ebXML 등록기/저장소 접근제어 시스템의 설계

성백호[0*], 차석일*, 박범대**, 신동규*

세종대학교 컴퓨터공학과*, 한국전산원**

{guardia, kiry, shindk}@gce.sejong.ac.kr {parkbd, }@nca.or.kr

Design of access control system for ebXML Registry/Repository

Baek-Ho Sung[0*], Hyoung-Seok Lee*, Dongsoo Kim**, Beomdae Park**, Dong-il Shin*, Dong-kyoo Shin*

*Dept. of Computer Engineering, Sejong University

**National Computerization Agency

요 약

XML을 기반으로 한 ebXML의 기술이 점차 확대되어 가고 있다. ebXML에서 등록기/저장소는 비즈니스를 수행하기 위한 정보의 등록, 발견, 저장 등을 위해 사용된다. 등록기/저장소는 비즈니스 수행을 위한 거래 당사자간의 합의문 및 각종 XML 문서들을 생성 및 저장하는 시발점이라는 측면에서 보안이 상당히 강조되어야 할 부분이다. 본 논문에서는 ebXML 등록기/저장소에서 XACML을 이용한 ebXML 자원 접근제어 시스템을 설계하였다.

1. 서 론

XML이 인터넷상에서 비즈니스를 위한 커뮤니케이션에서 확고한 자리매김을 함에 따라 보안의 필요성이 강조되고 있다. 여기에 비즈니스 파트너들 간의 거래를 지원하는 XML기반의 ebXML에서도 XML을 주도하고 있는 W3C와 OASIS는 XML Signature [1], SOAP [2], XKMS [3], SAML [4]과 XACML [5]등 현재 표준화하거나 진행중인 XML의 보안을 위한 명세를 ebXML에 적용하여 보안의 강화를 도모하고 있다. SAML은 인증과 승인 서비스 프레임워크를 제시하고 있고 XML기반 공개키 서비스를 위해 XKMS의 개발이 진행되고 있으며 XML을 이용하는 비중이 커지면 리소스에 대한 정확한 접근을 제어가 필요하게되고, 이에 대한 정책 수립이 중요한 문제로 제기되면서 XACML이 제안되었다.

이와 같은 외국의 보안 부분 강화와 맞물려 국내에서도 ebXML 보안 부분의 중요성을 인식하고 있지만 아직 연구 및 구현은 미흡한 실정이다. 그리고 이 보안 부분 중 ebXML의 자원 접근 제어에 대한 부분은 명세만 나와 있고 아직 적당한 시스템 시나리오나 설계가 거의 없다. 따라서 본 논문은 XACML과 등록기/저장소 [6]의 접목으로 자원에 대한 접근을 제어하여 기존 ebXML 명세에 보안 부분을 강화한 ebXML 등록기/저장소 접근제어 시스템을 설계하였다.

2. 관련연구

2.1 등록기/저장소(Registry/Repository)

등록기/저장소는 제출기관(Submitting Organization)에 의해 제출되어진 각종 정보들이 안전하게 저장될 수 있는 환경을 제공한다. 이러한 방식으로 등록기/저장소에 저장된 정보들은 ebXML을 기반으로 한 B2B 파트너 쉽 형성과 거래를 지원하기 위하여 사용된다. 제출기관이 등록한 정보는 XML Schema, XML 문서, 프로세스 설명, 핵심 컴포넌트, 컨텍스트 설명, UML모델, 당사자에 대한 정보를 비롯하여 소프트웨어 컴포넌트 정보가 될 수도 있다 [6]. 현재 2001년 12월 18일에 Registry Information Model은 2.0이, 2002년 4월 Registry Services Specification 2.0 버전이 표준 문서로 나와 있다 [7].

2.2 등록기/저장소 접근제어

등록기의 사용자는 AccessControlPolicy에 기반을 둔 미리 정의한 역할(role)에 의해 등록기/저장소의 자원에 접근을 할 수 있다. 역할의 종류는 등록기에 있는 자신이 소유한 모든 자원을 접근할 수 있는 ContentOwner, 등록기의 모든 자원에 접근 할 수 있는 RegistryAdministrator, 등록기의 모든 자원에 읽기(Read-only)로만 접근이 가능한 RegistryGuest등 3가지가 있다. 모든 사용자는 문건이 제출되면 등록기는 메시지에 있는 증명서로 인증을 하

고 ContentOwner로 역할을 부여한다. 읽기만 하는 클라이언트에겐 RegistryGuest의 역할을 부여한다. 현재 표준 문서로 2002년 4월 Registry Services Specification 2.0 버전이 나와 있다 [7].

2.3 관련 XML기반 보안 기술

1) SAML(Security Assertion Markup Language)

SAML은 경쟁 관계에 있던 S2ML과 AuthXML을 통합하여 e-비즈니스 프레임워크에서 인증(Authentication) 및 승인(Authorization)을 위한 표준적인 보안 기술을 제공하기 위해 OASIS의 STTC(Security Services Technical Committee)가 제안한 XML 기반의 보안 기술이다. SAML은 새로운 보안 기술을 제안하는 것이 아니라 기존의 보안 기술을 활용해 구성되며, 특정 업체의 방식이 아닌 표준적인 메시지 형식과 전송 프로토콜을 사용함으로써 플렛폼이나 솔루션 등에 독립적인 인증 및 속성확인, 승인 등의 서비스를 제공해 줄 수 있게된다.

SAML은 인터넷상에서의 자원 요청자에 대한 인증, 승인, 속성 확인 등을 수행하는 역할을 하며 이는 XML 기반의 다른 보안 기술들(XML 전자서명, XML Encryption, XKMS, XACML 등)과 통합되어 전체 보안 시스템을 구성하는 일부 요소로서 기능을 가진다. SAML 명세는 Assertion, 프로토콜, 바인딩으로 구성되어 있다. Assertion은 인증 및 승인 정보를 포함하는 XML 기반 구조를 가진다. 또한 Assertion의 인증을 위해 XML 전자서명을 적용한다. SAML 프로토콜은 XML 기반의 메시지 형태로서 요청 및 응답의 쌍으로 구성되어 각 Assertion에 대한 전송을 담당한다. 일반적으로 Assertion은 SAML 프로토콜의 응답을 통해 얻어진다. SAML 바인딩은 SAML Assertion 요청 및 응답 프로토콜을 표준 메시지 전송 프로토콜과 연동함에 있어 처리되어야할 방식을 정의하고 있다. 현재 SOAP-over-HTTP 바인딩이 기본적으로 사용된다. 현재 oasis committee specification으로 표준화가 진행중이다 [4].

2) XACML(eXtensible Access Control Markup Language)의 개요

XACML은 XML문서에 대한 접근을 정책리스트를 이용하여 제어할 수 있는 XML기반의 언어이다. XACML TC(Technical Committee)에서는 XACML로 정의된 기술로 정책과 인증을 표현하기 위한 XML 스키마를 제공하고 있다. 이 정책에서의 리소스는 XML을 사용하여 표현되는 어떠한 객체도 될 수 있으며 XACML은 XPath [8]나 LDAP 등 다양한 프로토콜과 함께 바인딩하여 사용될 수 있으며 새로운 프로토콜과도 함께 사용될 수 있다. XACML은 인증시스템의 접근과 접근자 요청의 특징적인 역할에 대한 제어를 할 것으로 기대된다.

XACML은 크게 <Subject>, <Resource>, <Action>의 3가지 요소로 구성되는데 subject는 사용자의 ID 나 그룹, 또는 역할 등을 나타낼 수 있으며, Resource 요소는 subject가 접근할 데이터를 의미하며 그 데이터 참조로서 단일 XML 문서에서 개별 요소 수준까지 지정 할 수 있다. action은 4가지 수행 가능 동작으로 구성되며 각각은 읽기, 쓰기, 생성, 삭제 작업이다.

XACML은 공통의 산업 명세를 만드는 국제적인 컴소시엄인 OASIS(the Organization for the Advancement of Structured Information Standards)에 의해 표준화되고 있으며 가장 최근의 기술문서는 2002년 7월 12일 문서로 현재 Working Draft상태로 있으며 표준화 진행중이다 [9].

3. ebXML 등록기/저장소 접근제어 시스템의 설계

본 논문에서 설계의 개념은 다음과 같다. 로그인 후 전달된 메시지를 MSH가 복호화와 서명검증을 한다. 그리고 assertion에 따른 콘텐츠의 URI를 얻어온다. 그리고 콘텐츠를 변경할때는 XACML에 접근평가를 요청하고 여기서 받은 결과로 콘텐츠를 변경한다. 또한 설계의 전제 조건은 다음과 같다.

- 등록기/저장소, SAML 및 XKMS 웹 서비스에는 이미 클라이언트의 사용자 등록 정보(XKMS의 경우 사용자 키 정보)를 가지고 있다.
- 클라이언트 및 등록기/저장소는 SAML 및 XKMS 웹 서비스와 신뢰관계에 있다.
- 클라이언트의 CPP는 이미 등록기/저장소에 등록되어 있다.
- XACML Policy문서는 이미 사용자의 역할을 기반으로 작성되어 있다.
- deprecated된 문서에 대한 Update기능만 사용한다.

그림 1은 Registry/Repogitory Usecase의 개념도로 각각의 부분은 다음과 같다.

• 등록기/저장소 Guest, 등록기/저장소 User : 사용자는 레지스트리에 특별한 권한이 없는 게스트계정 사용자나 콘텐츠에 대한 읽기, 변경권한이 있는 일반 사용자를 말한다.

• 키 정보 획득, 인증 정보 획득 : 각 사용자는 로그인을 하는 과정에서 XKML에서 키를 획득한 후 SAML로부터 사용자 인증 정보를 받게 된다.

• SAML메시지 처리 : 로그인 한 사용자의 SAML assertion에는 이름을 포함한 역할정보가 있다. 이 정보는 로그인 한 사용자의 기본 인증 XML파일이나 세션에 저장되고, action, resource 정보와 함께 request 문서를 생성하여 레지스트리 객체에 접근할 때 XACML Processor의 입력파일로 사용된다.

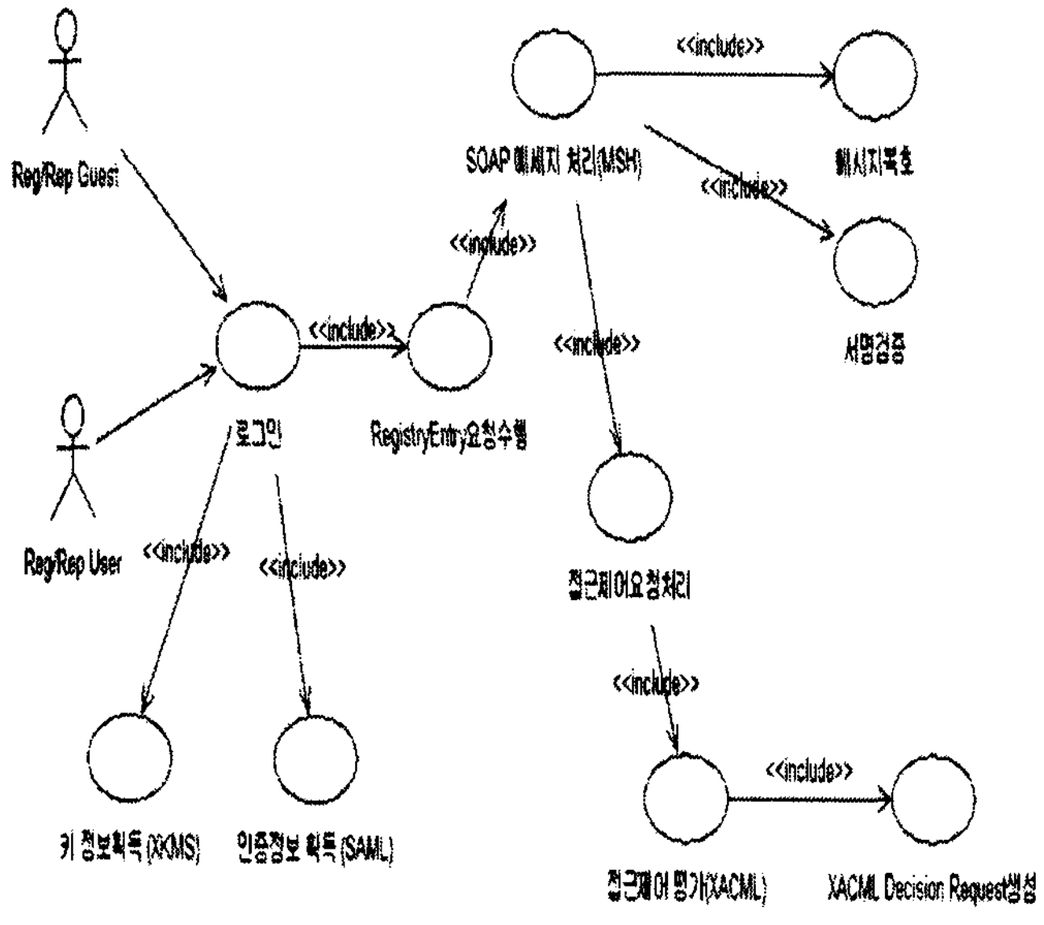

그림 1 Registry/Repogitory Usecase

• RegistryEntry 요청수행 : 로그인한 사용자는 객체에 접근을 시도하여 읽기, 변경 등 작업을 하기 위한 요청을 한다. 이때 Registry는 사용자에 적절한 접근 권한을 부여하기 위해 접근제어 평가를 수행하여 권한을 결정한다.

• SOAP메세지 처리: 사용자가 전송한 메시지는 보안을 위해 암호와 전자서명으로 감싸져 있다. 따라서 이 메시지로 요청을 수행하기 위해 암호를 풀고 전자서명을 인증해서 Registry에 접근하도록 한다.

• 접근 제어 평가 : 등록기/저장소에서 사용된 인증정보와 함께 접근하려는 resource와 action을 결정하여 프로세서에서 사용될 입력정보를 추출하고 권한 요구 파일을 생성하며 XACML프로세서에서 정책파일과 함께 접근권한이 결정되는 일련의 과정을 수행한다.

• XACML Decision Request생성 : 등록기/저장소에서 객체에 접근하려는 목적을 가진 메시지로 이 메시지는 객체에 접근하기 전에 XACML에 의해 정책을 평가받아야 하고, 만일 부여된 권한에 적합하지 않는 사항이라면 접근을 거부한다.

• 권한에 따라 객체에 접근 : XACML Processor에 의해 평가된 객체 접근 메시지는 정책에서 결정한 대로 read, write, update등의 접근 권한을 가지고 객체에 접근하게 되고 권한이 없다면 접근이 거부된다.

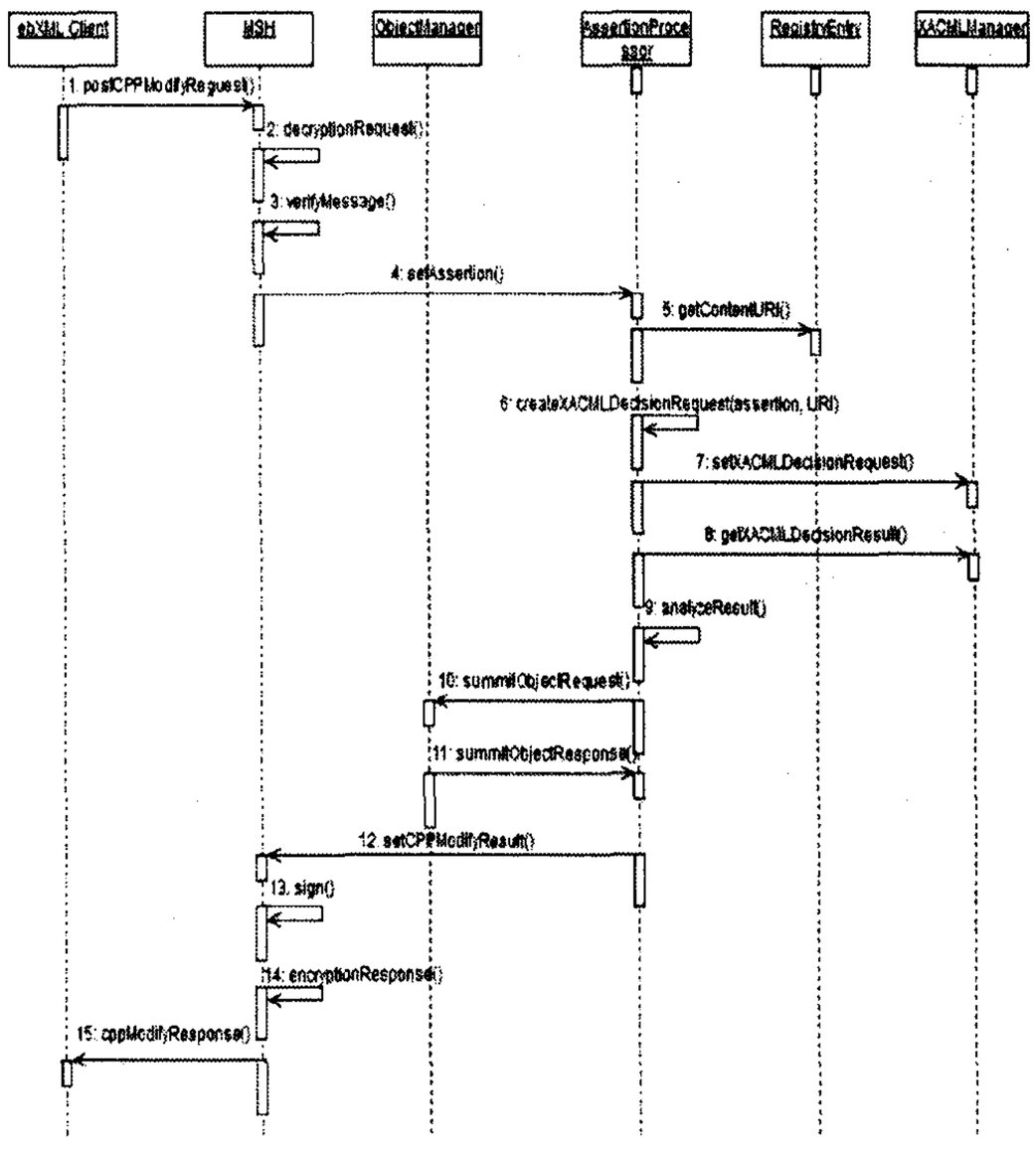

그림 2 Registry/Repogitory Sequence Diagram

그림 2는 Registry/Repogitory Sequence Diagram으로 client가 등록기/저장소로 전송한 메시지를 몇 가지 과정을 통해 처리하는 절차를 보여준다. 등록기/저장소는 MSH, ObjectManager, Assertion Processor, RegistryEntry등으로 구성되어 있다. MSH는 client로부터 온 메시지를 암·복호화와 전자서명을 인증한다. ObjectManager는 레지스트리에 저장된 문서의 업데이트를 수행한다. AssertionProcessor는 RegistryEntry로부터 콘텐츠 URI를 읽어온다. 그리고 이 콘텐츠에 대한 변경과 같은 작업

을 사용자로부터 요청 받으면 접근권한평가를 위한 요청을 생성하여 XACML Manager로 이 생성된 요청을 전송하여 결과를 응답 받아 분석한 후 ObjectManager로 변경요청을 한다. RegistryEntry는 실제컨텐츠의 메타정보를 가진 객체이다.

자세한 순서는 다음과 같다.

(1)postCPPModifyRequest() : ebXML Client로부터 등록기/저장소접근 메세지를 전송 받는다.

(2)decryptionRequest() : 암호화된 메시지를 복호화한다.

(3)verifyMassgae() : 서명을 검증한다.

(4)setAssertion() : Assertion을 설정한다.

(5)getContentURI() : RegistryEntry로부터 콘텐츠의 URI를 얻어온다.

(6)createXACMLDecisionRequest() : Assertion과 URI로 접근권한평가 요청을 생성한다.

(7)setXACMLDecisionRequest() : XACML Manager로 접근권한평가를 요청한다.

(8)getXACMLDecisionResult() : XACML Manager로 접근권한평가 결과를 응답받는다.

(9)analyzeResult() : 응답받은 접근권한평가를 분석한다.

(10)summitObjectRequest() : analyzeResult() 메소드로 분석한 접근권한평가를 바탕으로 콘텐츠의 변경을 요청한다.

(11)summitObjectResponse() : 변경 성공 여부에 대한 결과를 ObjectManager로부터 받는다.

(12)setCPPModifyResult() : 변경된 결과를 MSH로 전송한다.

(13)sign() : 변경된 메시지에 서명을 한다.

(14)encryptionResponse() : 서명된 메시지를 암호화한다.

(15)cppModifyResponse() : 암호화한 메시지를 ebXML Client로 전송하여 응답한다.

전체적인 ebXML Registry/Repository 기능은 client로부터 받은 요청을 암·복호화와, 서명검증과 접근제어평가에 따른 CPP의 변경작업을 수행한다.

데이터의 흐름은 시퀀스 다이어그램에서 보인 것과 같고 각 클래스는 그림3과 같고 역할은 다음과 같다.

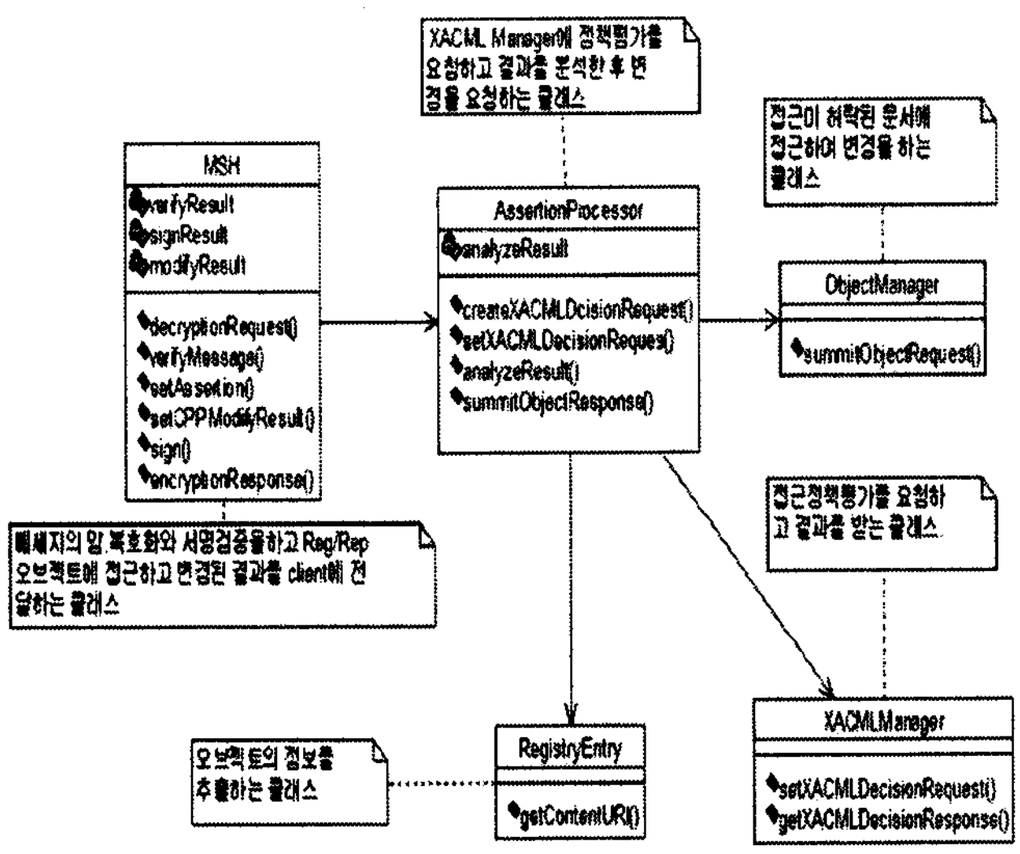

그림 3 ebXML Registry/Repository Class Diagram

• MSH : client로부터 온 암호화되어 온 메시지를 decryptionRequest()메소드로 복호화를 하고 verifyMessage() 메소드로 전자서명을 검증하고 AssertionProcessor의 setAssertion()메소드를 호출해서 assertion을 설정한다. 그리고 Update된 결과를 setCPPModifyResult() 메소드를 통해 AssertionProcessor로부터 넘겨받아 sign()메소드로서명을 하고 encryptionResponse()메소드로 암호화를 한 후 이 암호화한 메시지를 cppModifyResponse()메소드를 사용하여 client로 전송한다.

• ObjectManager : 이 클래스는 XACML을 통한 접근제어평가를 받고 AssertionProcessor로 분석한 결과를 summitObjectRequest()메소드로 레지스트리에 저장된 문서의 업데이트를 수행한다. 그리고 수행한 결과를 summitObjectResponse()메소드를 호출해서 AssertionProcessor로 전달한다.

• AssertionProcessor : getContentURI() 메소드를 호출하여 RegistryEntry로부터 콘텐츠 URI를 읽어온다. 그리고 이 콘텐츠에 대한 변경과 같은 작업을 사용자로부터 요청 받으면 createXACMLDecisionRequest()메소드를 사용하여 접근권한평가를 위한 요청을 생성하고 setXACMLDecisionRequest() 메소드를 호출하여 XACML Manager로 이 생성된 요청을 전송하여 결과를 getXACMLDecisionResult()메소드로 응답 받아 analyzeResult()메소드로 분석한 후 ObjectManager로 변경요청을 한다.

• RegistryEntry : RegistryEntry는 실제 컨텐츠의 메타정보를 추출하는 클래스이다. AssetionProcessor로부터 호출된 getContentURI()메소드로 컨텐츠의 메타정보를 전송한다.

4. 결론 및 향후 과제

등록기/저장소와 접근 제어 기술인 XACML의

접목은 CPP/CPA같은 ebXML자원의 접근을 제어하며 접근자의 특징적인 행위에 대한 정책을 수립하고 실행함으로써 효율적인 자원 관리를 할 수 있는 접근제어를 할 수 있다. 본 논문은 현재 개념만 나와 있고 실제적인 제품이 거의 없는 자원 접근 시스템에서 등록기/저장소와 XACML을 접목하여 자원 접근에 대한 보안시스템을 설계하였다. 향후 과제로 본 논문에서 설계한 접근제어 시스템을 구현하여 시스템의 효율성을 검증하는 것이다.

5. 참고문헌

[1] XML-Signature Syntax and Processing, http://www.w3.org/TR/xmldsig-core/

[2] XML Protocol working Group, http://www.w3c.org/2000/xp/Group/

[3] XML Encryption WG, http://www.w3.org/TR/xkms2/

[4] Security Assertion Markup Language, http://www.oasis-open.org/committees/security/docs/cs-sstc-core-01.pdf

[5] XML Key Management Specification (XKMS), http://www.w3.org/TR/xkms/

[6] OASIS/ebXML Registry Information Model v 2.0, http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebRIM.pdf

[7]OASIS/ebXML Registry Services Specification, v2.0, December 6, 2001, http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrs.pdf

[8] XML Path Language (XPath), http://www.w3.org/TR/xpath

[9]OASIS eXtensible Access Control Markup Language (XACML), http://www.oasis-open.org/committees/xacml/