情 報 保 護 學 會 誌
第 16 卷 第 5 號, 2006. 10

Active XML 기반 "전자의료기록 시스템"의 보안성 분석

김 일 곤*, Debmalya Biswas*

요 약

XML기반 웹서비스의 활성화와 더불어 효율적인 데이터 호출 및 통합을 위해, XML 문서 안에 웹서비스 호출 노드를 내장할 수 있는 Active XML (AXML) 문서가 개발되었다. 본 논문에서는 기존 전송계층 보안 프로토콜 및 WS-Security의 제한점을 방지하기 위한, AXML 문서의 내장형 웹서비스 호출과 XML-Encryption 및 XML-Signature 보안 표준을 통합한 데이터 암호화 및 전자서명 방식에 대해 소개한다. 또한, "전자의료 시스템" 예제를 통해 AXML 문서를 이용한 중첩된 웹서비스 호출 및 쿼리위임 방식에 대해 소개한다. 마지막으로, 기존 보안성 정형분석 방법의 확장을 통한 AXML 시스템의 보안 취약 가능성에 대해 언급하고자 한다.

Ⅰ. 서 론

XML기반 웹서비스 분야의 활성화와 더불어, 분산 환경에서 XML 기반 정보 데이터를 효율적으로 관리하고 통합하기 위한 연구의 필요성이 대두되었다. 이에 따라, P2P 환경 하에서 XML 문서 안에 웹서비스 호출 기능을 내장하여, 동적으로 서비스 제공자로부터 해당 정보를 호출하고 수신할 수 있는 Active XML (AXML) 언어가 제안되었다[1]. 또한, SSL과 같은 기존 전송계층 보안 프로토콜 및 SOAP 채널 보안 표준인 WS-Security[2]의 제한점으로 인하여, AXML 문서 안에 웹서비스 호출기능과 암호화 알고리즘을 통합하여, 해당 중요 데이터를 안전하게 전송하고 통합하고자 하는 연구가 진행되었다.

지난 수년간, 중요 자원 혹은 사용자 정보 보호를 위해 다양한 보안프로토콜들이 개발되어 졌다. 하지만, 대부분의 많은 보안 프로토콜들은 개발된 후, 시간이 지남에 따라 점차 보안 취약점들이 하나둘 발견되고 있다. 예를 들어, Needham-Schroeder 프로토콜은 제안 된지, 17년이 지나고서야 보안 취약점을 발견할 수 있었다[3]. 그 만큼 안전한 보안프로토콜을 설계하는 것은 매우 복잡하면서도 중요한 연구 분야이다.

이에 지난 수년간, 미국 및 유럽을 중심으로 한 선진연구기관에서는 정형기법을 이용하여, 설계단계에서부터 보안 프로토콜의 보안성을 검증하기 위한 연구가 진행되었다. 그 결과, 보안 프로토콜 설계상에서 생길 수 있는 보안 취약점을 검증하기 위해 다양한 방법론과 도구들이 개발되었다[4].

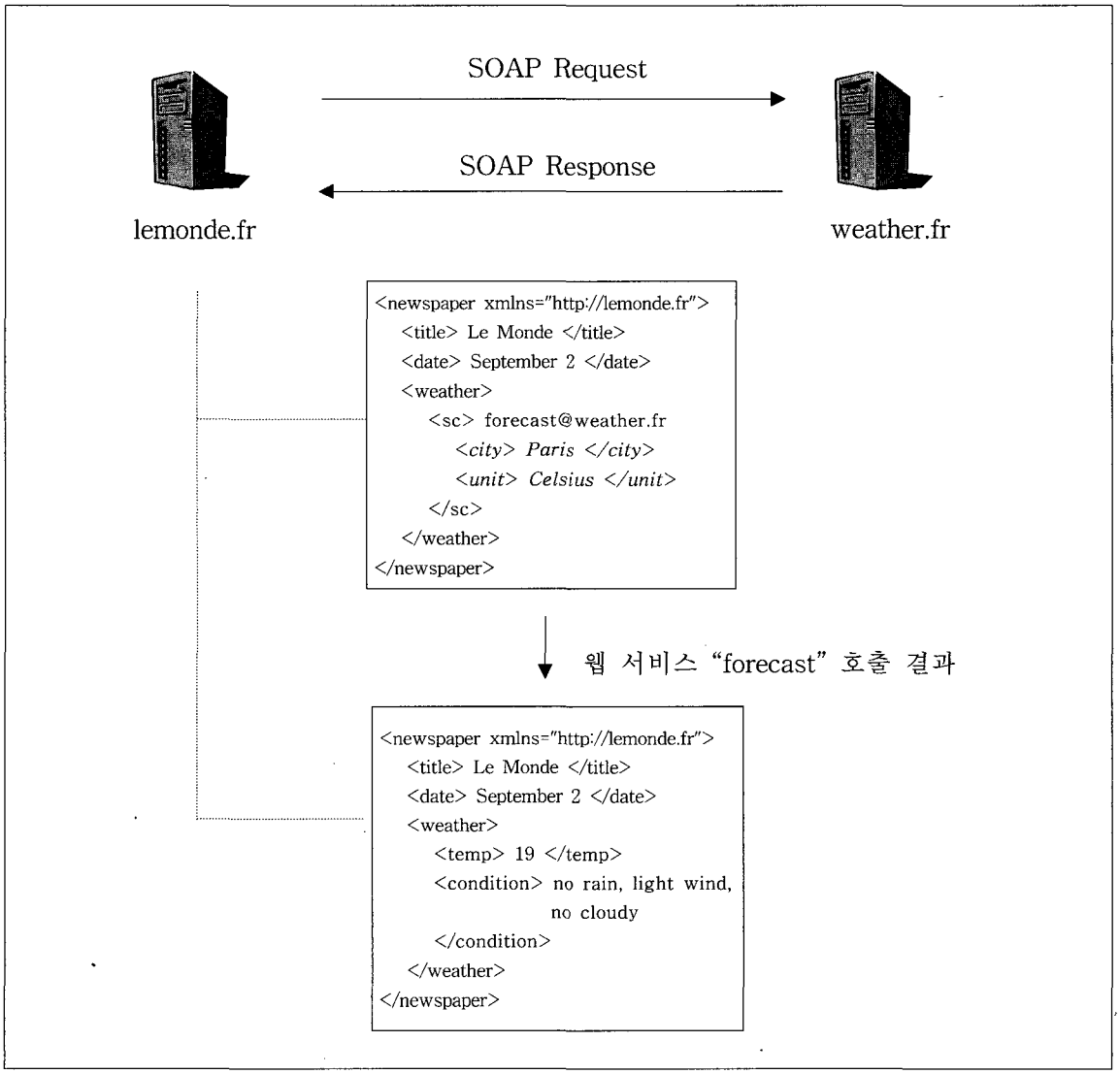
그중에서도, Casper[5], CSP[6] 및 FDR[7]을 이용한 정형분석 방법은 다양한 유.무선 프로토콜의 보안성을 분석에 적용결과, 설계 시 예측치 못한 보안 취약점을 찾아냄으로써, 효율적인 분석방법으로 자리 잡게 되었다[8]. 이에 본 연구에서는 AXML 시스템의 보안성 분석을 위해, Casper, CSP 및 FDR 방법을 채택하였다.

하지만, 기존의 분석 방법들은 메시지 기반 보안 프로토콜에 적용되었다. 이에 따라, AXML과 같은 문서 기반 웹서비스 호출 프로토콜의 분석을 위해서는 증요 보안 요구사항 파악 및 모델 추상화 기법 개발과 같은 기존 방법론에 대한 확장성이 요구된다.

본 연구에서는 AXML 기반 동적 웹서비스 문서보안 방식을 소개하고, AXML 시스템 보안성 정형 분석을 위한 프레임워크를 제시하고, 마지막으로 "전자의료기록 시스템 (Electronic Patient Record System)"의 사례연구를 통해 AXML에서 발생할 수 있는 보안 취약 가능성에 대해 지적하고자 한다.

본 논문의 구성은 다음과 같다. 제Ⅱ장에서는 AXML 문서의 구성 및 웹서비스 호출 방식에 대해 간

본 연구는 프랑스 INRIA 연구소 ARC-ASAX 및 RNRT-SWAN 프로젝트의 지원하에 수행되었음.
* IRISA/INRIA 연구소 ({ikim,dbiswas}@irisa.fr)



〔그림 1〕 AXML 문서를 이용한 웹서비스 호출

략히 요약 설명한다. Ⅲ장에서는 XML Encryption[9] 및 XML Signature[10] 보안표준과 AXML의 내장형 웹서비스 호출을 통합한 데이터 및 문서 보안 방식에 대해 소개한다. Ⅳ장에서는 AXML 문서 기반 "전자의료 기록 시스템" 예제를 소개한다. Ⅴ장에서는 AXML 시스템의 보안성을 분석하기 위한 프레임워크를 제시하고, 예제에 대한 보안성 분석을 통해 AXML 시스템의 보안 취약 가능성에 대해 언급한다. Ⅵ장에서는 기존의 관련연구를 소개하고, 마지막으로 Ⅶ장에서는 결론을 맺고자 한다.

Ⅱ. Active XML (AXML) 문서

1. 개 요

AXML 문서는 기본적으로 XML 언어를 바탕으로 구성되어져 있다. 따라서, XML 문서와 같이 웹 서비스 환경 하에서 P2P 호스트의 플랫폼의 상관없이, SOAP 채널을 통한 자유로운 데이터 교환이 가능하다.

AXML 문서가 XML 문서와 다른 가장 큰 차이점은 문서 안에 웹서비스를 호출할 수 있는 함수를 엘리먼트(element) 형태로 내장할 수 있다는 점이다. 이

내장형 웹서비스 함수 호출기능을 통해, 서비스 제공자 호스트에 정의되어 있는 웹서비스를 요청하게 되면, 웹서비스 평가(evaluation) 후에, 관련 데이터는 문서내의 서비스 요청이 호출된 엘리먼트 위치로 통합되어 진다.

예를 들어, [그림 1]에서 보는 바와 같이 newspaper 태그를 최상위 노드로 갖는 AXML 문서를 갖고 있는 lemonde.fr 호스트와 서비스 제공자 weather.fr 호스트가 있다. 이 AXML 문서 안에는 weather.fr 호스트에서 제공되는 forecast 웹 서비스를 호출하기위한 노드가 내장되어 있다; 〈sc〉 태그는 서비스 호출 (service call) 을 나타내기 위해 AXML 문서에서 사용되는 기호이다. 그리고, 〈sc〉 태그의 하위 노드는 해당 서비스 함수의 인자(parameter)를 나타낸다. 따라서, 〈city〉Paris 〈/city〉 와 〈unit〉Celsius〈/unit〉은 forecast 서비스 호출시 인자값으로 SOAP 채널을 통해, weather.fr 호스트에 전달되어 지며, 서비스 평가후에 파리(Paris)에 대한 현재 온도(Celsius 단위)에 대한 결과 (〈temp〉와 〈condition〉 노드)값이 본래 AXML 문서의 내장형 서비스 노드 위치로 통합되게 된다. 이러한 내장형 웹서비스 호출 방식을 통해, 필요시 마다 동적으로 해당 데이터를 호출할 수 있기 때문에, 기존 웹서비스 호출 방식에서 수많은 데이터를 유지, 관리 하는데 생기는 문제점을 해결 할 수 있다. 예를 들어, 파리의 온도는 매시간 변할 수 있기 때문에 동적 데이터 호출이 요구된다. 또한 해당 서비스 호출시 인자 값 변환을 통해 다양한 정보를 손쉽게 얻을 수 있다 (예, 〈city〉seoul〈/city〉 로 변환 가능).

2. 대수 기호를 이용한 AXML 서비스 호출방식의 의미론 정의

본 소단원에서는 간략한 대수 기호를 이용하여, AXML 문서의 웹서비스 호출 및 문서 교환방식에 대한 의미론을 표현하고자 한다.

정의 2.1. 웹서비스는 (p, s) 두 개의 튜플(tuple)로 구성되어 있다.

- $p \in P$: p는 P2P 환경에 속한' 한 호스트의 식별자를 의미한다.
- $s \in S$: s는 웹서비스의 명칭을 나타낸다.

AXML 문서, 서비스 이름, 내장형 서비스 함수, 서비스 호출결과 데이터에 대한 간략한 대수 표현식은 다음과 같이 표현 된다[11].

- $d@p$: 호스트 p가 소유한 문서 d
- $s@p$: 호스트 p에서 제공하는 웹서비스 s
- $f(para_1,\ldots, para_n)@p$: 호스트 p에서 제공되는 서비스를 호출하기 위한 함수 f

Ⅲ. AXML 내장형 웹 서비스 호출을 이용한 이용한 데이터 보안

본 단원에서는 기존 SSL (Secure Socket Layer)과 같은 기존 전송계층 (Transport Layer) 보안 프로토콜 및 SOAP 채널 보안 표준인 WS-Security의 제한점에 대해 살펴보고, AXML 문서의 내장형 서비스 호출방식을 이용한 데이터 보안 방식에 대해 소개하고자 한다.

1. 보안 제약사항

SSL : 데이터 전송 채널 단계의 메시지 보안을 제공하지만, 데이터 저장단계의 문서 보안은 제공하지 않는다. 또한 암호시 XML 문서의 모든 노드를 암호화해야 하며, 특정 태그 영역에 대한 암호화 기능을 제공하지 못한다. 뿐만 아니라, P2P 환경 하에서, 웹서비스 호출은 여러 호스트를 경유할 수 있는데, 이 경우 SSL은 지점간 (peer-to-peer) 보안을 제공하지만, 단대단(end-to-end) 보안을 제공하지 못한다. 이에 따라, 여러 호스트를 경유하는 전달되는 XML 문서의 경우, 공격자의 위장 및 도청 공격에 취약점을 갖고 있다.

WS-Security : 웹서비스 보안의 중요성과 더불어 제안된 SOAP 채널 암호화 표준 방식으로, XML Encryption과 XML Signature 암호화 표준을 이용하여 단대단(end-to-end) 보안을 제공한다는 장점을 갖고 있다. 하지만, 이 표준은 우선 SOAP 채널 단계의 메시지 보안만을 제공한다. 또한 웹서비스 호출시 서비스 함수의 원형 및 인자 값들에 대한 보안을 제공하기 위한 목적으로 사용되기 때문에, 문서에 대한 암호화 및 전자서명을 적용하기에는 부적합하다.

이에 따라, AXML 문서 보안을 위해, SOAP 채널

단계에서 뿐만 아니라, 데이터 문서저장 단계에서의 보안을 제공하기 위해, AXML의 내장형 서비스 호출 기능을 통하여, 해당 중요문서를 XML Encryption 및 XML Signature로 암호화하거나 전자서명할 수 있는 방식을 채택하였다.

XML Encryption 암호화 기술은 W3C의 워킹그룹에 의해 제안된 XML 문서 암호표준으로 다음과 같은 특징을 갖고 있다.

- 암호화 결과를 XML 형태로 생성 문서단위의 암호화 또는 선택적 노드단위의 암호화를 함께 지원
- 크게 〈Encrypted Method〉,〈KeyInfo〉 및 〈CipherData〉 태그로 구성
- 〈Encrypted Method〉 태그는 데이터 암호화 또는 키 암호화 알고리즘을 정의
- 〈KeyInof〉 태그는 암호키에 대한 정보 제 공
- 〈CipherData〉 태그는 암호화된 데이터에 대한 정보 제공

XML Signature은 XML 문서기반 전자서명 표준기술로 송수신자 사이에 교환되는 문서에 대한 인증, 무결성 및 부인봉쇄 서비스를 제공하며, 다음과 같은 특징이 있다.

- 전자서명 결과를 XML 형태로 생성
- 문서단위의 서명 또는 선택적 노드단위의 전자서명 가능
- 크게 〈SignedInfo〉, 〈SignatureValue〉 및 〈KeyInfo〉 태그로 구성
- 〈SignedInfo〉 태그는 전자서명 대상 및 알고리즘에 대한 정보 명시
- 〈SignatureValue〉 태그는 전자서명된 값을 명시
- 〈KeyInfo〉 태그는 전자서명을 확인(validate)하는데 필요한 개인키 정보명시

XML Encryption 및 XML Signature에 대한 상세한 자료는 참고문헌[9-10]을 참조하기 바란다.

Ⅳ. 전자의료기록 시스템

1. 가정 및 시나리오

A 병원에서 근무하고 있는 의사 "김"은 어느날 척추 디스크로 불편을 호소하는 한 환자의 방문을 받는다. 환자는 그동안 다른 지방에서 근무하는 동안 B 병원에서 허리 디스크 문제로 인하여, 수 차례 의사의 진단과 치료를 받아 왔으며, 현재는 A 병원이 위치한 지방에서 근무지를 옮겨온 상황이다. 의사 "김"은 환자와의 면담 후, 환자에 대한 본격적인 진료를 하기에 앞서, 웹서비스 호출을 통해 환자에 대한 전자의료기록 문서를 살펴보고자 한다.

〔그림 2〕에서 보는 바와 같이 의사 "김"은 자신의 컴퓨터에 저장되어 있는 AXML 문서를 이용하여, 전자환자기록 문서를 수신 받게 되며, 상세한 서비스 호출 절차는 다음과 같은 순서로 이루어 진다; 의사 "김"은 p_1, A 병원은 p_2, B 병원은 p_3 등으로 간략히 표현된다.

순서 1 : p1은 AXML 문서의 서비스 호출 노드 (sc : diagnosis@p_2로 표기)를 통하여, p2에 diagnosis 웹 서비스를 호출하게 된다. 이때, 서비스 호출시 전달되는 인자는 쿼리 q1으로 다음과 같이 표현되어 있다가 가정한다.

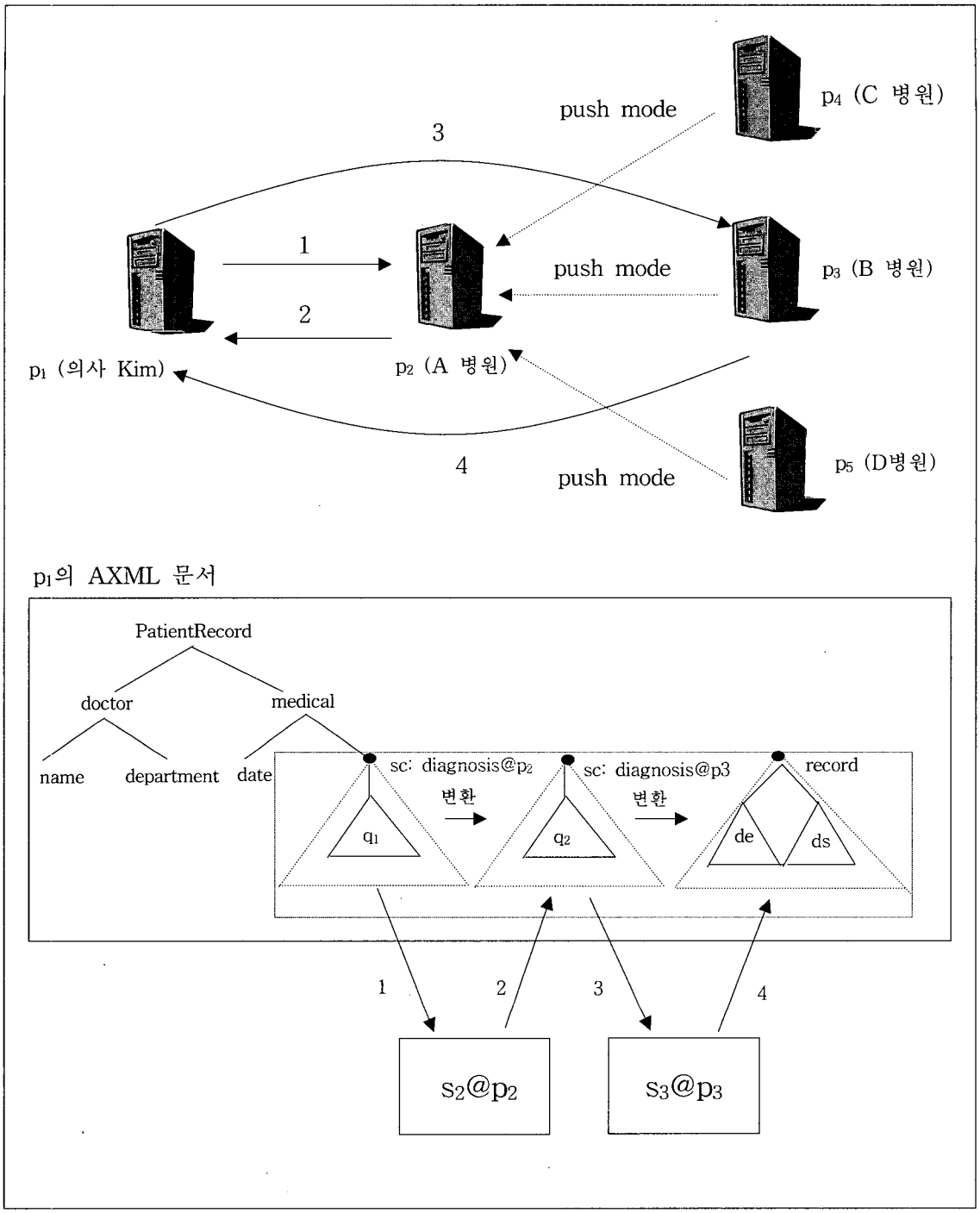
q_1 = *PatientRecord[name="Lee", id="123-456-789"]/**

위의 q_1의 표현식은 간단히 말해서, PatientRecord 태그를 최상위 노드로 갖는 문서에서 환자의 이름이 "Lee"이고 식별번호가 "123-456-789"인 문서의 모든 노드정보를 요청하는 쿼리이다. 쿼리 표현식에 대한 보다 상세한 내용은 XPath[12] 연구 자료를 참조하기 바란다.

순서 2 : p_1으로부터 서비스 호출을 받은 p_2는 우선 접근통제 규칙에 정의에 따라, p_1의 쿼리 q_1에 대한 필터링 작업을 진행하게 된다. 본 시나리오에서는 p_1이 PatientRecord 노드 이하 모든 하위노드에 접근할 수 있다고 가정한다. 이 경우, 요청된 이전 쿼리 q_1은 접근통제 규칙의 필터링 작업에 의해, 새로운 쿼리 q_2로 새롭게 변형된다. 접근통제 규칙에 의한 쿼리 변형(query transformation) 과정 또한 AXML 문서의 새로운 특징이라 할 수 있다.

새롭게 변형된 q_2는 아래와 같이 표현되어 진다.

q_2 = *PatientRecord[name="Lee", id="123-456-789"]/(name ∪ ID ∪ visit/(MD ∪ diagnosis ∪ xray))*



〔그림 2〕 AXML 기반 전자의료기록 시스템 구성환경 및 서비스 호출 순서

p_2에서는 변형된 쿼리 q_2를 평가(evaluate) 하게 되며, 해당 환자에 대한 정보가 없음을 알게 되었다고 가정한다. 이 때, p_2는 이미 다른 병원 p_4 및 p_5 등에 환자기록 문서에 대해 웹 서비스 업데이트 요청을 등록한 상태였다 (1)push mode를 통한 서비스 등록).

```
<record>
   <name>Lee</name>
   <ID>123-456-789</ID>
   <visit date = 2005, April 11>
      <MD> Chul-Sun Park </MD>
      <diagnosis> vertebral f. </diagnosis>
      <xray> abseqelalsjeiiqlauxa ... </xray>
   </visit>
</record>
```

추상화 ⟶ d

```
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'>
   <EncryptionMethod Algorithm='http://www.w3.org/2000/09/xmlenc#3des-cbc'/>
   <ds: KeyInfo xmlns:de = 'http://www.w3.org/2000/09/xmldsig#'>
      <EncryptedKey xmlns='http://www.w3.org/2001/04/xmlenc#'>
         <EncryptedMethod Algorithm='http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
         <ds:KeyInfo xmlns:de='http://www.w3.org/2000/09/xmldsig#'>
            <ds:KeyName>
               Dr. Kim's Public Key
            </ds:KeyName>
         </ds:KeyInfo>
         <CipherData>
            <CipherValue>A23B45C67...</CipherValue>
         </CipherData>
         <CarriedKeyName>
            Shared Key with Dr. Kim
         </CarriedKeyName>
      <EncryptedKey>
      <ds:KeyName>Shared Key with Dr. Kim</ds:KeyName>
   <ds:/KeyInfo>
   <CipherData>
      <CihperValue>ERbGASIseiKJOOesqeuSILis.... </CipherValue>
   </CipherData>
</EncryptedData>
```

추상화 ⟶ $d_e = \{d\}_k, \{k\}_{PK(p1)}$

```
<Signature xmlns='http://www.w3.org/2000/09/xmldsig#'
   <SignedInfo>
      <SignatureMethod Algorithm='http://www.w3.org/07/xmldsig#rsa-sha'/>
   <Reference URL=''>
      <DigestMethod Algorithm = 'http://www.w3.org/2000/07/xmldsig#sha'/>
      <DigestValue>6iwwx3rvEPOSSvKMjUR2vsqUQS</DigestValue>
   </Reference>
   <SignedInfo>
   <SignatureValue>MCOFFvalSSJRsrk=....</SignatureValue>
   <KeyInfo>
      <KeyName>Integrity Key of Hospital C </KeyName>
</Signature>
```

추상화 ⟶ $d_s = \{SHA(d)\}_{Sign(p3)}$

〔그림 3〕 환자진료기록 문서 d, 암호화된 문서 d_e 및 전자서명된 문서 d_s

1) AXML 문서는 서비스 요청자가 일단 해당 웹서비스에 대한 등록을 신청하면, 서비스 요청자는 정기적으로 필요정보를 수신 받게 된다.

p_2는 p_3 병원이 해당 환자의 진료기록을 보관하고 있다는 사항을 확인하게 된다.

p_2는 p_1이 p_3에 q_2를 요청할 수 있도록 하기 위해서, q_2를 p_2의 개인키로 전자서명 하여, 결과값을 p_1에게 반환하게 된다. 즉, p_2가 p_1에게 쿼리의 사용을 위임하기 위한 과정을 나타내고 있다.

순서 3 : p_1은 p_2로부터 전자 서명된 쿼리 q_2를 수신받은 후, 다시 내장된 서비스 호출 노드(diagnosis@p_3로 표기)를 이용하여, p_3에 diagnosis 웹서비스를 호출하게 된다.

순서 4 : p_3는 p_1의 전자 서명된 쿼리 q_2를 수신한 후, B 병원의 공개키를 이용하여 쿼리 q_2가 B 병원에 의해 위임되었음을 확인하게 된다. 따라서, 쿼리 평가 후에 환자의 전자의료기록 문서 d를 암호화하고 전자서명한 문서 d_e 와 d_s를 각각 p_1에게 반환한다.

〔그림 3〕에서 보는 바와 같이, XML 언어로 표현된 환자기록 문서 d는 암호화 문서 d_e로 변경될 때, XML Encryption 표준에 따라 암호화 되게 된다. 이해를 돕기 위해 암호화된 문서 d_e의 추상화 표현 $\{d\}_k$, $\{k\}_{PK(p1)}$ 대한 의미를 설명하면 다음과 같다.

$d_e = \{d\}_k, \{k\}_{PK(p1)}$: p3는 공유키 k를 생성, 이 키로 문서 d를 암호화 한 후, p1의 공개키로 k 키를 암호화하여 p_1에게 분배(〔표 1〕 참조).

그러고, 전자서명된 문서 ds는 XML Signature 표준에 따라 표현된다. 전자서명된 문서 d_s의 추상화 표현 $\{SHA(d)\}_{Sign(p3)}$ 대한 의미를 설명하면 다음과 같다.

$d_e = \{SHA(d)\}_{Sign(p3)}$: p3는 공유키 문서 SHA 메시지 다이제스트 함수를 통해 압축한 후, Sign 함수를 이용하여 p3의 개인키로 서명.

〔표 1〕 기호 및 의미

기 호	의 미
k	p_3와 p_1의 공유키
PK	공개키 함수
Sign	전자서명 함수
SHA	메시지 다이제스트 함수

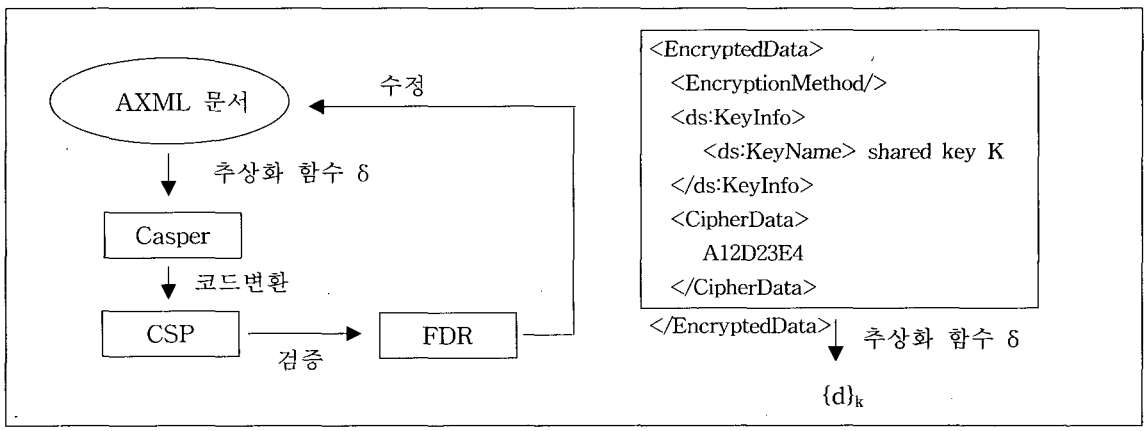
V. 보안성 분석

앞에서 미리 언급한 바와 같이, AXML 기반 전자의료기록 시스템의 보안성을 분석하기 위해 Casper, CSP 및 FDR을 이용한 정형분석 방법론을 사용하였다. 또한, AXML 문서의 웹서비스 호출 기능 및 XML Encryption과 XML Signature 표준을 이용한 데이터 암호화 특징을 고려하여 〔그림 4〕와 같은 보안성 분석 프레임워크를 제안하고자 한다.

첫째, AXML 문서가 주어졌을 때, δ 추상화 함수를 이용하여, 보안 컴포넌트를 추출한다. 예를 들어 AXML 문서의 웹서비스 호출 후, 어떤 문서 혹은 데이터 d를 공유키 k를 이용하여 XML Encryption 표준에 따라 암호화 하였을 경우, 추상화 함수 δ를 이용하면 〔그림 4〕와 같이 간략화된 추상화 보안기호로 나타낼 수 있으며, 이 기호는 공격자 모델 및 검증하고자 하는 보안요구사항 등과 함께 Casper 입력 명세로 표현되게 된다.

둘째, Casper의 자동 변환기능을 이용하여, CSP 프로세스 알제브라 코드로 변환한다.

셋째, FDR 모델체킹 도구를 이용하여, CSP 명세 모델내 AXML 문서의 보안모델이 해당 보안요구사항



〔그림 4〕 보안성 분석 프레임워크

을 만족시키는지 검증한다. 보안요구사항을 위반하였을 경우, CSP 프로세스 알제브라 기호형태의 반례를 보여주어 보안 취약점을 분석 및 수정하도록 도와준다.

1. 보안요구사항

본 논문에서는 전자의료기록 시스템 예제에서 사용된 '쿼리 위임(query delegation)'을 위함 보안 요구사항을 정의하고 분석하였다.

쿼리 위임은 웹서비스의 동적 데이터 호출을 위해, 해당 쿼리를 다른 호스트에게 위임하기 위해 사용되는 기술로 쿼리 위임자(query delegator)와 쿼리 대리인(query delegatee) 으로 구성된다.

앞에서 보여진 전자의료기록 시스템 예제에서 p_2는 쿼리 q_2를 자신의 개인키로 서명하여 p_2에게 전달하기 때문에 p_2는 쿼리 위임자이며, p1는 쿼리 대리인 역할을 수행하여 최종적으로 암호화되고 전자 서명된 환자기록 문서 $d_{es} = d_e + d_s$를 수신하게 된다.

본 논문에서 언급된 쿼리 위임 예제는 환자기록 문서 d의 기밀내용을 쿼리 대리인이 열람 할 수 있도록 허가하기 위한 목적으로 사용되었다.

따라서, 만일 다음의 두 보안요구사항을 만족시킬 경우, 본 논문에서는 쿼리 위임이 안전하게 수행되었다고 간주한다.

비밀성 : 환자기록 문서 d의 기밀내용이 p_1과 p_2를 제외한 허가받지 않은 사용자에게 알려져서는 않된다.

인증 : p_1은 p_3로부터 환자기록 문서 d를 받았음을 인증하여야 한다.

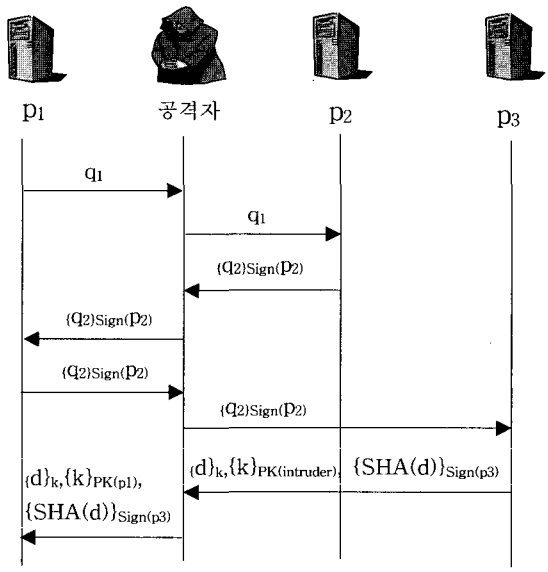
2. 공격자 모델

기존의 보안 프로토콜에서의 공격자 모델은 다음과 같은 공격 행위를 갖고 있다고 가정되어 왔다.

- 네트워크상의 전송 메세지 도청
- 정상적인 호스트로 위장하여 가짜 메시지 전송

이 더불어, 본 논문에서 정의된 공격자 모델 p_i는 다음과 같은 공격행위를 추가적으로 갖고 있다고 가정하고 있다.

- XML로 표현된 엘리먼트 분석
- 악의적인 서비스를 포함한 가짜 환자기록 문서 d'의 생성



〔그림 5〕 능동적 공격시나리오

기존의 CSP 명세에서의 공격자 모델은 도청한 메시지로부터 중요 정보를 추론하기 위해, 기본적으로 5가지 추론 규칙을 사용하였다[3].

예를 들어, B ⊢ m 표현식은 공격자가 메시지의 집합 B로부터 메시지 m을 추론할 수 있다는 규칙을 나타낸다. 따라서, 기존의 추론 규칙에서는 공격자가 공유키 k로 암호화된 메시지 $\{m\}_k$을 알고 있고, 공유키 k를 알고 있다면, 공격자는 원래 메시지 m을 추론할 수 있으며, 다음과 같은 추론규칙을 통해 표현할 수 있다.

$$B \vdash \{m\}_k \wedge B \vdash k \Rightarrow B \vdash m$$

본 논문에서는 XML Encryption 및 XML Signature 표준에 따라 표기된 문서 d에 대한 공격자 모델을 적용하기 위해 추론규칙을 확장하였다. 예를 들어, 공유키로 k로 암호화된 어떤 트리 구조형 데이터 $\{d\}_k$를 알고 있고, 공유키 k에 대한 정보를 갖고 있다면, 데이터 d를 알아 낼 수 있으며, 다음과 같은 추론 규칙을 통해 표현할 수 있다.

```
B ⊢ 〈EncryptedData〉
    〈EncryptionMethod.../〉
    〈KeyInfo〉
    〈KeyName〉shared key k〈/KeyName〉
    〈/KeyInfo〉
    〈CipherData〉
```

〈CipherValue〉A12D34E〈/CipherValue〉
〈/CipherData〉
〈/EncryptedData〉

∧ B ⊢ 〈KeyInfo〉
〈KeyName〉shared key k〈/KeyName〉
〈/KeyInfo〉
⇒ B ⊢ 〈document〉 ... 〈/document〉

3. 추상화 보안모델

앞에서 언급한 추상화 함수 δ를 전자의료기록 시스템 예제에 적용하면, 다음과 같은 추상화 보안모델을 생성할 수 있다. 추상화 암호 컴포넌트의 기호 및 의미에 대해서는 〔표 1〕을 참조하기 바란다.

메시지 교환 순서
1. p1 → p2 : q1
2. p2 → p1 : $\{q2\}_{Sign(p2)}$
3. p1 → p3 : $\{q2\}_{Sign(p2)}$
4. p3 → p1 : $\{d\}_k$, $\{k\}_{PK(p1)}$, $\{SHA(d)\}_{Sign(p3)}$

본 논문의 페이지 사정상, 4번째 메시지를 생성하는 예제를 통해 추상화 함수 δ의 활용방법에 대해 설명하고자 한다.

암호화 전자 서명된 환자기록 문서 $d_{es} = d_e + d_s$에 δ 함수를 적용하면 다음과 같은 절차에 의해 추상화 메시지 $\{d\}_k$, $\{k\}PK_{(p1)}$, $\{sha(d)\}_{Sign(p3)}$를 생성할 수 있다.

$\delta(d_e)$
⇒ δ(〈EncryptedData〉...〈/EncryptedData〉
⇒ δ(〈CipherData〉...〈/CipherData〉
⇒ δ(〈KeyInfo〉...〈/KeyInfo〉),
δ(〈EncryptedMethod.../〉)
⇒ δ(〈KeyName〉...〈/KeyName〉),
δ(〈EncryptedKey〉...〈/EncryptedKey),
δ(〈EncryptedMethod .../〉)
⇒ $\{d\}_k$, δ(〈CipherData〉...〈/CipherData〉)
⇒ $\{d\}_k$, δ(〈KeyInfo〉...〈/KeyInfo〉),
δ(〈EncryptedMethod.../〉)
⇒ $\{d\}_k$, δ(〈KeyName〉)...〈/KeyName〉),
δ(〈EncryptedMethod.../〉
⇒ $\{d\}_k$, $\{k\}_{PK(p3)}$

$\delta(d_s)$
⇒ δ(〈Signature〉...〈/Signature〉)
⇒ δ(〈SignatureValue〉...〈/SignatureValue〉)
⇒ {δ(〈SignedInfo〉...〈/SignatureValue〉)}
δ(〈KeyInfo〉...〈/KeyInfo〉)
⇒ {δ(〈SignatureMethod../.〉),
δ(〈Reference...〉...〈/Reference〉)}
δ(〈KeyInfo〉...〈/KeyInfo〉)
⇒ {δ(〈DigestMethod〉...〈/DigestMethod〉)}
δ(〈KeyName〉...〈/KeyName, Sig)
⇒ $\{SHA(d)\}_{Sign(p3)}$

∴ d_{es} ⇒ δ(de) + δ(d_s) ⇒ $\{d\}_k$, $\{k\}_{PK(p3)}$, $\{SHA(d)\}_{Sign(p3)}$

4. 보안요구사항 분석 결과

앞의 단원에서 언급한, 보안모델, 보안요구사항 및 공격자 모델을 Casper를 이용하여 추상화 모델로 표현한 후, CSP 코드로 변환한 다음, FDR 도구에 입력하여 보안요구사항에 대한 모델체킹을 실행한 결과 공격 시나리오를 발견할 수 있었다.

순서 1 : 공격자는 p_1과 p_2 호스트 사이의 메시지를 도청하여, 쿼리 q_1에 대한 응답 메시지인 $\{q_2\}_{Sign(p2)}$를 가로챈다.

순서 2 : p_3 호스트로 위장하여, p_1으로 부터의 쿼리 메시지 $\{q_2\}_{Sign(p2)}$를 가로챈 후, 웹서비스 호출을 통해 쿼리 메시지를 송신한다.

순서 3 : 쿼리 메시지를 수신한 후, p_3는 공격자 intruder의 식별자를 확인하게 된다. 하지만, p_3는 최초 웹서비스의 출발 호스트가 누구인지 알 수 있는 방법이 없기 때문에, Sign(p2) 값을 확인한 후, 공격자 intruder가 p_2로부터 q_2를 사용하도록 위임받았다고 믿게 된다. 따라서, 환자기록 문서 d를 자신이 생성한 공유키 k로 암호화하고 키 k를 공격자의 공개키 PK(intruder)로 암호화하여 공격자에게 전달할 수 있게 된다.

순서 4 : 공격자는 자신의 개인키로 공유키 k를 알아내고, 이 복호화 키를 이용하여 이와 더불어

환자기록 문서 d의 내용을 획득하게 된다. 이때, 만약 공격자가 문서 d의 내용을 변경하고 악의적인 내장형 웹서비스 호출노드를 포함한 문서 d'를 생성하여 공유키 키로 암호한 메시지 $\{d'\}_k, \{k\}_{PK(p1)}$를 p_1에게 전송하게 되고, p_1이 문서의 변경유무를 확인하지 않는다면, 공격의 피해범위는 보다 확산될 수 있다. 예를 들어, 변경 문서 d'가 순환 웹서비스 호출을 유도하여 p_1 호스트의 중요 자원처리 프로세서에 과부하를 유발할 수 있다. 혹은 변경된 환자의 진료카드 내용으로 인하여 p_1 (의사 "김")에게 환자에 대한 잘못된 의료진단 정보를 제공할 수 있게 된다.

이런 보안 취약점을 미연에 방지하기 위해서는 기본적으로 위임된 쿼리 메시지에 타임 스탬프를 표시하여 공격자의 쿼리 재사용 공격을 차단하고, P1은 위임된 쿼리 메시지 $\{q_2\}_{Sign(p2)}$에 자신의 서명을 추가한 $\{\{q_2\}_{Sign(p2)}\}_{Sign(p1)}$을 p_3에 송신하여 웹서비스를 호출한 최초 사용자를 확인할 수 있도록 하는 방법을 채택하는 것이다.

Ⅵ. 관련연구

프랑스의 INRIA Futurs 연구소는 지난 3년간 웹서비스의 동적 데이터 검색 및 통합기능을 가능하게 하는 AXML 문서를 개발하고 그 기능을 확장하는 연구를 진행해 오고 있다[13-16].

AXML 문서는 기본적으로 스키마 레벨의 데이터 타입 보안에 의존하고 있다. 예를 들어, 웹서비스를 호출한 한 호스트가 해당 데이터 결과로 AXML 문서를 수신하였을 경우 결과 데이터 노드 값이 스키마에 미리 정의되어 있는 데이터 타입과 일치하는지 여부를 결정하는 방법이다.

스키마 레벨의 데이터 타입 보안은 궁극적으로 수신 데이터에 대한 암호화 및 인증 기능등을 제공하지 못하여 데이터 비밀성, 인증 및 접근통제 에 대한 보안문제를 야기 시킬 수 있다.

이에 따라, AXML 문서의 비밀성 및 접근통제에 대한 보안연구가 처음으로 진행되었다[17]. 또한 AXML 문서에 적합한 새로운 접근통제모델을 통해, 허가받지 않은 사용자로부터 비밀정보 데이터를 보호하는 기술을 제안되었다[18].

기존 보안 프로토콜의 보안성 검증에 대한 연구는 웹서비스의 중요성과 더불어, XML 보안 및 웹서비스 프로토콜 보안에 대한 검증 방향으로 연구가 점차 확장되기 시작하고 있다.

Eldar Kleiner은 SOAP 채널 보안 표준인 WS-Security 명세를 Casper 입력으로 자동변환하고 FDR 도구를 이용하여 보안성을 검증하는 방법을 제시하였다[19-20].

Llanos Tobarra는 Casper 및 FDR 도구를 이용하여, 마이크로소프트사에서 제안된 SOAP 채널 보안 표준인 Web Services Enhancement (WSE)[21]의 보안 취약점을 분석하였다[22].

Bhargavan는 pi-calculus 정형명세 언어를 이용하여 SOAP 기반 보안 프로토콜을 명세하고 보안 취약점을 검증할 수 있는 도구 TulaFale를 개발하였다[23].

본 논문의 주된 연구내용은 WS-FM'06 학회에 발표되었으며[24], 본 특집호에서는 최근 웹서비스 환경에서 활성화되고 있는 전자의료기록 시스템 의 시나리오 및 보안 문제점을 소개하도록 작성되었다.

Ⅶ. 결 론

본 논문에서는 AXML 문서의 내장형 웹서비스 호출과 XML Encryption 및 XML Signature 보안 표준을 통합하여, 수신 데이터에 비밀성 및 무결성을 보장하는 보안 메커니즘에 대해 설명하였다. 또한, AXML 기반 "전자의료기록 시스템" 예제를 통해 동적 환자진료 문서에 대한 웹서비스 호출 절차 및 쿼리 위임 시나리오에 대해 소개하였다. 뿐만 아니라, 기존 보안 프로토콜에 적용되어온 보안성 정형분석 방법을 확장하여, AXML 보안 시스템 예제에 적용해 본 결과, 분산 서비스 환경에서 쿼리 위임 방식에 의해 허가받지 않은 사용자가 중요정보에 접근할 수 있는 보안 취약점을 발견할 수 있었다.

참 고 문 헌

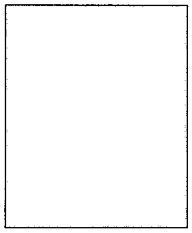
[1] Active XML (AXML) Home Page, http://activexml.net, 2003.

[2] IBM, Microsoft, and VeriSign, Web Services Security (WS-Security), Version 1.0, 2002.

[3] G. Lowe, "Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR", Proceedings of TACAS,

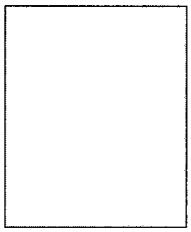
number 1055 in LNCS. Springer, 1996.
[4] 김일곤, 최진영, 김현석, 전철욱, 강인혜, "보안 프로토콜 안전성 분석을 위한 정형적 방법론", 정보보호학회 논문지 : 제15권, 제1호, pp.17~27, 2005.
[5] G. Lowe, "Casper: A compiler for the analysis of security protocols", 10th IEEE Computer Security Foundations Workshop, 1997.
[6] C.A.R. Hoare, Communicating Sequential Processes, Prentice-Hall, 1985.
[7] Formal Systems(Europe) Ltd, FDR2 User Manual, 1999.
[8] P. Ryan, S. Schneider, "Modelling and Analysis of Security Protocols", Addison-Wesley, 2001.
[9] D. Eastlake, J. Reagle, T. Imamura, B. Dillaway, E. Simon, XML-Encryption syntax and Proceeding, W3C Recommendation, 2001.
[10] D. Eastlake, J. Reagle, D. Solo, M. Bartel, J. Byer, B. Fox, B. LaMacchia, E. Simon, XML-Signature Synatax and Proceedings, W3 Recommendation, 2002.
[11] S. Abiteboul, I. Manolescu, F. Taropa, "A Framework for Distributed XML Data Management", Proceedings of EDBT, pp.1049~1058, 2006.
[12] XML Path language (XPath) Version 1.0. W3C Recommendation, http://www.w3c.org/TR/xpath, 1999.
[13] S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo, R. Weber, "Active XML: Peer-to-peer data and web services integration(demo)", Proceedings of 28th VLDB, pp.1087~1090, 2002.
[14] S. Abiteboul, O. Benjelloun, B. Cautis, I. Manolescu, T. Milo, N. Preda, "Lazy Query Evaluation for Active XML", Proceedings of ACM SIGMOD, pp. 227~238. 2004.
[15] S. Abiteboul, O. Benjelloun, T. Milo, "Positive Active XML", Proceedings of ACM PODS, 2004.
[16] S. Abiteboul, I. Manolescu, F. Taropa, "A Framework for Distributed XML Data Management", Proceedings of EDBT, pp.1049~1058, 2006.
[17] S. Abiteboul, O. Benjelloun, B. Cautis, and T. Milo, "Active XML, Security and Access Control", Proceedings of SBBD, pp.13~22, 2004.
[18] S. Abiteboul, B. Alexe, O. Benjelloun, B. Cautis, I. Fundulaki, T. Milo, and A. Sahuguet, "An Electronic Patient Record on Steroids : Distributed, Peer-to-Peer, Secure and Privacy-conscious", Proceedings. of 30th VLDB, pp.1273~1276, 2004.
[19] E. Kleiner, A.W. Roscoe, "Web Services Security: a Preliminary Study using Casper and FDR", Proceedings of ARSPA, 2004.
[20] E. Kleiner, A.W. Roscoe, "On the Relationship between Web Services Security and Traditional Protocols", Proceedings of DIMACS Workshop on Security of Web Services and E-Commerce, 2005.
[21] Microsoft, Microsoft Web Services Enhancements (WSE) 2.0, http://msdn.microsft.com/webservices/webservices/building/wse/default.aspx
[22] L. Tobarra, D: Cazorla, F. Cuartèro, Gregorio Diaz, "Application of Formal Methods to the Analysis of Web Services Security", Proceedings of International Workshop on Web Services and Formal Methods. pp.215~229. 2005.
[23] K. Bhargavan, C. Fournet, A. D. Gordon, R. Pucella, "TulaFale: A security tool for web services", In International Symposium on Formal Methods for Components and Objects (FMCO'03), pp.197~222. 2004.
[24] I. Kim, D. Biswas, "I. Kim and D. Biswas, Application of Model Checking to AXML System's Security : A Case Study," Proceedings. of International Workshop on Web Services and Formal Methods (WS-FM'06), pp.242~256, 2006.

〈著者紹介〉



김 일 곤 (Il-Gon Kim)

2000년 2월 : 경기대학교 영어영문학과 졸업
2002년 2월 : 고려대학교 컴퓨터학과 석사 졸업
2005년 2월 : 고려대학교 컴퓨터학과 박사 졸업
2005년~현재 : INRIA-IRISA 연구소 postdoc
관심 분야 : 소프트웨어 공학, 보안 프로토콜, 웹서비스 보안, 데이터베이스



Debmalya Biswas

2000년 : 인도 Amravati대학 컴퓨터학과 졸업
2003년 : 인도 Infosys Technologies사의 소프트웨어 엔지니어
2005년 : 캐나다 Memorial 컴퓨터학과 졸업
2002년~현재 : INRIA-IRISA 연구소 박사과정
관심 분야 : 소프트웨어 공학, 데이터 베이스, petri-net, 웹서비스