



Research Manuscript Title

AN ENHANCED FINE GRAINED MODEL FOR VIEWING XML DOCUMENTS

J.Jamila¹, M.Annapoorani²

¹PG Scholar, ²Assistant Professor,
M.E. Department Of Computer Science And Engineering
Institute Of Road And Transport Technology,
Erode-638 316,Tamilnadu,

E-Mail: jamibtecit@gmail.com

JUNE – 2016

www.istpublications.com

AN ENHANCED FINE GRAINED MODEL FOR VIEWING XML DOCUMENTS

J.Jamila¹, M.Annapoorani²

¹PG Scholar, ²Assistant Professor,
M.E. Department Of Computer Science And Engineering
Institute Of Road And Transport Technology,
Erode-638 316,Tamilnadu,

E-Mail: jamibtecit@gmail.com

ABSTRACT

We introduce an XML access control enforcement mechanism that exploits the advantages of both pre-processing and view-based mechanisms. The fine-grained views that are defined for access control rule(s), instead of users/roles. With the increasing usage of XML on information sharing over the Internet, a mechanism for defining and enforcing XML access control is demanded, such that only authorized entities can access the sets of XML data that they are allowed to. The research interests in these areas have grown significantly in recent years. Various access control enforcement solutions have been proposed, each with its inherent advantages and disadvantages. Yet, there is still no solution that can provide superior performance in all situations. In this paper, we present hyxac, a hybrid approach to enforce XML access control. Hyxac integrates the two most popular categories of XML access control enforcement mechanisms, and earns the benefits from both. In particular, hyxac first preprocesses user queries by rewriting queries and removing parts violating access control rules, and evaluates the re-written queries using sub views, if they are available. In hyxac, views are not defined on a per-role basis. Instead, a sub-view is defined for each access control Rule, and roles sharing identical rules will share sub-views. Moreover, hyxac dynamically allocates memory and secondary storage resources to materialize and cache sub-views to improve query performance. We have conducted extensive experiments, and the results show that hyxac improves query processing efficiency while optimizes the use of system resources.

Keywords: XML, access control, view

I. INTRODUCTION

Generally, data mining is the process of analyzing data from different perspectives and summarizing it into useful information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or pattern among dozens of fields in large relational databases. Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis. Data mining uses sophisticated mathematical algorithms to segment the data and evaluate the probability of future events. Data mining is also known as Knowledge Discovery in Data (KDD).

The key properties of data mining are:

- Automatic discovery of patterns
- Prediction of likely outcomes
- Creation of actionable information
- Focus on large data sets and databases

II.BACKGROUNDANDPRELIMINARIES

Secure and Selective Dissemination of XML Documents[1].XML (eXtensible Markup Language) has emerged as a prevalent standard for document representation and exchange on the Web. It is often the case that XML documents contain information of different sensitivity degrees that must be selectively shared by user communities. Approach consists of encrypting different portions of the same document according to different encryption keys, and selectively distributing these keys to the various users according to the access control policies. In this paper, show that the number of encryption keys that have to be generated under this approach is minimal and present architecture to support document distribution.

Compressed accessibility map: Efficient access control for XML[2].A compressed accessibility map (CAM) as a space and time-efficient solution to the access control problem for XML data. A CAM compactly identifies the XML data items to which a user has access, by exploiting structural locality of accessibility in tree-structured data. Secure XML Querying with Security Views[3].A novel paradigm for specifying XML security constraints and investigates the enforcement of such constraints during XML query evaluation. The approach is based on the novel concept of security views, which provide for each user group an XML view consisting of all and only the information that the users are authorized to access. Security views effectively protect sensitive data from access and potential inferences by unauthorized users, and provide authorized users with necessary schema information to facilitate effective query formulation and optimization. XML Access Control Using Static Analysis[4].An access control policy, query expression, and an optional schema, static analysis determines if this query expression is guaranteed not to access elements or attributes that are permitted by the schema but hidden by the access control policy. Static analysis can be performed without evaluating any query expression against an actual database. Run-time checking is required only when static analysis is unable to determine whether to grant or deny access requests.

III.EXISTING SYSTEMS

A. Engine-level methods

Engine-level methods tag each node with an accessible-users-list and incur high overhead on the XML engines.

B. View-based approaches

Create for each role a view consisting of all the accessible data and evaluate queries against the view. Create a separate Copy of data for each role. A view contains exactly the set of Xml nodes that can be accessed by the role.

C. Pre-processing approaches

User queries are first modified into “safe” ones requesting only authorized data before they are answered against the XML document.

D. Post-processing methods

Evaluate the queries on the database, get “unsafe” results, and then discard all the unauthorized nodes.

E. Disadvantages

- View maintenance becomes an issue.
- It is non-trivial to maintain and synchronize a large number of views.
- Query processing could be slow when the documents are large especially when caching or indexing is not we will supported in the xml DBMS.
- High redundancy of data storage
- Query performance is low.

IV PROPOSED SYSTEM

A. Fine-grained view

- Fine-grained view for every positive access control rule (ACR). Hence, accepted queries are evaluated against the corresponding fine-grained views
- Fine-grained view management allows sub-views to be shared across multiple roles, thus reducing the number of views stored in the DBMS.
- Hyxac’s fine-grained views are created in a per-rule basis, instead of per-role basis.

B. Dynamic View Management Scheme

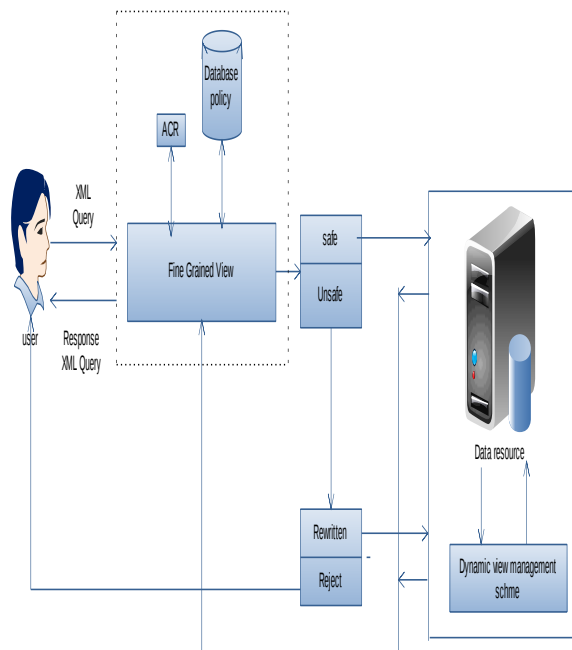
- Hyxac employs a cost-benefit model and greedy approximation to implement a dynamic view management scheme, so as to achieve an optimal query performance under the constraints of the available resources.

C. Advantages

- It provides efficient access control and query processing.
- Maximizes the utilization of available resources with dynamic view management.
- Exploiting the advantages of several access control enforcement mechanisms.
- Reducing the number of views stored in the DBMS.
- Hyxac is designed for query-answering scenario, and attempts to integrate both access control and view-based query processing.
- Maintaining the view's easy compare to the existing system.
- Query processing doesn't affected by size of the document.

V SYSTEM ARCHITECTURE

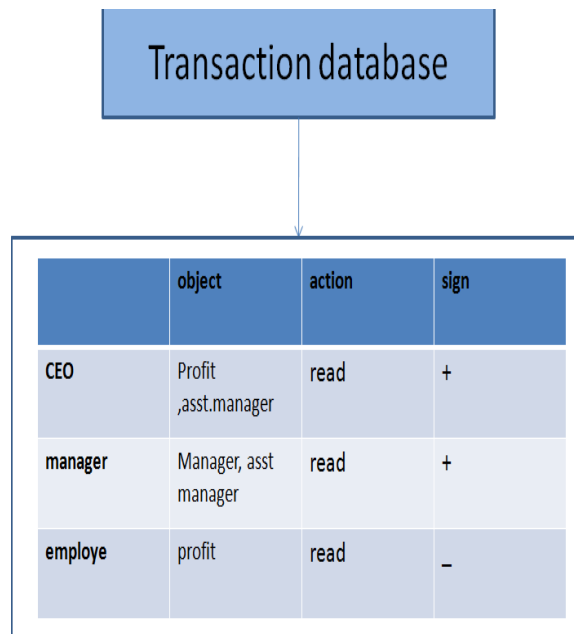
System design is the process of defining the architecture, components, modules, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to hybrid xml development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. System design is therefore the process of defining and developing systems to satisfy specified requirements of the user.



VI MODULE DESCRIPTION

A. ACR Vs Transaction database

In this module we create an access control rule for the each attribute in the transaction database. The access control rule that contain the subject, object, action, sign .where the subject denotes the role who is authorized (or denied) to access the data, the object is a set of attributes to be accessed, the action is the operation that can be performed on the object by the given subject (ex: read, write, update etc.) and the sign specifies whether the action can be performed on the object or not.



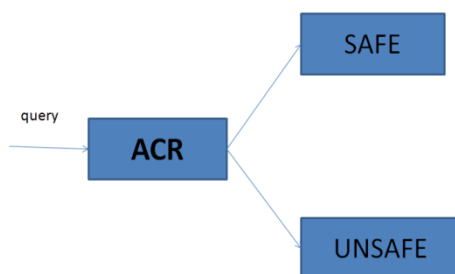
B. User searching into query

In the module we got an input as user enter searching words, then we convert the user searching into the query. After converting the query, then only we can process the query and retrieve the data for the user searching in the database



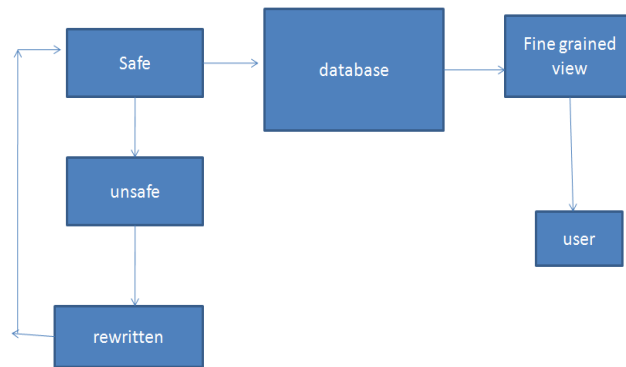
C. Analyzing the query

In this module the queries are analyzed by the set of access control rules (ACR) in the database. The query that satisfy the access control rule in the database that are safe queries the query are not satisfy the access control rule mean that are unsafe queries



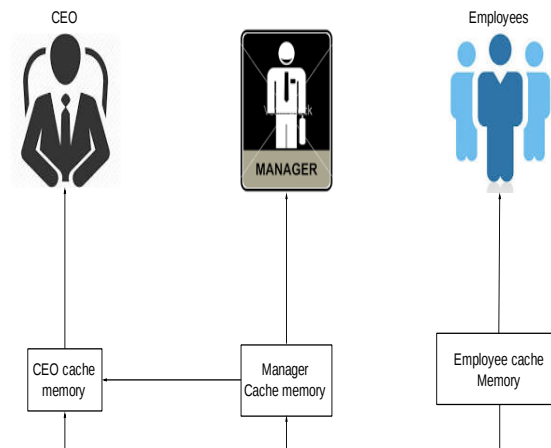
D. Data retrieval

In the module the safe queries are processed in the data base based on the query retrieve data from the data base as fine grained view. Here, the unsafe queries are rewrite into the safe one and then retrieve the data from the database for that query.



E. Improve the query performance using cache memory:

In this module we use the cache memory to improve the query performance. using cache memory under the ACR(Access control rule).Because in organization environment CEO was having rights use their own, employees and manager cache memory, Managers use their own and employee cache. The employees are having access only its own cache memory. By using the cache memory we avoid the process delay to fetch the data from the database. It improves the query performance of the hybrid xml scheme.



VII CONCLUSION

- Hyxac exploit the advantages of two different types of XML access control mechanisms.
- Answers the safe queries with fine-grained views.
- Improve the query performance using the cache memory.

IX REFERENCES

1. E. Bertino and E. Ferrari, "Secure and selective dissemination of XML documents," ACM Trans. Inf. Syst. Security, vol. 5, no. 3, pp. 290–331, 2002.
2. M. Kudo and S. Hada, "Xml document security based on provisional authorization," in Proc. ACM 7th Conf. Comput. Commun. Security, 2000, pp. 87–96.

3. T. Yu, D. Srivastava, L. V. S. Lakshmanan, and H. V. Jagadish, "Compressed accessibility map: Efficient access control for XML," in Proc. 28th Int. Conf. Very Large Databases, China, 2002, pp. 478–489
4. W. Fan, C.-Y. Chan, and M. Garofalakis, "Secure xml querying with security views," in Proc. ACM SIGMOD Int. Conf. Manage.Data, 2004, pp. 587–598.
5. M. Murata, A. Tozawa, M. Kudo, and S. Hada, "Xml access control using static analysis," in Proc. ACM Conf. Comput. Commun. Security, 2003, pp. 73–84.
6. ManognaThimma, Fang Liu, Jingqiang Lin and Bo Luo "HyXAC: Hybrid XML Access Control Integrating View-Based and Query-Rewriting Approaches" in IEEE Transactions on knowledge and engineering, VOL 27, NO.8, AUGUST 2015.
7. R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the generalized assignment problem," Inf. Process. Lett., vol. 100, no. 4, pp. 162–166, 2006.
8. C. Chekuri and S. Khanna, "A polynomial time approximation scheme for the multiple knapsack problem," SIAM J. Comput., vol. 35, no. 3, pp. 713–728, 2005.
9. Y. Diao and M. J. Franklin, "Highperformance XML filtering: An overview of YFilter," IEEE Data Eng. Bulletin, vol. 26, no. 1, pp. 41–48, Mar. 2003.
10. M. Thimma, T. K. Tsui, and B. Luo, "Hyxac: A hybrid approach for xml access control," in Proc. ACM 18th Symp. Access Control Models Technol., 2013.
11. H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems. New York, NY, USA: Springer, 2004.
12. J. Foster, B. Pierce, and S. Zdancewic, "Updatable security views," in Proc. 22nd IEEE Comput. Security Found. Symp., Jul. 2009, pp. 60–74.
13. B. Luo, D. Lee, W.-C. Lee, and P. Liu, "Qfilter: Fine-grained run-time xml access control via nfa-based query rewriting," in Proc. ACM Conf. Inf. Knowl. Manage., Washington, D.C., USA, Nov. 2004, pp. 543–552.
14. G. Kuper, F. Massacci, and N. Rassadko, "Generalized xml security views," in Proc. 10th ACM Symp. Access Control Models Technol., 2005, pp. 77–84.