모바일 환경에서의 디지털 서명을 위한 XML 정규화 시스템

우뢰 · 홍현우 · 윤화묵 · 최봉규 · 정회경

배재대학교 컴퓨터공학과

A XML Canonicalization System for Digital Signature on Mobile Environment

Yu Lei · Xian-Yu Hong · Hwa-Mok Yoon · Bong-Kyu Choi · Hoe-Kyung Jung

Dept. of Computer Engineering, Paichai University

E-mail : {yulei0622, , hongxianyu, bonkyu1963.choi, hkjung}@pcu.ac.kr, hmyoon@kisti.re.kr

요 약

이동 통신기술의 발달로 모바일 환경에서 대량의 데이터 전송이 가능해졌고, 이를 기반으로 다양한 모바일 서비스가 제공되고 있다. 특히 전자서명을 사용하여 제공되는 서비스들은 XML로 기술된 전자서명 정보를 단말 간에 송수신한다. 이때 다양한 물리적 문서형태를 허용하는 XML의 특성은 어플리케이션에서의 전자서명 유효성 검증 문제를 유발한다. 이는 W3C에서 제정한 Canonical XML 1.0 표준의 XML 정규화를 통해 해결이 가능하지만, XML 네임스페이스에서의 속성 상속 문제로 인해 제한적인 해결책만을 제시한다. 이를 해결하기 위해 W3C에서 Canonical XML 1.1을 표준화 중에 있으며 Candidate 권고안까지 진행되어 표준화를 앞두고 있다.

이에 본 논문에서는 모바일 환경에서의 보다 폭넓은 XML 정규화를 지원하기 위해 W3C에서 표준화 중인 Canonical XML 1.1 표준을 기반으로 XML 문서의 논리적 상호 동등성을 보장하는 XML 정규화 시스템을 설계 및 구현하였다.

ABSTRACT

Along with the developing of mobile communication technology, plenty of data transmission turn into possible in mobile environment. As the foundation, it can provide diverse mobile service. Especially the service which use electronic signature, and as the transmission of XML technology among the terminal digital signature information. By now, allowed plenty of validity confirmation questions that the digital signatures about the characteristic induced application of physical property XML. For this question, it can be solved through XML by Canonical XML 1.0 standards provided in W3C. But, because the question in the XML namespace attribute inheritance, proposed the restrictive solution. In order to solve this problem, proposes candidate plan of Canonical XML 1.1 standardized in W3C, and also even more standardization.

In this paper, in order to support the widespread XML standardization in the moving environment. Standardizes Canonical XML 1.1 standard as the underlies, safeguarding the theoretical mutual identity of the XML documents, and constructing and realizing the XML standardization system.

키워드

XML, Canonical, 디지털 서명

I. 서 론

현재 이동 통신기술의 발달로 모바일 환경에서 대량의 데이터 전송이 가능해 졌고, 이를 기반으로 동영상, 사진, 게임, 금융, 교육, 교통 정보 등의 다양한 서비스가 제공되고 있다. 특히 전자서명을 기반으로 서비스를 제공하는 금융 관련 서비스에서는 사용자에게 결제 서비스를 제공하기 위해 W3C에서 제정한 XML 1.0을 사용하여 전자결제 문서의 생성과 단말간의 송수신을 수행한다[1]. 이때 동일한 의미를 표현하는 XML 문서는 다양한 형태의 문서로 표현될 수 있는 특성으로 인해 어플리케이션에서 전자서명 유효성 검증 시 검증 오류를 유발한다. 이는 W3C에서 표준화한 Canonical XML 1.0 권고안을 통해 XML문서가 물리적으로도 동일한 형태임을 보장하여 해결이 가능하다[2]. 그러나 Canonical XML 1.0은 XML

네임스페이스에서의 속성 상속 문제로 인해 제한적인 해결책만을 제시한다[3]. 이를 해결하기 위해 W3C에서 Canonical XML 1.1을 표준화 중에 있으며 Candidate 권고안까지 진행되어 표준화를 앞두고 있다[4].

이에 본 논문에서는 모바일 환경에서 모든 서비스의 XML 처리가 가능하도록 XML 정규화를 지원하기 위해 W3C에서 표준화 중인 Canonical XML 1.1 표준을 기반으로 XML 문서의 물리적 동등성을 보장하는 XML 정규화 시스템을 설계 및 구현하였다.

II. 관련연구

2.1 Canonical XML 1.0

Canonical XML 1.0은 논리적으로 동일한 XML 문서를 물리적인 형태에서도 동일성을 보장하는 규칙이다. 이때 XML 문서의 물리적 동일성을 보장하는 작업을 정규화라고 한다. XML 문서 정규화는 Canonical XML 1.0에 기술된 규칙과 절차에 의해 진행되며 해당 과정을 통해 만들어진 XML 문서를 정규 XML이라고 한다. XML 정규화 규칙은 다음과 같다.

- 문서는 UTF-8로 인코딩
- 구문분석 전에 입력 행 종료는 #xA로 정규화
- 검증 처리기 규칙에 따라 속성 값 정규화
- 문자 참조와 파싱된 개체 참조는 대치
- CDATA는 문자 콘텐트로 대치
- XML 선언과 DTD 제거
- 빈 엘리먼트는 시작/종료 태그 쌍으로 변환
- 문서 엘리먼트 외부의 공백 문자와 시작 태그와 종료 태그 사이의 공백 문자 정규화
- 문자 콘텐트 안의 모든 공백 문자는 보존 (행 종료 정규화 동안 제거되는 문자 제외)
- 속성 값 구분자는 " (double quote)로 설정
- 속성 값과 문자 콘텐트의 특수 문자들은 문자 참조로 대치
- 기본속성 각 엘리먼트에 추가
- 각 엘리먼트의 네임스페이스 선언과 속성들을 알파벳 순서로 정렬
- 각 엘리먼트의 네임스페이스 선언을 루트 엘리먼트에 삽입 후 각 엘리먼트에서 제거

상위 규칙에 따라 XML 문서 정규화 시 각 노드의 종류에 따라 처리가 이루어지며 그 종류는 다음과 같다.

- 루트 노드
- 엘리먼트 노드
- 네임스페이스 노드
- 속성 노드
- 텍스트 노드
- PI(Processing Instruction) 노드
- 주석 노드

루트노드는 XML 문서의 최상위 엘리먼트이다. 엘리먼트 노드는 루트 엘리먼트의 모든 하위 엘리먼트이다. 네임스페이스 노드는 모든 엘리먼트의 집합을 정의하는 노드집합이다. 속성 노드는 공백, 노드의 QName, 등호(=), 인용부호("), 수정된 스트링 값으로 이루어진다. 텍스트 노드는 스트링 값을 그대로 사용한다. 단, &는 & 로 모든 여는 꺽쇠 괄호(<)는 <로 괄호(>)는 > 로 모든 인용부호(")는 "로, horizontal tab은 &#x9로 line feed는 &#xA로 carriage return은
 로 대치된다. PI 노드는 외부 어플리케이션을 참조할 수 있는 노드이다. 주석 노드는 주석 없는 정규 XML을 생성할 경우 아무 것도 생성하지 않는다.

2.2 Canonical XML 1.1

Canonical XML 1.1은 Canonical XML 1.0의 모든 표준을 준수한다. Canonical XML 1.0과의 차이점은 xml:base 속성이 추가된 점과 xml:id가 수정된 점이다[5]. 차이점은 다음과 같다.

- xml:base는 XML 엘리먼트를 URI(Uniform Resource Identifiers)로 연결하는 수단을 제공한다.
- C14N-Issues에 따라 개선한 xml:base 속성을 처리한다[6].
- xml:id는 XML 문서의 엘리먼트에 대한 고유 식별자를 나타내는 속성에 대한 공식 규약 이다.

URIs and IRI(Internationalized Resource Identifiers)는 거의 모든 XML 어플리케이션에 있어서 중요하다. XInclude와 기타 문서 결합 방식과 외부 리소스를 참조하는 많은 기술들을 포함하여, XML 엔터티들에 대한 접착제와 같은 역할을 한다. URI는 또 다른 URI와 비교하여 지정될 수 있고, XML 문서에 상대 URI를 지정한다면 프로세서에 의해 문서가 로딩되었던 방식에 따라 변환된다. 간혹, 문서 작성자는 URI가 변환되는 방식에 대해 더 많은 제어권을 갖고 싶어 하며, xml:base는 이 같은 컨트롤을 제공한다. 사용자는 XML 엘리먼트를 기본 URI와 연결하여 그 엘리먼트 내의 상대 URI가 문서 엔터티의 URI 보다 오버라이드된 기본에 따라 변환되도록 한다.

많은 XML 어플리케이션은 엘리먼트를 구분하는 메커니즘을 필요로 한다. 예를 들어, XHTML 엘리먼트에 대한 아이디는 문서 내의 연결에 사용된다. 이는 HTML 문서의 앵커 이름이 내부 링크에 사용되는 방식과 비슷하다. DTD는 하나의 속성의 아이디 유형을 선언하는 메커니즘을 제공하지만, 프로세서가 모든 경우를 다룰 수 없다는 점에서 문제가 있다. 이 같은 복잡함 때문에 어떤 속성이 아이디 구조를 갖고 있는지

를 표현하기가 어렵다. xml:id Version 1.0은 xml:id 라는 이름을 가진 속성(xml 접두사는 특별한 XML 네임스페이스와 연결)이 유연한 어플리케이션에 의해 고유 식별자로서 취급된다는 규약을 나타내는 간단한 명세이다.

III. 시스템 설계

본 시스템의 구조를 그림 1에 나타내었다.

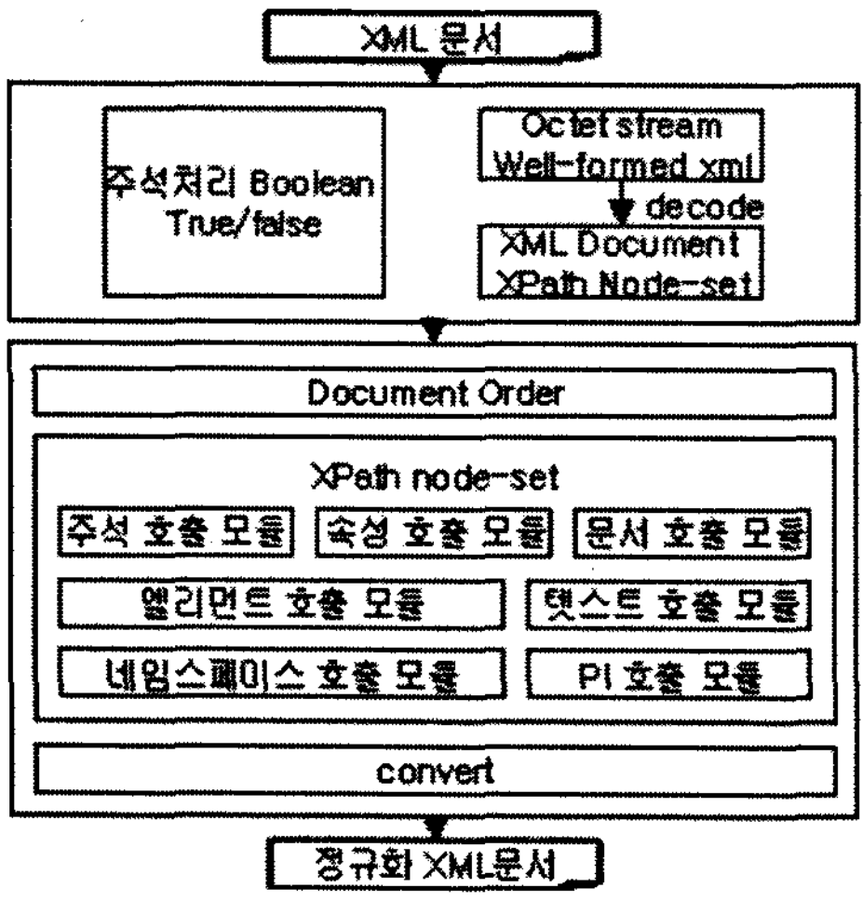

그림 1. 시스템 구조도

W3C에서 정의한 Canonical XML 1.1에 따르면 입력으로 두 가지 인자를 받아들이도록 되어 있다. 첫번째 인자로는 주석을 넣을 것인지 결정하는 주석처리 불리언(Boolean)값이고 두 번째 인자로는 XML 문서를 옥텟스트림으로 받아들일 것인지, 노드셋으로 받아들일 것인지를 결정한다.

두 번째 인자는 문서를 받아들이는 형식인데, 그 값이 문서 형식이라면 바로 SerializeNode 모듈로 보내지고, 파일 형식으로 받아들였다면 문서 형식으로 변환한 뒤에 보내진다. SerializeNode 모듈은 받아들인 노드의 형식과 5가지 노드의 형식(문서, 엘리먼트, 텍스트, PI, 주석)을 비교하여 그 중 일치하는 노드에 따라 분기하는 역할을 수행하고 이 중 엘리먼트 노드에서는 그에 해당하는 네임스페이스와 속성 노드를 처리한다. 분기된 노드는 그에 대응하는 처리 모듈로 보내지게 되고 각 처리 모듈은 DOM을 이용하여 원본 XML 문서로부터 가져온 데이터와 해당 노드에 적절한 텍스트를 혼합하여 임의의 텍스트에 쓰는 형식이다[7].

본 시스템의 흐름을 그림 2에 나타내었으며, 처리순서는 다음과 같다. 문서 노드 처리에서는 문서의 가장 최상위 노드인 문서 노드를 받아들여 자신이 가진 자식 노드의 수만큼 반복하여 처리한다.

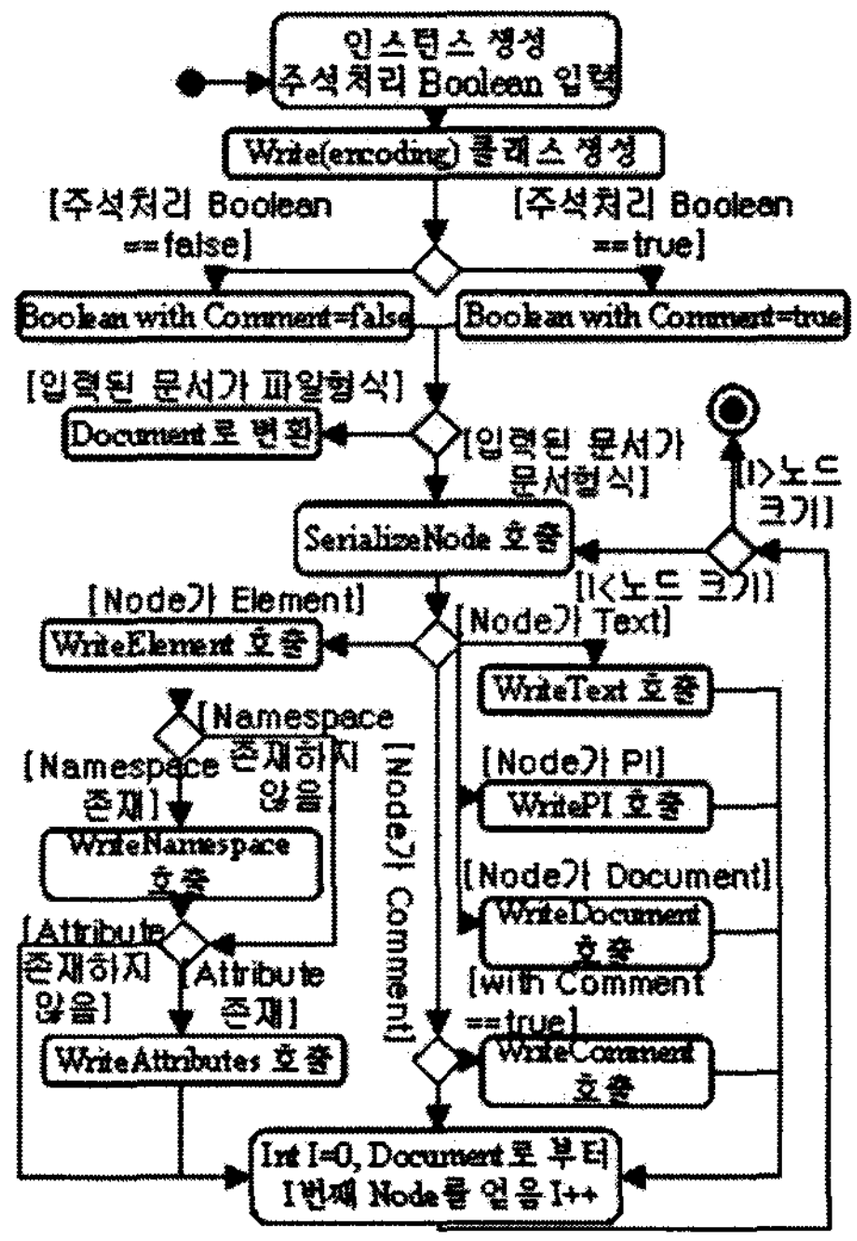

그림 2. 시스템 흐름도

네임스페이스 노드 처리는 엘리먼트 노드 처리에서 호출된다. 이 처리 모듈은 정렬된 네임스페이스 노드를 받아들여 순서대로 순차적으로 처리한다. 엘리먼트 노드 처리에서는 그 엘리먼트에 속한 네임스페이스 노드와 속성 노드를 같이 처리한다. 속성 노드 처리 또한 엘리먼트 노드 처리에서 호출되는 모듈로써 정렬된 속성들을 순차적으로 받아들여 처리한다. 텍스트 노드처리에서는 명세에서 지정한 <, >, &, &#x9, &#xA, &#xD 네 문자를 각 문자의 참조로 대치하도록 DOM을 이용하여 텍스트를 가져와 그 텍스트를 문자 단위로 나누어 비교, 대체한다. 처리 지시자 노드의 경우 '<'와 '?'를 작성하고 DOM을 이용하여 처리 지시자의 이름과 공백을 작성한 뒤 처리 지시자의 텍스트를 작성하고 '?'과 '>'를 작성한다. 주석 노드의 경우 '<'와"!--"를 작성하고 역시 DOM을 이용하여 주석 노드로부터 텍스트를 가져와 작성한 뒤 "--"과 '>'를 작성한다.

IV. 시스템 구현

본 시스템의 개발환경은 IBM-PC 호환 컴퓨터의 Windows XP Pro SP2 운영체제에서 개발하였고, 모발일 플랫폼개발을 위해 SKT IDE Tool, Visual C++ 6.0 SP6을 사용하였다.

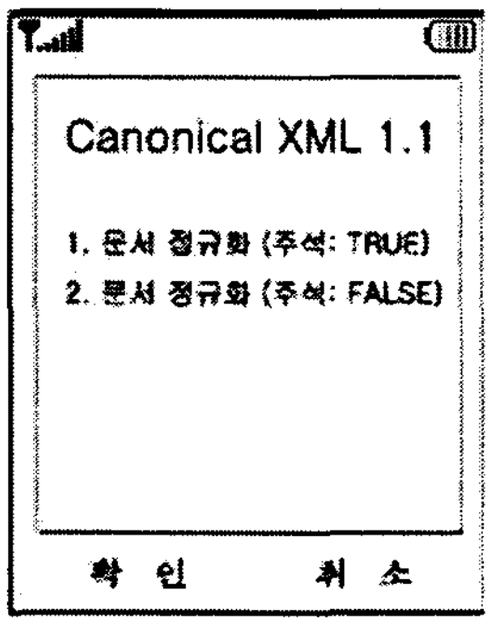

그림 3. 사용자 인터페이스

그림 3은 모바일환경에서 테스트를 위한 사용자 인터페이스 화면이다. 인터페이스의 창에는 정규화 시킬 원본 XML 문서를 보여주고 그 상단의 도구막대의 정규화를 위한 "문서 정규화(주석:TRUE)" 또는 "문서 정규화(주석:FALSE)" 선택해서 버튼을 확인하여 발생하는 이벤트로 정규화 모듈을 호출하여 정규화한다.

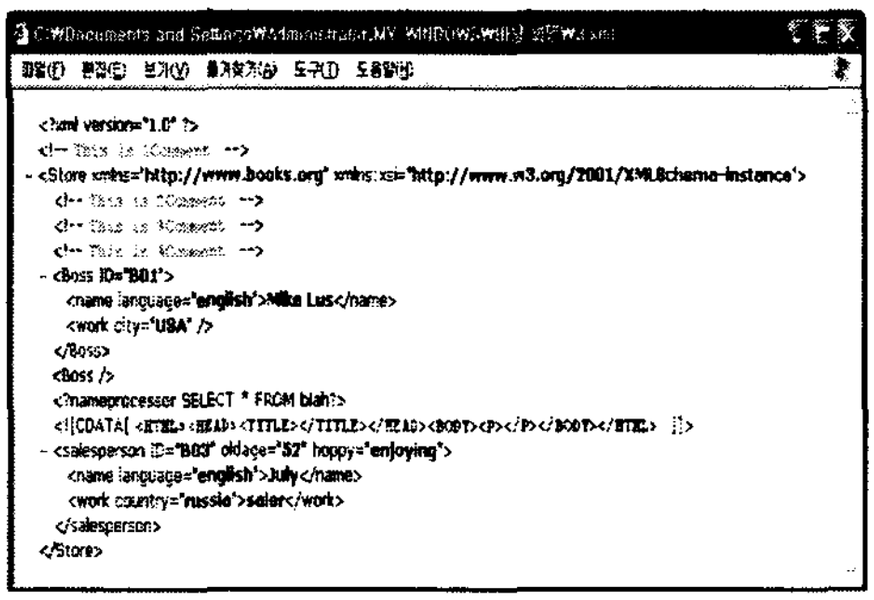

그림 4. 입력 XML 문서

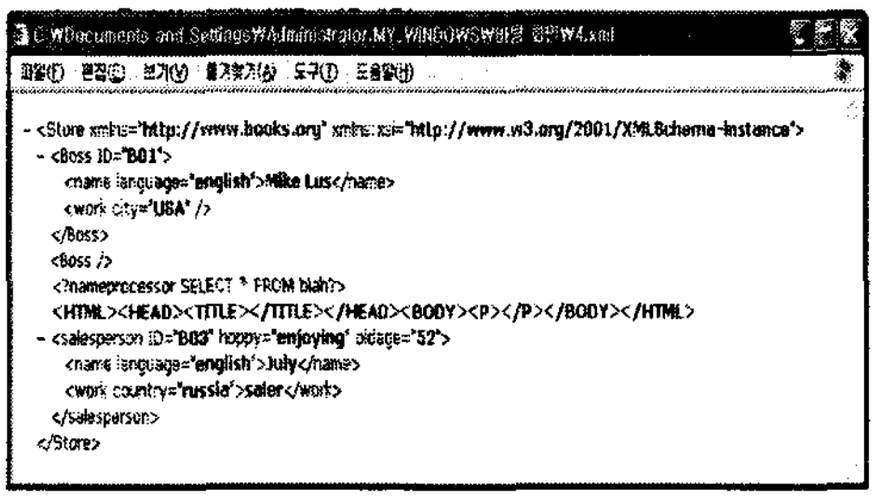

그림 5. 변환된 XML 문서

그림 4와 그림 5는 입력문서와 본 알고리즘을 이용하여 정규화시킨 문서이며 지면 관계상 개행된 부분이 있다. 변환 결과를 살펴보면, XML 선언부가 제거되었음을 볼 수 있고, 시작 태그 내의 네임스페이스 사이에 존재하는 불필요한 공백문자와 개행 문자가 제거되었음을 볼 수 있다. 시스템 입력부분에서 주석 노드의 경우에는 주석을 추가하도록 선택하여 결과 문서에는 주석이 포함되었음을 볼 수 있다. 만약 선택을 "문서 정규화(주석:FALSE)"로 설정하고 정규화 시킨다면 주석이 포함되지 않는다. CDATA 섹션 처리는 그 내부 값은 참조되는 엔터티 값으로 대체시켜야 한다는 명세에 따라 이 노드를 일반 텍스트와 같이 취급하여 텍스트 노드 처리에서 처리하여 해당하는 문자들이 대체되었음을 확인할 수 있다. 엘리먼트에 존재하는 속성들과 네임스페이스의 순서가 네임스페이스, 속성 순으로 정렬되고 여러 속성들은 알파벳순으로 정렬되었음을 볼 수 있다.

V. 결 론

본 논문에서 제안한 시스템의 특징은 Canonical XML 1.1 표준 명세를 적용시킴으로써 다양한 XML 파서로 인한 상호 운용성 문제를 해결하였고 작성자에 따라 논리적으로는 동일하지만 물리적으로 상이할 수 있는 XML 문서를 동일한 물리적 구조를 가지도록 함으로써 좀 더 정교하게 정규화된 문서로 변형할 수 있다. 또한 본 시스템을 모듈화 하여 생성함으로써 다른 여러 응용프로그램에서도 이 모듈을 사용하여 XML 문서를 정규화 시킬 수 있다. 따라서 XML 문서의 정규화를 통하여 디지털 서명 시스템에서의 사용이 용이할 뿐만 아니라, XML 문서 교환 시 물리적 동일성이 요구되는 많은 응용분야에서의 핵심 요소 기술로 사용될 수 있으리라 사료된다.

향후 연구방향으로는 현재 웹 서비스에서 전송 프로토콜로 사용되는 SOAP(Simple Object Access Protocol)을 정규화 시키기 위한 연구가 진행되어야 하며, 또한 XML 전자서명 시스템과 연계하여 통합형 웹 서비스 보안 모델에 관한 연구가 필요하다.

참고문헌

[1] W3C, "eXtensible Markup Language 1.0", 2006
[2] W3C, "Canonical XML Version 1.0", 2001
[3] W3C, "Namespace in XML 1.0", 2006
[4] W3C, "Canonical XML Version 1.1", 2007
[5] W3C, "XML Base", 2001
[6] W3C, "Known Issues with Canonical XML 1.0 (C14N/1.0)", 2006
[7] W3C, "Document Object Model (DOM) Level 1", 2000