# XML 문서의 안전한 브로드캐스팅을 위한 동적인 레이블링 기법*

김민정[0] 고혜경 이상근
고려대학교 컴퓨터학과
{cara2847, ellefgt, yalphy}@korea.ac.kr

# Dynamic Labeling Scheme for Secure Broadcasting of XML Document

Min-Jeong Kim[0] Hye-Kyeong Ko SangKeun Lee
Department of Computer Science and Engineering, Korea University

요 약

XML이 데이터 표현과 문서 교환의 표준으로 떠오름에 따라 XML 문서에 대한 보안이 중요하게 되었다. 이 논문에서는 XML 보안을 위하여 W3C의 암호화 방법에 접근 제어 기법을 적용시킨 XML Pool Encryption 방법을 기반으로 XML 문서의 구조 정보를 빠르고 쉽게 파악할 수 있는 새로운 레이블링 기법을 제안한다. 제안하는 기법은 계층 구조의 특성을 갖는 XML 문서에서 하위 정보의 위치에 상위 정보의 위치를 포함시킴으로써 문서 일부에 대해서도 구조 정보의 유추가 가능하다. 또한 XML 문서의 변화(문서의 갱신, 수정, 삭제) 시에도 문서 전체의 레이블링을 변화시키지 않고, 변화하는 부분에 대해서만 새로운 레이블을 부여한다는 장점이 있다. 실험 결과에서는 제안된 기법이 XML 문서상에서의 위치 정보를 찾는 데에 효율적임을 보여준다.

## 1. 서 론

XML(eXtensible Markup Language)[13]이 웹상에서 정보를 교환하기 위한 문서 표현의 표준으로 자리잡아가면서 정부나 기업 등에서 사용하는 문서도 XML로 표현되고 있다. 뿐만 아니라 XML은 ebXML, 웹서비스 및 차세대 인터넷 기술인 Semantic Web 등에서 문서의 표준으로 정착되고 있어 그 사용이 급격하게 증가하고 있다. 이에 따라 XML 문서의 보안에 대한 관심이 크게 증가되었다. W3C의 XML 표준화 그룹[10]에서는 XML 문서에 대한 보안 서비스 제공을 위하여 XML 문서에 대한 전자서명[12], 키 관리[14], 암호화[11] 등에 관한 표준화 작업을 진행하고 있다. 또한, XML 문서가 갖는 구조적 특성을 활용하여, XML의 부분적 정보의 의미에 따른 사용자 별 접근 제어를 수행하도록 하는 XML 접근 제어 방법이 연구되고 있다[2, 4, 5]. XML Pool Encryption[6]에서는 XML 암호화 방법에 XML 접근 제어를 적용하여 중요한 정보에 대한 부분적인 암호화 및 사용자 별 접근 제어가 가능하도록 하고 있다. 사용자들은 부분적으로 암호화 된 XML 문서를 받으며, 암호화된 정보 중, 자신의 권한에 해당되는 부분에 대해서 복호화 한 후 XML 문서로 재구성하여 해당 내용을 볼 수 있게 된다. 문서를 재구성하는 과정에서 XML 문서를 이루는 정보 간의 구조 정보 파악을 위하여 XML 문서를 이루는 기본 단위인 엘리먼트에 대하여 의미 있는 구조 정보를 부여한다.

이 논문에서는 XML 문서에 대한 구조 정보의 표현으로 엘리먼트 단위의 레이블링 기법을 제안한다. 제안하는 레이블링 기법은 상위 엘리먼트의 위치 정보를 하위 엘리먼트의 위치 정보에 포함시킴으로써 XML 문서의 재구성 과정 시에 필요한 구조 정보 파악을 빠르고 쉽게 한다. 또한 XML 문서에 새로운 정보가 추가되었을 때 문서의 일부 혹은 전체가 아닌 그 정보에 대해서만 새로운 레이블을 부여한다.

본 논문의 구성은 다음과 같다. 2장에서는 지금까지 연구되고 있는 XML 암호화와 관련된 연구에 대해 살펴보고, 3장에서는 XML 문서의 안전한 브로드캐스팅을 위하여 논문에서 제안하는 레이블링 기법에 대해서 설명한다. 4장에서는 다양한 실험을 통하여 제안하는 레이블링 기법의 성능을 평가한다. 마지막으로, 5장에서는 결론을 제시하며 향후 연구에 대하여 기술한다.

## 2. 관련연구

XML 문서에 대한 보안을 위해 XML 암호화 표준화 그룹[10]에서는 XML 암호화 방법이 진행되고 있으며, 세밀한 암호화를 위하여 XML 암호화 방법에 XML 접근 제어 방법을 적용한 XML Pool Encryption[6]이 연구되고 있다.

### 2.1 XML 암호화

W3C의 XML 암호화 표준화 그룹[10]에서는 XML이 문서 교환의 표준으로 자리잡으면서 부각되어온 보안 문제를 해결하기 위하여 XML 문서에 대한 보안 서비스를 제공하고 있다. XML 암호화는 서브 트리 단위로 암호화를 하여, 암호화될 노드가 선택되면 그 하위 노드도 함께 암호화한다. 또한, XML 암호화는 하나의 XML 문서에 대해서 사용자 권한에 따라 문서의 내용을 다르게 볼 수 있도록 여러 명의 사용자에 대해서 암호화하는 중복 암호화(super encryption)를 제공한다.

### 2.2 XML Pool Encryption

W3C의 XML Encryption[11]은 XML 문서 암호화를 위한 표준을 제공하고 있지만, 몇 가지 문제점을 지니고 있다. XML Encryption은 서브트리 기반으로 암호화를 하므로 암호화가 필요 없는 하위 정보가 암호화의 범위에 포함된다. 또한 하나의 XML 문서를 여러 명의 사용자에 대해서 중복 암호화를 하기 때문에 사용자는 받아본 문서 상에 자신이 볼 수 없는 암호화된 부분이 있다는 것을 알게 되어 비밀 정보의 존재를 유추할 수 있다. 이러한 문제점을 해결하기 위하여 XML Encryption[11] 방법과 XML 문서에 대한 세밀한 접근을 제공하는 XML Access Control[5]을 적용한 XML Pool Encryption[6] 방법이 연구되었다. 이 방법은 엘리먼트 단위의 암호화를 제안하고 있으며, 암호화되어야 할 엘리먼트들을 모아서 암호화 풀(pool)에 저장하여 관리한다. 암호화가 필요한 엘리먼트들을 따로 관리함으로써 XML 문서에 대한 세밀한 암호화와 사용자에 따라 서로 다른 암호화가 가능하다.

#### 2.2.1 XML Pool Encryption의 레이블링 기법

XML Pool Encryption[6]에서는 노드의 위치 정보를 표현하기 위하여 "Trees in SQL" [3]에서 제안한 Adjacency List Mode (ALM)을 수정한 "Modified Adjacency List Mode (MALM)" 방법을 사용한다. 모든 엘리먼트는 <left, right>의 번호를 가지며, 상위 엘리먼트는 하위 엘리먼트들을 포함하는 값을 갖는다. MALM은 암호화되는 노드와 그렇지 않은 노드를 분리하여 레이블링 한다. 암호화되지 않은 노드에 대해서는 일정한 간격의 번호를 매기며, 암호화되는 노드는 매겨진 번호 사이의 임의의 번호로 레이블링을 하므로 노드 사이에 다른 노드들이 암호화 되

* 본 결과물은 정보통신부의 정보통신 기초기술연구지원산업 (정보통신연구진흥원)으로 수행한 연구결과입니다.

었다는 것을 유추해 낼 수 없다. 그러나 MALM 기법은 XML 문서상에서 암호화된 노드의 위치 파악을 위해 레이블을 비교해야하는 노드의 수가 많으며, 최악의 경우 XML 문서의 모든 노드를 검색해야 한다.

## 3. 제안하는 레이블링 기법

XML 문서의 안전한 브로드캐스팅을 위하여 연구되었던 XML Pool Encryption에서 사용된 노드 레이블링 방법의 단점을 보완하여 동적인 갱신을 지원하는 새로운 레이블링 기법을 제안한다. 풀(pool)에 저장되어 있는 암호화된 노드가 복호화 된 후, XML 문서에서의 알맞은 위치를 찾기 위해 노드 간의 관계를 파악하는 것이 중요하다. 제안하는 레이블링 기법은 노드 간의 관계를 빠르게 파악할 수 있고, 암호화된 노드를 분리하여 레이블링하지 않으므로 정책(policy)의 변화가 있어도 모든 노드의 레이블링을 다시 하지 않는 장점이 있다. 또한 노드마다 유일한 레이블링을 부여하여 XML 문서의 동적인 갱신을 지원한다.

### 3.1 레이블링의 기본 구성

모든 노드의 레이블은 그림 1과 같이 크게 세 가지 구성요소로 구분되어 유일한 레이블을 구성한다.

| (C1) Level component | (C2) Inherited label component from parent node | (C3) Sibling order component |
|---|---|---|

그림1. 레이블의 구성 요소

- Level component(C1)-XML 트리 상에서 해당 노드의 레벨을 의미
- Inherited label component(C2)-부모 노드 레이블 중에서 C1을 제외한 부분을 상속 받는다.
- Sibling order component(C3)-형제 노드들간의 상대적인 위치

이 세 개의 구성요소들은 "." 로 연결되어 각 노드에 대한 유일한 레이블이 생성된다.

### 3.2 XML 문서의 레이블링 기법

XML 트리를 구성하는 노드에 대한 레이블링은 루트노드와 루트노드가 아닌 노드의 두 가지 경우로 수행된다.

정의 1. 루트 노드 $r$에 대한 레이블

$L(r) = C1_{root\,r} \,.\, C2_{root\,r} \,.\, C3_{root\,r} = 1.nil.1$

정의 2. 루트가 아닌 노드 $x$에 대한 레이블

$L(x) = C1_{node\,x} \,.\, C2_{node\,x} \,.\, C3_{node\,x}$

$C1_{node\,x}$ = 현재 노드의 레벨

$C2_{node\,x} = C2_{parent\_node}$ 와 $C3_{parent\_node}$ 를 연결시킨다.

$C3_{node\,x} = C3_{node\,x}$의 형제 순서

그림 2는 제안된 방법으로 XML문서를 레이블링한 결과를 나타낸다.

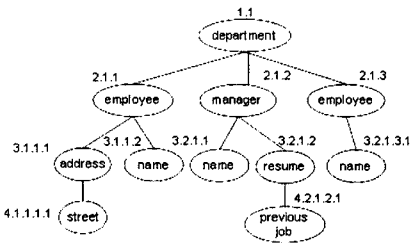



그림2. 레이블링 된 XML 트리

### 3.3 XML 문서의 동적인 갱신을 지원하는 레이블링 기법

XML 문서에 삽입되는 노드에 대하여 새로운 레이블을 부여하며, 이 때 삽입되는 위치에 따라, 가장 왼쪽에 삽입되는 경우, 가장 오른쪽에 삽입 되는 경우, 그리고 두 노드 사이에 삽입되는 세 가지 경우로 나눌 수 있다. 세 가지 경우에 대한 노드 레이블링은 그림 3과 같다. 또한 삽입되는 노드의 위치에 따른 새로운 레이블은 그림 4에 기술된 알고리즘 1 에 의해 새로운 노드에 부여된다.

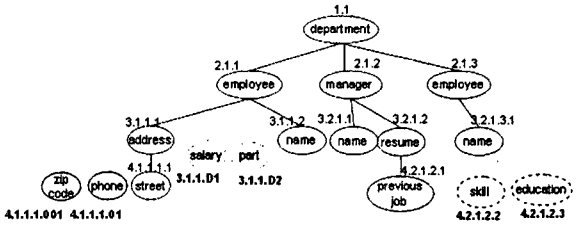



그림 3. 새로운 노드가 삽입된 레이블

**알고리즘 1**

**Input** : $pos$ which is the position to insert into the new node $p_1, p_2, \cdots p_n$
**Output** : label of the new node $p$ $L(p)$

For($i$<$n$){
  **If**($pos$ is the leftmost)
    search for the label $L(x)$of current leftmost node $x$
      $C1_{p_i}$ and $C2_{p_i}$ are the same label of $L(x)$, respectively
      $C3_{p_i}$ is made to be inserted to '0','00',.. in front of $C3$
  **else if** ($pos$ is the rightmost)
    search for the label $L(x)$of current rightmost node $x$
      $C1_{p_i}$ and $C2_{p_i}$ are the same label of $L(x)$, respectively
      $C3_{p_i}$ is made to be increased by one of $C3_{p_{i-1}}$
  **else if** ($pos$ is between nodes)
    search for the label $L(x)$and $L(y)$ which are continuous nodes
    (node $y$ is located on the right of node $x$)
      $C1_{p_i}$ and $C2_{p_i}$ are the same label of $L(x)$, respectively
      $C3_{p_i}$ is made to be inserted to '$D_1$' .. '$D_n$' the end of the $C3_{p_{i-1}}$
}

그림 4. 동적인 갱신을 지원하는 알고리즘

### 3.4 XML 문서상에서의 노드의 위치 정보

제안하는 레이블링 기법은 풀(pool)에 저장되어 있는 암호화된 노드에 대하여 XML 문서상에서 해당 되는 위치 정보를 찾기 위하여 노드간의 관계를 빠르게 파악한다.

*부모-자식 관계*

노드 $x$가 노드 $y$의 부모 노드일 경우 두 노드의 레이블은 다음 관계를 만족한다.

1. $C1_{parent\_node\,x} = C1_{child\_node\,y}$ 의 레벨을 하나 감소한다.
2. $C2_{parent_node\,x} = C2_{child_node\,x}$의 마지막 부분을 제거한 substring 이다.
3. $C3_{parent\_node\,x} = C2_{child\_node\,x}$ 의 마지막 부분이다.
4. $C2_{parent\_node\,x}$ 과 $C3_{parent_node\,x}$를 연결한 문자열 = $C2_{child_node\,y}$이다.

*조상-후손 관계*

노드 $x$가 노드 $y$의 조상 노드일 경우 두 노드의 레이블은 다음 관계를 만족한다.

1. $C1_{ancestor\_node\,x} < C1_{descendant\_node\,y}$ 이고,
2. $C2_{ancestor\_node\,x} = C2_{descendant\_node\,y}$ 에서 길이 $C1_{descendant\_node\,y} - C1_{ancestor\_node\,x}$ 만큼에 해당되는 부분 문자열이다.
3. $C3_{ancestor\_node\,x} = C2_{descendant\_node\,y}$ 에서 길이 $C1_{ancestor\_node\,x}$ 만큼에 해당되는 부분 문자열의 마지막 문자이다.
4. $C2_{ancestor_node\,x} \subset C2_{descendant_node\,y}$을 만족한다.

*형제 관계*

노드 $x$와 노드 $y$가 형제노드일 경우 두 노드의 레이블은 다음 관계를 만족한다.

1. $C1_{right\_node\,x} = C1_{left\_node\,y}$ 이고,
2. $C2_{right\_node\,x} = C2_{left\_node\,y}$ 이며,
3. $C3_{right\_node\,x} = C3_{left\_node\,y}$ 보다 크기가 1 크다.

## 4. 실험 및 평가

논문에서 제안하는 레이블링 기법에 대한 성능 평가를 위하여 암호화되어 풀(pool)에 저장된 노드가 복호화 된 후에 XML 트리에서의 위치를 찾는데 걸리는 시간을 측정하였고, 이를 MALM 기법과 비교 하였다.

### 4.1 실험데이터

성능 평가를 위하여 0.1MB~6.9MB 크기의 XMark[8] 데이터 셋과 DBLP[7] DTD를 사용하여 IBM Generator[1]를 통해 0.025MB~1.4 MB 의 XML 문서를 생성하였다. 풀(pool)에 저장할 노드를 선택하기 위

하여 XPath[9] 표현식을 사용하였다.

### 4.2 실험 결과

제안하는 레이블링 기법의 성능 평가를 위하여 풀(pool)에 저장되어 있는 노드의 위치를 찾는데 걸리는 시간을 측정 하였다. 이 때 걸리는 시간과 노드의 개수와의 관계를 찾기 위하여 암호화 되는 노드의 개수 및 노드의 위치를 찾기 위하여 비교되는 노드의 개수를 측정하였다.

#### 4.2.1 암호화되는 노드의 개수

그림 5는 두 개의 데이터 셋에 대하여 XPath로 표현된 위치의 노드를 암호화 할 경우, 이에 만족하는 노드의 개수를 나타낸 것이다.

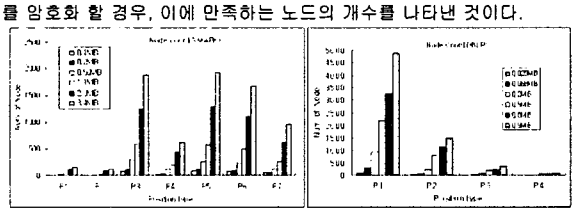


그림 5. 위치에 따른 암호화 노드의 개수

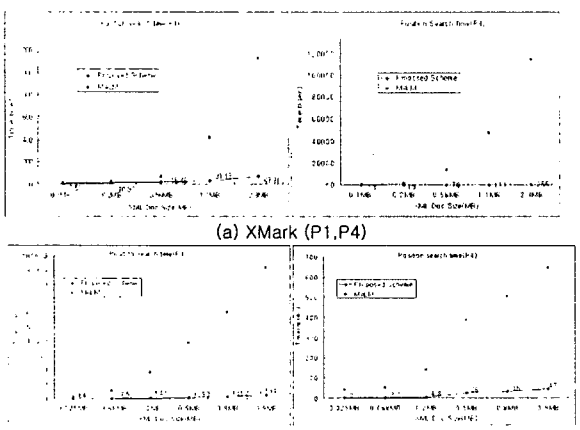
#### 4.2.2 암호화 되는 노드의 개수와 위치를 찾는데 걸리는 시간과의 관계

암호화 되는 노드는 풀(pool)에 저장되며, 복호화 되어 XML 문서에서 알맞은 위치를 찾기 위하여 노드의 레이블 비교가 필요하다. MALM 기법은 문서상에서 노드의 위치를 찾기 위하여 자신의 레이블에서 "left" 값보다 작은 값을 갖는 노드의 모든 레이블을 비교 해야 한다. 반면, 제안하는 기법은 위치를 찾으려는 노드의 레이블 만으로도 부모, 조상 노드의 레이블을 유추해 낼 수 있으므로 다른 노드들과의 레이블을 비교하지 않아도 된다.

표 1. 비교 노드의 개수

| | XMark (0.1MB) | | | DBLP (0.029MB) | |
|---|---|---|---|---|---|
| | 제안기법 | MALM | | 제안기법 | MALM |
| P1 | 12 | 84 | P1 | 94 | 6,197 |
| P2 | 3 | 21 | P2 | 31 | 2,064 |
| P3 | 79 | 22,452 | P3 | 9 | 701 |
| P4 | 20 | 2,514 | P4 | 3 | 61 |
| P5 | 91 | 42,572 | | | |
| P6 | 80 | 36,306 | | | |
| P7 | 55 | 24,557 | | | |

표 1을 통하여 암호화 되는 노드의 개수가 많을수록, 위치를 찾기 위해 비교해야 할 노드의 수가 많음을 알 수 있다.



(a) XMark (P1,P4)

(b) DBLP(P1, P4)

그림 6. 문서의 크기에 따른 노드의 비교 검색 시간

그림 6 (a),(b)는 각각 두 개의 데이터 셋에 대하여, Position Type P1,P4 이 나타내는 위치에 대한 문서의 크기에 따른 노드의 비교 검색 시간을 보여준다. 이를 통하여 문서의 크기가 클수록 암호화 되는 노드의 개수가 증가하며, 레이블을 비교하는 노드의 개수가 많으므로 위치를 찾는데 필요한 시간도 증가함을 알 수 있다.

## 5. 결 론

XML 문서가 웹 환경에서 데이터를 표현하는 표준으로 자리 잡으면서 네트워크를 통하여 분산되고 공유될 수 있기 때문에 XML 문서에 대한 보안이 중요하게 되었다. 이에 대하여 W3C 에서는 반 구조적 특성을 갖는 XML 문서를 서브 트리 단위로 암호화가 가능하도록 하는 XML Encryption[11]을 제공한다. 또한 XML Encryption 이 갖는 몇 가지 한계점을 극복하기 위하여 XML 문서에 대한 다양한 레벨에서의 접근 제어 방법을 XML Encryption 에 적용한 XML Pool Encryption [6] 방법이 연구되었다.

이 논문에서는 웹 환경에서 XML 문서의 안전한 브로드캐스팅을 위한 동적인 레이블링 기법을 제안하였다. XML Pool Encryption 은 XML 문서를 브로드캐스팅 할 때 암호화 되는 노드를 문서에서 분리하여 풀(pool)에 저장하여 암호화되지 않은 문서의 일부와 함께 사용자에게 보낸다. 이 때 사용자들이 풀(pool)에 있는 암호화된 노드들 중에서 자신의 권한에 맞는 노드를 복호화 하여 XML 문서 형태로 볼 수 있도록 하기 위해서는 복호화된 노드의 위치를 찾을 수 있어야 한다. 제안하는 레이블링 기법은 XML 문서상에서 노드의 알맞은 위치를 찾기 위하여 노드 간의 계를 빠르게 파악할 수 있다. XML 문서를 구성하는 노드에 대해 매겨지는 레이블은 부모 노드의 레이블 일부를 계승 받음으로써 레이블 자체에 부모 및 조상 정보를 포함하고 있다. 그렇기 때문에 노드들 간의 관계 파악 시 모든 노드들과의 레이블 비교가 필요 없으며, 임의의 한 노드의 레이블 만으로도 조상과 부모 노드의 정보를 유추할 수 있다.

제안된 방법과 MALM 방법을 비교한 결과 암호화된 노드의 개수에 따라 노드의 위치를 찾기 위해 비교되는 노드의 수가 제안된 방법이 훨씬 적었고, 문서의 크기에 따른 노드의 검색 시간에 있어서도 제안된 방법이 좋은 성능을 보였다.

## 참고문헌


[1] S. Abiteboul, P. Bunneman, and D. Suciu, "Data on the Web:From Relations to Semistructured Data and XML". Morgan Kaufmann, 1999.

[2] E. Bertino, S. Castano, E. Ferrari, M. Mesiti, "Controlled Access and Dissemination of XML Document". *ACM Web Information and Data Management*, pages 22-27, 1999.

[3] Joe Celko, "Trees in SQL". http://www.intelligententerprise.com/001020/celko1_1.shtml (October 2000).

[4] E. Damiani, S.De Capitani di Vimercati, S.Paraboschi, and P.Samarati, "Securing XML Documents". *EDBT*, pages 121-135, 2000.

[5] E. Damiani, S.De Capitani di Vimercati, S.Paraboschi, and P.Samarati, "A Fine-Grained Access Control System for XML Documents". *ACM Transactions on Information and System Security*, 5(2):169-202, 2002.

[6] Christian Geuer-Pollmann, "XML Pool Encryption". *ACM Workshop on XML Security*, pages 1-9, 2002.

[7] Michael Ley. DBLP database web site. http://informatik.uni-trier.de/ley/db, 2000.

[8] A. Schmidt, F. Waas, M. L. Kersten, M. J. Carey, I. Manolescu, and R. Busse. Xmark : A benchmark for xml data management. *VLDB*, pages 974-985, 2002.

[9] XPath. http://www.w3c.org/TR/XPath

[10] W3C. "XML Encryption WG". http://www.w3.org/Encryption/2001/

[11] W3C. "XML Encryption Syntax and Processing". W3C Recommendation. http://www.w3.org/TR/xmlenc-core/ (December 2002).

[12] W3C. "XML-Signature Syntax and Processing". W3C Recommendation. http://www.w3.org/TR/xmldsig-core/ (February 2002).

[13] W3C. "eXtensible Markup Language (XML) 1.0". World Wide Web Consortium (W3C). http://www.w3.org/TR/REC-xml/ (February 2004).

[14] W3C. "XML Key Management Specification (XKMS 2.0)". W3C Recommendation. http://www.w3.org/TR/xkms2/ (June 2005).