홈 네트워크 환경에서 이동 에이전트의 역할에 기반한 접근제어 프레임워크 설계 및 안전성 평가

정 용 우[†] · 고 광 선[††] · 김 구 수[†††] · 엄 영 익[††††]

요 약

홈 네트워크 환경은 가정내의 다양한 디지털 기기들이 네트워크로 통합한 최첨단 생활환경으로써, 이동 에이전트는 이러한 홈 네트워크 환경에서의 새로운 컴퓨팅 요소로써 활용될 것으로 기대된다. 특히, 이동 에이전트의 이동성과 비동기적 수행능력은 가정내의 디지털 기기들을 제어하고 관리하기 위해 발생하는 네트워크 트래픽을 감소 시킬 수 있다. 그러나, 이동 에이전트를 홈 네트워크 환경에서 적용하기 위해서는 이동 에이전트에 대한 접근제어가 반드시 필요하다. 기존의 홈 네트워크 시스템에서는 홈 서버를 이용하여 사용자에 대한 접근제어를 수행한다. 홈 서버는 디지털 기기와 사용자의 권한을 명시하는 접근제어 목록을 이용하여 홈 네트워크로 접근하는 사용자에 대한 접근제어를 수행한다. 이를 위해 홈 서버는 디지털 기기와 사용자의 권한 간의 최신 정보를 저장하기 위해 주기적으로 접근제어 목록을 갱신하는 추가적인 연산을 수행한다. 따라서, 본 논문에서는 홈 네트워크 환경에서 이동 에이전트의 역할에 기반한 접근제어 프레임워크(Secure-KAgent)를 보인다. 본 프레임워크는 Role-Based Access Control(RBAC)을 기초한 접근 권한의 관리가 가능하다. 또한, 본 논문에서 제안하는 롤 티켓(Role ticket)을 이용함으로써 이동 에이전트에게 안전한 역할 분배를 보장한다.

키워드 : 이동 에이전트, 홈 네트워크, 역할 기반 접근제어, RBAC

Design and Safety Analysis of a Role-Based Access Control Framework for Mobile Agents in Home Network Environments

Youngwoo Jung[†] · Kwang Sun Ko[††] · Gu Su Kim[†††] · Young Ik Eom[††††]

ABSTRACT

A home network is a residential local area network in which digital home appliances are connected with each other. Applying the mobile agent technology to the home network is expected to provide a new computing model. In particular, mobility and asynchronous ability of mobile agent can be used to reduce network traffic generated for managing home appliances. However, in order to apply the mobile agent concept to the home network, access control for mobile agents is necessary. In the existing home network system, there is one special server, sometimes called home server. This server generally has mapping tables to be updated periodically, which describes access control lists between users' authorities and corresponding devices. In this paper, we propose a role-based access control framework with mobile agents in home networks. This framework, called Secure KAgent framework, is designed and implemented based on KAgent system. It has two main characteristics: to control access permissions based on Role-Based Access Control (RBAC) scheme and to safely assign roles to mobile agents by role tickets.

Key Words : Mobile Agents, Home Networks, Role-Based Access Control, RBAC

1. 서 론

홈 네트워크 환경[1,2]은 댁내의 PC를 비롯한 다수의 기기들이 네트워크로 통합된 첨단 생활 환경으로써 사용자는 네트워크를 통하여 댁내의 디지털 가전 기기를 제어 및 통제할 수 있다. 홈 네트워크 환경과 외부 네트워크를 연결하는 역할을 담당하는 홈 서버는 이러한 외부의 통신 채널을 통하여 홈 네트워크로 접근하는 사용자에 대한 인증 및 접근 제어를 담당할 뿐만 아니라, 댁내의 디지털 가전 기기의 통제 및 제어에 필요한 다양한 기능을 수행한다.

이동 에이전트[3,4]는 네트워크 상에서 스스로 이동하면서 사용자를 대신하여 작업을 수행하는 컴퓨터 프로그램이다. 이동 에이전트는 비동기적으로 작업을 수행하며, 네트워크 환경

※ 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기술개발사업의 지원에 의한 것임(2007-0266-0100).
† 준 회 원 : 성균관대학교 전자전기컴퓨터공학과 석사과정
†† 정 회 원 : 성균관대학교 이동통신공학과 연구교수
††† 정 회 원 : 동양대학교 정보통신공학부 교수
†††† 종신회원 : 성균관대학교 정보통신공학부 교수
논문접수 : 2007년 5월 11일, 심사완료 : 2007년 10월 2일

변화에 동적이고 유연하게 적응함으로써 작업에 필요한 통신 비용의 절감을 가능하게 한다. 이와 같은 특성을 지닌 이동 에이전트를 홈 네트워크 환경에 적용함으로써 홈 네트워크 환경에서 발생하는 네트워크 트래픽을 줄일 수 있을 것으로 기대된다.

그러나, 이동 에이전트를 홈 네트워크 환경에 적용하기 위해서는 홈 네트워크 환경으로 접근하는 이동 에이전트에 대한 접근제어가 반드시 선행되어야 한다[5,6,7]. 예를 들어 범죄를 목적으로 홈 네트워크에 접근하는 이동 에이전트가 댁내에 사람이 있는지를 확인하기 위해 센서 정보를 획득하려고 하면 홈 네트워크는 에이전트 소유자의 신원이 홈의 구성원이 아님을 확인하고 센서 정보에 대한 접근을 제한 해야 한다. 일반적인 홈 네트워크 시스템에서는 홈 서버는 디지털 기기와 사용자의 권한을 명시하는 접근제어 목록을 이용하여 홈 네트워크로 접근하는 사용자에 대한 접근제어를 수행한다. 이를 위해 홈 서버는 디지털 기기와 사용자의 권한 간의 최신 정보를 저장하기 위해 주기적으로 접근제어 목록을 갱신하는 추가적인 연산을 수행해야 한다.

본 논문에서는 홈 네트워크 환경에서 이동 에이전트의 역할에 기반한 접근제어 프레임워크인 Secure-KAgent 시스템을 제안한다. Secure-KAgent 시스템은 유비쿼터스 환경에 맞게 개발된 경량화된 이동 에이전트 시스템인 KAgent 시스템[8]을 확장한 것으로서 홈 네트워크 환경에 적합한 인증과 접근제어 기능이 강화되었다. Secure-KAgent 시스템은 이동 에이전트의 인증을 위해 공유키 기반 인증 기법을 사용하며 이동 에이전트 역할 기반 접근제어 기법(RBAC: Role-Based Access Control)[9,10,11]을 적용한다. Secure-KAgent 시스템은 RBAC을 적용함으로써 홈 서버에서 일괄적으로 관리되던 접근제어 목록을 각 디바이스가 독립적으로 관리할 수 있도록 분산시킨다. 따라서 홈 서버에서 접근제어 목록을 갱신하기 위해 필요한 연산 비용을 감소시킬 수 있다. 뿐만 아니라 각 디바이스마다 독립적으로 서비스 접근제어 정책을 유지함으로써 보다 더 능동적인 서비스 접근제어가 가능하며, 접근제어 정책의 특별한 수정 없이 새로운 장비를 홈 네트워크 환경에 추가할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 이동 에이전트 시스템에 관련된 기존 연구들을 살펴보고 3장에서는 제안한 Secure-KAgent 시스템의 접근제어 기법에 대해 설명한다. 4장에서는 제안한 접근제어 시스템의 성능을 평가한 뒤 마지막으로 5장에서는 결론을 맺는다.

2. RBAC을 적용한 이동 에이전트 시스템

본 절에서는 역할 기반 접근제어 기법을 적용한 대표적인 이동 에이전트 시스템에 대해서 언급하고 각각의 이동 에이전트 시스템의 특징과 활용 가능한 응용 환경 등에 대해서 살펴본다.

RCACM (Role-based Context Aware Coordination Model)[12]은 중국의 Jiao Tong 대학에서 개발한 이동 에이전트 시스템이다. RCACM은 분산된 자원, 서비스 및 객체 사이의 통신을 일관된 방법으로 지원하는 튜플 스페이스(tuple space)로 접근하는 이동 에이전트의 역할에 기반한 접근제어 기법을 제안한다. 튜플 스페이스는 튜플 스페이스의 환경정책에 부합한 역할을 가진 이동 에이전트에 대해서만 접근을 허가하며, 접근이 허가된 이동 에이전트들은 튜플에서 제공하는 자원을 활용하여 협업함으로써 적절한 결과를 도출한다. 튜플 스페이스를 구성하는 각각의 오브젝트(object)와 이동 에이전트는 모두 Java를 이용하여 작성되었다. RCACM은 군사적인 전략 시뮬레이션이나 원격 정보 검색 등의 협업이 필요한 다양한 환경에서 활용된다.

SoD(Sea of Data)[13]는 스페인의 UAB(Universitat Autonoma de Barcelona)에서 개발한 이동 에이전트 시스템이다. SoD는 MARISM-A(Mobile Agents with Recursive Itinerary and Secure Migration)를 확장하여 개발한 시스템으로 이동 에이전트를 이용한 전자상거래 환경에서 필요한 보안 기술을 제안한다. SoD는 SPKI(Simple Public Key Infrastructure)를 이용한 이동 에이전트의 인증과 이동 에이전트의 역할에 기반한 접근제어, 그리고 이동 에이전트에게 할당된 권한의 위임이 가능하다.

이탈리아의 Bologna 대학과 Ferrara 대학에서 공동으로 개발한 SOMA(Secure and Open Mobile Agent)[14]는 신임장을 이용한 이동 에이전트에 대한 공개키 기반의 인증 기법과 이동 에이전트의 역할에 기반한 접근제어 기법을 통해 악의를 가진 이동 에이전트로부터 호스트를 보호한다. 또한 분산 환경에서 나타날 수 있는 이동 에이전트에 대한 호스트의 공격으로부터 TTP(Trusted Third Party)와 MH(Multiple-Hops)를 이용한 보안 기법을 제안한다. SOMA는 이러한 보안 기술을 전자상거래 환경에 적용함으로써 이동 에이전트를 활용한 안전한 전자상거래 환경을 제공한다.

이탈리아의 UNIMORE(Modena e Reggio Emilia)대학에서 개발한 BRAIN(Behavioral Roles for Agent Interaction)[15]은 이동 에이전트를 이용한 대규모의 분산 시스템 환경에서 나타나는 요구 사항을 수용하고 이동 에이전트간의 협업에서 발생할 수 있는 문제를 해결하기 프레임워크로서 MASIF, RMI, CORBA, JXTA등을 지원한다. 특히, BRAIN은 이동 에이전트에게 부여된 역할을 기반으로 이동 에이전트간의 상호 작용 및 협업을 관리한다. BRAIN을 구성하는 각각의 오브젝트는 Java로 작성되었으며, XML을 이용하여 이동 에이전트에게 할당할 수 있는 역할을 기술한다.

RMAA(a Role-based Mobile Agent Approach to support e-democracy)[16] 역시 UNIMORE 대학에서 개발한 이동 에이전트 시스템으로 이동 에이전트를 이용한 e-Democracy 환경에서 이동 에이전트의 역할에 기반한 접근제어 기법을 제안한다. 특히, 실시간시에 이동 에이전트에게 동적으로 역할을 할당함으로써 환경 변화에 능동적으로 대처할 수 있는 접근제어 기법을 제안한다.

위에서 설명한 이동 에이전트 시스템은 모두 Java를 이용하여 작성되었으며, 이를 통해 뛰어난 이식성을 제공한다. 특히, 위 시스템들에서 사용한 역할 기반 접근제어는 권한의 관리를 분산시킴으로써 보다 유연하고 효과적인 접근제어가 가능하도록 한다. 그러나 이들 대부분의 시스템들은 이동 에이전트가 생성 되는 시점에 역할을 할당한다. 특히 홈 네트워크와 같이 각 디바이스 마다 서로 다른 접근제어 정책을 사용하는 환경에서는 해당 홈 네트워크 환경에 맞는 역할을 동적으로 할당할 수 있어야 하기 때문에 위에서 언급한 이동 에이전트 시스템들은 홈 네트워크에 적용하기에 한계를 나타낸다. 따라서 본 논문에서는 홈 네트워크 환경에서 적합한 이동 에이전트의 역할에 기반한 접근제어 기법을 제안한다.

3. Secure-KAgent 시스템

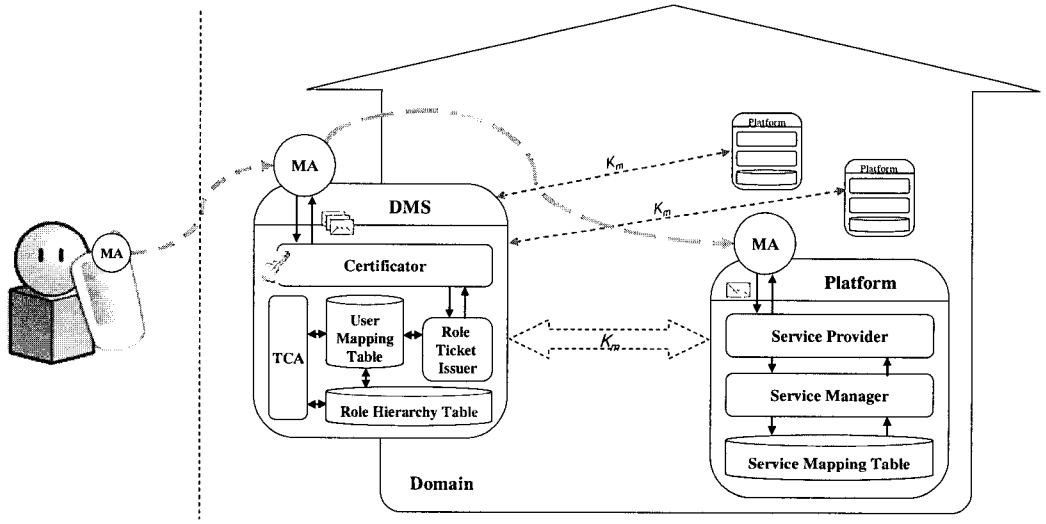
Secure-KAgent 시스템은 홈 네트워크 환경에서 활용 가능한 이동 에이전트 시스템으로써 Java를 기반으로 개발 되었으며, 데스크 탑 환경에서는 J2SE 1.3.2 이상의 버전을, 임베디드 장비에서는 J2ME Personal Profile/CDC 환경을 필요로 한다. Secure-KAgent 시스템은 시스템 내부로 접근하는 이동 에이전트에 대해 공유키 기반 인증 기법과 역할 기반 접근제어 기법을 이용한 보안 기능을 제공한다. 본 장에서는 Secure-KAgent 시스템의 접근제어 기법에 대해서 구체적으로 설명한다.

3.1 Secure-KAgent 접근제어 구조

Secure-KAgent 시스템은 DMS(Domain Management Server), 플랫폼, 그리고 이동 에이전트로 구성된다. 그림 1은 Secure-KAgent 시스템의 접근제어 구조를 나타낸다.

도메인은 하나의 접근제어 정책을 기반으로 DMS에 의해 관리되는 영역을 의미한다. DMS는 도메인의 게이트웨이로써 도메인에 접근 하는 이동 에이전트에 대한 인증 및 롤 티켓의 발급을 수행한다. 또한 DMS는 도메인에 속한 플랫폼들을 관리한다. DMS는 각 플랫폼과 보안 채널을 형성하며 이를 통해서 도메인 공유키(k_m)를 배포한다. 플랫폼은 이동 에이전트의 역할에 기반한 접근제어를 수행하며, 이동 에이전트에게 서비스를 제공한다. 표 1은 Secure-KAgent 시스템에서 서비스 접근제어를 위해 필요한 구성요소이다.

이동 에이전트가 플랫폼으로부터 서비스를 제공 받는 과정은 다음과 같다. DMS는 도메인으로 접근하는 이동 에이전트를 대상으로 인증을 수행한다. DMS의 Certificator(Cert)는 이동 에이전트의 authenticator를 이용하여 인증을 수행한다.



MA : Mobile Agent, DMS : Domain Management Server, TCA : Table Control Agent, K_m : Shared Key

(그림 1) Secure-KAgent 시스템의 접근제어 구조

〈표 1〉 Secure-KAgent 시스템의 서비스 접근제어 구성요소

위 치	구성 요소	기 능
DMS	Certificator (Cert)	도메인으로 접근하는 이동 에이전트에 대한 인증을 수행한다.
	Role Ticket Issuer (RTI)	인증에 성공한 이동 에이전트에게 롤 티켓의 생성 및 발급을 담당한다.
	User Mapping Table (UMT)	사용자의 신원에 기반하여 이동 에이전트에게 발급 가능한 역할들을 정의한다.
	Role Hierarchy Table (RHT)	각 역할들간의 상속 관계를 정의한다.
	Table Control Agent (TCA)	UMT와 RHT를 관리함으로써 정보의 신뢰성을 유지한다.
Platform	Service Provider (SP)	롤 티켓에 대한 인증과, 이동 에이전트에게 서비스를 제공하는 역할을 한다.
	Service Manager (SM)	이동 에이전트의 역할을 바탕으로 서비스에 대한 접근 권한을 확인한다.
	Service Mapping Table (SMT)	각 역할들과 서비스 접근 권한에 대한 사상 관계를 정의한다.

$$Authenticator = \{ AID \mid HPID \mid MD \mid TS \mid Cert_{K_U} \mid Sign_{K_R} (H(AID \mid HPID \mid MD \mid TS))\}$$

(그림 2) Authenticator의 구조

DMS는 인증에 성공한 이동 에이전트에게 롤 티켓을 발급한다. DMS의 RTI(Role Ticket Issuance)는 UMT(User Mapping Table)과 RHT(Role Hierarchy Table)을 참조하여 롤 티켓을 생성하고 발급하는 역할을 담당한다.

롤 티켓을 발급 받은 이동 에이전트는 플랫폼으로 이주하여 서비스를 요청한다. 플랫폼의 Service Manager(SM)는 이동 에이전트의 롤 티켓을 참조하여 요청한 서비스에 대한 접근 권한을 확인한다. 플랫폼의 SMT(Service Mapping Table)는 이러한 역할과 접근 권한에 대한 정보를 관리한다. Service Provider(SP)는 SM을 통해서 권한이 확인된 이동 에이전트를 대상으로 서비스를 제공한다.

3.2 이동 에이전트에 대한 인증

DMS는 도메인으로 들어오는 모든 이동 에이전트를 대상으로 인증을 수행한다. 인증은 이동 에이전트의 authenticator를 기초로 하여 이루어진다. Authenticator는 홈 플랫폼(Home Platform)에서 이동 에이전트가 생성될 때 만들어지며, 이동 에이전트 내부에 저장된다. 그림 2는 authenticator의 구조를 나타낸다.

Authenticator는 이동 에이전트의 ID인 AID와 홈 플랫폼의 ID인 HPID(Home Platform ID) 그리고 이동 에이전트의 실행 코드의 MD(Message Digest)를 포함한다. AID는 이동 에이전트가 생성될 때 홈 플랫폼으로부터 부여되며 시스템 내에서 유일하다. MD는 이동 에이전트의 클래스 파일들에 대하여 단방향 해시 함수를 적용하여 생성된 비트열이다. Authenticator는 이러한 값들을 홈 플랫폼의 개인키로 서명한 전자 서명(Digital Signature)과 이를 확인하기 위해 필요한 인증서를 포함한다. 또한 TS(Time Stamp)는 authenticator의 유효 기간을 나타낸다.

이동 에이전트는 DMS에게 authenticator를 전송한다. DMS의 Cert는 이동 에이전트의 authenticator를 이용하여 인증을 수행한다. 인증은 authenticator의 서명을 복호화하고 그 값의 무결성을 검증함으로써 이루어 진다. 복호화 과정에서 사용되는 공개키는 홈 플랫폼의 인증서로부터 획득한다. Authenticator를 통한 인증이 성공되면, Cert는 RTI에게 롤 티켓의 발급을 요청한다.

3.3 롤 티켓 발급

DMS의 RTI는 인증에 성공한 이동 에이전트를 대상으로 롤 티켓을 발급한다. RTI는 UMT와 RHT를 참조하여 이동 에이전트에게 발급 가능한 롤 티켓을 생성하고 이를 할당한다. UMT는 사용자와 역할간의 관계를 정의함으로써 특정 사용자에게 어떠한 역할을 부여할 수 있는지에 대한 정보를 XML기반으로 관리한다. UMT는 도메인 형성 시, 관리자에 의해 생성된다. 그림 3은 홈 네트워크 환경에서 사용자와 역할간의 관계를 정의한 UMT를 보인다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT UserMappingTable((UserMappingTableSpec)+)>
<!ELEMENT UserMappingTableSpec(User)*>
<!ATTLIST User Name ID #REQUIRED>
<!ELEMENT User (Role)*>
<!ATTLIST User Name ID #REQUIRE+D>
<!ELEMENT Role EMPTY>
<!ATTLIST Role RoleName CDATA #REQUIRED
```

(a) 사용자와 역할간의 관계를 표현하기 위한 DTD

```
<?xml version "1.0" encoding="UTF-8"?>
<!DOCTYPE UserMappingTable SYSTEM "UserMappingTable.dtd">
<UserMappingTable>
<UserMappingTableSpec Type = "DClabUser">
    <User Name="YoungwooJung">
        <Role RoleName="SystemAdmin"/>
        <Role RoleName="FamilyMember"/>
    </User>
    <User Name="WonheeKim">
        <Role RoleName="FamilyMember"/>
    </User>
    <User Name="HyunsookPark">
        <Role RoleName="FamilyMemberAdult"/>
    </User>
    <User Name="GusungJung">
        <Role RoleName="FamilyMemberAdult"/>
    </User>
    <User Name="SKKU">
        <Role RoleName="PublicServant"/>
    </User>
</UserMappingTableSpec>
</UserMappingTable>
```

(b) 사용자와 역할간의 관계를 표현한 XML

(그림 3) 홈 네트워크 환경에서의 User Mapping Table

도메인 내에서 사용되는 각 역할들은 계층 관계를 유지한다. RHT는 이러한 역할들간의 계층 관계를 정의한다. 역할들간의 계층 관계는 권한의 상속을 가능하게 한다. 예를 들어, 그림 3에서 사용한 "FamilyMemberAdult"와 "FamilyMemberTeenager"는"FamilyMember"의 권한을 상속 받는다. Secure-KAgent 시스템에서는 이러한 역할들간의 계층 관계를 XML로 표현하고 관리한다. 그림 4는 홈 네트워크 환경에서의 RHT를 위한 DTD를 나타낸다.

RTI는 UMT와 RHT를 참조하여 이동 에이전트에게 발급 가능한 역할들을 확인하고, 각 역할에 따라 롤 티켓을 발급한다. 롤 티켓은 이동 에이전트의 ID 와 역할의 이름 그리고 서명한 부분으로 구성된다. 서명은 이동 에이전트의 ID와 역할의 이름 그리고 MD에 해시 함수를 적용한 뒤, 도메인 공유키로 암호화 하여 수행한다. 그림 5는 롤 티켓의 구조를 나타낸다.

RTI는 각각의 역할마다 롤 티켓을 따로 발급한다. 다수의 역할에 대해 하나의 롤 티켓을 발급하게 되면 이동 에이전트가 서비스 요청을 위해 플랫폼에게 롤 티켓을 전달할 때 불필요한 정보까지 플랫폼에게 노출된다. 따라서 이를 방지 하기 위해 하나의 롤 티켓은 하나의 역할에 대해서만 발급된다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT RoleHierarchyTable (RoleSpec)+>
<!ELEMENT RoleSpec (Role)+>
<!ATTLIST RoleSpec Type ID #REQUIRED>
<!ATTLIST RoleSpec MaxDepth CDATA #REQUIRED>
<!ELEMENT Role ((Extends)*, (SSOD)*, (DSOD)*)>
<!ATTLIST Role RoleName ID #REQUIRED>
<!ELEMENT Extends EMPTY>
<!ATTLIST Extends RoleName IDREF #REQUIRED>
<!ELEMENT SSOD EMPTY>
<!ATTLIST Extends RoleName IDREF #REQUIRED>
<!ELEMENT DSOD EMPTY>
<!ATTLIST Extends RoleName IDREF #REQUIRED>
```

(그림 4) 홈 네트워크 환경의 Role Hierarchy Table을 위한 DTD

$$T = (AID \mid NAME_{role} \mid TS \mid E_{K_m}(H(AID \mid NAME_{role} \mid TS \mid MD)))$$

(그림 5) 롤 티켓의 구조

3.4 이동 에이전트의 역할에 기반한 서비스 접근제어

롤 티켓을 발급 받은 이동 에이전트는 플랫폼으로 이주하여 서비스를 요청한다. 이동 에이전트는 서비스 요청에 필요한 롤 티켓과 서비스 관련 메시지들을 플랫폼에게 전달한다. 플랫폼의 SP는 롤 티켓에 대한 무결성 검사를 수행한다. 무결성 검사는 롤 티켓에 이루어진 DMS의 서명을 확인함으로써 이루어진다. 그림 6은 롤 티켓에 대한 서명 확인 알고리즘을 보인다.

SP는 (1)단계를 통해서 롤 티켓의 TS를 확인하고 유효한 롤 티켓으로 확인되면 (2), (3), (4) 단계를 통하여 롤 티켓에 저장된 정보를 추출한다. (2), (3) 단계에서 추출된 값은 이동 에이전트로부터 전달 받은 MD와 함께 (5)단계에서 해시함수를 적용한다. 또한 도메인 공유키를 이용하여 DMS의 서명을 복호화한다. SP는 (5)단계와 (6)단계에서 나온 결과 값을 비교한다. 비교한 결과 값이 같지 않을 경우, SP는 부적합한 롤 티켓으로 판단하고 이동 에이전트의 서비스 접근을 거부한다. 롤 티켓의 무결성이 확인되면, SP는 이동 에이전트가 요청한 서비스 ID와 롤 티켓으로부터 얻은 역할의 이름을 SM에게 전송한다.

SM는 SMT를 확인하여 이동 에이전트가 요청한 서비스에 대한 권한을 확인한다. SMT는 각각의 역할에 따른 서비스의 접근 권한에 대한 정보를 관리한다.

```
verifyRoleTicket(MD', T) {
(1)   if (verifyTS (TS'=getTS(T)) == false)
          return false;
(2)   AID'          = getAID(T) ;
(3)   NAME_role'    = getRoleName(T) ;
(4)   Sign'         = getSign(T) ;
(5)   a = H(AID' | NAME_role' | TS' | MD') ;
(6)   b = D_Km (Sign') ;
(7)   if (a == b)
          return true ;
      else
          return false;
}
```

(그림 6) 롤 티켓에 대한 서명 확인 알고리즘

Secure-KAgent 시스템은 이러한 역할과 권한과의 사상정보를 XML을 이용하여 관리한다. SM는 요청한 서비스에 대한 권한이 확인되면 SP로 하여금 이동 에이전트에게 서비스를 제공할 수 있도록 권한 정보를 전송한다. SP는 이동 에이전트에게 서비스 ID와 권한 정보를 전송함으로써 이동 에이전트에게 서비스에 대한 권한을 부여한다.

4. 제안 기법 검증

본 장에서는 본 논문에서 제안한 이동 에이전트의 역할에 기반한 서비스 접근제어 기법을 검증한다. 본 장은 DMS에서의 이동 에이전트 인증과정, 이동 에이전트에게 롤 티켓을 발급하는 과정 그리고 플랫폼에서의 접근제어 과정에 대한 검증으로 구성된다. 검증에 앞서 본 기법에서 사용한 전자 서명은 충분히 안전하다고 가정하며, 따라서 전자 서명은 공격자의 위변조가 불가능하다고 가정한다. 표 2는 제안 기법 검증에 사용되는 Notation을 나타낸다.

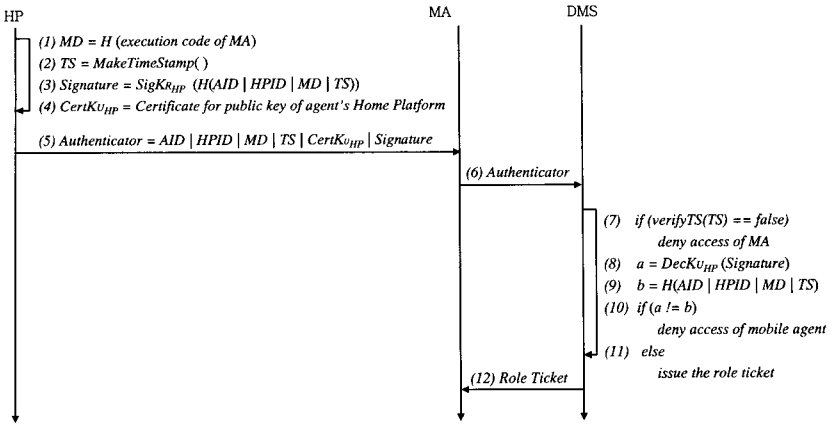
4.1 DMS의 이동 에이전트 인증에 대한 검증

DMS는 도메인으로 접근하는 이동 에이전트에 대한 인증을 수행한다. 그림 7은 DMS에서의 이동 에이전트 인증 프로토콜을 보인다.

홈 플랫폼은 (1), (2), (3), (4) 단계를 통하여 생성된 값을 이용하여 authenticator를 생성한다. 생성된 authenticator는 (5)단계와 같이 이동 에이전트 내부에 저장되며 인증을 위해 DMS에게 전송된다. DMS는 authenticator의 TS를 확인하고 유효한 authenticator로 확인되면 홈 플랫폼의 공개키를 이용

〈표 2〉 제안 기법 검증에 사용되는 Notation

Notation	설 명
$(k_R, k_U)_i$	홈 플랫폼 i의 개인키(k_R)와 공개키(k_U) 쌍
$Sigk_{Ri}(m)$	메시지 m을 홈 플랫폼 i의 개인키 k_R로 서명
$Deck_{Ui}(m)$	메시지 m을 홈 플랫폼 i의 공개키 k_U로 복호화
k_m	도메인 공유키
$M(a_1, a_2, \cdots, a_n)$	원소 $a_1, \cdots, a_n$으로 구성된 메시지를 생성
$H(m_1, m_2, \cdots, m_n)$	메시지 $m_1, \cdots, m_n$ 에 대한 One-way hash 값을 생성
$Ext(m, a_i)$	메시지 m으로부터 원소 a_i를 추출
$Adv(m)$	메시지 m에 대한 공격자의 공격
$A \rightarrow B:m$	A로부터 B로의 메시지 m 전송



(그림 7) DMS에서의 이동 에이전트 인증 프로토콜

하여 서명을 복호화한다. 이 때 사용되는 공개키는 홈 플랫폼의 인증서로부터 획득한다. 또한, DMS는 authenticator로부터 추출한 AID, HPID, MD에 해시함수를 적용한다. DMS는 인증을 위해 (8)단계와 (9)단계를 통해 얻은 결과값을 비교한다. 비교한 결과값이 같지 않을 경우, DMS는 인증에 실패한 것으로 판단하고 이동 에이전트의 도메인 접근 요구를 거부한다. 비교한 값이 같을 경우, DMS는 authenticator의 무결성을 확인하는 동시에 홈 플랫폼의 인증서를 통하여 authenticator가 HPID로부터 발급되었음을 확인하게 된다. 인증에 성공하면 Cert는 RTI에게 롤 티켓 발급을 요청한다.

정리 1. 본 인증 프로토콜은 메시지 변형 공격으로부터 안전하다.

증 명:

HP : $DF_0=M(AID, HPID, MD, TS)$
$SF_0=Sigk_{Ri}(H(DF_0))$
$m_0=M(DF_0, SF_0)$

$HP \rightarrow MA$: m_0

$MA \rightarrow DMS$: m_0

DMS : $COMPARE(H(Ext(m_0, DF_0)), Deck_{Ui}(Ext(m_0, SF_0)))$

if $m_{adv} = Adv(m_0)$ then

DMS: $COMPARE(H(Ext(m_{adv}, DF)), Deck_{Ui}(Ext(m_{adv}, SF)))$

$Deck_{Ui}(Ext(m_{adv}, SF)) = Deck_{Ui}(Ext(m_0, SF_0))$

(∵Signature can not be modified)

Therefore, $H(Ext(m_{adv}, DF)) \neq Deck_{Ui}(Ext(m_0, SF_0))$

따라서 DMS는 원본 메시지 m_0가 변형되었음을 확인하고 인증을 거부한다.

정리 2. 본 인증 프로토콜은 메시지 재전송 공격으로부터 안전하다.

증 명:

$Adversary$: $m_{captured}= Adv(m_0)$

$Adversary \rightarrow MA$: $m_{captured}$

$MA \rightarrow DMS$: $m_{captured}$

DMS : $COMPARE(Ext(m_{captured}, TS), TIME)$: $m_{captured}$

Therefore, DMS makes sure expired TS of $m_{captured}$: $m_{captured}$

따라서 DMS는 원본 메시지 m_0가 재전송 되었음을 확인하고 인증을 거부한다.

정리 3. 본 인증 프로토콜은 메시지 송, 수신 부인 공격으로부터 안전하다.

증 명:

HP : $DF_0=M(AID, HPID, MD, TS)$
$m_0=M(DF_0, Sigk_{Ri}(H(DF_0)))$

$HP \rightarrow MA$: m_0

$MA \rightarrow DMS$: m_0

DMS : $COMPARE(H(Ext(m_0, DF_0)), Deck_{Ui}(Sigk_{Ri}(H(DF_0))))$

Therefore, HP signs a message with its private key and DMS make sure a message with public key of Home platform i

따라서 전자서명의 일반적인 성질에 의해 HP와 DMS는 각각 송, 수신을 부인할 수 없다.

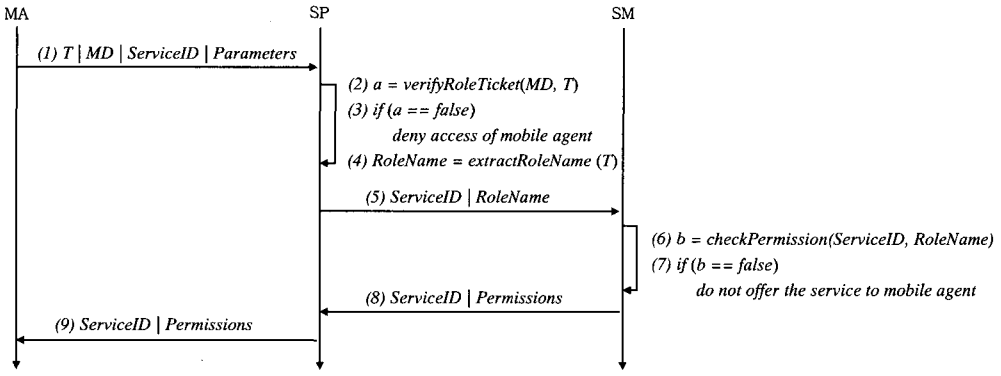
4.2 이동 에이전트에게 부여된 역할에 대한 검증

DMS는 롤 티켓을 통하여 이동 에이전트에게 역할을 부여한다. 롤 티켓은 DMS의 UMT와 RHT를 기반으로 이동 에이전트에게 부여된다. UMT와 RHT는 도메인이 구성될 때 관리자에 의해서 생성되며, TCA에 의해 관리됨으로써 역할에 관련된 정보의 무결성과 신뢰성을 보장한다. RTI는 UMT와 RHT를 참조하여 롤 티켓을 생성하고 이는 이동 에이전트 내부에 저장되며, 이 과정은 DMS로부터 제공되는 안전한 보안 메커니즘을 통해 이루어진다.

4.3 플랫폼에서의 접근제어에 대한 검증

플랫폼의 SP와 SM은 롤 티켓을 이용하여 이동 에이전트에 대한 접근제어를 수행한다. 그림 8은 플랫폼에서의 접근제어 프로토콜을 나타낸다.

이동 에이전트는 롤 티켓(T), 이동 에이전트의 MD, 요청한 서비스에 대한 ID 그리고 서비스 이용에 필요한 파라미터를 (1)단계와 같은 형태로 SP에게 전송한다. SP는 (2)단계를 통하여 롤 티켓에 대한 무결성을 확인한다. 적합한 롤 티켓으로 판단되면, SP는 요청한 Service ID와 (4)단계에서 추출한 역할의 이름을 SM에게 전송한다. SM는 전송 받은 역할의 이름을 참조하여 요청한 서비스에 대한 접근 권한을 확인한다. 역할에 대한 접근 권한이 확인되면, SM는 Service ID와 확인된 권한을 SP에게 전송한다. SP는 이동 에이전트



(그림 8) 플랫폼에서의 접근 제어 프로토콜

에게 서비스 ID와 권한 정보를 전송함으로써 이동 에이전트에게 서비스에 대한 권한을 부여한다.

정리 4. 본 접근제어 프로토콜은 메시지 변형 공격으로부터 안전하다.

증 명:

$RTI \rightarrow MA$: $DF_0=M(AID, NAME_{role}, TS)$
$SF_0 = Sigk_m(H(DF_0, MD))$
$T=M(DF_0, SF_0)$

MA : $m_0=M(T, MD, ServiceID, Parameters)$

$MA \rightarrow SP$: m_0

SP : $t_0=Ext(m_0, T)$
$COMPARE(H(Ext(t_0, DF_0), MD), Deck_m(Ext(t_0, SF_0)))$

if $t_{adv} = Adv(t_0)$ *then*
$DMS : COMPARE(H(Ext(t_{adv}, DF), MD), Deck_m(Ext(t_{adv}, SF)))$
$Deck_m(Ext(t_{adv}, SF)) = Deck_m(Ext(t_0, SF_0))$
$(\because Signature\ can\ not\ be\ modified\ )$
$Therefore,\ H(Ext(t_{adv}, DF), MD) \neq Deck_m(Ext(t_0, SF_0))$

따라서 SP는 원본 메시지 t_0가 변형되었음을 확인하고 MA에 대한 권한 할당을 거부한다.

정리 5. 본 접근제어 프로토콜은 이동 에이전트 코드 변조 공격으로부터 안전하다.

증 명:

$RTI \rightarrow MA$: $DF_0=M(AID, NAME_{role}, TS)$
$SF_0 = Sigk_m(H(DF_0, MD_{RTI}))$
$T=M(DF_0, SF_0)$

MA : $m_0=M(T, MD_{MA}, ServiceID, Parameters)$

$MA \rightarrow SP$: m_0

SP : $t_0=Ext(m_0, T)$
$COMPARE(H(Ext(t_0, DF_0), MD_{MA}), Deck_m(Ext(t_0, SF_0)))$

if $MD_{adv} = Adv(MD_{MA})$ *then*
$DMS : COMPARE(H(Ext(t_0, DF_0), MD_{adv}), Deck_m(Ext(t_0, SF_0)))$
$Deck_m(Ext(t_0, SF_0)) = Deck_m(SF_0)$
$= Deck_m(Sigk_m(H(DF_0, MD_{RTI})))$
$= H(DF_0, MD_{RTI})$
$Therefore,\ H(Ext(t_0, DF_0), MD_{adv}) = H(DF_0, MD_{adv})$
$\neq H(DF_0, MD_{RTI})$

따라서 SP는 MA의 코드가 변조되었음을 확인하고 MA에 대한 권한 할당을 거부한다.

정리 6. 본 인증 프로토콜은 메시지 재전송 공격으로부터 안전하다.

증 명:

$Adversary$: $m_{captured} = Adv(m_0)$

$Adversary \rightarrow MA$: $m_{captured}$

$MA \rightarrow SP$: $m_{captured}$

SP : $t_0=Ext(m_{captured}, T)$
$COMPARE(Ext(t_0, TS), TIME)$

$Therefore,\ SP\ makes\ sure\ expired\ TS\ of\ m_{captured}$

따라서 SP는 원본 메시지 t_0가 재전송 되었음을 확인하고 MA에 대한 권한 할당을 거부한다.

5. 결 론

본 논문에서는 홈 네트워크 환경에서 이동 에이전트의 역할에 기반한 접근제어 프레임워크인 Secure-KAgent 시스템을 제안하였다. 본 시스템은 RBAC을 기초한 접근제어를 수행함으로써 이동 에이전트의 역할에 따라 그에 적합한 권한을 부여하고 서비스를 제공한다. 이를 위해 본 시스템은 롤 티켓을 이용함으로써 이동 에이전트에게 안전한 권한 부여를 보장한다.

본 논문에서는 이에 대한 검증을 위해 롤 티켓의 발급 과정과 이를 이용한 서비스 접근제어 과정의 안전성을 평가하였다. 특히, 롤 티켓 발급 과정에서 나타날 수 있는 메시지 변형 공격, 메시지 재전송 공격 등의 안전성을 평가하였다. 또한, 이동 에이전트를 이용한 서비스 접근제어 과정에서 나타날 수 있는 이동 에이전트 코드 변조 공격 등에 대한 안정성 역시 평가하였다.

참 고 문 헌

[1] B. Rose, *Home Networks: A Standard Perspective*, IEEE Communication Magazine, pp. 78-85, 2001.

[2] Digital Home Working Group, Digital Home. White Paper, http://www.dhwg.org, June 2003.

[3] A. Fuggetta, G. Picco, and G. Vigna, "Understanding code mobility," *IEEE Trans. on Software Eng.*, Vol.

24(5), pp. 352-361, 1998.

[4] A. Aneiba and J. S. Rees, "Mobile Agent Technology and Mobility," *Proc. of the 5th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet 2004)*, 2004.

[5] G. Karjoth, D. Lange, and M. Oshima, "A Security Model for Aglets," *IEEE Trans. on Internet Computing*, Vol. 4, pp. 68-77, 1997.

[6] N. M. Karnik and A. R. Tripathi, "A Security Architecture for Mobile Agents in Ajanta," *Proc. of the 20th Int'l Conf. on Distributed Computing Systems*, 2000.

[7] M. A. Mazlan, A. Samsudin, and R. Budiarto, "Secure Groups Communication for Mobile Agents Based on Public Key Infrastructure," *Proc. of the 9th Asia-Pacific Conf. on Communications (APCC 2003)*, 2003.

[8] H. J. Cho, G. S. Kim, K. C. Kim, H. S. Shim, and Y. I. Eom, "Development of Lightweight Mobile Agent Platform for URC Environments," *Proc. of the 2nd Int'l Conf. on Ubiquitious Robots and Ambient Intelligence*, 2005.

[9] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," *ACM Trans. on Information and System Security (TISSEC)*, Vol. 4(3), pp. 224-274, 2001.

[10] G. Cabri, L. Ferrari, and L. Leonardi, *Applying security policies through agent roles: A JAAS based approach*, Science of Computer Programming (Article in Press).

[11] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Younman, "Role-based access control model," *IEEE Trans. on Computer*, Vol. 20(2), pp. 38-47, 1996.

[12] T. Xinhuai, Z. Yaying, and Y. Jinyuan, "RCACM: Role-Based Context Coordination Model for Mobile Agent Applications," *Proc. of the 2nd Int. Workshop on Grid and Cooperative Computing*, 2003.

[13] G. Navarro, S. Robles, and J. Borrell, "Role-Based Access Control for E-commerce Sea-of-Data Applicaton," *Proc. of the Information Security Conference 2002*, 2002.

[14] A. Corradi, R. Montanari, and C. Stefanelli, "Security Issues in Mobile Agent Technology," *Proc. of the 7th IEEE Workshop on Future Trends of Distributed Computing Systems*, 1999.

[15] G. Cabri, L.Ferrari, and F. Zambonelli, "Role-Based Approaches for Engineering Interactions in Large-scale Multi-Agent System," *Post-Proc. of Advances in Software Engineering for Large-Scale Multiagent Systems(SELMAS 03)*, 2003.

[16] G. Cabri, L. Ferrari, and L. Leonardi, *A role-based mobile-agent approach to support E-democracy*, Science of Computer Programming (Article in Press).

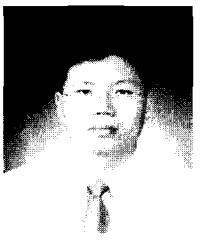
정 용 우

e-mail : withdubu@ece.skku.ac.kr

2006년 성균관대학교 정보통신공학부(학사)

2006년~현 재 성균관대학교 대학원 전자전기컴퓨터공학과 석사과정

관심분야 : 이동 에이전트, 이동 에이전트 시스템 보안, 유비쿼터스 컴퓨팅 등



고 광 선

e-mail : rilla91@ece.skku.ac.kr

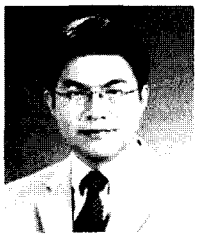
1998년 성균관대학교 정보공학과(학사)

2004년 성균관대학교 대학원 전기전자및컴퓨터공학부(공학석사)

2007년 성균관대학교 대학원 전자전기컴퓨터공학과(공학박사)

2007년~현 재 성균관대학교 대학원 이동통신공학과 연구교수

관심분야 : 정보보호, 리눅스, 네트워크 등



김 구 수

e-mail : gusukim@dyu.ac.kr

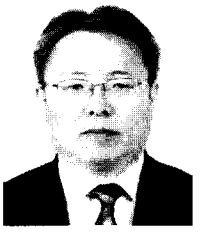
1994년 성균관대학교 정보공학과 (학사)

1996년 성균관대학교 정보공학과 (공학석사)

2006년 성균관대학교 정보통신공학과 (공학박사)

2007년~현 재 동양대학교 정보통신공학부 교수

관심분야 : 분산 컴퓨팅, 이동 에이전트 등



엄 영 익

e-mail : yieom@ece.skku.ac.kr

1983년 서울대학교 계산통계학과(학사)

1985년 서울대학교 전산과학과(이학석사)

1991년 서울대학교 전산과학과(이학박사)

2000년~2001년 Dept. of Info. and Comm. Science at UCI 방문교수

2005년 한국정보처리학회 학회지 편집위원장

1993년~현 재 성균관대학교 정보통신공학부 교수

관심분야 : 분산 컴퓨팅, 이동 컴퓨팅, 이동 에이전트, 시스템 보안, 운영체제, 내장형 시스템 등