情報保護學會論文誌
第16卷 第5號, 2006. 10

SPKI/SDSI 인증서를 이용한 Jini 기반의 안전한 이벤트 서비스 설계 및 구현*

박 희 만,[1†] 김 인 수,[1] 이 영 록,[1] 이 형 효,[2] 노 봉 남[1]
[1]전남대학교, [2]원광대학교

Design and implementation of Jini-based secure event service using SPKI/SDSI certificate

HeeMan Park,[1†] InSu Kim,[1] YoungLok Lee,[1] HyungHyo Lee,[2] BongNam Noh[1]
[1]Chonnam National University, [2]WonKwang University

요 약

컴퓨팅 기기들이 점점 산재되고, 기기들의 이동성 또한 증가되면서, 동기적이고 직접적인 접대점 통신 모델은 여러 형태의 분산 컴퓨팅 환경에 적용하기에 적합하지 않다. 이벤트 서비스와 같은 간접통신 모델을 이용하면, 분산 환경에서의 응용들 사이의 결합도를 감소시키고 많은 정적인 요구들을 제거할 수 있는 이점이 있다. 본 논문에서는 신뢰할 수 있는 유비쿼터스 환경을 만들기 위해 Jini 기반의 안전한 이벤트 서비스를 설계하고 구현한다. 구현된 이벤트 서비스는 내용기반의 검색이 가능하고, 정당한 권한을 지닌 제공자와 소비자만이 이벤트를 주고받을 수 있도록 한다. 이벤트 서비스에 인증과 인가 기능을 제공하기 위해 SPKI/SDSI 인증서를 이용하고, 이벤트 제공자와 소비자의 분리, 이동기기의 연결 끊김 지원, 내용기반의 검색을 위해 Jini의 자바스페이스를 확장하여 구현한다.

ABSTRACT

As computing devices become ubiquitous and increasingly mobile, it becomes obvious that a synchronous and direct peer-to-peer communication model is not sufficient in distributed computing environment. Using an indirect event service instead of other traditional communication model has an advantage of decreasing the coupling of applications in a distributed environment and removing the need for many static dependencies. In this paper, we design and implement the secure event service for providing secure ubiquitous computing environment. The Secure Event Service implemented enables users to perform content-based event retrieval, and supports only eligible event comsumer and event producer can publish and receive events through the secure event service. SPKI/SDSI certificate is used for supporting authentication and authorization in the secure event service. In order to provide a content-based event retrieval, an asynchronous communication between event producers and consumers, and a disconnectedness support for mobile devices, we modify and expand the JavaSpace package.

Keywords : *Event Service, SPKI/SDSI Certificate, Jini, JavaSpace*

접수일: 2006년 3월 29일; 채택일: 2006년 7월 24일
* 본 연구는 정보통신부 대학 IT연구센터 육성, 지원사업의 결과로 수행되었습니다.
† 주저자, hareup@lsrc.chonnam.ac.kr

Ⅰ. 서 론

유비쿼터스 컴퓨팅 환경에서 상호 작용하는 응용들은 기존의 연결된 네트워크 컴퓨팅 환경에서처럼 직접적이고 결합적인 통신을 하기에는 많은 제약들이 있다. 유비쿼터스 응용들은 네트워크 사이를 이동하기도 하고, 어느 순간 전원이 방전되어 네트워크상에서 사라지기도 하며, 단일 처리 운영체제에서 네트워크에 연결된 응용이 초점을 잃음으로써 연결이 끊길 수도 있다. 따라서 응용간의 통신에도 이와 같은 동적인 특성을 반영하여 간접적이고 분리된 통신을 제공하여야 한다. 상호 작용하는 응용들 간의 분리와 간접성을 획득하는 가장 일반적인 방법은 이벤트 서비스를 이용하는 것이다.

연결이 자주 끊길 수 있는 응용을 위해서 이벤트 서비스는 이벤트 소비자를 위해 에이전트와 같은 역할을 해야 한다. 이벤트 생산자의 입장에서 이벤트 소비자에 대한 정보를 알지 못하더라도 이벤트를 발행할 수 있어야하며, 이벤트 소비자 또한 이벤트 생산자에 대한 정보를 모르더라도 이벤트를 수신하는데 문제가 없어야 한다. 마지막으로 이러한 일련의 이벤트 전달 과정은 안전해야 한다는 것이다.

본 논문에서는 신뢰할 수 있는 유비쿼터스 환경을 만들기 위해 지니 기반의 안전한 이벤트 서비스를 설계하고 구현한다. 구현된 이벤트 서비스는 내용기반의 검색을 가능하게 하고, 정당한 권한을 지닌 제공자와 소비자만이 이벤트를 주고받을 수 있도록 한다. 서비스에 인증과 인가 기능을 제공하기 위해 SPKI/SDSI(Simple Public Key Infrastructure / Simple Distributed Security Infrastructure) 인증서를 이용하고, 제공자와 소비자의 분리, 이동기기를 위한 연결 끊김 지원 그리고 내용기반의 검색을 위해 지니(Jini)의 자바스페이스(JavaSpace)를 확장하여 구현한다.

Ⅱ. 관련연구

2.1 유비쿼터스 환경에서의 이벤트 서비스

유비쿼터스 컴퓨팅 환경의 미들웨어인 Gaia [1], M3-RTE [2] 에는 구성요소간의 메시지 통신을 위한 이벤트 서비스들을 포함하고 있다.

Gaia의 이벤트 서비스 [3,4] 는 서로 다른 개체들 사이에서 분리된 통신을 제공하기 위한 모델을 제공한다. 이벤트 서비스는 Orbacus-Corba [5] 의 기초적인 이벤트 서비스를 사용하여 통신한다. Gaia의 이벤트 서비스는 OMG(Object Management Group)의 COS(Common Object Service) [6] 모델을 구현하였다. Gaia의 이벤트 통신은 이벤트 공급자와 이벤트 소비자들이 이벤트 서비스라는 중간 매체를 이용하여 서로에게 메시지를 전달한다. 중간 매체인 이벤트 서비스를 이용함으로써 직접통신에서 크게 나타났던 이벤트 공급자와 이벤트 소비자 간의 결합도를 낮추었고, 서로에 대한 정보에 의존하지 않고서도 통신을 할 수 있다. 하지만 네트워크상에서 전달되는 메시지의 순서를 장담할 수 없으며, 통신의 복잡성이 증가하게 된다. 또한 채널의 생성은 이벤트 서비스가 채널을 중앙 관리해야 하는 단점이 있고, 하나의 채널은 하나의 이벤트 타입을 전송하는 통로이므로 채널을 이용하는 이벤트 제공자와 소비자 사이에는 이벤트 타입에 대한 의존성이 발생하게 된다.

컴포넌트 기반의 Gaia 미들웨어에는 도메인에 추가되는 컴포넌트들에 대한 Gaia 인증 메커니즘 모듈(Gaia Authentication Mechanisms Modules : AMM) [7,8] 을 가지고 있다. Gaia에서는 인증모듈이 컴포넌트의 식별자를 얼마나 신용하는지에 따라 신용도를 달리 정하고, 접근제어에서 이 신용도를 활용할 수 있도록 한다. 또한 정책구성을 위해서는 역할 기반 접근제어(Role Based Access Control : RBAC) [9,10] 모델을 이용하고 있다. 결과적으로 Gaia 미들웨어에서는 인증되고 정당한 접근권한을 가진 컴포넌트들만이 Gaia 이벤트 서비스를 이용할 수 있다. 하지만 Gaia 미들웨어에서는 이벤트 서비스를 접근하기위한 인증모듈은 존재하지만 이벤트 서비스를 통해 전달되는 이벤트에 대한 인증 및 접근제어에 대한 기술은 없다. 즉, 정당한 컴포넌트들만이 Gaia 이벤트 서비스를 접근할 수 있지만, 이벤트 서비스에 접근 가능한 부정한 컴포넌트가 고의로 잘못되고 부정확한 이벤트를 생성시켜 네트워크를 범람시킬 수 있는 것과 또 다른 컴포넌트에게 보내지는 이벤트에 대한 도청을 막을 수는 없다.

M3-RTE 미들웨어도 Gaia와 마찬가지로 컴포넌트 기반의 미들웨어이다. 컴포넌트 사이에 메시지 전달을 위해 이벤트 서비스를 사용한다. M3-RTE 이벤트 서비스는 이벤트 전달을 위해 클라이언트-서버 구조를 사용한다. 클라이언트들은 이벤트 서버 프로세스와 세션을 설정한 후 이벤트를 보내거나 다른 컴포넌트들에 의해 보내진 이벤트를 받을 수 있다.

M3-RTE의 클라이언트들은 이벤트의 생산자나 소비자가 동일한 세션 안에 있기 때문에 세션 안에서 익명으로 활동할 수 있는 장점이 있다. M3-RTE에는 정책관리자가 존재하여 각 컴포넌트에 대한 역할기반의 접근제어를 수행한다. 이벤트 서비스 또한 컴포넌트로 타 컴포넌트가 이벤트 서비스에 접근하기위해서는 관련 역할을 획득하여야한다. 하지만 Gaia에서와 마찬가지로 이러한 접근방법도 이벤트 서비스에 대한 접근제어 방법은 제공하지만 전달되는 이벤트 자체에 대한 접근제어 기술을 제공하지는 않는다.

2.2 지니 기술

미들웨어인 지니[11]은 새로운 서비스가 네트워크 상에 연결될 수 있도록 하고, 클라이언트는 추가적인 설정을 하지 않고도 이들 서비스들을 즉시 이용할 수 있게 한다. 서비스가 수정된 경우에도 클라이언트는 이를 이용하는데 문제가 없다.[12]

하지만 지니에는 응용들 사이에서 간접통신을 제공하기 위한 이벤트 서비스가 없다. 대신 자바 객체를 저장할 수 있는 자바스페이스가[13,14] 있다. 자바스페이스는 다른 지니 서비스처럼 조회를 통해 발견되고 프락시를 통해 사용되는 완전한 지니 서비스의 형태로서 일반적인 데이터 저장 서비스가 아닌 객체를 저장할 수 있는 서비스이다. 자바스페이스의 목적은 객체들을 위한 파일 시스템을 제공하는 것으로 지니를 이해하는 클라이언트들과 서비스 사이에서 객체를 공유하는 방법을 제공한다. 자바스페이스는 강력한 타입검사나 이동 코드 및 안전한 실행 등을 포함하여 객체들을 저장하기 때문에 자바 객체의 모든 이점을 살릴 수 있다.

2.3 SPKI/SDSI 인증서

SPKI/SDSI는 기존의 PKI 기반의 융통성 없고 복잡한 문제를 단순화하기 위해 연구되었다. SDSI의 특징 중 두드러진 것은 "지역이름공간(local name space)"라는 개념을 도입해서 공개키의 소유자는 각각 그 공개키에 기반을 둔 지역이름공간을 생성할 수 있다는 것이다. 다시 말하면 누구든지 자신이 보유한 주체들의 공개키에 자신의 공개키에 기반을 둔 지역이름을 연결하는 이름 인증서를 발행할 수 있다. 또한 이들 지역이름을 이용하여 새로운 이름 인증서를 발행할 수 있다.[15]

SPKI는 이름 인증서와 권한 인증서를 각각 구별하여 명세할 수 있는 융통성을 제공한다.[16] 예를 들면 한 사용자의 권한은 수시로 바뀔 수가 있는 관계로 기존 PKI 인증서에 권한을 부여하는 통합방식인 X.509 인증서는 한 사용자의 권한이 바뀔 때마다 매번 그 사용자에게 새로운 인증서를 발행해야하는 불편함이 있다. 또한 SPKI는 주체(principal)의 구별을 이름이 아닌 공개키로 함으로써 주체의 신상정보에 따라 수시로 변경될 수 있는 주체의 전역이름을 그 주체의 구별 대상으로 하는 X.509 인증서의 문제점을 해결한다.[17] 필요한 권한 인증서들을 인증서 캐시에서 찾아주는 "Certificate chain discovery"라는 알고리즘[18]은 서버가 SPKI/SDSI로 보호된 정보를 클라이언트가 요청 할 때 클라이언트는 그 정보를 취득할 권한이 있다는 것을 서버에게 증명해보이기 위해 사용한다. 즉, 클라이언트가 서버에 증명할 권한 인증서와 관련된 이름 인증서들을 찾아내는 알고리즘이다.

Ⅲ. 이벤트 서비스 요구사항과 보안 위협

3.1 이벤트 서비스 요구사항

사용자는 더 이상 컴퓨팅 장치나 서비스에 구속되지 않고, 저마다 특색 있게 장치와 서비스를 이용한다. 유비쿼터스 컴퓨팅 환경에서는 장치, 사용자, 응용의 관점에서 기존의 컴퓨팅 환경과는 다른 특징을 나타낸다.[19] 장치들은 네트워크 사이를 넘나들며 이동하고, 응용은 장치 사이를 넘나든다. 장치들의 이동성은 다른 네트워크 범위들 사이에 이동하는 장치들 간의 연결을 유지하거나, 네트워크 연결 끊김을 처리해야 하는 문제를 발생시킨다. PDA (Personal Digital Assistants)와 같은 현재의 모바일 기기들은 분산 네트워크 통신이 일어날 수 있는 방법에 충분히 영향을 미칠 정도로 자주 끊기는 것이 특징이다.

간접통신 패러다임은 생산자가 소비자의 위치와 능력에 대한 걱정 없이도 데이터를 보낼 수 있다. 즉, 생산자는 소비자의 연결 끊김 등에 관계없이 이벤트를 발행할 수 있다. 또한 소비자는 메시지의 특정 그룹에 구독신청을 하게 되므로 공급자에서 생산된 메시지는 많은 목적지에 전달된다. 연결끊김 기기들에게 이벤트나 통지 배달을 위해서는 이벤트 저장을 위한 데이터 저장소가 요구된다. 이벤트 소비자와

이벤트 서버간의 연결이 끊어지면 이벤트 서버는 이벤트를 저장소에 저장하고 다음에 다시 이벤트 소비자가 접속해 오면 그 해당 이벤트를 여러 가지 방법으로 전달해 줄 수 있다.

장치를 넘나드는 응용을 위하여 JVM(Java virtual machine)과 같은 플랫폼에서 제공하는 코드 이동이 필요하다. 이것은 또한 이종의 실행환경에서 실행시간에 코드 이주와 같은 능력이 요구된다. 이와 비슷하게 소프트웨어 컴포넌트들을 분산시키기 위해서는, 단지 분산통신의 투명성만 제공해주는 CORBA와 같은 플랫폼보다도 더 월등한 이동성 및 컴포넌트 간의 조정을 제공할 수 있는 플랫폼이 필요하다. 지니 기술은 코드 이주 능력을 제공한다.

유비쿼터스 컴퓨팅 환경에서 사용자들은 똑같은 장치와 똑같은 응용을 갖고서도 똑같은 정보를 받고자하지는 않는다. 사용자마다 선호하는 정보가 다르고 취사선택이 자유롭다. 내용기반 통신은 데이터를 필터링하고 관심 있는 내용을 선택하기 위한 훌륭한 방법을 제공한다. 필터링은 이벤트 소비자에서 수행되는 것이 아니라 서버에서 수행된다. 필터링이 이벤트 서버에서 수행되기 때문에 데이터 흐름을 줄일 수 있다. 자바스페이스의 어트리뷰트(Attribute) 매칭 기술은 이벤트 필터링 기술로 적합하다.

3.2 이벤트 메시지에 대한 보안 위협

기존의 이벤트 서비스에서 이벤트 생산자는 이벤트 정보를 제공받는 이벤트 소비자의 식별자를 알지 못하고 알 필요도 없다. 이와 같은 기존의 이벤트 서비스는 이벤트 소비자에 대한 구독 제어가 없어 이벤트 소비자는 발행되는 모든 이벤트에 대해 구독하는 것이 가능하다. 하지만 이벤트 생산자의 입장에서 발행하는 이벤트는 정당한 소비자만이 구독할 수 있고, 다른 소비자들로부터 비밀이 지켜지기를 희망할 것이다.

역으로 이벤트 소비자는 자신이 구독하는 이벤트를 제공하는 이벤트 생산자에 대한 정보를 알지 못한다. 이벤트 생산자에 대한 이벤트 발행에 대한 제어가 없어 이벤트 생산자는 어떤 이벤트든지 발행할 수 있다. 하지만 이벤트 소비자의 입장에서 제공받은 이벤트가 정당한 생산자로부터 발행된 것이고 중간에 위조되거나 변조되지 않았음을 보장받기를 원할 것이다.

앞서 말한 두 가지 보안 이슈는 생산자와 소비자가 서로를 인증하는 문제와 키관리 문제와 연관되어 있다. 인증은 행위를 하는 주체를 파악하는 것으로 이벤트 서비스를 포함하는 유비쿼터스 네트워크에서는 쉽지 않다. 이벤트를 생산하는 시점에 이벤트 생산자는 이벤트 소비자가 어떤 개체가 될지 알지 못하고, 반대로 이벤트 소비자는 이벤트 구독을 요청하는 시점에 이벤트 생산자가 어떤 개체가 될지 알 수 없기 때문이다. 본 논문에서는 신뢰할 수 있는 제3자에 의해 공개키를 인증하는 공개키 기반구조(Public Key Infrastructure : PKI)와 비슷하게 제3자인 이벤트 서비스에 의해 이벤트 생산자와 이벤트 소비자를 간접 인증하는 방법을 사용한다.

또 하나의 보안이슈는 인증된 이벤트 생산자가 잘못되고 부정확한 이벤트로 네트워크를 범람시키는 것과 인증된 소비자에 의한 이벤트 도청 문제이다. 지금까지의 많은 유비쿼터스 미들웨어는 이벤트 서비스 자체에 대한 접근제어에 관심이 있다. 인증되어지고 이벤트 서비스에 접근할 수 있는 역할을 받은 컴포넌트들은 모든 이벤트 타입에 대해 발행권한과 구독권한을 얻게 된다. 이벤트 서비스에 접근할 수 있는 역할이 모든 이벤트 타입에 대한 이벤트 발행 권한 또는 이벤트 구독 권한으로 이어져서는 안 된다. 이벤트 서비스에 인증된 생산자나 소비자라 할지라도 이벤트 타입에 따른 발행권한과 구독권한의 제어와 더불어 이벤트에 대한 검색, 수정 등의 권한 제어가 필요하다.

마지막 보안이슈는 유비쿼터스 환경에서 공개키 기반구조에 의한 안전한 통신채널 확보의 어려움이다. 여러 네트워크 사이를 이동하는 사용자 또는 장치, 혹은 많은 응용들이 가상공간에서 정보를 안전하게 주고받기 위해서는 안전성과 신뢰성을 확보해야한다. 가상공간에서 생산자와 소비자는 서로의 신분확인은 물론, 이들이 주고받는 정보의 내용이 변경되지 않고 타인에게 노출되지 않는다는 보장을 받아야 비로소 서로 간에 행한 전자거래에 대한 결과에 승복할 수 있다. 이처럼 전자거래의 안전성과 신뢰성을 확보하기 위한 기술 중 가장 보편적으로 이용되고 있는 것은 X.509 공개키 기반구조이다.

하지만 X.509 공개키 기반구조는 루트에서부터 시작하는 계층적 구조의 전역이름(global name)을 이용하고 있으므로 사용자가 소속되어 있는 회사나 부서의 이동에 따른 전역이름 변경이 자주 일어날 수 있을 뿐만 아니라 유비쿼터스 컴퓨팅 환경에서는 인증서 발급과 관리에 있어서도 문제가 발생한다. 유비

쿼터스 컴퓨팅 환경에서는 온라인 서버의 부재 가능성이 있다. 그것은 네트워크 환경이 인증서를 검증할 수 있는 인증기관과 연결이 단절된 경우일 수도 있고, 갑자기 생겨나고 없어지는 에드혹(adhoc) 네트워크처럼 인증기관의 역할을 수행할 만한 컴퓨팅 능력을 가진 서버가 네트워크상에 없을 수도 있음[20,21]을 의미한다. 이러한 문제를 해결하기 위해서는 온라인 서버가 없을 수도 있음을 가정한 상황에서 안전한 통신 기반구조가 필요하다. SPKI/SDSI 인증서를 이용하면 온라인 서버가 없는 가운데서도 기존의 네트워크에서 적용했던 것과 유사하게 인증과 접근제어를 할 수 있다.[22]

Ⅳ. 안전한 이벤트 서비스의 설계 및 구현

4.1 자바스페이스를 수정한 이벤트 서비스

본 논문에서 구현한 이벤트 서비스는 이벤트 생산자와 소비자의 낮은 결합도 제공을 위해 자바스페이스를 사용한다. 자바스페이스는 객체를 저장하는 생산자가 객체를 소비하는 소비자의 정보를 모르더라도 객체를 저장할 수 있고, 반대로 저장된 객체를 소비하는 소비자는 객체를 저장하는 생산자에 대한 정보를 모르고도 저장된 객체를 소비할 수 있는 자바스페이스의 특징은 간접통신의 그것과 같다. 본 논문에서는 자바스페이스에 저장되는 객체를 이벤트로 이용한다. 다만 자바스페이스에서 이벤트를 일으킨 객체를 구독신청한 소비자에게 바로 반환하지 않는 부분을 수정하여 이벤트 서비스 (JavaSpace based Event Service : JES)를 구현하였다.

4.2 안전한 이벤트 서비스의 구성요소

이벤트 서비스를 사용하여 보내지는 모든 이벤트는 그림 1의 이벤트 클래스를 상속하여 만들어진다.

```
Class Event {
  String name;       /*이벤트 이름*/
  String source;     /*이벤트 생산자URI*/
  Integer gen_time;  /*이벤트 생산시간*/
  Integer seq_no;    /*이벤트 생산번호*/
  Integer duration;  /*이벤트의 유효기간*/}
```

그림 1. 이벤트 클래스

이벤트 클래스를 상속받은 하위 이벤트 클래스의 이름은 바로 그 클래스가 만들어내는 이벤트의 타입이 된다. 이벤트 이름은 이벤트 타입과 결합하면 주제기반 이벤트 서비스에서의 채널과 같다.

이벤트 소비자와 이벤트 생산자 모두는 자신이 정당한 사용자임을 증명하기 위해 "Certificate chain discovery"라는 알고리즘을 사용하여 자신이 보유하고 있는 인증서 캐시에서 권한 인증서와 이름 인증서들을 찾아내어 SeJES에 자신의 권한을 증명하게 된다.

SSCM(SPKI/SDSI Certificate Manager)은 이벤트 소비자와 이벤트 생산자가 SeJES에 대해 정해진 연산을 수행할 수 있는 권한이 있음을 증명한 인증서 경로를 검증하는 역할을 수행한다. SSCM도 마찬가지로 "Certificate Chain Discovery" 알고리즘을 포함하고 있다. SSCM은 알고리즘을 통해서 얻어진 값을 LRC와 ERC에게 반환한다.

LRC(Listener Registration Controller)는 이벤트 소비자가 이벤트 구독요청시 이벤트 리스너를 JES에 등록할 책임을 지고 있으며, 이벤트 소비자가 이벤트 검색할 때 정당한 사용자인지를 검사한다. LRC는 이벤트 소비자에게서 받은 이벤트 타입과 인증서 묶음을 SSCM에게 보내 권한 검사를 의뢰하고, 그 반환 값을 통해 이벤트 소비자의 등록여부를 결정한다. 이벤트 소비자의 이벤트 리스너 등록 요청과 이벤트 검색 요청에 대해서 이미 등록된 이벤트 소비자인지를 점검한 후 요청을 수행한다.

ERC(Event Registration Controller)는 이벤트 생산자를 등록하고 관리할 책임이 있고, 이벤트 생산자가 정당한 사용자인지를 검사한다.

JES는 이벤트 생산자가 이벤트 객체를 발행하면 그 이벤트 객체에 구독신청한 이벤트 소비자에게 비동기적으로 이벤트 객체를 직접 전달한다. 그리고 이벤트 소비자가 연결이 끊겼던 기간 동안 받지 못했던 이벤트들을 검색할 수 있게 한다.

4.3 안전한 이벤트 서비스 구현

4.3.1 이벤트 생산자/소비자 등록 절차

이벤트 클라이언트(이벤트 생산자 또는 이벤트 소비자)는 이벤트를 발행하거나 구독신청 하기 위해서는 자신이 해당 이벤트에 대한 정당한 발행 권한 또는 구독 권한이 있음을 증명해야만 한다. 그림 2는 이벤트 클라이언트가 이벤트에 대한 정당한 권한이

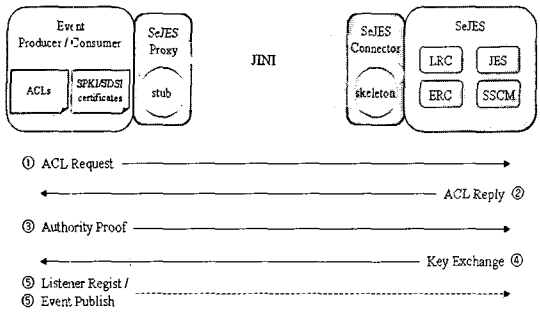


그림 2. 이벤트 생산자/소비자 등록 절차

있음을 증명하는 절차이다. 그림 2의 각 단계는 지니 환경에서의 원격 메소드 호출에 의한 것이다.

그림 2의 각 단계는 지니 환경에서의 원격 메소드 호출로 이루어진다. 각 단계의 데이터들은 모두 자바 객체이며, 원격 메소드의 인수 또는 반환값이 된다.

1단계 : 접근 제어 목록 요청

이벤트 클라이언트는 이벤트 서비스에 이벤트 발행 권한 또는 구독 권한을 얻기 위하여 이벤트 서비스에게 해당하는 접근 제어 목록을 요구한다. 접근 제어 목록을 요청할 때 이벤트 클라이언트는 이벤트 서비스에게 이벤트 템플릿을 제공한다.

이벤트 템플릿은 그림 4의 이벤트 클래스를 상속하며, 구독하기를 원하는 이벤트 타입과 이벤트 매칭 규칙을 담고 있는 자바 객체이다. 이벤트 타입은 이벤트 템플릿의 클래스명이고, 템플릿 내의 널 값이 아닌 데이터 필드의 값은 매칭을 위해 사용된다. 데이터 필드가 널 값인 경우 임의의 값과 매칭되는 와일드 카드로 사용된다. 접근 제어 목록을 요청할 때 이벤트 생산자는 이벤트 템플릿의 "name" 필드에 생산자 등록 요청(Producer Registration Message : PRM) 식별자인 "PRM"으로 한다.

2단계 : 접근 제어 목록 응답

이벤트 서비스는 이벤트 클라이언트로부터 받은 이벤트 템플릿에 관련된 접근 제어 목록을 검색하고, 이벤트 클라이언트의 세션 식별자를 생성한다. 그림 3은 접근 제어 목록 응답 클래스이다. 이벤트 서비스는 세션 식별자와 이벤트 템플릿 쌍을 저장하고, 접근 제어 목록과 세션 식별자를 이벤트 클라이언트에게 반환한다. 접근제어목록은 그 구조가 권한인증서 구조와 같다.

```
Class Challenge {
    SessionID sessionID; /*세션 식별자*/
    AuthCert [] ACLs;   /*접근제어목록*/ }
```

그림 3. 접근제어목록 응답 클래스

접근 제어 목록은 어떤 이벤트 클라이언트가 어떤 이벤트에 대해 발행 또는 구독권한이 있는지를 규정하고 있다. 접근 제어 목록은 이벤트 서비스에 의해 생성된 것과 이벤트 생산자에 의해 생성된 것이 있다. 이벤트 서비스는 어떤 이벤트 생산자가 어떤 이벤트를 발행할 수 있는지를 규정하고, 이벤트 생산자는 자신이 생산한 이벤트에 대해 어떤 소비자가 구독권한이 있는지를 규정하고, 그 권한을 이벤트 서비스에게 위임할 수도 있다.

세션 식별자는 이벤트 서비스가 이벤트 클라이언트를 구별하고, 암호화 계산을 줄이고 이전의 세션을 재사용하기 위한 필드이다. 세션 식별자는 4단계에서 클라이언트에게 제공되는 유효시간동안만 이벤트 서비스에 저장되고 이후 파기된다.

3단계 : 권한 증명

이벤트 클라이언트는 이벤트 서비스로부터 받은 접근제어목록과 자신이 저장하고 있는 인증서를 가지고 권한증명을 수행한다. 이벤트 클라이언트는 이벤트 서비스의 접근제어목록으로부터 이벤트 클라이언트 자신에 이르는 인증서 경로를 생성함으로써 권한증명을 한다. 인증서 경로는 세션 식별자와 함께 이벤트 서비스에 제공된다. 이때 이벤트 생산자는 자신이 생산할 이벤트에 대한 접근제어목록도 함께 제공한다.

인증서 경로는 이벤트 서비스의 접근제어목록으로부터 이벤트 클라이언트에게 이르는 인증서들을 순서

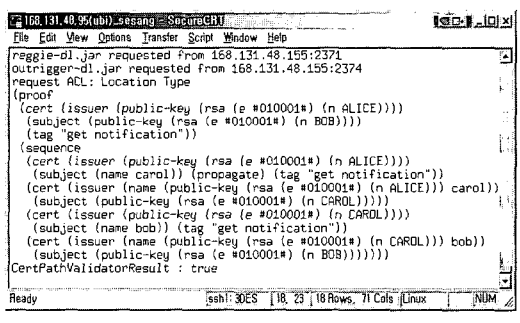


그림 4. 인증서 경로 검증

적으로 나열하는 것이다. 인증서 경로를 만들기 위해 "Certificate chain discovery"라는 알고리즘이 사용된다. 인증서 경로에는 클라이언트의 공개키가 포함되고, 클라이언트의 서명이 추가된다.

그림 4은 이벤트 소비자가 권한 증명을 하고 등록 요청하면, 이벤트 소비자가 SeJES에게 보낸 인증서 경로가 정당한 것인지를 SeJES.SSCM이 검증하는 과정이다.

4단계 : 키 교환

그림 5는 공개키로 암호화된 세션키를 응답하는 클래스이다.

```
Class SecureSession{
    byte[] sessionKey; /*세션키 암호화값*/
    byte[] nonce;   /*Nonce 암호화값*/ }
```

그림 5. 키교환 응답 클래스

이벤트 서비스는 클라이언트가 증명한 인증서 경로를 "Certificate chain discovery 알고리즘"으로 검증하고, 세션키와 Nonce를 생성하여 2단계에서 저장된 세션 식별자와 이벤트 템플릿 쌍에 덧붙여 저장한다. Nonce와 세션키 쌍은 클라이언트의 공개키로 암호화하여 반환한다. Nonce는 특정한 순간을 증명하기 위해 생성된 값이다. Nonce는 재생공격(replay attack)으로부터 방어를 위해 사용된다. 세션키는 공개키 암호화 비용을 줄이기 위해 클라이언트와 이벤트 서비스 사이의 한 세션동안 사용되는 비밀키이다.

5단계 : 이벤트 발행과 구독

이벤트 생산자는 4단계를 통해 얻은 세션 식별자, Nonce, 세션키를 통해 안전하게 이벤트를 발행할 수 있고, 그 이벤트는 3단계에서 자신이 발행한 접근제어목록에 대해 권한 증명을 할 수 있는 소비자만이 이벤트를 받을 수 있음을 보장받는다. 이벤트 소비자는 세션 식별자, Nonce, 세션키를 통해 안전하게 리스너를 이벤트 서비스에 등록할 수 있고, 3단계에서 자신이 권한 증명을 한 이벤트를 안전하게 구독할 수 있다.

그림 6은 이벤트 생산자가 자신을 SeJES에 등록하고 이벤트를 발행하는 과정이고, 그림 7은 이벤트 소비자가 이벤트 구독신청을 하고 실제로 이벤트를 수신하고 있는 과정이다.

4.3.2 발행과 구독의 제한

SeJES의 모든 함수는 모든 클라이언트에게 개방적이다. 하지만 클라이언트가 모든 이벤트를 발행하고 구독할 수 있는 것은 아니다. 이벤트 생산자와 이벤트 소비자는 자신이 권한을 증명한 이벤트에 대해서만 발행과 구독이 가능하다. SeJES는 이벤트 생산자가 특정 이벤트에 발행권한이 있음을 SPKI/SDSI 인증서와 ACLs를 통해서 증명하도록 하고, SeJES는 그 증명을 검증한다. 검증된 이벤트 생산자와 특정 이벤트를 한쌍으로 관리함으로써 이벤트 생산자는 권한이 증명된 이벤트만을 발행할 수 있도록 제어한다. 한편 이벤트 생산자는 자신이 발행하는 이벤트에 대해 ACLs를 기술한다. SeJES는 특정 이벤트에 구독을 요구하는 이벤트 소비자에게 이벤트 생산자가 기술한 ACLs에 대한 권한 증명을 요구하고, 그 증명을 검증한다. 이와같이 SeJES는 권한이 검증된 이벤트 소비자만이 특정 이벤트를 구독할 수 있도록 제어한다.

모든 이벤트 생산자(주체)는 SeJES(객체)에 대

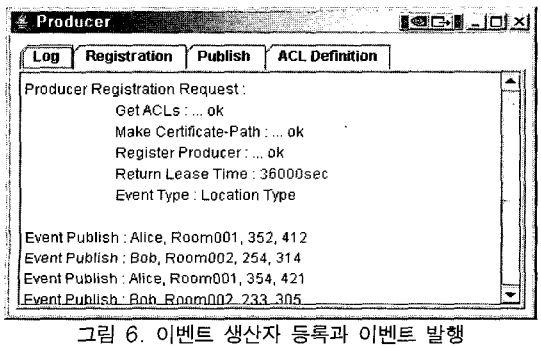


그림 6. 이벤트 생산자 등록과 이벤트 발행

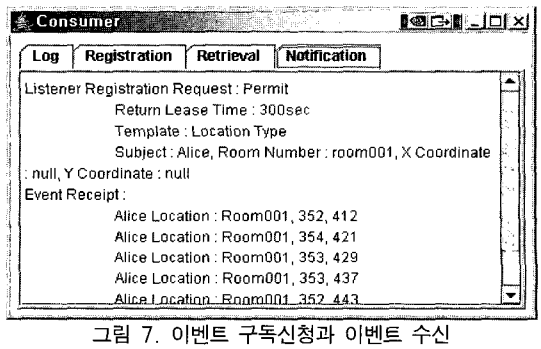


그림 7. 이벤트 구독신청과 이벤트 수신

표 1. Gaia 이벤트 서비스와 SeJES의 비교

	Gaia 이벤트 서비스	SeJES
간접통신	지원	지원
다수 이벤트 채널	지원	가상 채널
이벤트 타입	고정	고정되지 않음
이벤트 저장	CORBA	자바스페이스
이벤트 유효기간	설정안됨	설정가능
단말 간의 인증	없음	간접인증
이벤트 접근제어	없음	SPKI/SDSI
접근제어 오버헤드	없음	있음

해 publish()(연산)을 수행할 수 있지만, 자신에게 허용된 이벤트타입으로 제한된다. 이와 유사하게 이벤트 소비자는 SeJES에 대해 subscribe()을 수행할 수 있지만, 자신에게 허용된 이벤트타입으로 제한된다. 예를 들어 TVEvent 타입의 이벤트 발행권한은 가지고 있지만, RadioEvent 타입의 이벤트 발행권한을 갖지 못한 이벤트 생산자는 publish (tvEvent, ...)로 TVEvent 타입의 이벤트를 발행할 수 있지만, publish(radioEvent, ...)로 RadioEvent 타입의 이벤트를 발행할 수는 없다. 이와 마찬가지로 TVEvent 타입의 이벤트 구독권한은 가지고 있지만, RadioEvent 타입의 이벤트 구독권한을 갖지 못한 이벤트 소비자는 subscribe(tvEvent, ...)로 TVEvent 타입의 이벤트를 구독할 수 있지만, subscribe (radioEvent, ...)로 RadioEvent 타입의 이벤트를 구독할 수는 없다.

V. 시스템 분석 및 결론

Gaia의 이벤트 서비스는 CORBA 이벤트 저장소를 이용하여 구현된 반면에, 본 논문에서의 이벤트 서비스는 Jini의 JavaSpace를 수정하여 만들어진다. 하지만 가장 큰 차이점은 Gaia의 이벤트 서비스는 이벤트에 대한 접근제어가 없지만, 본 논문에서의 SeJES는 SPKI/SDSI 인증서와 ACLs를 이용하여 이벤트에 대한 접근 제어가 가능하다는 것이다. 표 1은 Gaia의 이벤트 서비스와 본 논문의 안전한 이벤트 서비스인 SeJES를 비교한 것이다.

첫째로 Gaia 이벤트 서비스와 SeJES 모두 분리된 통신과 간접적 정보 전달이 가능하다. Gaia 이벤트 서비스와 SeJES 모두는 이벤트 소비자와 이벤트 생산자 사이에서 이벤트의 중계 역할을 하면서 둘 사이에 분리된 통신을 가능케 한다. 분리된 통신으로 이벤트 소비자들은 이벤트 생산자에 대한 정보를 모르고도 필요한 이벤트를 얻을 수 있다. 하지만 Gaia 이벤트 서비스는 여전히 채널의 이름에 기초해서 이벤트 소비자와 이벤트 생산자가 결합되기 때문에 완전하게 분리된 모델은 아니다. 다시 말하면, Gaia 이벤트 소비자들은 특정 이벤트를 받기 위해서는 반드시 특정 채널에 가입해야만 하고, 소비자가 원하지 않더라도 동일 이벤트 채널에 가입한 이벤트 생산자들이 발행한 모든 이벤트를 받을 수밖에 없다. 하지만 SeJES에서 채널은 가상적인 것으로 이벤트 생산자와 이벤트 소비자 사이에는 직접적인 채널이 없으며, 가상적인 이벤트 채널은 SeJES와 이벤트 소비자간의 이벤트 리스너 등록으로 만들어진다. 이 채널은 순전히 이벤트 소비자의 주도하에 생성되며, 이렇게 생성된 가상채널은 이벤트 생산자에게 영향을 미치지 않는다.

둘째로 SeJES는 객체 임대기간을 이용한 이벤트 유효기간 설정이 가능하다는 것이다. 이벤트 서비스의 이벤트 저장기능은 다른 한편으로 저장된 이벤트의 폭증을 가져올 수 있는데, 이를 막기 위해 이벤트마다 유효기간을 기술할 수 있다. 이벤트 유효기간은 SeJES에 이벤트가 저장되어 있을 최대 시간이다. 그 시간이 지나면 이벤트는 SeJES에서 지워진다. 이 기능은 저장된 이벤트의 폭증을 막을 수 있을 뿐만 아니라, 이벤트 유효기간 설정의 주체가 이벤트 생산자라는 점에서 의미없는 이벤트가 서버에 계속 남아 있음으로 해서 가져올 수 있는 이벤트 무결성 문제를 유효기간의 자유로운 설정으로 해결할 수 있는 장점이 있다. 한편, SeJES가 기반으로 하고 있는 자바스페이스의 객체 임대 협상을 통해 SeJES의 성능을 넘어선 지나친 이벤트 유효기간의 설정이 되지 않도록 할 수 있다.

셋째로 SeJES는 Gaia 이벤트 서비스에는 없는 이벤트 서비스에 의한 생산자와 소비자의 간접인증 방법을 제공한다. 앞서 기술한 바대로 SeJES에서의 이벤트 생산자와 이벤트 소비자는 서로에 대한 정보가 없고, 가상채널 형성에 영향을 미치지도 않는다. 이벤트를 생산하는 시점에 이벤트 생산자는 이벤트 소비자가 어떤 개체가 될지 알지 못하고, 반대로 이벤트 소비자는 이벤트 구독을 요청하는 시점에 이벤트 생산자가 어떤 개체가 될지 알 수 없다. SeJES는 이벤트 생산자 등록절차를 통해서 특정 이벤트를 발행할 수 있는 이벤트 생산자임을 인증할 수 있고,

비슷하게 이벤트 소비자의 등록절차를 통해 특정 이벤트를 구독할 수 있는 이벤트 소비자임을 인증할 수 있다. 이벤트 생산자와 이벤트 소비자는 스스로가 이벤트를 주고받을 상대방을 인증하지는 않지만, 자신이 발행한 이벤트가 정당한 소비자에게만 전달됨을 SeJES를 통해서 보장받을 수 있고, 자신이 구독하는 이벤트가 정당한 생산자로부터 왔음을 SeJES를 통해서 보장받을 수 있다.

넷째로 SeJES는 이벤트 단위의 접근제어 기능이 있다. 이것은 "이벤트 생산자나 소비자가 SeJES에게 어떤 연산을 수행할 수 있는가"와는 다른 것이다. SeJES는 이벤트 생산자가 특정 이벤트에 발행권한이 있음을 SPKI/SDSI 인증서와 ACLs를 통해서 증명하도록 하고, SeJES는 그 증명을 검증한다. 검증된 이벤트 생산자와 특정 이벤트를 한쌍으로 묶어 관리함으로써 이벤트 생산자는 권한이 증명된 이벤트만을 발행할 수 있도록 제어한다. 한편 이벤트 생산자는 자신이 발행하는 이벤트에 대해 ACLs를 기술한다. SeJES는 특정 이벤트에 구독을 요구하는 이벤트 소비자에게 이벤트 생산자가 기술한 ACLs에 대한 권한 증명을 요구하고, 그 증명을 검증한다. SeJES는 권한이 검증된 이벤트 소비자만이 특정 이벤트를 구독할 수 있도록 제어한다.

제안된 방법에는 보안을 위한 오버헤드가 분명히 존재한다. 이벤트 클라이언트가 세션을 형성하기 위한 등록과정에서 인증서 체인 생성 연산과 공개키 암호화 연산이 추가된다. 등록이후 이벤트 발행 또는 구독을 위한 세션키(대칭키)에 의한 암호화에 의한 오버헤드도 발생한다.

마지막으로 SeJES는 계층적인 온라인 인증서버 없이도 인증과 접근제어를 수행한다. 유비쿼터스 컴퓨팅 환경에서는 인증과 인가를 위한 온라인 인증 서버를 사용하기 어려울 수 있음을 고려하여, SeJES는 온라인 인증 서버 없이도 인증과 권한 증명이 가능한 SPKI/SDSI 인증서를 사용하여 안전한 통신 채널을 확보한다.

본 논문에서는 유비쿼터스 환경에서 안전한 이벤트 통신 환경을 구축하고자, 자바스페이스를 이용하여 이벤트 서비스를 구현하고 SPKI/SDSI 인증서를 이용하여 안전한 이벤트 통신 채널을 확보하는 방법을 보였다. 본 연구는 이벤트 서비스 사이의 부하를 조절하는 기능과 이벤트 서비스의 갑작스러운 고장에 대비하는 결함 감내 기능 등을 추가하여 이벤트 관리자로 확장할 수 있다.

참 고 문 헌

[1] Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt "Gaia:A Middleware Infrastructure to Enable Active Spaces." *In IEEE Pervasive Computing* 74-83, Dec 2002

[2] A. Rakotonirainy, J.Indulska, W.W Loke, A.Aaslavsky "Middleware for Reactive Components: An Integrated Use of Context, Roles, and Event Based Coordination", Lecture Notes In Computer Science, Vol. 2218 . 77-98

[3] Bhaskar Borthakur, "Event Service User Manuel", white paper, October 29, 2001

[4] Bhaskar Borthakur, "The Event Service, white paper", October 2001

[5] Object-Orientated Concepts Inc, "ORBacus for C++ and Java". 1998.

[6] Object Management Group, "The Common Object Request Broker : Architecture and Specification", 2.0 ed., July 1995.

[7] Roy Campbell, Jalal Al-Muhtadi, Prasad Naldurg, Geetanjali Sampemane, M. Dennis Mickunas, "Towards Security and Privacy for Pervasive Computing" LECTURE NOTES IN COMPUTER SCIENCE 2609, 2003

[8] Shiva Chetan, Anand Ranganathan, and Roy H. Campbell, "Towards Fault Tolerant Pervasive Computing", In IEEE Technology and Society. Volume: 24, No. 1, pp 38-44, Spring 2005.

[9] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramoul, "Proposed Nist Standard for Role-Based Access Control," ACM Transactions on Information and System Security,

Volume 4, Issue 3, pp. 224-274, August 2001.

[10] Ravi S. Sandhu, Pierangela Samarati, "Access Control: Principle and Practice," IEEE communications Magazine, pp. 40-49, September 1994.

[11] Sun Microsystems, "Jini(TM) Architecture Specification", Sun Microsystems, 1997~2000

[12] W.Keith Edwards, W.Edwards, "Core Jini", Pearson Education, 2000

[13] Philip Bishop and Nigel, "JavaSpaces IN PRACTICE", *Addison-Wesley*, 2003

[14] Sun Microsystems, "Jini(TM) Technology Core Platform Specifications", *Sun Microsystems*, 2005

[15] 이영록 "RBAC 및 비밀통신 기능을 갖는 SPKI/SDSI HTTP 보안서버", 한국정보보호논문지, 2003

[16] Carl M. Ellison, Bill. Frantz, Butler. Lampson, Ron Rivest, Brian M. Thomas, Tatu Ylonen, "SPKI Certificate Theory" RFC2693, September 1999.

[17] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, Tatu Ylonen, "SPKI Examples," Internet-Draft. March 1998.

[18] Dwaine Clarke, Jean-Emile Elien, Carl Ellison, Matt Fredette, Alexander Morcos, and Ronald L. Rivest, "Certificate Chain Discovery in SPKI/SDSI," Journal of Computer Security, vol9, Dec 2000.

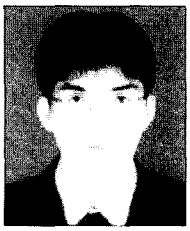
[19] Norman, D., "The invisible computer: why *good* products can fail, the personal computer is so complex, and information appliances are the solution", *MIT Press*, 1998.

[20] 최성재, 김용우, 이홍기, 송주석, 양대헌, "모바일 애드혹 네트워크의 안전하고 효과적인 최적의 인증경로 탐색 기법", 한국정보과학회 논문지, 2005.

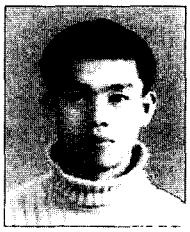
[21] K.Sanzgiri, D.LaFlamme, B.Dahill, B.N.Levine, C.Shields, E.M.Belding Royer,"Authenticated routing for ad hoc networks", IEEE Journal on Selected Areas in Communications, March, 2005

[22] G. Navarro, J. Borrell, J. A. Ortega-Ruiz, S. Robles, "Access control with safe role assignment for mobile agents", AGENTS: International Conference on Autonomous Agents, July 2005.

〈著者紹介〉

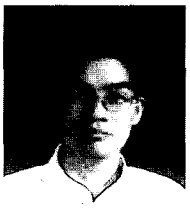


박 희 만 (Hee-Man Park) 학생회원
2003년 2월: 전남대학교 전기공학과 졸업
2006년 2월: 전남대학교 정보보호학과 석사
2006년 3월~현재: 전남대학교 정보보호학과 박사과정
〈관심분야〉네트워크 보안, 유비쿼터스 보안

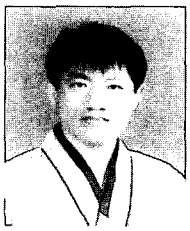


김 인 수 (In-Su Kim) 학생회원
2002년 2월: 전남대학교 전산학과 졸업
2005년 2월: 전남대학교 정보보호학과 석사
2005년 3월~현재: 전남대학교 정보보호학과 박사과정
〈관심분야〉네트워크 보안, 상황인지 컴퓨팅, 유비쿼터스 보안

이 영 록 (Young-Lok Lee) 정회원
1986년 2월 : 전남대학교 계산통계학과 졸업
1990년 2월 : 전남대학교 전산통계학과 한국대학교 전자공학과 석사
2003년 2월 : 전남대학교 전산학과 박사
2003년 3월 ~ 현재 전남대학교 리눅스시스템연구센터 연구교수
〈관심분야〉 유비쿼터스 보안, 전자상거래 보안, 보안모델, 정보보호 시스템



이 형 효 (Hyung-Hyo Lee) 정회원
1987년 2월 : 전남대학교 계산통계학과 졸업
1989년 2월 : 한국과학기술원 전산학과 석사
2000년 2월 : 전남대학교 대학원 전산학과 박사
1990년 ~ 1997년 : 삼보컴퓨터 기술연구소, 한국통신 연구개발원
2001년 3월 ~ 현재 : 원광대학교 정보·전자상거래학부 조교수
〈관심분야〉 보안모델, 네트워크보안, 전자상거래보안



노 봉 남 (Bong-Nam Noh) 정회원
1978년 2월 : 전남대학교 수학교육과 졸업
1982년 2월 : KAIST 대학원 전산학과 석사
1994년 2월 : 전북대학교 대학원 전산과 박사
1983년 ~ 현재 전남대학교 컴퓨터정보학부 교수
2000년 ~ 리눅스 보안 연구센터 소장
〈관심분야〉컴퓨터와 네트워크 보안, 정보보호시스템, 전자상거래 보안, 사이버사회와 윤리