The University of Sheffield
Department of Computer Science

Developing a Dynamic Trust-Based Access Control Model for XML Databases.

Norah Farooqi and Dr. Siobhán North
{acq10nf, S.North}@sheffield.ac.uk

Date
17-10-2011
Abstract
XML (Extensible Markup Language) databases are an active research area in databases since they have become widely used recently. The topic of Security in XML database is important to protect sensitive data and provide secure environments to users. This report gives a general background for XML and reviews the literature of access control for XML databases. The research proposal presents Trust-Based Access Control for XML Databases. Trust-based access is an established technique in many fields, such as networks and distributed systems, but has not previously been used for XML databases. In trust-based access control, user privileges are calculated dynamically depending on the user’s trust value. Applying the technique to XML databases should have many advantages over current techniques such as role based access.
Table of Contents
Chapter 1: Introduction ....................................................................................................................... 1
Chapter 2: XML Databases’ Background .......................................................................................... 2
  2.1. Introduction.......................................................................................................................... 2
  2.2. XML concepts..................................................................................................................... 2
  2.3. XML Syntax......................................................................................................................... 3
    2.3.1. Elements...................................................................................................................... 3
    2.3.2. Attributes.................................................................................................................... 3
    2.3.3. Comments................................................................................................................... 3
    2.3.4. Well-formed file............................................................................................................ 3
  2.4. Schemas............................................................................................................................... 4
  2.5. Tree structure....................................................................................................................... 4
  2.6. Xpath.................................................................................................................................... 5
  2.7. XML Databases.................................................................................................................... 7
  2.8. Conclusion ......................................................................................................................... 7
Chapter 3: Access control for XML Databases .................................................................................. 8
  3.1. Introduction........................................................................................................................ 8
  3.2. Access control concepts...................................................................................................... 8
  3.3. Access control types........................................................................................................... 9
    3.3.1. Discretionary access control model (DAC).............................................................. 9
    3.3.2. Mandatory access control model (MAC)............................................................... 10
    3.3.3. Role based access control (RBAC)........................................................................ 10
    3.3.4. Trust based access control...................................................................................... 11
  3.4. Access control techniques for XML databases ................................................................... 13
    3.4.1. Node filtering.......................................................................................................... 14
    3.4.2. Query rewriting....................................................................................................... 15
    3.4.3. Other techniques..................................................................................................... 16
  3.5. Conclusion ......................................................................................................................... 17
Chapter 4: The Research Proposal .................................................................................................... 18
  4.1. Introduction......................................................................................................................... 18
  4.2. Research Motivations ........................................................................................................ 18
    4.2.1. XML Databases’ Importance............................................................................... 18
    4.2.2. Security in XML Databases ............................................................................... 18
    4.2.3. Trust Based Access Control ............................................................................... 19
  4.3. The Research Hypothesis ................................................................................................. 19
  4.4. The Research Objectives and Contributions.................................................................... 19
4.4.1. Apply the trust-Based access approach to XML databases .............................................. 19
4.4.2. Extend dynamic access control to XML databases .............................................................. 19
4.4.3. Improve the access control security level for XML databases and user performance .................................................. 20
4.5. Research Proposal .................................................................................................................... 20
4.6. Conclusion ................................................................................................................................ 23
Chapter 5: Conclusion ....................................................................................................................... 24
References ............................................................................................................................................ 25
List of Figures

1: The bank XML database. ................................................................. 2
2: The tree of the bank XML database........................................... 5
3: Using the position privilege in user view (Gabillon, 2004)............... 13
4: The access control model was proposed by Damiani et al. (2008)... 15
5: The Architecture of Trust-Based Access Control Model for XML Databases... 19

List of Tables

Table 1. Basic marks are used in Xpath expression (W3Schools)... 5
Table 2. Xpath axes (W3Schools) ...................................................... 6
Chapter 1: Introduction

Recently, XML has been widely used in many applications such as the web. XML has the ability to store, exchange, transfer and retrieve the data. It is human and machine readable at the same time which improves its level of flexibility. Hundreds of researches are investigating XML databases. Most research focuses on either storage strategies or query performance. As storage and retrieval techniques are important, the security aspect is also vital. The security research in XML database is a key to protect data from unauthorised processes and misuse. Access control manages the access to data and controls users’ rights. A number of recent researchers in the topic of security have developed access control techniques for XML databases. This report studies the existing access control for XML database and aims to address their problems. Also, it attempts to put forward a research proposal to improve access control for XML databases.

The report covers a general basic background about XML in chapter 2. Chapter 3 provides a literature review of access control for XML databases. Chapter 4 explains the research proposal and outlines its architecture. The conclusion is about the main point of the research highlighted in chapter 5.
Chapter 2: XML Databases’ Background

2.1. Introduction

This chapter gives a general background about XML. It explains XML concepts, syntax and schema. It shows how an XML file can be structured as a tree and how to navigate it by using XPath.

2.2. XML concepts

The Extensible Mark-up Language (XML) has been widely used for the structured data representation (W3C; W3Schools; Abiteboul, Buneman et al. 2000; TotalXML 2011). It was derived from SGML in 1996 and recommended by W3C in 1998 (W3C; TotalXML 2011). XML differs from HTML since it focuses on storing and transferring data rather than controlling data appearance. There are many advantages of using XML since it is a self-describing language which gives the users the freedom to create their own tags. It is known for its flexibility due to its format. It is also readable by most platforms, so it can be shared between different applications. It is a simple text based language and portable data format (W3C; W3Schools; Abiteboul, Buneman et al. 2000; Harold and Means 2002; Ray 2003).

XML files have many components including: elements (e.g. <customer>), attributes (e.g. birthdate="16-6-1987") and comments (e.g. <!--Written by Norah 2/4/2011 -->) (W3Schools; Walsh 1998; Abiteboul, Buneman et al. 2000). Figure 1 shows the XML document for a bank database. The three main components of the XML file are explained in the next section.

```
<?xml version="1.0"?>
<!-- written by Norah on 2/4/2011 -->
<Bank>
  <customer birthdate="16-6-1987">
    <name>john smith</name>
    <mobile>07777777</mobile>
    <address>city name, street name, post code</address>
    <balance>£2000</balance>
    <card>
      <number>192837465</number>
      <start date>11-11-2010</start date>
      <end date>11-11-2012</end date>
      <security code>222</security code>
    </card>
    <transaction id="3333">
      <type>draw</type>
      <amount>£80</amount>
    </transaction>
    <transaction id="3334">
      <type>credit card payment</type>
      <amount>£40</amount>
    </transaction>
  </customer>
  <customer birthdate="7-8-1950">
    ...
  </customer>
  <customer birthdate="27-9-1970">
    ...
  </customer>
</Bank>
```

Figure 1. The bank XML database.
2.3. XML Syntax

2.3.1. Elements

The XML element is considered to be the basic component of the file. Normally, it includes the opening tag (<customer>) and the closing tag (<customer>), but it can be an empty tag (<customer/>). The value of the element can be a text, other elements or both. Elements can include attributes such as <customer birthdate="16-6-1987"> (Walsh 1998; Abiteboul, Buneman et al. 2000).

2.3.2. Attributes

The XML attributes provide information about the element which is not usually changed. They are placed inside the opening tag and consist of its name and its value. The attribute value is placed inside quotation marks. There are also reference attributes which are used as pointers for the element itself or to other elements (Walsh 1998; Abiteboul, Buneman et al. 2000).

Attributes should be distinguished from elements. Attributes are used for specific and static values. The value should be a string which is cited between quotation marks. On the other hand, it is better to represent dynamic and changeable data as elements. As mentioned before, the element value can be a string or other sub-elements or both. Attributes are more difficult to use and maintain than elements, so it is useful to limit using them (W3Schools; Abiteboul, Buneman et al. 2000; Ray 2003).

2.3.3. Comments

XML is a simple and clear language but as with any language, comments are used to clarify the complexity of the code or adding note for the writer or the external reader. The comment syntax is exactly the same as in HTML (e.g. <!--Written by Norah 2/4/2011--> ) (Tizag; W3Schools; Harold and Means 2002; Ray 2003).

2.3.4. Well-formed file.

Although XML is flexible and gives freedom to users to create their own tags as they want, there are some basic rules that should be followed.

- An XML file should have one root element.
- Starting and ending tags should be matched and they are case sensitive.
- Nested elements should be ordered in the right order; the last opening tag should be closed first and then the previous opening tag closed later. For example, the first element contains another element so the second element should be closed before the first element as shown in the following:
Chapter 2: XML Databases’ Background

- The value of attribute should be placed between quotation marks.
- An attribute name inside an element should be unique.

When the XML file conforms to all these rules, it is called well-formed XML file. (W3Schools; Abiteboul, Buneman et al. 2000; Harold and Means 2002; Ray 2003).

2.4. Schemas

There are several XML schemas which can be used with XML files such as DTD, XML Schema, RELAX NG and Schematron. They are used to store the file structure and show the elements’ relationships. The most popular ones are DTD and XML Schema (Abiteboul, Buneman et al. 2000; Lee and Chu 2000; Harold and Means 2002; Ray 2003).

DTD is the oldest type that is used to describe the XML file structure. It lists all the contents: elements and attributes. It can be defined inside the XML file as internal DTD or outside the XML file in a separate file as external DTD. The external DTD is used more than the internal because it can be related to many XML files and can define their syntax. XML Schema was proposed to overcome DTD limitations such as its namespace and lack of data types. XML Schema is more readable than DTD. Also, it can be defined as an internal XML Schema or an external file (Abiteboul, Buneman et al. 2000; Harold and Means 2002; Ray 2003).

2.5. Tree structure

An XML document is usually represented as a tree which starts by the root element and branches to many sub-trees. The tree has nodes that reflect XML file components such as elements and attributes (W3Schools; Abiteboul, Buneman et al. 2000; Harold and Means 2002; Ray 2003). The tree idea is derived from graph theory. An XML file can have only one tree representation (Ray 2003).

This tree shows the XML document components and reflects its structure in a graphical form. There are many relationships between nodes defined using the tree. The root element is called the parent and it has many children in the lower level. The children at the same level are called siblings (W3Schools; Abiteboul, Buneman et al. 2000; Harold and Means 2002; Ray 2003). All these relations are described in the Xpath section in detail. Figure 2 shows XML tree for the XML document that is mentioned in Figure 1.
2.6. Xpath

XML path language (Xpath) defines the XML file as a tree. It uses path expressions to navigate the tree and address specific parts of XML tree, either a node or a set of nodes. The path expression is similar to the file systems. It was recommended by W3C in November 1999. It is used in the Query language to retrieve data from the XML tree such as Xpath 2.0 and XQuery. It defines nodes’ relations: parent, child, sibling, ancestor and descendant. Parent and child relation is that each child has only one parent and the parent can have no children or a number of children. Children in the same level under the same parent are called siblings. The ancestor relation goes up from the parent until it reaches the root. In contrast, descendant goes down from the child until it reaches the leaves. (Tizag.; W3C; W3Schools; Harold and Means 2002; Ray 2003).

Xpath expressions show the location of elements and attributes in XML files. The syntax of Xpath includes many special marks to identify the node (Tizag.; W3C; W3Schools; Harold and Means 2002; Ray 2003); the basic marks are explained in table 1 (W3Schools).

Table 1. Basic marks are used in Xpath expression (W3Schools).

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nodename</code></td>
<td>Selects all child nodes of the named node</td>
</tr>
<tr>
<td><code>/</code></td>
<td>Selects from the root node</td>
</tr>
<tr>
<td><code>//</code></td>
<td>Selects nodes in the document from the current node that match the selection no matter where they are</td>
</tr>
<tr>
<td><code>.</code></td>
<td>Selects the current node</td>
</tr>
<tr>
<td><code>..</code></td>
<td>Selects the parent of the current node</td>
</tr>
<tr>
<td><code>@</code></td>
<td>Selects attributes</td>
</tr>
</tbody>
</table>
Here are some examples of Xpath expressions for the bank database which was mentioned before in figure 1:

This expression can be used to access the mobile node which is child of customer node and descendant of bank node.

```
/bank/customer/mobile
```

This expression is used to select identifier (id) attribute of a transaction node.

```
/bank/transaction@id
```

Xpath expressions can also have predicates which are used to find specific nodes and values. It uses square brackets marks “[ ]” to write predicates (Tizag.; W3C; W3Schools; Harold and Means 2002; Ray 2003). For example, to find all transactions that have an amount over £50, the predicate must be used in the Xpath expression as follows:

```
/bank/customer/transaction [amount>50]
```

Also, Xpath expressions have many other marks that are used to find unknown nodes and many operators that are used to select paths. It uses axes to identify group of nodes that are related to the specific node (Tizag.; W3C; W3Schools; Harold and Means 2002; Ray 2003). Table 2 shows Xpath axes names and results (W3Schools).

Table 2. Xpath axes (W3Schools).

<table>
<thead>
<tr>
<th>AxisName</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>ancestor</td>
<td>Selects all ancestors (parent, grandparent, etc.) of the current node</td>
</tr>
<tr>
<td>ancestor-or-self</td>
<td>Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself</td>
</tr>
<tr>
<td>attribute</td>
<td>Selects all attributes of the current node</td>
</tr>
<tr>
<td>child</td>
<td>Selects all children of the current node</td>
</tr>
<tr>
<td>descendant</td>
<td>Selects all descendants (children, grandchildren, etc.) of the current node</td>
</tr>
<tr>
<td>descendant-or-self</td>
<td>Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself</td>
</tr>
<tr>
<td>following</td>
<td>Selects everything in the document after the closing tag of the current node</td>
</tr>
<tr>
<td>following-sibling</td>
<td>Selects all siblings after the current node</td>
</tr>
<tr>
<td>namespace</td>
<td>Selects all namespace nodes of the current node</td>
</tr>
<tr>
<td>parent</td>
<td>Selects the parent of the current node</td>
</tr>
<tr>
<td>preceding</td>
<td>Selects everything in the document that is before the start tag of the current node</td>
</tr>
<tr>
<td>preceding-sibling</td>
<td>Selects all siblings before the current node</td>
</tr>
<tr>
<td>self</td>
<td>Selects the current node</td>
</tr>
</tbody>
</table>

Xpath expressions can be classified into two types: absolute and relative paths. The absolute path for a specific node is the full path starting from root until reaching the node. The relative path is shorter and is based on a specific node only (Tizag.; W3C; W3Schools; Harold and Means 2002; Ray 2003).
2.7. **XML Databases**

XML is considered as a type of database since it has the ability to store and retrieve data like other types of databases. The XML database is categorised into either an enabled XML database or a native XML database. The enabled XML database stores data in traditional databases. On the other hand, an XML file is considered as the basic unit in native XML databases which deal with the files directly. In native XML databases, the file is represented as a tree and it can be stored as objects (Staken 2001; Bourret 2005; Harold 2005). This research will focus only on native XML databases.

2.8. **Conclusion**

XML is a vast topic and cannot be covered from all aspects in this limited report. The basic points are listed in this chapter to give a suitable background to understand the research proposal. The next chapter describes access control concepts and explains some techniques in access control for XML database.
Chapter 3: Access control for XML Databases

3.1. Introduction

XML databases are widely used in many different areas. Databases are used to store, retrieve and provide data and information in an organised manner. They are multi-users systems, meaning they can be accessed by millions of users and can provide a huge amount of data. This large amount of data needs to be controlled, managed and organised. In addition, much of this data is sensitive and personal. All data and especially confidential data need to be protected and saved in a secure environment. Therefore, XML databases should manage data securely to protect user rights and data privacy (Izadi, Asadi et al. 2007; Li and Hong 2008; Gollmann 2011).

The main aim of XML security technologies is to protect information and ensure users have proper authorisation. These technologies include XML signature, XML encryption and XML access control (Jo, Kim et al. 2005; Ardagna, Damiani et al. 2007; Myint 2010; Gollmann 2011). This report focuses on access control technology and investigates current research and proposals in this area to improve access control in XML databases. The following sections introduce access control concepts and compare their different types. Access control techniques that are applied to XML databases are discussed in detail.

3.2. Access control concepts

Access control is an important topic in security and it is applied to many computer fields such as operating systems, networks and databases. This research focuses on access control for XML databases. In general, access control models control and manage users’ access to XML nodes or attributes according to policy rules. They prevent unsecure access that breaches data security. The following sections cover policy rules concepts and the access control structure.

Access control has security policy rules that define the access instructions for users and specify who can access what under which privileges. They are also called authorisation specifications (Gabillon 2004; Qi, Kudo et al. 2005; Di Vimercati, S.Foresti et al. 2008). The term ‘privileges’ which is frequently used in this topic means the right that is given to the subject to perform actions and operations on objects (Damiani, di Vimercati et al. 2005; Gollmann 2011). Usually the main syntax of policy rules is represented as <subject, object, action>. Subjects are entities that can access and achieve resources and data through requests. Each subject may be a user’s identification, his location, IP address or a combination of them; it can also be a group of users. Objects can be defined as items that are accessed through requests. Each Object refers to an XML document or single node and attribute. Action refers to privileges such as read (browse) and write (insert, update, delete); it is also called access mode (Chan, Qing et al. 2004; Gabillon 2004; Di Vimercati, Marrara et al. 2005; Gabillon 2005; Jo, Kim et al. 2005; Di Vimercati, S.Foresti et al. 2008; Li and Hong 2008; Gollmann 2011)
Access authorisation is sometimes also defined by adding sign and type factors to the main syntax and appear as <subject, object, action, sign, type> where sign reflects that the access permission is either permitted or denied. The sign is represented by positive + or negative - marks. The type refers to the authorisation type, which is local, recursive, hard, soft or a mixture of them. Many other factors such as time have been added to this syntax by various authors. These policies can relate to reading, writing and positioning privileges (Park, Costello et al. 2004; Gabillon 2005; Jo, Kim et al. 2005; Qi, Kudo et al. 2005; An and Park 2007; Di Vimercati, S.Foresti et al. 2008).

There are two main traditional access control structures that store and manage access rights: an access control matrix and access control list. An access control matrix contains a set of subjects, actions and objects. Each subject is represented as a row and each object is represented as a column in the matrix. Cells in the matrix may contain actions’ values given to the subject in the same row to execute this action on the object in the same column or they may be empty. These empty cells consume space and this matrix does not work efficiently with a large amount of data because the update process becomes difficult. Access control list (ACL) solves part of the problem by storing object rights rather than both the subject and object. Each object has a list that contains all subjects that can access it and stores the actions’ values (Sandhu and Samarati 1994; Qi, Kudo et al. 2005; Gollmann 2011) . In the following section, the main access control categories are discussed.

3.3. Access control types

Access control models can be categorised into two main types: the discretionary access control model (DAC) and the mandatory access control model (MAC) (Zhu, Jin et al. 2007). However, some authors categorise it into three core categories: discretionary access control model (DAC), mandatory access control model (MAC) and role based access control (RBAC) (Sandhu and Samarati 1994; Jeong, Kim et al. 2003; Wang and Osborn 2004; Rashid, Basit et al. 2010). In addition to these traditional categories, there are some new types: attribute based access control and trust based access control. Likewise, there are many proposed types for access control such as functional based access control, non-discretionary access control (NAC), and purpose based access control (Chan, Qing et al. 2004; Wang and Osborn 2004; Byun, Bertino et al. 2005; Qi, Kudo et al. 2005; Sun, Wang et al. 2010) . The basic types are explained in the following sections.

3.3.1. Discretionary access control model (DAC)

This category of access control depends on the subject identity to manage the access process to databases. This type is characterised by its flexibility, so the subject can grant access control rights to the other subject. Depending on this flexibility, discretionary access control is implemented in several applications and is adopted by the majority of commercial database management systems (DBMSs). However, the passing of access rights to other users may cause illegal access to sensitive information or permit malicious attacks on databases. This discretionary access control loses control to disseminate data and to check prohibited flows of
unauthorised subjects (Sandhu and Samarati 1994; Jeong, Kim et al. 2003; Chan, Qing et al. 2004; Bertino and Sandhu 2005; Zhu, Jin et al. 2007; Rashid, Basit et al. 2010).

3.3.2. Mandatory access control model (MAC)

In MAC, both subjects and objects are classified into multi-levels depending on the object’s content. Security levels of objects reflect their sensitivities and all security levels have labels. These security levels can be in ordered or unordered labels. The order levels are usually classified into TopSecret, Secret, Confidential and Unclassified. The relation between these levels is defined as Topsecret > Secret > Confidential > Unclassified. Some models define other order label sets like Protected > Private > Public. On the other hand, unordered labels can be defined as names that are used in the same area such as Technique, Human resource, Financial, etc. (Sandhu and Samarati 1994; Jeong, Kim et al. 2003; Chan, Qing et al. 2004; Bertino and Sandhu 2005; Zhu, Jin et al. 2007; Duong and Zhang 2008; Rashid, Basit et al. 2010).

Access operations in mandatory access control depend on two principles: read-down and write-up. Read-down concept means that users can read only objects at their level or lower. Write-up concept means that users can write only at that level or higher. However, the write-up principle causes problems for data integrity because the subject in the low level can write data in the high level without having privileges to read data in that level, so it may overwrite data. Some systems overcome this issue by eliminating the write-up concept and making the write process for the same level only. Moreover, defining and classifying security levels in organisations is a difficult process. Therefore, this type is not adopted widely by database management systems (DBMSs) (Sandhu and Samarati 1994; Jeong, Kim et al. 2003; Chan, Qing et al. 2004; Bertino and Sandhu 2005; Zhu, Jin et al. 2007; Rashid, Basit et al. 2010).

3.3.3. Role based access control (RBAC)

Role based access control divides subjects (users) according to their roles and responsibilities in the system. Privileges and rights to access objects are assigned to roles and then subjects are assigned to roles depending on their jobs. The authorisation process management becomes easier and more effective by being broken down into two steps rather than assigning privileges to subjects directly. If the user’s role changes, it will be easy to revoke the old role and assign him a new role without consuming time in changing privileges. In addition, the relation between role and subject is a many to many relationship that means the role can be assigned to a subject or group of subjects and the subject can have one role or a group of roles. Usually role based access control has a role hierarchy that relates to all the roles in the systems.

The administrator defines the operation and role for each user and he can control the authorisation roles by adding constraints on execution processes. This category is the most popular one in large-scale systems and it has a great influence in the access control area. However, it may face privilege abuse from subjects who misuse their roles (Sandhu and Samarati 1994; Hitchens and Varadharajan 2001; Jeong, Kim et al. 2003; Wang and Osborn
2004; Bertino and Sandhu 2005; Byun, Bertino et al. 2005; Park and Giordano 2006; Rashid, Basit et al. 2010).

3.3.4. Trust based access control

Lately, trust based access control has become established and is used in many areas, such as networks and virtual organisations. Many researches have focused on developing and improving it. Trust based access control is discussed in detail because it is central to the research proposal discussed in chapter 3. The term ‘trust’ is defined from different aspects such as sociology, psychology, mathematics, and economics. Trust has been used in different disciplines under different perspectives to match implementation needs. Trust depends on beliefs, operations, and recommendations. It requires effort and time to achieve but it can be lost quickly and easily. Trust is taken from the real world and applied to the digital world. Trust concepts used in the access control topic mean that subjects can trust entities such as other subjects, applications, and firms according to past histories, operations, behaviours experiences, and recommendations over time (Abdul-Rahman and Hailes 1997; Cahill, Gray et al. 2003; Almenarez, Marin et al. 2004; Almenarez, Marin et al. 2005; Almenarez, Marin et al. 2006; Lin, Vullings et al. 2006; Feng, Lin et al. 2008; Jia, Collins et al. 2009; Ma, Logrippo et al. 2010; Zhao, Liu et al. 2010; Singh 2011).

There are many advantages of using trust. The main feature is that it is dynamic; trust values are changeable and can be increased or decreased according to the subjects’ behaviour, history, and operations. It is subjective, which means each subject can be trusted by many subjects with a different trust value. Each subject trusts itself by default. Moreover, trust relations are asymmetrical which means this relationship can be different in both directions. For example, subject A can trust subject B but at the same time B does not trust subject A. It is transitive under conditions, which means it can be transferred from one entity to others. For example, subject A trusts subject B and at the same time subject B trusts subject C. If subject A trusts subject B as leader, then subject A can trust subject C indirectly. In addition, it is dependent on both past and present time and context (Abdul-Rahman and Hailes 1997; Cahill, Gray et al. 2003; Almenarez, Marin et al. 2004; Ryutov, Zhou et al. 2005; Lin, Vullings et al. 2006; Xing, Cui et al. 2010; Zhao, Liu et al. 2010; Singh 2011).

Trust values can reflect the relations between two sides. Trust value can be measured in several ways. It can be in a range between 0 to 1 where 0 means distrust and 1 is trusted (Almenarez, Marin et al. 2004; Almenarez, Marin et al. 2005; Tran, Hitchens et al. 2005; Almenarez, Marin et al. 2006; Feng, Lin et al. 2008; Jia, Collins et al. 2009; Zhang and Rao 2010; Singh 2011). It also can be defined using real numbers such as between 1 to 10 where 1 means low trust and 10 is high trust (Abdul-Rahman and Hailes 1997; Zhao, Liu et al. 2010). It can be described by using levels such as L1, L2... L5 where L1 means distrust and L5 means completely trusted (Lin, Vullings et al. 2006; Singh 2011). Trust value can be called trust degree when using a specific range value, and trust level when using level concepts (Abdul-Rahman and Hailes 1997; Almenarez, Marin et al. 2004; Almenarez, Marin et al. 2005; Tran, Hitchens et al. 2005; Almenarez, Marin et al. 2006; Lin, Vullings et al. 2006; Feng, Lin et al. 2008; Jia, Collins et al. 2009; Ma, Logrippo et al. 2010; Zhang and Rao 2010;
Zhao, Liu et al. 2010; Singh 2011). Trust based access control depends on a trust management system, which automatically calculates and updates the trust values of users. Trust values rely on users’ behaviours, users’ histories, users’ credits, and users’ operations. Users can access resources through trust values and levels (Ryutov, Zhou et al. 2005; Almenarez, Marin et al. 2006; Lin, Vullings et al. 2006; Ma, Feng et al. 2008; Xing, Cui et al. 2010).

Trust relationship between two entities can be categorised to be either a direct relation or indirect relation. Direct trust relations depend on the past interactions, experiences, and operations between two subjects without using any other external resources. Since not all subjects know others, each direct relation can be assigned an initial value and then change over time according to actions. The direct trust relation is reliable when evaluating the trust value. On the other hand, there is the indirect relation that depends on recommendations from a third party. For example, subject A is a recommender who recommends subject B to interact with subject C because there was a good experience between subject A and subject C in the past (Abdul-Rahman and Hailes 1997; Almenarez, Marin et al. 2004; Almenarez, Marin et al. 2005; Tran, Hitchens et al. 2005; Ma, Feng et al. 2008; Singh 2011).

Some modules for trust based access control depend on both direct and indirect relations to calculate the trust value. Other systems add some other factors such as reputation, contribution, and domain trust (Abdul-Rahman and Hailes 1997; Tran, Hitchens et al. 2005; Ma, Feng et al. 2008; Singh 2011). Trust based access control models use different techniques to calculate trust values depending on their system design, needs, and goals.

Many equations have been proposed and used to calculate the trust value between entities. The general equation is quoted here to clarify how the trust value is calculated between two entities. The standard formula is:

\[
T_{p_1 \rightarrow p_2} = \frac{\sum_{i=1}^{n} [o_i(B_i) - e_i(B_i)]}{t}
\]  

(1)

This equation (1) calculates the trust value between P1 and P2 from P1’s prospective. Where \( t \) is the time for previous operations between them; \( i \) reflects the number of interactions; \( B_i \) is behaviours in \( i \) interactions; \( o_i \) is the observed value for these behaviours; \( e_i \) is expected value for these behaviours (Lin, Vullings et al. 2006).

In addition, there are many techniques to recalculate the trust value depending on the past and present operations. One of these equations is:

\[
T_i = \begin{cases} 
T_{i-1} + \omega \cdot V_{a_{i}} (1 - T_{i-1}) & \text{if } a_{i} > 0 \\
T_{i-1} (1 - \omega) + \omega \cdot V_{a_{i}} & \text{else} 
\end{cases}
\]

(2)
In formula (2), $T_i$ means the recalculated trust value and $T_{i-1}$ means the previous value. Where $\omega$ is the strictness factor that connects with subjects’ disposition in the past and the present and $\omega$ is the given difference percentage. $V_{ai}$ is the weight variable that depends on past operations and is calculated according to action weight (Almenarez, Marin et al. 2006; Zhang and Rao 2010).

Some trust based access control modules use flowcharts to explain how their systems work in each step, from requesting until accessing resources (Tran, Hitchens et al. 2005; Feng, Lin et al. 2008; Ma, Feng et al. 2008). Some models implement their work by using SUN’s XACML API that depends on XACML. XACML is standard for eXtensible Access Control Markup Language that writes and describes the access policies in XML language (Almenarez, Marin et al. 2005; Zhang and Rao 2010; Singh 2011).

### 3.4. Access control techniques for XML databases

The only widely used method to apply any of the access control for XML database types which were mentioned in the previous section (3.3) uses Xpath (see section 2.6). The majority of traditional and proposed access control types for XML database such as role based, mandatory based, purpose based and function based focus on processing of access to XML files and elements by using Xpath (Hitchens and Varadharajan 2001; Wang and Osborn 2004; Qi, Kudo et al. 2005; Zhu, Jin et al. 2007; Li and Hong 2008; Landberg, Rahayu et al. 2010; Sun, Wang et al. 2010). Xpath is used as a basic technique to access a node or set of nodes in XML databases. Besides using Xpath to access specific nodes, there are many techniques that are used to describe how the access process is done and explain user’s access permission and privileges.

Many different models for XML access control that manage users’ access to data have been proposed and developed by using the main techniques: node filtering (Yu, Srivastava et al. 2002; Gabillon 2004; Yu, Srivastava et al. 2004; Gabillon 2005) and query rewriting (Mohan, Sengupta et al. 2005; Mohan, Klinginsmith et al. 2006; Mohan, Sengupta et al. 2006; Damiani, Fansi et al. 2007; Damiani, Fansi et al. 2008; Byun and Park 2010). Likewise, many other techniques have been used to develop access control systems such as labelling and path indexing (An and Park 2007; Duong and Zhang 2008; Duong 2010; An and Park 2011). It seems these techniques consider access control from a processing query perspective and do not consider applying different types of access control to XML databases. To understand the access control topic for XML, these techniques are discussed in the following sections.
3.4.1. Node filtering.

In simple terms, Node filtering means scanning and passing the whole tree and giving each node a positive or negative sign. The node filtering technique depends on access policies to create a user view which then can be integrated with user queries (Gabillon 2004; Gabillon 2005; Damiani, Fansi et al. 2008; Byun and Park 2010). This technique consumes a large amount of storage space and needs high maintenance since the filtering process is repeated many times for each user or group of users (Damiani, Fansi et al. 2008; Byun and Park 2010).

Many access control systems for XML depend on this technique to develop their access control system (Damiani, De Capitani di Vimercati et al. 2000; Damiani, Di Vimercati et al. 2000; Damiani, Samarati et al. 2001; Damiani, Vimercati et al. 2002; Yu, Srivastava et al. 2002; Gabillon 2004; Yu, Srivastava et al. 2004; Gabillon 2005; Di Vimercati, S.Foresti et al. 2008). Some of these models support just the read privilege (Damiani, De Capitani di Vimercati et al. 2000; Damiani, Di Vimercati et al. 2000; Damiani, Samarati et al. 2001; Yu, Srivastava et al. 2002; Yu, Srivastava et al. 2004) and others support both read and write privileges (Damiani, Vimercati et al. 2002; Gabillon 2004; Gabillon 2005; Di Vimercati, S.Foresti et al. 2008). A novel privilege called position privilege has been introduced to solve some issues that were addressed in node filtering techniques by other alternative models. The position privilege makes a separation between the node’s existence and its content. In other words, it allows the user to know about the node’s existence without knowing the node’s label and value. The node has a RESTRICTED label and figure 3 explains the position privilege work (Gabillon 2004; Gabillon 2005).

![Figure 3. Using the position privilege in user view (Gabillon, 2004)](image-url)
A compressed accessibility map (CAM) approach for XML database access control is proposed to solve the storage problem and speed up the process in this technique (Yu, Srivastava et al. 2002; Yu, Srivastava et al. 2004). This map gathers nodes that have similar accessible attributes together. The compressed tree looks smaller than the original one and has fewer nodes. The model can provide a fast response to determine the user’s rights. The main function of this map is to determine whether access is allowed or denied for each user. It can be defined as \( M = H \times U \times A \), where \( M \) refers to the map, \( H \) refers to the database tree, \( U \) refers to the users and \( A \) refers to the access modes. This approach contains algorithms to find the best compressed accessibility map to reduce the storage space consumed. The compressed accessibility map is created for each user and each access mode (Yu, Srivastava et al. 2002; Yu, Srivastava et al. 2004). Although this approach tried to reduce the storage space by reducing the tree size, it still consumes a large amount of storage due to the presence of many users’ maps (Duong and Zhang 2008; Duong 2010).

There are various additions to the node filtering technique. One of these additions is that the authorisation model can deal with authorisations in one XML document or a group of XML documents by working with DTD and XML schema (Damiani, De Capitani di Vimercati et al. 2000; Damiani, Di Vimercati et al. 2000; Damiani, Samarati et al. 2001; Di Vimercati, Marrara et al. 2005). The access control system can add the access rules to the XML file or XML schema in separate files (Di Vimercati, Marrara et al. 2005).

Node filtering is one of the two main approaches which describes how the access process is done in XML databases. The other approach is known as ‘query rewriting’.

### 3.4.2. Query rewriting.

The query rewriting technique depends on transforming possibly unsafe queries into safe ones which can then access the data. This technique uses much more time for rewriting process. Several models have been proposed depending on this technique.

Some systems that use the query rewriting technique create annotated schemas by using many attributes (Mohan, Sengupta et al. 2005; Mohan, Klinginsmith et al. 2006; Mohan, Sengupta et al. 2006; Damiani, Fanti et al. 2007; Mohan, Sengupta et al. 2007; Damiani, Fanti et al. 2008). The access attribute provides the ‘allow’ or ‘deny’ right to the subject to access the object. The condition attribute includes a number of predicates. The third attribute, ‘dirty’, indicates that access to some node’s descendants may be denied. A user schema view can easily be created from the annotated schema. In addition, finite state automation is used in rewrite automation process. Rewriting the query can be done by using automation and except operator (Damiani, Fanti et al. 2007; Damiani, Fanti et al. 2008). Sometimes, primitives are used to reflect the security specification by using an algebraic language (Mohan, Sengupta et al. 2005; Mohan, Klinginsmith et al. 2006; Mohan, Sengupta et al. 2006; Mohan, Sengupta et al. 2007). These procedures are applied for read and write (insert, delete, update) privileges. Figure 4 shows all the steps in the query rewriting process.
The access control can take into account both the nodes’ values and nodes’ relationships. A virtual security view can be used with query rewriting approaches (Mohan, Sengupta et al. 2005; Mohan, Klinginsmith et al. 2006; Mohan, Sengupta et al. 2006; Mohan, Sengupta et al. 2007).

3.4.3. Other techniques

Some access control models do not depend on either node filtering or query rewriting techniques. These models use several methods which avoid repeated processes for users. They aim to define a standard access control system which is suitable for all users rather than find a particular access mode for each user and repeat the process several times (Kitagawa and Yoshikawa 2005; Duong and Zhang 2008; Duong 2010).

Access control system can depend on policy tables’ technique. Due to the policy tables’ size and their numbers, a simplification and unification process is performed and improve the access decisions by reducing the time consumed. The policy table consists of two columns: pathID and flag; each one of these tables represents only one role. The pathID for each node is taken from the path information table which gives each path a number (ID). The flag in the policy table reflects whether access is allowed (+) or denied (−).
The simplification process is used to minimise the table size by representing only the minimum pathID when sequential numbers have the similar access result. Then the unification stage occurs by creating the role table which consists of role name and roleID. Finally, the unifying policy table (UPT) is generated from the role table and policy tables. The UPT table includes pathID and flag key number. PathID refers to the node in the XML tree and the flag key number is a product of the roleID which only allows access to this node. It seems that the system depends on many tables which means using a relational database to implement access control for the XML database. It may face some problems when implementing the write privilege. For example, the delete processes need to change data in many tables which may consume much more time (Kitagawa and Yoshikawa 2005).

Also, access control model can define access control concept in files. The XML Access Authorization file (XAA) included all XML elements and their access levels. The access authorization of user groups is defined in the XML Group Authorization (XGA). In this technique, the access level was classified as: public < private < protected. Many different symbols were used to refer to access level such as “#” to refer to the protected level. When this method is compared with the node filtering technique according to the number of scanned nodes and response time, it demonstrates a good performance due to its speed and accuracy (Duong and Zhang 2008; Duong 2010).

3.5. Conclusion

This chapter reviews the related background to the research proposal. It describes the access control types and highlights both their advantages and disadvantages. It discussed trust based access control model in detail since it is the main point of the proposal. Several techniques are described for accessing XML databases. It seems all traditional and proposed access control types for XML database used XPath. The next chapter identifies the research proposal and defines the research hypotheses and objectives.
Chapter 4: The Research Proposal

4.1. Introduction
Both chapter 2 and 3 offer a background to XML databases and access control concepts. After reviewing the past and current research, the motivations and objectives of this research are pointed out in this chapter. The research hypothesis is defined. This chapter also describes, in detail, the system architecture and components.

4.2. Research Motivations
There are various motivations behind this research proposal and each one is discussed below:

4.2.1. XML Databases’ Importance
In the last decade XML has become established and used in a wide range of areas such as the web, businesses, information systems, and databases (W3C; W3Schools; Abiteboul, Buneman et al. 2000; Champion 2001; Vakali, Catania et al. 2005). Much research has been carried out to investigate and develop this technology. XML is used to store, transfer, and manipulate data. As mentioned in chapter 1, XML has many advantages; it is readable for both humans and machines. It is flexible, simple, and self-descriptive. Users create their tags according to their needs (W3C; W3Schools; Abiteboul, Buneman et al. 2000; Champion 2001; Vakali, Catania et al. 2005). Therefore, one of the motivations is that XML databases are active and update areas in the databases. As a result of this flexibility, it has the ability to improve and develop.

4.2.2. Security in XML Databases
Having data is a power but it must be dealt with in an appropriate and accurate manner. Much of the research on XML focuses on storage strategies and query performance. Although data storage and retrieval techniques are important, so is security and, in comparison, this is a neglected area. XML databases are multi-user systems, meaning they can be accessed by millions of users and can provide a huge amount of data. Much of this data is sensitive and personal. Confidential data need to be protected and saved in a secure environment. Security research for XML databases is crucial in protecting data from unauthorised processes and misuse. Hence, access control is vital in security. There has been much research carried out to investigate and develop this topic but still there are quite a number of points that need to be investigated.
4.2.3. Trust Based Access Control
As described in chapter 2, many different models for XML database access control have been proposed and developed. Access control systems for XML databases can be categorised into three core categories: discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC) (Hitchens and Varadharajan 2001; Wang and Osborn 2004; Zhu, Jin et al. 2007). Most traditional access control models protect data from the malicious activities of outside users but cannot protect the data from insiders (Chagarlamudi, Panda et al. 2009). Research has highlighted that damage caused by insiders is more harmful than that of outsiders (Park and Giordano 2006). The insider threat is a huge topic in data security and many methods have been proposed to identify misuse behaviour, yet there has been no work on dynamic updates to access privileges in relation to trust for XML databases (Farooqi and North 2011).

Lately, trust-based access control has been established and used in many areas, such as networks and virtual organisations. It depends on a trust management system, which automatically calculates and updates the trust values of users. Trust values rely on users’ behaviours, users’ histories, users’ credits, and users’ operations. Users can access resources through trust values and levels (Almenarez, Marin et al. 2006; Lin, Vullings et al. 2006; Ma, Feng et al. 2008; Xing, Cui et al. 2010; Farooqi and North 2011; Singh 2011).

4.3. The Research Hypothesis
Since the trust-based access control approach applies to many areas of research, such as networks, distributed systems, pervasive devices, and virtual organisations (Abdul-Rahman and Hailes 1997; Cahill, Gray et al. 2003; Almenarez, Marin et al. 2004; Almenarez, Marin et al. 2005; Almenarez, Marin et al. 2006; Lin, Vullings et al. 2006; Feng, Lin et al. 2008; Jia, Collins et al. 2009; Ma, Logrippo et al. 2010; Zhao, Liu et al. 2010; Singh 2011), it may also apply to the XML database research field. Depending on the trust management features, it may improve access control to an XML database by providing an appropriate security level to protect sensitive and confidential data. It may possibly make the access process more reliable through evaluating users’ histories and executing dynamic update permissions for the access control system.

4.4. The Research Objectives and Contributions
Considering the problems of the traditional access control model, the research approach aims to:

4.4.1. Apply the trust-Based access approach to XML databases
Since the trust-based approach is considered to be one of the new types of access control systems (Lin, Vullings et al. 2006; Ma, Feng et al. 2008), its benefits can be applied to the research area of XML databases (Farooqi and North 2011).

4.4.2. Extend dynamic access control to XML databases
One of the most important features of trust-based access control is that it is dynamic. Compared with the traditional approaches to access control, which are static, trust-based access control can make access systems responsive and active (W3C ; W3Schools ; Abiteboul, Buneman et al. 2000; Champion 2001; Vakali, Catania et al. 2005; Farooqi and North 2011).
4.4.3. Improve the access control security level for XML databases and user performance

This model improves data security by evaluating users’ histories and operations. This approach extends the established work by considering error factors in calculating trust values. Therefore, users’ permissions and privileges can change in response to their behaviour (Farooqi and North 2011).

4.5. Research Proposal

The research proposes applying trust-based access control to XML databases, as mentioned in the objectives section. The architecture of the proposed trust-based access control for XML databases is shown in Figure 5. It is based on direct trust and ignores indirect trust. Direct trust focuses on users’ operations and errors. Indirect trust depends on recommendations and therefore is largely irrelevant in this context (Farooqi and North 2011).
The model consists of two main parts: the trust model and the access control model. The trust model evaluates users’ activities and calculates the trust value for them. The evaluation process depends on the recordings of some operations and all errors in the Xlog file. The Xlog file is temporary for each session or defined time. The file is made temporary to save time and storage space in the recording and searching processes. The error detector then finds errors in the Xlog file and assigns a weight to them according to the error policy file. The error policy file has its rules to calculate the error weight (Farooqi and North 2011).

The majority of trust-based access control models define their policy to calculate trust value according to their system’s needs. They specify which factors are considered in their calculations and then define the rules and equations (see chapter 3, section 3.3.4). This research proposes two ways to identify the error weight, either by calculating the error percentage by comparing the number of errors with the whole operations or by defining a specific number for the error for each weight.

The first proposed method:

\[
\text{Errors per cent (EP)} = \frac{\text{Number of errors}}{\text{number of operations}}.
\]

- If \( EP = 0 \) then Error weight (EW) = 0
- If \( 0 < EP < 0.25 \) then Error weight (EW) = 0.25
- If \( 0.25 < EP < 0.5 \) then Error weight (EW) = 0.5
- If \( 0.5 < EP < 0.75 \) then Error weight (EW) = 0.75
- If \( 0.75 < EP \leq 1 \) then Error weight (EW) = 1

The second proposed method:

Identify errors number in sessions.

- If \( EV_1 < EP > EV_2 \) then Error weight (EW) = 0.25
- If \( EV_2 < EP > EV_3 \) then Error weight (EW) = 0.5
- If \( EV_3 < EP > EV_4 \) then Error weight (EW) = 0.75
- If \( EV_4 < EP > EV_5 \) then Error weight (EW) = 1

These specific operations will be defined in the operation policy file. Likewise, the operation evaluator assigns the appropriate weight to misused operations defined in the operation policy file and recorded in the Xlog file. Defining specific operations is difficult because the rules are different from system to system and from one organisation to another. Therefore, operations are defined in general and can be changed or extended according to system and organisation needs. The rules to capture misused operations can be defined according to organisation policy. At this initial stage of the research, one of the proposed rules to capture a misused operation is measuring when users are accessing others’ permissions illegally. The rules to evaluate ‘misuse’ may change and develop according to the research processes (Farooqi and North 2011). It is described as follows:

Where Bad Operations’ Number is (BOP) and Bad Operations’ Weight is (BOW); BV values from BV1 to BV5 are specified and defined according to system strategy and organisation policy.
Chapter 4: The research Proposal

If $BV_1 < BOP$, then $BOW = 0.25$.

If $BV_2 < BOP$, then $BOW = 0.5$.

If $BV_3 < BOP$, then $BOW = 0.75$.

If $BV_4 < BOP$, then $BOW = 1$.

After recording errors and operations and calculating their weight, the trust calculator determines the trust value depending on the suggested equation in this proposal:

$$TV = ETV \times ETVW + AV \times AVW + GOV \times GOVW.$$ (1)

In (1):

- $TV =$ Trust Value.
- $ETV =$ Existing Trust Value.
- $ETVW =$ Existing Trust Value Weight.
- $AV =$ Accuracy Value = $1 - $ Error Weight (EW).
- $AVW =$ Accuracy Value Weight.
- $GOV =$ Good Operations Value = $1 - $ Bad Operations Value (BOV).
- $GOVW =$ Good Operations Value Weight.

All ETVW, AVW, and GOVW percentages can change according to the organisation’s policies. For example, AVW can be 10% if the organisation does not consider the error factor as an important element or AVW can be 50% if the organisation considers the error rate to be very important. The trust value is updated automatically in the users’ access permission policy file. The XML database’s access permission policy file describes the trust values required to access XML nodes. These nodes can be accessed individually or in groups by views. To decide which technique is better, some experiments should be performed. The access manager sub-model matches two files to manage a user’s right to access the requested data. The access decision maker allows or denies user’s access to XML databases according to the results. All these proposed rules, calculations and the equations may change and develop depending what occurs in the future processes (Farooqi and North 2011).

Here is an example to clarify how the system works. If we consider $ETVW$ (50%) + $AVW$ (20%) + $GOVW$ (30%), then the complete weight = 100%. Supposing $ETV = 0.75$; if $EW = 0.25$ then $AV = 0.75$; if $BOV = 0.50$ then $GOV = 0.50$. Applying (1), we find the new TV, as in:

$$TV = 0.75 \times 0.50 + 0.75 \times 0.20 + 0.50 \times 0.30$$

$$TV = 0.375 + 0.15 + 0.15$$

$$TV = 0.675$$

TV is decreased by 0.075 if $EW = 0.25$ and $BOW = 0.50$. In this instance, the user can access XML database nodes that match $TV \leq 0.675$. 
4.6. Conclusion
This chapter highlighted the research motivations and defined the objectives. The research hypothesis is formulated and the system architecture is drawn. The proposed trust calculations may be developed and change in the future according to the different stages of the research.
Chapter 5: Conclusion

The report provides a literature review on access control models for XML databases. It focuses attention on several exiting access control systems and compares them. Both advantages and disadvantages of the existing models are discussed. All traditional types of access controls are static and they capture misuse from outsiders without paying attention to insider misuse. It is clear that the main techniques’ node filtering and query rewriting suffered from problems which cost too much time require. This leads many recent researches to propose different techniques to solve the existing problems.

The proposed research discusses how trust-based access control can apply to XML databases. It describes the advantages of using trust to protect personal data and provide a range of values for accessibility to data. It evaluates users’ behaviour and changes their access value depending on this behaviour. It tries to improve the access control performance for XML database by providing dynamic system.

It is important to tackle security problems in XML databases to reduce misuse and attacks. Also, improving the access control work is vital to protect the sensitive data and provide a secure environment to users.
References


References


Harold, E. and W. S. Means (2002). XML IN A NUTSHELL. O'REILLY.


References


