제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

# SAML 기반의 SSO 및 B2B

진승헌*, 최대선*, 한근희**, 이승민***, 정태의***
*한국전자통신연구원
**공주대학교 응용수학과
***서경대학교 컴퓨터과학과
e-mail : tejeong@skuniv.ac.kr

# A SAML-based SSO and B2B

Jin-Seung Hyen*, Dae-Seon Choi*, Keun-Hee Han**,
Seung-Min Lee***, Tae-Eui Jeong***
*Electronics and Telecommunications Research Institute
**Dept. of Applied Mathematics, Kong-Ju University
*** Dept. of Computer Science, Seo-Kyeong University

요 약

일반적으로 Web Service 란 표준 인터넷 프로토콜을 이용하여 외부에 노출된 비즈니스 기능을 프로그램적으로 접근하는 방식으로 간단한 메소드 호출뿐만 아니라 복잡한 비즈니스 프로세스까지 수행이 가능하고, 한번 배포된 Web Service 는 인터넷으로 접근할 수 있는 곳이면 어디서든지 접근 및 호출이 가능하다. 현재 Web Service 구현에 있어 최대의 당면 과제는 보안문제로서. 인터넷을 이용해서 이동하고 있는 많은 양의 데이터를 안전하게 지킬 수 있는 방법의 모색에 초점이 맞추어져 있다. 이러한 보안 문제를 해결하기 위해 SAML(Security Assertion Markup Language), XKMS(XML Key Management Specification), XAML(Transaction Authority Markup Language), BTP(Business Transaction Protocol), XLANG 등을 이용한 여러 가지 방법이 시도되고 있다. 본 논문에서는 SAML 을 이용한 보안 솔루션을 분석하고, SAML 을 이용하여 SSO, Back Office Transaction 등의 어플리케이션을 구축하여 시뮬레이션 결과를 보이고자 한다.

## 1. 서론

일반적으로 Web Service 란 표준 인터넷 프로토콜을 이용하여 외부에 노출된 비즈니스 기능을 프로그램적으로 접근하는 방식이다. Web Service 는 간단한 메소드 호출뿐만 아니라 복잡한 비즈니스 프로세스까지 수행이 가능하고, 한번 배포된 Web Service 는 인터넷으로 접근할 수 있는 곳이면 어디서든지 접근 및 호출이 가능하다. Web Service 는 컴포넌트 기반 개발과 Web 을 통합한 솔루션을 제시한다. 컴포넌트와 같이 Web Service 는 내부의 표현 방식을 외부에 숨겨서 내부의 변경 및 수정이 외부 클라이언트에 영향을 주지 않도록 만들어지며, 웹과 같이 표준 인터넷 프로토콜과 표준 포맷으로 접근이 가능하다. 또한 Web Service 는 e-business 어플리케이션 개발의 새로운 패러다임으로 모든 것을 자체적으로 포함하며 (Self-contained), 자체 기술적이며(self-describing), 모듈화 되어있어 Web 을 통해 공개되고 위치를 찾고 호출될 수 있다. 이러한 Web Service 구현에 있어서 가장 큰 이슈가 되는 것은 보안 문제이다. 인터넷을 이용해서 이동하고 있는 많은 양의 데이터를 안전하게 지킬 수 있는 방법의 모색에 초점이 맞추어져 있다. 보안 문제를 해결하기 위해 여러 가지 방법이 시도되고 있으며, SAML(Security Assertion Markup Language), XKMS(XML Key Management Specification), XAML(Transaction Authority Markup Language), BTP(Business Transaction Protocol), XLANG 등을 예로 들 수 있다. 본 논문에서는 SAML 을 이용한 보안 솔루션을 분석하고 SAML 을 이용한 어플리케이션을 시뮬레이션 하고자 한다.

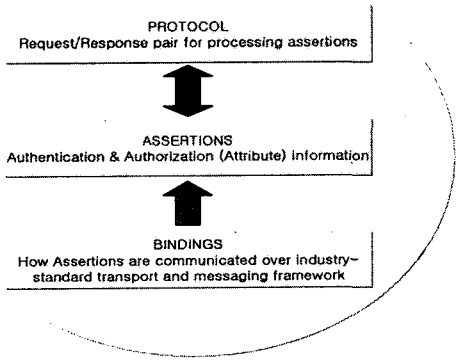
본 논문의 구성은 다음과 같다. 먼저 2 장에서 SAML 의 구조와 내용 분석을 기술하였고, 3 장에서는 SSO 와 B2B 의 흐름을 보았다. 4 장에서는 Simulation 을 기술하였고, 마지막으로 5 장에서 결론을 맺었다.

## 2. 관련 연구

SAML (Security Assertion Markup Language)은 인터넷 상의 비즈니스 보안 정보 교환용 XML[1]기반의 표준으로 다른 시스템 간의 보안 서비스 상호 운용이 가능하고 XML 로 된 정보를 기술하는 공통 언어이다. 웹 상의 거래가 B2C, B2B 등으로 광범위해지고 거래 시작 사이트와 거래 종료 사이트가 달라지므로 다양한 거래를 공유할 수 있는 보안 정보가 요구된다. 따라서 공통 언어로서 상호 운용성과 각종 프로토콜과의 호환성을 갖춘 개방 솔루션 및 자원 접근을 용이하게 하는 SSO(Single Sign On)기능 들을 제공하고 있다. SAML 은 OASIS XML 보안 서비스 기술 위원회 (OASIS XML – Security Service Technical Committee)에서 표준화 작업을 진행하고 있다. Browser-driven interaction, XML Message transfer, Remote authorization 의 use case 를 기본으로 한다. Browser 기반의 SSO 환경에서는 사용자의 디렉토리 중복이나 동시성이 요구되지 않는다. 보안 정보를 가지고 이동하게 된다. 결과적으로 소스 web site 에서 온 사용자는 destination web site 에서 다시 등록절차를 거치지 않아도 되는 것이다. 그래서 사용자 정보는 e-business 네트워크에서 복잡하게 중복되지 않아도 된다. 또한 XML 메시지 전송 환경에서 SAML 은 XML digital signature[2]를 기반으로 하는 authentication 과 attribute 를 기본으로 하는 authorization 을 제공한다.

### 2.1 SAML 의 구조

SAML[3]은 assertions, protocol, bindings 의 3 가지 부분으로 구분된다. Protocol 부분은 assertion 을 수행하기 위해 request 및 response 의 쌍으로 이루어 진다. Assertion 은 authentication 과 authorization 에 관련된 정보를 포함하고 있다.




<그림 1> SAML 의 구성

### 2.2 Assertion

Assertion 은 크게 3 가지로 나눌 수 있다. Authentication assertion 은 issuer(속성보증서의 issuer)에 의하여 subject 가 특정 시간에 특정 방법에 의해 인증되었다는 것을 확인 시켜준다. Authorization decision assertion 은 subject 가 특정한 객체(object)에 접근을 요청하였을 때 이러한 요청이 제공된 증거에 의하여 어떠한 결과를 낳았는지에 대해서 결정된 상태를 나타낸다. Attribute assertion 은 해당 subject 가 표시된 속성들을 가지고 있음을 확인시킨다.

이런 assertion 들은 element 들의 집합으로 이루어져 있다. Element 는 simple type 과 complex type 이 있는데 단순히 element 하나로 이루어지는 것을 simple type, 다른 Element 를 포함하고 참조하는 것을 complex type 이라고 한다. Simple type 은 <IDType>, <IDReference_Type>, <Assertion>, <Statement>, <SubjectStatement>등이 있다. 세부 특성을 보면 <IDType>은 assertion, request 및 responses 등에 대한 identifier 부여하고, 중복되지 않아야 한다. 동일할 확률 $2^{-128}$ 보다 작아야 한다고 정의 되어 있지만 $2^{-168}$.이 기본 확률로서 사용된다. Identifier 를 부여하는 선언과정은 재실행 없이 단 한 번만 이루어 진다. <IDReferenceType>은 <IDType>의 identifiers 를 참조하는데 사용하고, <DecisionType>은 authentication decision statement 의 상태로써 report 될 수 있는 값으로 정의한다. Decision 은 Permit, Deny, Indeterminate 의 세가지 상태를 갖는다. <Assertion>은 complex type 으로 attribute 로는 MajorVersion, MinorVersion, AssertionID, Issuer 및 IssuerInstance 를 가진다. <Condition>은 new conditions 에 대한 extension point 로써의 역할을 수행한다. <Advice>는 issuer 가 제공하고자 하는 추가정보 포함할 수 있어서 주로 주석을 내용으로 한다. <Statement> element 는 다른 application 의 JSAML assertion 기능을 재사용하며, complex type 으로 많은 element 들을 포함할 수 있다. <SubjectStatement>는 statement 를 subject 로 구분한다. <SubjectConfirmation>은 authenticate 된 subject 를 허가하는 데이터를 제공한 subject 의 정보를 가진 attribute 를 포함한다. <ConfirmationMethod>는 subject 를 인증하는데 사용되는 인증 프로토콜을 가리키는 URI 를 포함하며, <SubjectConfirmationData>는 특정한 인증 프로토콜에 의하여 사용될 수 있는 부가적인 정보를 포함한다. <ds:KeyInfo>는 subject 를 보관하고 있는 암호 키를 지정하는 XML Signature[XMLSig]를 가지고 있다. 어떤 종류의 statement 가 포함되는가에 따라서 assertion 의 형태가 달라지는데 각 종류의 내용을 살펴보면 다음과 같다. <AuthenticationStatement>는 특정한 시간에 특정한 방법에 의하여 subject 의 신원이 issuer 에 의해 검증되었다는 것을 확인하는 역할을 하며, 포함하고 있는 attribute 는 SubjectLocality 와 AuthorityBinding 이다. SubjectLocality 는 authenticate 된 subject 에서 비롯된 system entity 에 대한 DNS domain 이름과 IP address 를 정의하고 있다. AuthorityBinding 은 authentication _Statement 를 받은 신용할 수 있는 party 를 지칭한다. <AuthorizationDecisionStatement>는 접근 허용 여부 판단한 결과를 나타내는데, attribute 로 action 과 evidence 를 갖는다. Action 은 subject 가 요청하는 시스템 사용 방법을 포함하고, evidence 는 접근 권한 부여 여부의 판단에 사용된 assertion 을 포함한

다. 마지막으로 <AttributeStatement>는 명시된 subject 가 명시된 attribute 들을 가지고 있음을 보여주는데 AttributeDesignator 와 Attribute 를 attribute 로 가진다. AttributeDesignator 는 attributeNamespace 에서 attribute 이름을 구분하고, Attribute 는 attribute value 를 포함한다.

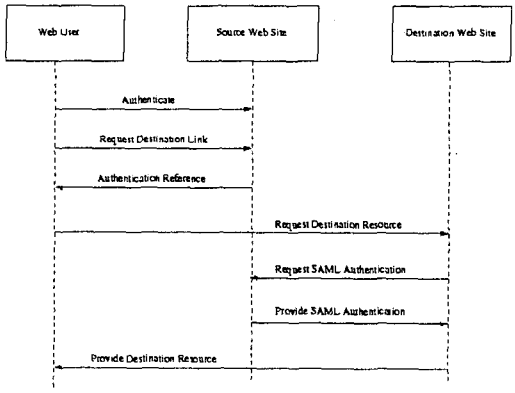
### 2.3 Protocol

SAML Protocol 은 request 와 response 두 개의 쌍으로 이루어 진다. PEP (Policy-enforcement Point)와 PDP (Policy-Decision Point)의 사이, authentication authority 와 client program 의 사이, 그리고 attribute authority 와 client program 사이에서 사용된다. <Request>는 authentication, attribute, 그리고 authorization queries 를 attribute 로 포함하고 있고, 각 request 는 공용되는 response 에 각각 매치된다. Protocol 들도 assertion 과 동일하게 여러 element 들로 구성된다. <RequestAbstractType>, <Request>, <ResponseAbstract _Type>, <Response>, <Status>등을 element 로 가진다. <RequestAbstractType>은 RequestID, MajorVersion, MinorVersion, IssueInstant 를 attribute 로 가진다. Request 에서 기본적으로 header 처럼 가지게 되는 형식이며 그 뒤에 어떤 <Request> element 가 붙는지에 따라 해당 request 로 바뀐다. 뒤에 올 수 있는 element 는 <Query>, <SubjectQuery>, <Authentication _Query>, <AttributeQuery>, <AuthorizationDecisionQuery>, <AssertionIDReference>, <AssertionArtifact> 등이 있다. Response 도 이와 유사한 형태를 가지는데, ResponseID, InResponseTo, MajorVersion, MinorVersion, IssueInstant, Recipient 를 attribute 로 가진 <ResponseAbstractType>이 있고, 그 뒤에 <Status>가 붙는다. <StatusCode>, <StatusMessage>, <StatusDetail>의 내용을 포함해서 상태를 표현한다. 각각의 response 는 해당 subjcet 의 element 와 매칭이 되게 설계 되어진다.

## 3. Application

### 3.1 SSO (Single Sign-On)

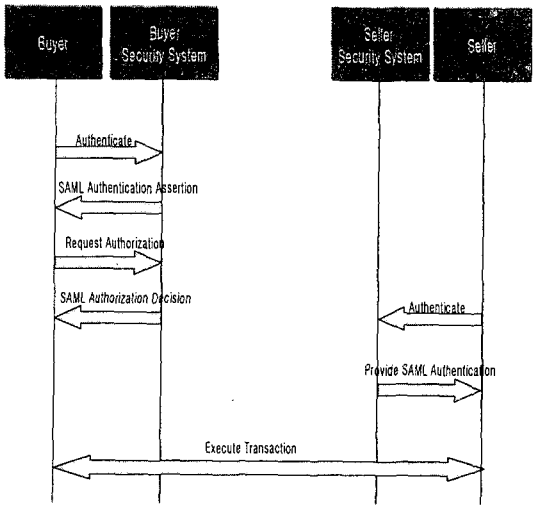
SSO 는 단 한번의 로그인 만으로 기업의 각종 시스템이나 인터넷 서비스에 접속하게 해주는 보안 응용 솔루션이다. 각각의 시스템마다 인증 절차를 밟지 않고도 1 개의 계정만으로 다양한 시스템에 접근할 수 있어 ID, 비밀번호에 대한 보안 위험 예방과 사용자 편의 증진, 인증 관리비용의 절감 효과가 있다. 클라이언트 SSL 인증서와 S/MIME 인증서가 포함된 SSO 솔루션으로 개인 키 데이터베이스에 있는 하나의 키로 로그인하고, 다른 비밀번호 없이 SSL 사용 서버에 접근할 수 있다. SAML 은 이런 SSO 를 기본으로 하고 있다. <그림 2>는 SSO 의 흐름을 나타내고 있다.



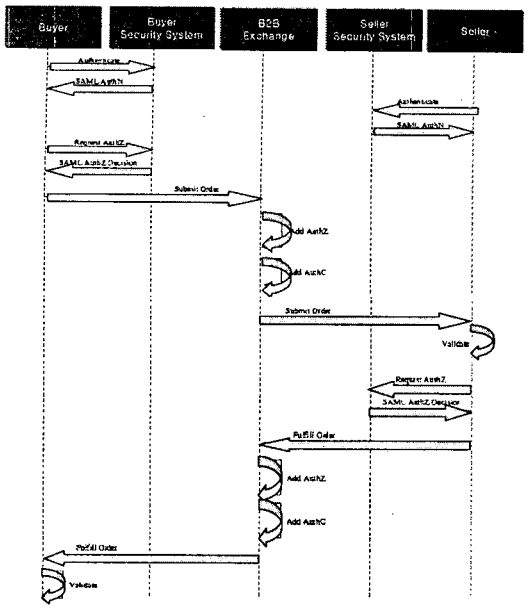

<그림 2> SSO 흐름

### 3.2 B2B

모든 B2B 는 SSO 를 기본 구조로 한다. Buyer 와 seller 에 각각 security system 이 있는 Back Office Transaction, 각각의 security system 이 합쳐진 Back Office Transaction with Third Party Security, 각각의 Security System 이 있고 그 사이에 서로 교환전송 할 수 있게 한 Intermediary Add 등으로 나눌 수 있다. Back Office Transaction 들은 비교적 간단하게 이해 할 수 있지만, Intermediary Add 같은 경우에는 B2B Exchange 서버가 Buyer 와 Seller 사이에서 오가는 주문이나 확인을 중간에서 받아서 다시 확인해주는 절차를 거치기 때문에 다른 종류에 비해 복잡하다. <그림 3>과 <그림 4>는 B2B 종류들의 흐름을 그림으로 나타낸 것이다.



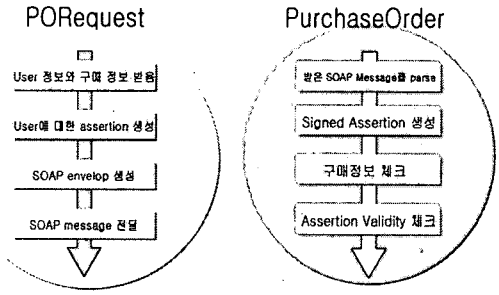

<그림 3> Back Office Transaction




<그림 4> **Intermediary Add**

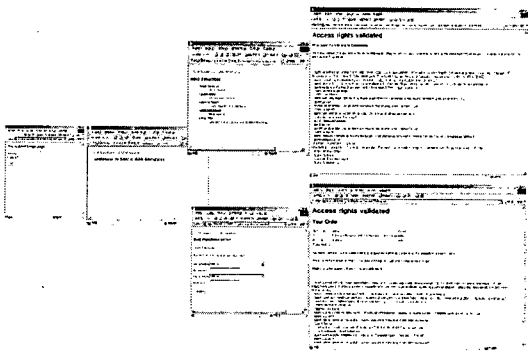
## 4. Simulation

SAML 을 바탕으로 하여 SSO 와 B2B 를 구현하였다. Netegrity 사의 JSAML-Toolkit[4]을 기반으로 하여 Apache Tomcat 4.0 으로 web server 를 서블릿을 사용하였다. 로그 파일 분석으로 세부적인 서블릿의 흐름을 보면 주문을 받은 상태에서부터 해당 서블릿이 동작을 하며, user 의 assertion 생성이 만들어지고 그 assertion 을 SOAP(Simple Object Access Protocol)을 이용하여 전송하게 된다. 그 메시지를 받은 서블릿은 SOAP message 로부터 assertion 과 구매 내용을 추출하여 validity 를 체크하고 그에 대한 응답을 보내준다. 주문에 대해 validity 가 제대로 되면 거기에 해당되는 응답을 해주게 된다. 다음 <그림 5>은 그에 해당되는 서블릿의 흐름을 보여주고 있다.




<그림 5> **Servlet 흐름도**

<그림 6> 은 시뮬레이션 된 web page 화면이다. SSO 와 B2B 를 보여준다. SSO 를 보면 최초의 로그인 한 번으로 링크되어 있는 다른 사이트로 이동할 수 있다. 해당 링크별로 레벨이 있으며 로그인시에 레벨에 맞게 주어진 권한을 이용하여 링크된 웹사이트로 이동하게 된다. B2B 의 경우에는 주문을 할 수 있게 폼이 주어지는데 주어진 폼에서 주문을 하게 되면 <그림 5>에서 보여진 해당 서블릿들이 동작을 하게 되면서 동작이 이루어지며, 결과를 웹페이지 형태로 나타내게 된다.



<그림 6> **Simulation 화면**

## 5. 결론

현재 web service 구현에 있어 최대의 당면 과제는 보안문제로서. 인터넷을 이용해서 이동하고 있는 많은 양의 데이터를 안전하게 지킬 수 있는 방법의 모색에 초점이 맞추어져 있다. 이러한 보안 문제를 해결하기 위해 SAML(Security Assertion Markup Language), XKMS(XML Key Management Specification), XAML (Transaction Authority Markup Language), BTP(Business Transaction Protocol), XLANG 등을 이용한 여러 가지 방법이 시도되고 있다. 본 논문에서는 SAML 과 이를 이용한 보안 솔루션을 분석하였고, SAML 을 이용하여 SSO, Back Office Transaction 등의 어플리케이션을 구축하여 그 세부 내용을 시뮬레이션 하였다.

## 참고문헌

[1] Time Moses, et.al., "Security Assertions Markup Language"
[2] D. Eastlake et at., "XML-Signature Syntax and Processing", World Wide Web Consortium.
[3] "Oasis Security Services Use Cases and Requiredments", draft-sstc-saml-reqs-01, 2001.
[4] "JSAML Toolkit", Netegrity White paper, 2001, http://www.netegrity.com