다중 보안레벨 사용자 간에 교환 가능한 XML 문서 보안에 대한 연구

김수희, 김문권, 권태경
세종대학교 정보보호연구실

Securing Exchangeable XML Documrnts for Users

Suhee Kim, Moonkwon Kim and Taekyoung Kwon
Information Security Lab., Sejong University.

요 약

XML(eXtensible Markup Language)은 현재 웹에서 데이터 처리의 표준으로 자리 잡고 있다. 하지만 XML-Encryption을 통해 해결하기 어려운 문서 기밀성과 무결성에 대한 관리적인 문제가 발생한다. 본 논문에서는 시스템 내부 사용자가 한 번의 데이터 요청을 통해 여러 외부 사용자의 접근권한에 맞게 신뢰 가능한 데이터를 전달하며, 모든 사용자는 자신이 가지고 있는 그룹키를 통해 일회성인 레벨키를 생성하여 자신의 접근권한에 맞는 데이터를 볼 수 있는 XML 문서 보안 방법을 제안한다.

I. 서론

XML은 W3C(World-Wide Web Consor -tium)로부터 정의된 데이터의 구조를 기술하는데 사용되는 마크업 언어이다[1]. XML이 웹의 표준 문서 교환 방식으로 자리 잡게 됨에 따라 XML문서 보안에 관한 관심이 점점 높아지고 있다. 이에 W3C 에서는 XML 보안의 표준으로써 XML-Encryption을 제안하였다. 하지만 단순히 하나의 XML 문서를 여러 사용자의 권한에 따라 접근제어하기에는 어려움이 있다. 기존의 접근제어를 이용한 방법들은 DB나 DTD레벨에서의 방식들이다. 따라서 하나의 XML문서를 여러 사용자가 공유하는 방법에 대해서는 고려하지 않았다. 본 논문은 효과적이고 안전한 XML 문서 전달을 위해, 키 전달 과정 없이, 사용자의 권한에 따라 암호화된 XML 문서에 접근할 수 있는 방법을 제안한다. 또한 전자서명을 통한 데이터 무결성 검증 방법과 자신의 그룹키를 통해 유도한 레벨키를 사용하는 암·복호화 방법을 제시한다.

II. 관련연구

2.1 XML-Encryption

XML-Encryption은 2002년 12월에 W3C가 표준완료된 것으로 XML문서 내에서 기밀성을 보장하기 위해 만들어 졌다. XML문서 내용의 중요한 부분을 암호화 하여 문서전체를 암호화 하지 않아도 되도록 만들어 암·복호화에 다른 계산량을 줄이고 효율성을 높였다[2].

2.2 XML Signature

XML-signature는 XML문서에 대해 XML 형태의 서명을 생성하고 검증할 수 있는 전자서명 표준으로 W3C와 IETF가 공동으로 개발하였으며 전자문서에 대해 인증, 데이터 무결성, 부인방지 등의 기능을 제공한다. 이것의 기본적인 특징은 문서 전체뿐만 아니라 XML문서의 특정부분을 서명하는 것으로 특정 부분에 대한 무결성을 확인하기 위한 것이다[3].

2.3 XML문서의 접근제어

XML문서를 접근제어 하는 방법에는 크게 DTD기반의 접근제어 방식과, XML-Encryp- tion을 이용하는 방법 두 가지로 나눌 수 있다. 각 방식에는 장단점이 있는데, DTD기반의 방식[4]은 XML 문서를 읽을 때 접근제어를 통해 가져오기 때문에 XML 문서 자체의 접근제어는 불가능하다. 그에 반해 XML-Encryption을 이용한 방식에는 계산에 따른 부하가 있다는 단점이 있지만 XML 문서의 데이터를 안전하게 보관하고 전달 할 수 있다는 장점이 있다.

XML-Encryption을 이용한 방식에는 또다시 DB기반의 방식과 문서자체에 접근제어를 하는 방식으로 나눌 수 있다. DB기반 방식[5],[6]은 DB로부터 데이터를 가져올 때 접근제어를 통해 가져오기 때문에 XML 문서 자체의 접근제어는 하지 못한다. 그래서 문서 자체에 접근제어를 하는 방식인 슈퍼암호화와 다중암호화에 대해 알아보고 새로운 체인암호화 방식을 제안한다.

2.3.1 Super 암호화

이것은 많은 사용자가 하나의 문서를 접근할 때

* 본 연구는 보건복지부 보건의료기술진흥사업의 지원에 의하여 이루어진 것임. (0412-MI01-0416-0002)

접근권한 규칙에 따라 제안된다. 이것은 중요한 정보를 중요도에 따라 여러번 중첩해서 암호화 하는 방법으로 방식은 매우 간단한다. 하지만 상위 사용자는 모든 하위레벨 키들을 가지고 있어야 하고 암호화와 복호화 시 많은 계산량을 가지므로 큰 메모리 공간이 필요하다. 이러한 여러 가지 이유로 실제 적용하기에는 많은 무리가 따르는 방법이다[7].

2.3.2 다중 암호화

super 암호화와 가장 큰 차이점이라면 중요도에 따라 여러번 암호화 하는 것이 아닌 중요도에 따라 각각의 레벨에 따른 키로 암호화 한다는 점이다. 이것은 super encryption의 단점중 하나인 계산량은 현저하게 줄여 주지만 여전히 상위 사용자는 하위레벨의 모든 키들을 가지고 관리 하여야 한다는 단점은 여전히 존재 한다[7].

III. 신뢰 가능한 접근권한 기반의 XML 문서 전달

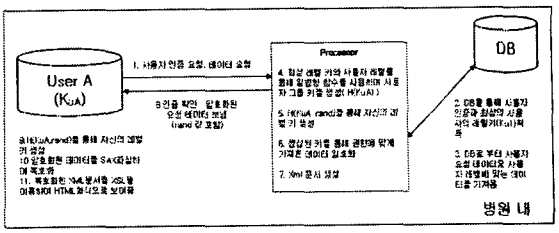
본 논문은 시스템 내부 사용자에 의해 역할기반 접근제어 정책을 통해 암호화된 하나의 XML 문서를 외부 여러 사용자에게 신뢰성 있는 전달을 수행하는 것이다. 제안하는 기법은 우선 다음과 같은 환경을 가정한다.

●사용자 그룹의 레벨에 따라 볼 수 있는 문서의 기밀 정도가 다르다.
●레벨은 최상위부터 최하위까지 한 방향으로 이루어져 있다.
●모든 사용자는 자신의 그룹 키를 스마트카드로 만들어진 ID 카드를 통해 가지고 있다.
●시스템 외부 사용자에 의한 데이터 요청이 빈번이 발생한다.

3.1 구조

접근제어는 인증을 거친 사용자가 원하는 정보를 요청하면 시스템은 사용자의 권한을 확인하고 요청한 자료를 권한에 맞는 부분만 전달하기 위한 것이다.

우선 문서를 암·복호화하기 위해서는 접근 권한 규칙에 따른 각 레벨별 키가 필요하다. 사용자가 자신의 권한에 맞는 데이터를 갖기 위해서는 자신 이하의 모든 레벨 정보를 볼 수 있게 각 레벨의 키를 가지고 있어야 한다. 이러한 경우 한 사용자가 여러 개의 키를 가져야 하기 때문에 키 관리의 어려움이 발생한다. 이를 해결하기 위해 여기서 그룹키는 일방향 키 체인[8]을 이용하여 자신 이하의 모든 그룹키를 유도할 수 있다. 해쉬 함수는 역함수가 존재하지 않는 것으로 인코딩은 가능하지만 디코딩은 불가능한 함수이다. 이 성질을 이용하여 최상위 그룹키를 일방향 키 체인을 통해 모든 레벨의 그룹키를 생성할 수 있다. 또한 각 그룹키와 랜덤값을 해쉬 함수를 통해 각 레벨키를 생성하여 데이터를 암·복호화 할 수 있다. 랜덤값은 데이터 요청 시 서버에서 만들어 데이터에 부착시켜 전달한다. 이 특징을 이용하여 모든 사용자는 단지 자신의 그룹키 하나만을 가지고 자신의 레벨이하의 모든 문서를 암·복호화 할 수 있다.



[그림 1] 자신만이 사용할 문서 요청

제안하는 기법은 시스템 서버에 데이터를 요청하는 사용자가 다른 사용자에게 문서 전달할 것인지의 여부에 따라 두 가지 방법으로 수행될 수 있다. 정보 요청 사용자가 자신만 보기위한 문서를 요청할 경우에는 이 사용자의 권한에 해당하는 데이터만 이 사용자의 레벨키로 암·복호화 하는 단순한 방법을 사용한다. 하지만 정보 요청 사용자가 다른 사용자에게도 전달할 문서를 요청하는 경우에는 각 레벨키 생성 방법이 요구되지만, 모든 사용자에게 서버로부터 전달받은 하나의 XML문서 전달을 통해 사용자는 자신의 접근권한 정보를 모두 볼 수 있을 뿐 만 아니라 전자서명을 통한 데이터의 무결성 보장하며 키를 전달할 필요가 없다. 다음은 이 두 가지 구조를 구분하여 자세히 설명한다.

3.1.1 정보 요청 사용자만 사용할 문서를 요청하는 경우

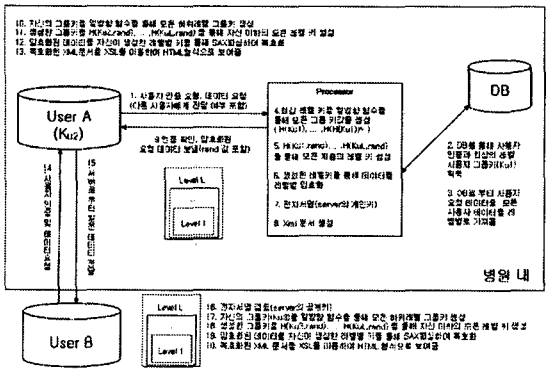
첫 번째로 정보 요청 사용자가 단지 자신만이 볼 데이터를 서버에 요청할 경우이다.

사용자가 시스템에 사용자 인증 및 데이터 요청을 하면 서버는 데이터베이스에서 해당 사용자와 요청 정보가 존재하는지 확인을 한다. 사용자 인증을 한 후 요청 정보 중 사용자의 권한 정책에 맞는 데이터와 사용자의 레벨정보, 최상위 그룹키를 데이터베이스로부터 가져온다. 최상위 그룹키를 사용하여 일방향 키 체인을 통해 사용자의 그룹키를 생성한다. 이 키와 생성한 랜덤값을 해쉬 함수를 사용하여 새로이 사용자의 레벨키를 만들어 사용자의 권한 정책에 맞는 데이터를 암호화한다. 그리고 암호화한 XML 문서에 랜덤값을 함께 전달한다. 사용자는 자신이 가지고 있는 그룹키를 문서와 함께 전달되어 온 랜덤키와 함께 해쉬 함수를 사용하여 자신의 레벨키를 생성한 후 XML 문서를 복호화한다.

3.1.2 정보 요청 사용자가 다른 사용자에게도 전달할 문서를 요청하는 경우

두 번째로 서버에 정보를 요청하는 사용자가 자신이 획득한 정보를 자신뿐만 아니라 여러 레벨의 다른 사용자에게 전달할 수 있는 데이터를 서버에 요청하는 경우이다.

이전과 마찬가지로 정보 요청 사용자가 시스템에 사용자 인증 및 데이터 요청을 하면 서버는 데이터베이스에서 해당 사용자와 요청 정보가 존재하는지 확인을 한다. 하지만 사용자 인증을 한 후 데이터베이스로부터 요청 정보 중 사용자의 접근권한 정책에 해당하는 정보만 가져오는 것이 아니라 어떤 레벨의 사용자라도 볼 수 있는 요청 정보의 모든 데이터를 최상위 그룹키와 함께 가져온다. 최상위 그룹키를 일



[그림 2] 다른 사용자와 함께 사용할 문서 요청

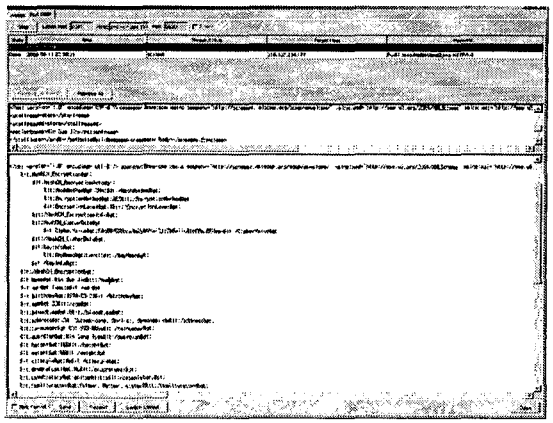
방향 키 체인을 사용하여 모든 사용자의 그룹키를 생성한다. 각각의 그룹키와 랜덤값을 해쉬 함수를 통해 일회성인 모든 레벨키를 생성하여 데이터의 레벨에 따라 알맞은 레벨키로 부분별 암호화한다. 그리고 서버의 개인키를 통해 전자서명을 한 후 암호화한 XML 문서에 랜덤값과 서버의 공개키를 함께 전달한다. 서버에 정보를 요청한 사용자는 자신이 가지고 있는 그룹키를 일방향 키 체인을 통해 자신 이하의 모든 레벨 그룹키를 생성한다. 그리고 문서와 함께 전달되어 온 랜덤키와 함께 해쉬 함수를 사용하여 자신을 포함한 모든 하위레벨키를 생성한 후, 자신의 접근권한에 해당하는 문서를 볼 수 있다. 또한 다른 사용자가 서버로부터 획득한 문서를 가진 사용자에게 이 문서를 요청하면 인증 후 획득한 데이터를 그대로 전달한다. 서버로부터 간접적으로 데이터를 획득한 사용자는 데이터를 받으면 우선 서버의 공개키를 통해 전자 서명을 검증한 후 이전에 서버로부터 직접 데이터를 획득한 사용자가 했던 방법을 사용하여 얻은 문서 중 자신의 권한에 맞는 데이터만을 복호화하여 볼 수 있다.

IV. 환경설정 및 구현

본 연구에서 제안하는 접근권한 기반의 신뢰 가능한 XML문서 전달은 다음과 같은 컴퓨팅 환경에서 구현되었다.

- ●Computer environment
  CPU: Intel 3.2G RAM: 1G OS: Microsoft windows server 2003 enterprise edition
- ●Language : Java
- ●developing environment : eclipse
- ●Web Server : Apache tomcat 5.x, apache-axis web service server

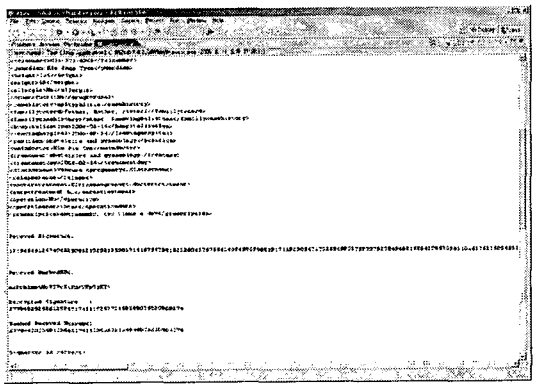
본 논문은 가상의 병원 시스템에서 사용자가 환자에 대한 정보를 요청 때 다른 사용자에게도 전달할 때 접근권한에 따라 데이터를 전달하는 경우를 가정하여 구현하였다. 우선 원장, 과장의사, 담당의사, 수련의, 간호사의 5단계로 문서의 중요도에 따라 접근권한을 다르게 주어서 최고 level 1부터 최하 level 5까지로 나타내어 문서의 접근을 제한한다. 그리고 DB에 여러 환자들의 데이터가 저장되어있다. 여기서는 담당의사가 민수진이라는 환자의 정보를 가져와 간호사에게 전달하는 과정을 보여준다.



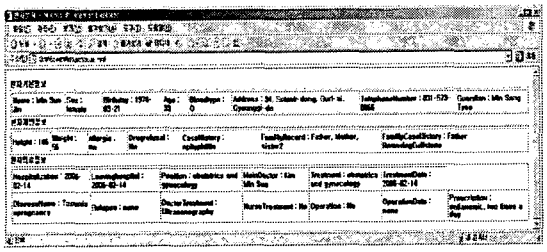
[그림 3] TCPMON을 통해 본 SOAP메세지

[그림 4] 서버에서 인증 후 데이터 처리 과정

담당의사가 웹서비스 서버에 java로 작성된 클라이언트 프로그램을 통해 자신의 아이디와 패스워드를 SOAP메세지 형태로 보낸다. 서버는 사용자의 인증 요청 시 사용자의 아이디와 패스워드가 일치하는지를 DB 검색을 통하여 확인 하여 인증을 하는데 이는 [그림 3] TCPMON을 통해 SOAP 메세지를 보고 확인할 수 있다. 인증된 사용자는 민수진이라는 환자 정보를 다른 사용자들에게도 전달할 것이라는 것과 함께 서버에 이 환자정보를 요청하며 이러한 데이터 처리 과정은 [그림 4]와 같다. 서버는 이 환자에 대한 모든 정보와 병원에서 최상위 레벨인 level 1 그



[그림 5] 클라이언트에서 문서를 받고 검증하는 과정



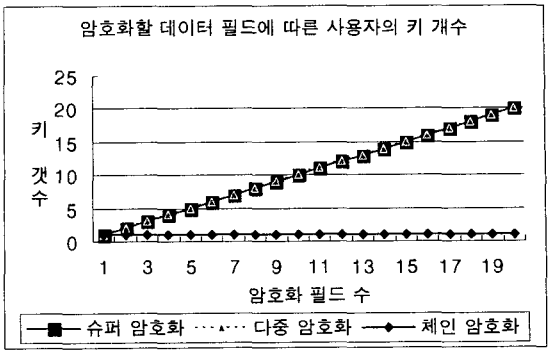
[그림 6] 웹브라우저를 통해 보여지는 복호화된 XML문서

룹키를 DB로부터 획득한다. level 1 그룹키를 일방향 키 체인을 이용하여 level 5까지의 그룹키를 생성한다. 랜덤값을 하나를 생성하여 각 레벨의 그룹키와 함께 해쉬하여 새로운 각각의 그룹 레벨키를 생성하여 접근권한에 따라 XML문서를 암호화하며, 서버의 개인키를 통해 전자서명한 후 랜덤값과 함께 전달한다. 담당의사는 자신이 가진 level 3 그룹키를 일방향 함수를 사용하여 level 4, level 5 그룹키를 생성한 후 문서에 부착된 랜덤값과 획득한 모든 그룹키를 해쉬하여 level 3, level 4, level 5의 레벨키를 생성하여 자신의 접근권한에 맞는 데이터까지 복호화할 수 있다. 이때 자신보다 상위 레벨의 키로 암호화 되어 있는 부분은 볼 수 없다. 다음으로 담담의사가 간호사에게 민수진이라는 환자정보를 보내고자 할 때 의사와 간호사가 볼 수 있는 부분의 정보는 다르지만 서버로부터 획득한 데이터를 그대로 전달하여도 간호사는 자신의 접근권한에 맞는 부분만을 볼 수 있다. 우선 간호사는 서버의 공개키를 통해 전자서명을 검증한 후 자신이 가지고 있는 본인의 그룹키를 통해 이전에 담당의사가 한 것과 같은 방법으로 자신의 접근권한에 해당하는 정보까지 복호화 할 수 있다. 복호화 된 문서는 XSL스타일 시트를 이용하여 [그림 6]과 같이 사용자에게 HTML 형태로 보여준다.

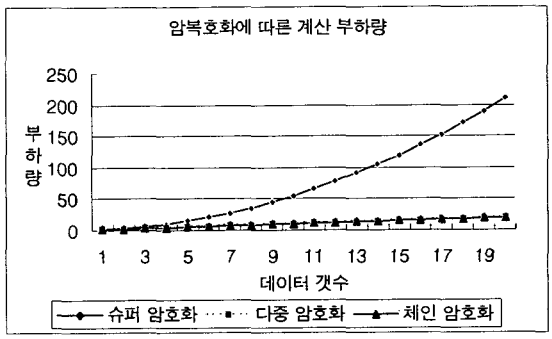
V. 성능 비교 및 분석

본 논문에서 사용하는 체인 암호화의 효율성을 XML-Encryption 기반의 접근제어 방식들과 함께 그래프 형태로 비교 및 분석 해 보았다.

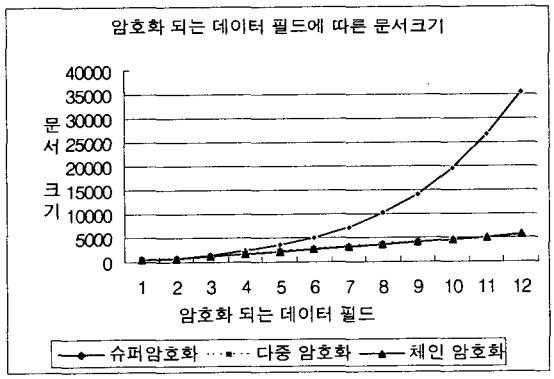
[그림 7]에서는 암호화 하는 데이터에 따른 사용자가 가져야할 키 개수를 비교 하고 있다. 기존의 방법들은 데이터가 많아질수록 데이터에 따른 키가 필요하



[그림 7] 암호화할 데이터 필드에 따른 사용자의 키 개수



[그림 8] 암복호화에 따른 계산 부하량



[그림 9] 암호화 되는 데이터 필드에 따른 문서 크기

기 때문에 사용자가 가질 키 개수가 선형적으로 증가하지만 체인 암호화는 모든 사용자가 단지 자신의 그룹키 하나만을 가짐으로서 일정하게 유지 된다.

[그림 8]에서는 암·복호화에 따른 계산량을 표로 나타낸 것이다. 슈퍼암호화의 경우 레벨이 높은 데이터일수록 총 데이터 레벨의 횟수만큼 반복적으로 암호화되기 때문에 암·복호화 계산량이 크다. 다중 암호화와 체인 암호화에서 각 데이터는 자신의 레벨에 따른 키로 한 번씩만 암·복호화 함으로써 계산량은 현저히 적다.

[그림 9]에서는 암호화 되는 데이터 필드에 따른 문서의 크기를 비교 한 것으로 슈퍼암호화는 문서의 크기가 암호화 횟수에 따라 지수적으로 증가하지만

다중 암호화와 체인 암호화는 선형적으로 증가한다. 그래프에 잘 나타나지 않았지만 미세하게 체인암호화가 다중 암호화 보다는 문서의 크기가 조금 더 크다. 하지만 앞에서 비교하였듯이 다중 암호화는 사용자가 가져야할 키 개수가 많으므로 본 논문에서 사용한 체인 암호화가 보다 효율적인 것을 볼 수 있다.

VI. 결론 및 향후연구 방향

본 논문은 문서 전달 시 정보의 기밀성에 따른 접근권한 문제를 키 전달 없이 효과적으로 해결하기 위한 방법을 제안하였다. 모든 사용자는 자신의 그룹키를 ID카드에 저장하여 가지고 있다. 하지만 실제로 데이터 암·복호화에 사용되는 키은 문서와 함께 전달되어오는 랜덤값을 통하여 한 문서에서만 쓸 수 있는 일회성 키를 사용한다. 따라서 서버로부터 다른 문서를 전달받을 경우 다른 키를 생성하여 사용하는 결과를 가져온다. 또한 효과적인 데이터 전달 방법을 위해 다른 사람들에게 데이터를 전달할지의 여부에 따라 두 가지 방식으로 데이터를 가져온다. 첫 번째 경우는 간단한 방법으로 암·복호화가 쉽게 이루어지며, 두 번째 경우는 키를 생성하는 복잡한 과정이 있긴 하지만 암호화된 문서 하나만을 다른 여러 사람들에게 키 없이 전달하여 권한에 따른 접근제어를 할 수 있다. 그리고 전자서명을 통해 데이터가 전달되었을 때의 손상 여부를 알 수 있으므로 안전하다. 본 연구는 사용자의 그룹 레벨에 따라 볼 수 있는 문서의 기밀 정도가 다름을 가정하여 구현하였다. 하지만 모든 문서가 항상 한방향의 레벨에 따라 기밀 정도가 달라지는 경우만 있는 것은 아니며, 레벨 뿐만 아니라 정책 또한 함께 사용해야 하는 문서가 발생할 경우에는 데이터 암호화 시 제안하는 스킴을 통해 해결하기 어려움으로 이에 대해 고려해 보아야 한다.

[참고문헌]

[1] D. Hunter, K. Cagle, D. Gibbons, N. Ozu, J. Pinnock and P. Spencer,. "beginning XML", 25 October 2000

[2] T. Imamura, B. Dillaway and E.Simon, "XML Encryption Syntax and Processing", W3C Recommendation 10 December 2002, http://www.w3.org/TR/xmlenc-core/

[3] D. Eastlake, J. Reagle and D. Solo, "XML-Signature Syntax and Processing", W3C Recommendation 12 February 2002, http://www.w3.org/TR/xmldsig-core/

[4] E. Damiani, S. Vimercati, S. Paraboschi and P. Samarati, "Securing XML Documents" EDBT2000, March 27-31, 2000.

[5] C. Geuer-Pollmann, "XML Pool Encryption," In Workshop On Xml Security Proceedings of the 2002 ACM workshop on XML security, Nov.22, 2002.

[6] 최동희, 박석, "접근제어 정책 구현을 위한 주체 기반 XML 암호화", 데이터베이스연구회, 2004.

[7] E.Damiani, S.Vimercati, S.Paraboschi and P.Samarati, "Securing XML Documents", EDBT2000, 2000..

[8] Y. Hu, M. Jakobsson and A. Perrig, "Efficient Constructions for Onw-Way Hash Chains," In proceedings of ACNS 2005, LNCS 3531, pp. 423-441, 2005.