유비쿼터스 컴퓨팅 환경에서 속성 인증서를 이용한 단일/멀티 도메인 인증

(Authentication for Single/Multiple Domain using Attribute Certificates)

이 덕 규 † 박 희 운 †† 이 임 영 †††
(Deok-Gyu Lee) (Hee-Un Park) (Im-Yeong Lee)

요 약 유비쿼터스 컴퓨터 환경은 사용자가 언제 어디서나 컴퓨터를 이용할 수 있도록 네트워크를 통해 상호 연결된 보이지 않는 수많은 컴퓨터(Invisible Computer)가 편재되고 사용자가 원하는 대로 쉽게 이용할 수 있는 컴퓨팅 환경을 지향하고 더 나아가서는 사용자가 원하는 컴퓨팅을 컴퓨터 스스로 알아서 (상황인식 Context Awareness) 제공하는 스마트 환경이다. 이러한 유비쿼터스 컴퓨팅은 보안에 있어 특히 취약한 면을 많이 내포하고 있다. 그 중에서도 분산된 다양한 컴퓨팅 기기들이 도처에 존재함으로 인해 사용자 주변에 있는 기기 중에서 사용자 혹은 서버에 인증된 기기로의 위장공격 등이 가능해진다. 이에 본 논문에서는 다음과 같은 특징을 가지는 방식을 제안한다. 디바이스의 이동을 통한 인증 모델을 제시한다. 이는 개인의 작은 디바이스가 다른 사용자의 공간(MD: Multi Domain)로 이동하였을 경우 디바이스를 통한 인증을 실현하는 방식과 사용자 자신의 공간(SD: Single Domain)으로 이동하였을 경우 디바이스를 통한 인증을 실현하는 두 가지 방식을 제안한다.

키워드 : 유비쿼터스, 멀티 도메인, 단일 도메인, 인증

Abstract The Ubiquitous computer environment is thing which invisible computer that is not shown linked mutually through network so that user may use computer always is been pervasive. Intend computing environment that can use easily as user wants and it is the smart environment that user provides context awareness that is wanting computing environment. This Ubiquitous computing contains much specially weak side in security. Masquerade attack of that crawl that is quoted to user or server among device that is around user by that discrete various computing devices exist everywhere among them become possible. Hereupon, in this paper, proposed method that have following characteristic. Present authentication model through transfer or device. Suggest two method that realize authentication through device in case of moved to method(MD: Multi Domain) and user ownself space(SD: Single Domain) that realize authentication through device in case of moved user's direct path who device differs.

Key words : Ubiquitous, Multi Domain, Single Domain, Authentication

1. 서 론

유비쿼터스 컴퓨터 환경은 사용자가 언제 어디서나 컴퓨터를 이용할 수 있도록 네트워크를 통해 상호 연결된 보이지 않는 수많은 컴퓨터(Invisible Computer)가 편재되고 사용자가 원하는 대로 쉽게 이용할 수 있는 컴퓨팅 환경을 지향하고 더 나아가서는 사용자가 원하는 컴퓨팅을 컴퓨터 스스로 알아서(상황인식 Context Awareness) 제공하는 스마트 환경이다. 이러한 유비쿼터스 컴퓨팅은 다음과 같은 특징을 가진다. 첫째로 특정 목적의 특정 사용자들을 위하여 분산된 다양한 컴퓨팅 기기들이 존재하며, 둘째로 네트워크를 통해 단절없이 연결된 컴퓨팅 기기들이 존재하게 된다. 셋째로는 사용자의 눈에 띄지 않으며 사용자는 인간화된 인터페이스

· 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음
† 정 회 원 : 순천향대학교 전산학과
hbrhcdbr@sch.ac.kr
†† 비 회 원 : 한국정보보호진흥원 연구원
hupark@kisa.or.kr
††† 종신회원 : 순천향대학교 정보기술공학부 교수
imylee@sch.ac.kr
논문접수 : 2004년 2월 4일
심사완료 : 2005년 2월 14일

만을 느끼는 특징을 가지게 된다. 마지막으로 인터넷 안의 가상공간이 아닌 실제 세계의 공간이다[1-4].

이러한 유비쿼터스 컴퓨팅은 보안에 있어 특히 취약한 면을 많이 내포하고 있다. 그 중에서도 분산된 다양한 컴퓨팅 기기들이 도처에 존재함으로 인해 사용자 주변에 있는 기기 중에서 사용자 혹은 서버에 인증된 기기로의 위장공격 등이 가능해진다. 또한 사용자가 인증된 프로그램에 대해서 기기로의 설치를 인가하여도, 악의적인 목적을 가진 코드가 사용자 주변의 컴퓨팅 능력이 없는 기기로의 전송이 발생할 수 있다. 많은 유비쿼터스 컴퓨팅 기기들이 메모리에 대한 효과적인 보호방안을 갖고 있지 않기 때문에 실제적인 사용자 정보(인증 정보)가 보관된 메모리에 대한 공격이 가능해진다[5]. 이러한 것들은 컴퓨팅 능력을 지닌 디바이스에 의해 암호나 전자 서명을 통하여 보안 될 수 있지만, 컴퓨팅 능력을 지니지 않은 디바이스에 있어서는 암호나 전자 서명을 통해 안전해 질 수 없기 때문에 잠재적인 위험이 항상 내재되어있다고 할 수 있다. 또한 하나의 디바이스가 사용자를 벗어나 다른 사용자의 공간으로 이동되었을 경우 이동된 공간에서의 사용자 인증이 원활히 수행되어야 한다. 다른 사용자의 공간에서 다른 디바이스가 이전 사용자의 인증 정보로 인증을 실시하지 않고 이동한 곳의 사용자의 인증 정보를 가지고 수행될 수 있기 때문이다[6].

이에 본 논문에서는 다음과 같은 특징을 가지는 방식을 제안한다. 디바이스의 이동을 통한 인증 모델을 제시한다. 이는 개인의 작은 디바이스가 다른 사용자의 공간(MD: Multi Domain)로 이동하였을 경우 디바이스를 통하여 인증을 실현하는 방식과 자신의 공간(SD: Single Domain)으로 이동하였을 경우 디바이스를 통하여 인증을 실현하는 두 가지 방식을 제안한다. 첫 번째 방식에서 각각의 디바이스는 사용자의 다른 인증 정보를 보관하게 되는데 스마트 디바이스에는 최소의 인증 정보를 보관하고 컴퓨팅 능력을 가지고 있는 디바이스에는 최대의 인증 정보를 보관토록 한다. 스마트 디바이스의 경우 최소의 인증 정보를 보관하게 됨으로 인해 스마트 디바이스에게 최초로 인증 정보를 제공해준 Hub를 통해 인증 정보를 전달하고 전달받게 된다. Hub는 사용자 혹은 서버가 될 수 있다. 본 논문은 총 5장으로 구성되며 2장은 유비쿼터스 컴퓨팅에서 인증을 위한 요구사항, 3장은 유비쿼터스 컴퓨팅 연구 동향, 4장에서는 단일 도메인과 멀티 도메인에서 스마트 디바이스를 이용한 인증 방식을 제안한다. 5장은 기존연구와 제안방식과의 비교 분석을 통해 제안방식의 효율성을 검증하고, 6장에서 결론으로서 마치도록 한다.

2. 유비쿼터스 컴퓨팅 연구 동향과 PMI 개요

유비쿼터스의 대부로 불리는 PARC(Palo Alto Research Center)의 Mark Weiser가 1993년 발표한 논문 중에 컴퓨터의 진화과정을 컴퓨터 기술과 인간과의 관계 변화를 중심으로 보고, 제 1의 물결인 메인프레임시대(1대의 고가의 컴퓨터를 다수가 공유), 제 2의 물결인 퍼스널 컴퓨터 시대(1인 1대의 컴퓨터 사용), 그리고 광역 분산 컴퓨팅을 제공하는 인터넷 시대를 거쳐 유비쿼터스 사회(다양한 사람들이 내장형의 다양한 컴퓨터를 의식하지 않고 네트워크를 통해 사용)로 나누는 동시에 컴퓨터 기술에 있어 제 3의 물결로 정의하였다. 그리고 우리 생활 중의 주요한 컴퓨터 인터페이스 기술이 보이지 않는 인터페이스(Invisible Interfaces)를 사용하여 우리 주변에 스며들어 일상생활에 통합되는 기술(Calm Technology)의 등장을 언급하면서, 이러한 기술변화를 통해 새로운 문화 즉, 유비쿼터스 컴퓨팅의 출현을 주장하였다[2,7].

유비쿼터스 컴퓨팅에서의 주요 연구는 다음과 같이 이뤄지고 있다. 유비쿼터스 컴퓨팅 시대의 유비쿼터스 네트워크는 PC, 서버 중심의 협회의 네트워킹에서 AV 기기, 정보가전, 휴대전화, 게임기, 제어기기 등과 같은 다양한 기기가 접속됨으로 인하여 소형화 기술, 휴대전화기술, 정보가전기술, 전자제어기술, 네트워킹제어기술 등이 주요한 원천 기술로 대두되었다. 이중에서 어디서나 안전하게 컴퓨터를 사용할 수 있는 기술로서 개인 인증 기술과 보안 기술을 들 수 있다. 이러한 개인 인증에 관한 연구는 각 국가의 프로젝트에 따라 연구가 진행되어왔다. 각각의 도메인 상에서는 인증에 관한 해결책으로 제시하고 있는 연구는 현재 진행되고 있지 않은 실정이다. 다음 장에서는 기존의 유비쿼터스 컴퓨팅의 일반적인 흐름을 알아보고 기존방식에 대해 설명한다.

2.1 유비쿼터스 컴퓨팅 흐름

유비쿼터스 컴퓨팅 관련 프로젝트는 MIT의 Oxygen[8], UC Berkeley의 Endwavour, Washington 대학의 Portolano[9], Georgia Tech & Ogi의 Infosphere, CMU의 Aura 프로젝트[10], 일본의 TRON프로젝트[11] 등이 있다. 산업체에서의 가장 대표적인 경우가 IBM의 Websphere 제품[12]과 HP의 Cooltown[13], MS의 Easy Living[14]이라 할 수 있다[3,15,16].

MIT의 Oxygen Project의 특징은 매우 동적이며, 다양한 인간 활동을 지원하기 위함이고, 많은 기술적인 문제점을 해결해야한다. 사용자와 시스템 기술들을 조합하여 편재하며, 인간 중심형 컴퓨팅 기술을 가능하게 하며, 음성, 비전 기술들을 이용하여 시간과 노력을 절약시킨다. 임베디드 디바이스를 이용한 분산처리 기반 컴

퓨팅 시스템이라 할수 있다.

Washington 대학의 Portolano 프로젝트는 사용자의 의도에 따른 다중 사용자 인터페이스 기능을 가지며, 네트워크에 기초한 수평적 계층적 서비스 기능을 제공한다. 그리고 엑티브 네트워크, 분산처리 기반 구조 기능을 특징으로 한다.

CMU의 Aura 프로젝트는 사용자의 집중도를 떨어뜨리지 않고 작업할 수 있는 컴퓨팅 환경구성을 주요 목표로 하고 있다. Aura 프로젝트를 수행하기위한 요소기술은 다음과 같으며, 보안에 관련된 연구는 키 설립, 선택적 제어, 위치 정보에 대한 보호등 개인적인 프라이버시에 집중되는 것을 알 수 있다.

IBM의 유비쿼터스 컴퓨팅은 모든 네트워크 상에서 임의의 장치를 사용하여 어떤 정보라도 전달하며, 개인화기능을 이용하면 사용자가 선택하는 것으로 정보를 전달하는 일을 뜻한다.

2.2 JARM 방식

2002년에 Jalal등은 사용자 인증 레벨 개념을 가진 방식을 제안하였다[1]. 여기에서 각각의 디바이스에 사용자 인증 정보가 다르게 존재할 수 있다. 즉, 스마트 디바이스인 PDA, 시계, 스마트 반지와 같은 디바이스에 사용자의 정보를 반지에는 최소의 인증 정보를 PDA에는 중간 정도의 인증 정보를 보관 할 수 있다. 그러나 이 방식에서 하나의 디바이스가 사용자의 도메인에서 벗어나 다른 도메인으로 들어갔을 경우 다른 도메인의 사용자 정보를 가지고 있더라 하더라도 새로운 도메인의 사용자 인증 정보를 따라가 결국 다른 도메인에서 이동하여 온 사용자는 사용에 있어 제약이 생기게 된다. 이 방식은 같은 도메인 내에서의 사용자의 디바이스 각각이 인증 정보를 가지고 있으므로 사용자가 하나의 디바이스를 통해서 인증을 받을 수 있고 사용자 주변의 모든 디바이스를 레벨 인증 정보를 통하여 모두 인증받을 수 있는 특징을 가지고 있다. 이 방식에서는 레벨 인증 정보를 위해 신임값을 통한 인증을 다단계로 분할하고 있는데 신임을 각 디바이스에서 획득하기 위해 각 디바이스에 맞는 인증 프로토콜을 통하여 신임값을 전달하고 있다. 하지만 이러한 신임값을 갖고 디바이스를 인증하는 것은 스마트 디바이스에 대해서는 효율적인 인증을 제공하지만 전체적인 인증을 위해서나 스마트 인증에 대한 확인을 위해 상위 디바이스가 필요한 경우가 발생하게 된다. 이때 신임값을 갖는 중간 디바이스나 스마트 디바이스 상위의 디바이스가 다른 곳에 위치하거나 분실하였다면 전체적인 인증은 불가능하며 분실 디바이스 이하의 디바이스에 대해서는 새로이 신임값을 분배해야하는 문제점이 발생하게 된다. 각각에 대하여 다시 살펴보면 다음과 같다.

1. 전체 인증 정보는 디바이스 N까지의 신임값 합과 일치한다.
2. 어떤 한 디바이스가 이동 혹은 분실되면, 이하 디바이스에 대해 전체적인 인증은 불가능하다

JARM 방식에 대해 자세히 기술하면 다음과 같다.

유비쿼터스 컴퓨팅 환경에서 사용자는 여러 가지 디바이스를 통하여 인증을 받을 수 있다. 하나의 디바이스를 통해 인증이 이뤄질 수 있고 작은 디바이스는 상위의 디바이스로 인증 정보를 전송하는 다단계 과정을 통해 인증을 이뤄질 수 있다. 다단계 인증 과정 중에 각 디바이스에 어떻게 신임할 것인가가 가장 큰 문제이다. 예를 들어, 패스워드를 어느 하나의 인증 디바이스에게 제공한 후 이것을 디바이스가 사용하였을 때, 제공된 패스워드가 다른 디바이스에서 어느 정도 신뢰되어 개체를 인증하는 것은 각 디바이스 선택에 의해 좌우된다.

각 디바이스에 신임값을 전달하는 것은 각 디바이스에 맞는 프로토콜을 통하여 이뤄질수 있다. 만약 사용자가 하나의 인증 방법을 사용하기 원할 때 신임값은 포괄적으로 사용할 수 있다. 이 방식에서 사용되는 신임값의 예제는 다음과 같다.

$$C_{net} = 1-(1-C_1)(1-C_2)\cdots(1-C_n)$$

여기서 C_{net}은 사용자의 신임값이 되며, $C_1, C_2, \ldots, C_n$은 각 디바이스의 신임 값이 된다. 본 방식은 기존 분산 시스템의 인증 방법으로 사용되었던 Kerberos를 사용한다. 하지만 유비쿼터스 환경에 맞도록 개선하여 사용하게 된다. 여기서 활동 공간(AD: Active Domain)은 인증을 위한 도메인으로 Kerberos와 동일하게 구성한다. 이 엑티브 도메인은 세 개의 인증 컴포넌트들로 구성하게 된다. 첫 번째 컴포넌트는 인증 서버(AS: Authentication Server)로서 엑티브 도메인 내의 SSO를 지원하게 된다. 두 번째 컴포넌트는 티켓 발급 서버(TGS: Ticket Granting Server)로 엑티브 도메인 내에 사용자가 접근할 수 있는 티켓 발급을 담당한다. 세 번째 컴포넌트는 데이타베이스이며, 엑티브 도메인내의 모든 사용 인증에 관한 필요정보를 보관한다.

Step 1. 활동 공간(Active Domain)내에 스마트 디바이스들이 login등을 검출하는 것을 기지국으로 봉쇄할 수 있다.

Step 2. 사용자는 자신을 충분히 인증해 줄 수 있는 SAP(Space Authentication Portal)로 이동한다.

Step 3. SAPs 자체는 사용자 인증을 제공하면서 프라이버시를 제공하기 위해 충분한 정보를 제공하지 않는다. 하지만, 사용자의 정보는 보안 서버와 통신할 수 있는 Lighthouse에서 가지고 있다.

Step 4. 성공적인 인증에는 Kerberos와 같은 활동

공간에 사용할 수 있는 사용자에 대한 TGT(Ticket Granting Ticket)가 있어야 한다.

Step 5. 사용자는 "고정된" 워크스테이션의 사용이 필요치 않다. 대신 활동 공간에서 서비스를 이용하도록 접근이 가능한데, 직접적으로 사용할 수 있는 어떠한 디바이스를 이용함으로써 서비스에 상호 작용할 수 있다.

Step 6. 통신은 Lighthouse가 사용자의 위치정보 제공을 막기 위해 Mist 프로토콜(MIT대학의 프로토콜)을 사용하면서 일어난다.

Step 7. TGT는 Target 사용자를 위해 Lighthouse에서 저장해둔다. 또한 Lighthouse는 필수 서비스에 접근하기 위해 티켓들을 요청한 TGS와 통신할 수 있다.

Step 8. 최종적인 신뢰와 더불어 사용권한에 대한 내용을 담고 있는 티켓내의 정보를 이용하여, 서비스는 스마트 디바이스 소유자에게 권한을 줄 것인지 아닌지 결정을 할 수 있다.

Step 9. 최종적으로 활동공간의 로그오프 혹은 티켓의 폐기, 티겟의 말소, 디바이스 활동 공간 벗어난 경우는 종료된다.

2.3 PMI(Privilege Management Infrastructure) 개요

기존의 PKC(Public Key Certificate)는 정보보호 서비스를 사용하는 사용자의 신원을 인증하는 기능을 한다. 하지만 다양한 웹상의 서비스는 사용자 권한에 따라 서로 다른 기능을 제공하려는 움직임을 보이고 있다. 이런 상황에서 사용자 신원의 확인은 물론 사용자의 속성 즉, 권한, 지위, 임무 등에 관한 정보를 기록할 필요가 발생했다[17-19].

사용자 속성의 정보를 제공하려는 방법의 하나로 기존의 PKI기반에서 사용하던 X.509 인증서의 확장 필드

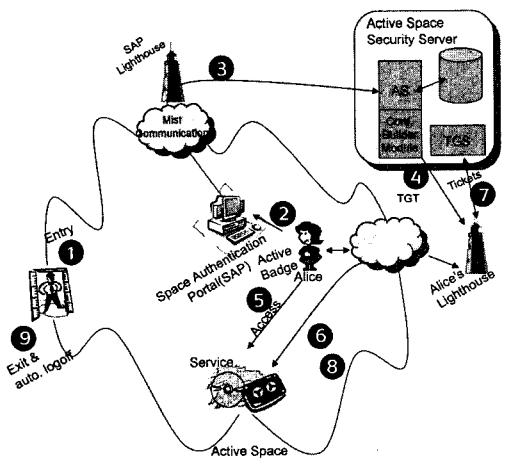

그림 1 JARM 방식의 인증 흐름도

를 이용하는 방안이 제안되어 있다. 하지만 이를 사용할 경우에 또 다른 문제가 발생한다. 일반적으로 각 개체에게 주어지는 권한에는 유효기간 존재한다. 하지만 사용자 신원에 비해 사용자에게 부여되는 권한은 더 자주 변하므로 인증서에 비해 사용자 속성의 유효기간이 더 짧다. 따라서 새로운 속성을 적용하기 위해서 이미 발급된 인증서를 폐기하고 새로운 인증서를 재발급 받아야 한다. 또, 사용자 신원은 전체에게 신뢰 받는 하나의 대리기관에서 받아서 모두 적용할 수 있지만, 사용자의 속성은 적용하려는 곳 마다 다르기 때문에 기존의 인증서를 사용할 경우 적절한 인증서를 항상 재발급 받아야 하는 단점이 생긴다. 이러한 문제들을 해결하고자 AC(Attribute Certificate)를 사용한다. 이는 사용자의 속성을 기록하고 인증하는 또 다른 인증서에 해당한다. PMI의 각각의 구성요소와 사용모델에 대해 설명한다.

1) 구성요소들

일반적인 권한 관리 모델은 객체, 권한 소유자, 권한 입증자 등으로 구성된다. 객체는 접근 제어 응용과 같이 보호되어야 할 자원을 의미하는데, 이러한 객체들은 각자의 메소드를 지닌다. 예를 들어 방화벽 같은 객체는 '개체 허용'과 같은 메소드를, 파일 시스템 상의 파일은 읽기, 쓰기, 실행 등의 메소드를 지닌다.

어떤 사용자가 지닌 권한은 그 권한을 소유한 사람의 신임 정도를 반영한다. 각 개체에게 부여되는 권한은 AC(s)에 캡슐화 되어 있거나 PKC의 확장영역에 하나의 필드로 기록된다.

권한정책은 객체가 지닌 메소드의 보안 민감도나 사용 환경을 고려하여 각 개체에게 적합한 권한의 정도를 부여하는 방법을 제시한다. 권한 정책은 무결성과 인증 서비스가 제공되어야 한다. 전달 정책을 세우는데 있어서 여러 경우가 존재하는데, 권한이 실제로는 전달되지 않게 하고 입증자의 환경에 맞게 부분적으로 사용하게 할 수도 있고, 시스템 내의 모든 개체에게 알려지고 전달되게 할 수도 있다.

권한 정책은 각 서비스에 대한 권한들의 수용을 위한 경계를 제시한다. 즉 소유자가 객체에 접근하기 위해 적합한지를 판단해야 할 때 입증자는 그 정책을 사용하여 결정한다.

환경 변수들은 권한 입증자가 결정을 내릴 때 지역적으로 그 환경에 맞게 설정할 수도 있는데, 이를 나타내기 위해서 사용한다.

객체 메소드의 보안에 대한 민감도는 전달되는 문서나 처리해야 할 요구들의 속성을 의미한다. 예를 들어 문서 내용이 어느 정도의 보안 사항인지를 나타내는 것을 의미한다. 이는 AC나 연관된 보안 레이블 등에 기록될 수 있다. 객체가 사용되는 상황에 따라 메소드의 민

감도는 사용되지 않을 수도 있다.

2) AA와 SOA

AA(Attribute Authority)와 CA(Certification Authority)는 논리적인 경우와 대부분의 물리적인 경우에서 서로 완전히 독립적이다. 신원(Identity)을 만들고 유지하는 것은 PMI와 구분되어 PKI를 기반으로 이루어진다. 그러므로 전체 PKI가 구축되고 나서 PMI를 구축하게 된다.

SOA(Source of Authority)는 일련의 권한 할당에 책임을 지는 권한 주장자에 의해서 신뢰되는 개체이다. SOA는 자신이 AA가 되어서 다른 개체에게 인증서를 발급하기도 하며, PKI 기반에서의 root CA와 같은 역할을 하므로, SOA로부터 서명된 인증서는 권한 입증자에게 신뢰를 준다.

3) PMI 사용 모델

AC를 이용하여 접근 제어 응용에 사용할 경우 처리해야 하는 여러 상황이 발생한다. 이를 어떻게 처리하는지를 보여준다. PMI에는 크게 관제모델, 권한위임모델, 역할 기반 모델로 나누어진다.

3. 유비쿼터스 컴퓨팅 요구사항

'유비쿼터스(Ubiquitous)', '퍼패이시브(Pervasive)', 또는 'Invisible computing' 등으로 불리는 유비쿼터스 컴퓨팅은 사용자가 컴퓨터를 이용하고 있음을 자각하지 못하고 일에 집중하게 할 수 있도록 인간중심의 컴퓨팅 기술로 등장하였다. 이러한 유비쿼터스 컴퓨팅은 디지털 데이타가 가지는 모든 특성을 가지고 있기 때문에 그 편리함과 유용함에도 불구하고 많은 문제점들을 내포하고 있으며 이러한 문제점을 완벽히 해결하지 않고서는 사용될 수가 없다. 즉, 사용자 주변에 여러 대의 디바이스를 이용하기 때문에 사용자의 정보를 대량으로 복사하여 확인 혹은 인증되지 않은 디바이스에 사용자의 정보를 삽입할 수 있으며, 수집된 정보를 네트워크 상으로 위·변조를 통하여 악의적인 목적으로 이용될 수 있다. 이러한 유비쿼터스 컴퓨팅이 가지고 있는 자체 특성과 사용 환경으로 인해 여러 가지 공격이 가능하며 불법적이고, 악의적인 사용 등으로 인해 유비쿼터스 컴퓨팅의 발전에 큰 장애를 초래할 가능성이 많다. 따라서 이러한 문제점을 해결하기 위해 유비쿼터스 컴퓨팅 개발시에는 다음과 같은 요구 조건을 반영하여야 한다.

(a) Mobility(이동성)
: 사용자가 가지고 있는 스마트 디바이스(인증 정보가 포함된)는 이동하여 모든 서비스에 사용될 수 있다.

(b) Entity Authentication(개체 인증)
: 사용자가 SM_A를 가지고 Domain_A를 벗어나더라도 Domain_B에서 SM_A의 정보만을 이용하여 인증을 받을 수 있다.

(c) Corresponding Entity Authentication(동일한 개체에 대한 인증)
: Domain_A에 Device_B가 위치해 있을 때, B와 동일한 개체임을 확증하도록 제공하는 것이다. 이 인증은 하나의 도메인에 여러 디바이스가 접속되었을 때, 이전 사용자의 개체를 통하여 디바이스의 인증을 실현하는 것이다. 이러한 인증은 다양한 등급의 보호기능을 제공할 수 있다.

(d) Data Outgoing Authentication(데이타 발신처 인증)
: Domain_A에 의해 제공 될 때, 데이타의 발신처를 요구하는 Domain_B에서의 Device_A가 실제 디바이스라는 것을 Domain_A에 의해 제공된다. 이 인증은 인증 데이타의 발신처에 대한 확증을 제공한다. 그러나 이 인증은 데이타의 중복 혹은 변경에 대한 보호기능은 제공되어서는 안된다.

(e) Connection/Non-connection Confidentiality(접속/비접속 기밀성)
: 접속 기밀성은 Domain_A 상에서 Device_B는 사용자 데이타에 대한 기밀성을 제공해야한다. Domain_A는 B의 정보를 받아 상위로부터 최종 인증을 받는다. 비접속 기밀성은 B의 디바이스는 어떤 도메인과 연결되기 전까지 사용자 데이타에 대한 기밀성을 제공해야 한다.

4. 제안 방식

지금까지 기존 유비쿼터스 컴퓨팅 환경과 JARM 방식, 그리고 PMI에 대해 살펴보았다. 기존 유비쿼터스에 관한 연구는 많이 진행되었고 현재 또한 진행되고 있지만, 가장 활발히 연구되고 있는 분야는 보안 관련 분야보다는 통신과 관련된 분야가 많이 연구되고 있는 실정이다, 보안 관련 연구는 또한 단독으로 연구되는 것이 아니라 전체적인 포괄분야에서 일부로서 다뤄지고 있다. 이에 본 논문에서는 기존 유비쿼터스 연구 중에서 인증만을 단독적으로 연구한 JARM 방식을 비교 연구 대상으로 선정하였다. 따라서 현재 시점에서 다시 기존 연구를 살펴보면 많은 연구가 진행되고 발표되었으리라 생각하지만 본 저자가 연구한 시점에서 연구되었던 논문을 중심으로 비교 분석하도록 하겠다. 기존 연구를 토대로 유비쿼터스 컴퓨팅이 기본적으로 만족시켜주어야 하는 이동성, 개체 인증, 동일한 개체에 대한 인증, 데이타 발신처 인증, 접속/비접속 기밀성 등의 요구조건들도 함께 만족시켜야 함을 알았다. 이에 대해 본 논문에서는 위의 요구사항을 만족하는 방법으로 PMI를 적용하여 멀티 도메인에서 스마트 디바이스에 대한 인증을 실현하였다. 본 논문에서 PMI를 적용하였는데 이는 유비쿼

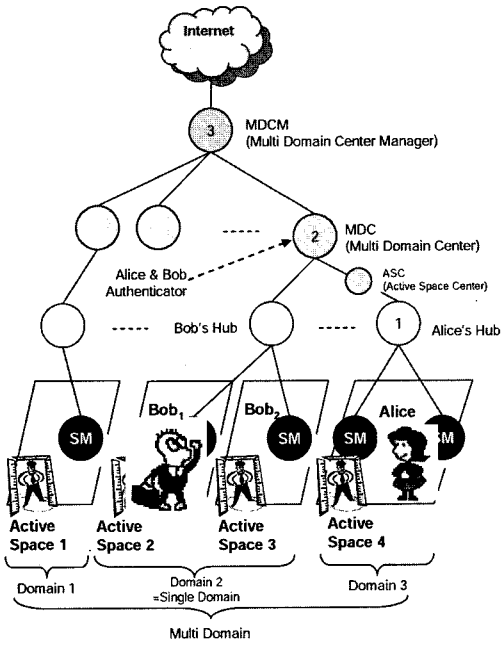

그림 2 제안 방식 전체 구조

터스에서의 디바이스들은 각각의 계산능력, 저장능력 등 여러 부분에서 많은 제약성을 가지고 있다. 그럼에도 불구하고 앞에서 살펴본 요구사항에도 만족하여야한다. 이를 만족하기위해 현재 사용하고 있는 암호 시스템에서 PMI를 적용하면 디바이스의 능력과 요구사항 모두를 만족시킬 수 있다. 각각의 디바이스는 PMI 인증서를 가지고 각각의 인증은 물론 접근 제어가 가능하게 되므로 각각의 디바이스에 대해 인가된 행위만 허락할 수도 있다. 이와 같은 장점 때문에 디바이스에 PMI 인증서를 이용하여 본 방식을 제안하려한다. 그림 2는 본 논문에서 제안하고 있는 방식의 전체적인 흐름도이다. 개괄적인 흐름에 대하여 제안 방식의 고려사항을 살펴본 뒤 기술하도록 한다.

4.1 제안 방식의 고려사항

본 제안방식은 디바이스가 멀티 도메인으로 이동하더라도 이전 도메인의 사용자 정보를 통하여 이동한 도메인에서 사용하는 것을 목표로 하고 있다. 이에 따라 제안 방식에서는 다음과 같은 사항이 고려되어져야 한다.

- 사용자의 디바이스가 단독으로 이동이 능하며 이동한 디바이스는 다른 디바이스와 연계하여 사용할 수 있다.
  : 이는 사용자의 스마트 디바이스가 멀티 도메인으로 이동하였을 경우, 스마트 디바이스는 이동한 도메인에서의 디바이스와 연계하여 어떤 서비스를 받을 수 있다. 이러한 경우 서비스를 위해 필요한 디바이스는 단지 서비스를 위해 존재하게 되며 사용자 정보는 스마트 디바이스로부터 추출되어져야하는 것을 의미한다.
- 사용자의 디바이스는 같은 공간에서는 허브(Hub: 사용자의 모든 디바이스 정보를 가지고 있는 곳)를 이용하여 인증을 받고 다른 공간으로 이동하였을 경우 MDC(Multi Domain Center)를 통해 인증을 받는다.
  : 사용자의 디바이스들이 사용자의 도메인에 있을 경우 사용자 활동공간을 묶어주는 허브를 통해서 인증을 받을 수 있지만 사용자의 공간을 벗어난 경우 스마트 디바이스는 사용자의 허브를 통해서 인증을 받을 수 없다. 스마트 디바이스가 이동할 시에는 허브 이외의 방법으로 인증을 시행해야하는데 이때 멀티 도메인에서 스마트 디바이스에 대한 인증을 하는 것이 MDC이다.
- 최초 인증 정보는 MDC에서부터 사용자 허브를 통해 스마트 디바이스에 발급된다.
  : 인가된 사용자가 자신의 허브에 디바이스를 등록하는 과정을 의미한다. 최초로 사용자가 인증 정보를 MDC로부터 생성하면 그 정보를 허브에 보관하고 이를 다시 스마트 디바이스에 인증 정보를 발급한다. 이때 상위 MDC는 사용자에게 제공한 인증 생성정보를 통해 스마트 디바이스를 인증한다.
- 사용자의 위치정보에 대한 프라이버시는 현 단계에서 고려하지 않는다.

전체적인 흐름은 그림 2에서와 같이 진행된다. 단일 도메인과 멀티 도메인에 있어 사전에 수행하는 작업은 사용자 등록 및 디바이스 등록 작업이다. 이 작업은 최초 MDC가 사용자에 PMI 인증서를 발급할 수 있도록 제공한다. 그 다음으로 수행할 수 있는 작업으로 단일 도메인에서의 인증과 멀티 도메인에서의 인증인데, 우선 단일 도메인에서의 인증은 다음과 같다. Bob은 자신내부에서 활동할 수 있는 활동 공간(Active Space)이 나뉘게 되는데 여기서는 Bob1(Active Space2)과 Bob2(Active Space3)로 기술하며 이 둘의 공간을 합쳐 Domain_2(single Domain)이라한다. Bob의 Device_B가 AS2를 벗어나 AS3로 이동하는 경우를 단일 도메인에서의 인증으로 최초 AS2에 있던 Bob의 SM_B는 자신의 Hub에 이동을 요청하고 Device_B와 Hub에 인증 정보를 전송하고 AS3에 인증을 요청한다. 이때 AS2는 자신이 가지고 있던 SM_B의 인증 정보를 AS3에 전송하여 효율적인 인증이 되도록 한다. 요청받은 AS3는 AS2에게서 받은 인증 정보와 Hub에게서 제공받은 인증 정보를 비교하여 SM_B가 AS3에서 활동할 수 있도록 인증한다. 이로써 단일 도메인에서의 인증이 완료된다. 멀티 도메인 인증은 사용자 Bob이 이동하여 Alice의 Domain_A로 이동

하였을 경우 Bob의 SM_B에 대해 인증 과정을 거쳐야만 Alice의 Device_A를 이용할 수 있다. 이런 경우 Bob의 SM_B는 자신의 Hub에 이동 신호를 보내고 Alice의 AS로 이동한 뒤 인증을 요청하게 되는데 이때 인증 정보는 Alice의 Hub를 통해서 요청하게 된다(그림 2의 1 참조). Alice의 Hub를 거쳐 MDC에 인증을 요청(그림 2의 2)하게 되는데 만약 Bob의 정보가 MDC에 존재하지 않는다면 상위 개체인 MDCM에 요청하여 인증을 수행한다. 이와 같이 전체 시스템은 유기적으로 상위에 포함되어있고 연결되어있지 않은 개체가 있다할지라도 인터넷을 통하여 다른 지역의 디바이스가 와도 인증 과정은 수행될 수 있다.

4.2 시스템 계수

다음은 본 방식에서 사용되는 시스템 계수에 대해 설명한 것이다.

- * : (SM : Smart Device, D: Device, SD: Single Domain, MDC : Multi Domain Center, A : Alicer, B : Bob, MDCM: MDC Manager, ASC: Active Space Center)
- Cert* : *의 공개키를 포함한 인증서
- PCert* : *의 공개키를 포함한 PMI 인증서
- n : PMI 인증서 최대 발급 갯수
- AP : 유효기간(Available Period)
- r: 사용자 Hub가 생성한 난수
- i: 사용자가 발급한 디바이스
- E*() : *의 키로 암호화
- pw : Password
- R* : *의 권한
- ID* : *의 Identity
- H() : 안전한 해쉬함수
- Hub* : *의 Hub

4.3 제안 방식

다음은 본 제안 방식에 대한 자세한 흐름을 기술한다. 첫 번째 방식은 사용자가 자신의 스마트 디바이스를 가지고 자신의 도메인으로 이동하여 이동한 공간의 디바이스를 이용하고자 할 때 자신의 인증 정보가 포함된 스마트 디바이스를 통해 인증을 받게 되는 방식이다. 두 번째 방식은 사용자가 자신의 스마트 디바이스를 가지고 단일 도메인이 아닌 멀티 도메인으로 이동하여 그곳의 디바이스를 이용하고자 할 때 자신의 인증 정보가 포함된 스마트 디바이스를 통해 인증을 받게 되는 것인데 각각의 방식에 대한 그림이다(그림 3, 그림 4 참조).

1) 사용자 등록 및 디바이스 등록

Single/Multi Domain에서 사용하기 위해 사용자는 초기에 여러 디바이스에 인증 정보를 가지도록 해야 한다. 이때 사용자는 MDC(Multi Domain Center)로부터 인증서를 받고 이를 Hub를 통해 자신의 디바이스에 PMI(Privilege Management Infrastructure)를 발급한다. PMI 인증서를 발급하여 스마트 디바이스에 저장토록 하는데 이때 PMI 인증서를 발급은 MDC와 상호 협정한 방법에 따라 인증서를 발급한다.

Step 1. 사용자 A의 Device_A 인증 정보 생성을 위해 다음과 같은 과정이 필요하게 된다. MDC가 사용자 A에게 n개의 PMI인증서를 생성할 수 있는 인증서를 발급한다. PMI 인증서는 사용자 A의 ID, 권한, PMI 인증서에 대한 유효기간으로 구성된다.

$$MDC \rightarrow Hub_A: Cert_A[ID_A, R_A, n, AP]$$

Step 2. 사용자 A는 발급 받은 인증서를 이용하여 Device_A들과 SM_A들에 PMI 인증서를 발급한다. 포함된 인증서는 상위 인증서로의 경로를 포함한다.

$$Hub_A \rightarrow SD_A(or\ Device_A): PC_A = PCert_A$$

$$[ID_A, H(Cert_A \mid r), i] \parallel AP$$

$$SD_A: E_{PK_{DOC}}[PC_A]$$

$$SD_A\ install: E_{PW(orPIN)}[E_{PK_{DOC}}[PC_A]]$$

Step 3. 사용자 A는 자신의 Device_A에 발급된 인증서에 대한 사항을 MDC에 통보한다. 후에 A의 SM_A PMI 인증서가 멀티 도메인에서 사용되었을 경우 SM_A 내의 PMI 인증서 경로로써 사용자 A를 인증 한다.

$$Hub_A \rightarrow MDC: E_{PK_{DOC}}[H(Cert_A \mid r), r, i)]$$

2) 단일 도메인 상에서의 인증

사용자 A의 Hub에서의 SM_A가 AS_A1에서 이동하여 AS_A2에 Device_A2를 사용하고자 할 경우 SM_A는 기존 정보를 그대로 이용하게 된다.

Step 1. SM_A는 최초 공간(AS_A1: Active Space)에 존재하고 있다가 이동이 발생할 경우 이동 신호를 Device_A1에게 보낸다.

$$SD_{A_1} \rightarrow Device_{A_1}: Signal(Outgoing)$$

Step 2. Device_A1은 Hub_A에 SM_A의 이동을 알린다.

$$Device_{A_1} \rightarrow Hub_A: E_{PK_{Hub}}[Device_{A_1}, E_{PK_{Hub}}[PC]]$$

Step 3. 또한 Device_A1은 이동할 Device_A2에 SM_A의 정보를 전송한다.

$$Device_{A_1} \rightarrow Device_{A_2}: [Device_{A_1} \| E_{PW(OrPIN)} E_{PK_{Hub}}[PC]]$$

Step 4. Device_A1로부터 제공받은 인증 정보를 이용하여 Device_A2는 정보와 함께 Hub_A에 전송한다.

$$Device_{A_2} \rightarrow Hub_A: [Device_{A_2} \| E_{PW(OrPIN)} E_{PK_{Hub}}[PC]]$$

Step 5. Hub_A 또한 Device_A1로 제공받은 인증 정보를 확인하고 Device_A2에게서 제공받은 정보와 비교하여 SM_A의 인증을 허락한다.

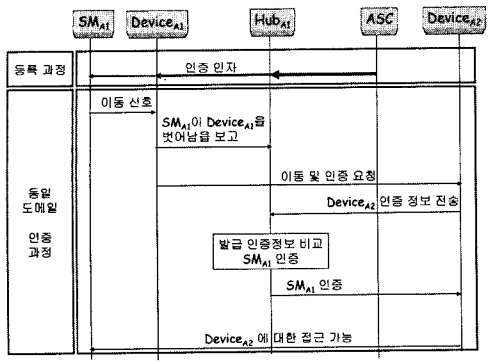

그림 3 단일 도메인에서의 제안 방식 흐름도

$$Hub_A : E_{PW(OrPIN)} E_{PK_{Hub}}[PC] = E_{PK_{Hub}}[PC]$$

$$Compare: (Device_{A_1}) E_{PK_{Hub}}[PC] \doteq (Device_{A_1}) E_{PK_{Hub}}[PC]$$

Step 6. Hub$_A$는 확인을 완료하고 SM_A에 대한 인증을 수락한다.

$$Hub_A \rightarrow Device_{A_2} : [Device_{A_2} \| Auth_{SD}]$$

Step 7. SM_A는 자신의 값을 제공하고 비교한 뒤 이동 공간의 Device_A2에 대한 사용을 허락한다.

3) 멀티 도메인 상에서의 인증

Domain_A의 SM_A가 Domain_B로 이동하여 사용자 A의 정보를 사용하여 Domain_B의 Device_B를 사용하고자 할 경우 SM_A는 사용자 A의 정보를 그대로 이용하게 된다.

Step 1. 자신의 도메인(Domain_A)에 Device_A를 통해 이동 신호를 보낸다. Hub$_A$은 SM_A로부터 이동 신호를 받으면 자신의 공간 목록에서 삭제한다.

$$SD_{A_1} \rightarrow Hub_A : Signal(Outgoing)$$

$$Hub_A : SD_{DeviceList} \rightarrow \nabla ete[SD_{A_1}]$$

Step 2. Hub$_A$은 자신의 Domain_A에서 벗어났음을 MDC에 알린다. 만약 다른 MDC로의 이동일 경우 MDCM에 알려준다.

$$Hub_A \rightarrow MDC : (ID_A, i)$$

$$MDC \rightarrow MDCM : (ID_A, i)$$

Step 3. SM_A는 새로운 도메인(Domain_B)에 위치하였음을 알린 후에 Domain_B의 Device_B에 인증 요청을 한다.

$$SD_{A_1} \rightarrow Hub_B : Signal(Ongoing)$$

$$SD_{A_1} : E_{PW}[E_{PK_{DDC}}[PC_A]] = E_{PK_{DDC}}[PC_A]$$

$$SK_{A_1} \rightarrow Device_B : E_{PK_{DDC}}[PC_A]$$

$$Device_B \rightarrow Hub_B : E_{PK_{Hub_B}}[PC_B, E_{PK_{DDC}}[PC_A]]$$

Step 4. Domain_B의 Hub$_B$는 자신의 Device_B의 인증 정보가 맞는지 확인한다.

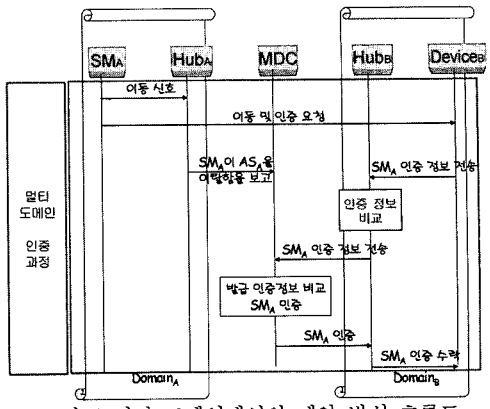

그림 4 멀티 도메인에서의 제안 방식 흐름도

$$Hub_B : E_{SK_{Hub}}[E_{PK_{Hub_B}}[PC_B, E_{PK_{DDC}}[PC_A]]] = PC_B, E_{PK_{DDC}}[PC_A]$$

$$Hub_B : PC_B' \stackrel{?}{=} PC_B$$

Step 5. Hub$_B$는 자신의 Device_B 인증 정보가 통과하면 SM_A의 인증 정보를 MDC에 전송한다.

$$Hub_B \rightarrow MDC : (ID_B, E_{PK_{DDC}}[E_{PK_{DDC}}[PC_A] \mid ID_B])$$

Step 6. MDC는 Domain_B 사용자에게 발급한 인증 정보로부터 생성된 정보인지 확인한다. 확인이 완료되면 SM_A에 대한 인증을 수락한다.

$$MDC : E_{PK_{DDC}}[PC_A] \| ID_B$$

$$MDC : PC_A' = PCert_A[ID_A, H(Cert_A \mid r), i]$$

Step 7. Hub$_2$는 수락 받은 SM_A에 대한 인증을 자신의 Domain_B 상에서 수락하고 Domian_B에서 사용될 Device_B에 대한 사용을 허락한다.

5. 제안 방식 고찰 및 비교 분석

본 장에서는 제안한 프로토콜을 사용자 등록 및 디바이스 등록과 멀티 도메인 상에서의 인증 부분으로 나누어 분석하고 기존에 제시된 프로토콜과 비교 분석한다. 본 방식은 SM_A가 AS1로 이동하여 인증을 받는 것이 아니라, SM_A가 Domain_B로 이동하였을 경우, 어떻게 사용자 A의 정보를 이용하여 Device_B를 사용하도록 인증할 것인가에 대한 연구이다. 기존의 연구에서는 디바이스에 따른 인증 정보를 다르게 부여함으로써 해결하려 하고 있다. 하지만 기존의 방식에는 전체 인증 정보를 획득하는데 있어 모든 디바이스가 있어야한다는 단점이 존재한다. 이것은 만약 디바이스를 분실하였다면 인증 정보를 획득하지 못하게 된다.

본 제안 방식과 기존의 방식을 비교 분석한 것이다. 각각에 대해 자세히 기술하면서 제안방식의 장점을 설명하겠다.

• Mobility(이동성):

표 1 제안 방식 비교 분석표

	이동성	개체인증	동일한 개체에 대한 인증	데이타 발신처 인증	접속/비접속 무결성
JARM	○	×	×	×	○
제안방식 (단일 도메인)	○	○	○	△	○
제안방식 (멀티 도메인)	○	○	○	△	○

사용자가 가지고 있는 하나의 디바이스(인증 정보가 포함된)가 이동하여 모든 서비스를 사용할 수 있다. 본 제안 방식에서는 디바이스내에 PMI 인증서를 포한하고 있기 때문에 사용자의 다른 디바이스와 떨어져 단독으로 이동이 가능하며, 다른 디바이스와 연계하여 사용이 가능하다.

• Entity Authentication(개체 인증)

스마트 디바이스가 자신의 도메인을 벗어나더라도 멀티 도메인에서 이전 사용자의 정보만을 이용하여 인증을 받을 수 있다는 것이다. 사용자가 디바이스가 자신의 PMI인증서를 이용하기 때문에 이동성은 보장되지만 인증서 자체가 단독으로 아무런 보호조치 사용되지 않는다면 불법적인 사용자에 의한 인증 또한 수락될 것이다. 본 제안 방식에서는 이러한 문제점을 해결하고자 각각의 PMI 인증서에 접근할 때 디바이스 접근과 인증서 접근 장치(예, 패스워드, PIN 번호) 등을 통하여 접근할 수 있다.

• Corresponding Entity Authentication(동일한 개체에 대한 인증)

Domain_A에 디바이스가 위치해 있을 때, Device_B에 대해 B와 동일한 개체임을 확증하도록 제공하는 것이다. 이 인증은 하나의 도메인에 여러 디바이스가 접속되었을 때, 이전 사용자의 개체를 통하여 디바이스의 인증을 실현하는 것이다. 이러한 인증은 다양한 등급의 보호기능을 제공할 수 있다. 스마트 디바이스가 이동된다 하더라도 스마트 디바이스 내에 저장된 $E_{PW(orPIN)}[E_{PK_{DOC}}[PC_A]]$을 이용하여 인증을 받을 수 있으며 또한 내부의 인증서인 PMI 인증서는 사용자 A로부터 나온 인증서로 동일한 개체에 대한 인증을 할 수 있다.

• Connection/Non-connection Confidentiality(접속/비접속 기밀성)

접속 기밀성은 Domain_A상의 Device_B는 사용자 데이타에 대한 기밀성을 제공해야한다. Domain_A는 B의 정보를 받아 상위로부터 최종 인증을 받는다. 비접속 기밀성은 Device_B는 어떤 도메인과 연결되기 전까지 사용자 데이타에 대한 기밀성을 제공해야 한다.

본 논문에서는 이러한 것을 해결하고자 인증서에 접근할 수 있는 암호를 설정함으로써 스마트 디바이스 내부에 있는 인증서로의 접근을 차단할 수 있다. 이것은 접속되었을 때의 기밀성과 비접속일 때의 기밀성을 제공할 수 있다.

• Integrity(무결성)

무결성은 각 디바이스에서 이용되는 데이타에 대해 제공하여야 한다. 본 방식에서는 각각의 디바이스에 속성 인증서를 가지고 있으며 이를 통한 각각의 디바이스는 상위 도메인으로부터 인증과 상위 디바이스 접근에 대해 인가받게 된다. 이때 디바이스에서 제공되는 데이타에 대해 무결성을 제공할 수 있어야 하는데, 본 방식에서는 각각의 디바이스가 상위 도메인에서 검증하고 이를 다시 접근하고자 하는 디바이스에 제공하여 접근을 허락하게 된다. 따라서 각각의 속성 인증서는 자신이 상위의 인증서를 올바르게 가지고 있음을 증명할 수 있으므로 무결성을 제공한다고 할 수 있다.

6. 결 론

인터넷의 빠른 확산은 언제, 어디서나 접근 가능한 유비쿼터스 컴퓨팅 환경의 필요성을 제기하고 있다. 유비쿼터스 컴퓨팅 환경에서 사용자는 구체적으로 원하는 것을 명시하지 않아도 필요한 서비스를 접속방식 등과는 무관하게 제공받을 수 있어야한다.

사용자 주변의 디바이스를 연결하고 사용자에게서 인가된 디바이스는 어느 곳에서도 이용이 가능해야 한다. 하나의 디바이스가 사용자를 벗어나 다른 사용자의 공간으로 이동되었을 경우 이동된 공간에서의 사용자 인증이 원활히 수행되어야 한다. 다른 사용자의 공간에서 다른 디바이스가 이전 사용자의 인증 정보로 인증을 실시하지 않고 이동한 곳의 사용자의 인증 정보를 가지고 수행될 수 있기 때문이다.

본 논문에서는 위와 같은 문제를 해결하고자 스마트 디바이스의 이동에 따른 인증을 위해 Hub, ASC, 그리고 MDC라는 개체를 이용하여 계산 능력이 없는 디바이스에 대해 PMI 인증서를 발급함으로써 상위 디바이스에 인증 정보를 제공함으로써 해결하였다. 또한 스마트 디바이스가 멀티 도메인에 이동하여 인증을 요청할 경우 스마트 디바이스가 요청하는 것이 아니라 스마트 디바이스가 속한 도메인내의 디바이스에 요청하고 이를 다시 디바이스가 MDC에 인증을 요청하는 형식으로 이루어져 있다. 이는 사용자 자신의 도메인에서는 Hub를 통하여 인증을 실현하고 멀티 도메인으로 이동시에는 상위에 있는 MDC로써 실현한다. 본 제안 방식은 기존 디바이스에 대한 인증 문제를 해결하려 노력하였다. 향후 발전방향으로는 사용자 프라이버시를 보호하는 측면으로써 사용자의 위치 정보 노출로 인한 사용자 프라이

버시 보호, 키의 간소화 (여러 서비스에 동일하게 사용될 수 있는 키에 관한 연구), 넓은 대역폭을 요구하는 데이타에 대한 원활한 서비스 측면에서 다가설 수 있으리라 본다.

참 고 문 헌

[1] Jalal Al-Muhtadi, Anand Ranganathan, Roy Campbell, and M. Dennis Mickunas,"A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments," ICDCSW '02, pp.771-776, 2002.

[2] Mark Weiser,"Hot Topics: Ubiquitous Computing," IEEE Computer, October 1993.

[3] Sanjay E. Sarma, Stephen A. Weis and Saniel W. Daniel, White Paper:RFID Systems, Security & Privacy Implications, AUTO-ID Center, MIT, Nov, 2002.

[4] 이덕규, 박희운, 이임영, "유비쿼터스 컴퓨팅에서의 도메인간 인증에 관한 연구", 한국정보보호학회 충청지부, pp. 21-34, 2003. 8.

[5] 이임영, 이재광, 소우영, 최용락, "컴퓨터 통신 보안", 도서출판 그린, 2001.

[6] 이근호, "유비쿼터스 컴퓨팅 환경에서의 정보보호", Symposium on Information Security 2003, pp. 629-651. 2003.

[7] M. Roman, and R. Campbell, "GAIA: Enabling Active Spaces," 9th ACM SIGOPS European Workshop, September 17th-20th, 2000, Kolding, Denmark.

[8] Oxygen Project home page. http://oxygen.lcs.-mit.edu/

[9] Portolano home page. http://portolano.cs.washington.-edu/

[10] Aura Project home page. http://www-2.cs.cmu.edu/~aura/

[11] TRON Project home page. http://www.tron.org-/index-e.html

[12] IBM Websphere home page. http://www-3.ibm.-com/software/info1/websphere/index.jsp?tab=highlights

[13] CoolTown home page. http://www.cooltown.hp.com

[14] Microsoft Research, EasyLiving Website, http://www.-research,microsoft.com/easyliving

[15] 이성용, 정현수, "Ubiquitous 연구 동향 및 향후 전망", Worldwide IT 제 3 권 7호, pp. 1-12, 2002.

[16] 이윤철, "최근 홈 네트워크 기술동향 및 시장 전망", ITFIND 주간기술동향(TIS-03-20) 1098호, pp. 22-33. 2003.

[17] A. Aresenault, S. Tuner, Internet X.509 Public Key Infrastructure, Internet Draft, 2000. 11.

[18] ITU-T, Draft ITU-T RECOMMANDATION X.509 version4, ITU-T Publications, 2001. 5.

[19] S. Farrell, R. Housley, An Internet Attribute Certificate Profile for Authorization, Internet Draft, 2001.

이 덕 규

2001년 2월 순천향대학교 컴퓨터공학과 졸업. 2003년 2월 순천향대학교 전산학과 석사. 2003년 3월~현재 순천향대학교 전산학과 박사과정. 관심분야는 Broadcast Encryption, DRM, EKE

박 희 운

1997년 2월 순천향대학교 컴퓨터공학부 학사. 1999년 2월 순천향대학교 전산학 전공 석사. 2002년 2월 순천향대학교 전산학전공 박사. 2002년 1월~현재 한국정보보호진흥원 선임연구원. 관심분야는 암호이론, 컴퓨터 · 네트워크 보안

이 임 영

1981년 8월 홍익대학교 전자공학과 졸업 1986년 3월 오사카대학 통신 공학 전공 석사. 1989년 3월 오사카대학 통신공학 전공 박사. 1989년 1월~1994년 2월 한국전자통신연구원 선임 연구원. 1994년 3월~현재 순천향대학교 정보기술공학부 교수