情報保護學會論文誌
第18卷 第4號, 2008. 8

안전한 웹서비스를 위한 멀티 에이전트 기반의 확장된 SAML 위임 모델*

김 규 일†, 원 동 호, 김 응 모‡
성균관대학교

An Extended SAML Delegation Model Based on Multi-Agent for Secure Web Services*

Kyu Il Kim†, Dong Ho Won, Ung Mo Kim‡
SungKyunKwan University

요 약

웹 서비스는 네트워크상에서 서로 다른 종류의 컴퓨터들 간에 상호운용을 지원하기 위한 시스템이고 서비스 지향적 분산 컴퓨팅 기술의 일종이다. 이러한 환경에서 보안은 가장 중요한 문제 중 하나이다. 공격자는 아무런 인증 없이 사용자의 비밀정보를 노출 시킬 수 있다. 또한 사용자들은 웹 서비스를 이용하기 위해 반드시 그들 대신 서비스를 처리할 에이전트에게 그들의 권한 모두를 혹은 그 중 일부분을 일시적으로 위임해야만 한다. 이것은 사용자의 비밀정보가 에이전트를 통해 외부에 노출되는 결과를 초래한다. 우리는 에이전트를 통한 비밀정보의 노출을 막고 서비스의 기밀성과 단언정보의 무결성을 제공하기 위해 XML 암호화와 XML 전자서명 방식을 이용한다. 그리고 XACML 기반의 웹 서비스 관리 서버를 통해 웹 서비스 제공자들과의 서비스정책의 상호운용을 수행한다. 우리는 역시 멀티 에이전트들 간의 위임을 통해 웹 서비스 제공자들에게 전달될 위임 단언을 정의하기 위해 SAML을 확장한다.

ABSTRACT

Web service is defined to support interoperable machine to machine interaction over a network and defined as distributed technologies. Recently in web service environment, security has become one of the most critical issues. An attacker may expose user privacy and service information without authentication. Furthermore, the users of web services must temporarily delegate some or all of their behalf. This results in the exposure of user privacy information by agents. We propose a delegation model for providing safety of web service and user privacy in ubiquitous computing environments. In order to provide safety of web service and user privacy, XML-based encryption and a digital signature mechanism need to be efficiently integrated. In this paper, we propose web service management server based on XACML, in order to manage services and policies of web service providers. For this purpose, we extend SAML to declare delegation assertions transferred to web service providers by delegation among agents.

Keywords : SAML, XACML, Agent

Ⅰ. 서 론

웹 서비스는 XML을 기반으로 SOAP, UDDI, WSDL 등의 공개 표준을 이용하여 B2B(Business to Business)를 쉽게 통합하고, B2C(Business to Consumer)간의 상호작용을 제공하기 위하여 고안되었다. 웹 서비스 제공자(Web Service Provider)들은 자신들의 서비스를 WSDL(Web Service Description Language)[17]을 이용하여 UDDI(Universal Description, Discovery and Integration) 레지스트리에 등록할 수 있고, 사용자들은 자신의 조건에 맞는 서비스를 찾고, 이용할 수 있다. 이러한 분산 환경에서 사용자들은 웹 서비스를 검색하고 이용하기 위하여 멀티 에이전트를 이용한다. 멀티 에이전트들은 사용자의 자격(Credential)정보를 전달하고, 사용자로부터 받은 정보를 가지고 사용자 대신 웹 서비스를 수행한다. 그리고 효율적인 웹 서비스를 제공하기 위하여 에이전트들은 자신의 업무를 다른 에이전트들에게 위임한다. 그러나 사용자는 웹서비스 사용권한을 얻거나, 웹 서비스를 수행하기 위해 자신의 비밀(Privacy)정보와 서비스 정보를 에이전트에게 노출 시켜야 하고, 이는 사용자의 비밀정보가 외부에 노출 될 수 있다는 결과를 초래한다. 우리는 사용자의 비밀을 보호하고 서비스의 기밀성과 무결성을 제공하기 위해서 XML 관련 보안기술들인 XML 전자서명과 XML 암호화 방식[14]를 사용한다. 그리고 에이전트를 통해 웹 서비스 제공자들에게 위임단언을 전달하기 위해서 SAML(Security Assertion Markup Language)을 확장한다. 또한 웹 서비스 제공자들과 상호운용하며, 인증된 사용자들에게 적합한 웹 서비스를 찾아 이용할 수 있는 역할(Role)을 부여하기 위해 XACML(eXtensible Access Control Markup Language) 기반의 웹 서비스 관리 서버를 사용한다. SAML과 XACML은 OASIS 표준으로 싱글 사인 온(Single Sign On)[15,16], 신뢰(Trust)관리, 인증, 권한부여 등의 영역을 포함하고 있으며, 웹 서비스 뿐만 아니라 다른 영역에서도 사용될 수 있다. 우리는 OASIS의 공개 표준인 SAML을 확장하여 정의하는 것을 특징으로 안전한 웹 서비스를 위한 모델을 제안한다.

접수일 : 2008년 1월 23일; 수정일 : 2008년 4월 11일;
채택일 : 2008년 5월 28일
* 본 연구는 정보통신부 및 정보통신 연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행하였습니다.(IITA-2008-C1090-0801-0028) 또한 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스 컴퓨팅 및 네트워크 원천기반 기술개발사업의 지원에 의해 수행하였습니다.
† 주저자, kisado@ece.skku.ac.kr
‡ 교신저자, umkim@ece.skku.ac.kr

Ⅱ. 관련 연구

2.1 SAML(Security Assertion Markup Language)

SAML[1]은 도메인 간에 사용자 정보를 안전하게 교환하기 위해 만들어진 OASIS 표준 확장 언어로 정보 수신, 전송 및 공유와 관련된 모든 기능을 제공한다. 이전에는 대부분 단일 시스템이 액세스 제어 결정에 필요한 모든 정보를 소유하고 감사 추적에 기록할 모든 데이터를 가지고 있다는 가정 하에 시스템을 설계하였다. 그러나 대규모 분산 시스템은 항상 다양한 제품을 사용하는 여러 조직에 의해 구축된다. 이 경우 사용자가 다른 방법을 통해 다른 인증기관에서 인증을 받을 수도 있다. 또한 서로 다른 인증기관은 사용자 속성 및 특성에 대해 서로 다른 정보를 보유하게 된다. 이러한 모든 기능과 정보를 중앙에서 관리하는 것이 반드시 실용적이지 않다. SAML은 인증 및 사용자 속성을 표시하기 위해 표준 포맷을 제공하고 이를 요청하고 수신하기 위한 프로토콜을 제공한다. SAML은 단언(Assertion)을 사용하여 웹 SSO를 제공할 수 있는 방법을 정의하였다. 단언은 신뢰할 수 있는 제 3의 기관에 의해 작성된 것으로 시스템이나 애플리케이션 간에 교환되는 정보를 뜻하고 인증 및 승인에 관련된 정보가 XML로 인코딩되어 있다. 단언은 다음과 같이 세 가지 다른 형태로 단언들을 정의한다.

- 인증(AuthnStatment) : 주체(Subject)가 특정 시간 및 장소에서 특정 방법(패스워드, 하드웨어 토큰, X.509 공개키) 등으로 인증되었다는 것을 나타낸다.
- 인가(AuthzDecisionStatment) : 주체가 자원에 접근할 수 있도록 권한 부여 및 거부 되었다는 것을 나타낸다.
- 속성(AttributeStatment) : 주체가 주어진 속성(Attribute)/속성 값(Attribute Value)과 연관되어 있다는 것을 나타낸다.

SAML은 분산 환경에서 통합 ID 관리를 구현하기 위한 유용한 매커니즘을 제공하고 일반적인 여러 가지

시나리오를 상세하게 지정하여 뛰어난 상호 운용성을 제공한다. 또한 고유 요구 사항 및 향후 발생하게 될 요구를 처리할 수 있도록 확장이 가능하다.

2.2 XACML(eXtensible Access Control Markup Language)

XACML[2,11] 역시 OASIS 표준으로 엑세스 제어 정책을 표시하기 위한 언어이다. 대부분의 컴퓨터 전문가들은 사용 권한 또는 엑세스 제어 목록(ACL)에 기반을 둔 엑세스 제어에 익숙하다. 하지만 이러한 매커니즘은 실제 시스템에서 요구되는 복잡한 정책을 표시할 수 있는 기능이 부족하다. 그 결과 엑세스 제어 정책이 애플리케이션 코드에 포함되기도 한다. 그럴 경우 정책을 변경하거나 심지어 실행 중인 정책을 찾는 것조차 매우 어려워진다. XACML은 실제로 이용할 수 있는 모든 정보를 사용하여 리소스에 대한엑세스를 허용할 것인지 결정할 수 있다. 또한 이러한 결정에 요청된 데이터를 90일 이후 제거하라는 것과 같은 추가 의무를 결합할 수 있다. XACML은 리소스 내용과 같은 리소스의 속성이나 날짜, 시간, 위치와 같은 환경적 요소를 바탕으로 결정을 내릴 수 있고 또한 역할 또는 그룹 구성원과 같이 요청과 관련된 측정의 속성을 고려할 수 있다. 여기에는 요청을 하는 측뿐만 아니라 데이터 또는 요청에 대한 매개물을 수신하는 측도 포함될 수 있다.

XACML 2.0은 2005년 2월에 OASIS 표준으로 승인되었으며, 정책을 만드는 관리자가 여럿인 대규모 환경에 접합하다. SAML이 다른 엑세스 제어 시스템과 함께 작동하는 것과 같이 XACML을 SAML과 함께 사용하거나 SAML 없이 사용할 수도 있지만 특정 기능을 함께 작동해야 사용할 수 있도록 지정되어 있다.

하지만 현재 위의 기술을 적용한 연구를 보면 다음과 같은 문제점을 지니고 있다. Navarro, et al[4]에서는 권한 위임을 제공하기 위해서 SAML의 SubjectStatment 엘리먼트를 SubjectDelegation Statement로 재정 하였다. 그러나 이것은 SAML 1.1/2.0 Assertion specifications에서 지원되지 않기 때문에 유연성(Flexibility)의 문제가 있었다. V.Welch[5]는 위임을 제공하기 위하여 X.509 인증서(Certificate)를 확장시킨 Proxy X.509 인증서를 정의하며 이 방법은 Globus 프로젝트[6]에서 이용되고 있다. Proxy X.509 인증서는 상당히 유용한 방식이나 웹 서비스를 위한 상용화 툴들은 이러한 인증서들을 올바르게 인식하지 못한다. J. Y. Hu[7]는 PKI(Public Key Infrastructure)를 기반으로 하여 에이전트 시스템을 어떻게 구성하는가를 기술하였다. 위임 관련해서 사슬규칙(Chain-ruled)위임, 임계치(Threshold)위임, 조건부(Conditional) 위임을 고려하였다. 그러나 이러한 방법은 모바일 에이전트 시스템이나 이기종의 멀티 에이전트 시스템에는 적용하기 어렵다.

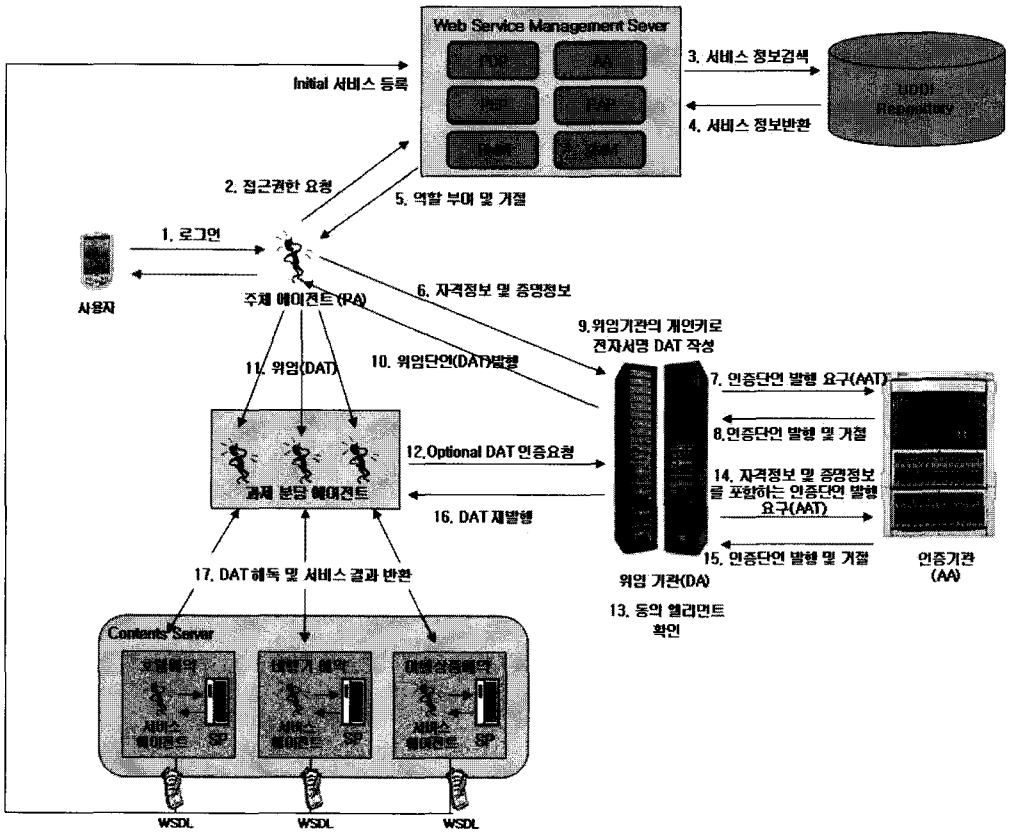
Wang, Del Vecchio[9]은 권한 위임을 제공하기 위해서 SAML을 확장하였다. 그들은 SAML AttributeStatement 엘리먼트에 권한 위임에 관한 정보를 표현하기 위해 Attribute 엘리먼트들을 추가하여 위임단언을 선언하였다. 그리고 위임단언에 위임자(delegator)의 X.509V3 서명정보를 포함하여 SAML 토큰을 선언하고, SAML 토큰을 WSS(Web Service-Security) SOAP (Simple Object Access Protocol)[19] 헤더에 추가하여 웹 서비스와 그리드 서비스들 간의 직/간접적(Direct/Indirect)으로 권한을 위임하고 검증하는 방법을 제안하였다. 그러나 제안된 방법은 간접적(Indirect)인 권한위임의 경우 위임의 횟수만큼 SAML 토큰의 개수도 많아지기 때문에 위임을 검증하기 위한 시간복잡도(Time Complexity)가 증가한다. 따라서 본 논문은 웹 서비스를 이용하는 데 있어서 사용자의 프라이버시 정보를 보호하면서 사용자에 의해 관리될 수 있는 에이전트들 간의 권한 위임모델을 제안한다.

Ⅲ. 에이전트를 적용한 위임 모델

3.1 멀티에이전트를 이용한 위임 모델

[그림 1]은 웹서비스에서 에이전트를 기반으로 안전한 웹서비스를 지원하기 위한 위임모델이다. 위임 모델에서 가장 우선시 정의해야 할 사항으로 첫째, 웹 서비스 관리서버와 웹 서비스 제공자들은 인증/위임 기관과 신뢰관계(Trust Relationship)를 갖는다. 신뢰를 통해 인증/위임기관이 인증/인가한 사용자/에이전트들을 신뢰할 수 있다. 둘째, 사용자는 인증기관에 미리 등록되어 있고, 웹 서비스관리 서버에 계정을 갖고 있다. 따라서 웹 서비스 사용자로써의 역할(Role)을 가지고 있다. 셋째, 웹 서비스 관리 서버가 인가된 사용자들에게 부여하는 역할은 사용자의 검색조건에 맞는 웹 서비스 제공자들의 역할과 공유되는 역할이다.

이들 세 가지 요소를 전제로 제안 모델은 사용자가 자



〔그림 1〕 멀티에이전트를 이용한 위임 모델

기 자신의 특권(Privilege)의 위임을 관리할 수 있다. 에이전트들은 사용자와 에이전트를 식별하여 인증(Authentication)하는 인증기관(Authentication Authority)과 위임 받고자 하는 에이전트 위임을 인가하는 위임기관(Delegation Authority)의 도움을 받아 사용자에 의해 위임 받은 특권에 따라 서비스 이용을 제어한다.

위임모델에서 사용되는 기본적인 컴포넌트를 설명하면, 웹 서비스 관리서버(WSMS)는 사용자와 서비스 제공자를 관리한다. 웹 서비스 제공자들과 상호작용하고 주체의 요청을 처리하는 시스템이다. 주체(Principal)는 역할을 가지고, 웹 서비스를 수행하기 위해 자신의 특권을 에이전트들에게 일시적으로 위임하는 사용자이다. 역할은 웹 서비스 관리서버에게 특정 조건들과 주체의 자격정보와 함께 웹 서비스 요청을 함으로써 획득된다. 주체 에이전트(Principal agent Pa)는 주체와 상호작용하고 주체대신 다른 에이전트들과 상호 작용하는 소프트웨어이다. 과제분담 에이전트(Carrier(Task) agent Ca)는 그것이 오직 위임단언을 전달하는 것만 제외하고 주체 에이전트와 유사하고 서비스 에이전트(Service agent Sa)는 그것이 웹 서비스 제공자의 에이전트인 것만 제외하고 주체 에이전트와 유사하다.

인증기관(AA)은 주체와 에이전트를 인증하는 기관이다. 인증기관은 주체와 에이전트가 인증되었는지 나타내는 인증단언을 발행한다. 위임기관(DA)은 주체의 특권을 에이전트들에게 인가하는 위한 기관이다. 위임기관은 자신이 주체의 권한을 주체/에이전트에서부터 에이전트들에게 인가한다는 것을 보증하는 위임단언을 발행한다. 위임단언을 발행하기 전에 위임기관은 반드시 인증기관을 통해 위임 받을 에이전트의 인증 단언을 획득해야 하고 주체 에이전트에게 위임 단언발행에 대한 동의를 얻어야 한다.

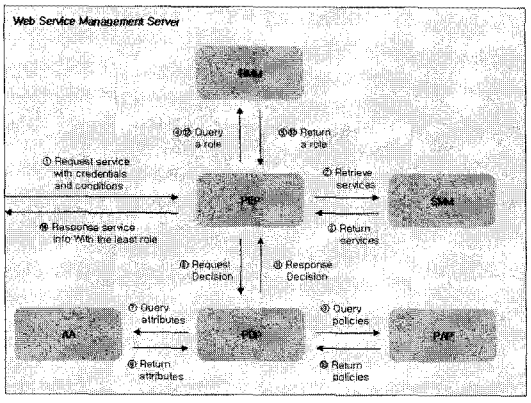
3.2 웹 서비스 관리서버(Web Service Management Server WSMS)

웹 서비스 관리서버(WSMS)은 XACML 모델을 기

반으로 한다. 웹 서비스 관리서버는 웹 서비스 제공자들의 서비스를 등록하고 인증기관에 의해 인증된 주체의 검색조건에 따르는 적합한 서비스를 UDDI(Universal Description Discovery and Integration) 레지스트리[8]에 찾아 역할(Role)을 부여하는 시스템이다. WSMS는 주체들에게 적합한 역할을 부여하기 위해 관련된 서비스 제공자들과 상호 작용하여 정책들을 모으고, 서비스를 이용할 수 있는 최소한의 특권(Least Privilege)을 가진 역할을 부여한다. 역할 부여는 역할 계층(Role Hierarchy)[3,13]을 고려하는데 만약 서비스를 이용할 수 있는 최소한의 특권이 주체가 가진 역할보다 상위 계층이면 역할은 부여되지 않는다. 그리고 부여된 역할(Role)[18]은 특정한 제한시간(Time Constraint)을 가지는데, 제한시간이 지난 역할은 유효성을 상실한다.

[그림 2]는 주체의 요청에 관한 웹 서비스 관리서버의 절차를 나타낸다. WSMS의 모듈은 다음과 같이 구성된다. 정책 시행 모듈(Policy Enforcement Point PEP)은 주체 역할 요청을 받고 관련 모듈과 상호 작용 후 그에 적합한 접근 결정을 시행하는 모듈이다. PEP는 자격정보와 함께 주체의 유효성을 확인하고 웹 서비스 정보를 리턴한다. 그리고 결정 상태의 결정에 의해 적어도 하나의 역할을 할당한다.

정책 결정 모듈(Policy Decision Point PDP)는 주체의 조건 검색에 의한 UDDI 레지스트리로부터 검색된 서비스를 웹 서비스 제공자의 정책을 평가한다. PDP는 주체/웹 서비스 제공자의 역할이 허가인지 아닌지에 대해 결정하고 PEP에게 권한결정을 보낸다. 정책 관리 모듈(Policy Administration Point PAP)은 등록된 웹 서비스 제공자의 정책을 관리하는 모듈이다. 속성인가

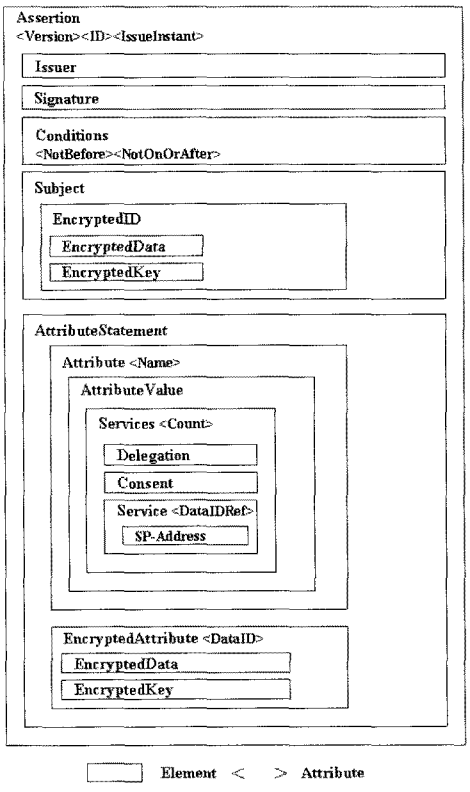


〔그림 2〕 웹 서비스 관리서버(WSMS)

(Attribute Authority AA)는 주체의 신원을 위한 속성을 묶는 모듈이고, 역할 관리 모듈(Role Management Module RMM)은 주체의 역할을 관리하는 모듈이다. RMM에서 주체의 역할은 서비스의 사용을 위한 요구된 역할보다 높아야 하며 적어도 하나의 역할을 할당받아야 한다. 서비스 관리 모듈(Service Management Module SMM)은 WSDL과 함께 웹 서비스 제공자의 서비스를 등록하는 모듈이다. 또한 주체의 검색 조건에 의한 서비스를 검색한다.

3.3 위임 단언 구조

웹 서비스 SAML 2.0 명세서는 위에서 언급했던 것처럼 다음과 같이 세 가지 다른 형태의 단언들을 정의한다. 인증 선언(Authentication Statement)은 주체가 어떤 수단(Password, Hardware Token, X.509 공개키)에 의해 이미 인증 되었다는 것을 의미하고 인가 결정 선언(AuthorizationDecision Statement)은 주체가 접근 요청에 대해 접근이 허가되었는지 거부되었는지를 나타



〔그림 3〕 위임 단언 구조

```
<Assertion ID="_a75adf55-010d-dadsl42-tchadb434d"
          IssueInstant="2008-01-05T02:46:02Z" Version="2.0">
  <Issuer>http://www.delegation-authority.com</issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
     digital signature of delegation authority </ds:Signature>
  <Conditions NotBefore="2008-01-05T02:46:02Z"
              NotOnOrAfter="2008-01-05T02:55:00Z">
  <Subject>
    <EncryptedID>
      <xenc:EncryptedData> ... </xenc:EncryptedData>
      <xenc:EncryptedKey> ... </xenc:EncryptedKey>
    </EncryptedID>
  </Subject>
  <AttributeStatement>
    <Attribute Name="DeleInfo1">
      <AttributeValue>
        <Services count="2">
          <Delegation>true</Delegation>
          <Consent>true</Consent>
          <Service DataIDRef="Serv1">
            <SP-Address>http://www.SP1.com</SP-Address>
          </Service>
          <Service DataIDRef="Serv2">
            <SP-Address>http://www.SP2.com</SP-Address>
          </Service>
        </Services>
      </AttributeValue>
    </Attribute>
    <EncryptedAttribute DataID="Serv1">
      <xenc:EncryptedData> ... </xenc:EncryptedData>
      <xenc:EncryptedKey> ...</xenc:EncryptedKey>
    </EncryptedAttribute>
    <EncryptedAttribute DataID="Serv2">
      <xenc:EncryptedData> ... </xenc:EncryptedData>
      <xenc:EncryptedKey> ... </xenc:EncryptedKey>
    </EncryptedAttribute>
  </AttributeStatement>
</Assertion>
```

〔그림 4〕 위임 단언의 예

낸다. 마지막으로 속성 선언(Attribute Statement)은 대상이 주어진 속성들의 값들과 관련되어 있다는 것을 나타낸다. SAML 명세서는 위임을 위한 스키마는 직접적으로 정의되어 있지 않다. 그러나 다행이 SAML은 스키마의 확장을 허용한다. 위임단언을 생성하기 위해서 이러한 특성을 이용할 수 있다. 본 논문은 위임에 관한 정보를 담기 위해 속성 단언(Attribute Assertion)을 확장한다. [그림 3]에서는 위임단언의 구조를 기술한다.

위임단언 Issuer 엘리먼트는 위임단의 발행자이다. 위임단언 발행은 위임기관에서만 발행하기 때문에 Issuer는 위임기관(DA)라 볼 수 있다. Signature는 위임단언의 무결성을 증명하기 위한 위임기관의 전자서명이다. Conditions는 위임단언의 유효기간을 나타내고 Not-Bofore와 NotOnOrAfter 속성(Attribute)들의 유효 기간을 나타낸다. Subject는 주체로서 이름 식별자(Name identifier)같은 주체의 정보를 포함하고 EncryptedID는 인증기관의 공개키에 의해 암호화된 형태의 이름 식별자로서 사용자의 비밀보호(Privacy Protection)을 제공한다. Name은 속성의 이름이고 Count는 서비스의 개수를 말한다. Delegation은 위임 단언을 받은 에이전트가 다른 에이전트들에게 과제를 위임하는 것이 허가되었는지 여부를 나타낸다. 만약 이것이 true이면 재귀적인 위임의 자격이 부여된다. Consent는 위임기관(DA)이 위임 단언을 발행하는데 있어서 사용자의 동의를 얻었는지 여부를 나타낸다. 만약 이것이 true이면 동의는 얻어진 것이다.

Service는 IDRef 속성에 의해 EncryptedAttribute를 참조할 수 있고 SP-Address는 서비스 제공자(Service Provider) URL를 뜻한다. 끝으로 EncryptedAttribute는 서비스를 수행하기 위한 주체의 입력 데이터이고 이것은 WSDL의 입력(Input)에 따른다. 이것은 서비스의 기밀성(Confidentiality)을 제공하기 위하여 서비스 제공자의 공개키로 암호화 된다.

[그림 4]는 위임단언의 예제를 보여준다. 위임단언에서 <EncryptedData>와 <EncryptedKey>는 XML 암호화 명세서[10]에 기술되어 있기 때문에 논문에서는 설명하지 않는다.

3.4 위임 단언 발행 및 검증

웹 위임 단언은 위임기관에 의해 발행되고 웹 서비스 제공자에 의하여 검증된다. 위임 기관은 인증기관에서 인증된 주체 에이전트의 자격정보의 유효성을 검증하고 발행 위임단언 발행 알고리즘(Delegation Assertion Issue algorithm)을 적용하여 위임 단언을 발행한다. 웹 서비스 제공자는 위임 단언 검증 알고리즘(Delegation Assertion Verification algorithm)을 적용하여 위임단언의 유효성을 검증한다.

제안한 두 알고리즘에서 표기법은 다음과 같이 나타낸다. 위임단언(DAT)은 위임기관에 의해 발행된 위임단언(Delegation Assertion Ticket)이고 인증단언(AAT)은 인증기관에 의해 발행된 인증단언(Authentication Assertion Ticket)을 나타낸다. '?'은 주체 에이전트(Pa), 과제분담 에이전트(Ca) 그리고 서비스 제공자 에이전트(Sa)를 포함하는 에이전트를 뜻한다. '!'은 웹 서비스 제공자 및 인증기관(AA)과 위임기관(DA)을 포함하는 기관을 의미한다. PK!/PK!-은 '!'의 공개키 및 개인키를 의미한다. 에이전트는 서비스 사용에 성능 향상하도록 그들의 권한을 위임할 수 있다. 예를 들어, 엘리스 주체(P)는 서울에 가는 항공 예약하기를 원한다. 또한 비즈니스 기간 동안 호텔을 잡기 원한다. 엘리스는 첫째로 그녀의 에이전트에게 그녀의 권한을 위임한다.

그녀의 직무를 수행하는 주체에이전트 Pa는 그녀의 서비스 요구 수가 많은 만큼 다른 에이전트에게 그녀의 권한을 위임한다. 이것은 SPKI/SDSI[12]가 위임 유효성 검사를 위해 순차적 증명 연결 방식을 형성하기 때문에 SPKI/SDSI 매커니즘보다 더 효율적이다. 그 결과 검증측면에서 시간 복잡도(Time complexity)가 증가한다.

그러나 제안 모델은 웹 서비스 제공자는 위임 체인을 요구하지 않는다. 왜냐하면 Pa가 Ca에게 위임을 할지라도 위임기관에 의해 관리되기 때문에 위임단언의 증가 없이 Sa가 단언을 검증할 때 오직 하나의 위임단언의 유효성 검사만 하면 되므로 시간을 단축 할 수 있다. 웹 서비스 제공자는 오직 검증을 위한 위임단언(DAT)만을 요구한다. 각 서비스 제공자는 위임단언(DAT)의 엘리먼트 EncryptedAttribute가 수령인 정보를 포함하기 때문에 수령인 주체 에이전트(Pa)에게 직접적으로 전달된다. 주체 에이전트가 과제 분담 에이전트에게 위임하기 위해서는 반드시 동의 엘리먼트 속성 값이 'true'인 위임단언을 포함해야 한다.

먼저 [그림 5]에서 위임단언 발행 알고리즘을 설명하면 다음과 같다. 위임기관(DA)은 주체 에이전트(Pa)나 주체 에이전트로부터 위임 받은 과제분담 에이전트(Ca)로부터 위임단언(DAT) 요청을 받는다. 만약 위임기관(DA)이 주체 에이전트(Pa)로부터 직접 위임단언(DAT) 요청을 받을 시에는 위임 기관(DA)이 주체 에이전트

Algorithm 1 : Delegation Assertion Issue algorithm

```
// DAT : Delegation Assertion
// AAT : Authentication Assertion
// PK! : The Public Key of '!'
// PK!- : The Private key of '!'
// C? : The Credentials of '?'
// C_Pa : The Credentials of Pa
/* ? : It denotes agents containing Principal agent (Pa), Carrier(task) agent (Ca),
and Service agent (Sa)*/
/* ! : It denotes Service Provider (SP) and authorities containing
Delegation Authority (DA), Authentication Authority (AA), Authentication Authority Ticket (AAT)*/

Input : C_Pa / C_a and DAT
Output : DAT / Reject

(1) DA gets the DAT request from Pa or other agents.
(2) If (DA receives with C_Pa from Pa)
      Then
(3)     DA request AA to issue AAT with C_Pa
(4)     If (AAT is not returned) Then
(5)       Return Reject
        Else
(6)       Set Delegation element is true.
(7)       Issue DAT signed by PK_DA-
      Else
(8)       If Sa don't trust Ca, DA confirms DAT's Consent element.
(9)       If (Consent element is false) Then
(10)        Return Reject
          Else
(11)        DA request AA to issue AAT with C_a
(12)        If (AAT is not issued) Then
(13)          Return Reject
            Else
(14)          Decrypts AAT using PK_DA-
(15)          Query to Subject whether consent is true or not.
(16)          If (Consent is true) Then
(17)            Repeat Step 6 ~ Step 7
              Else
(18)            Set Consent element is false
(19)            Return Reject
```

[그림 5] 위임단언 발행 알고리즘

(Pa)의 자격정보 및 증명 정보를 직접 받는다.

위임기관(DA)은 인증기관에서 주체 에이전트 자격정보 및 증명정보를 담은 인증단언(AAT)을 발행할 것을 요구한다. 만약 인증단언(AAT)이 발행되지 않으면 거절을 표시한다. 인증기관이 주체 에이전트의 자격정보 및 증명정보를 담은 인증단언(AAT)을 발행하면, 위임기관은 개인키로 전자 서명한 위임 단언(DAT)을 발행한다. 위임기관의 개인키로 전자 서명된 위임단언(DAT)은 이에 대응하는 공개키를 가진 에이전트만이 그 서명을 검증할 수 있다. 전자 서명된 위임단언(DAT)의 공개키로 검증된다면 이 문서는 위임기관의 개인키로 전자 서명된 것임을 알 수 있다.

Sa(서비스 에이전트)는 Pa로부터 넘겨받은 Ca의 DAT를 검사하여 서비스를 제공한다. 하지만 만일 Sa가 Ca에 대한 정보가 없을 경우, Sa가 Ca를 신뢰할 수 없으므로 Sa의 요구에 의해 Ca는 위임기관에게 DAT를 재발행하게 된다. 이때 Ca는 Pa의 자격정보와 증명정보를 받지 않고 오직 DAT만을 위임받은 상태이기 때문에 위임기관은 Ca의 에이전트 정보를 알 수 없으므로 인증기관에게 이들 정보를 포함한 인증단언 발행을 요구하게 된다. 위임기관(DA)은 위임단언(DAT)의 동의(Consent) 엘리먼트를 확인하여 동의 엘리먼트가 거짓이면 거절을 표시하고, 참이면 위임기관(DA)은 인증기관에게 Pa, Ca 및 Sa를 포함하는 에이전트의 자격정보 및 증명 정보를 담은 인증단언(AAT)을 발행할 것을 요구한다. 마찬가지로, 만약 인증단언(AAT)이 발행되지 않으면 거절을 표시한다. 인증기관이 상기 에이전트들의 자격정보 및 증명 정보를 담은 인증단언(AAT)을 발행하면, 인증기관은 위임기관의 공개키로 인증단언(AAT)을 암호화한다. 그리고 암호화된 인증단언은 위임기관의 개인키로 암호를 해독하고 위임단언을 발행한다.

또한 위임기관(DA)은 주체 에이전트(Pa)에게 동의 엘리먼트가 참인지 문의하여 참이면 위임기관의 개인키로 서명된 동의 엘리먼트가 True인 위임단언(DAT)을 주체 에이전트로부터 위임받은 과제분담 에이전트(Ca)에게 위임단언(DAT)을 재발행하고 만약 거짓이면 동의 엘리먼트를 False로 하여 거절한다.

다음으로 [그림 6]에서처럼 위임단언을 검증하는 알고리즘을 설명하면 다음과 같다.

서비스 제공자(SP)는 웹 서비스 제공자의 에이전트(Sa)로부터 위임단언(DAT)을 얻고 위임기관의 공개키를 이용하여 위임기관의 서명을 확인한다. 서명이 유효

Algorithm 2 : Delegation Assertion Verification algorithm

```
Input : DAT
Output : Success / Fail

(1) Service Provider (SP) gets the DAT from SA.
(2) SP confirms the signature of DA in DAT using PK_DA
(3) If (Signature is not valid) Then
(4)    Return Fail
    Else
(5)    Decrypts encrypted input data using PK_SP-
(6)    Confirms input data and process it
(7)    If (Result value is successful) Then
(8)       Return Success
       Else
(9)       Return Fail
```

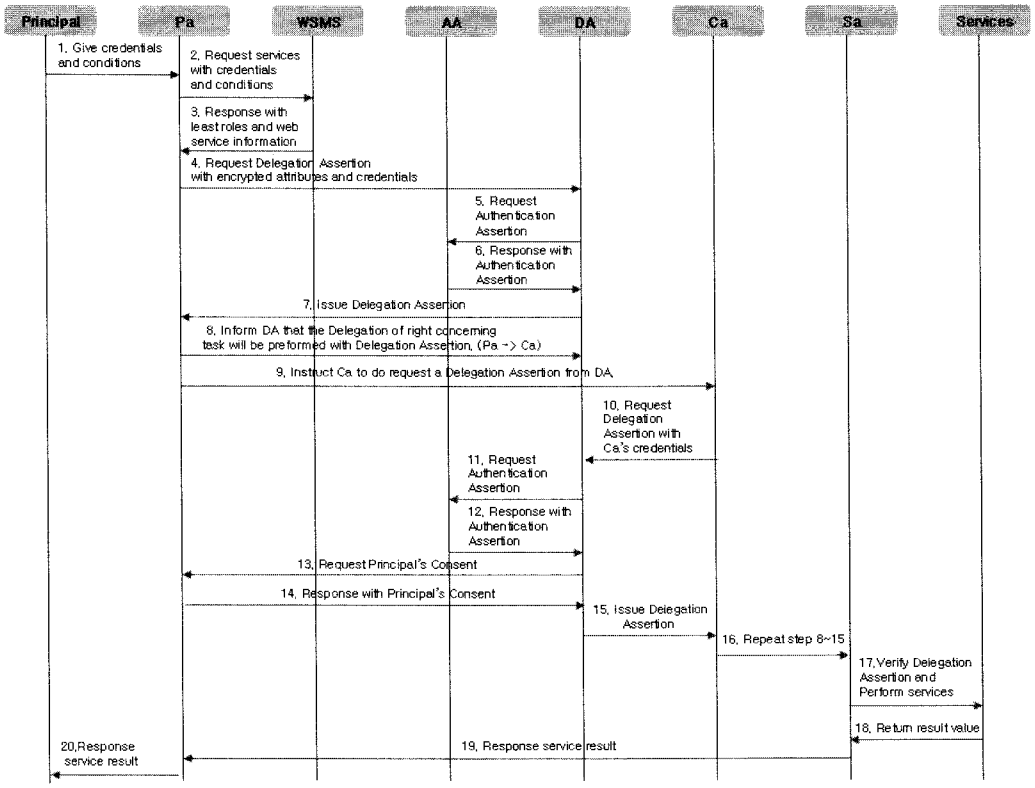
〔그림 6〕 위임단언 검증 알고리즘

하지 않으면 실패를 나타낸다. 서명이 유효하면 서비스 제공자의 개인키를 이용하여 암호화된 서비스 입력 데이터를 해독, 확인하고 이를 처리한다. 결과 값이 성공적이면 성공을 표시하고, 그렇지 않으면 실패를 표시한다.

3.5 멀티 에이전트를 이용한 위임모델의 상호작용

[그림 7]은 멀티 에이전트를 적용한 위임모델의 순서도표(Sequence Diagram)를 기술한다. 이장에서는 각 단계를 보다 자세히 설명한다.

1. 주체(P)는 원하는 웹 서비스를 받기 위하여 주체 에이전트(Pa)에게 자격정보와 증명정보를 준다.
2. 주체 에이전트(Pa)는 웹 서비스 관리서버(WSMS)에게 웹 서비스 제공자에 의해 등록된 서비스 정보를 요청한다.
3. 웹 서비스 관리서버(WSMS)는 주체 에이전트(Pa)의 검색조건에 대응하는 서비스 정보를 검색하고 Pa에게 웹 서비스를 이용할 수 있는 최소한의 특권(Least Privilege)을 가진 역할을 부여한다.
4. 주체 에이전트(Pa)는 다른 에이전트에게 Pa의 권한을 안전하게 위임하기 위해 위임기관(DA)에게 위임 단언을 요청한다. Pa는 주체의 개인키에 의해 암호화된 자격정보와 주체의 프라이버시를 보호하기위한 웹 서비스 제공자의 공개키로 암호화된 속성을 위임기관(DA)에게 보낸다. 위임기관(DA)은 위임된 에이전트는 신뢰성과 위임단언을 기반으로 위임기관에 의해 안전하게 관리된다는 것을 보증한다.
5. Pa에서 암호화된 자격정보를 받으면 위임기관은 인증기관(AA)에게 인증 단언을 요청한다.
6. 인증기관(AA)은 인증기관의 공개키로 암호화된 주체의 자격정보를 개인키로 복호화한다. 만약 인증기관이 위임기관으로부터 자격 유효성을 확인하면, 인증기관은 주체의 유효성을 보증하는 인증단언을 발행한다. 반대로 주체 유효성이 확인되지 않는다면 인증단언 발행을 거절한다.
7. 인증기관에서 의해 인증단언이 발행하면, 위임기관(DA)은 주체에게 위임기관의 개인키로 전자 서명한 위임단언(DAT)을 발행한다.
8. 주체 에이전트의 권한을 위임하기 위해 Pa는 Pa의 권한들을 과제분담 에이전트(Ca)에게 위임하고 위임기관에게 알린다.
9. 주체 에이전트는 Ca에게 위임단언을 주고, Ca를 위한 새로운 위임 단언 요청할 수 있도록 위임기관에게 전달한다. Ca는 위임단언이 위임기관에 의해 관리되기 때문에 주체 에이전트의 위임단언 사용은 부적당하다고 볼 수 없다. 다만 제한시간(Time Constraints)을 가진다.
10. Sa가 과제분담 에이전트(Ca)를 신뢰하지 못할 경우, Ca는 위임기관에게 DAT인증을 요청한다.
11. 위임기관은 위임단언 내의 동의 엘리먼트의 속성값을 확인한다. 만약 동의 엘리먼트가 사실이면 인증기관에게 인증단언을 요청하고 Ca에게 위임단언을 재 발행한다. 만약 동의 엘리먼트가 거짓이라면 위임단언을 발행하지 않는다.
12. 인증기관은 Ca의 자격 정보의 유효성을 확인하고 주체 에이전트의 공개키를 사용하여 위임단언의 주체 엘리먼트를 복호화한다. 만일 Ca의 자격정보가 유효하면 인증기관은 Ca를 위한 인증단언을 발행한다.
13. 주체 에이전트의 공개키로 위임단언을 복호화한 위임 기관은 Pa에서부터 동의 여부를 알 수 있다. 위임기관은 Ca를 위한 위임단언의 동의 엘리먼트 확인을 위해 Pa의 동의를 요청한다.
14. 주체 에이전트는 동의 엘리먼트 설정 여부를 위임기관에게 보낸다.
15. 위임기관은 Ca에게 위임단언을 발행한다. 만일 주체 에이전트가 위임동의를 하면 Ca의 위임단언 동의 엘리먼트가 True이고 반대로 거짓이다.
16. 주체 에이전트에서 위임받은 과제 분담 에이전트(Ca)는 많은 사용자 요구 서비스에 다른 임무 분업 에이전트들에게 더 많은 위임이 필요하다. 따라서 과제 분담 에이전트 절차 8~15로 반복하여 실행할 수 있을 것이다. 그러나 Ca는 주체 에이



〔그림 7〕 위임모델의 순서도표

전트의 동의 없이 그의 권한을 위임할 수 없다. 즉 위임 단언의 최초의 발행은 주체 에이전트가 지니고 나중에 과제 분담이 필요할시 Ca에게 위임을 주기 때문에 주체 에이전트의 동의 없이 권한을 위임할 수 없다.

17. 마지막으로 웹 서비스 제공자의 에이전트(Sa)는 Ca에서부터 요청 서비스를 받는다. Sa는 위임기관의 공개키로 Ca의 위임단언의 유효성을 검사한다. 만약 Ca의 위임단언이 유효하면 Ca 위임단언에서 암호화된 속성을 넘겨주고 Sa는 웹 서비스 제공자에게 웹 서비스를 요청한다.
18. 웹 서비스 제공자는 서비스 제공자의 개인키로 암호화된 서비스 입력데이터를 해독한다. 웹 서비스 제공자는 주체로부터 웹 서비스를 처리하고 Sa에게 서비스의 결과를 반환한다.
19. 서비스 에이전트(Sa)는 주체 에이전트에게 웹 서비스 결과를 이동한다. Sa는 Ca의 위임단언에서 서비스 입력 데이터가 수령 에이전트를 포함하기 때문에 주체 에이전트에게 직접적으로 이동할 수 있다.
20. 주체 에이전트는 주체에게 웹 서비스의 결과를 반환한다.

무엇보다 상호작용의 연속은 신뢰할 수 없는 에이전트 중에 위임단언 동적 획득을 포함하기 때문에 보안관점에서 약하게 보여질 수 있다. 하지만 주체는 항상 에이전트 중 위임을 모니터링 할 수 있다. 또한 제안 모델은 XML기반 암호화와 전자 서명을 기반으로 하기 때문에 보안에서 어떠한 위험성 없이 웹 서비스를 개발할 수 있다.

Ⅳ. 분석 및 평가

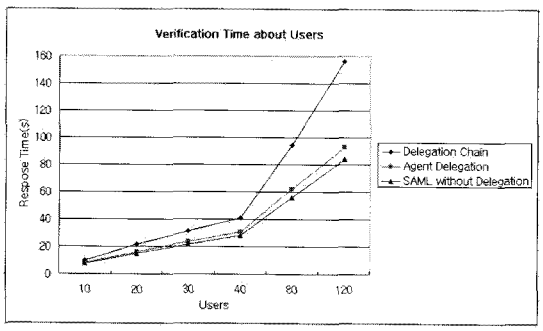
본 절에서는 제안 에이전트 기반 위임 기법의 특징을 분석하고 사용자가 인증 받는데 걸리는 시간을 측정한다.

제안 기법은 다음과 같이 여러 측면을 고려하여 위임 모델을 분석하면 다음과 같다. 첫째, OASIS의 표준인 SAML 2.0을 기반으로 하여 위임단언을 정의함으로써 SAML 기반의 다른 시스템들과의 유연성(Flexibility)

을 제공하였다. 둘째, 확장성 측면에서 W3C에서 권고하는 XML 암호화와 전자서명 기술을 적용하여 사용자의 프라이버시의 노출 없이 권한을 위임할 수 있는 방법을 제시하였다. 셋째, 신뢰할 수 있는 기관들의 관리와 사용자의 동의를 기반으로 한 권한위임으로 웹 서비스의 효율성을 증가시켰다.

성능 평가를 위한 하드웨어의 사양은 인텔 듀얼 코어 CPU와 메모리 2GB에서 수행하였고 Window 2003 Server Service Pack2의 운영체제와 웹 서버구축에 필요한 IIS(Internet Information Server) 6.0을 설치하였다. 또한 SAML 파싱 및 API를 구현하기 위해서 JAVA JDK 1.5.0_06의 도구를 사용하였다.

[그림 8]과 같이 사용자의 로그인 수를 10. 20. 30. 040, 80, 120명으로 분류하고 로그인에 비례하여 발행한 위임단언을 인증하는데 걸리는 검증 시간을 비교 측정하였고 또한 위임 체인(Delegation Chain)과 제안 기법은 위임 횟수를 사용자의 수만큼 랜덤(Random)하게 발생시켜 측정하였다. 먼저 위임 체인일 경우, 10명을 인증하는데 걸리는 시간은 9.8초에서부터 120명의 검증시간은 156.5초 1회 사용자의 위임단언의 검증시간은 평균 1.18초이고 계속 증가함을 알 수 있다. 이것은 사용자 증가에 따른 권한 위임의 횟수가 증가하게 되면 위임 횟수만큼 SAML 토큰의 개수도 늘어나기 때문으로 분석된다. 하지만 제안 방법은 평균 0.78초로 위임을 적용하지 않은 SAML 단언과 별 차이 없을 정도로 시간 복잡도(Time Complexity)를 크게 감소시켰다. 이것은 위에서 언급했던 것처럼 에이전트로부터 받은 위임단언의 유효성(Validity)만을 검증하면 되기 때문으로 해석된다. SAML는 단언(Assertion)을 전체 또는 중요한 부분만 암호화할 수 있기 때문에 앞으로 검증 시간을 좀 더 앞당길 수 있을 것으로 본다.



[그림 8] 위임방식에 따른 검증시간 비교

V. 결 론

현재 가장 큰 이슈 중 하나는 보안 문제이다. 지난 몇 년 동안 정보 보안과 관련된 수많은 새로운 표준들이 개발되었다. 그 중 주목할 만한 표준은 SAML이다. SAML는 인증 및 권한부여에 사용되는 보안 서비스이며 SSO(Single Sign-On)에 활용될 수 있다. SSO는 하나의 보안 영역에서 인증받은 주체가 다른 보안 영역에서 재인증 과정 없이 서비스를 제공받을 수 있고 전자거래를 위한 분산처리와 분산된 접근제어를 지원하는 다른 애플리케이션에서도 활용될 수 있다. 하지만 SAML 기반으로 위임에 관한 기존 연구는 위임의 횟수가 증가하게 되면 많은 비용과 노력이 요구되는 문제점을 안고 있었다.

이에 반해 제안 기법은 멀티 에이전트를 통한 웹 서비스의 이용에 있어서 사용자의 프라이버시 정보를 보호하면서 사용자에 의해 관리될 수 있는 에이전트들 간의 권한 위임모델을 제안하였다. 제안 모델은 OASIS의 공개 표준인 SAML을 확장된 위임단언을 정의하고, W3C의 권고안인 XML 암호화와 전자서명 메커니즘을 적용하여 권한 위임 시 멀티 에이전트 간의 신뢰를 기반으로 사용자의 프라이버시 정보의 노출을 방지하였다. 그리고 인증기관과 위임기관의 관리 하에 사용자의 동의를 기반으로 위임 단언이 발행되기 때문에 웹 서비스 제공자는 에이전트로부터 받은 위임단언만 검사하면 되므로 검증 시간을 크게 향상 시킬 수 있었다. 제안 논문은 앞으로 분산 환경에서 사용자의 프라이버시를 고려한 에이전트 기반의 웹 서비스 분야에서 많이 활용될 것이다.

참고문헌

[1] OASIS "Profile for the OASIS Security Assertion Language(SAML)V2.0" *OASIS Standard*, 15 March 2005.

[2] OASIS "eXtensible Access Control Markup Language (XACML)V2.0" *OASIS Standard*, 1 February 2005.

[3] D. Ferraiolo, R. Kuhn, "Role-Based Access Control" *Proceedings of 15th National Computer Security Conference*, 1992.

[4] G. Navarro, B.S. Firozabadi, E.Rissanen and

J.Borrell, "Constrained delegation in XML-based Access Control and Digital Rights Management Standards", *Communication, Network, and Information Security*, 2003.

[5] V.Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlaman, S. Tuecke, J. Gawor, S. Meder and F. Siebenlist, "X.509 Proxy Certificates for Dynamic Delegation", *In 3rd Annual PKI R&D Workshop*, 2004.

[6] Hidehito Gomi, Makoto Hatakeyama, Shigeru Hosono, Satoru Fujita, "A Delegation Framework for Federated Idenity Management", *In Proceedings of DIM workshop*, 2005.

[7] Y. J Hu, "Some thoughts on agent trust and delegation", *In Proceedings of the fifth International Conference on Autonomous Agents*, 2001.

[8] Juan Dai, Rolbert Steele, "UDDI Access Control", *In Proceedings of the Information Technology and Applications*, 2005.

[9] Jun Wang, David Del Vecchio, Marty Humphrey, "Extending the Security Assertion Markup Language to Support Delegation for Web Services and Grid Services", *IEEE International Conference on Web Services*, 2005.

[10] XML Encryption Syntax and Proceeding, http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

[11] C. A. Ardagna, E. Damiani, S. De Capitani di Vimercati, P.Samarati, "XML-based Access Control Language", *Elsevier Information Security Technical Report*, 2004.

[12] D. Clark, J. Elien, C. Ellision, M.Fredette, A. Marcos, R. Rivest, "Certificate Chain Discovery in SPKI/SDSI", *ACM Journal of Computer Security*, 2001.

[13] R. Sandhu, E. Coyne, H. Fenstein, and C. Youman, "Role-Based Access Control Models", *IEEE Computer*, February 1996.

[14] XML Signature, http://www.w3.org/TR/xmldsig-core

[15] V. Semar, "Single Sign-On Using Cookies for Web application. Proceedings", *IEEE 8th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise (WET ICE 99)*, 1999.

[16] B. Pfitzmann, B. Waidner, "Token-based web Single Sign-On with Enabled Clients", *IBM Research Report RZ 3458(93844)*, Nonmember 2002.

[17] W3C, Web Service Description Language (WSDL) 1.1, W3C Note, March 2001. http://www.w3.org/TR/wsdl.html

[18] X. Feng, L. Guoyuan, H. Hao, and X. Li, "Role-Based Access Control System for Web Services." *In Proceedings of the 4th International Conference on Computer and Information Technology (CIT 04)*, 2004.

[19] OASIS, "Web Service Security: SOAP Message Security 1.0", *OASIS Standard*, March 2004.

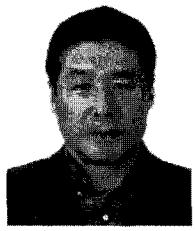
〈著者紹介〉

김 규 일 (Kyu-il Kim) 정회원

2003년 2월 : 원광대학교 컴퓨터정보통신과 졸업
2005년 2월 : 성균관대학교 컴퓨터공학과 석사
2005년 3월~현재 : 성균관대학교 컴퓨터공학과 박사과정
<관심분야> 정보보호, 접근제어, 웹서비스 보안

원 동 호 (Dong-ho Won) 종신회원

1976년~1988년 : 성균관대학교 전자공학과(학사, 석사, 박사)
1978년~1980년 : 한국전자통신연구원 전임연구원
1985년~1986년 : 일본 동경공업대 객원연구원
1988년~2003년 : 성균관대학교 교학처장, 전지전자 및 컴퓨터공학부장, 정보통신대학원장, 정보통신기술연구소장, 연구처장.
1996년~1998년 : 국무총리실 정보화추진위원회 자문위원
2002년~2003년 : 한국정보보호학회장
현재 : 성균관대학교 정보통신공학부 교수, 한국정보보호학회 명예회장, 정보통신부지정 정보보증기술연구센터 센터장
<관심분야> 암호이론, 정보이론, 정보보호



김 응 모 (Ung-mo Kim) 정회원

1981년 2월 : 성균관대학교 수학과 졸업
1986년 2월 : Old Dominion University Computer Science 석사
1990년 2월 : Northwestern University Computer Science 박사
2001년 1월 : 서울시 지방공무원 교육원 자문위원
2002년 2월 : 경기도청 정보시스템 보완 및 확대사업 평가위원, 문화관광부 자문위원
2002년~2006년 : 한국정보처리학회 편집위원
1990년 9월~현재 : 성균관대학교 정보통신공학부 컴퓨터공학전공 교수
<관심분야> 정보보호, 접근제어, 데이터 마이닝