OSGi 환경에서 XML 전자서명을 이용한 번들 인증

임희영° 김영갑
고려대학교 컴퓨터학과 소프트웨어 시스템 연구실
{tonight24°, ygkim}@software.korea.ac.kr

Bundle Authentication using XML Signature in the OSGi Service Platform

Hee-Young Lim° Young-Gab Kim
Software System Lab. Dept. of Computer Science and Engineering, Korea University

요 약

현재 XML은 데이터의 표현에 있어서 가장 유연하고 독립성 있는 언어로 자리잡고 있으며 다양한 분야에서 활용되고 있다. 이러한 추세에 따라 OSGi 서비스 플랫폼 환경에서도 여러 서비스들이 자신의 데이터를 XML로 표현하고 있으며, 이를 기반으로 서비스 번들의 인증을 XML 전자 서명을 생성하여 수행할 수 있다. 현재 OSGi 에서는 PKI기반 서비스 번들 인증 메커니즘을 이용하고 RSH 프로토콜을 권고하고 있다. 그러나 저장 공간이나 연산이 제한된 환경에 있는 OSGi플랫폼 내에서 작동하는 데에는 어려움이 있다. 따라서 본 논문에서는 JAR파일로 제공되는 서비스 번들에 대해 별도의 연산이나 인증과정 없이 XML 전자서명을 생성하여 서비스 번들을 인증하는 메커니즘을 제시한다.

1. 서 론

OSGi(Open Services Gateway Initiative)는 개방형 서비스 게이트웨이의 표준을 지향하는 업체들이 모여 만든 표준화 단체로, 전 세계적으로 퍼져있는 컴퓨터 위주의 인터넷을 가정의 가전제품과 점검기기들에 연결하여 다양한 서비스를 제공받을 수 있게 하고 이를 확산시키기 위한 목적으로 조직된 그룹이다.

이러한 OSGi 플랫폼 환경의 서비스는 기존의 네트워크 서비스와는 달리 번들이라는 자체 설치가 가능한 컴포넌트 형태로 제공되어 동적으로 배치되며, 다른 서비스와의 상호 작용도 자주 일어난다. 이러한 특성들로 인하여 네트워크상에서 인증되지 않은 오퍼레이터에 의해 악의적인 서비스가 배치되거나 서비스가 변질 될 위험이 있다. 현재 OSGi 서비스 플랫폼 릴리즈 3에서는 PKI(Public Key Infrastructure)[1] 기반 서비스 번들 인증 메커니즘과 RSH(Remote Communication in a Secure way based on HTTP)[2] 프로토콜을 사용할 것을 권고하는 기본적인 보안 모델 방향을 제시하고 있으나, 저장 공간이나 연산이 제한된 자원을 가지고 있는 OSGi 서비스 플랫폼에서 작동하는 데에는 성능의 저하가 예상된다. 따라서 본 논문에서는 이러한 한계점을 극복하기 위하여 XML 전자서명[3]을 이용한 서비스 번들 인증 메커니즘을 제안한다.

다양한 요구사항에 대한 효율적인 데이터의 조작과 처리를 충족시키기 위하여 등장한 XML(eXtensible Markup Language)는 이제 새로운 차세대 언어로 자리를 잡아가고 있으며 이러한 추세에 발맞추어 OSGi 서비스 플랫폼 릴리즈 3.0[4]에서도 XML Parser Service 1.0을 명세하고 있다. 따라서 본 논문에서는 기존 연구로 진행된 서비스 제공자와 서비스 게이트웨이의 상호 인증 메커니즘[5]에서 생성된 키를 이용하여 서비스 번들에 대한 XML 전자서명을 생성해 서비스를 인증하는 방법을 제안한다.

2. 관련 연구

2.1 OSGi

2.1.1 OSGi 서비스 플랫폼

OSGi 서비스 플랫폼의 전체적인 구조는 그림 1과 같다.

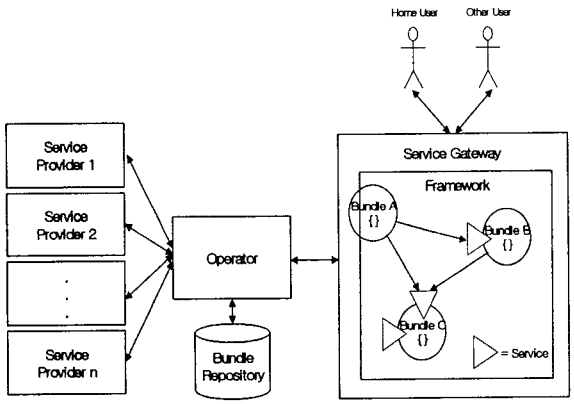

그림 1 OSGi 서비스 플랫폼 구조

서비스 제공자(Service Provider)는 가정에 제공할 서비스를 제공하는 전문 업체를 의미한다. 오퍼레이터(Operator)는 독립적인 서비스 업체의 형태로 존재하면

서 각 가정의 서비스 게이트웨이와 서비스를 관리하는 전문 업체이다. 그리고 서비스 게이트웨이(Service Gateway)는 각 가정에 설치되어 있는 홈 게이트웨이를 의미한다. 사용자는 서비스 게이트웨이를 사용하는 주체로서 다양한 디바이스와 유무선 인터넷을 이용하여 서비스를 사용할 수 있다.

2.1.2 OSGi 보안 구조[6]

서비스 번들의 안전한 전송을 위해서 장치를 인식하고 초기 설정을 하는 부트스트래핑 단계에서 오퍼레이터는 특정 서비스 게이트웨이를 인증해야한다. 그러나 서론에서도 언급했듯이 오퍼레이터와 서비스 게이트웨이의 상호 인증 및 서비스 번들 인증에 대한 구체적인 방안은 제시되지 않았다.

현재 OSGi 에서는 PKI기반 서비스 번들 인증 메커니즘을 이용하고 RSH 프로토콜을 권고하고 있다. 그러나 PKI기반 서비스 번들 인증의 경우 공개키 연산뿐 아니라 인증서의 유효성 검사도 수행해야 하므로 저장 공간이나 연산이 제한된 환경에 있는 OSGi플랫폼 내에서 작동하는 데에는 어려움이 있다. 또한 RSH프로토콜은 MAC(Message Authentication Code)[7]을 이용하여 오퍼레이터와 서비스 게이트웨이 사이에 전송되는 모든 데이터를 암호화하고 인증하므로 서비스 번들의 크기에 따라 성능의 저하가 예상된다. 따라서 본 논문에서는 이러한 한계점을 극복하기 위하여 XML 전자서명을 이용한 서비스 번들 인증 메커니즘을 제안한다.

2.2 XML 전자서명

2.2.1 XML 전자서명의 개요

전자서명이란 전자화 된 문서의 메시지 내용이 수정 및 변조되지 않았음을 보장하는 동시에 메시지를 보내고 받는 이가 올바른 사용자라는 것을 확인 할 수 있도록 하는 인증 방식을 말한다.

XML 서명은 무결성(Integrity), 메시지 인증(Message Authentication), 서명자 인증(Signer Authentication) 기능을 제공한다. XML 서명은 특정 문서 전체에 대한 서명 또는 부분에 대한 서명이 가능하다. 뿐만 아니라 하나의 XML서명으로 텍스트 데이터, 그림과 같은 바이너리 데이터 등 여러개 자원에 서명을 할 수 있는 장점을 가진다. 또한 XML의 경우 문서에 수신자가 생성한 다이제스트와 서명 값이 포함되어 있으므로 데이터를 메시지와 서명으로 분리하여 다이제스트 값을 계산할 필요가 없다.

2.2.2 XML 전자서명 기본 구조

기본 적인 구조는 그림 2와 같다.

서명해야 하는 각각의 데이터는 <Reference> 엘리먼트에서 URI로 지정한다. <Transform> 엘리먼트에서는 참조하는 데이터에 대한 다이제스트를 만들기 전에 해야하는 작업을 나열하며, <DigestValue> 엘리먼트는 데이터의 실제 다이제스트 값을 표현한다. <SignedValue> 엘리먼트에서는 <SignedInfo> 엘리먼트의 다이제스트 값을 암호화 한 값을 표현하며 <KeyInfo> 엘리먼트에는 서명을 검증할 키에 대한 정보를 입력한다. 이 논문에서는 기존에 연구되었던 키공유 메커니즘을 이용하여 공유 비밀키를 생성하기 때문에 <KeyInfo> 엘리먼트는 이 메커니즘에 의해 생성된 키 정보를 가진다.

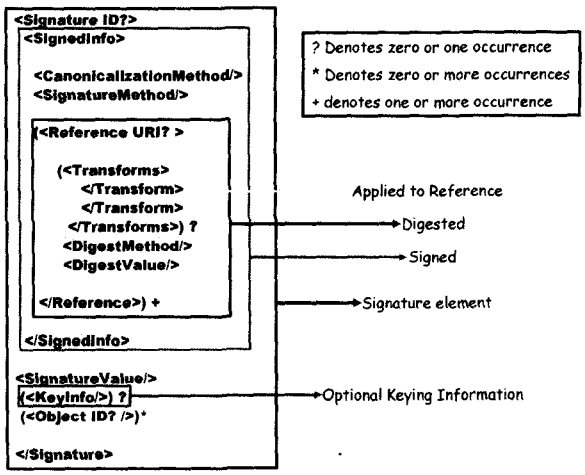

그림 2 XML 전자서명의 구조

3. 서비스 번들 인증 메커니즘

3.1 XML 전자서명 기반 서비스 인증 메커니즘

오퍼레이터와 서비스 게이트웨이 사이의 서비스 번들의 제공은 서명 된 JAR파일[8]에 대한 서명을 검증하여 이루어진다. 일반적인 JAR 서명 코드는 개발자의 개인키를 이용하여 작성되고, 개발자의 공개키와 인증서를 기반으로 검증된다. 그러나 이러한 인증은 제한된 시스템 자원을 가지고 있는 서비스 플랫폼 상에서는 부적합하다. 이러한 점을 고려하여 오퍼레이터와 서비스 플랫폼 간에 공유하고 있는 공유키를 이용하여 서비스 번들에 대한 XML 전자서명을 생성하는 것이 효율적이다. 앞에서 언급했던 메시지 무결성과, 메시지 인증을 만족하며 서명자 인증 기능까지 제공하는 동시에, 추가적인 연산이나 인증기관과의 유효성 검사도 필요없는 간단한 메커니즘으로 인해 서비스 인증이 효율적으로 이루어 질 수 있다.

그림 3는 XML 전자서명 기반 서비스 번들 인증 메커니즘은 나타낸다. 즉, 서비스 게이트웨이는 오퍼레이터로부터 XML 전자서명이 추가된 JAR 파일을 전송받고, 전자서명에 명시된 알고리즘과 공개키 정보를 사용하여 JAR파일을 검증한다. 이때 XML 전자 서명은 JAR파일 전체에 대한 전자 서명이 아니라 JAR파일 안에 있는 Menifest 파일에 대한 전자서명이다. 그렇기 때문에 서비스 제공자가 제공하는 JAR 파일 전체에 대한 전자 서명을 생성하는 것보다 훨씬 간편하게 전자 서명을 생성 할 수 있다. 그림 3에 표현된 서비스 인증과정을 자세히 살펴보면 다음과 같다.

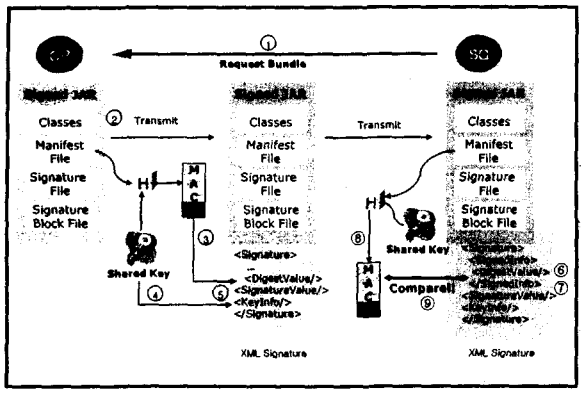

그림 3 XML 전자 서명을 이용한 서비스 번들의 인증

①서비스 게이트웨이가 오퍼레이터에게 서비스 번들을 요청하면 서비스 제공자는 요청한 서비스를 제공한다.

②서비스 제공자가 제공하는 서비스에 대해서 XML 전자서명을 생성하기 위해 JAR 파일들의 위치를 지정하는데, 이때 <Transform> 엘리먼트에서 JAR파일의 압축을 푸는 알고리즘은 지정한다. 따라서 <Reference> 엘리먼트의 URI는 JAR 파일의 Manifest를 지정할 수 있다.

③Manifest 파일의 다이제스트를 계산하여 <DigestValue> 엘리먼트에 기록한다. 이때 사용한 알고리즘은 <DigestMethod> 에 명시한다.

④이미 가지고 있던 공유 비밀키를 이용하여 전자 서명을 생성하고 <SignedValue> 엘리먼트에 기록한다.

⑤서명된 문서를 서비스 게이트웨이에서 받아 인증하기 위하여 키에 대한 정보를 제공한다. 여기서의 키는 부트스트래핑 단계에서 키 공유 메커니즘은 통하여 생성된 것이다. 이 정보를 <KeyInfo> 엘리먼트에 기록한다.

서명이 전달되면 이를 검증하는 방법을 <SignedInfo>에 있는 서명을 검증하는 것이다. 이 검증은 두개의 필수 프로세스로 구성되는데 <SignedInfo>에 대한 검증과 그 자식 엘리먼트인 <Reference>의 다이제스트 값에 대한 검증으로 구성된다.

⑥<SignedInfo> 엘리먼트의 다이제스트를 <SignatureMethod>에 명시된 알고리즘으로 계산한다.

⑦<SignatureValue>에 있는 전자서명 값을 <KeyInfo>에 있는 공개키를 이용하여 해독한 후 ⑥번의 값과 동일한지 확인한다.

⑧<SignedInfo>의 자식 엘리먼트의 <Reference>엘리먼트가 참조하고 있는 Menifest의 다이제스트를 계산한다.

⑨⑧번 값을 <DigestValue>값과 동일한지 검토한다.

서명이 완료된 형태는 그림 4와 같다. JAR파일 전체에 대한 서명이 아닌 Manifest 파일에 대한 전자 서명을 생성하므로써 제한된 자원의 서비스 플랫폼 상에서 보다 효율적으로 서비스 번들에 대한 인증을 수행할 수 있다.

```
<Signature Id="MyFirstSignature"
xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
  <CanonicalizationMethod
   Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-0010315"/>
  <SignatureMethod
   Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
  <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
   <Transforms>
   <Transform Algorithm="http://www.w3.org/TR/2001/
    REC-xml-c14n-20010315"/>
   </Transforms>
   <DigestMethod Algorithm="http://www.w3.org/2000/09/
    xmldsig#sha1"/>
   <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
  </Reference>
 </SignedInfo>
 <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
 <KeyInfo>
  <KeyValue>
   <DSAKeyValue>
    <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
   </DSAKeyValue>
  </KeyValue>
 </KeyInfo>
</Signature>
```

그림 4 XML 전자 서명

4. 결론 및 향후 연구

지금까지 XML 전자서명을 이용한 서비스 인증 과정에 대하여 살펴보았다. OSGi 환경에서 제한된 시스템에서의 PKI 기반 인증이나 RSH프로토콜 사용으로 인한 성능 저하 문제를 XML 전자 서명을 사용하여 해결 할 수 있었다. XML 전자 서명을 이용하므로써 추가적인 연산이나 인증기관과의 연동 없이 무결성, 메시지 인증, 서명자 인증 기능을 제공 할 수 있었고, 제공된 JAR파일에서 Manifest 파일에 대한 전자 서명을 생성하여 보다 간단하고 효율적으로 서비스 번들을 인증할 수 있었다.

향후 연구로는 제안한 인증 방법을 이용하여 실제적인 구현을 통해 검증하는 작업이 필요하며, 기존에 연구된 서비스 번들 인증 방법들과의 비교를 통하여 메커니즘을 좀 더 개선하고 확장하는 작업이 필요하다.

참고문헌

[1] Marc Branchaud, "A Survey of Public Key Infrastructures", Department of Computer Science, McGill University, Montreal, 1997.

[2] OSGi, "Secure Provisioning Data Transport using Http", RFC36, http://www.osgi.org/ , 2002.

[3] Blake Dournaee, "XML Security", Osborne, 2002

[4] OSGi, OSGi Service Platform, Release 3, http://www.osgi.org, 2003

[5] 김영갑, 문창주, 박대하, 백두권 , "OSGi 서비스 프레임워크 환경에서의 서비스 번들 인증 메커니즘", 정보과학회지, 제 29권, 제1호, page 868-870, 2002.

[6] OSGi, "RFC 18 - Security Architecture Specification" Draft, http://www.osgi.org/member, 2001.

[7] William Stallings , "Cryptography and Network Security" , Pearson Education, 2002.

[8] Sun, JAR Feature , http://java.sun.com/j2se/1.4/docs/guide/jar/, 2001.