OSGi Service Framework 환경에서 사용자 인증 방법

전경석* 문창주 박대하 백두권
고려대학교 컴퓨터학과 소프트웨어 시스템 연구실*
{hoi2040 mcj dhpark baik}@software.korea.ac.kr

User Authentication Mechanisms in The OSGi Service Framework Environment

Gyoung-Suck Jeon* Chang-Joo Moon Dae-Ha Park Doo-Kwon Baik
Software System Lab, Dept. of Computer Science & Engineering, Korea University

요 약

최근 들어 정보가전과 홈 게이트웨이를 사용하여 집안의 가전제품과 외부 네트워크를 연결해 다양한 서비스를 제공하려고 하는 시도가 많아졌으며, 이와 함께 홈 게이트웨이에 대한 표준화 작업이 여러 단체에서 진행 중에 있다. 본 논문은 홈 게이트웨이 환경에서 여러 보안관련 부분 중 사용자 인증에 관한 문제점을 해결하고자 한다. 홈 게이트웨이는 기존 PC 환경처럼 CPU, 메모리 등의 리소스들이 충분하지 않고 각 서비스를 사용하기 위해 매번 로그인하지 않아야 하는 등의 이유로 PC 환경에서 사용하는 보안 메커니즘을 그대로 적용하는데는 한계가 있다. 홈 게이트웨이의 특성과 위와 같이 대두되는 몇 가지 문제점을 본 논문에서 제시하는 사용자 인증 방법을 통해 해결하고자 한다.

1. 서 론

가정에서 사용하는 가전제품은 다양하다. 따라서 이러한 가전제품을 외부 네트워크와 연결하기 위해 홈 게이트웨이가 필요하다. 현재, 홈 게이트웨이에 대한 표준화 작업이 TIA TR41.5, ISO/IEC JTC21 SC25 WG1, DOCSIS 등 몇몇 기관 및 단체에서 이루어지고 있다. 그 중에서 가장 활성화되어 있는 단체는 OSGi(Open Service Gateway Initiative) 이다. 현재 OSGi Service Platform Release 2.0 까지 배포되었으며[1], 아직 보안에 관련된 많은 부분은 논의 중에 있다.

홈 게이트웨이는 가정의 가전제품과 외부의 점검기기들이 연결되어, 가정 또는 외부의 사용자들이 접속해서 다양한 서비스를 받게된다. 따라서, 보안의 관점에서 홈 게이트웨이의 안전성 여부는 기밀성, 권한부여, 사용자 인증 등 여러 가지 면을 충족시켜야 한다. 그 중에서 보안의 시작이라고 할 수 있는 것이 홈 게이트웨이에 접속한 사용자가 어떤 사용자이며, 어느 정도 믿을 수 있는가? 이다. 즉 사용자를 인증하는 부분이라 할 수 있다. 이는 모든 보안의 시작이며, 홈 게이트웨이에서도 마찬가지이다.

사용자 인증에 사용하는 방법으로 기존에 ID/PW, PKI (Public Key Infrastructure), Kerberos Protocol, 등 여러 가지 방법이 있다. 하지만, OSGi Framework에서 사용자 인증은 기존의 PC 환경과 다른 몇 가지 상황들이 고려되어야 한다. 첫째, 리소스의 제약이 따른다. PC환경처럼 빠른 중앙처리장치와 많은 메모리, 저장매체를 사용할 수 없다. 둘째, Single Sign On 이 적용되어야 한다. 즉 사용자는 한번 인증을 받은 후, 서비스를 사용할 때마다 매번 인증 받지 않을 수 있어야 한다. 셋째, 네트워크 제약성이 없어야 한다. 즉, 홈 게이트웨이에 연결될 수 있는 다양한 네트워크 상에서 작동해야 한다. 기존의 보안 알고리즘과 프로토콜로는 위와 같은 문제들을 동시에 만족시킬 수 없다. 따라서, 기존의 사용자 인증 방법을 확장하여, 위와 같은 고려사항을 동시에 만족하는 사용자 인증 방법을 제시하고자 한다.

2. 관련 연구

2.1 OSGi Framework

2001년 10월 OSGi Release2가 발표되었다[1]. 기본적인 Framework은 거의 완성되었으며, 보안에 대한 부분은 지금도 진행 중에 있다. 전체적인 구조는 그림 1. 과 같다.

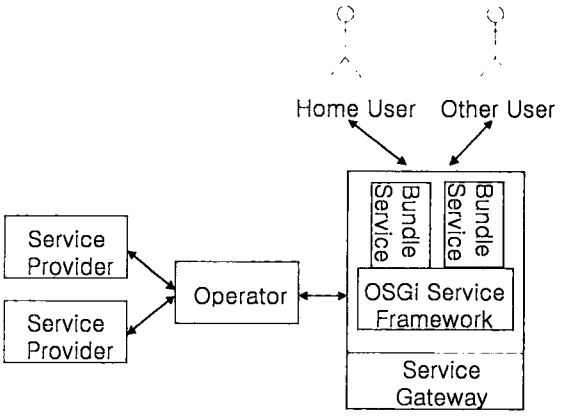

그림 1. OSGi Structure

OSGi는 Service Provider, Operator, Service Gateway, User 로 구성된다. 여기서, Service Gateway 와 사용자를 주목해야 한다. 하나의 Service Gateway에는 여러 개의 서비스를 위한 번들 들이 있고, 사용자는 Home User 와 Other User로 구분되어 진다. 이는 사용자 인증을 위한 메커니즘에 중요한 영향을 끼친다. 또한 하나의 Operator에 수많은 Service Gateway가 연결된다.

2.2 기존의 사용자 인증 방법

네트워크에 연결되어 있는 기존 가정의 PC 환경에서, 서버로부터 제공되는 서비스를 제공받기 위한 다양한 사용자 인증 방법이 존재한다.

먼저, 가장 일반적으로 사용되는 방법이 ID/PW를 이용한 방법이다. 이 방법은 여러 가지 측면에서 문제점이 있다. 먼저 공개된 네트워크상에 password의 보호장치가 아무것도 없어, 임의의 다른 사용자에게 노출될 위험이 있다. 또한 사용자가 다른 서비스를 사용하기 위해 매번 인증을 받아야 한다.[2]

두 번째로, PKI 인증 방법이 있다. 이는 공개 키, 개인 키 쌍으로 비대칭 키를 사용하는 인증방법이다. PKI는 사전에 out-of-band 방법을 통해 배포한 인증서를 기반으로 사용자를 인증한다. 이러한 비대칭 키를 사용하는 알고리즘으로 주로 사용되는 것은 RSA이다. 이러한 방법은 주로 강한 인증이 요구되는 곳, 즉 인터넷뱅킹 등에서 사용되는 것을 볼 수 있다. 하지만 PKI의 비대칭 키 연산은 DES와 같은 대칭 키에 비해 많은 자원과 연산 수행 시간이 걸린다. 또한, PKI는 many-to-many 인증관계에서 사용될 수 있도록 고안되었기 때문에, Service Gateway에서 복잡한 키 관리가 요구되어진다. 그리고 연관된 인증서에 필요한 신뢰 기반 구조를 제공하기 어렵다.[3][4][5]

세 번째로 Kerberos가 있다. Kerberos는 인증서버, 티켓 발행 서버로 분리되어, 사용자 인증을 한 후, TGT(Ticket Granting Ticket)를 받고, 이것을 사용하여 티켓 발생 서버에서 서비스 티켓을 받는다. 그리고 난 후 사용자는 이 서비스 티켓을 사용하여 원하는 서비스를 제공하는 서버에 접속하여 서비스를 사용할 수 있는 권한을 얻는다. 기본 구조는 그림 2. 와 같다.

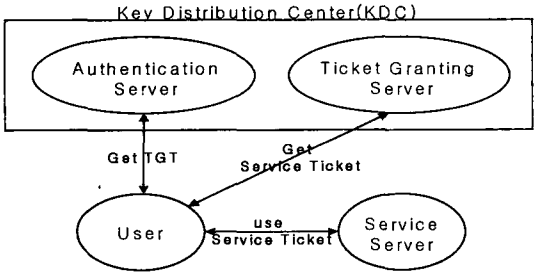

그림 2. Kerberos Protocol

Kerberos는 사용자 인증에서 대칭 키를 사용하고, 다수의 서비스를 받기 위해 하나의 서비스 티켓을 사용한다. 하지만, 사전에 Authentication Server와, Ticket Granting Server, Ticket Granting Server와 Service Server 사이에 공유 대칭 키를 가지고 있어야 한다.[6][7][8]

3. OSGi Service Framework 환경에서 사용자 인증 방법

본 논문에서는 기존의 사용자 인증방법 중 Kerberos Protocol을 OSGi 환경의 특성에 맞게 확장 적용한 사용자 인증 방법을 제시한다. OSGi 환경에서 Kerberos Protocol 의 Service Server에 해당하는 것이 Bundle Service이다. 이 Bundle Service는 Service Gateway에 존재하고, Bundle Service 와 Ticket Granting Server, Ticket Granting Server 와 Authentication Server 사이에 공유 대칭 키를 가져야 하는 Kerberos Protocol의 전제조건을 만족 시켜야 한다. 따라서, 다음 세 가지 방법을 생각해 볼 수 있다.

첫째 Operator에 KDC를 두는 방법이 있다(메커니즘-1). 둘째, Service Gateway에 KDC를 두는 방법이 있다(메커니즘-2). 셋째, KDC의 TGS(Ticket Granting Server)를 Service Gateway에 두고, AS(Authentication Server)를 Operator에 두는 방법이 있다(메커니즘-3). 본 논문에서는 메커니즘-3을 OSGi 환경에 적합한 사용자인증 메커니즘으로 제안한다.

3.1 사용자 인증 방법(메커니즘-3)의 구조

사용자 인증 방법(메커니즘-3)의 기본 구조는 그림 3. 과 같다.

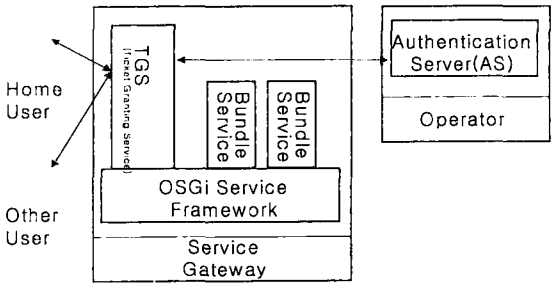

그림 3. 사용자 인증 방법(메커니즘-3)의 구조

OSGi 환경에서 Operator의 수는 하나 또는 극히 적은 수 가 된다. 또한 Operator가 Service Gateway들을 관리하는 역할을 하게 때문에 인증서버의 역할을 Operator가 하게된다. TGS(Ticket Granting Service)는 하나의 시스템 서비스로서, 사용자가 각 Bundle Service를 사용할 수 있는 서비스 티켓을 사용자에게 제공한다.

메커니즘-1은 AS와 TGS가 Operator에 있는 것이고, 메커니즘-2는 AS와 TGS가 Service Gateway에 있는 것이다.

3.2 공유 대칭 키 공유 방법(메커니즘-3)

메커니즘-3에서, Service Gateway가 켜지면, Bootstrapping 과정에서, Operator와 통신을 통해 TGS(Ticket Granting Service)를 다운, 설치, 실행시킨다. 이때, Operator는 TGS내에 Authentication Server 와 TGS 사이의 공유 대칭 키를 생성 내부 변수로 내장시킨다.

그런 다음 사용자는 TGS를 통해, Authentication Server와 통신하여 인증을 받고 TGT(Ticket Granting Ticket)을 받는다. TGS는 일종의 OSGi System Service로서, 각 번들의 서비스가 Service Gateway로 다운되어 설치 될 때마다, TGS와 Service 사이의 공유 대칭 키를 생성 저장한다.

위 두 과정을 통해 TGS는 Operator의 Authentication Server 와 Service Gateway의 Service들 간의 공유 대칭 키를 가지게 된다.

사용자는 Authentication Server로부터 인증을 받은 후 얻은 TGT를 이용해서 TGS에게 각 서비스에서 인증을 받기 위한 Service Ticket을 얻는다. 사용자는 이 Service Ticket을 사용하여 Service Gateway에 있는 각 번들의 서비스를 사용할 때마다. 서비스 내에서 인증을 받는데 사용한다. 즉, Service Gateway의 각 서비스를 이용할 때마다, Authentication Server로부터 인증을 받지 않는 것이다.

3.3 사용자 인증 프로토콜(메커니즘-3)

메커니즘-3에서 사용자 인증 프로토콜의 전체적인 과정은 그림 4. 와 같다.

U	사용자
S	서비스
TGS	서비스 티켓 발행 서비스
Kr	일 회 사용하는 대칭 키
Ka_b	a 와 b 사이에 공유 대칭 키
{M}Ka_b	Ka_b를 사용하여 암호화된 메시지
Ts#	time-stamps
Tauth	최초 인증 시간
Ta_b	a 와 b 사이에 사용되는 티켓

AS_REQ	: U, TGS, Ts1
AS_REP	: {Ku_tgs, TGS, Ts1}Ku, Tu_tgs
TGS_REQ	: U, S, Ts2, Tu_tgs, {auth}Ku_tgs
TGS_REP	: U, {Ku_s, S, Ts2}Ku_tgs, Tu_s
S_REQ	: Tu_s, {U, Ts3}Ku_s

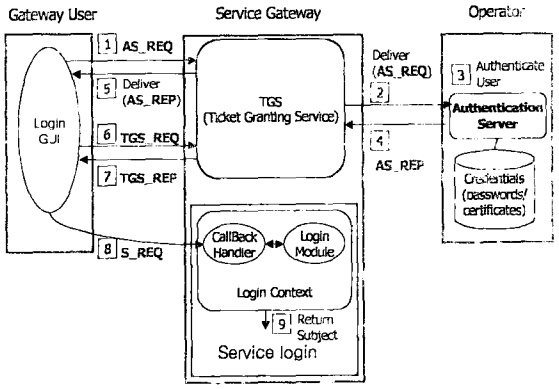

그림 4. 사용자 인증 프로토콜

그림 4.에 나와 있는 요청(REQ)과 그에 대한 결과(REP)의 상세 내용은 3.3.1, 3.3.2와 같다. 3.3.2에서 Tu_tgs는 TGT로서 사용자가 서비스 티켓을 TGS에 요청하기 위한 티켓이다. Tu_tgs는 "TGS,{Ku_tgs, U, Tauth}Ktgs"와 같이 구성되며, TGS가 Ktgs로만 복호화 할 수 있다. Tu_s는 서비스 티켓으로서 각 서비스에서 사용자가 Authentication Server로부터 인증 받았다는 것을 나타내는데 사용한다. 서비스 티켓(Tu_s)은 "S,{Ku_s, U, Tauth}Ks_tgs"와 같이 구성되며, 사용자는 복호화 할 수 없고, 사용자가 이용하고자 하는 서비스에서 Ks_tgs에 의해서만 복호화 된다.

그림 4.에서 "Service Login" 은 위에서 설명한 서비스 티켓(Tu_s)에 의해 각 서비스가 사용자의 인증여부를 검증하는 것이다. 각 서비스는 사용자에 대한 인증을 JAAS API를 통해 구현할 수 있다. 인증이 성공적으로 끝나면 JAAS API 의 Subject 객체를 리턴 되고, 서비스는 이 Subject 객체를 사용한다.[9]

이처럼 메커니즘-3을 사용하면, 사용자는 인증서버를 통해 한번만 인증을 받고, TGS로부터 받은 서비스 티켓(Tu,s)을 사용하여, Authentication Server로부터 인증 받았음을 나타냄으로서, 서비스 별들을 이용할 때마다. 매번 인증을 하지 않는다.

4. TGS와 AS의 위치에 따른 사용자 인증 방법 비교

메커니즘-1의 경우, TGS(Ticket Granting Service)가 Operator내에 존재하게 된다. 이럴 경우, Service Gateway의 부담은 줄어든다. 즉 사용자 인증에 있어서 Service Gateway의 추가적인 리소스 사용이 없다. 그러나, 네트워크 혼잡도는 그림 4. 에서 6, 7과정에 대한 Operator와의 통신이 필요하기 때문에 메커니즘-3에 비해 좀 더 혼잡해지게 된다. 또한, 각 Service Gateway들을 접속하는 사용자(Travel User)의 경우 각 Service Gateway별로 인증을 받아야 하는 치명적인 단점이 있다.

메커니즘-2의 경우, AS(Authentication Server)가 Service Gateway에 존재하게 된다. 이럴 경우, 네트워크 혼잡도는 그림 4. 에서 2, 4 과정이 없어지게 됨으로서 줄어들 수 있다. 하지만, Service Gateway가 인증서버의 역할을 함으로 인해 많은 자원(CPU, 메모리, 저장공간)과 공유 대칭 키 관리를 필요로 한다. 또한 Travel User의 경우 각 Service Gateway별로 인증을 받아야 한다.

메커니즘-3은 많은 자원과 공유 대칭 키 관리를 필요로 하는 AS를 Operator에 두고, Service Gateway에 TGS를 둠으로서, Service Gateway에서 자원부담을 최소화하고, 사용자가 한번 인증을 받은 후, TGT를 사용하여 각 Service Gateway에서 서비스를 사용하고자 할 때마다 인증을 받지 않는다. 아울러, 네트워크의 응용계층에서 구현되기 때문에, SSL처럼 특정 네트워크 프로토콜의 제약과 연관성 없이 동작할 수 있다.[10]

5. 결론 및 향후 연구 과제

본 논문에서는, 기존의 PC환경과 다르게, OSGi 환경에서 여러 요구사항들을 만족시키는 사용자 인증 시스템을 제안하였다. 이것은 기존의 보안에 있어서 검증된 프로토콜인 Kerberos를 OSGi 환경에 맞춰 보완 확장하였다.

현시점에도 OSGi 환경에서의 보안부분은 진행 중에 있다. 추후, 홈 게이트웨이에서 서비스 사용자의 시스템 자원에 대한 권한 부여(authorization)에 대한 연구가 필요하다.

< 참고 문헌 >

[1] "OSGi Service Platform Release 2 Specification" http://www.osgi.org/resources/docs/spr2book.pdf, 2001

[2] B.Clifford Neuman, Theodore Th'o "Kerberos : An Authentication Service for Computer Network", IEEE, Computer Magazine 32:9:33-38, September 1994

[3] Answers to Frequently Asked Questions about Today's Cryptography, 3.0 , http://www.rsa.com/rsalabs

[4] Marc Branchaud "A Survey of Public Key Infrastructures" 1997

[5] "RFC18-Security Architecture Specification" http://www.osgi.org

[6] "Kerberos: The Network Authentication Protocol" http://web.mit.edu/kerberos/www/

[7] "Single Sign-On Using Kerberos in Java" http://java.sun.com/j2se/1.4/docs/guide/security/jgss/singlie-signon.html

[8]John chung-i chung "Distributed Authentication in Kerberos Using Public Key Cryptography" http://www.ini.cmu.edu/NETBILL/pubs/pkda.html

[9] Charlie Lai Li Gong "User Authentication and Authorization in the Java Platform" Computer Security Applications Conference, December 1999

[10] "The SSL Protocol Version 3.0" http://www.netscape.com/eng/ssl3/draft302.txt