Supporting XML Security Models using RDBMS: A Vision

Dongwon Lee
Wang-Chien Lee
Peng Liu

Penn State University

Outline

- Motivation
- Related Work
- Framework and Issues
- Conclusion
Motivation

- Needs for secure exchange and evaluation of XML data increase
- Still majority of XML data/documents are originally from RDBMS
- RDBMS has solid support for security already
- Many recent works re-invent the wheels
  - New XML Security Models, but
  - Since original data are from RDBMS, and RDBMS has a good support for security model, why don't we use it?

Dongwon Lee

<table>
<thead>
<tr>
<th>XML Security Model</th>
<th>Relational Security Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Databases</td>
<td>RDBMS</td>
</tr>
</tbody>
</table>

Overview of Proposal

- XML data in RDBMS
  - Either XML documents are converted into RDBMS, or
  - They are originally stored there
- Security Admin specifies access controls in XML security model
- Users issue XML queries given XML view/schema
- Security check and query evaluation are done in RDBMS or Middleware atop RDBMS, not in XML side
- Valid and secure answers are returned to users

Dongwon Lee
Outline

- Motivation
- Related Work
- Framework and Issues
- Conclusion

Dongwon Lee
Related Work I: Security Models

- Relational security model: Multilevel, Discretionary, etc
- Most RDBMS implements discretionary model
  - GRANT / REVOKE
  - Table or column level authorization
  - Views for indirectly supporting contents-based access control
  - Role-based access control

- XML security model: several proposed recently, no consensus or standard yet – active research area
  - We pick one by Samarati et al [TISSEC 02]
- No XML databases fully support the proposed XML security model – only research prototypes

Samarati’s XML Security Model

- A Fine-Grained Access Control System for XML Documents
- Model: 5-tuple (subject, object, action, sign, type)
- Users submit an XML query $Q$ about an XML document $D$
- DOM tree $T$ is built from $D$
- Access-denied nodes (based on the pre-defined policies) are removed from $T$, yielding to a refined DOM tree $S$ (i.e., view)
- $Q$ is processed against $S$ and answers are returned to users
Related Work II: Conversion

- See Sekar’s Survey this morning [XSymp03]

- Conversion between XML documents and Relational data
  - Active research area

- Structure-based; use XML schema for conversion
  - STORED [SIGMOD 98], Inlining [VLDB 99], ...

- Data-based
  - Node/Edge [Data Eng. Bulletin 99], XRel [TOIT 01], ...

- We pick the XRel [TOIT 01]

---

Dongwon Lee

---

XRel’s Path-based Conversion

- View XML documents as a set of Paths
- Map such paths onto several tables in RDBMS
- Node information is captured via Root-to-leaf path
- Capture ancestor-descendent relationship via Region (ie, ordering scheme)

Element(docID, pathID, start, end, index, reindex)
Attribute(docID, pathID, start, end, value)
Text(docID, pathID, start, end, value)
Path(pathID, pathexp).

---

Dongwon Lee
Related Work

- We use
  - XML security model – Samarati’s Model
  - XML-RDBMS conversion method - XRel

- Our choice made at this point is arbitrary

- Finding which particular model or conversion methods are the best is one of the research issues

Outline

- Motivation
- Related Work
- Framework and Issues
- Conclusion
(1) Using RDBMS's Security Support

- A1: (Admin1, /A/B, Read, +)
- A2: (Admin2, /A/C, Read, +)
- A3: (Admin3, /A/B[./D>5], Read, +)
- A4: (Admin4, /A/B[./D>5]/E, Read, +)

Fine-grained access control can specify a set of nodes to force the security precisely.

- `<!ELEMENT A (B+, C, X)>`
- `<!ELEMENT B (D, E)>`
- `<!ELEMENT C (#PCDATA)>`
- `<!ELEMENT D (#PCDATA)>`
- `<!ELEMENT E (#PCDATA)>`
- `<!ELEMENT X (#PCDATA)>`

Inlining [VLDB99]

A (C, X)
B (D, E, fk_a)
(1) Using RDBMS’s Security Support

**A1:** /A/B

**A2:** /A/C

GRANT SELECT to USER Admin1 ON B

GRANT SELECT to USER Admin2 ON A(C)

---

(1) Using RDBMS’s Security Support

CREATE VIEW tmp AS
SELECT * FROM B
WHERE D>5;

GRANT SELECT to USER Admin3 on tmp;

**A3:** /A/B[./D>5]

---

Dongwon Lee
Issue: Granularity discrepancy between XML and relational security models

Issue: Model-to-model mapping (eg, expressive power, etc)
  - More theoretical study is needed
(2) Middleware Approach

- A underlying RDBMS is used as a simple storage
  - Its security support may or may not be used

- A middleware atop RDBMS handles security check via Query-Rewriting
  - Incoming queries from users are XML format
  - XML access controls are specified via XML queries too
  - Therefore, one can view the access control as additional query constraint over the incoming query

Dongwon Lee

---

(2) Middleware Approach

- Query Q, Access control rule R

- $Q + R \Rightarrow Q'$
- Answers that satisfy $Q'$ are:
  - Secure w.r.t R
  - Valid w.r.t Q

Dongwon Lee
(2) Middleware Approach

- XML security authorization table:
  - R1: (s1, /a/b, read, +, Recursive)
  - R2: (s1, /a/d, read, +, Recursive)

- Query Q: /a//c

- Q’: /a/ [b | d] // c
(2) Middleware Approach

- XML security authorization table:
  - R1: (s1, /a/b, read, --, Recursive)
  - R2: (s1, /a/d, read, --, Recursive)

- Query Q: /a//c

- Q': /a//c – (/a/b//c union /a/d//c)

Dongwon Lee

---

(2) Middleware Approach

- Issues...

- How to rewrite Q to Q'?
  - Which access control rule R to use?
  - How to quickly find?
  - Optimization issue for Q'
  - Security-specific features’ effect? (eg, recursive vs. local)
  - ...

Dongwon Lee
Outline

- Motivation
- Related Work
- Framework and Issues
- Conclusion

Conclusion

- L³ project
  - http://nike.psu.edu/l3/

- Preliminarily thoughts to support XML security models using RDBMS
  - Many approaches
  - Many issues
  - More details in the paper

- Hope to have more researchers interested in the topic
Samarati’s XML Security Model II

Def: 5-tuple (subject, object, action, sign, type), where:
- **Subject**: to whom the authorization is granted
- **Object**: nodes (returned by URI:XPath) to which the authorization is applied
- **Action**: read (select) or write (insert/delete/update)
- **Sign**: \{+,-\}, where + (access granted) and – (access forbidden)
- **Type**: \{LDH, RDH, L, R, LD, RD, LS, RS\}; i.e., local, recursive, …
XML Tree Example

Fig. 2. An XML tree.

Dongwon Lee

Tables

<table>
<thead>
<tr>
<th>docID</th>
<th>pathID</th>
<th>start</th>
<th>end</th>
<th>value</th>
<th>NodeID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>22</td>
<td>28</td>
<td>Michael</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>45</td>
<td>52</td>
<td>Franklin</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>125</td>
<td>172</td>
<td>Comparative Analysis of the XML Schemas Languages</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>205</td>
<td>241</td>
<td>Dongwon</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>223</td>
<td>230</td>
<td>Lee</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>264</td>
<td>269</td>
<td>Wesley</td>
<td>21</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>286</td>
<td>287</td>
<td>W.</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>305</td>
<td>307</td>
<td>Chu</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>345</td>
<td>347</td>
<td>As</td>
<td>27</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>357</td>
<td>359</td>
<td>XML</td>
<td>29</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>370</td>
<td>690</td>
<td>is emerging as the ...</td>
<td>30</td>
</tr>
</tbody>
</table>

Path

<table>
<thead>
<tr>
<th>pathID</th>
<th>pathType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>#some</td>
</tr>
<tr>
<td>2</td>
<td>#is Gaul</td>
</tr>
<tr>
<td>3</td>
<td>#is Gaul/first</td>
</tr>
<tr>
<td>4</td>
<td>#is Gaul/finally</td>
</tr>
<tr>
<td>5</td>
<td>#is Gaul/articles</td>
</tr>
<tr>
<td>6</td>
<td>#is Gaul/articles/article</td>
</tr>
<tr>
<td>7</td>
<td>#is Gaul/articles/article/@category</td>
</tr>
<tr>
<td>8</td>
<td>#is Gaul/articles/article/title</td>
</tr>
<tr>
<td>9</td>
<td>#is Gaul/articles/article/authors</td>
</tr>
<tr>
<td>10</td>
<td>#is Gaul/articles/article/authors/author</td>
</tr>
<tr>
<td>11</td>
<td>#is Gaul/articles/article/authors/author/first</td>
</tr>
<tr>
<td>12</td>
<td>#is Gaul/articles/article/authors/author/finally</td>
</tr>
<tr>
<td>13</td>
<td>#is Gaul/articles/article/authors/author/middle</td>
</tr>
<tr>
<td>14</td>
<td>#is Gaul/articles/article/summary/keyword</td>
</tr>
</tbody>
</table>

Dongwon Lee