

Security-Conscious XML Indexing



Yan Xiao, Bo Luo, [Dongwon Lee](#)

The Pennsylvania State University
U.S.A.



DASFAA 2007

Agenda

- Background
- Contribution
- Access Control Model
- Five Indices
- Experimental Results
- Conclusion

Motivation

- Secure exchange and sharing of XML data needs XML access controls
- Secure XML query processing = locating XML nodes that:
 - Satisfy query constraints
 - Do not violate security policies
- In this paper, we propose various indexing schemes that speed up this secure XML query processing

Background

Two-tier Coarse Indexing Scheme for MLS Database Systems

(Sushil Jajodia. DB Security, 1998)

- Two-tier Indexing Scheme :
 - Improve query response time
 - Reduce the storage required for indexing
- Three Index Structures:
 - Single-level Index
 - Global Multilevel Index
 - Two-tier Coarse Index

Background

Indexing and Querying XML Data for Regular Path Expressions

(Quanzhong Li, Bongki Moon, 27th VLDB Conference, 2001)

- Numbering Scheme
- XISS - a system for indexing and storing XML data based on a numbering scheme.
 - Three major index structures (element index, attribute index, structure index)
- Path-join algorithms:
 - Decompose regular path expressions
 - Path-join algorithms

Contribution

- Add Access Control Function to XISS
- Five Index Structures
 - Global Index (GI)
 - Single-level Index (SLI)
 - Minimum-Security Index (MSI)
 - Skip-Record Index (SRI)
 - Skip-Forward Index (SFI)
- Analysis and Experimental Results

Access Control Model

- Multilevel Security Model
 - Total order
 - L security levels from 1 (lowest) to L (highest)
- Each XML element and attribute will be specified a security level.
 - Assign security levels in DTD/Schema Assign security levels in XML documents
Eg: ``
 - SL in XML documents can overwrite that in XML schema
 - If no security is assigned, security level is 1
- XML query with specified level is written as
 - `{L}:Q` (e.g., `{3}://a//b`)



Access Control Model

- User will be assigned a security level.
- Only XML data whose security requirement is **lower than** or **equal to** user's authorization can be returned.

```
<?xml version="1.0" encoding="utf-8"?>
<books>
  <book>
    <author>
      <file s_title="1" s_price="1" s_unit="1">XML Indexing</file>
    </author>
    <title s_title="2">XML Indexing</title>
    <price s_price="3" unit="USD" s_unit="3">50</price>
  </book>
</books>
```

User's assigned security level ≥ 3 , can access *title, price, unit.*

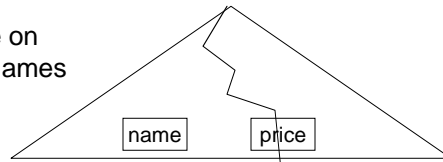
User's assigned security level < 2 , can not access *title, price, unit.*

User's assigned security level = 2, can access *title.* cannot access *price, unit.*



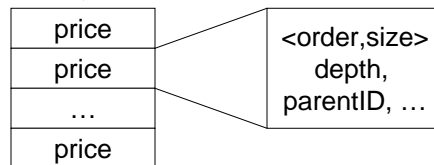
XISS Index Structure

B+ Tree on element names



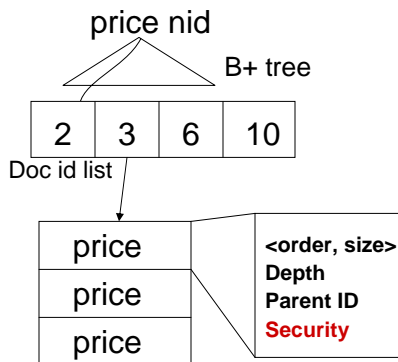
DOC ID list
2 | 5 | 10 | ... | 34 | -1

Element list



Global Index (GI)

Query: {3}://price

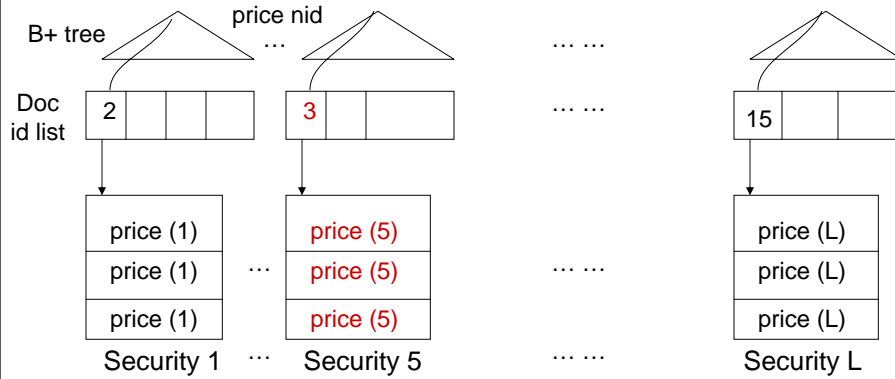


Query Step:

1. Find a document list for the element;
2. Security check for the element.

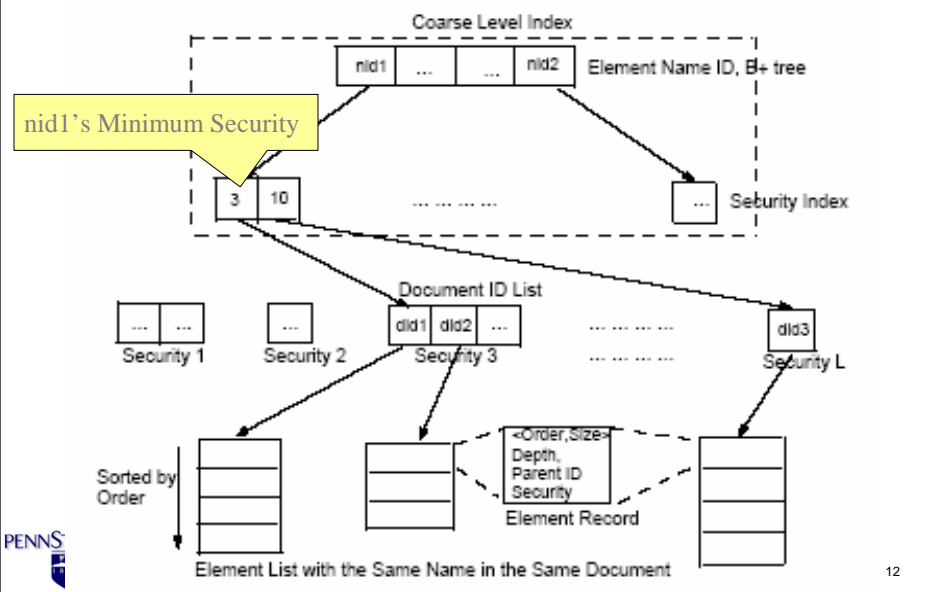
Disadvantages: Security check at element level
Not efficient for single-level queries

Single-level Index (SLI)

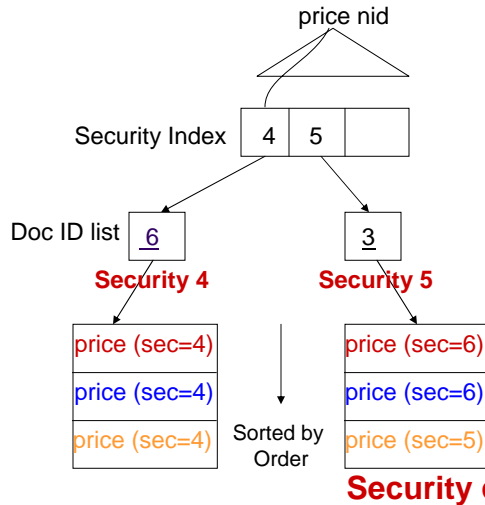


- Single-level Index: Separate Index at each security level.
- Disadvantages: Use more storage room
Not efficient for range queries

Minimum Security Index (MSI)



Minimum Security Index (MSI)



Query:

- If user's security < 4, no documents need to be checked.
- If user's security = 4, check document 6 only.
- If user's security >= 5 check document 6 and 3.

Security check at document level



Disadvantages: not effective if the element has big confidentiality difference within a document.

Skip-Record Index (SRI)



Query: {3}://price

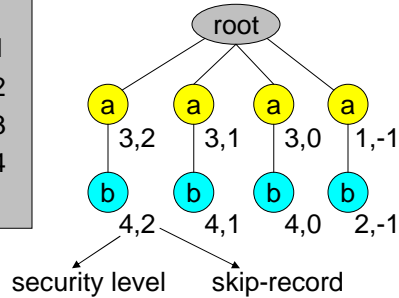


Skip-Forward Index (SFI)

```

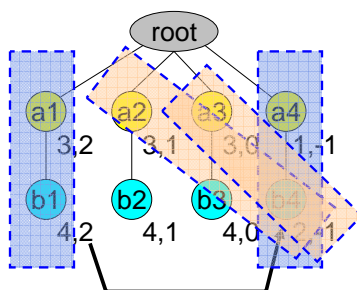
<root>
  <a s_a=3><b s_b=4/> </a> // a1, b1
  <a s_a=3><b s_b=4/> </a> // a2, b2
  <a s_a=3><b s_b=4/> </a> // a3, b3
  <a s_a=1><b s_b=2/> </a> // a4, b4
</root>

```



- Query: $\{3\}://a/b$
 - Get all “a” nodes with satisfactory constraints
 - Get all “b” nodes with satisfactory constraints
 - Sort-merge two lists and return all final “b”

Skip-Forward Index (SFI)



Monotonic security model:
ancestors' SL \leq
descendents' SL

- $\{3\}://a/b$
 - a list: a1(3,2), a2(3,1), a3(3,0), a4(1,-1)
 - b list: b1(4,2), b2(4,1), b3(4,0), b4(2,-1)
- a1-b1 pair
 - Sort-merge: satisfy “/” relationship
 - Security check: b1’s SL > 4: X
- a1-b2 & a1-b3 pairs: skipped
- a2-b4 pair
- a3-b4 pair ...

Experiment and Results

Mnemonic	Monotonic Model	Non-Monotonic Model
Uniform Security Distribution	UM	UNM
Skewed Security Distribution (More low security level data)	S1M	S1NM
Skewed Security Distribution (More high security level data)	S2M	S2NM

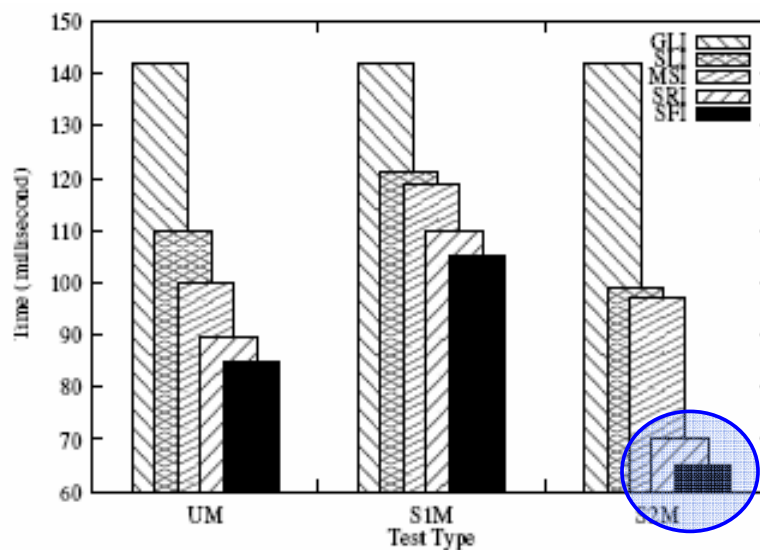
Table 2: Test sets.

security	1	2	3	4	5	6	7	8	9	10	11	Total
UM	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	66000
S1M	11000	10000	9000	8000	7000	6000	5000	4000	3000	2000	1000	66000
S2M	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000	66000

Table 3: Security distribution for monotonic model (site/*/listitem(security))



Experiment and Results



Conclusion

- Proposed five variations of indices that support security checks for multi-level based XML access controls
- SRI or SFI was able to improve other variations up to 130% at best
- Future work
 - Extend to other state-of-the-art XML indexing schemes
 - Extend to DAC or RBAC models