Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 12, December 2014, pg.245 – 253

RESEARCH ARTICLE

Security Enforcement and Query Forwarding While Preserving System Wide Privacy

Prof. K.D.Yesugade[1], Pooja S. Bhosale[2], Sayali P. Gavhane[3]

[1]Dept. of Computer Engineering, Bharati Vidhyapeeth's College of Engineering, For Women, Pune, India

[2]Dept. of Computer Engineering, Bharati Vidhyapeeth's College of Engineering, For Women, Pune, India

[3]Dept. of Computer Engineering, Bharati Vidhyapeeth's College of Engineering, For Women, Pune, India

[1] Kiran_yesugade@yahoo.com , [2] poojabhosale21@gmail.com , [3] sayaligavhane55@gmail.com

ABSTRACT- As we know that mutual data shared in distributed information sharing environment. Information brokering system (IBS) used to access information from data server. These IBS systems are on demand information access and has brokering overlay. Brokers redirects the data requester's queries to requested data servers .In IBS systems the Brokers are not always trustable one because they may adopt access control for data confidentiality. With the purpose of providing privacy to user data and metadata we propose new system called as Privacy Preserving Information Brokering system (PPIB).PPIB is the Broker coordinator overlay network.

It consists of two methods i.e. Automaton Segmentation and query segment encryption. This PPIB approach prevents two attacks as attribute correlation attack and inference attack which was present in existing IBS systems.

Keywords- Access control, Automaton segmentation, Information sharing, Query segment encryption

I. INTRODUCTION

Information sharing is becoming increasingly important in many organizations (e.g. government Agencies, supply management, digital libraries, RHIO, Banking etc.). For example In Banking, Customer's account information is gathered and maintained in local databases. While customer want to open loan account then we need to access the information they may potentially come from any remote division to check whether that customer has already loan account in same or different branches of that bank. In Distributed information sharing there are some challenges are :(a) Handling data heterogeneity where data format and data management systems are not structured and incompatible with each other.(b) Achieving scalable , secure and agile remote access of distributed data.(c) Location discovery.(d)Modern business application handling.

To face these challenges federation based information brokering technologies are proposed. Solution to integrating incompatible data while preserving semantics, the Extensible Mark-up Language (XML) documents are used by brokers and distributing information. We can handle query brokering and access control as two issues: Query brokering issues concentrates on performance and cost while access control concerns data confidentiality.

The existing IBS systems works on two extremes: (1) In traditional distributed database system, all the clients lost autonomy and they are managed by unified DBMS.(2)In query answering model , peers are fully autonomous but there is coordination among system. So that user creates one to one client server connection for information sharing. For many emerged applications these extremes are not suitable. For example In Banking application information is shared in controlled and conservative manner due to some legal and security reasons.

II. EXISTING SYSTEM

In our study, the distributed systems providing data access using the set of brokers is termed as Information Brokering System (IBS).

As shown in fig.1. Databases of different organizations are connected using collection of brokers, and metadata about server location and data summary are pushed on local brokers. The local brokers then advertise some of metadata to other brokers present in the network. Every query of user is sent to local brokers. This query gets routed following the metadata of all brokers and reaches to right database. In this way large no of information sources in different organizations are loosely federated to provide fused, clear and on demand data access.

The IBS approach provides autonomy of server and scalability, This arise privacy concerns as brokers are no longer assumed trustable they may be compromised by outsiders or abused by insiders.

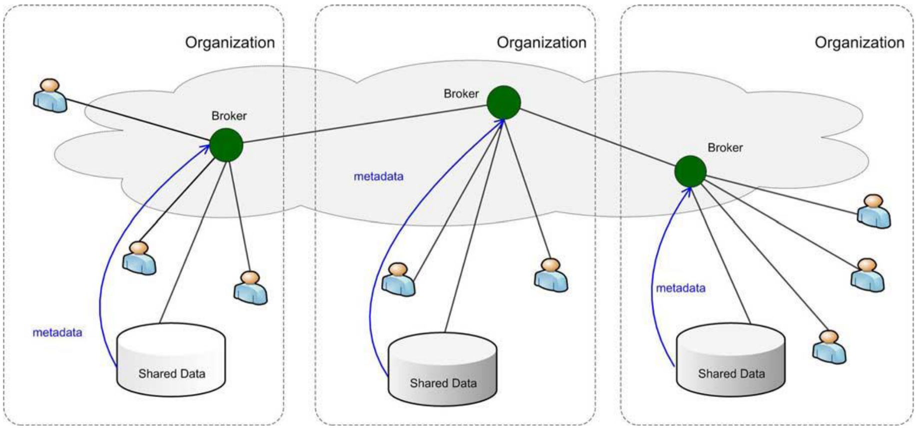

Fig 1.Information Brokering System(IBS)

III. PROBLEMS

A. Vulnerabilities and threat model:

There are three types of stakeholders as: data owner, data providers and data requestor. Each of these has its own privacy: (i) the privacy of data owner is identifiable data and the information carried by this data. The data owners wants their information should not be accessed by any unauthorised user.(ii) Data owner stores collected data and two types of metadata i.e. Routing metadata and Access Control metadata for brokering the data. These both metadata are the privacy of data provider. (iii)Data requestor uses or discloses identifiable and private information in the querying process.

The attacks can be classified into two types: The Attribute correlation attack and the Inference attack.

1) Attribute correlation Attack: The Query typically contains several predicates which describe sensitive and private data condition. The attacker intercepts these query which is having multiple or composite predicates expressions, The attacker can ”correlate” the corresponding attributes to infer sensitive data about data owner. These types of attacks are known as attribute correlation attack.

2) Inference attack: Privacy of data owner leaks when an attacker obtains more than one types of sensitive information. And further he learns explicit and implicit knowledge related to stakeholders. By implicit means the attacker infers the fact by” guessing". E.g. attacker can guess the identity of a requestor from his query location (IP address). By explicit knowledge, identity of data owner could be learned from query content. For making inference attacker can also obtain publically available information.

The three reasonable inferences from three distinct combination of private information (i) the attacker can infers about where who is or who is interested in what or something about data owner, from query location and query content. (ii)From query location and data location the attacker infers about who is interested in what type of data. (iii) From query content and data location, the attacker infers which server contains which data.

IV. PROPOSED SYSTEM

For overcoming untrustable brokering system problem we proposed new approach called Privacy Preserving Information Brokering (PPIB). PPIB is a broker-coordinator overlay infrastructure having two components: Brokers and Coordinators. The brokers are responsible for user authentication and query forwarding. The coordinators are connected in tree like structure.

They enforce access control and query routing based on brokering metadata. To prevent corrupted coordinators from inferring private information we are using two schemes: (a) *Automaton Segmentation*- To segment the query brokering autometa. (b) *Query segment encryption*- To encrypt corresponding query segments.

As shown in Fig.2, data servers and data requestors are connected to the system through local brokers. Brokers and coordinators are interconnected. Function of broker is to authenticate requestor and hide requestor's identity from other brokering components. Coordinators are responsible for content based query routing and access control enforcement. For privacy consideration we cannot let a coordinator to hold complete rule instead divide rules into segments using Automaton segmentation scheme, then assign each segment to coordinators.

For enforcing secure query routing coordinators operates collaboratively. To protect query content we propose query segment encryption scheme. So that coordinators cannot see sensitive predicates. Last but not least, there is a separate brokering server is assumed which is called as Central Authority (CA) which is responsible for key management and metadata maintenance.We show that PPIB provides privacy protection for on demand information brokering with little overhead and large scalability.

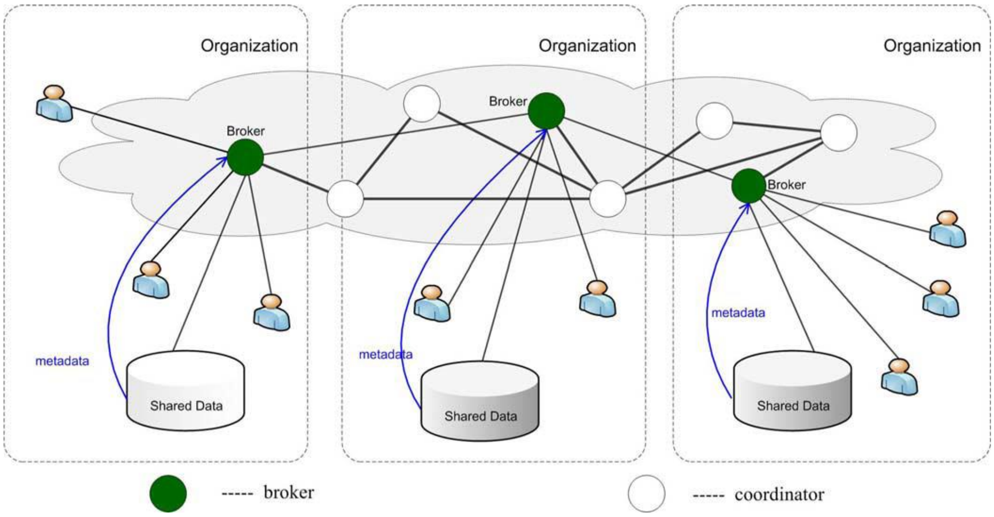

Fig.2. Architecture of PPIB

V. PRIVACY PRESERVING QUERY BROKERING SCHEME

In the existing system, if QBroker is not fully trustable or if they are compromised by outsiders then there may be various issues related to the privacy of data owner and data requestor. To address such kind of problem we are going to introduce Privacy Preserving Information Brokering (PPIB) system. In Privacy Preserving Information Brokering (PPIB), there are two important Schemes i.e. *Automaton Segmentaion* and *Query Segment Encryption.* The *Automaton segmentation* scheme is used to divide QBroker into multiple components. These components are logically independent from each other. So that it is efficient to process only the small piece of information i.e. User queries. By collaborating all this information from multiple components, we can have original and meaningful information.

In *Query Segment Encryption* scheme, it allows to encrypt the small piece of information with the help of different keys.

A. Automato Segmentation:

In Distributed Information Brokering, many organizations are connected to each other. So, they can share the information among them. There are different organizations hence we are assuming global schema and local schema, each organization can have its own local schema. Global schema is used to align and to merge the local schema.

The basics idea behind the *Automaton Segmentation Scheme* is to divide the global automaton into no. of segments. These segments are independent from each other but still they are physically connected. These segments are allocated to different brokering servers for further processing.

1)Segmentation: Non-deterministic Finite Automata (NFA) is the important part in the segmentation. So that each segment can hold one or more NFA states. The largest distance between two NFA states of same segment is denoted by *granularity level.* If *granularity level* large then it states that each segment is holding more than one NFA states. It helps to reduce the no. of segments. The processing complexity and privacy requirements should be considered while deciding the *granularity level* .As privacy protection is main goal, the *granularity level* should always less than or equal to 2.Once the segmentation is over then to have logical connection between those segments we are using *Heuristic Segmentation Rule.* This rule states that: (i) if there are many NFA states in one segment then these states should be connected through Parent-Child link.

(ii) The segment cannot hold the sibling NFA states without having parent state. (iii) There should be separate segment for Accept state of original global automaton. Last state of each segment should be *Dummy accept state* and It is necessary to ensure that all segments are logically connected. The segment which is holding the child state in original global automaton is pointed by those *dummy accept states.* We are using following algorithm for query segment encryption

Algorithm:

```
Input:  Automaton state S

Output:  Segment address i.e. add

For each symbol k in S.StateTransTable do

add=deploySegment(S.StateTransTable(k).nextState)

DS=createDummyAcceptState()

DS.nextState ←add

S.StateTransTAble(k).nextState←DS

End for

Seg=createSegment()

Seg.addSegment(S)

Coordinator=getCoordinator()

Coorinator.assignSegment(Seg)

Return Coordinator.add
```

2)Deployment: Different coordinators are used to hold multiple logical segments. But the no of required coordinators should be less. To achieve this task different segments are deployed on same coordinator. This is done with the help of port numbers and to identify the unique segment the tupple <coordinator, port>is used. Once the deployment is over the tree structure of the coordinator is formed. According to the relative position of the segment, coordinators are connected to create the tree structure.

In this tree then, root is one of the coordinator which is holding the root state of global automaton and leaf nodes are those nodes who are holding the accept states. Then the query is processed with the help of this coordinator tree. The process starts from root coordinator and first at all the XPath step of query and tokens at the root coordinator are compared. If match found then query is send to the next coordinator until it is accepted by leaf coordinators. In case, if the input XPath step does not matched with the stored tokens then such queries are immediately rejected and dropped.

3) Replication: Replication is mainly used in distributed system. All queries are first processed at root coordinator; it can be the reason of failure and also reduces performance. To overcome such problem we need to replicate the root coordinator and the coordinators at higher level in trees. Central Authority (CA) performs very vital role in this process.CA is responsible to create and to revoke the replicas. The set of replicas of the coordinator i.e. maintained by CA. This set of replicas can be predefined or dynamically managed as per the requirement.

*4) Handling the Predicates:*To handle the predicates NFA is used. Each of the NFA states has a predicate table. This predicate table is connected to the each of its child state and stores the predicate symbol (psymbol) .*Lookup-and-Attach* method is used to handle the predicates.

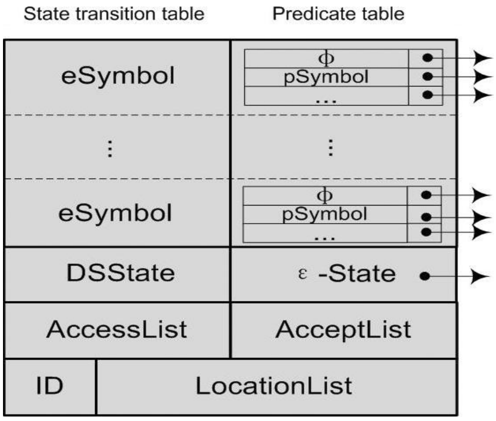

Fig3. Data structure of NFA state

Example: Consider the query as:

Q=/site/regions//item [@id="30"]/name

In the following fig. it is shown that how the query is processed.

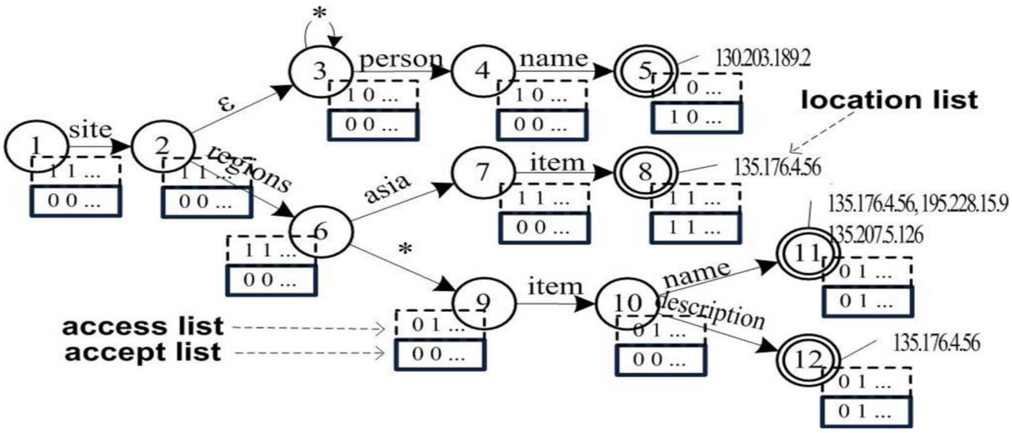

Fig.4.The state transition graph

The given query is accepted at the state11.after that it is rewritten into safe query and then it is passed to three different data servers. As shown in fig. from index rule, the data servers having the address as 195.228.155.9 and 135.176.4.56 are not holding the requested data. In such case routing the query to those data servers can create more workload i.e. overhead. To overcome this problem we are going to introduce a new scheme to handle the Value-based predicates in XML query.

Structural predicates (containing twigs) are not handled by this scheme. so that data structure of the original NFA state is changed. This is done by adding new fields to the predicate table i.e. condition, location, and type. And p symbol stores predicate token, condition holds the test condition, type $\in$ {R, I}.It denotes that the predicate is from ACR or from index rule. Location holds the address of index rule. During the processing, if XPath step of query does not holding any predicate then the processing is done in *lookup-and-attach* way.

If there is the predicate in XPath step, then the record of the predicates are retrieved from predicate table and then it is send to the *predicate directory server*. Then test cases are evaluated. In this testing process, to determine that if the query passes access control testing, the predicates in the query are first compared with ACR predicates. Then Query predicates are compared with predicates which are given by the index rule. If accepted query is assigned to multiple location lists then it is send to the data servers in the list.

B. Query Segment Encryption:

One can guess the information which is requested from the query content. So it is vary essential to hide this query content from other brokering components. For this purpose we are comparing the query content with index rule and access control rule for authorization and to rout the query among multiple components.

The *automaton segmentation* scheme allows to divide the query into small pieces. Then this piece of information is encrypted. This scheme also allows each coordinator to decrypt the corresponding piece of information. *Pre-encryption, Post-encryption and commutative encryption* methods are used.

1)Level based pre-encryption: Using automaton segmentation scheme, each query is processed by set of coordinators with the help of coordinator tree. If query is encrypted using public key of coordinator then it gives the guarantee that each query segment is decrypted and processed by the required component. So each coordinator does not have enough information for inference, but when they are collaborated then there are chances of inference. To address such problem, the piece of information is encapsulated based on static and publicly known information i.e. global schema. The tree structure is generated from the XML schema. In this tree, level of the node is defined as its distance from the root node. The XPath step in the XPath expression is associated with the level *i*, if and only if the corresponding node in the tree is at level *i* position. The nodes positioned at the same level share the pair of keys i.e. public and private keys {*pk, sk*}. After Automaton segmentation is over, each segment are assigned a private key of level *i*, sk_i. In the *pre-encryption,*XPath steps of the query is encrypted from root coordinator with the help of public level keys{pk_1,pk_2,.....pk_n}.If coordinator have multiple segments which contains XML nodes of *k* levels, then it is necessary to decrypt the first *k* unprocessed XPath steps of the query.

2)Post-Encryption:*The query segments which are already processed should be hidden from other coordinators in later processing for this purpose *post-encryption* is required. In normal scheme, it is assumed that all data servers share the pair of public and private keys{pk_{DS},sk_{DS}} and pk_{DS} is known to all the coordinators. Firstly the query segment is decrypted with private key, then authorization and indexing is performed .After that corresponding segment is encrypted.

*3)Commutative Encryption:*Generally this type of encryption is used for '//' handling. When query gas descendent or self axis then it is called as mismatching problem is occurred at coordinator which takes the '//' XPath step as input. Because '//' XPath step can recursively accept multiple tokens until match is found. Consider the following example.

In this, new commutative level key e_i is assigned to each level and it is commutative with original key. Then the commutative level key of level *i* shared by the nodes positioned at *i+1* level. The commutative encryption is performed at the first coordinator at which XPath step in the query is found and when NFA state of coordinator matches the token"//" then the process is ended. To denote the stage of the query, a flag $\in$[0, 3] is attached to query.

VI. MODULES

A. Admin module:

Admin module performs vary important role like registration of data owner, data requestor, broker and coordinator. It also registers the organization and manages the database. In our system it manages the database based on account holder details and its records.

B. Coordinator module:

It operates in between two end users i.e. data owner and data user. Firstly data owner submits the details of account holder. Data user searches for the data which is required and stored at data server. The sends request for receiving the required data. Then coordinator sends the key to the data user and data is received via the brokers. Usually they are responsible for access control and query routing.

C. Broker module:

The broker module acts in between the coordinator and data users. The request sent from data requestor is verified by the broker and then it is forwarded to different coordinators. Again the data is passed from the coordinator and it is given to end user. Usually it is responsible for user authentication and query forwarding.

D. User module:

The user can be of two types i.e. data user and data owner. The data users are requesting for some data And data owner are those whose records are maintained in the database. The coordinator gives the requested data to data user through brokers. Then this data is verified with the help of secret key and then it us available for the user.

VII. MAINTENANCE

A. Key management:

Central Authority (CA) is considered as highest level of trust. It is responsible for key maintenance. CA has the global view related to all rules so than only CA has the idea of those rules. It performs important role in the automaton segmentation and key management. In this system four types of keys are required. i.e. Query session key K_Q, Public and private level keys {*pk,sk*}, commutative level key {*e, d*}, public and private level keys for data servers{$pk_{DS,}sk_{DS}$}.The query session key K_Q *is* create by the user and remaining three types of keys are created and also maintained by the CA.The data servers are considered as the unique party and they shares the pair of private and public keys. Each of the coordinator has their own keys. During the automaton segmentation and deployment process, for the coordinators at each level CA generates pair of key and assigns the private key to the segments.

B. Brokering server Join or Leave:

Brokers and coordinator are the important brokering components. Both of them are allowed to join and leave the system dynamically as per the requirement.AS we have seen earlier, local brokers are the entry point for the system. It is used for storing the address of root coordinator and their replicas, and then forwards the query among multiple coordinators. When new broker wish to join the system, firstly he need to inform to CA and then receives the current address list from CA and sends his own address to local brokers. When any one of the broker wants to leave the system it needs to only send the leave message to all the local brokers. This join and leave process is simple and easy for the brokers but it is somewhat difficult in case of coordinator. When new coordinator wants to join the system, he needs to send the join request to the CA.Then CA authenticates it and assigns the automaton segment it. For this, it is necessary to consider the trust level and the load balance requirement of that coordinator. Then CA gives the corresponding private level keys and to update the location list which is attached to the parent coordinator it send the *serverJoin(add)* message. When coordinator wish to leave the system, it informs to the CA that he want to leave the system. Then CA makes decision that, weather to use existing coordinator or to take new one in place of that leaving coordinator. This decision is taken based on the heuristic rules and curentnload of the coordinators.

Then to replace the leaving coordinator to another, CA broadcasts *serverLeave(add1,add2)* message. Also CA considers the failure coordinator as the leaving coordinator and replaces it with new one.

C. *Metadata update:*

If changes are done in the access control policy or in the data distribution of the organization, it is necessary to update the ACR and index rules.

1) Index rule: The local server needs to send the update message to add or to remove the set of data objects, for this it sends the message in form of *DataUpdate(object,address,action)* to CA.where object is XPath expression and it is used to describe the set of XML nodes, Address is location of the data object, and action is the action which is to be performed it can be 'add' or 'remove'.CA sends the update message to root coordinator to add the data object. Then message passes through the coordinator network until reaches to the leaf coordinator. At that time address is putted on location list.Simillarly data object is removed and also address is removed from the location list.

2) Access Control Rule: If any changes are done in the access control policy then it is described by set of positive or negative access control rule. So the message *ACRUpdate(role,object,type)* is sent to the CA. CA is responsible for forwarding this message to the root coordinator. From that root coordinator, the *object* in XPath expression is processed by the each coordinator. This is done with help of state transition table. Then generates the new automaton with new ACR as: if at any particular NFA state message stops, then that state is changed as accept state for that rule. After that all of the leaf and descendent coordinators are retrieved and then location list is attached to the accept state. It will generate the new automaton segment and it will assign his segment to the new coordinator if message is accepted by the existing leaf coordinator. Finally location list at the original leaf coordinator is copied to the new coordinator.

VIII. ADVANTAGES

1) PPIB system is efficient and scalable because end-to-end query processing performance and system's scalability is evaluated.

2) PPIB system provides the security to the data.

3) It performs query routing while preserving the privacy.

4) Load balancing and site distribution is handled by the system.

IX. CONCLUSION

Privacy issues of user and data during the design stage 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. We claim that our 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.

Many directions are ahead for future research. First at present, 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.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

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

[5] N. Koudas, M. Rabinovich, D. Srivastava, and T. Yu, "Routing XML queries," in *Proc. ICDE'04*, 2004, p. 844.

[6] S. Rizvi, A. Mendelzon, S. Sudarshan, and P. Roy, "Extending query rewriting techniques for fine-grained access control," in *Proc. SIGMOD'04*, Paris, France, 2004, pp. 551–562.

[7] T. Yu, D. Srivastava, L. V. S. Lakshmanan, and H. V. Jagadish, "Compressed accessibility map: Efficient access control for XML," in *Proc. VLDB*, China, 2002, pp. 478–489.

[8] B. Luo, D. Lee, W. C. Lee, and P. Liu, "Qfilter: Fine-grained runtime XML access control via NFA-based query rewriting enforcement mechanisms," in *Proc. CIKM*, 2004, pp. 543–552.

[9] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Proc. CRYPTO'07*, Santa Barbara, CA, USA, pp. 535–552.

[10] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in *Proc. ICDCS*, Minneapolis, MN, USA, 2011, pp. 383–392.

[11] E. Bertino, S. Castano, and E. Ferrari, "Securing XML documents with author-x," *IEEE Internet Comput.*, vol. 5, no. 3, pp. 21–31, May/Jun. 2001.

[12] G. Koloniari and E. Pitoura, "Content-based routing of path queries in peer-to-peer systems," in *Proc. EDBT*, 2004, pp. 29–47.

[13] A. Carzaniga, M. J.Rutherford, andA. L.Wolf, "Arouting scheme for content-based networking," in *Proc. INFOCOM*, Hong Kong, 2004, pp. 918–928.