情 報 保 護 學 會 誌
第 13 券 第 4 號, 2003. 8

XML 기반 접근제어 기술 동향

김 주 한*, 문 기 영*

요 약

XML 기반 접근제어 기술(eXtensible Access Control Markup Language, XACML)의 목적은 인터넷 상의 접근제어 서비스를 위한 다양한 제품들 및 그 제품들의 서로 다른 환경들 사이에서 일관되게 적용할 수 있는 권한부여(authorization) 정책을 제공하고, 그 정책을 통하여 기존의 다양한 환경 및 방식을 가진 접근제어 제품들에 상호운영성을 제공하기 위한 것이다.

XACML의 구성은 접근제어를 위한 XML 기반의 언어로 접근제어 정책(policy) 언어와 요청/응답(request/response) 언어로 되어 있으며 OASIS(Organization for the Advancement of Structured Information Standards)에서 표준화가 진행중이다. 정책 언어는 누가 언제 무엇을 할 수 있는 지를 기술하는 접근 제어 정책들을 표현하는 데에 사용되며, 요청/응답 언어는 특정 접근이 허용되는 지에 대한 질의를 표현하거나 그 질의에 대한 응답을 기술하는 데에 사용된다. 본 논문에서는 XACML의 대한 기술 소개와 표준화 등의 기술 동향을 분석한다.

Ⅰ. 서 론

XML이 ebXML, 웹 서비스 및 차세대 인터넷 기술인 Semantic Web 등에서 문서 표준으로 정착되고 있어 그 사용이 급격하게 늘어가고 있다. 그에 따라 XML 문서 및 데이터에 대한 효율적인 정보보호가 필요하게 되어 XML 정보보호기술이 등장하게 되었다.

기존의 정보보호기술들은 XML 문서 처리 및 XML 데이터 교환 등에 있어 중간 단계에서 XML이 아닌 다른 형태의 문서 및 데이터 형태로 변환하여 처리함으로써 XML 사용 및 확장에 장애가 되어 왔다.

XML 정보보호기술은 기존의 정보보호기술들과는 달리 그 처리 결과들이 XML 형태를 가짐으로써 일관되고 쉽게 XML이 가지고 있는 장점들을 살리면서 XML 문서 및 인터넷 상의 어떤 자원들도 XML 형태로 보호할 수 있는 장점을 갖는다.

이 논문에서 소개될 XML 기반 접근제어 기술(이하, XACML)은 XML 정보보호기술 중의 하나로서 자원들 혹은 접근 요청 개체들에 권한부여(authorization)를 통해 자원들에 대한 접근 제어(access control)를 하는 XML 기반의 언어이다. 또한, 다양한 접근제어 제품들에게 일관되게 적용될 수 있는 권한부여 정책들을 위한 통합 언어를 제공함으로써 광범위한 관리 및 권한부여 제품들에게 상호운영성을 제공한다.[1]

XACML은 특정 자원에 접근하기 위해 전송되는 요청자의 속성들(예로, SAML assertions, Java permission, 또는 WS-Security tokens 등)을 안전한 메커니즘들과 결합시켜 웹 서비스, J2SE 및 다른 전자상거래 환경들의 권한부여 인프라를 구성하게 하는 요소기술이기도 하다.

이 논문은 다음과 같이 구성된다. 이 논문의 2장에서 XML 정보보호기술의 개념, XACML의 개념 및 범위와 관련된 동향에 관해 설명한다. 3장에서는 XACML의 구성 및 처리 규칙 등을 기술하고 4장에서는 관련 연구, 제품들 및 및 표준화 동향을 분석한다.

Ⅱ. XACML 개요

1. XML 정보보호기술의 개념

XML 정보보호기술은 기존의 암호 및 보안 기술과

* 한국전자통신연구원 능동보안기술연구팀(juhankim, kymoon@etri.re.kr)

그 기술들의 특징에 맞는 XML 어휘와 처리 규칙을 정의한 XML 기술을 결합시켜 만든 것으로 XML 장점을 살려 유연하고 확장성이 강한 것이 특징이다.

다음은 XML 정보보호 표준으로 무결성과 서명 해법을 위한 XML 전자서명[3], 기밀성을 위한 XML Encryption[8], 공개 키 등록과 위치와 검증을 위해 XML Key Management(XKMS)[4], 인증과 인가 정보를 교환하고 속성에 대한 주장(Assertion)을 운반하기 위한 프로토콜인 Security Assertion Markup Language(SAML)[6], 접근 제어 규칙을 정의하는 XML Access Control Markup Language (XACML)[5], 그리고 사생활 정책과 선호를 정의하는 Privacy Preferences(P3P)를 위한 Platform[9]을 포함한다. 이들 XML 정보보호의 활용으로는 ebXML의 보안과 웹서비스 보안(WS-Security)[7], Digital Rights Management(eXtensible Rights Markup Language 2.0 - XrML)[10] 보안 등이 있다.[2]

다음은 주요 XML 정보보호기술들이다.

〔표 1〕 XML 정보보호기술 분류

기술 분류	서비스 유형
XML 전자서명	전자서명, 인증, 무결성, 부인방지
XML 암호기술	기밀성
XKMS	효율적인 키 관리
SAML	인증, 인가
XACML	접근 제어

2. XACML의 개념

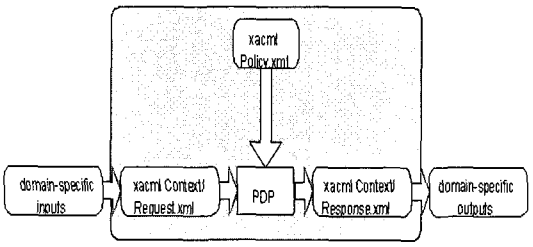
XACML은 XML로 기술된 정책 언어와 접근 제어 결정 요구/응답(request/response) 언어로 구성된 접근 제어에 대한 세계적인 표준이다.

정책 언어는 규칙, 정책 및 정책셋 등에 관한 통상적인 접근 제어 요구사항들에 대해 기술하고 있으며 함수들, 데이터 타입들 및 조합 논리(combining logic) 등에 대해서도 정의하고 있다.

요청 언어는 어떤 개체(subject)가 특정 자원(resource)에 대해서 특정한 동작(action)을 수행할 수 있는 지에 대한 질의를 구성할 수 있게 하고 응답 언어는 요청에 대한 결과를 표현하는데 사용하며 응답은 허용(permit), 거절(deny), 부정(indeterminate), 비적용(not applicable) 등의 4가지 결과로 표시된다.

XACML은 OASIS의 표준으로 XACML 기술위원회에서 2003.2월에 발표되었으며, 이 기술위원에서는 XACML 요구사항 분석서 초안, 여러 응용에서 접목 사용할 수 있도록 하기 위한 다양한 사용 사례에 대한 초안들, 다른 표준들과 함께 사용할 때 필요한 프로파일에 대한 초안 등에 작업도 진행중이다.

다음 〔그림 1〕은 XACML에서 정의하고 표준 영역을 나타낸 것이다.



〔그림 1〕 XACML 문맥(context)

XACML에서 정의하고 있는 부분은 그림 1에서의 어두운 부분으로 정책 언어와 응답/요청 언어임을 알 수 있다. 또한, 요청 언어의 문맥으로 들어온 요청을 정책 언어로 기술된 정책들을 기준으로 처리하고 다시 응답 언어로 그 결과를 생성하는 정책 결정 부분(Policy Decision Point, PDP) 등도 표준에 기술되어 있다.

어떤 접근 제어 어플리케이션이 XACML과 호환된다고 하면 요청하는 문맥이 XACML의 요청 언어로 기술되고, 응답 받은 XACML로 기술된 문맥을 이해할 수 있음을 뜻한다. 호환되지 않는 접근 제어 어플리케이션이라면 요청/접근 문맥을 XACML 요청/응답 언어로 기술된 문맥으로 바꿀 수 있는 변환이 필요하다.

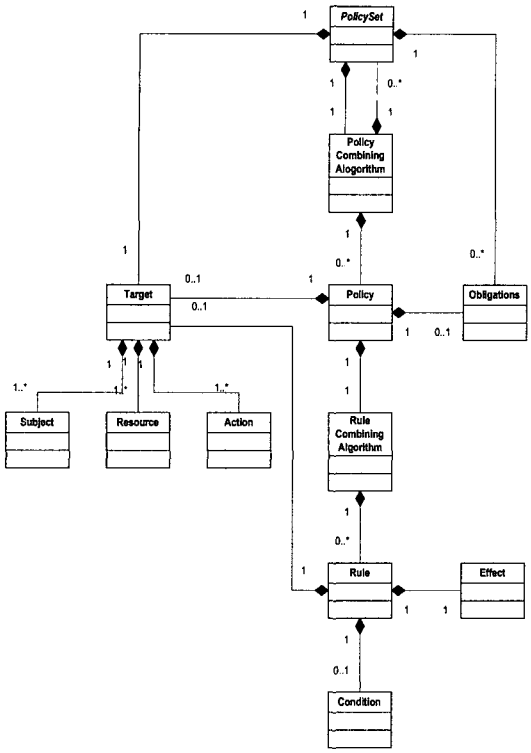
Ⅲ. XACML 구성 및 처리 절차

1. XACML 구성

XACML은 정책 언어와 요청/응답 언어로 구성되어 있다. 정책 언어는 가장 기본이 되는 규칙(Rule), 여러 규칙들을 포함하는 정책(Policy) 및 정책들의 집합인 정책셋(PolicySet)으로 구성된다.

다음 〔그림 2〕는 정책 언어에 대한 XML 스키마를 구조도로 표현한 것이다.

〔그림 2〕에서처럼 규칙, 정책 및 정책셋은 개체, 자원, 동작으로 정의된 타깃(target)을 포함하고 있다. 이 타깃은 PDP에서 요청 문맥을 읽고 결정(decision)을 함에 있어서 요청 문맥의 개체, 자원 및 동작의 내용을 각 규칙들, 정책들, 정책셋들의 타깃의 내용인



〔그림 2〕 정책 언어 모델

개체, 자원 및 동작과 비교하여 적정한 규칙들, 정책들 및 정책셋들을 찾는 데에 사용된다. 예를 들어, 어떤 사용자가 특정 자원에 대해 접근을 요청을 하기 위해 요청 문맥에 개체에 사용자에 대한 정보를, 자원에는 요청하는 자원의 정보를 그리고 동작에는 그 자원을 가지고 하는 행동(예로, read, write 등)을 기술하여 보내고, PDP는 요청 문맥상의 그러한 정보들과 일치하는 규칙들, 정책들 혹은 정책셋들을 찾아내는 것이다.

이렇게 요청 문맥에 해당하는 규칙들이 하나가 아니라 다수일 때는 각 규칙들의 결과들을 규칙 조합 알고리즘(rule combining algorithm)을 통하여 조합하여 결과를 낸다. 정책의 경우에는 정책 조합 알고리즘(policy combining algorithm)이 따로 있어 요청 문맥에 해당하는 정책들의 결과들을 조합하여 그 결과를 내고 응답 문맥에 넣는다.

그러나, 특정 규칙의 타깃이 요청 문맥에 해당되더라도 규칙에 포함되어 있는 조건(condition)에 어긋나면 그 요청 문맥에 대한 그 규칙의 결과는 비적용으로 된다. 규칙이 요청 문맥에 해당되고 조건을 만족하면 그 규칙의 결과는 규칙 요소 안에 있는 효과(effect)에 정의된 허용(permit)이나 거부(deny)가 된다.

다음 〔표 2〕는 요청 문맥에 따른 규칙의 평가표이다.

〔표 2〕 규칙 평가표

Target	Condition	Rule Value
"Match"	"True"	Effect
"Match"	"False"	"NotApplicable"
"Match"	"Indeterminate"	"Indeterminate"
"No-match"	Don't care	"NotApplicable"
"Indeterminate"	Don't care	"Indeterminate"

요청 문맥에 대한 정책에 대한 평가는 규칙에 대한 평가보다 복잡하다. 정책 안에는 많은 규칙들이 존재할 수 있으며 이 규칙들의 결과들을 어떤 규칙 조합 알고리즘에 적용하는가에 따라 그 결과가 달라지기 때문이다.

정책셋에 대한 평가는 정책에 대한 평가와는 또 다르다. 정책은 여러 규칙들의 조합 알고리즘으로 평가되지만, 정책셋은 그런 정책들의 결과들을 어떤 정책 조합 알고리즘에 적용하는가에 따라 그 결과가 달라진다.

다음 〔표 3〕은 대표적인 조합 알고리즘들이다.

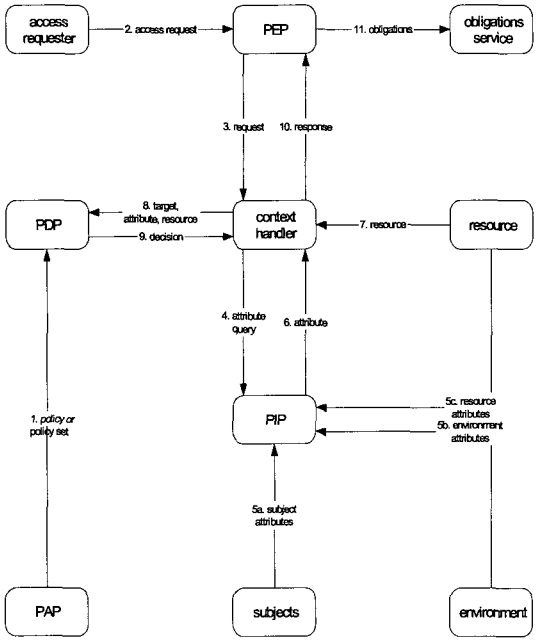
〔표 3〕 대표적인 조합 알고리즘들

조합 알고리즘	적용 방법	비고
Deny-Overrides	하나의 deny가 나오면 결과는 deny	규칙, 정책 모두에 사용
Permit-Overrides	하나의 permit이 나오면 결과는 permit	규칙, 정책 모두에 사용
First-Applicable	제일 처음 나온 결과가 최종 결과가 됨	규칙,정책 모두에 사용
Only-one-Applicable	적용 가능한 정책이 하나 이상이면 부정이 됨	정책 조합에만 사용

XACML 문맥을 구성하는 또 하나의 요소는 의무조항(obligations)이다. 이 의무조항은 정책에 포함되는 항목이며, 이 의무 조항을 포함한 한 정책의 평가가 자원에 대한 특정 동작을 허가하더라도 이 의무조항을 실행하지 않거나 이 의무조항을 이해할 수 없다면 그 정책에 대한 결과는 거부가 된다.

2. XACML 처리 절차

〔그림 3〕의 절차는 XACML 사용에 있어 전형적인 사용 절차를 나타낸 것이다.



〔그림 3〕 XACML의 데이터 흐름도

1. 정책 관리 부분(Policy Administration Point, PAP)에서 들어올 요청 문맥을 분석하여 미리 정책들이나 정책셋들을 작성한다.
2. 한 요청자, 즉 개체가 특정 자원에 대해 어떤 동작을 원할 때, 이 요청자는 그 자원을 보호하는 시스템(예로, 웹 서버, 파일 시스템 등)인, 정책 결정 부분(Policy Enforcement Point, PEP)에게 자원 허가에 대한 질의를 한다.
3. 이 PEP는 개체, 자원, 동작의 속성들을 선택적으로 포함할 수 있는 이 PEP 자체의 형식으로 요청자의 질의를 바꾸어 문맥처리기(context handler)로 보낸다. 이를 받은 문맥처리기는 4,5,6과 7 과정을 통해 XACML 요청 문맥으로 변환한다.
4. 개체, 자원 및 환경 속성들은 정책 정보 부분(Policy Information Point, PIP)으로 요청하여 얻을 수 있다.
5. PIP는 요청 받은 속성들을 구해온다.
6. PIP는 요청해온 속성들을 문책처리기로 반환한다.
7. 선택적으로, 문맥처리기는 요청 문맥의 자원을 포함할 수 있다.
8. 문맥처리기는 타깃을 포함한 결정 요구(decision request)를 PDP에 보낸다. PDP는 적용 가능한 정책을 찾고 필요한 속성들을 검색한다. 선택적으로 문맥처리기로부터 자원을 검색할 수도 있다. 그런 후 PDP는 찾은 정책을 평가한다.
9. PDP는 권한 부여 결정과 의무조항을 담고 있는 응답 문맥을 생성 문맥처리기에 반환한다.
10. 문맥처리기는 권한 부여 결정을 포함한 응답 문맥을 PEP 자체적인 응답 형식으로 변환하여 전송한다.
11. PEP는 의무조항(obligations)을 수행한다. PEP가 의무조항을 이해할 수 없거나 수행하지 못하면 접근을 거부해야 한다.
12. (〔그림 3〕에는 없지만) 자원에 대한 접근이 허용되면, PEP는 자원에 대한 접근을 허용해야 한다. 그렇지 않으면, 접근을 거부해야 한다.

위의 절차에서는 PEP가 XACML과 호환되지 않아도 되는 시스템인 것을 알 수 있다. PEP와 PDP 사이의 문맥처리기가 PDP의 XACML 요청/응답 문맥을 PEP의 자체의 요청/응답 형식으로 변환을 하고 있기 때문이다. 즉, 중간의 문맥처리기가 기존의 접근 제어 시스템에서 XACML 사용을 할 수 있게 하는 어댑터 역할을 하고 있다.

Ⅳ. XACML 연구동향

1. 표준화 동향

XACML의 표준화는 위에서 언급한 바와 같이 OASIS에서 XACML 기술위원회에서 진행 중이다. XACML 기술위원회는 IBM, Entrust, Open-Network Technology 및 Sun microsystems 등의 회사 등의 참여해 있으며 2001년에 표준화활동이 시작되어 2003년 2월에 XACML를 발표하였다.

현재 XACML과 관련되어 공식적으로 발표된 문서는 명세서, 정책 스키마(policy schema) 및 문맥 스키마(context schema) 등이 있으며 비공식적인 문서로는 호환성 테스트를 위한 테스트 벡터들이 있다.

그 외의 문서들로는 현재 발표된 명세의 기능 등을 보다 확장에 관련된 초안들과 다른 XML 표준들, 프로토콜 및 기타 다른 표준들과 연동하여 사용할 수 있도록 하는 프로파일 및 바인딩 초안들(예로, XML 전자서명, LDAP, ebXML 및 전송 프

로토콜 등)이 있다. 또한, XACML을 각각의 다른 분야의 응용 제품들(예로, DRM, ebXML 저장소, 온라인 서버 및 WorkFlow 등)에 적용하기 위한 적용 방안(use-case)등을 설계한 초안들이 있다.

XACML 명세, 여러 프로파일 및 바인딩 초안들 그리고 다양한 분야에 적용시키기 위한 적용 방안 등의 초안들은 아직 설계의 단계이거나 혹은 개념만 정립해 놓은 단계 등이 대 부분이다. XACML 명세의 경우에는, 새로운 조합 알고리즘들에 대한 특성, 구조적으로 기술될 수 있는 자원, 규칙에 대한 참조 및 기존 스키마의 수정 및 확장 등의 추가 설계 및 확장 등의 표준화 작업이 필요하다. 또한, XACML 명세 상에 정의PDP는 분산 환경에서 운영이 가능한데, 이 때 각 PDP들은 자신들에 해당하는 정책들을 어떻게 생성하고 분산할지 그리고 요청 문맥이 들어왔을 때, 어떤 PDP들에게 결정 요구를 해야되고 각 PDP들의 결정을 어떻게 조합할지 등에 대한 방법 등에 관한 연구도 필요하다.

프로파일이나 바인딩 초안들도 현재까지는 XML 전자서명에 대한 프로파일만 비교적 자세하게 소개될 뿐, 나머지 프로파일 혹은 바인딩 초안들은 아직까지는 개념 정립 상태이고 자세하게 기술된 것들이 없다. 또한, 여러 분야에 대한 XACML의 적용 방안에 대한 초안들도 마찬가지이다.

따라서, 앞으로의 OASIS의 XACML 기술위원회의 표준화 작업은 XACML 명세의 수정 및 확장, 프로파일과 바인딩 및 여러 적용 방안 등에 구체적인 설계 및 기술에 관한 방향으로 이루어 질 것이다.

2. 제품 동향

XACML 제품들에는 Sun Microsystems 사와 Jiffy Software 사가 개발한 제품이 XACML 기술위원회의 홈페이지에 소개되어 있다.

SUN 사의 XACML 제품[11]은 오픈소스로 자바 2 플랫폼 이상의 JDK 1.4 혹은 그 이상의 버전으로 구현되었다. 이 제품에는 XACML 명세 상의 필수 구현 부분들, 특히 정책들의 파싱, 요청과 응답의 관리 및 요청에 대한 정책들의 적용성 결정 및 정책들에 대한 정책들의 프로세싱 등을 구현하였다. 그러나, 이 제품은 XACML의 핵심만을 구현한 것으로 SAML, LDAP 등의 다른 보안 관련 재품들과의 연동성을 제공하지 않고 있다. 또한, XACML 명세 상에 언급되어 있는 수많은 데이터 타입들 및 함수들에 대한 모든 구현이 필요하며 새로운 데이터 타입들이나 함수들의 첨가가 쉽게 될 수 있도록 유연한 설계가 필요하다. Jiffy Software의 XACML에 관한 구현 제품[12]은 C++로 구성되어 있으며 오픈소스가 아니기 때문에 실행파일로 공개되어 있으며, 리눅스 x86과 윈도우즈2000, XP에서 실행가능하다. 이 회사는 미국과 영국의 개발자들이 모인 가상 회사로, 이 회사의 제품은 아직 정식 버전이 아닌 알파버전으로 XACML 명세가 OASIS 표준이 되기 전에 나온 것이다.

그밖에 많은 제품들이 있으나 XML 기반으로 접근제어를 하는 것은 거의 없다. 다만, 일본 IBM사에서 만들었던 XACL(XML Access Control Language)은 현재의 XACML의 모태가 되는 XML 기반 접근제어의 모델이되는 제품이다.

앞으로 나올 제품은 XACML 명세를 구현하는 것뿐만 아니라, 다른 XML 정보보호기술을 연동하여 사용할 수 있는 종합적인 정보보호를 제공할 수 있어야 한다. 또한, PKI연동, PMI 연동, LDAP 연동, 키 및 인증서 관리, 속성 관리 등의 정보보호를 위해 필요한 주변 기술들을 같이 제공하는 통합 XML 정보보호기술 제공이 가능해야 한다.

V. 맺음말

XML이 전자상거래를 위한 문서 표준 및 차세대 인터넷의 문서 표준으로 자리잡아 가고 있고 XML의 사용이 늘어감에 따라, XML의 장점을 유지하면서 정보보호가 가능한 XML 정보보호기술에 대한 표준들을 W3C와 OASIS에서 개발중이다.

XACML은 OASIS의 XACML 기술위원회에서 표준화 작업 중인 XML 정보보호기술 중의 하나이며, 권한부여를 통한 자원의 접근을 제어하는 XML을 기반으로 한 접근제어 기술이다. 권한부여는 규칙, 정책 혹은 정책셋을 통해 이루어진다. 이에 대한 실행은 접근 요청을 하는 요청 문맥에 의해 이루어진다. 그리고 그 결과는 응답 문맥으로 전송된다. 정책의 관점에서 XACML은 정책의 생성, 정책 평가를 위한 데이터 수집, 정책의 평가 및 정책의 집행 등의 순서로 진행된다. 권한부여는 위의 과정 중에서 정책의 생성 시에 이루어지고, 권한부여의 결정은 그 이 후 두 단계들을 통해 이루어지고 마지막으로 실행은 정책의 집행 단계에서 이루어진다.

XACML은 현재도 계속 표준화가 진행되는 기술

이며 이제 개발의 초기단계로 XACML 명세만 나왔을 뿐이다. 다른 XML 정보보호기술들과 사용하기 위한 프로파일 개발뿐만 아니라 전송 프로토콜들과 사용하기 위한 바인딩 기술 등의 많은 부분들이 앞으로의 과제로 남아있다. 또한, 기술적으로는 XACML의 다른 어플리케이션에 어떻게 적용되어 사용될 수 있는지를 보여주는 적용 방안 설계 등의 작업도 필요하다. 또한, 분산환경 등에서의 각각의 PDP들의 정책들에 대한 생성, 분류, 평가 및 집행 등의 효율적인 정책관리 방법도 연구되어야 한다.

PKI 및 PMI 연동, LDAP 연동, 키 및 인증서 관리, 속성 관리 등의 정보보호를 위해 필요한 주변 기술들과의 연동 및 개발 작업이 필요하다.

이러한 기술들이 모두 개발되고 통합되면 XACML을 포함하는 XML 정보보호기술은 무결성, 기밀성, 인증, 부인방지, 접근제어 등의 XML을 기반으로 하는 통합 보안서비스를 전자상거래 및 차세대 인터넷 등에 적용시킬 수 있게 된다.

참 고 문 헌

[1] "XACML Access Control Markup Language Ratified as OASIS Open Standard", CBDI-Forum, 25 Feb 2003

[2] 문기영, 손승원, "XML 정보보호 개요", 한국정보처리학회, 제10권 제2호, 2003.

[3] Mark Bartel, John Boyer, Barb Fox, Brian LaMacchia and Ed Simon, "XML Signature Syntax and Processing", http://www.w3.org/TR/xmldsig-core/,2002.

[4] Mark Bartel, John Boyer, Barb Fox, Brian LaMacchia and Ed Simon, "XML Key Management Specification(XKMS 2.0)", http://www.w3.org/TR/xmldsig-core/, Mar. 2002.

[5] OASIS, "OASIS extensible Access Control Markup Language Working Draft 14", http://www.oasis-open. org/ommittees/xacml/docs/,Jun. 2002

[6] OASIS, "Security Assertion Markup Language, http://www. oasis-open.org/committees/security/,Jan. 2003.

[7] OASIS, "Web Services Security (WS-Security) Version 1.0, http://www-106.ibm.com/developerworks/library/Ws-secure/,Apr. 2002.

[8] Takeshi Imamura, Blair Dillaway and Ed Simon, "XML Encryption Syntax and Processing", http:// www.w3.org/TR/xmlenc-core/,2002.

[9] W3C, "The Platform for Privacy Preferences 1.0 Specification", http://www.w3.org/TR/P3P, Apr. 2002.

[10] XrML.org, "extensible rights Markup Language(XrML) 2.0 Specification",http://www.xrml.org/, Nov. 2001.

[11] Sun's XACML Implementation, http://sunxacml.sourceforge.net/.

[12] Jiffy XACML, http://www.jiffysoftw-are.com/index.htm.

〈著 者 紹 介〉

김 주 한 (Ju-han Kim)
정회원

1997년 2월 : 충남대학교 컴퓨터과학과 졸업
1999년 2월 : 충남대학교 컴퓨터과학과 석사
2000년 8월~현재 : 한국전자통신연구원 능동보안기술연구팀
관심분야 : XML 정보보호, 저작권 보호, 전자상거래 보안

문 기 영 (Ki-young Moon)
정회원

1986년 2월 : 경북대학교 전자공학과 졸업
1989년 2월 : 경북대학교 전자공학과 석사
1992년 1월~1994년 3월 : (주)대우정보시스템 기술연구소 대리
1994년 3월~현재 : 한국전자통신연구원 능동보안기술연구팀 선임연구원
관심분야 : XML 정보보호, 전자상거래 보안, 분산시스템, 트랜잭션