Performance Evaluation of Trust Based Access Control for XML Databases

Norah Farooqi, Siobhan North
Department of Computer Science
The University of Sheffield
Sheffield, United Kingdom

Abstract

In order to improve security and provide dynamic access control for XML databases, we developed trust based access control for XML databases. Trust based access control for XML databases manages the access policy depending on users' trustworthiness and prevents unauthorized processes, malicious transactions and misuse from both outsiders and insiders. Trust scores are updated on the basis of users' histories. Privileges are automatically modified and adjusted over time depending on user behavior. In this paper, a practical trust based access control module for XML databases is evaluated. The dynamic access control has been tested from security, scalability, and performance perspectives. The experimental results illustrate the flexibility of trust values and the scalability of the system with small to large XML databases and with various numbers of users.

1. Introduction

The Extensible Markup Language (XML) is commonly used to store, transfer, present, and retrieve data in many applications. In all applications and especially in platforms such as business and medical applications, XML databases can contain a huge amount of sensitive, personal, and important data. These data should be handled in a secure manner to prevent loss or misuse. Thus, security research in XML databases is important and an active area that needs further investigation.

One of the main approaches to guarantee the security in any system is to apply access control. The access control model manages access to data and prevents unauthorized processes. Many access control models have been proposed and used. The traditional types are discretionary access control (DAC), mandatory access control (MAC), and role base access Control (RBAC) [1-5]. There are many other types that are non-traditional, such as function based access control and purpose based access control [6, 7]. Some of these models have been applied to provide a secure environment for XML databases. Although each model is designed differently to manage and implement the access policies, it seems that most of them prevent unauthorized access and misuse from outsiders. XML databases, like other databases, may also face internal threats and these can be more dangerous because insiders know the system better than outsiders [8]. Internal users may abuse their role and take advantage of their position in the system.

Trust based access control has become established in many applications [9-13]. It uses a trust management system that automatically calculates users' trust values. The trust values are updated according to an evaluation of the user's history. In our previous work [14-19], we proposed the use of trust based access control for XML databases which could provide dynamic access control and solve misuse problems from both outsiders and insiders.

In this paper, we discuss the implementation of our trust based access control approach for XML databases and the practical difficulties encountered. The system consists of two modules: a trust module and an access control module. The key idea in this research is to test each module separately and evaluate how the entire system works together in terms of scalability and performance. We improve and extend the trust calculation to four equations; each of which is used for specific cases. The system processes are categorized into two main functions: access supervision and trust maintenance. Different experiments were run to evaluate the system and find its limitations.

The remainder of the paper is organized as follows. Section 2 describes both the trust module and the access control module and explains the processes cycle in the entire system. Section 3 explains the evaluation of both modules and the whole system. It illustrates the results of several experiments that have been run with different size databases and different numbers of users. Section 4 forms the conclusion and suggests future work.

2. A practical trust based access control module

In this section, a practical trust based access control module for XML databases is described. This

system is dynamic and responsive to users' history of errors and bad transactions. The module consists of two main parts: the trust module and the access control module. The trust module is responsible for recording errors and bad transactions, evaluating them, and calculating the new trust value. The access control module is responsible for the access permission policy and access decisions. Both modules are explained in detail and how each works separately is described. Then the whole system combining both modules is explained.

2.1. The trust module

The trust module is constructed of many parts that work together to achieve the main goal of calculating users' trust values. These parts consist of an operation recorder, an error detector, an operation evaluator, and a trust calculator. The operation recorder records errors and bad transactions in the Xlog file [16]. The Xlog file is designed to be dynamic and to be stored temporarily for a set period to reduce storage and improve searching performance. Both the error detector and operation evaluator work in the light of the error policy file and the operation policy file. Each of these policy files has the role of defining what an error or a bad transaction is. The trust calculator uses the data recorded by both the error detector and the operation evaluator. The main goal of the trust calculator is to compute a new trust value. The trust value depends on the user history of malicious bad transactions and error factors.

The new Trust Value (TV) is generated using three values: Existing Trust Value (ETV), Bad Transaction Factor (BTF), and Error Factor (EF). Each factor is multiplied by a weight that reflects the importance of the factor in the system and shows to what extent the factor affects the final Trust Value (TV). Each weight is a percentage that shows how much the factor will affect the general equation and the new Trust Value (TV). The weights are Existing Trust Value Weight (ETVW), Bad Transaction Factor Weight (BTFW), and Error Factor Weight (EFW).

Both Error Factor Weight (EFW) and Bad Transaction Factor Weights (BTFW) range is between 1 % and 10%. The Existing Trust Value Weight (ETVW) range is between 80% and 99%. Range values are selected to keep the Trust Value (TV) within suitable bounds. The maximum for both error and bad transaction weights is 10% and not higher because the aim of the system is to adjust user privilege according to behavior and not to block user access completely. For example, if this weight was high, such as 70%, then the Trust Value (TV) would drop suddenly and dramatically and may cause other access problems. The Existing Trust Value Weight (ETVW) is regarded as the basic value to calculate the new Trust Value (TV). The new Trust Value (TV) is derived from the previous existing one and this explains why this weight should be in the range between 80% and 99%.

The Trust Value (TV) increases when there are no bad transactions or errors but it drops markedly when the Bad Transaction Factor (BTF), Error Factor (EF), or both increase. There are four different equations to calculate the Trust Value and each one applies to specific cases. Trust Value equations are:

Where EF=0 and BTF=0 then
TV=ETV*ETVW+ (1-EF)*EFW+ (1-BTF)*BTFW. (1)

Where EF>0 and BTF=0 then
TV=ETV*ETVW-EF*EFW. (2)

Where EF=0 and BTF>0 then
TV= ETV*ETVW-BTF *BTFW. (3)

Where EF>0 and BTF>0 then
TV=ETV*ETVW-EF*EFW-BTF*BTFW. (4)

Equation 1 is used to calculate Trust Value (TV) when there are no errors or bad transactions and this increases the Trust Value (TV) slightly. If there are errors or bad transactions, (2) or (3) are used to calculate TV. In general, if there are errors or bad transactions the TV should decrease in all cases. As a consequence, the Error Factor (EF) is subtracted from the Existing Trust Value (ETV) in (2). The same principle applies to (3) when there are only bad transactions without errors; the Bad Transaction Factor (BTF) is subtracted from Existing Trust Value (ETV). Equation 4 is used when there are both errors and bad transactions. It subtracts both Error Factor (EF) and Bad Transaction Factor (BTF) from Existing Trust Value (ETV) to find the new Trust Value (TV). Table I illustrates the calculation of Trust Value for some general cases when the Existing Trust Value (ETV) = 0.5.

Table1. The calculation of trust value.

ETV	ETVW	EF	EFW	BTF	BTFW	TV
0.5	85%	0	5%	0	10%	0.575
0.5	85%	0.25	5%	0.25	10%	0.387
0.5	85%	0.5	5%	0.5	10%	0.350
0.5	85%	0.75	5%	0.75	10%	0.312
0.5	85%	1	5%	1	10%	0.275

2.2. Access control module

The access control module consists of the access manager and the access decision maker. The access manager deals with access permission policies that primarily depend on Trust Value (TV). These policies are divided into subject policy by assigning Trust Value (TV) to the subject and object policy,

which concentrates on giving each item of data the appropriate Trust Value (TV).

The access decision maker handles the XML query and then either permits or denies the request. The final decision depends directly on defined access permission policies in the access manager. The Trust Value (TV) of the user is compared with the Trust Value (TV) of the required XML data. If the Trust Value (TV) for users equals or is larger than the Trust Value (TV) of data, then the user is allowed to access the data; otherwise access is denied.

2.3. The entire system

The trust module is connected to the access control module to form the whole system. This combination works to provide a secure environment to access XML databases depending on the evaluation of the users' history. Combining these two modules to work together generates the system processes that relate to both modules at the same time. These processes can be characterized into two main classifications: access supervision and trust maintenance. The access supervision process is always run for each access into the system. The trust maintenance process occurs frequently, depending on the organization's policy for updating the users' Trust Value and making the system responsive, therefore it could run weekly, daily or hourly.

The access supervision process contains a series of small processes. This series starts by accessing XML data and detecting both errors and bad transactions, then recording them in the log. The trust maintenance process also consists of a sequence of small processes. These processes evaluate errors and bad transactions, calculating a new Trust Value and updating users' privileges. The whole system is explained in Figure 1. In the following section, the results of using this approach are discussed.

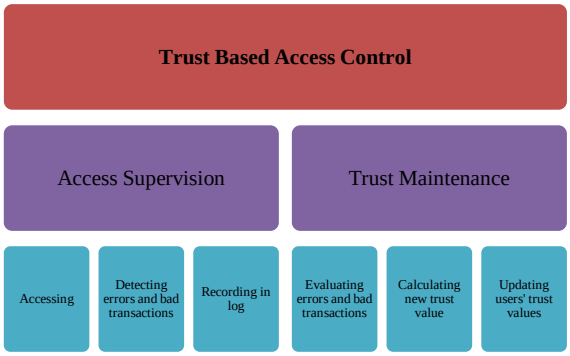

Figure 1. The system's processes.

2.4. The objectives of policy files

The policy files contain rules. In this section, the goals of using policy files are described. Policy of the organization owning the data will differ from one organization to another. Organizations that hold a lot of sensitive and personal data have strict policy rules to provide a high level of security. Other organizations may be not concerned with security issues but they may value processes speed and so need more flexible and simple policy rules.

In this experimental trust based access control for an XML databases system, some very basic policy rules are recorded in policy files but they can easily be extended to cover more complicated policies. The system divides the policy rules into sub rules and records them in individual policy files. The idea of these divisions is to make the policy rules clear and easy to update and change. Each policy file contains a group of rules related to a specific part in the system.

The policy file is a standard XML file and has defined tags that related to the policy rules. Policy files are written in XML. There are other defined access policy languages but they do not fit with trust based access control properties. Furthermore, XML is the original and other access languages are derived from it. Using XML makes the policy file clear, easy to use, and flexible. At the same time, ensuring the file type is consistent with the rest of the system.

There are five different policy files in the system: error, operation, trust, XML databases' access permission, and user access permission. The error policy file and the operation policy file have the rules to define an error and a bad transaction and are used by the error detector and the operation evaluator to capture errors and bad transactions. The trust policy file provides the rules to calculate the trust value for the trust calculator. Both the XML database access permission policy file and user access permission policy file are required by the access manager. The XML databases' access permission policy file contains the trust rules to access the XML databases by assigning each node an appropriate trust value. The user permission access policy file contains the trust value for each user in the system. This file is relatively dynamic because the trust values for users change over time according to their behavior.

2.5. Boundary management

Boundary management aims to define basic rules to avoid anomalous situations that may arise in the system over time. As in any database's system, the administrator manages the policy rules mentioned in the previous section and assigns the appropriate trust values for data and initial trust values for users based on their role. The administrator must be authorized to handle any exceptional circumstances that may occur in the system.

One of the critical points in the system is handling calculation trust value for users. Since, the trust

value is dynamic and updated over time; it may cause other access problems such as a trust value dropping until it blocks the user's access completely. The boundaries for trust values are designed to control the change in the trust value and to make sure the user's access is with the appropriate ranges.

There are two main boundaries: the maximum and the minimum boundaries. Both values are defined according to the organization policy and the role of users in the system. The maximum value is the highest trust value that the user can reach and the minimum value is the lowest trust value the user can fall to.

These boundaries control the powers and permissions of users' access in the system. For example, the trust value for the manager can be between a maximum value 1 and a minimum value 0.75 (1>= TV>=0.75). Within this range the trust value for the manager can change according to behavior. The trust value can increase to the maximum or decrease to the minimum. Boundary management can be easily improved to cover other organizations' strategies to handle future risks that the system may face.

3. Experiment results

In this section, we describe our experiments with a practical trust based access control module for XML databases. These experiments were completed in different stages. First, we tested each of the trust module and the access control module separately and analyzed the results. Then we tested the whole system to discover how both modules work together. The main goal of our experiments is to evaluate the system performance and scalability.

All experiments were performed on a PC with 2.40 GHz Intel® Core™ i5 CPU, 4 GB of main memory, and Windows 7 operating system. The system is implemented using Java Language (JDK 1.7.0) and Net Beans IDE 7.0.1 platform framework.

We used three real-world XML databases with different file sizes, structures, and depths. This is because real-world datasets are simple but contain realistic data whereas the data in benchmarks are synthetic. The benchmarks are usually used to test the management systems of XML databases and focus on storage techniques and query processing. Since this system handles security issues, using natural datasets is more appropriate.

These databases used were SIGMOD Record, NASA, and Treebank from [20]. SIGMOD Record includes real data for articles published by the ACM SIGMOD website. NASA contains genuine astronomical data. It is part of the GSFC/NASA XML Project. Treebank includes English sentences annotated for linguistic structures. This database is partially encrypted but it does not affect the XML structure at all. This dataset is considered an interesting case for evaluation experiments due to its deep recursive structure. In addition, these experiments were run with a number of user access permission policy files with different numbers of users. Table 2 shows the features of each dataset used in the experiments.

Table2. Details of the datasets.

Database name	Database size	Number of nodes	Number of levels
SIGMOD Record	467 KB	15263	6
NASA	23 MB	532963	8
Treebank	82 MB	2437667	36

3.1. Evaluating the Trust Module

The main goal of the trust module experiments is to study the Trust Value performance. Trust Value is changed depending on the Existing Trust Value, Error Factor, Bad Transaction Factor, and their weights. In general, these experiments show how the Trust Value is affected by these factors from two perspectives. The first viewpoint shows the change in Trust Value depending on the error and bad transaction factors. The second point explains how the weight values affect calculation of the Trust Value.

All case studies for the trust module are tested using three starting points for the Existing Trust Value (ETV): 0.75, 0.50, and 0.25. Table 3 shows the change in Trust Value depending on various values of errors without any bad transactions. These Trust Value results when the weights are ETVW= 80%, EFVW= 10%, and BTFW=10%.

Table 4 illustrates the calculations of TV when there is a bad transaction without an error. The results for TV are the same as in Table 3 because the same weights for errors and bad transaction are used. Both the error and bad transaction factor reduce the TV.

Table3. The results with only the Error factor.

ETV	ETVW	EF	EFW	BTF	BTFW	TV
0.75	%80	0.25	%10	0	%10	0.575
0.50	%80	0.25	%10	0	%10	0.375
0.25	%80	0.25	%10	0	%10	0.175
0.75	%80	0.5	%10	0	%10	0.55
0.5	%80	0.5	%10	0	%10	0.35
0.25	%80	0.5	%10	0	%10	0.15
0.75	%80	0.75	%10	0	%10	0.525
0.5	%80	0.75	%10	0	%10	0.325
0.25	%80	0.75	%10	0	%10	0.125

0.75	%80	1	%10	0	%10	0.5
0.5	%80	1	%10	0	%10	0.3
0.25	%80	1	%10	0	%10	0.1

Table4. The results with only the bad transaction factor.

ETV	ETVW	EF	EFW	BTF	BTFW	TV
0.75	%80	0.25	%10	0	%10	0.575
0.50	%80	0.25	%10	0	%10	0.375
0.25	%80	0.25	%10	0	%10	0.175
0.75	%80	0.5	%10	0	%10	0.55
0.5	%80	0.5	%10	0	%10	0.35
0.25	%80	0.5	%10	0	%10	0.15
0.75	%80	0.75	%10	0	%10	0.525
0.5	%80	0.75	%10	0	%10	0.325
0.25	%80	0.75	%10	0	%10	0.125
0.75	%80	1	%10	0	%10	0.5
0.5	%80	1	%10	0	%10	0.3
0.25	%80	1	%10	0	%10	0.1

Figure 2, Figure 3 and Figure 4 demonstrate the variation in Trust Value with different values for both errors and bad transactions. Figure 2 explains the results when the error and bad transaction weights are at the maximum value and the Existing Trust Value Weight is at its minimum, so: ETVW=80%, EFW=10%, and BTFW=10%. The results show how the Trust Value is changed significantly due to the effects of these weights.

Figure 3 shows the results when the error and bad transactions' weights are at the minimum and the Existing Trust Value Weight is at its maximum, therefore ETVW=99%, EFW=1%, and BTFW=1%. Using these weights means the new Trust Value is only slightly affected by errors and bad transactions.

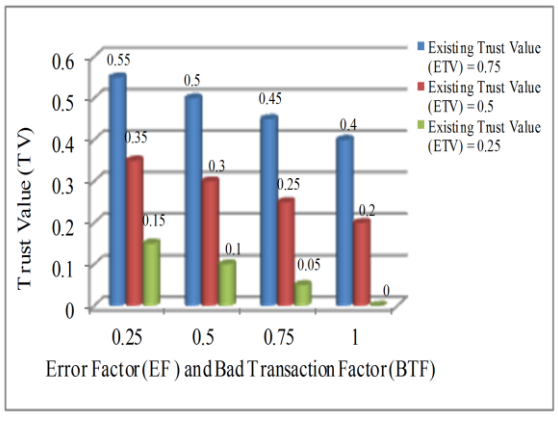

Figure2. The results when errors and bad transactions' weights are at the maximum.

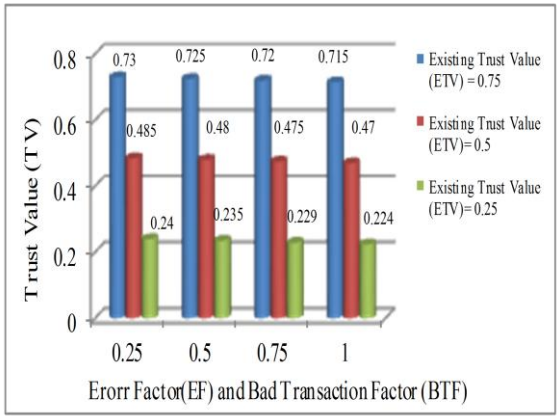

Figure3. The results when errors and bad transactions' weights are at the minimum.

These weights are flexible and can be changed according to the organization's policy. Compared to the results in Figure 2 and Figure 3, we recommend the weights of ETVW=85%, EFW=5%, and BTFW=10%. The Error Factor Weight is selected to be 5% in the middle range value because an error cannot be as harmful as a bad transaction. For the Bad Transaction Factor Weight, we recommend that the weight be the highest value allowed, which is 10%, because a bad transaction could reflect malicious intent. The results of these recommended weights are shown in Figure 4. It appears the Trust Value changes regularly and markedly for all starting points of the Existing Trust Value.

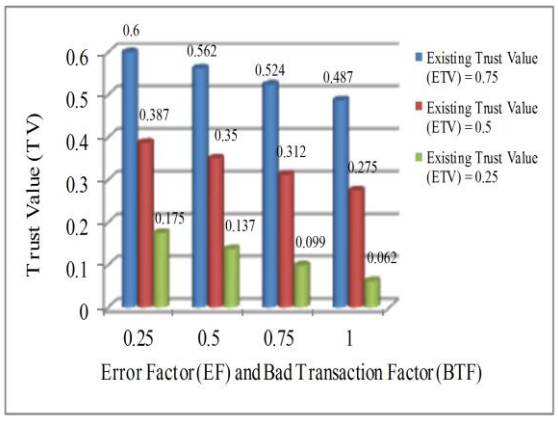

Figure4. The results when errors and bad transactions' weights are at the recommended weights.

Having demonstrated the decrement in the Trust Value (TV) depending on various errors and bad transaction factors, the increment in TV is now discussed. Table V displays the increment in Trust Value (TV) when there are no errors and bad transactions at all. The calculations are made with

the minimum, the maximum and recommended weights.

Table5. The results with only the error factor.

ETV	ETVW	EF	EFW	BTF	BTFW	TV
0.75	%80	0	%10	0	%10	0.8
0.50	%80	0	%10	0	%10	0.6
0.25	%80	0	%10	0	%10	0.4
0.75	%85	0	%5	0	%10	0.787
0.50	%85	0	%5	0	%10	0.575
0.25	%85	0	%5	0	%10	0.362
0.75	%90	0	%5	0	%5	0.775
0.50	%90	0	%5	0	%5	0.55
0.25	%90	0	%5	0	%5	0.325
0.75	%98	0	%1	0	%1	0.755
0.50	%98	0	%1	0	%1	0.51
0.25	%98	0	%1	0	%1	0.265

3.2. Evaluating the access control module

Access experiments were used to evaluate the performance and scalability of the access control module. From a scalability perspective, our access module worked with small to large databases. Similarly, the access module was tested with different sized users' access permission policy files that contain 50, 100, or 1000 users.

To evaluate the access process performance, we measured the access time taken with different databases and different numbers of users. The access time consists of the search time for both the user and data trust values and the retrieval time from the original databases. The time needed to find a user's Trust Value is affected by the number of users. The time consumed to find data Trust Value is slightly affected by the XML databases structure. The retrieval time is mainly affected by the size of the databases. Figure 5 shows the access time for a specific node in three databases with different numbers of users. In general, the total access time is affected significantly by the retrieval time, which means that the final access time is dependent mainly on the size of the database.

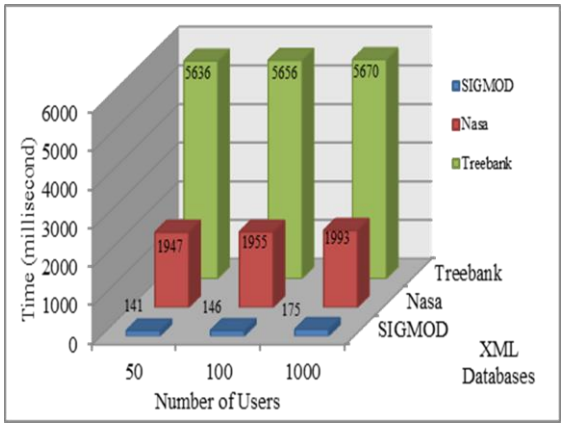

Figure5. The access time using a variety of different sized users

3.3. Evaluating the entire module

In these experiments, we tested the performance of both modules working together. We tested the access supervision process (see section 2.C) by evaluating the time needed to finish this process. The time includes the access time (see previous section), the time for detecting errors and bad transactions, and then the time required for recording in the Xlog file.

The performance of the supervision process was tested through two queries. The first query attempts to delete the root node from the XML databases (Deleting/RootNode). As a result, the access is denied and the operation is recorded in the Xlog file as a bad transaction. The time consumed to complete the supervision process for this query is relatively short because the retrieval time from the original databases does not include any access. The time needed includes the search time for the Trust Values for both users and data and the time required for logging. In this case study, the time is slightly affected by the number of users. For three different datasets, the time consumed is around 87 milliseconds when the number of users is 50. The time is 96 milliseconds for 100 users and around 128 milliseconds for 1,000 users.

The second query is about retrieving a specific node from the XML databases (//NodeName). The supervision process takes longer to execute the second query than the first. The final access decision for the second query is to permit it, so that the access time includes the retrieval time, which depends on the XML database size. The time is between 155 and 183 milliseconds for SIGMOD Record. The time range for NASA is from 2,039 milliseconds with 50 users to 2,086 milliseconds with 1,000 users. The total time significantly increased in Treebank due to the large XML database. It reached 6,101 milliseconds with 1,000 users.

The comparative results of the time consumed for supervising the process in both situations (Q1 & Q2) with 1,000 users are summarized in Table 6.

Table 6. The time consumed for processing the access supervision process

The query type	SIGMOD Record	NASA	Treebank
Q1	126	129	130
Q2	183	2,086	6,101

The trust maintenance process was tested with different numbers of users, errors, and bad transactions factors. The total time includes the evaluation time for errors and bad transactions recorded in the Xlog file, the calculation time and the time to update privileges, which are recorded in the user's access permission policy file. It seems that the time is only slightly affected by the number of users in the system. As a result of making the Xlog file dynamic and retained for a certain period, the errors and bad transaction factors' values in the Xlog file do not affect the time markedly. The time consumed for processing the trust maintenance is explained in Figure 6.

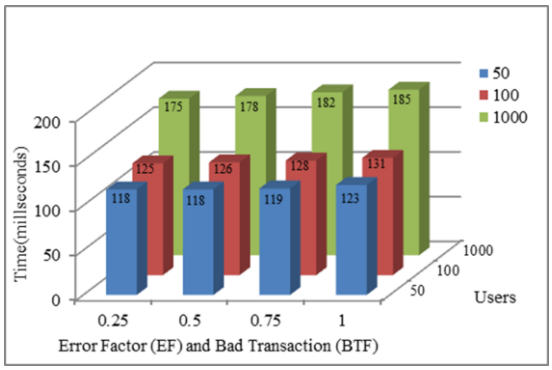

Figure6. The time consumed for processing the trust maintenance process.

4. Conclusions

In this paper, we have evaluated our trust based access control module for XML databases practically in terms of efficiency, scalability, and security. The system provides dynamic access updating users' trust value according to their behavior. It improves the security environment for access to XML databases by detecting misuse from both outsiders and insiders. We first tested the trust module and the access control module individually and then evaluated the whole system performance. To demonstrate the scalability and the performance of our practical system, we executed experiments using different sizes of real XML databases and with different number of users. In general, the experimental results show that the system works and scales well.

The experiments of the trust module show the flexibility of the calculation trust value by changing factors' weights. Access control module results display the access scalability for various XML databases' sizes and various user access permission policies' file sizes. The results of the whole system illustrate that the supervision access process is markedly affected by XML databases' size when the access is permitted since the access time includes the retrieval time. The trust maintenance process result is only slightly affected by the number of users and is not markedly affected by error and bad transactions factors due to the temporal design of the Xlog file, which makes maintenance fast. The results of testing this new dynamic access module for XML databases suggest it is worth persevering with this work. We now plan to compare our practical trust based access control with other existing access control systems.

5. References

[1] M. Hitchens and V. Varadharajan, "RBAC for XML Document Stores", in Information and Communications Security, Lecture Note in Computer Science, vol. 2229, S. Qing, T. Okamoto and J. Zhou, Eds. Springer Berlin/Heidelberg, 2001, pp. 131-143.

[2] J. Wang and S. L. Osborn, "A role-based approach to access control for XML databases", in the Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies, Yorktown Heights, New York, USA, 2004, pp. 70-77.

[3] H. Zhu, R. Jin and K. Lu, "A flexible mandatory access control policy for XML databases", in the Proceedings of the Second International Conference on Scalable Information Systems, Suzhou, China, 2007.

[4] H. Zhu, R. Jin and K. Lu, "A practical mandatory access control model for XML databases", Information Sciences, vol. 179, 2008, pp. 1116-1133.

[5] Z. Rashid, A. Basit, and Z. Anwar. "TRDBAC: Temporal reflective database access control", in the 6th International Conference on Emerging Technologies (ICET), 2010, pp. 337-342.

[6] N. Qi, M. Kudo, J. Myllymaki and H.Pirahesh, "A function-based access control model for XML databases", in Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM: Bremen, Germany, 2005, pp. 115-122.

[7] L. Sun, H.Wang, R.Jururajin and S. Sriprakash, "A Purpose Based Access Control in XML Databases System", in the 4th International Conference on Network and System Security (NSS), 2010, pp. 486-493.

[8] M. Chagarlamudi, B. Panda and Y. Hu, "Insider Threat in Database Systems: Preventing Malicious Users' Activities in Databases", in 2009 Sixth International Conference on Information Technology: New Generations, ITNG '09, 2009, pp. 1616-1620.

[9] A. Lin, E. Vullings and J. Dalziel, "A Trust-based Access Control Model for Virtual Organizations", in Fifth International Conference on Grid and Cooperative Computing Workshops, GCCW '06, 2006, pp. 557-564.

[10] F. Almenarez, A. Marin, D. Diaz and J. Sanchez, "Developing a model for trust management in pervasive devices", in Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on, 2006, pp. 267-271.
[11] X. Ma, Z. Feng, C. Xu and J. Wang, "A Trust-Based Access Control with Feedback", in International Symposiums in Information Processing (ISIP), 2008, pp. 510-514.
[12] X. Han-fa, C. Bing-liang and X. Li-lin, "A mixed access control method based on trust and role", in the Second IITA International Conference on Geoscience and Remote Sensing (IITA-GRS), 2010, pp. 552-555.
[13] S. Singh, "Trust Based Authorization Framework for Grid Services", *Journal of Emerging Trends in Computing and Information Sciences*, vol. 2, 2011, pp. 136-144,.
[14] N. Farooqi and S. North, "Trust Based Access Control for XML Databases", The 6th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE Xplore, Abu Dhabi, UAE, 2011, pp. 764-765.
[15] N. Farooqi and S. North, "Developing a Dynamic Trust Based Access Control Model for XML Databases", Department of Computer Science Research Memoranda CS-11-09, University of Sheffield, UK, 2011.
[16] N. Farooqi and S. North, "Logging in XML Databases: Xlog File for Trust Based Access Control", in World Congress on Internet Security Conference (WorldCIS-2012), IEEE Xplore: Ontario, Canada, 2012, pp. 174-175.
[17] N. Farooqi and S. North, "A Performance Evaluation of Logging in XML Databases Using an Xlog File for Trust Based Access Control", *International Journal of Intelligent Computing Research (IJICR)*, Volume 3**,** 2012, pp.337-341.
[18] N. Farooqi and S. North, "Evaluation of Access Process in Trust Based Access Control for XML Databases", The 6th Saudi Scientific international conference (SIC), Brunel University, London, UK, 2012.
[19] N. Farooqi and S. North, "Evaluation of Practical Trust Based Access Control for XML Databases", The 7th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE Xplore, London, UK, 2012, pp. 336-340.
[20] XML Data Repository [Online]. University of Washington. Available: http://www.cs.washington.edu/research/xmldatasets/ [Accessed 30/6/2012].

6. Acknowledgements

This paper is supported by Umm Al Qura University on behalf of the Higher Ministry of Education of Saudi Arabia.

