KIPS Tr. Comp. and Comm. Sys.
Vol.5, No.9 pp.217~222 pISSN: 2287-5891



http://dx.doi.org/10.3745/KTCCS.2016.5.9.217

# Efficient Data Publishing Method for Protecting Sensitive Information by Data Inference

Hye-Kyeong Ko†

## ABSTRACT

Recent research on integrated and peer-to-peer databases has produced new methods for handling various types of shared-group and process data. This paper with data publishing, where the publisher needs to specify certain sensitive information that should be protected. The proposed method cannot infer the user's sensitive information is leaked by XML constraints. In addition, the proposed secure framework uses encrypt to prevent the leakage of sensitive information from authorized users. In this framework, each node of sensitive data in an eXtensible Markup Language (XML) document is encrypted separately. All of the encrypted data are moved from their original document, and are bundled with an encrypted structure index. Our experiments show that the proposed framework prevents information being leaked via data inference.

Keywords : Data, Encrypted Structure Index, Sensitive Information, Inference, Data Publishing

# 데이터 추론에 의한 민감한 정보를 보호하기 위한 효율적인 데이터 출판 방법

고 혜 경†

## 요 약

최근의 통합 시스템 및 P2P에 대한 데이터베이스의 연구는 다양한 공유된 그룹 및 프로세스 데이터를 위한 새로운 방법들이 개발되었다. 본 논문에서는 XML 제약에 의해 유출될 수 있는 민감한 정보에 대한 사용자의 유추를 원칙적으로 차단하고 권한 부여가 되지 않은 사용자로부터 민감한 정보가 유출되지 않도록 암호화 방법을 이용하여 안전한 데이터 출판 프레임워크를 제안한다. 제안된 프레임워크에서는 XML 문서 내의 민감한 데이터의 각각의 노드는 따로 분리하여 암호화하고 암호화된 모든 데이터들은 본래의 문서로부터 분리되어 민감한 데이터의 각각의 노드는 따로 암호화된다. 암호화된 모든 데이터들은 원래의 문서로부터 분리하여 암호화된 구조 인덱스로 묶어 보호된 데이터를 출판한다. 실험 결과로 제안된 프레임워크는 익명의 사용자로부터 데이터 추론을 통한 사용자 정보 누설을 방지함을 보여준다.

키워드 : 데이터, 암호화된 구조 인덱스, 민감한 정보, 추론, 데이터 출판

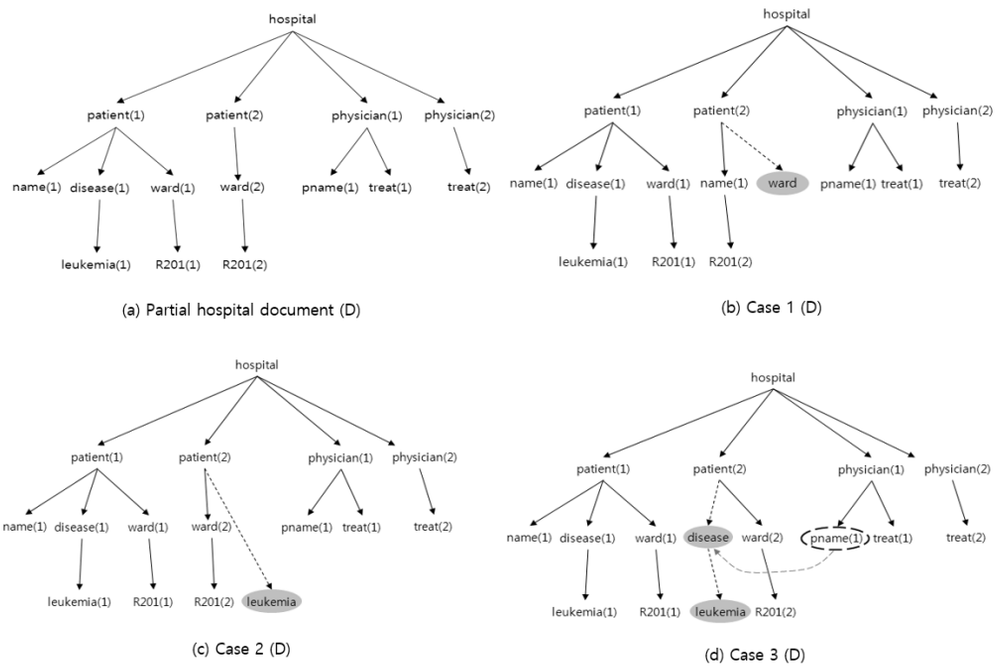
## 1. 서 론

최근의 연구들은 주로 데이터 통합과 P2P 데이터베이스에 관한 연구들로 다양한 공유된 그룹 및 프로세스 데이터를 처리하기 위한 새로운 방법들이 연구되고 있다[1-3]. XML (eXtensible Markup Language)을 획기적으로 개선하여 만든 언어로 웹 페이지 구축 기능과 검색 기능 등이 향상되었고, 웹페이지의 추가와 작성이 편해졌으며 클라이언트 시스템의 복잡한 데이터 처리를 쉽게 할 수 있다[4]. 이 밖에 HTML은 웹 페이지에서 데이터베이스처럼 구조화된 데이터를 지원할 수 없지만 XML은 사용자가 구조화된 데이터베이스를 뜻대로 조작할 수 있는 장점을 갖고 있다[5].

인터넷의 빠른 발전과 함께 웹상에서 출판되는 데이터의 양은 방대하게 증가하고 있으며, 최근 출판된 XML 문서들도 사용자 기반 어플리케이션이 증가하면서 보안에 대한 중요성이 증가되고 있다. 이러한 어플리케이션에서 데이터 소유자는 웹에서 다른 사용자에게 데이터를 출판할 수 있는 권한을 갖는다[6].

만일 데이터 소유자가 안전하지 않게 데이터를 출판한다

※ 이 논문은 2016년도 정부 (미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.NRF-2014R1A1A3051552).
† 정 회 원 : 성결대학교 컴퓨터공학부 조교수
Manuscript Received : August 9, 2016
Accepted : August 29, 2016
* Corresponding Author : Hye-Kyeong Ko(hkko@sungkyul.ac.kr)




(a) Partial hospital document (D)

(b) Case 1 (D)

(c) Case 2 (D)

(d) Case 3 (D)

Fig. 1. Hospital document inferred using XML constraints (The gray circles represent inferred nodes)

면 사용자는 본인이 가진 일반적인 정보의 유추를 이용하여 권한이 부여되지 않은 문서의 정보를 출판된 XML 문서로부터 알아낼 수 있다. 예를 들어 익명의 다른 사용자로부터 병원의 환자 정보와 같은 민감한 정보의 추론이 가능하기 때문에 보안의 위협이 발생하게 된다.

본 논문에서 제안하는 안전한 데이터 출판을 위한 프레임워크는 암호화 기술에 기반을 두어 권한 부여가 되지 않은 사용자로부터 민감한 정보를 유추할 수 없게 정보를 보호할 수 있다.

제안된 프레임워크에서 XML 문서 내의 각각의 민감한 정보는 따로 암호화 되고 암호화된 민감한 정보들은 원래의 문서로부터 분리되어 암호화 정보 셋 (Encrypted information set)으로 이동되어 저장된다. 그리고 암호화 된 노드들의 구조적 정보를 나타내는 암호화 구조 인덱스 (Encrypted structure index)로 묶이게 된다.

본 논문의 구성은 다음과 같다. 2장은 관련 연구로 기존 데이터베이스의 보안 모델에 대해 기술하고, 3장에서는 XML 제약 (constraint)에 의해 유추될 수 있는 데이터 추론에 대해 설명한다. 또한 4장에서는 본 논문에서 제안하는 노드 암호화 규칙들과 제안된 프레임워크에 대해서 자세하게 기술한다. 5장에서는 실험 평가를 통하여 제안된 방법의 성능 평가에 대해 논한다. 마지막으로 6장에서는 결론에 대해 기술하고 본 연구의 향후 연구에 대하여 기술한다.

## 2. 관련 연구

데이터베이스 보안에 관한 연구는 현재까지 광범위하게 연구되는 중이며 최근의 몇 가지 연구들은 데이터 출판에 관한 방법을 제안하였다[3, 6-8].

Miklau는 출판된 XML 문서에 접근 제어를 하기 위한 암호화 기반 접근 제어 방법을 연구하였다[8]. 이 방법은 접근 제어 정책의 명세화를 이용하는 방법으로 출판된 문서에 민감한 데이터를 정의하기 위해서 XQuery의 확장을 이용한 프레임워크를 설계하였다.

Yang은 의미적인 XML 제약 (constraint) 표현으로 공유 지식 방법을 연구하였다. 이 방법에서 사용자는 세 가지 타입의 XML 제약을 이용하여 데이터를 추론할 수 있고 정보의 누설 없이 주어진 단일 XML 문서의 부분적인 문서를 찾을 수 있는 새로운 알고리즘을 개발하였다[6]. 본 논문에서는 Yang의 연구와 다르게 문서 내의 암호화 된 부분을 사용자가 감지할 수 없고 허락된 사용자만이 접근할 수 있도록 프레임워크를 설계하였다.

Lee는 암호화 된 XML 데이터에 대하여 질의를 효율적으로 처리하기 위한 질의-인지 복호화 (Query-aware decryption) 개념을 제안하였다[7]. 본 논문은 Lee의 연구와 다르게 안전하고 빠른 구조적인 XML 보안 메커니즘을 제공한다.

## 3. XML 제약

### 3.1 XML 제약(Constraint)의 세 가지 종류

XML 제약은 문서 내의 노드들에 의해서 만족되는 관계를 명세화하고 DTD와 XML 스키마를 이용하여 정의할 수 있다[6].

XML 제약은 '조건 → 사실'이라는 형태로 표현할 수 있다. 만일 XML 문서에 '조건'이 만족된다면 그 때 '사실'은 반드시 XML 문서 내에 참이 된다. 세 가지 타입의 XML

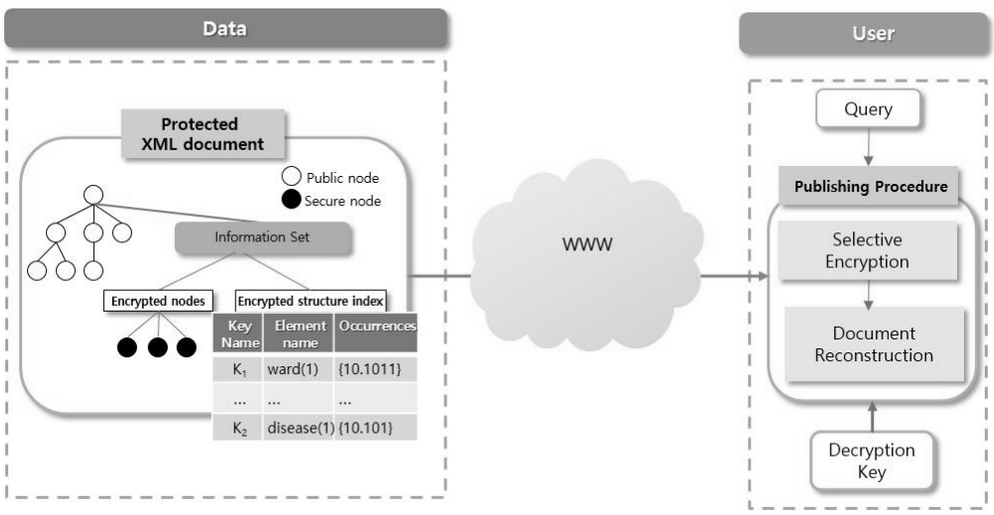



Fig. 2. Example of proposed secure framework

제약은 다음과 같다[6].

1) 자식 제약 : $P \rightarrow P/P'$로 표현되고 $P$의 모든 노드는 $P'$의 자식을 반드시 갖는다는 것을 의미한다.
2) 자손 제약 : $P \rightarrow P//P'$로 표현되고 $P$의 모든 노드는 $P'$의 자손을 반드시 갖는다는 것을 의미한다.
3) 함수적 종속성 제약 : $P/P_1 \rightarrow P/P_2$로 표현되고 여기서 $P$, $P_1$, $P_2$는 문서에 따르는 비어 있지 않은 부분 트리를 나타낸다. 어떤 두 개의 부분 트리 $T_1$과 $T_2$가 경로 $P/P_1$에 매칭 된다고 할 때, 만일 $P_1$이 경로 내에 동일한 값을 갖고 있다면 그 때 $P/P_2$에 매치되는 부분 트리는 동일하다는 것을 의미한다.

예를 들면, 자식 제약 '*//patient* → *//patient/ward*'는 엘리먼트를 가져야 함을 나타낸다. 자손 조건 '*//patient//leukemia*'는 각각의 *patient* 엘리먼트가 자손으로써 *leukemia* 엘리먼트를 반드시 가져야함을 나타낸다.

### 3.2 XML 제약에 의한 추론

이러한 XML 제약을 이용하여 사용자는 일반적인 질의의 결과로부터 정보를 추론할 수 있다. Fig. 1(a)에서 문서 $D$가 사용자에게 출판되었다면 사용자는 *patient*(1)의 *ward* 노드의 실제 위치를 알고 있고 *patient*(2)의 자식인 *ward* 노드의 위치를 추론할 수 있다. Fig. 1(c)의 Case 2에서 사용자들은 새로운 노드의 정확한 위치를 알 수는 없지만 *leukemia* 노드가 있다는 것을 인지할 수 있다.

## 4. 제안된 프레임워크

데이터의 출판에서 데이터의 소유자는 데이터를 출판한 뒤에 데이터에 대한 제어권을 잃게 된다[8]. Fig. 2는 본 논문에서 제안하는 안전한 데이터 출판을 위한 프레임워크를 나타낸다.

Fig. 2에서 각각의 사용자는 등록 단계 동안에 사용자 등록을 하고 데이터 소유자는 등록된 사용자들의 접근 권리에 따라서XML 문서에 주석 처리를 한다.

제안된 프레임워크에서 사용자는 문서 전체를 복호화 할 필요가 없고 미리 결정된 자신의 권한에 맞게 출판된 문서의 부분을 선택적으로 접근 할 수 있다.

제안된 프레임워크는 사용자의 권한에 따라 다음과 같은 구성요소로 동작한다.

- 권한 등록 : 엑세스할 수 있는 정보를 기록한 곳으로 사용자의 접근 권한은 XPath 표현식에 의해서 표현된다. 예를 들면, *physician*에 관련된 권한은 *//patient/name*, *//patient/ward*와 같이 표현할 수 있다.
- 권한 : 권한은 정보를 보여주는 것으로 view, navigate, 그리고 browse-all의 세 가지 권한을 제공한다.
- 키 할당 : 제안된 프레임워크에서 각각의 노드는 암호화 프로시저에 의해서 유일한 노드 키로 암호화된다. 노드 키를 사용자에게 보내기 위해서 노드 키들은 사용자 권한에 따라 그룹화 된다.

### 4.1 노드 암호화 규칙

제안된 프레임워크에서는 XML 제약에 의해 유추될 수 있는 정보를 보호하기 위해서 세 가지의 노드 암호화 규칙에 의해서 공개 노드와 민감한 노드를 구별한다.

1) 자식 암호화 규칙 : 만일 노드가 자식 제약 ($P \rightarrow P/P'$)을 만족한다면 그 노드의 모든 자식을 민감한 노드로 선택하여 반드시 암호화한다.
2) 자손 암호화 규칙 : 만일 $Q$와 $Q'$이 자손 제약 ($P \rightarrow P//P'$)을 만족한다면 그때 $Q$와 $Q'$를 민감한 노드로 선택하여 반드시 암호화한다.
3) 함수 종속성 암호화 규칙 : 만일 $Q$, $Q_1$, $Q_2$가 함수 종속성 제약 ($P/P_1 \rightarrow P/P_2$)을 만족한다면 그 때, $Q$, $Q_1$, $Q_2$를 민감한 노드로 선택하여 반드시 암호화한다.

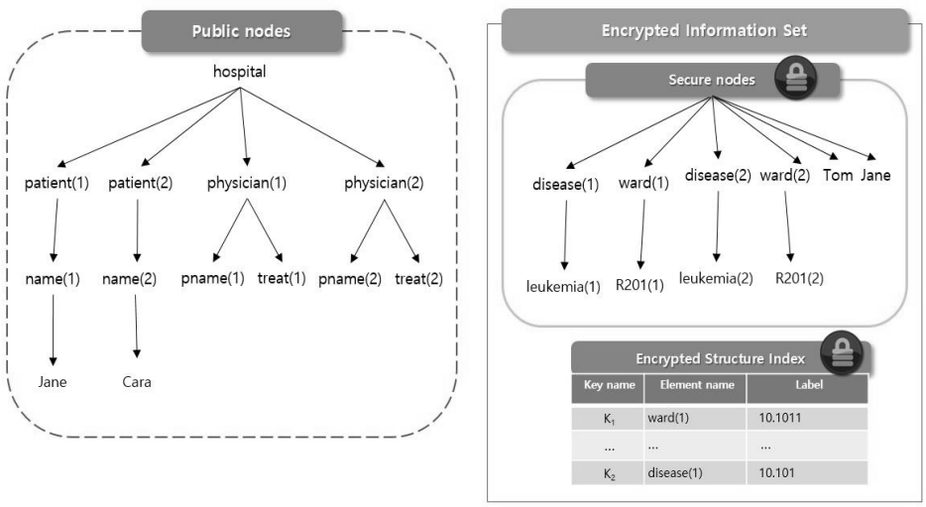



Fig. 3. Example of protected XML document with encrypted information set

예를 들면, 공개 노드와 민감한 노드를 구분하여 자식 제약을 갖는 *//patient* → *//patient/ward*에서 각각의 *patient* 노드가 반드시 *ward* 노드를 갖고 있다면 Fig. 3에서 관련된 노드 *ward*(1)과 *ward*(2)를 반드시 암호화한다.

만일 *//physician/pname* → *//physician/treat*에서 동일한 *department*의 *physician*은 동일한 *disease*를 치료하기 때문에 관련된 노드들 *ward*(1), *disease*(1), *ward*(2), *disease*(2), *Tom*, *Mary*, *Jane* 그리고 *Steve*를 반드시 암호화한다.

Table 1. The encrypted structure index for Fig. 3

| Key name | Element name | Label |
|---|---|---|
| K1 | *hospital* | *null* |
| K1 | *patient(1)* | *10* |
| K1 | *name(1)* | *10.10* |
| K1 | *Jane* | *10.10.10* |
| K3 | *disease(1)* | *10.101* |
| K3 | *leukemia(1)* | *10.101.10* |
| K3 | *ward(1)* | *10.1011* |
| K3 | *R201(1)* | *10.1011.10* |
| K1 | *physician(1)* | *1110* |
| K1 | *pname(1)* | *1110.10* |
| K2 | *Tom* | *1110.10.10* |
| K1 | *treat(1)* | *1110.110.10* |
| K3 | *Jane* | *1110.110.10* |
| | ... | ... |

### 4.2 암호화 정보 셋

제안된 프레임워크에서 시스템은 XML 문서의 민감한 부분을 암호화한다. XML 문서의 민감한 부분을 암호화하기 위해서 원래의 문서에서 노드들이 선택되고 권한이 부여되지 않은 사용자들로부터 원래의 문서 구조를 숨길 수 있다.

민감한 노드들은 노드 암호화 규칙에 따라서 선택되고 Fig. 3과 같이 암호화된 정보 셋으로 이동된다. 암호화된 정보 셋은 암호화된 민감한 노드들과 구조 인덱스로 구성된다. 복호화 후에 각각의 민감한 노드는 원래의 문서 내의 정확한 위치를 찾아가기 위해서 암호화된 구조 인덱스를 이용한다.

암호화된 정보 셋으로 이동된 민감한 노드들이 복호화 시 원래의 위치를 찾기 위해서 본 논문에서는 IBSL 레이블링 방법 [9]를 이용한다. 제거 단계에서 민감한 노드들은 XML 문서에서 분리되어 이동되고 공개 노드들은 레이블 정보 없는 문서로 출판된다. 만일 공개 노드들이 레이블 정보를 갖고 있다면 사용자는 레이블 정보를 통해서 암호화된 노드의 위치를 추측할 수 있고 공개 노드들의 구조적 정보를 유추할 수 있다.

### 4.3 XML 문서의 질의 처리

암호화된 구조 인덱스를 이용하여 원래 XML 문서의 구조적 정보를 포함하여 복호화 되는 민감한 노드들의 원래의 위치를 확인할 수 있다.

Table 1은 Fig. 3의 문서를 이용하여 암호화된 구조 인덱스의 예를 나타낸다. 암호화된 구조 인덱스 내의 각각의 엔트리는 키 이름, 엘리먼트 이름, 레이블로 구성된다. 예를 들면, *disease*(1)는 *10.101*로 레이블링되고 키 *K1*을 이용하여 암호화된다. *Tom*은 *1110.10.10*으로 레이블링 되고 키 *K2*를 이용하여 암호화된다. 사용자 질의에 따라서 복호화되는 민감한 노드들의 문서내의 원래의 위치를 찾기 위해서 입력값으로 공개 문서, 암호화된 정보 셋, 질의, 키를 이용한다. 본 논문에서는 Algorithm 1을 이용하여 복호화 된 노드의 자식 노드들을 재배치 할 수 있다.

Algorithm 1에서 공개 문서의 노드는 레이블이 되지 않고 암호화된 구조 인덱스의 정보를 이용하여 복호화 된 노드들의 구조 정보를 알 수 있다. 어떤 노드가 복호화 되는 노드의 부모 노드인지를 먼저 확인하고 만일 부모 노드가 이미 자식 노드를 갖고 있다면 복호화 되는 노드가 몇 번째

**Algorithm 1.** Query Processing

**Input**: 공개 문서, 암호화된 정보 셋, 질의, 사용자 키
**Output**: 질의 결과

**begin**
01: 암호화된 인덱스에 대하여 질의를 처리한다.
02: 사용자 키를 이용하여 암호화된 노드들과 암호화된 구조 인덱스를 복호화 한다.
03: 복호화 된 XML 문서에 대하여 질의를 만족하는 엘리먼트들을 찾는다.
04: 공개 문서 내에 찾은 엘리먼트들을 재배치한다.
05: 암호화된 구조 인덱스에 저장된 레이블들의 관계를 파악하여 엘리먼트들의 위치를 결정한다.
**end**

자식인지를 이미 존재하는 형제 노드와 비교하여 원래의 위치를 찾을 수 있다. 만일 부모 노드가 자식을 갖고 있지 않다면 복호화 된 노드는 부모 노드에 직접적으로 바로 추가된다.

**예제 4.1** 사용자가 키 *K1*, *K2*, *K3*을 가지고 있다고 가정하자. Table 1의 암호화된 구조 인덱스를 이용하여 Fig. 3의 암호화된 정보 셋에 대하여 다음의 질의 Q: '*//patient*(1)/*disease*(1)'을 처리하려고 한다. 질의 처리는 첫 번째로 *K1*, *K2*, *K3*을 이용하여 암호화된 구조 인덱스와 암호화된 노드들을 복호화 한다. 다음으로 키 *K1*과 *K3*을 이용하여 엘리먼트 *patient*(1), *disease*(1)의 구조 인덱스를 복호화 한다. 노드 *disease*(1)은 레이블 *10.101*을 갖고 레이블 관계에 의해서 *disease*(1)의 부모는 레이블 *10*을 갖는 *patient*(1)가 된다.

## 5. 성능 평가

본 연구에서는 성능 평가를 위해서 제안된 방법과 Miklau[8] 방법을 비교하여 실험 평가하였다. 실험에서 데이터 소유자는 XML 문서를 암호화하고 사용자에게 공개 문서와 암호화된 XML 문서로 출판하였다.

제안된 기법과 Miklau 기법[8]은 XML security Suite[10]과 128-비트 키의 Advanced Encryption Standard (AES)[11]을 Java로 구현하였다. 실험은 Windows 7에서 실행되었으며 4GB의 RAM, 3.40GHz 펜티엄 프로세서에서 수행되었다.

또한 실험에서 사용되는 XML 문서를 생성하기 위해 XMark[12] 데이터 셋을 사용하였다.

본 논문에서는 암호화 할 노드를 선택하기 위해서 XPath [13] 표현식을 사용하였다. 또한 신뢰 구간을 얻기 위해 동일한 실험을 20회 반복하여 실시하였다. Table 2는 암호화 할 노드를 나타내는데 사용된 XPath 표현식을 나타낸다. 실험 평가에서는 암호화된 구조 인덱스의 크기와 노드들의 개수와의 상관관계를 파악하기 위해서 비교된 노드들의 수를 살펴보았다.

Fig. 4는 2MB의 XMark 데이터로부터 생성된 암호화된 XML 문서와 암호화된 구조 인덱스의 크기를 나타낸다.

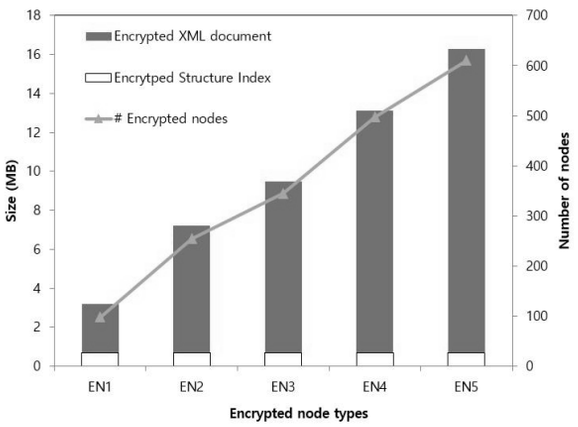



Fig. 4. The size of encrypted structure index

실험에서 구조 인덱스가 암호화 되었을 때 키 이름, 엘리먼트 이름, 레이블은 암호화된 스트링으로 재배치되었다. 실험 결과에서 볼 수 있듯이 암호화된 구조 인덱스의 크기는 암호화된 XML 문서와 비교하여 암호화, 복호화, 출판 시에 발생하는 오버헤드의 크기는 상대적으로 크지 않음을 알 수 있었다.

결과에서 보듯이 암호화된 구조 인덱스의 크기는 암호화된 XML 문서의 크기의 *1/22* 정도의 작은 크기를 나타내었다. 암호화된 XML 문서의 크기는 암호화 되는 노드들의 수가 증가할수록 커졌지만 암호화된 구조 인덱스의 크기는 커지지 않았다.

Table 2. Encrypted nodes ($EN_n$)

| Encrypted node | XPath expression |
|---|---|
| EN1 | //item/* |
| EN2 | //item/description/parlist[listitem/text] |
| EN3 | //item//parlist[.//mailbox]/ |
| EN4 | //*/text/keyword |
| EN5 | /*//keyword |

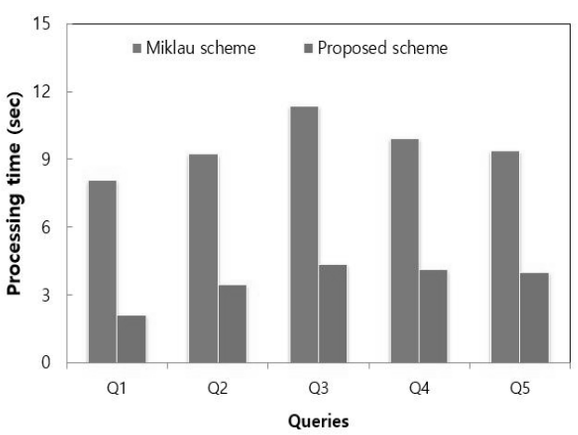



Fig. 5. Query processing time

실험 평가에서는 보호된 문서의 질의 처리 시간을 측정하기 위해서 질의의 종류에 따라서 암호화된 노드들의 복호화되는 시간을 측정하였다. Table 2에 나타난 XPath 표현식을 이용하여 암호화된 노드를 질의로 이용하였다.

제안된 방법과 Miklau 기법[8]의 질의 처리 시간을 비교하기 위해서 XML 문서 내의 노드들의 위치를 찾기 위한 처리 시간을 측정하였다. Fig. 5에서 제안된 기법은 XMark 데이터로 생성된 2KB의 XML 문서에서 모든 질의들에 대해 질의 처리 시간이 효율적으로 빠른 것을 볼 수 있다.

실험 결과 제안된 기법은 복호화 된 노드들의 위치를 비교할 때 Miklau 기법보다 3배 정도 효율적으로 노드를 검색할 수 있다는 것을 알 수 있었다.

## 6. 결 론

본 논문에서는 출판되는 민감한 정보를 보호하기 위한 프레임워크를 제안하였다. 제안된 프레임워크는 사용자의 추론에 의해 정보의 유추 없이 XML 문서가 출판될 수 있도록 하기 위해서 암호화된 정보 셋과 암호화된 구조 인덱스를 이용하였다.

안전하고 효율적인 출판을 위해서 암호화된 구조 인덱스는 원래의 XML 문서의 구조적 정보를 요약하여 저장하였고 질의 결과 내에 포함되는 암호화된 엘리먼트들은 암호화된 구조 인덱스를 이용하여 저장하였다.

실험 결과 암호화된 구조 인덱스의 사이즈는 암호화되는 XML 문서의 사이즈보다 현저하게 작았고 기존의 방법보다 사용자 질의 처리 시간을 효율적으로 줄일 수 있었다.

향후 연구로는 멀티 레벨 접근 제어를 위한 안전한 그룹 키 기반 관리 방법의 연구가 계속되어야 하겠다.

## References


[1] Z. G. Ives, A. Y. Lyer, J. Madhavan, R. Pottinger S. Saroiu, I. Tatarinov, S. betzler, Q. Chen, E. Jaslikowska, J. Su, and W. Yeung, “Self-Organizing Data Sharing Communities with SAGRES,” in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, p.582, 2003.

[2] W. S. Ng, B. C. Ooi, K. L. Tan, and A. Zhou, “A P2P-based System for Distributed Data Sharing,” in *Proceedings. of 19th International Conference on Data Management*, pp.633-644, 2003.

[3] M. Stonebraker, P. M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, Staelin, and A. Yu, “Mariposa: A Wide-area Distributed Database System,” *VLDB Journal*, Vol.5, No.1, pp.48-63, 1996.

[4] Extensible Markup Language (XML) 1.0 [Internet], http://www.w3.org/TR/REC-xml.

[5] E. Bertino, S. Castano, E. Ferrari, and M. Mesiti, “Controlling Access and Dissemination of XML Document,” in *Proceedings of 2nd International Workshop on Web Information and Data Management*, pp.22-27, 1999.

[6] X. Yang and C. Li, “Secure XML Publishing without Information Leakage in the Presence of Data Inference,” in *Proceedings of the 30th International Conference on Very Large Data Bases*, pp.96-107, 2004.

[7] J. G. Lee and K. Y. Whang, “Secure Query Processing against Encrypted XML Data Using Query-Aware Decryption,” *Information Science*, Vol.176, No.13, pp.1928-1947, 2006.

[8] G. Miklau and D. Suciu, “Controlling Access to Published Data Using Cryptography,” in *Proceedings. of the 29th International Conference on Very Large Data Bases*, pp.898-909, 2003.

[9] H.-K. Ko and S. Lee, “A Binary String Approach for Updates in Dynamic Ordered XML Data,” *IEEE Transactions on Knowledge and Data Engineering*, Vol.22, No.4, pp.602-607, 2010.

[10] IBM XML Security Suite [Internet], http://www.alphaworks.ibm.com/xmlsecuritysuite.

[11] J. Dacmen and V. Rijmen, “The Block Cipher Rijndael,” in *Proceedings of the International Conference on Smart Card Research and Applications*, pp.277-284, 1998.

[12] A. Schmidt, F. Wass. M. L. Kersten, M. J. Carey, L. Manolescu, and R. Busse, “A Benchmark for XML Data Management,” in *Proceedings. of 28th International Conference on Very Large Data Bases*, pp.974-985, 2002.

[13] XML Path Language (XPath) [Internet], http://www.w3.org/TR/1999/REC-xpath-19991116.



고 혜 경

e-mail : hkko@sungkyul.ac.kr

2008년 고려대학교 컴퓨터학과(이학박사)

2013년~현 재 성결대학교 컴퓨터공학부 조교수

관심분야 : 빅데이터 관리, 정보보호, 온톨로지 데이터 관리, 데이터 마이닝