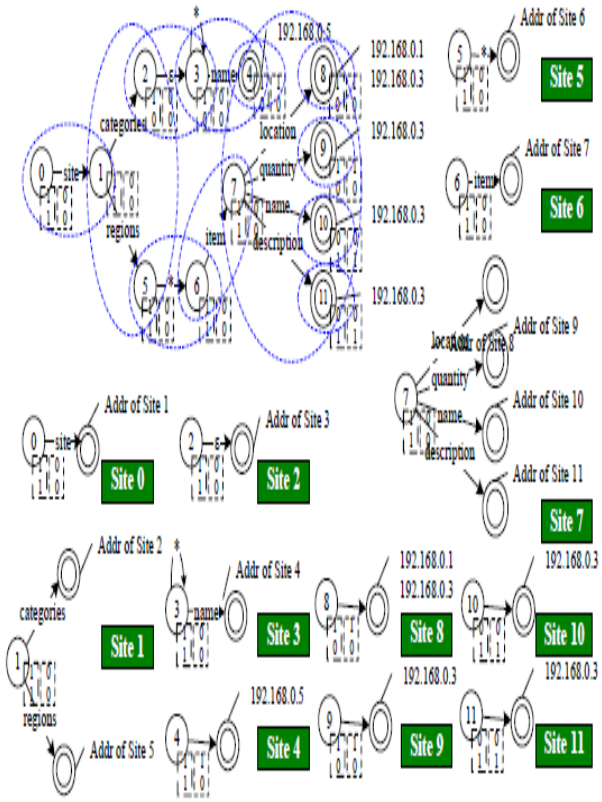

Fig 6: Example to illustrate the Automaton Segmentation Scheme

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

To illustrate how decentralized automata enforces access control, let us use the query Q: "/site/regions/asia/item [name='Abacavir']/location". When Q arrives at Site 0, the first XPath step "/site" is accepted. As the dummy accept state of Site 0 points to Site 1, Q is forwarded to Site 1. Then, the second XPath step "/regions" is accepted and the corresponding dummy accept state directs the remaining query to Site 5. There, Site 5 accepts "/asia" (wildcard "*" matches any input token) and forwards Q to Site 6. At Site 6, element name "item" is first accepted. Since the automaton segment does not carry any predicate states, the predicate from Q is kept as it is. Finally, "/location"is accepted at Site 7, and Site 10 forwards the query to data server at 192.168.0.3. Note that, "Abacavir" is a medicine. Therefore, the query Q, as well as related data and metadata, are all highly private and sensitive information. Under the automaton segmentation scheme, metadata privacy is preserved by dividing metadata into multiple sites.

E. Query Segment Encryption Scheme

To protect user/data privacy that may be revealed by the queries, a query segment encryption scheme is adopted, which is a good instance that combines data avoidance principle (i.e. encrypting sensitive data) with multilateral security principle (i.e. multiple parties cooperate to take one task, while each party only holds one share of sensitive information).

When an XPath query is being processed at a particular state in the NFA, the query content naturally splits into two parts: XPath steps that has been processed by NFA (accepted or rewritten), and XPath steps to be processed. Although the whole query will be forwarded to the coordinator who holds the next NFA state, NFA will only take the unprocessed steps as input. The idea of query segment encryption scheme is to encrypt the processed part of a query so that subsequent coordinators have only an incomplete view of the query content. For encryption, a trusted authority is needed for key distribution and management. In this scheme, trustee is the central authority. The notions used for encryption are defined as follows:

Both the public and private keys of an XML query are denoted as PubQ and PrivQ, respectively; then the corresponding encryption and decryption of string M are denoted as Encrypt (M, PubQ) and Decrypt (M, PrivQ), respectively; for the symmetric encryption scheme, the encryption and decryption applied to message M with secret key K are denoted as EK (M) and DK (M), respectively. The user submits an XML query with each segment encrypted by the corresponding public level keys, and a unique session key KQ and KQ is encrypted with the public key of the data servers to encrypt the reply data. When an XPath query Q = S_1S_2...Sn first arrives at the root-coordinator, it becomes the input to the automata segment. Suppose automata segment takes s1, first it decrypt its segment by its public key generated from central authority, generates S1′ and reaches the dummy accept state (when s1 is accepted, S1′ = S1, when S1 is rewritten, S1′ <> s1). The root-coordinator then requests a new PubQ from the central authority and encrypts S1′ as (EK_1 (S1′), Encrypt (K_1, PubQ)), where K_1 is the secret key of the root-coordinator. Both encrypted part and the remaining query (S2…Sn) are forwarded to the next coordinator. If the query passes all intermediate-coordinators and reaches the leaf-coordinator, the whole query will be encrypted as E_{K1} (S1′), E_{K2} (S2′), ...,E_{Kn}(Sn′). Thus, the entire query content is hidden from the leaf-coordinator. Once it arrives at the accept state of any leaf-coordinator by decrypting only its segment, the query is sent to destined data servers by encrypting whole thing with the public key of the data server. After decryption, again data server decrypts the secret keys (K1, ...,Kn) of the coordinators with the private key from the central authority, and then decrypts all the encrypted segments (S1…Sn) of the query with these secret keys. After decryption, the data server evaluates the query and returns the data, encrypted by K_Q , to the broker that originates the query.

F. Phases of Query Brokering Process

The query brokering process in four phases is shown in figure 7. Where users and data servers of multiple organizations are connected via a broker coordinator overlay. In particular, the brokering process consists of four phases:

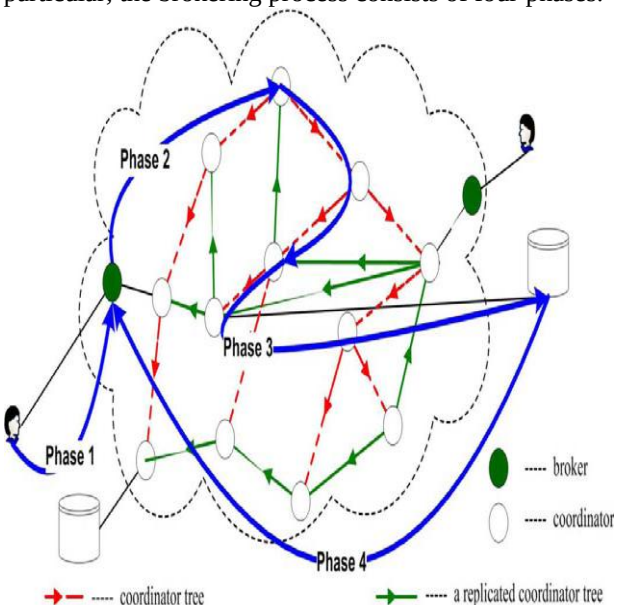

Fig 7: Query Brokering Process in 4 Phases

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 so that broker cannot see the query content, 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 of the authenticated user to attach to the encrypted query;

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

(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.

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 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.

VI. RESULTS

In this section, the performance of proposed PPIB system is analyzed using end-to-end query processing time and system scalability. In experiments, coordinators are coded in Java and results are collected from coordinators running on a Windows desktop. The XMark XML document and DTD is used.

X: Number of keywords at a query broker; Y: Time (ms)

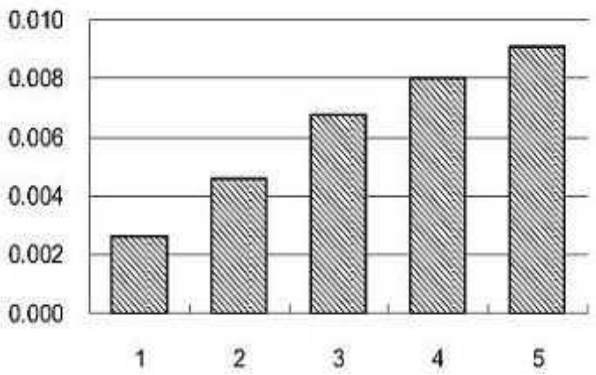

Fig 8: Average query brokering time at a coordinator

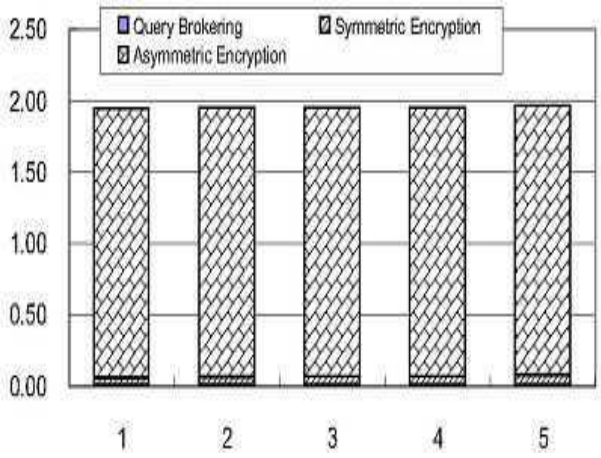

Fig 9: Average Symmetric and Asymmetric encryption Time

X: number of access control rules; Y: number of coordinators.

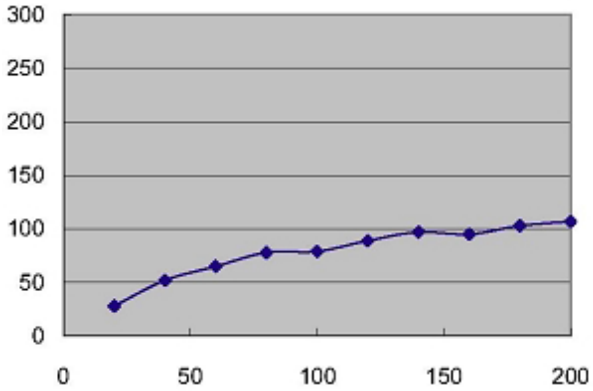

Fig 10: Using simple path rules for system scalability for number of coordinators.

X: number of queries in a unit time; Y: number of total segments in the system.

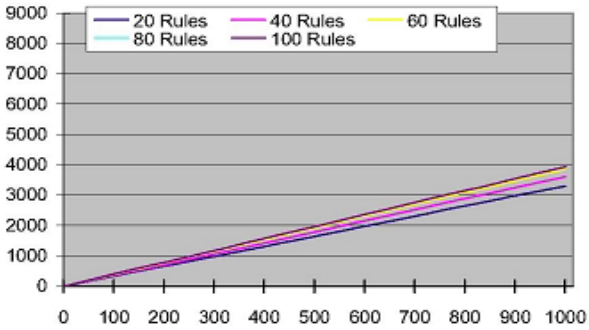

Fig 11: Using simple XPath rules and simple XPath queries for system scalability for number of query segments.

VII. CONCLUSION

Privacy issues of user and data is considered and concluded that existing information brokering systems suffer from a spectrum of vulnerabilities associated with user privacy, data privacy, and metadata privacy. In this paper, PPIB proposed architecture is discussed, a new approach to preserve privacy in XML information brokering. By using automaton segmentation scheme, within network access control and query segment encryption, PPIB put together security enforcement and query forwarding at the same time as providing comprehensive privacy protection. It can be claimed that this analysis is very resistant to privacy attacks. Node-to-node 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] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, Cool Streaming/DONet: A data-driven overlay network for efficient live media streaming," in Proc. IEEE INFOCOM, Miami, FL, USA, 2005, vol. 3, pp. 2102–2111.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

[3] A. C. Snoeren, K. Conley, and D. K. Gifford, "Mesh-based content routing using XML," in Proc. SOSP, 2001, pp. 160–173.

[4] N. Koudas, M. Rabinovich, D. Srivastava, and T. Yu," Routing XML queries," in Proc. ICDE'04, 2004.

[5] 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.

[6] M. Franklin, A. Halevy, and D. Maier, "From databases to data spaces: A new abstraction for information management," SIGMOD Rec., vol. 34, no. 4, pp. 27–33, 2005.

[7] F. Li, B. Luo, P. Liu, D. Lee, P. Mitra,W. Lee, and 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.

[8] 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.

AUTHOR'S PROFILE

Ravindra A.N. Completed B.E (CSE) from STJIT College Ranebennur, Karnataka in 2008 and pursued M.Tech (Automation and Robotics) in Malnad College of engineering Hassan, Karnataka. His main research interests include Network Security, Networking, and Operating systems.

Harshitha A. V. Completed B.E (ISE) from Malnad College of Engineering Hassan, Karnataka in 2012 and pursuing M.Tech (CSE) in Rajeev Institute of Technology Hassan, Karnataka. Her main research interests include Network Security, Networking and Data Mining.