情報保護學會論文誌
第14券 第5號, 2004. 10

홈 네트워크 환경에서 다중 도메인을 지원하는 공유키 및 공개키 기반의 이동 에이전트 인증 기법*

김 재 곤†, 김 구 수, 엄 영 익‡
성균관대학교

Shared Key and Public Key based Mobile Agent Authentication Scheme supporting Multiple Domain in Home Network Environments*

Jae-gon Kim†, Gu Su Kim, Young Ik Eom‡
Sungkyunkwan University

요 약

홈 네트워크 환경은 컴퓨터 시스템, 디지털 가전, 이동 단말과 같은 디지털 홈 디바이스들을 연결한 네트워크 환경으로 정의된다. 이러한 홈 네트워크 환경에는 홈 네트워크 디바이스들을 모니터링하고 제어하기 위한 수많은 지역 및 원격 상호작용이 발생할 것이며 홈 게이트웨이는 이러한 환경에서 통신 병목 지점이 될 것이다. 컴퓨팅 디바이스들 간을 스스로 이동하면서 사용자를 대신하여 작업을 수행할 수 있는 이동 에이전트를 홈 네트워크 환경에 적용함으로써 홈 네트워크 환경에서 발생하는 원격 상호 작용 및 네트워크 트래픽을 줄일 수 있다. 이동 에이전트 인증은 이동 에이전트 개념을 홈 네트워크 환경에 적용하기 위해 먼저 해결되어야 할 문제로서, 홈 네트워크 디바이스 및 자원에 대한 권한 부여 및 접근 제어의 전제가 되는 기술이다. 기존의 이동 에이전트 시스템들은 낮은 컴퓨팅 능력을 갖고 있는 디지털 디바이스로 구성된 홈 네트워크 환경에 적절하지 않은 공개키 기반 인증 기법을 주로 적용한다. 본 논문에서는 단일 홈 도메인을 위한 공유키 기반 이동 에이전트 인증 기법을 제안하고 이를 공개키 기반 인증 기법을 이용하여 다중 도메인 환경으로 확장시킨다. 본 제안 기법은 공유키 암호화 기법을 이동 에이전트의 도메인 내부 인증에 적용함으로써 공개키 암호화 기법을 이용하는 인증에 비하여 가벼운 연산만으로 인증을 수행한다.

ABSTRACT

The home network environment can be defined as a network environment, connecting digital home devices such as computer systems, digital appliances, and mobile devices. In this kind of home network environments, there will be numerous local/remote interactions to monitor and control the home network devices and the home gateway. Such an environment may result in communication bottleneck. By applying the mobile agents that can migrate among the computing devices autonomously and work on behalf of the user, remote interactions and network traffics can be reduced enormously. The mobile agent authentication is necessary to apply mobile agent concept to the home network environments, as a prerequisite technology for authorization or access control to the home network devices and resources. The existing mobile

접수일 : 2004년 6월 2일 ; 채택일 : 2004년 10월 8일
* 본 논문은 과학기술부 프론티어사업의 유비쿼터스컴퓨팅 및네트워크원천기반기술개발 과제로 수행된 결과입니다.
† 주저자 : angel77@ece.skku.ac.kr
‡ 교신저자 : yieom@ece.skku.ac.kr

agent systems have mainly used the public key based authentication scheme, which is not suitable to the home network environments, composed of digital devices of limited computation capability. In this paper, we propose a shared key based mobile agent authentication scheme for single home domain and expand the scheme to multiple domain environments with the public key based authentication scheme. Application of the shared key encryption scheme to the single domain mobile agent authentication enables to authenticate the mobile agent with less overhead than the public key based authentication scheme.

keyword : *Mobile Agent, Authentication, Home Network, Shared Key*

Ⅰ. 서 론

홈 네트워크 환경은 댁내의 PC 관련 기기들은 물론 냉장고, 세탁기, 디지털 TV 등의 가전 기기 그리고 휴대폰, PDA 등의 이동 단말 기기들을 원격 접속 및 원격 제어가 가능하도록 연결한 네트워크 환경으로서 기존의 인터넷, 모바일 통신, 그리고 방송 통신이 상호 운영성이 있는 네트워크로 통합되는 첨단 생활환경이다[1,2]. 가정의 PC가 한 대나 두 대에 불과했던 기존의 홈 환경과는 달리 여러 디바이스들이 네트워크로 연결되는 홈 네트워크에서는 이들 디바이스들을 제어하고 모니터링하기 위한 수많은 원격 상호작용이 발생할 것이며 이는 홈 네트워크를 외부와 연결해 주는 게이트웨이에 상당한 부하를 발생시킬 것이다.

이동 에이전트는 네트워크 상에서 스스로 이동하면서 사용자 또는 다른 개체 대신 행동할 수 있는 컴퓨터 프로그램으로서 실행 중에 자원이 있는 노드로 이동하여 작업을 수행할 수 있는 특징을 가지고 있다[3]. 그러한 이동 에이전트를 홈 네트워크 디바이스의 제어나 모니터링에 사용함으로써 홈 네트워크와 외부와의 원격 상호 작용의 횟수를 줄이고 홈 네트워크들 간의 트래픽을 줄일 수 있다[4]. 또한 이동 에이전트가 사용자를 추적하면서 사용자가 즐겨 사용하는 응용이나 중요한 데이터를 함께 이동시킴으로써 사용자에게 최적화된 컴퓨팅 환경을 자동적으로 만들어 주는 새로운 서비스도 제공할 수 있다[5].

이동 에이전트 인증 문제는 이동 에이전트를 홈 네트워크에 적용하기 위해서 먼저 해결되어야 할 문제이다. 호스트로 이동되어 올 에이전트를 인증함으로써 에이전트가 사용할 수 있는 홈 네트워크 자원에 대한 접근 제어가 가능해진다. 예를 들어 범죄를 목적으로 홈 네트워크 외부로부터 이주해 온 악의를 가진 에이전트가 댁내에 사람이 있는지를 확인하기 위해 센서 정보를 획득하려고 하면 홈 네트워크는 에이전트 소유자의 신원이 홈의 구성원이 아님을 확인하고 센서 정보에 대한 접근을 제한해야 한다. 또 다른 예로서 가족 내의 미성년자가 생성한 에이전트가 홈 네트워크 디바이스를 이용하여 성인정보에 접근하려고하면 홈 네트워크는 에이전트 소유자의 신원을 확인하여 성인이 아님을 확인하고 디바이스에 대한 제어권을 제한해야 한다.

지금까지 이동 에이전트 인증 기능을 지원하는 여러 시스템들이 개발되어 왔으며 이들 대부분이 공개키에 기반한 인증 기법을 사용한다[6,7]. 그러나 공개키 기반 인증 기법은 매우 많은 연산을 필요로 하기 때문에 데스크탑 컴퓨터보다 제한적인 성능을 갖는 홈 네트워크 디바이스에 적용하기에 적절하지 않다.

본 논문에서는 공개키 암호화 기법보다 적은 비용이 소요되는 공유키 기반 인증 기법을 적용함으로써 홈 네트워크 환경에 알맞은 이동 에이전트 인증 기법을 제안한다. 또한 공개키 기반 인증 기법을 사용하여 여러 홈 네트워크 간에서 사용될 수 있도록 이를 확장한다. 본 논문의 구성은 다음과 같다. 2장에서는 인증의 대상, 인증 알고리즘 그리고 그룹키 관리 기법에 관한 기존 연구들을 살펴본다. 3장에서는 보안 요구 사항 및 시스템 구성을 설명한다. 4장에서는 제안 인증 기법을 설명하고 5장에서는 제안 인증 기법의 유효성을 검증한 뒤 마지막으로 6장에서는 결론 및 향후 계획을 설명한다.

Ⅱ. 관련 연구

2.1 인증 대상의 종류

이동 에이전트의 이주 요청의 종류는 이주 결정을 내리는 주체에 따라서 이동할 에이전트 자신이 이주를 결정하는 경우와 에이전트를 실행하고 있는 플랫폼이 이동 에이전트의 이주를 결정하는 경우로 나뉘어 질 수 있다. 각각의 경우에 따라서 에이전트에 대한 인증을 수행할 것인지 아니면 플랫폼에 대한 인증

을 수행할 것인지가 달라진다. Shimson Berkovits는 이를 place hand-off/delegation과 agent hand-off / delegation으로 분류하였다[8]. Place hand-off / delegation은 플랫폼이 자신이 실행시키고 있는 에이전트를 이주시킬 것을 결정하는 경우에 해당하며, 목적지 플랫폼은 이주 요청을 전송한 플랫폼에 대하여 인증을 수행한다. Agent hand-off/delegation은 에이전트 스스로 이주 결정을 내릴 경우에 해당하며 목적지 플랫폼은 이주 요청을 전송한 에이전트에 대하여 인증을 수행한다.

2.2 인증을 위한 암호화 기법

인증을 위해 사용될 수 있는 암호화 기법으로는 공유키 암호화 기법과 공개키 암호화 기법이 있다[9]. 공유키 암호화 기법을 이용한 인증 방법은 인증 요청자가 공유키로 메시지를 암호화하면 상대방은 동일한 공유키로 메시지를 복호화함으로써 양자가 동일한 키를 공유하고 있음을 증명한다. 그러한 이유로 공유키 기반의 인증 기법은 키를 공유하고 있는 그룹에 대한 인증이다. 공유키 암호화 기법이 갖는 장점은 암호화 및 복호화 속도가 공개키 알고리즘에 비해 매우 빠르다는 점이다. 그러나 공유키 인증 기법은 공유되는 키가 그룹 밖으로는 절대적으로 비밀이 유지되어야 한다는 점 때문에 전역적으로 사용되기 어렵다. 반면 공개키 암호화 기법을 이용한 인증 방법은 한 키는 공개되어 있고 또 다른 키는 비밀이 유지되는 키쌍을 이용한다. 비밀키를 이용하여 메시지를 암호화하고 공개키를 이용하여 복호화함으로써 메시지가 비밀키를 유일하게 보유하고 있는 주체에 의하여 만들어졌다는 것을 인증한다. 공개키 기반 인증 기법의 장점은 공개키는 누구라도 소유할 수 있기 때문에 공개키 기반 인증을 네트워크 상에서 간단하게 전역적으로 수행할 수 있다는 점이다. 그러나 앞서 말한 바와 같이 공개키 알고리즘은 지수 연산과 같은 매우 많은 연산을 수행하기 때문에 공유키 암호화 기법에 비하여 느리다는 단점이 있다. 최근에는 그러한 공개키 알고리즘의 단점을 극복하기 위하여 타원곡선 암호시스템(ECC)[10], NTRU[11,12], 고차 잉여류를 이용한 공개키 암호 시스템[13] 등이 연구되고 있다.

2.3 그룹키 관리 기법

그룹키 관리 기법으로는 SKDC(Simple Key Distribution Center), GDH(Group Diffie-Hellman), LKH(Logical Key Hierarchy), OFT (One-way Function Tree) 기법 등이 있다[14,15]. 본 논문의 제안 인증 시스템에서는 OFT를 그룹키 관리 기법으로 가정하므로 본 절의 나머지 부분에서는 OFT 그룹키 관리 기법에 대하여 설명한다.

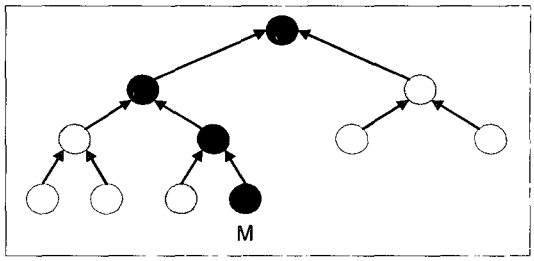


그림 1. OFT 키 트리

OFT 그룹키 관리 기법은 그림 1에서 보이는 이진 트리에 상향식으로 일방향 함수를 적용하여 키를 생성한다. 단말 노드는 키 그룹 멤버의 비밀키를 갖고, 중간 노드는 키 암호화 키를 갖고, 루트 노드는 그룹키를 갖는다. 키 트리에서 각 노드 x는 노드키(node key) K_x와 블라인드 노드키(blinded node key) K_x'를 갖는다. K_x'는 $K_x'=g(K_x)$에 의하여 생성되며 이때 $g()$는 일방향 함수이다. 중간 노드 x의 노드키 K_x는 $K_x=f(K_{left}', K_{right}')$에 의하여 생성되며 여기서 K_{left}'는 x의 왼쪽 자식 노드의 블라인드 노드키를 의미하고 K_{right}'는 x의 오른쪽 자식 노드의 블라인드 노드키를 의미하고 $f()$는 혼합함수로서 bit-XOR 연산을 사용한다. 그룹 멤버는 자신에게 해당하는 단말 노드의 비밀키와 루트 노드까지의 경로상의 형제 노드들의 블라인드 노드키들을 관리하고 함수 g와 f를 사용하여 경로상의 노드키와 루트키인 그룹키를 계산할 수 있다. 그림 1에서 M으로 표기된 단말 노드는 회색 노드들의 블라인드 노드키를 알고 있고 자신의 비밀키와 회색 노드들의 블라인드 노드키들과 함수 g, f를 이용하여 경로상의 노드들의 키들을 계산하고 최종적으로 루트노드의 그룹키를 계산할 수 있다. 키 관리 서버는 전체 트리를 관리하며 그룹 멤버가 새로이 추가되거나 삭제되어 노드 x의 블라인드 노드 키 Kx'가 바뀌면 새로운 Kx'를 노드 x의 형제 노드의 노드키 K_s로 암호화하여 브로드 캐스팅을 수행한다. x의 블라인드 노드 키 Kx'가 바뀌게 되어 영향을 받는 노드들은 노드 x의 형제 노드의 서브그룹에 해당하는 노드들이며 이들만이 K_s를 계산할 수 있으므로 변화된 Kx'를 알 수 있

게 한다. OFT 그룹키 관리 기법은 악의적인 공격자가 이전 세션의 그룹키에 대한 정보를 알고 있더라도 이후의 그룹키를 계산하지 못하게 하는 forward secrecy 및 공격자가 그룹키에 대한 정보를 가지고 있어도 이전 세션의 그룹키를 계산하지 못하도록 하는 backward secrecy의 조건을 만족시킨다.

Ⅲ. 보안 요구사항 및 시스템 구성

3.1 보안 요구사항

홈 네트워크 환경에서 이동 에이전트 인증을 위한 보안 요구사항은 이동 에이전트 보안의 요구사항과 홈 네트워크 환경에 특화되는 요구사항으로 구성된다. 이동 에이전트 보안의 요구사항은 비밀성, 무결성, 가용성, 책임성의 네 가지 사항으로 구성된다[16]. 비밀성을 만족시키기 위하여 이동 에이전트는 계좌 비밀번호와 같은 비밀 정보를 암호화하지 않고 이동해서는 안 된다. 이와 함께 이동 에이전트는 암호화를 위하여 사용되는 비밀키를 가지고 이동해서는 안된다[8]. 무결성을 만족시키기 위하여 에이전트의 실행 코드 및 상태 정보가 이동 에이전트의 이동 중에 정당하지 못한 방법으로 수정되지 않았음을 보장해야 한다. 가용성을 만족시키기 위하여 이동 에이전트가 이주하면서 발생할 수 있는 오류들에 대한 결함 허용 기능들이 사용된다. 책임성을 만족시키기 위하여 이동 에이전트 인증 및 권한 부여와 함께 에이전트의 동작에 대한 감사 및 로그 기능들이 사용된다. 홈 네트워크 환경에 특화되는 보안 요구사항으로는 홈 네트워크 디바이스의 계산 능력과 저장 공간의 제약을 고려하여 계산량 및 정보 전송량을 최소화시켜야 한다는 것과 홈 네트워크에 거주하는 구성원이나 방문자의 특성을 고려한 접근 제어 정책이 수립되어야 한다는 것이 있다[17].

본 논문에서 제안하는 인증의 구조는 외부로부터 홈 네트워크로 이주해 오는 에이전트가 공개키 인증을 받아서 확인된 신원에 근거하여 인증 티켓(Authentication Ticket)을 발급 받고 홈 네트워크 안의 디바이스들 간을 이주하는 경우에는 인증 티켓을 통하여 인증 받는 형태를 갖는다. 홈 네트워크 디바이스들은 특정 역할(role)을 가진 이동 에이전트의 이주를 허용하는 디바이스들의 그룹들로 구성되고 이동 에이전트는 공개키 인증을 통하여 특정 역할의 멤버임을 인증 받고, 해당 그룹의 인증 티켓을 발급 받음으로써 그룹에 속한 디바이스들 사이를 이주할 수 있는 권한을 부여 받게 된다. 예를 들어 설명하면, 가정 내의 전력 사용 및 수도 사용량 등에 대한 검침을 하기 위해 홈으로 이주하여 온 에이전트는 공개키 인증을 수행 받고 계량기 검침 그룹에 해당하는 인증 티켓을 발급 받아 댁내의 여러 계량기 디바이스로 이동할 수 있는 권한을 부여 받는다. 홈 네트워크의 관리자는 홈 네트워크의 여러 디바이스들을 그룹으로 구성하고 그룹 마다 각각의 인증 티켓을 사용하게 함으로써 검침 그룹의 인증 티켓을 발급 받은 이동 에이전트가 개인의 사생활 정보나 비밀 정보를 담고 있는 가족 그룹에 속한 디바이스들로는 이주를 하지 못하도록 한다.

3.2 시스템 구성

본 논문에서 제안하는 인증 시스템에서, 한 가정의 홈 네트워크는 단일 홈 도메인이라고 불리며 이동 에이전트, 플랫폼, 플랫폼 그룹 그리고 DMS(Domain Management Server)로 구성되고 각각에게는 *AID*(Agent ID), *PID*(Platform ID), *GID*(Group ID), *DID*(Domain ID)가 부여된다. 그림 2에서 제안 인증 시스템의 구성도를 보인다.

이동 에이전트는 실행 중에 자신의 컨텍스트를 이주시킬 수 있는 능력과 지능성을 가진 소프트웨어이며 플랫폼은 이동 에이전트의 생성, 실행, 이주 및 자원 접근 환경을 제공하는 소프트웨어로서 홈 네트워크 디바이스 상에서 작동된다. 홈 도메인 내의 플랫폼들은 DMS에 의하여 관리되며 DMS는 단일 도메인 내의 플랫폼들에 대한 정보를 관리하는데 사용되는 리스트인 DPL(Domain Platform List)을 유지한다. DPL의 각 엔트리는 표 1과 같은 필드들로 구성된다.

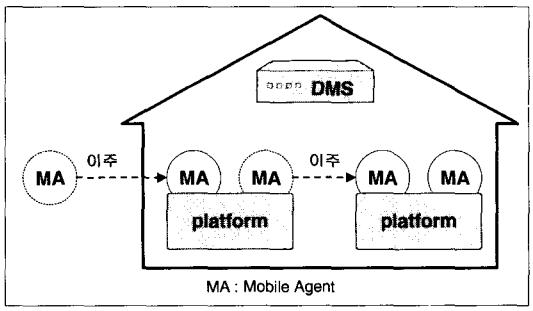


그림 2. 인증 시스템 구성도

표 1. DPL(Domain Platform List) 엔트리의 필드

필드명	설 명
PID	전역적으로 유일한 플랫폼의 ID
key	DMS와 플랫폼간의 공유키
groups	플랫폼이 가입되어 있는 플랫폼 그룹들의 ID

홈 도메인 내의 플랫폼들은 특정 역할을 갖는 이동 에이전트에게만 이주를 허용하는 플랫폼 그룹으로 구성되어 이동 에이전트의 이주 범위를 제한한다. 홈 네트워크에서의 역할의 구성은 본 논문의 범위를 벗어나는 주제이므로 본 논문에서는 언급하지 않는다. 그룹에 속한 플랫폼들은 그룹 내에서 공유되는 그룹키를 이용하여 이동 에이전트에게 인증 티켓(Authentication Ticket)을 발급한다. 인증 티켓에 관해서는 본 절의 끝에서 다시 설명한다. 본 논문에서는 그룹키의 생성 및 관리가 2.3절에서 설명한 OFT를 통하여 수행된다고 가정한다.

플랫폼 그룹의 관리는 각 홈 도메인별로 있는 DMS를 통하여 분산 관리되며 서로 다른 도메인과 그룹을 공유하지 않는다. DMS는 도메인 내의 플랫폼 그룹들에 대한 정보를 유지하는 DGL(Domain Group List)을 유지한다. DGL의 각 엔트리는 표 2와 같은 필드로 구성되며 DMS 관리자에 의하여 편집되어 그룹이 추가되거나 삭제될 수 있고, DGL의 각 엔트리에 인증 티켓을 발급 받을 수 있는 에이전트의 홈 플랫폼이 추가되거나 삭제될 수 있다.

각각의 이동 에이전트는 {*AID*, *HPID*, *AGID*, *GA*, *LAT*}로 구성되어 있는 신임장(credential)을 갖는다. *HPID*(Home *PID*)는 해당 에이전트를 생성시킨 홈 플랫폼의 *PID*, *AGID*(Agent *GID*)는 에이전트가 현재 참가하고 있는 플랫폼 그룹의 *GID*이다. *GA*(Global Authenticator)는 에이전트가 홈 도메인 외부로부터 이주하여 올 때 에이전트를 인증하는데 사용되며 다음과 같은 방법으로 생성된다.

표 2. DGL(Domain Group List) 엔트리의 필드

필드명	설 명
GID	전역적으로 유일한 플랫폼 그룹의 ID
key	그룹에 가입되어 있는 플랫폼들 간에 공유되는 키
platforms	인증 티켓을 발급 받을 수 있는 에이전트의 홈 플랫폼의 PID

$$GA = E_{K_{RS}}(AID||HPID||D)$$

K_{RS}는 에이전트의 홈 플랫폼의 비밀키이다. D는 에이전트의 실행 코드 부분에 대한 메시지 다이제스트로서 에이전트 프로그램의 무결성을 보장하기 위한 용도로 사용된다. 이동 에이전트는 신뢰성이 불확실한 여러 플랫폼들 간을 이주하기 때문에 비밀키와 같은 비밀 정보를 가지고 이동할 수 없다. 따라서 challenge-response-protocol과 같은 replay 공격 방지 기법[18]을 사용할 수 없다. 대신에 D를 K_{RS}로 서명함으로써 에이전트의 행동에 대한 무결성을 보장하여 공격자가 GA를 유용하고 에이전트의 실행 코드를 수정하여 악의적인 행동을 하지 못하도록 할 수 있다.

LAT(Loacal Authentication Ticket)는 플랫폼 그룹 내에서 이주하는 에이전트를 인증하는데 사용되는 인증 티켓이다. 에이전트가 *LAT*를 보유하고 있다는 것은 에이전트가 플랫폼 그룹 내에서 이주할 권한을 갖고 있음을 의미한다. 에이전트가 생성되거나 플랫폼 그룹에 참가할 때 *LAT*는 다음과 같은 방법으로 생성되어 에이전트에게 발급된다.

$$LAT = E_{K_G}(AID||AGID||D)$$

K_G는 에이전트가 현재 참가하고 있는 플랫폼 그룹의 공유키이다. *LAT*를 K_G로 복호화함으로써 플랫폼들은 플랫폼 그룹의 *LAT*가 에이전트에게 발급되었다는 사실을 인증한다. D의 역할은 GA의 D의 역할과 동일하다.

다음 장에서는 이러한 시스템 구성에 기반하여 이루어지는 인증 과정을 설명한다.

Ⅳ. 제안 인증 기법

4.1 다중 도메인 환경에서의 플랫폼 인증

플랫폼의 결정에 의하여 이동 에이전트가 외부 도메인으로부터 이주해 오면 에이전트를 전송하는 플랫폼에 대한 인증이 수행된다. 그림 3에서 다중 도메인 환경에서의 플랫폼 인증 과정을 설명한다.

플랫폼 A는 이동 에이전트를 전송하고자 하는 플랫폼이고 플랫폼 B는 에이전트가 전송될 목적지 플랫폼이며 두 플랫폼은 서로 다른 도메인에 위치해 있

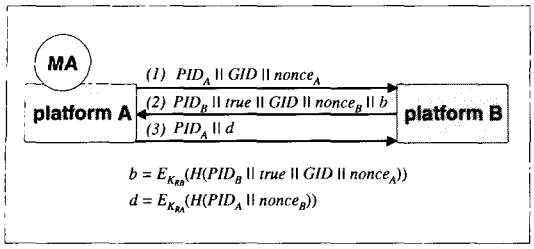


그림 3. 다중 도메인 환경에서의 플랫폼 인증

다. 이 경우의 인증은 chanllenge-response 프로토콜을 이용한다. $nonce_A$와 $nonce_B$는 각각 플랫폼 A와 플랫폼 B가 생성하는 난수와 생성 시간 값이 더해진 값으로서 인증 정보들이 엿듣기 공격에 의해 중간에 가로채어져서 유용되는 것을 방지하는 용도로 사용된다. 플랫폼 A는 그림 3의 (1)과 같이 자신의 PID인 PID_A와 인증 티켓을 발부 받기 원하는 플랫폼 그룹의 GID를 플랫폼 B에게 전송한다. 플랫폼 B는 자신이 보유하고 있는 DGL에서 GID를 갖는 엔트리를 찾고 *platforms* 필드에 PID_A가 존재하는지 확인한다. 만약 DGL에 GID에 해당하는 엔트리가 존재하지 않거나 PID_A를 *platforms* 필드에서 찾을 수 없다면 플랫폼 B는 해당 플랫폼 그룹에 대한 인증 티켓의 발부가 불가능하다는 것을 *false*를 전송하여 플랫폼 A에게 알린다. 그렇지 않다면 (2)에서 보이는 바와 같이 플랫폼 B의 개인키를 이용하여 서명을 수행하고 플랫폼 A에게 전송한다. PID_B는 플랫폼 B의 PID이며 *true*는 GID 플랫폼 그룹에 대한 인증 티켓 발급이 가능하다는 것을 의미한다. 플랫폼 A는 B의 공개키를 이용하여 서명을 확인하고 (3)에서 보이는 바와 같이 자신의 비밀키로 서명을 수행하여 B에게 전달한다. 플랫폼 B는 A의 공개키를 이용하여 서명을 확인하고 (1)에서 전송된 PID_A와 같은 PID_A가 전송되었는지 검사하여 플랫폼 A의 신원을 확인하면 플랫폼 B는 플랫폼 A로부터 에이전트 이주를 받아들이고 GID에 해당하는 플랫폼 그룹의 공유키를 이용하여 인증 티켓을 발급하여 이주된 에이전트의 LAT로 설정한다.

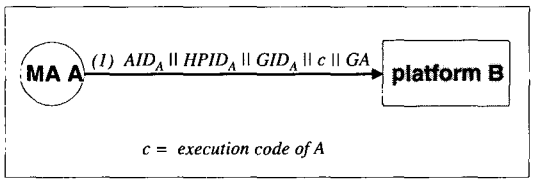


그림 4. 다중 도메인 환경에서의 에이전트 인증

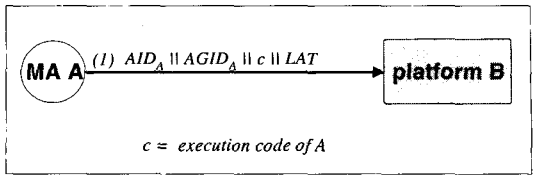


그림 5. 단일 도메인 환경에서의 그룹 인증

4.2 다중 도메인 환경에서의 에이전트 인증

이주될 이동 에이전트가 스스로의 결정에 의하여 외부 도메인으로부터 이주해 오면 이동 에이전트에 대한 인증이 수행된다. 그림 4에서 다중 도메인 환경에서의 에이전트 인증 과정을 설명한다.

에이전트 A는 다른 도메인에 위치하고 있는 플랫폼 B로 이주 결정을 내린 이동 에이전트이다. 에이전트 A는 플랫폼 B로 이주하기 위하여 그림 4의 (1)과 같은 정보를 B에게 전송한다. AID_A는 에이전트 A의 ID, $HPID_A$는 A를 생성한 홈 플랫폼의 PID 그리고 GID_A는 목적지 도메인에서 A가 참가할 플랫폼 그룹의 ID, c는 에이전트 A의 실행코드로서 GA에 이루어져 있는 서명을 확인하는데 쓰이는 메시지 다이제스트 D를 생성하는데 사용되며 또한 에이전트 A의 이주가 완료된 후 A의 실행 코드로 사용된다. 메시지 다이제스트 D에 대해서는 3.2절에서 설명하였다. 플랫폼 B는 $HPID_A$에 해당하는 플랫폼의 공개키를 이용하여 GA에 이루어져 있는 서명을 확인함으로써 에이전트를 생성한 플랫폼의 신원을 확인하고 GID_A에 해당하는 플랫폼 그룹의 공유키를 이용하여 인증 티켓을 발급하여 이주된 에이전트 A의 LAT로 설정한다.

4.3 단일 도메인 환경에서의 그룹 인증

단일 홈 도메인에서 이동 에이전트가 현재의 플랫폼 그룹의 범위 내에서 이주하고자 하면 이동 에이전트에 대하여 그룹 인증이 수행된다. 그림 5에서 단일 도메인 환경에서의 그룹 인증 과정을 설명한다.

에이전트 A는 같은 플랫폼 그룹에 소속되어 있는 플랫폼 B로 이주될 에이전트이다. 에이전트 A는 그림 5의 (1)과 같은 정보를 플랫폼 B에게 전송한다. AID_A는 에이전트의 ID, $AGID_A$는 현재 A가 참가하고 있는 플랫폼 그룹의 ID, c는 에이전트 A의 실행코드, LAT는 에이전트 A가 플랫폼 그룹에 참가하면서 발급 받은 플랫폼 그룹의 인증 티켓이다. 이

동 에이전트는 여러 플랫폼들 간을 이주하는 특성을 지니고 있기 때문에 비밀키를 지니고 이주할 수 없다. 대신에 3.2절에서 설명한 바와 같이 에이전트의 실행 코드에 대한 메시지 다이제스트에 그룹키로 암호화를 수행하여 에이전트에게 고유한 LAT를 발급하고 이를 이용하여 인증을 수행한다. 플랫폼 B는 $AGID_A$에 해당하는 플랫폼 그룹의 공유키로 LAT를 복호화 하여 AID_A // $AGID_A$ // $H(c)$와 비교함으로써 에이전트 A가 보유하고 있던 LAT가 플랫폼 그룹의 인증 티켓이 맞는지 확인한다. 만약 플랫폼 그룹의 인증 티켓이 맞으면 에이전트 A의 이주를 허용하고 그렇지 않으면 이주를 거부한다.

V. 제안 인증 기법 검증

본 장에서는 4장에서 설명한 제안 인증 기법을 검증한다. 표 3에서 본 제안 기법 검증에서 사용되는 함수를 보인다.

5.1 다중 도메인 환경에서 플랫폼 인증의 검증

플랫폼 A는 다른 도메인에 있는 플랫폼 B로 에이전트를 전송하려고 하고 플랫폼 B는 플랫폼 A에 대하여 인증을 수행할 때, 다중 도메인 환경에서 플랫폼 인증의 절차는 그림 6에서 보이는 바와 같다.

플랫폼 B의 개인키 K_{RB}를 알지 못하면 그림 6의 (3) 단계에서 b를 만들어 낼 수 없다. K_{RB}를 알고 있는 것은 플랫폼 B 뿐이므로 (5) 단계에서 플랫폼 B의 공개키로 서명을 확인함으로써 전송된 메시지가 플랫폼 B가 전송한 것이라는 사실을 확인할 수 있다. 마찬가지로 A의 개인키 K_{RA}를 알지 못하면 (9)

표 3. 기법 검증에서 사용되는 함수

기 호	설 명
find(GID, PID)	GID를 이용하여 DGL에서 플랫폼 그룹의 엔트리를 검색하고 검색된 엔트리의 platforms 필드에 PID가 존재하는지 검사하여, 존재하는 경우 true를, 그렇지 않은 경우 flase를 반환함
verify(a, b)	b를 이용하여 a의 서명을 확인함. 만약 서명에 대한 확인이 실패하면 에이전트 이주가 거부되고 그렇지 않은 경우 다음 단계로 넘어감
H(s)	s에 대한 메시지 다이제스트를 생성하여 반환함

(1) $A \rightarrow B$: $PID_A \| GID \| nonce_A$
(2) B : $a = find(GID, PID_A)$
(3) : $b = E_{K_{RB}}(H(PID_B \| a \| GID \| nonce_A))$
(4) $B \rightarrow A$: $PID_B \| a \| GID \| nonce_B \| b$
(5) A : $verify(b, H(PID_B \| a \| GID \| nonce_A))$
(7) : $if(a = false)$
(8) : *finish agent migration*
(9) : $d = E_{K_{RA}}(H(PID_A' \| nonce_B))$
(10) $A \rightarrow B$: $PID_A' \| d$
(11) B : $verify(d, PID_A' \| nonce_B)$
(12) : $if(PID_A = PID_A')$
(13) : *perform agent migration, get LAT*
(14) : *else*
(16) : *finish agent migration*

그림 6. 다중 도메인 환경에서 플랫폼 인증의 절차

에서 d를 만들어 낼 수 없다. K_{RA}를 알고 있는 것은 플랫폼 A 뿐이므로 (11) 단계에서 플랫폼 A의 공개키로 서명을 확인함으로써 전송된 메시지가 플랫폼 A가 전송한 것이라는 사실을 확인할 수 있다. 악의를 가진 공격자 C가 (1) 단계와 (10) 단계에서 전송되는 메시지를 엿들어서 replay 공격을 시도할 수 있다. C가 (1) 단계에서 엿들은 메시지를 B에게 보내면 B는 $nonce_A$에 서명한 뒤 C에게 다시 돌려준다. C가 마치 자신이 A인 것처럼 속이기 위해서 (9) 단계에서 d를 생성하여 B에게 전송해야 하지만 C는 K_{RA}을 알지 못하므로 d를 생성할 수 없다. 또한 (10) 단계에서 엿들었던 메시지를 플랫폼 B에게 다시 전송하여 자신이 A인 것처럼 속이려고 할 수 있지만 그러한 시도는 B가 전송하는 $nonce_B$값이 매번 달라지기 때문에 (11) 단계에서 서명 확인이 실패하게 된다. 플랫폼 A가 (1) 단계에서 전송하는 PID_A와 (10) 단계에서 전송하는 PID_A'를 다르게 함으로써 플랫폼 B를 속여서 플랫폼 A의 신원으로 발급받을 수 없는 인증 티켓을 발급받으려고 시도할 수 있다. 그러나 플랫폼 B가 PID_A를 저장하고 있다가 (12) 단계에서 PID_A'와 비교를 수행하여 다를 경우 이주를 중단시키므로 PID_A와 PID_A'를 달리하여 플랫폼 B를 속일 수 없다. 인증이 성공적으로 이루어지지 못하면 플랫폼이 전송하는 이동 에이전트는 플랫폼 그룹에 대한 인증 티켓을 발급받지 못한다. 인증 과정이 성공적으로 끝난다고 할지라도 확인된 A의 신원, 즉 에이전트의 전송 결정을 내린 플랫폼 A의 PID가 (2) 단계의 DGL에 등록되어 있지 않다면 플랫폼 그룹에 대한 인증 티켓을 발급받지 못한다. DGL의 내용은 DMS 관리자에 의해서만 수정될 수 있으므로 플랫폼 그룹에 대한 인증 티켓을 받

(1) S : $D_A = H(\text{execution code of } A)$
(2) : $HPID_A = PID_S$
(3) : $GA = E_{K_{RS}}(AID_A \| HPID_A \| D_A)$
(4) $A \rightarrow B$: $AID_A \| HPID_A \| GID_A \|$ execution code of $A \| GA$
(5) B : $D_A' = H(\text{execution code of } A)$
(6) : $a = AID_A \| HPID_A \| GID_A \| D_A'$
(7) : $verify(GA, a)$
(8) : $b = find(GID_A, HPID_A)$
(9) : $if(b = false)$
(10) : *finish agent migration*
(11) : *else*
(12) : *perform agent migration*
(13) : : *get LAT*

그림 7. 다중 도메인 환경에서 에이전트 인증의 절차

을 수 있는 에이전트는 DMS 관리자가 신뢰하는 플랫폼에 의해 이주가 결정된 에이전트로 국한된다.

5.2 다중 도메인 환경에서 에이전트 인증의 검증

플랫폼 S에서 생성된 에이전트 A는 다른 도메인에 있는 플랫폼 B로 이주하려고 하고 플랫폼 B는 에이전트 A를 인증하고자 할 때 다중 도메인 환경에서 에이전트 인증의 절차는 그림 7에서 보이는 바와 같다.

플랫폼 S의 개인키 K_{RS}를 알지 못하면 그림 7의 (3) 단계에서 GA를 만들지 못한다. K_{RS}를 알고 있는 것은 S뿐이므로 (7) 단계에서 S의 공개키로 서명을 확인함으로써 에이전트 A가 플랫폼 S에 의하여 생성된 에이전트라는 사실을 확인할 수 있다. 악의를 갖고 있는 에이전트 C가 (4) 단계에서 전송되는 메시지를 엿들어서 replay 공격을 시도할 수 있다. C는 (4) 단계에서 엿들은 메시지에서 A의 실행 코드 부분을 수정하여 플랫폼 B에게 전송하여 이주 요청을 시도한다. 그러나 GA에는 이동 에이전트의 실행 코드에 대한 메시지 다이제스트가 에이전트 A를 생성한 플랫폼 S의 K_{RS}로 서명되어있으므로 (7) 단계에서 서명의 확인은 실패하게 된다. 따라서 (4) 단계에서 전송되는 에이전트 A의 실행 코드를 수정하여 악의적인 행동을 하려는 시도는 불가능하다. (1) 단계에서 (7) 단계까지의 인증 과정이 성공적으로 이루어지지 못하면 이동 에이전트는 플랫폼 그룹에 대한 인증 티켓을 발급받지 못한다. 인증 과정이 성공적으로 끝난다고 할지라도 확인된 A의 신원, 즉 A를 생성한 플랫폼 S의 PID가 (8) 단계의 DGL에 등록되어 있지 않다면 플랫폼 그룹에 대한 인증 티켓을 발급받지 못한다. DGL의 내용은 DMS 관리자에 의해서만 수정될 수 있으므로 플랫폼 그룹에 대한 인증 티켓을 받을 수 있는 에이전트는 DMS 관리자가 신뢰할 수 있는 플랫폼에 의해 생성된 에이전트로 국한된다.

(1) S : $D_A = H(\text{execution code of } A)$
(2) : $LAT = E_{K_G}(AID_A \| AGID_A \| D_A)$
(3) $A \rightarrow B$: $AID_A \| AGID_A \|$ execution code of $A \| LAT$
(4) B : $D_A' = H(\text{execution code of } A)$
(5) : $verify(LAT, AID_A \| AGID_A \| D_A')$
(6) : *perform agent migration*

그림 8. 단일 도메인 환경에서 그룹 인증의 절차

5.3 단일 도메인 환경에서 그룹 인증의 검증

플랫폼 S에서 인증 티켓을 할당 받은 에이전트 A가 같은 플랫폼 그룹에 속한 플랫폼 B로 이주하려고 하고 플랫폼 B는 에이전트 A가 플랫폼 그룹의 인증 티켓을 보유하고 있는지 검증하고자 할 때 단일 도메인 환경에서 그룹 인증의 절차는 그림 8과 같다.

플랫폼 그룹의 공유키 K_G는 플랫폼 그룹의 멤버에게만 알려져 있으므로 그림 8의 (3) 단계에서 멤버가 아닌 플랫폼은 플랫폼 그룹의 LAT를 생성할 수 없다. 따라서 (5) 단계에서 LAT를 플랫폼 그룹의 공유키 K_G로 복호화한 값이 AID_A // $AGID_A$ // D_A'와 같다면 LAT가 플랫폼 그룹 멤버에 의해 만들어졌다는 것이 증명된다. LAT의 발급 여부는 그림 6의 (2) 단계와 그림 7의 (8) 단계에서 DMS 관리자에 의하여 수정되는 DGL 엔트리의 *platforms* 필드에 플랫폼이 등록되어 있느냐에 따라 결정된다. 따라서 에이전트가 플랫폼 그룹의 멤버에 의하여 만들어진 LAT를 갖고 있다는 것은 에이전트가 DMS 관리자로부터 플랫폼 그룹 내의 플랫폼들 간을 이주할 수 있는 권한을 부여받았음을 의미한다. 악의를 갖고 있는 에이전트 C가 그림 8의 (3) 단계에서 전송되는 메시지를 엿들어서 replay 공격을 시도할 수 있다. C는 (3) 단계에서 엿들은 메시지에서 A의 실행 코드 부분을 수정하고 플랫폼 B에게 전송하여 이주를 요청한다. 그러나 LAT에는 이동 에이전트의 실행 코드에 대한 메시지 다이제스트가 플랫폼 그룹의 공유키 K_G로 서명되어 있으므로 (5) 단계에서 서명의 확인은 실패하게 된다. 따라서 (3) 단계에서 전송되는 정보에서 A의 실행 코드를 수정하여 악의적인 행동을 하려는 시도는 실패하게 된다.

표 4. 인증에 소요되는 연산량 비교

	공개키 인증	공유키 인증
LAT 생성	TH + TE	TH + TS
LAT 서명 확인	TH + TD	TH + TS

TE : 공개키 기반 서명에 소요되는 시간
TD : 공개키 기반 서명 확인에 소요되는 시간
TS : 공유키 암/복호화에 소요되는 시간
TH : 해싱에 소요되는 시간

단일 도메인 환경에서의 그룹 인증은 공유키 암호화 기법을 이용하여 인증을 수행하므로 공개키 암호화 기법을 이용하는 인증에 비하여 가벼운 연산만으로 인증을 수행한다. 단일 도메인 환경에서의 그룹 인증 비용은 *LAT* 생성 비용과 *LAT* 서명 확인 비용으로 이루어진다. 표 4는 단일 도메인 환경에서의 공유키를 이용한 인증을 공개키를 이용하여 수행한다고 가정하였을 경우의 연산량과 본 논문에서 제안하는 공유키 인증의 연산량을 비교한 표이다.

표 5는 대표적인 공개키 암호화 알고리즘인 RSA, 해싱 알고리즘인 MD4, 공유키 암호화 알고리즘인 DES의 처리 속도를 하드웨어 수준 및 소프트웨어 수준에서 수행했을 때의 속도를 비교한 표이다[9].

표 5에 있는 바와 같이 공유키 암호화 알고리즘은 공개키 암호화 알고리즘에 비하여 약 1000배에서 5000배까지 더 빠르다. 따라서 본 논문에서 제안한 공유키를 사용하는 그룹 인증이 공개키 암호화 기법을 이용하는 인증보다 빠르게 인증을 수행한다고 할 수 있다.

Ⅵ. 결 론

본 논문에서는 홈 네트워크에 이동 에이전트를 적용하는데 필요한 이동 에이전트 인증문제를 해결하기 위해서 홈 네트워크에서의 이동 에이전트 보안 요구사항이 무엇인지를 밝히고 그에 따른 인증 기법을 제안하였다. 제안 인증 기법은 공개키 암호화 알고리즘에 비하여 약 1000배에서 5000배까지 더 빠른 공유키알고리즘을 적용함으로써 단일 도메인 환경에서의 그룹 인증에 사용되는 *LAT*의 생성 및 서명확인 과정을 가벼운 연산만으로 수행할 수 있다. 또한 다중의 홈 네트워크들 간에 발생하는 에이전트의 이주 요청에 대해서는 플랫폼에 의한 이주 요청과 이동 에이전트 자신에 의한 이주 요청으로 나누고 각각에 대하여 공개키 인증을 적용함으로써 이동 에이전트 인증을 다중의 홈 네트워크 환경으로 확장시킬 수 있도록 하였다. 향후 본 논문에서 제시한 기법을 기반으로 홈 네트워크 환경에서 이동 에이전트의 자원 및 서비스 접근 제어 기법의 개발이 가능할 것이다.

표 5. 암호화 연산 속도

	Hardware, bits/sec	Software, bit/sec/MIPS
RSA 암호화	220K	0.5K
RSA 복호화	-	32K
MD4	-	1300K
DES 암호와 및 복호화	1.2G	400K

참 고 문 헌

[1] Bill 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] Neeran M. Karnik, Anand R. Tripathi, "Agent Server Architecture for the Mobile-Agent System," PDPTA'98, pp. 66-73, July 1998.

[4] Jeong-Joon Yoo, Dong-Ik Lee, "Scalable Home Network Interation Model Based on Mobile Agents," PerCom'03, pp. 543-546, March 2003.

[5] Kazunori Takashio, Gakuya Soeda, Hideyuki Tokuda, "A Mobile Agent Framework for Follow-Me Applications in Ubiquitous Computing Environment," Distributed Computing Systems Workshop, 2001 International Conference, pp. 202-207, April 2001.

[6] N. Karnik, "Security in Mobile Agent Systems," Ph. D. dissertation, University of Minnesota, 1998.

[7] ObjectSpace, Inc. "Voyager Security Developer's Guide Table of Contents,"

http://www.recursionsw.com/products/voyager/voyager.asp.

[8] Shimshon Berkovits, Joshua D. Guttman, Vipin Swarup, "Authentication for Mobile Agents," Mobile Agents and Security LNCS, pp. 114-136, 1998.

[9] Buttler Lampson, Martin Abadi, Michael Burrows, Edward Wobber, "Authentication in Distributed Systems : Theory and Practice," ACM Transactions on Computersystems, pp. 265-310, November 1992.

[10] G.V.S. Raju, Rehan Akbani, "Elliptic curve cryptosystem and its applications," Systems, Man and Cybernetics, 2003. IEEE International Conference Vol. 2, pp. 1540-1543, Oct. 2003.

[11] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, "NTRU : A Ring-Based public Key Cryptosystem," LNCS Vol 1423, 1998

[12] 박현미, 강상승, 최영근, 김순자, "NTRU기반의 이동 통신에서의 인증 및 키 합의 프로토콜," 정보보호학회논문지, 12(3), June 2002.

[13] 이보영, 최연이, 주미리, 원동호, "무선통신 환경에서 사용 가능한 고차잉여류 문제에 기반을 둔 자체 인증방식," 정보보호학회논문지, 9(2), June 1999.

[14] David A. McGrew, Alan T. Sherman, "Key Establishment in Large Dyanmic Groups Using One-Way Function Trees," Cryptographic Technologies Group, TIS Labs at Network Associates, Inc., May 1998.

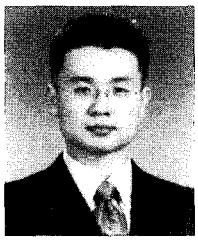
[15] 박영호, 이경현, "이동네트워크 환경에서의 그룹키 관리구조," 정보보호학회논문지, 12(2), April 2002.

[16] Wayne Jansen, Tom Karygiannis, "NIST Special Publication 800-19-Mobile Agent Security," National Institute of Standards and Technology Computer Security Division.

[17] Carl M. Ellison, "Home Network Security," Intel Technology Journal Vol. 6, November 2002.

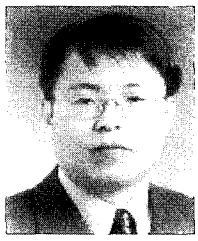
[18] Martin Adabi, Roger Needham, "Prudent Engineering Practice for Cryptographic Protocols," IEEE Transactions on Software Engineering, January 1996.

〈著者紹介〉



김 재 곤 (Jae-gon Kim) 학생회원
2004년 2월 : 성균관대학교 정보통신공학부 졸업(학사)
1994년 10월~현재 : 성균관대학교 컴퓨터공학과 석사과정
〈관심분야〉 이동 에이전트, 운영체제

김 구 수 (Gu Su Kim) 학생회원
1994년 2월 : 성균관대학교 정보공학과 졸업(학사)
1996년 2월 : 성균관대학교 정보공학과 졸업(석사)
2002년 3월~현재 : 성균관대학교 정보통신공학부 박사과정
〈관심분야〉 분산컴퓨팅, 이동 에이전트



엄 영 익 (Young Ik Eom) 정회원
1983년 2월 : 서울대학교 계산통계학과 졸업(학사)
1985년 2월 : 서울대학교 대학원 전산과학과 졸업(석사)
1991년 8월 : 서울대학교 대학원 전산과학과 졸업(박사)
2000년 9월~2001년 8월 : Dept. of Info. and Comm. Science at UCI 방문교수
현재 : 성균관대학교 정보통신공학부 교수
〈관심분야〉 분산시스템, 이동 컴퓨팅 시스템, 이동 에이전트, 시스템 소프트웨어 등