OSGi 서비스 플랫폼 기반의 인증 기법에 관한 연구

(A Study on Authentication Technique based on OSGi Service Platform)

이 창 욱 † 홍 원 기 †† 장 훈 †††
(Changuk I) (Wongi Hong) (Hoon Chang)

요 약 OSGi 서비스 플랫폼 환경에서는 개방된 게이트웨이에 대한 정당하지 않은 사용자의 침입과 허가되지 않은 자원의 접근과 같은 보안 취약성이 발생하므로 보안 아키텍처의 구축이 필수적이다. 본 논문에서는 OSGi 서비스 플랫폼 환경의 하드웨어적인 제약을 고려하여 대칭키를 이용한 자동화된 사용자 인증과 서비스 번들 인증 기법을 제안한다. OSGi 플랫폼 환경은 디바이스들의 입력 수단에 제약이 있으므로 패스워드 입력 과정을 생략하고 MAC address와 암호화된 식별자를 사용한 자동화된 사용자 인증 메커니즘을 연구하였다. 또한 서비스 번들을 위한 인증 메커니즘에는 대칭키를 이용하여 연산이 빠르고 시간 정보와 발급 정보가 담긴 티켓을 이용하여 안전한 인증 과정을 거칠 수 있도록 하였다. 이러한 두 가지 인증 메커니즘을 바탕으로 자원의 제약성을 해소하여 사용자와 관리자의 편의성을 높이고 패스워드 입력 대기 시간의 생략과 보안 인증에 필요한 연산을 줄여 OSGi 서비스 플랫폼 환경에서 효과가 있다고 분석하였다.

키워드 : OSGi, 보안, 사용자 인증, 번들 인증

Abstract The establishment of security architecture is essential because security vulnerabilities occur such as user's unjustifiable connection for the opened gateway and access to resources without permission in OSGi service platform environment. In this paper, it proposes a authentication technique for an Automatic user authentication which is used the Symmetric Key and the Service bundle authentication to consider the constraints of the hardware in the OSGi service platform environment. Typically, the type of entering a password is used for the user authentication mechanism however OSGi platform environment studies not entering the password but using MAC address and encrypted identifier of the automatic user authentication mechanism because the devices are limited in their input. In this paper, the Symmetric Key is used for bundle authentication mechanism. Therefore operation becomes quick and secure authentication process has been successfully completed by using the time data and a ticket which contains a license. Based on these two different authentication mechanisms, it could eliminate the constraints of resources and improve the convenience of users and administrators. Also it shows an effect from omitting the waiting time to enter a password and reducing operations which need for authentication in the OSGi service platform environment.

Key words : OSGi, Security, User authentication, Bundle authentication

† 정 회 원 : 숭실대학교 컴퓨터학과
changuk@ssu.ac.kr
†† 학생회원 : 숭실대학교 컴퓨터학과
ustino@esrl.ssu.ac.kr
††† 정 회 원 : 숭실대학교 컴퓨터학부 교수
hoon@ssu.ac.kr
논문접수 : 2008년 12월 31일
심사완료 : 2009년 5월 28일

정보과학회논문지: 시스템 및 이론 제36권 제5호(2009.10)

1. 서 론

OSGi(Open Service Gateway Initiative)[1]는 서비스를 로컬 네트워크의 장비에게 전달하고 전달된 서비스가 운용되는 개방형 스펙을 만드는 그룹이다. 일반적으로 홈 네트워크와 같은 내장형 시스템(embedded system)은 각기 다른 적용 분야와 통신방식에 의해서 구성되기 때문에 다양한 네트워크 환경을 모두 수용하기에 어려움이 있다. 이러한 문제점을 해결하기 위해서 OSGi 개방형 스펙을 이용한 게이트웨이 구성을 고려할 수 있다[2]. 하지만 OSGi 서비스 플랫폼 환경은 개방된 게이트웨이에 대한 정당하지 않은 사용자의 침입과 허가되지 않은 자원에 대한 접근과 같은 보안 취약성이 발생한다. 따라서 OSGi 환경에서는 인증 및 권한부여와 같은 보안 아키텍처의 구축이 필수적이다. 현재 학계에서는 OSGi 플랫폼 환경을 위한 향상된 보안 모델에 대한 연구가 진행 중이다[3]. 그리고 현재 OSGi 스펙에서는 보안 아키텍처에 관한 구체적인 명세화가 없기 때문에 OSGi 환경을 위한 사용자 인증 메커니즘과 번들 인증 메커니즘의 명세화가 필요하다[4].

OSGi 플랫폼 환경의 특성상 사용자 인증 서버와 서비스를 제공하는 번들 서버(bundle server) 그리고 게이트웨이를 각각의 물리적인 장비로 나누는 것은 시스템 개발의 복잡성을 초래할 수 있다. 또한 시스템 구현 비용에 대한 부담이 증가하므로 이러한 기능들을 하나의 서비스 게이트웨이에 통합하여 구성하는 것이 바람직하다. 따라서 본 논문에서는 인증 서버와 번들 서버를 서비스 형태로 구현하여 서비스 게이트웨이에 배치하였다.

그림 1은 서비스 게이트웨이에 인증 서비스와 번들 서비스를 구현한 OSGi 서비스 플랫폼 환경의 구성도이다. 로컬 네트워크에는 컴퓨터와 PDA, 스마트폰 또는

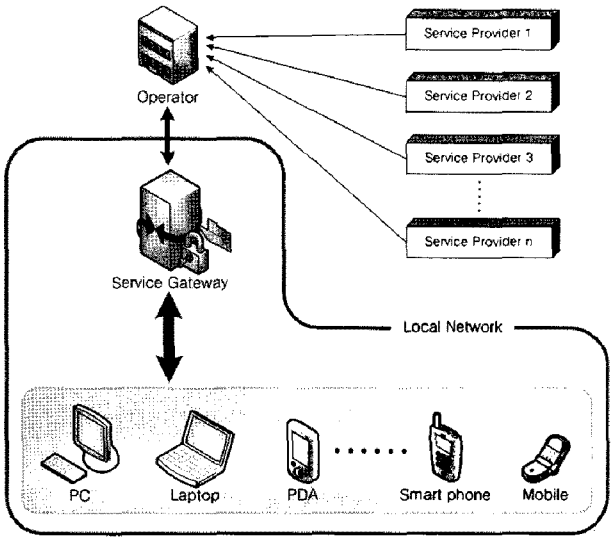


그림 1 OSGi 서비스 플랫폼 구성도

정보가전까지 다양한 이기종 디바이스들로 이루어지며 서비스 게이트웨이에는 OSGi 프레임워크에 인증 서버와 번들 서버의 기능을 번들 서비스 형태로 구현하였다. 오퍼레이터는 서비스 게이트웨이의 요청을 받아 필요한 번들 서비스를 서비스 제공자에게 받아서 배포하는 것이다.

본 논문에서는 OSGi 서비스 플랫폼을 위한 두 가지 인증 메커니즘을 제안한다. 첫 번째는 사용자 인증 메커니즘이다. OSGi 서비스 플랫폼은 일반적으로 소규모의 네트워크 구조로 구성되며 사용자의 디바이스에 문자 입력에 대한 제약이 따를 수 있다. 이러한 특성을 고려하여 기존의 패스워드 입력 방식이 아닌 사용자의 접속에 따라서 패스워드 입력 없이 인증 과정을 거치는 자동화된 사용자 인증 메커니즘을 제안한다.

두 번째는 OSGi 프레임워크의 번들에 대한 인증 메커니즘이다. OSGi 프레임워크의 번들이란 서비스를 제공하기 위한 배포단위로써 여러 서비스를 하나의 패키지로 묶은 JAR 파일[5]의 형태로 이루어져 있다. OSGi 프레임워크에서는 번들의 수명 주기(life cycle)를 관리하여 새로운 번들의 설치, 삭제, 실행 및 정지 등의 작업을 해주는데, 이러한 특성에 따라 번들은 서비스 게이트웨이 관리자에 따라서 동적으로 게이트웨이에 배치된다. 번들이 게이트웨이에 배치되는 과정에서 허락되지 않은 오퍼레이터의 접근으로 인한 불법적인 번들의 설치와 침입에 의한 번들의 변질 등의 문제가 발생될 수 있다. 따라서 번들이 배포되는 과정에서 번들 인증 메커니즘이 필요하다. 기존에 연구된 번들 인증 메커니즘은 XML(extensible markup language)을 이용한 번들 인증 방법이 있다[6].

현재 OSGi에서 권고하고 있는 서명된 JAR[7]와 PKI (Public Key Infrastructure)[8] 기반의 번들 인증 메커니즘은 OSGi 플랫폼 환경에 적용하기에는 어려움이 있다. 그 이유는 복잡한 연산 과정에 의해 자원의 소모가 많기 때문이다. 따라서 본 논문에서는 연산이 빠르고 소규모 네트워크 구조에 적합한 대칭키 기반의 번들 인증 메커니즘을 제안한다.

2. 사용자 인증 메커니즘

OSGi 서비스 플랫폼 기반의 네트워크 구조에서는 서비스 제공자와 디바이스들의 제약성을 고려하여야 한다. 이기종간의 적용 분야와 통신 방식에 대해서 일관된 서비스를 제공하기 위해 OSGi 서비스 게이트웨이가 그 역할을 담당한다. 일반적으로 기존의 보안 메커니즘은 사용자에게 디바이스의 식별자와 패스워드를 입력받아 인증 서버에 그 정보를 전달하여 정당한 사용자임을 판별하는 방식으로 구성된다. 그러나 무선 네트워크 환경에서 입력 수단이 제한적인 소형 디바이스에서는 사용

자가 접속을 원할 때마다 패스워드를 묻는 이러한 방식의 적용이 효율적이지 못하다.

본 논문에서는 자동화된 사용자 인증 메커니즘을 위하여 Kerberos 프로토콜[9]을 변형하여 적용하였다. Kerberos 프로토콜은 가장 기본적인 키(Key)를 이용한 암호화 방법으로써 인증(Authentication), 무결성(Identity), 데이터보안(Privacy)을 제공한다. Kerberos 프로토콜은 키를 배포하는 역할을 수행하는 KDC(Key Distribution Center)와 서비스를 제공받기 위한 티켓을 발행하는 TGS(Ticket Grating Service)로 구성되어 있다.

OSGi 플랫폼 환경은 불특정다수가 접근하는 웹서버와 달리 정해진 수의 클라이언트와 한정적인 공간에서 서비스가 이루어진다. 이러한 OSGi 환경에 맞추어 Kerberos 프로토콜의 KDC, TGS와 같은 서비스를 번들로 구현하여 하나의 서버에 기능을 통합하여 구현한다면 서버 구축의 효율성을 높일 수 있다. 따라서 본 논문에서는 전반적인 인증 서비스를 하나의 게이트웨이 서버에 번들 형태로 구성하여 OSGi 플랫폼 환경을 구축하였다.

그림 2는 본 논문에서 제안하는 Kerberos 프로토콜을 변형한 자동화된 사용자 인증 메커니즘의 순서도이다. 사용자 인증 메커니즘은 서비스 게이트웨이 인증 서비스의 사용자 인증 과정과 TGS의 서비스 티켓 발급 과정, 그리고 번들 서비스의 권한 부여 과정까지 총 3단계의 과정으로 이루어져 있다.

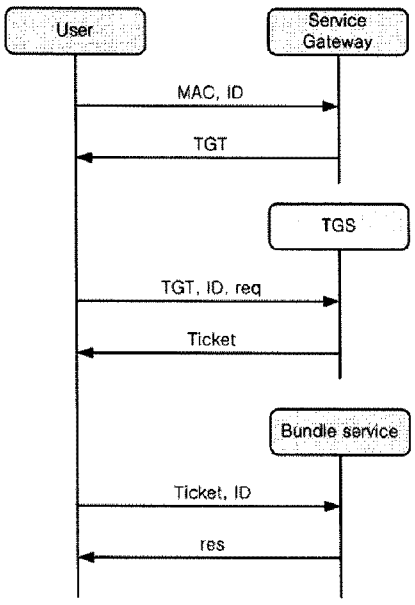


그림 2 사용자 인증 메커니즘 순서도

2.1 사용자 인증 과정

첫 번째 과정은 사용자 인증 과정이다. 보안 메커니즘의 가장 기본이 되는 과정으로 사용자가 서비스 게이트웨이에 사용자 정보를 전송하고 서비스 게이트웨이는 이를 토대로 사용자에게 인증 여부를 결정하는 과정이다.

Kerberos 프로토콜은 사용자의 패스워드를 비밀키로 이용하여 ID를 비밀키로 암호화 시켜서 서버에 전송하는 방식으로 구성된다. 그러나 본 논문에서는 자동화된 보안 메커니즘 구현을 위해서 패스워드 입력 방식을 채택하지 않았기 때문에 사용자 등록 과정에서 배포 받은 대칭키를 이용하여 암호화한 식별자와 MAC Address를 서비스 게이트웨이로 전송하는 방식을 연구하였다. 그림 3은 제안하는 사용자 인증 메커니즘의 사용자 인증 요청 과정을 나타낸 그림이다.

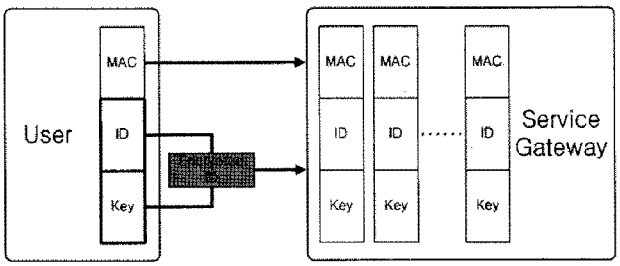


그림 3 제안하는 사용자 인증 메커니즘

그림 4는 사용자 인증 프로토콜을 나타낸 것이다. 본 논문에서는 메커니즘들의 명세화 된 프로토콜을 기술하고 도식화 하였다.

< 표기법 >

ID_U	사용자 U의 식별자
K_X	X의 비밀키
$K_{X,Y}$	X와 Y의 세션키
$K_X\{M\}$	비밀키 K_X로 암호화된 M

< 메시지 >

① $U \rightarrow SG$: $MAC \parallel K_U\{ID_U\}$

② $SG \rightarrow U$: $K_{SG}\{TGT\} \parallel K_U\{K_{U,SG}\}$

그림 4 사용자 인증 프로토콜

2.1.1 사용자 인증 요청 과정(U→SG)

사용자가 서비스 게이트웨이에 접속하고 구성원의 권한을 획득하기 위해 서비스 게이트웨이의 인증 서비스에 인증 요청을 하는 과정이다. 일반적으로 사용하는 패스워드 입력 과정을 생략하고 사용자와 서비스 게이트웨이간에 미리 등록된 MAC address와 식별자, 대칭키를 이용하여 자동화된 인증 과정을 거친다. 안전한 인증을 위해 사용자의 MAC과 함께 사용자의 대칭키 K_U를 이용하여 식별자를 암호화한 $K_U\{ID_U\}$를 서비스 게이트웨이에 전송한다.

2.1.2 서비스 게이트웨이의 사용자 TGT 발급(SG → U)

TGT란 티켓을 발행하는 TGS에 티켓 발행 요청을 하기 위한 티켓이다. 서버는 사용자로부터 받은 $K_U\{ID_U\}$를 서버가 가지고 있는 사용자의 MAC에 해당하는 대칭키를 이용하여 $K_U\{ID_U\}$를 복호화하고 복호화된 데이터가 서버가 저장하고 있는 ID와 일치할 시에 사용자에게 서버 자신의 비밀키 K_{SG}로 암호화 된 $K_{SG}\{TGT\}$를 발행한다. 그리고 TGS와의 통신 과정에서 사용하게 될 세션키를 발행하는데, 안전한 보안 과정을 위하여 사용자의 비밀키 K_U로 암호화한 세션키 $K_U\{K_{U,SG}\}$를 사용자에게 전송한다.

2.2 서비스 티켓 발급 과정

서비스 티켓 발급 과정은 사용자가 특정 서비스 이용에 필요한 티켓을 서버로부터 발급 받는 과정이다. 사용자가 특정 서비스에 접근하기 위하여 먼저 해당 번들 서비스에 접근할 수 있는 티켓을 발급 받기 위한 작업을 거쳐야 한다. 그림 5는 서비스 티켓 발급 프로토콜을 나타낸 것이다.

< 표기법 >
T TGS(Ticket Granting Service)
req Service request message

< 메시지 >
① $U \rightarrow T$: $K_{SG}\{TGT\} \parallel K_{U,SG}\{ID_U\} \parallel req$
② $T \rightarrow U$: $K_{BS}\{Ticket\} \parallel K_{U,SG}\{K_{U,BS}\}$

그림 5 서비스 티켓 발급 프로토콜

2.2.1 서비스 티켓 발급 요청(U → T)

TGS란 서비스 제공을 받기 위한 티켓을 발행해주는 서비스이다. TGS는 인증 서버와 물리적으로 다른 위치에 있을 수도 있지만 본 논문에서는 서론에서 얘기한 OSGi 환경의 특성상 물리적으로 동일한 서버에 하나의 번들 형태로 구성하였다.

사용자는 티켓을 발급받기 위해 TGS에 서비스 게이트웨이로부터 받은 $K_{SG}\{TGT\}$와 서비스 게이트웨이와의 세션키로 암호화된 사용자 식별자 $K_{U,SG}\{ID_U\}$, 그리고 티켓 발급 요청 메시지를 TGS에게 전송한다.

2.2.2 TGS의 사용자 티켓 발급(T → U)

사용자에게 티켓 발급 요청을 받은 TGS는 자신의 비밀키로 $K_{SG}\{TGT\}$를 복호화하고 사용자와 서버의 세션키로 암호화된 사용자의 식별자 $K_{U,SG}\{ID_U\}$를 복호화하여 사용자의 신원을 확인하면 TGS의 비밀키로 암호화된 티켓 $K_{BS}\{Ticket\}$을 사용자에게 발급하고 사용자의 세션키 $K_{U,SG}$로 암호화한 TGS와의 세션키 $K_{U,SG}\{K_{U,BS}\}$를 사용자에게 전송한다.

2.3 서비스 권한 부여 과정

그림 6은 서비스 권한 부여 프로토콜을 나타낸 것이다. 서비스 권한 부여 과정은 사용자 인증이 완료되어 서버로부터 발급받은 티켓을 이용하여 원하는 특정 번들 서비스의 권한을 부여받는 과정이다.

< 표기법 >
BS Bundle service
res Response message

< 메시지 >
① $U \rightarrow BS$: $K_{BS}\{Ticket\} \parallel K_{U,BS}\{ID_U\}$
② $BS \rightarrow U$: res

그림 6 서비스 권한 부여 프로토콜

2.3.1 서비스 권한 부여 요청(U → BS)

사용자는 원하는 번들을 이용하기 위해서 번들을 제공해주는 번들 서비스의 권한을 획득하여야 한다. 사용자는 권한 획득을 위한 TGS의 비밀키로 암호화된 서비스 티켓 $K_{BS}\{Ticket\}$과 세션키로 암호화된 식별자 $K_{U,BS}\{ID_U\}$를 번들 서비스에 전송한다.

2.3.2 번들 서비스의 사용자 권한 부여(BS → U)

번들 서비스는 사용자로부터 전송받은 암호화된 티켓을 복호화하여 유효한 티켓인지 판별 후 사용자에게 서비스 자원에 대한 권한을 부여하고 응답메시지를 전송한다. 사용자는 응답메시지를 보고 권한을 부여 받았는지 확인할 수 있다. 사용자는 번들 서비스로부터 받은 권한으로 번들서비스의 자원에 접근할 수 있다.

3. 번들 인증 메커니즘

현재 OSGi 스펙에 의하면 번들 인증에 서명된 JAR (signed JAR)와 PKI 기반의 인증 메커니즘 그리고 RSH 프로토콜[10]을 권고하고 있다. 그러나 PKI는 복잡한 연산 과정과 믿을 수 있는 서명자의 인증서와 그에 대한 인증 과정을 추가로 수행해야 한다. 이를 자원의 제약이 많은 OSGi 플랫폼 구조에 적용하기에는 어려움이 있다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 티켓을 사용한 대칭키 기반의 번들 인증 메커니즘을 제안한다.

대칭키는 암호화에 쓰이는 키와 복호화에 쓰이는 키가 동일하여 암호화한 사람의 메시지를 같은 키로 다른 사람이 복호화 할 수 있다. 대칭키는 이와 같이 암호화와 복호화에 같은 키를 사용함으로써 연산 속도가 빠르다는 특징이 있고, 이는 자원이 제한되어 있는 OSGi 플랫폼 환경에 적합하다.

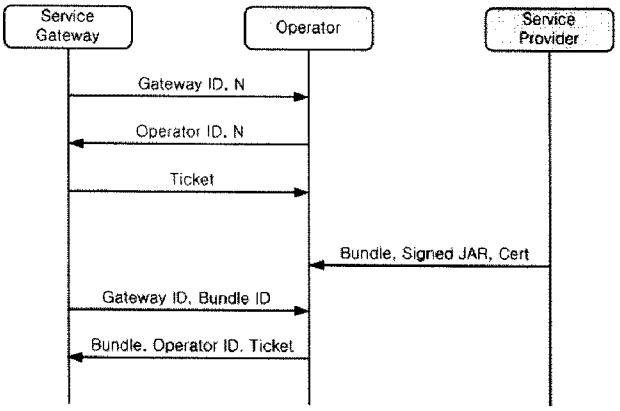


그림 7 번들 인증 메커니즘 순서도

그림 7은 본 논문에서 제안하는 대칭키 기반의 번들 인증 메커니즘이다. 번들 인증 메커니즘은 서비스 게이트웨이의 오퍼레이터 인증 과정과 서비스 게이트웨이가 오퍼레이터에게 번들을 요청하고 그 번들을 인증하여 전송 받는 번들 인증 과정까지 총 2단계의 과정으로 이루어져 있다.

3.1 오퍼레이터 인증 과정

오퍼레이터 인증 과정은 번들 인증을 하기 전에 오퍼레이터와 서비스 게이트웨이의 인증 작업을 하는 단계이다. 서비스 게이트웨이는 오퍼레이터 등록과정에서 사전에 자신의 키를 오퍼레이터에 보내주고 그 키를 바탕으로 인증 작업을 거친 후 번들 인증에 필요한 티켓을 발급해주는 과정으로 이루어져 있다. 오퍼레이터 인증 메커니즘에 대한 프로토콜은 그림 8과 같다.

< 표기법 >

ID_{SG} 서비스 게이트웨이 SG의 식별자

ID_{OP} 오퍼레이터 OP의 식별자

N_{SG} SG가 생성한 난수값

K_{SG} SG의 비밀키

< 메시지 >

① SG → OP : $ID_{SG} \| K_{SG}\{ID_{SG} \| N_{SG}\}$

② OP → SG : $ID_{OP} \| K_{SG}\{ID_{OP} \| N_{SG}\}$

③ SG → OP : $K_{SG}\{Ticket\}$

그림 8 오퍼레이터 인증 프로토콜

3.1.1 사전 키 공유 요청(SG→OP)

서비스 게이트웨이는 자신의 식별자 ID_{SG}와 새롭게 생성한 난수값 N_{SG}를 자신의 비밀키 K_{SG}로 암호화하여 자신의 식별자 ID_{SG}와 함께 오퍼레이터에게 전송한다. 여기서 난수값 N_{SG}는 서비스 게이트웨이의 부트스트래핑(bootstrapping) 과정에서 새롭게 생성하여 암호화의 신선도를 유지시켜주는 역할을 한다. 오퍼레이터는 받은 메시지를 미리 공유된 서비스 게이트웨이의 비밀키 K_{SG}로 복호화한다. 그리고 전송 받은 서비스 게이트웨이의 식별자와 메시지를 복호화하여 추출한 식별자가 같은 것인지 검사하여 복호화가 제대로 이루어졌는지 판단한다.

3.1.2 오퍼레이터 인증 요청(OP→SG)

오퍼레이터는 서비스 게이트웨이로부터 받은 난수값 N_{SG}를 자신의 식별자 ID_{OP}와 함께 서비스 게이트웨이의 비밀키 K_{SG}로 암호화하여 자신의 식별자와 함께 서비스 게이트웨이로 전송한다.

3.1.3 서비스 게이트웨이의 오퍼레이터 티켓 발급 (SG → OP)

서비스 게이트웨이는 오퍼레이터로부터 받은 메시지를 자신의 비밀키 K_{SG}로 복호화하여 난수값 N_{SG}를 추출한다. 자신이 생성한 난수값과 추출한 N_{SG}가 동일한지 검사하고 원하는 오퍼레이터가 맞는지 식별자를 확인한다. 그 후에 다시 오퍼레이터에게 티켓을 비밀키 K_{SG}로 암호화하여 전송한다. 여기서 티켓은 일정 시간 동안 사용할 수 있도록 시간 정보와 서비스 게이트웨이의 발급 정보들이 기록되어 있다.

3.2 번들 인증 과정

번들 인증 과정에 대한 프로토콜은 그림 9와 같다. 번들 인증 과정은 오퍼레이터가 서비스 제공자의 번들을 인증한 후 이 번들을 사전에 받은 티켓과 함께 서비스 게이트웨이에 전송하는 과정이다.

< 표기법 >

SP 서비스 제공자

KR_{SP} 서비스 제공자 SP의 개인키

SB_{SP} 서비스 제공자 SP의 서비스 번들

ID_{SB} 서비스 번들 SB의 식별자

$Cert_{SP}$ 인증 기관이 발급한 SP의 디지털 인증서

< 메시지 >

① SP → OP : $SB_{SP} \| ID_{SB} \| KR_{SP}\{H(SB_{SP} \| ID_{SB})\} \| Cert_{SP}$

② SG → OP : $ID_{SG} \| ID_{SB}$

③ OP → SG : $SB_{SP} \| ID_{SB} \| ID_{OP} \| K_{SG}\{Ticket\}$

그림 9 번들 인증 프로토콜

3.2.1 서비스 제공자 번들 인증(SP→OP)

서비스 제공자의 번들 인증 단계는 오퍼레이터가 서비스 제공자에게 서비스 번들을 제공받으면서 그에 대한 인증 작업을 하는 단계로써 서비스 게이트웨이에 번들을 제공하기 전에 별도로 수행한다. 오퍼레이터가 서비스 제공자에게 번들을 요청하면 서비스 제공자는 서명된 JAR 파일의 형태로 서비스 번들 SB_{SP}를 전송한다. 서명된 JAR는 해쉬 코드 $H(SB_{SP} \| ID_{SB})$를 서비

스제공자의 개인키 KR_{SP}로 암호화 한 서명 파일과 인증서 CertSP를 오퍼레이터에게 전송한다. 오퍼레이터는 전송 받은 인증서를 이용하여 서명 파일을 복호화하여 해당 번들의 인증을 수행한다.

3.2.2 게이트웨이 번들 요청(SG→OP)

서비스 게이트웨이는 오퍼레이터의 서버에 접속하여 필요한 번들의 식별자 ID_{SB}와 서비스 게이트웨이의 식별자 ID_{SG}를 오퍼레이터에 전송한다.

3.2.3 오퍼레이터 티켓 및 번들 인증(OP→SG)

오퍼레이터는 서비스 게이트웨이로부터 받은 번들의 식별자를 토대로 그에 해당하는 번들을 다시 보내주어야 한다. 오퍼레이터는 서비스 제공자로부터 받은 번들 SB_{SP}와 번들의 식별자 ID_{SB}, 오퍼레이터 식별자 ID_{OP}를 전송한다. 그리고 사전에 오퍼레이터 인증 메커니즘을 통해 서비스 게이트웨이로부터 전송받은 티켓 K_{SG}{Ticket} 덧붙여 전송한다. 서비스 게이트웨이는 전송받은 메시지들 중 티켓을 자신의 비밀키 K_{SG}로 복호화하여 티켓의 유효성을 검사한다. 티켓에는 발급 시간 정보와 발급한 게이트웨이의 정보 등이 포함되어있기 때문에 유효기간이 지난 티켓이나 다른 게이트웨이에서 발급 받았거나 변질된 티켓의 경우를 검출해 낼 수 있다. 티켓의 인증이 완료되면 오퍼레이터의 식별자를 이용하여 요청한 오퍼레이터로부터 온 번들인지 확인한 후 번들을 전송받는다.

4. 설계 및 구현

OSGi 보안 아키텍처의 사용자 인증 메커니즘은 클라이언트와 서비스 게이트웨이의 사용자 인증 서비스, 그리고 번들을 저장하고 사용자에게 제공해주는 번들 서비스로 이루어진다. 번들 인증 메커니즘은 오퍼레이터를 인증하는 서비스 게이트웨이의 오퍼레이터 인증 서비스와 서비스 제공자의 번들을 검증하고 서비스 게이트웨이에 제공하는 오퍼레이터의 번들 인증 서비스로 이루어진다.

보안 아키텍처의 구현은 Windows XP 운영체제에서 JDK 1.5를 사용하였다. OSGi 프레임워크는 Apache사의 Felix 1.2.1을 사용하였으며 암호화 연산은 JCE의 암호화 라이브러리를 사용하였다.

4.1 사용자 인증 시스템

사용자 인증 시스템은 서비스 게이트웨이에서 수행되는 사용자 인증 서비스와 클라이언트에서 수행되는 클라이언트 서비스, 그리고 번들 제공을 위해 번들 서버에 설치되는 번들 서비스로 구성된다.

그림 10은 사용자 인증 시스템의 클래스 다이어그램이다. 사용자 인증 서비스의UserAuthService 클래스에서 사용자의 접속 요청을 처리하고 SecurityService 클래스에서 사용자 인증을 처리한다. 클라이언트 서비스에서는 ClientConnectionService 클래스에서 서비스 게이트웨이에 접속 요청을 하고 사용자 식별자의 암호화는 SecretService 클래스의 EncryptionService 메소드에서 수행한다. 번들 서비스에서는 BundleService 클래스가 번들의 식별자를 관리하고 사용자의 번들 요청에 따라 번들을 선별하여 준다. VerifyTicketService 클래스는 사용자가 제시하는 티켓에 대한 인증 과정을 수행한다. BundleTransfer 클래스는 인증이 완료된 사용자에게 Bundle Context 데이터를 받아 사용자에게 원하는 번들을 전송한다.

4.2 번들 인증 시스템

번들 인증 시스템은 서비스 게이트웨이에서 번들을

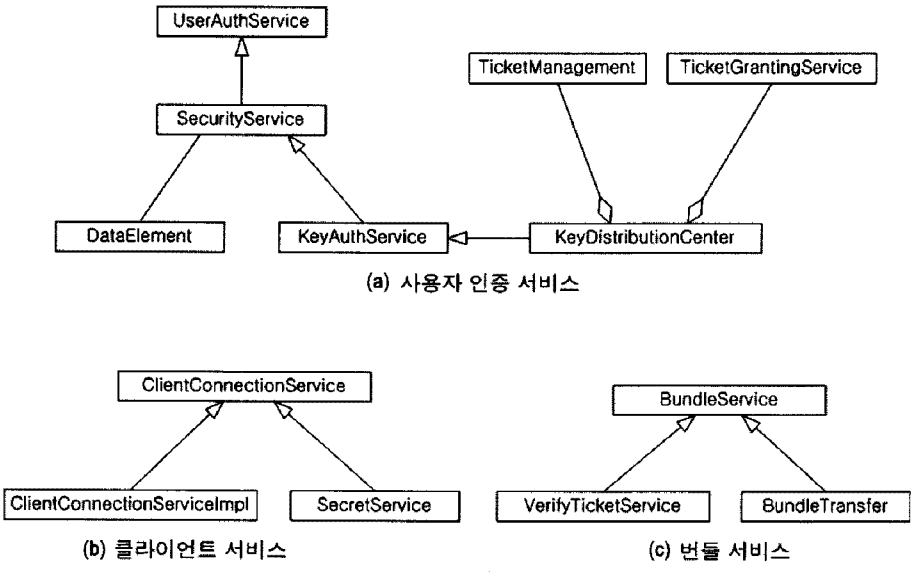


그림 10 사용자 인증 시스템 클래스 다이어그램

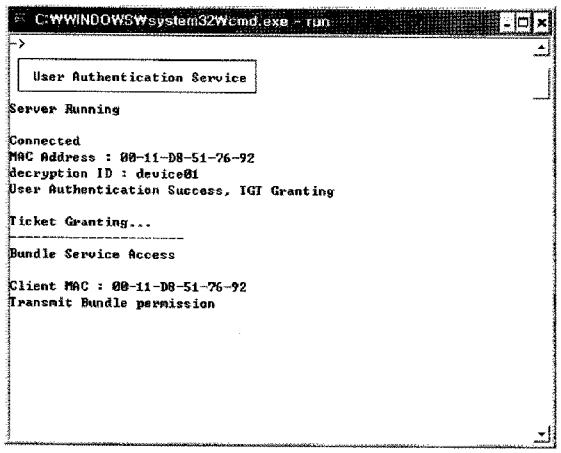


그림 11 사용자 인증 서비스 실행 화면

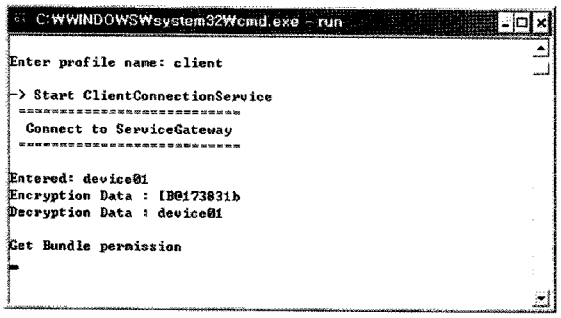


그림 12 사용자 접속 서비스 실행 화면

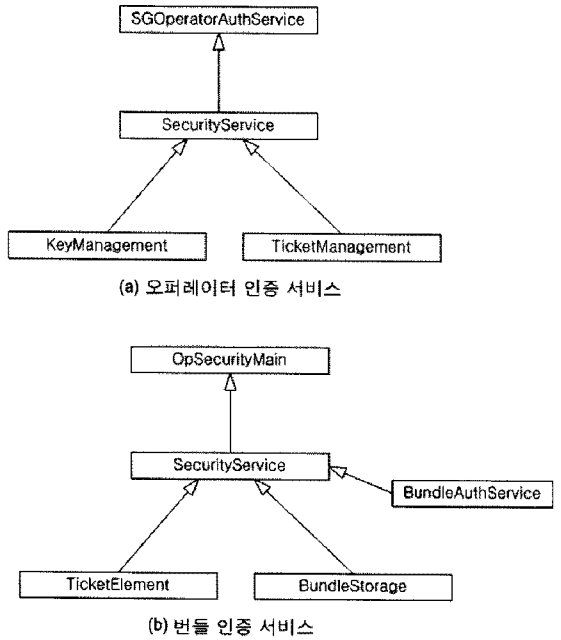


그림 13 번들 인증 시스템 클래스 다이어그램

제공받기 위해서 오퍼레이터에 대한 인증 과정을 수행하는 오퍼레이터 인증 서비스와 오퍼레이터에서 서비스 제공자의 번들을 인증하고 서비스 게이트웨이에 오퍼레이터 인증을 요청하는 번들 인증 서비스로 구성된다.

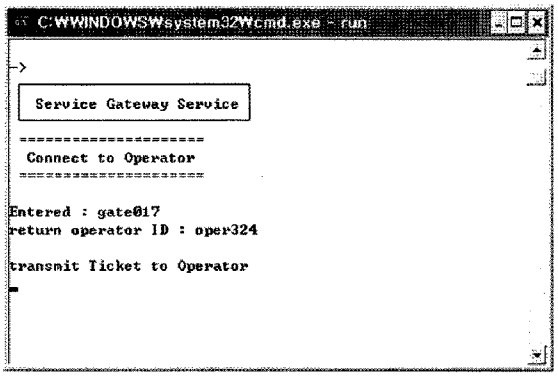


그림 14 오퍼레이터 인증 서비스 실행 화면

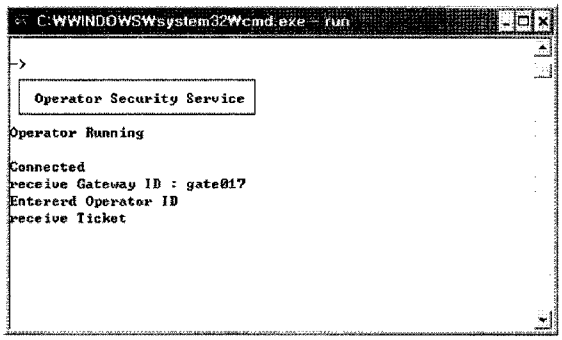


그림 15 번들 인증 서비스 실행 화면

그림 13은 번들 인증 시스템의 클래스 다이어그램이다. 오퍼레이터 인증 서비스의 SGOperatorAuthService 클래스에서는 번들을 제공받기 위해 오퍼레이터에 접속을 요청하는 작업을 수행한다. 사전에 검증된 오퍼레이터에게 대칭키를 배포하는 작업은 SecurityService 클래스에서 수행한다. 번들 인증 서비스의 BundleAuthService 클래스에서는 오퍼레이터가 서비스 제공자로부터 받은 번들이 공인 기관으로부터 검증된 것임을 확인하는 과정을 수행한다. 서비스 게이트웨이의 접속 요청은 OpSecurityMain 클래스에서 처리하며 SecurityService 클래스는 서비스 게이트웨이로부터 받은 암호화된 데이터를 복호화하는 작업과 사전에 배포 받은 대칭키를 관리한다. BundleStorage 클래스는 번들의 저장소 역할을 수행한다.

4.3 성능 평가

표 1은 제안하는 대칭키 기반의 자동화된 인증 메커니즘과 기존의 PKI 기반 인증 메커니즘에 대한 비교이다. 5가지의 기준으로 두 가지의 인증 메커니즘을 실험을 통해 비교하였다. 먼저 자동화 항목은 사용자의 입력과 대기시간이 없이 바로 인증을 과정을 거칠 수 있는가에 대한 내용이다. 제안하는 방식은 식별자를 암호화하는 방식으로 자동화된 인증 과정을 수행하였고 기존 방식은 사용자의 패스워드 입력과정이 필요하다. 인증 속도는 대칭키 기반이 별도의 추가적인 연산 과정이 필

표 1 인증 메커니즘의 비교

	제안하는 대칭키 기반	기존 PKI 기반
자동화	O	X
인증 속도	고속	저속
CPU 점유율	소 (4.1%)	대 (9.4%)
메모리 사용량	소 (2.7Mb)	대 (4.8Mb)
OP 재인증	티켓 유효기간 만료시	번들 전송시 항상

요한 PKI 기반에 비해 빠른 속도를 보여 주었다. CPU 점유율과 메모리 사용량은 자원이 부족한 OSGi 환경에서 중요한 요소이다. PKI 기반은 공개키 연산에 많은 자원이 소모되므로 제안하는 대칭키 기반의 인증 메커니즘에 비해 CPU 점유율과 메모리 사용량이 높게 측정되었다. 번들 전송시에 오퍼레이터 재인증 시점은 제안하는 대칭키 기반이 티켓 유효기한 만료시에 재인증을 수행하는 반면 기존 PKI 기반은 번들 전송시에 항상 인증 과정이 필요하였다.

위의 성능 평가 결과를 통해 제안하는 대칭키 기반의 자동화된 메커니즘이 PKI 기반의 기존 방식에 비해 성능이 뛰어나다는 것을 보여주었다. PKI는 키 관리의 용이성이 있지만 OSGi 환경의 특성상 하나의 서비스 게이트웨이에 접속하는 클라이언트 수가 많지 않으므로 제안하는 대칭키 기반이 OSGi 환경에 적합하다는 것을 알 수 있다.

5. 결 론

OSGi의 주목적은 정보 가전과 같은 내장형 시스템에 있다. 이러한 OSGi 플랫폼 환경은 정해진 소수의 사용자가 접속하며 디바이스들의 자원에 대한 제약이 많기 때문에 자원의 소모를 줄여야 한다. 또한 사용자가 커뮤니티 그룹에 접속할 때 다양한 디바이스의 입력 장치에 대한 특성을 고려하여야 한다.

본 논문에서는 OSGi 서비스 플랫폼의 특수한 환경에 맞춘 자동화된 보안 아키텍처를 제안하였다. OSGi 플랫폼을 기반으로 한 디바이스들의 제약사항을 고려하여 연산이 간단한 대칭키를 이용하여 효율성을 높이고 사용자의 식별자를 암호화하여 전송하는 방법을 사용하였다. 그로 인해 사용자의 패스워드 입력 과정을 생략할 수 있고, 로컬 네트워크에서 사용자의 접속만으로 인증 과정을 거쳐서 서비스의 권한을 부여하는 자동화된 사용자 인증 메커니즘을 구축하였다. 또한 번들 인증 메커니즘에서는 대칭키를 이용하여 검증된 오퍼레이터에게 서비스 게이트웨이의 라이센스 정보가 담긴 티켓을 발행하는 방법을 이용하였다. 티켓을 받은 오퍼레이터는 한 번의 인증 과정으로 티켓의 유효기한이 만료될 때까지 별도의 보안 관련 연산 과정이 없이 번들을 전송할 수 있다. 따라서 시스템 관리의 편의성과 자원이 부족한 OSGi 플랫폼 환경에서 자원 소모를 줄일 수 있는 장점이 있었다.

본 논문에서는 위에서 제시한 보안 메커니즘의 구성을 클래스 다이어그램을 통해서 보여주었고 제안한 보안 메커니즘이 사용자와 관리자의 편의성을 높이고 자원의 소모를 줄일 수 있어 OSGi 플랫폼 환경에서 효과가 있음을 보여 주었다.

참 고 문 헌

[1] OSGi Alliance, "OSGi Service Platform Release 4," http://www.osgi.org, 2007.

[2] 서대영, "OSGi를 중심으로 한 홈 게이트웨이 표준화 동향", 한국정보과학회, *2003년도 가을 학술발표논문집*, 제30권, 제2호, pp.337-339, 2003년 10월.

[3] Chi-Chih Huang, Pang-Chieh Wang, Ting-Wei Hou, "Advanced OSGi Security Layer," AINAW'07, vol.2, pp.518-523, May. 2007.

[4] 박대하, 김영갑, 문창주, 백두권, "OSGi 서비스 플랫폼 환경을 위한 보안 아키텍처", *한국정보과학회*, 제10권, 제3호, pp.259-272, 2004년 6월.

[5] Sun Microsystems, "Lesson: Packaging Programs in JAR Files," http://java.sun.com/docs/books/tutorial/deployment/jar/index.html, 2008.

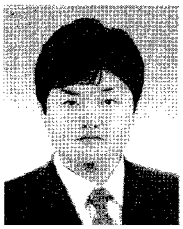
[6] H.-Y. Lim, Y.-G. Kim, C.-J. Moon, D.-K. Baik, "Bundle Authentication and Authorization using XML Security in the OSGi Service Platform," *ICIS'05*, pp.502-507, Jul. 2005.

[7] Sun Microsystems, "The Java Tutorials - Signing JAR Files," http://java.sun.com/docs/books/tutorial/deployment/jar/signing.html, 2008.

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

[9] B. Clifford Neuman and Theodore Ts'o, "Kerberos: An Authentication Service for Computer Network," *IEEE Communications Magazine*, vol.32, no.9, pp.33-38, Sep. 1994.

[10] OSGi Alliance, "OSGi Service Platform Service Compendium Release 4 - 110.8 Mapping To RSH Scheme," http://www.osgi.org, 2007.



이 창 욱

2006년 한세대학교 컴퓨터공학과 학사 졸업. 2009년 숭실대학교 대학원 컴퓨터학과 석사 졸업. 관심분야는 컴퓨터구조, 임베디드 시스템, 정보 보호, 영상 보안

홍 원 기

2005년 숭실대학교 컴퓨터학부 학사 졸업. 2007년 숭실대학교 대학원 컴퓨터학과 석사 졸업. 2007년~현재 숭실대학교 대학원 컴퓨터학과 박사과정. 관심분야는 메모리 테스트, VLSI 설계 및 테스트, 컴퓨터구조, 임베디드 시스템

장 훈

1987년 서울대학교 공대 전자공학과 학사 졸업. 1989년 서울대학교 공대 전자공학과 석사 졸업. 1993년 University of Texas at Austin 박사 졸업. 1994년 숭실대학교 컴퓨터학부 조교수. 2003년 숭실대학교 컴퓨터학부 부교수. 2008년~현재 숭실대학교 컴퓨터학부 정교수. 관심분야는 컴퓨터구조, 메모리 테스트, VLSI 설계 및 테스트, 임베디드 시스템