Pervasive Computing 환경을 위한 인증기법에 관한 연구

정칠현*, 송주석*

*연세대학교, 컴퓨터과학 · 산업시스템공학과

Authentication Mechanism for Pervasive Computing Environments

Chul-hyun Jung*, Joo-seok Song*

*Department of Computer Science and Industrial System Engineering Yonsei Univ.

요 약

시대의 흐름에 따라 컴퓨팅 환경에서도 많은 변화가 이루어져 왔다. 이는 컴퓨팅 환경이 초기 독립적인 환경에서 네트워크를 이용한 분산환경으로, 이어서 무선통신을 이용한 이동통신으로 발전되어 왔음을 의미한다. 이에 따라, 언제/어디에서나 원하는 자원에 접근할 수 있는 환경이 제공되게 되었고, 이동성을 지원하는 Mobile IP, Ad Hoc 등의 개념에 의해 제한된 환경에서의 컴퓨팅이 아닌 이동이 가능한 환경에서의 컴퓨팅으로 확장되기에 이르렀다. 현재에 이르리서는 디바이스의 소형화를 통해 사용자의 이동성을 극대화시키는 개념인 "Pervasive Computing"에 대한 연구가 활발히 진행 중에 있다.

이러한 Pervasive 컴퓨팅 환경에서도 보안기능에 대한 기능이 최근 중요시되고 있다. 특히 pervasive computing 환경은 기존의 컴퓨팅 환경과는 차별화 되는 특징들을 내포하고 있기 때문에 상호 인증의 제공이 필요하다. 본 논문에서는 Pervasive Computing 환경에서 현재 진행중인 인증 기법에 대한 연구들을 알아본 후, 이러한 인증기법들의 취약점을 보완할 수 있는 상호 인증을 고려한 새로운 인증 기법을 제안하고자 한다.

I. 서론

"Pervasive"란 "퍼지는, 보급하는, 배어드는, 스며드는"이란 뜻으로, Pervasive Computing 환경이라 함은 우리 주변에 널리 퍼져있는 유형, 무형의 디바이스를 통한 컴퓨팅을 가능하게 하자는 개념이다. 이러한 개념이 등장함으로 인해 좀더 편리한 컴퓨팅 환경으로의 전환이 이루어지게 되었다. 하지만 아직까지는 개념에 불과할 뿐 표준의 제정이나 구체적인 구현에는 미치지 못하고 있다. 따라서, 기초적인 요구사항들에 대한 연구나 사용되는 애플리케이션 및 기반구조에 대한 조사[1][2][3]를 제외한 실제적인 구조의 설계는 정해진 표준에 의한 것이 아닌 각 연구마다 환경을 설정하고 이에 맞는 구조를 설계하고 있는 현실이다.

또한 Pervasive Computing 환경에서의 보안에 관한 연구는 학계뿐만 아니라 기업체에서도 이루어지고 있기는 하지만 기존의 유선 망에서의 보안 기술을 기반으로 연구가 진행되고 있어 변화되는 환경에는 적합하지 않다. 현재 진행중인 연구의 대표적인 예로 Tiny SESAME[4]와 Vigil[5]을 들 수 있다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 이미 개발된 Tiny SESAME와 Vigil system의 인증기법에 대해 알아보고 3장에서는 상호 인증을 고려한 새로운 보안구조를 제안하겠으며, 4장에서 이를 기존 시스템의 구조와 비교 및 분석하여 5장에서 결론을 맺도록 하겠다.

II. 기존의 인증기법

1. Tiny SESAME의 인증기법

Smart Home 환경에서의 사용자 혹은 다른 디바이스와의 상호작용을 가능하게 하기 위해서는 가벼우면서도 분산환경에 적당한 보안 메커니즘이 필요하다. 이러한 요구사항에 부합하기 위하여 Sun Microsystem에서 Jini™ 기술을 이용하여 Tiny SESAME를 개발하였다. Tiny SESAME는 SESAME를 component 기반으로 Java로 구현되었기 때문에 경량화 되었다. SESAME는 공개키 기반 기술과 접근제어, 접근권한의 위임 등을 제공하는 Kerberos의 확장이다.

Tiny SESAME는 Kerberos의 인증체계를 따른다. Kerberos를 직접 사용하지 않는 이유는 Kerberos는 대칭키 암호 알고리즘을 사용하기 때문에 커다란 개방형 환경에는 맞지 않으며, 접근제어를 제공하지 않기 때문이다. 그래서, SESAME에서는 Kerberos를 확장시키면서 로그인을 위한 전자서명, 오프라인 사전 공격 방지,

사용자를 위한 접근 제어 권한의 제공, 각기 다른 키 관리와 분산 환경 서비스를 제공한다. 또한 인터넷 표준인 GSS-API(General Security Services Application Programming Interface)[6]를 제공한다. GSS-API는 일반적인 보안 서비스를 위한 프로그래밍 인터페이스의 표준이다.

Tiny SESAME는 단순암호와 무결성 검증 그리고 RBAC(Role-Based Access Control)을 제공하는 UIUC SESAME의 일부분이다. Tiny SESAME는 구조를 경량화하기 위해 컴포넌트 기반구조를 채택하였으며 현재 Tiny SESAME의 버전은 패스워드 기반의 인증을 통해 개체인증을 제공하고 있다.

2. Tiny SESAME의 문제점

디바이스의 경량화를 위해 동적인 component의 적재방식을 사용하였으나 이러한 component의 적재도 로컬메모리에서의 적재를 전제로 하고 있기 때문에 디바이스의 보조메모리를 필요로 하게 되며, Client side에서도 웹 브라우저와 같은 특별한 접근 도구가 필요하게 된다. 또한 사용자와 클라이언트 디바이스와의 인증이 결여되어 있기 때문에, 사용자의 이동성이 지원된다 하더라도 상호인증이 되지 않는 보안상 문제가 존재하게된다.

3. Vigil System의 인증기법

Vigil System은 분산 시스템에서 보안과 접근제어를 제공하기 위하여 설계되었으며, 대부분의 클라이언트가 hand-held 디바이스인 SmartSpaces에서 최적화 되어있다.

Vigil system 내에서의 모든 개체들은 시스템 내에 자신을 등록시킬 때, Security Agent에 자신의 인증서를 전달하고, Security Agent의 인증서를 돌려 받는다. 이 때, 주고받은 인증서는 자신이 스스로 검증하거나 로컬에 있는 Certificate Controller로 전송을 하여 검증할 수 있다. 인증서 내에는 신뢰하고 있는 인증기관의 리스트와 인증서의 검증 규칙이 들어있다. 그러므로 인증서가 신뢰된 인증기관으로부터 발행되었거나, 검증 규칙을 만족하면서 인증서가 폐지되지 않았다면 해당 인증서는 유효하다고 간주된다.

4. Vigil System의 문제점

여기서도 사용자의 이동성 제공에 따른 상호인증이 이루어지지 않기 때문에, 보안상 문제가 발생할 수 있다. 또한 디바이스에서의 직접적인 인증성의 검증을 하기 위해서는 디바이스내의 검증 모듈이 구현되어야 하는데, 이것은 디바이스를 경량화 한다는 기본 개념에 적당하지 않다.

그리고, 접근권한의 위임의 경우 해당 권한의 위임이 서로 상충되거나 위임에 의한 체인이 중간에 끊기게 되면 의도하지 않았던 보안상의 문제들이 일어날 수 있다.

III. 제안하는 시스템

새롭게 제안하는 시스템의 구조는 상호인증, 사용자의 이동성 그리고 분산된 보안 시스템의 구현을 고려하여 설계되었다.

1. 환경

서두에서도 언급했듯이, Pervasive Computing 환경은 크게 세 가지의 제약조건을 가지게 된다.

첫째, 사용자 및 디바이스의 이동성이 극대화 되어야하기 때문에 디바이스는 가능한 경량화, 소형화되어야 한다.

둘째, 사용자 및 디바이스의 이동은 예측이 불가능하며, 인증시 필요한 정보는 단 한곳에 저장되어야 한다.

셋째, 기존의 환경에서는 사용자가 물리적으로 접해있는 디바이스와의 인증이 생략되어 있었다. 그러나 Pervasive Computing환경에서는 인증이 이루어지지 않은 사용자-디바이스간, 사용자-Agent간의 인증이 이루어진 후에만 접근이 가능해야 한다.

2. 구조

사용자 및 디바이스는 단 하나의 Space에만 속한다. 따라서 자신의 HA에만 등록정보를 저장하게 된다. 이러한 정보는 사용자 및 디바이스가 최초 등록 시 자신의 Home Space의 Agent에 저장된다. 또한 사용자 및 디바이스가 이동을 하게 되어 인증이 필요로 하게 될 때는 항상 해당 사용자 및 디바이스의 HA를 통해서 인증이 이루어진다.

본 논문에서 제안하는 시스템의 구조는 그림 1과 같다.

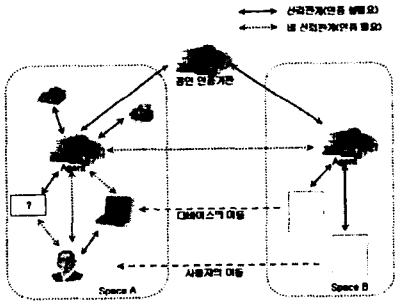

그림 1: 제안하는 시스템의 구조

3. 절차

1) 등록

사용자 및 디바이스는 최초 자신의 Home Space의 Agent(HA)에 자신의 존재를 등록하여야 한다.

사용자는 자신의 ID, Password를 등록하고, 자신과 HA간의 대칭키와 HA의 인증서를 받게되며 이를 Smart Card에 안전하게 저장한다.

디바이스는 자신의 ID를 등록하고, 자신과 HA간의 대칭키 그리고 HA의 인증서를 받아 안전한 장소에 저장한다.

2) 디바이스의 이동

Space A의 디바이스가 Space B로 이동하였을 때, 디바이스는 자신의 HA의 인증서를 FA에 제시한다. 이때 FA는 HA에게 해당 디바이스가 Space A에 등록되어있는 디바이스임을 질의하고 이에 대한 확인이 끝난 후, 디바이스에 임시적으로 ID와 대칭키를 부여한다.

3) 사용자의 이동

Space A의 사용자가 Space B로 이동하였을 경우, 다음과 같은 프로토콜을 이용하여 중간 경로상의 개체들과의 상호인증을 수행한다.

■ User(Smart Card)→Device

$$E_{K_{User-HA}}[ID_{User}, Password, Nonce_{User}], Cert_{HA}, ID_{User} \quad (1)$$

■ Device→FA

$$E_{K_{Device-FA}}[E_{K_{User-HA}}[ID_{User}, Password, Nonce_{User}], Nonce_{Device}, Cert_{HA}, ID_{User}], ID_{Device} \quad (2)$$

■ FA→HA

$$E_{KU_{HA}}[E_{K_{User-HA}}[ID_{User}, Password, Nonce_{User}], Nonce_{FA}, Cert_{FA}, ID_{User}] \quad (3)$$

■ HA→FA

$$E_{KU_{FA}}[Nonce_{FA}+1, E_{K_{User-HA}}[Nonce_{User}+1]] \quad (4)$$

■ FA→Device

$$E_{K_{Device-FA}}[Nonce_{Device}+1, E_{K_{User_{HA}}}[Nonce_{User}+1]] \quad (5)$$

■ Device→User(Smart Card)

$$E_{K_{User_{HA}}}[Nonce_{User}+1] \quad (6)$$

(1)에서 사용자와 HA간에 공유하고 있는 대칭키를 이용하여 사용자의 ID와, 패스워드 그리고 인증의 검증을 위한 Nonce값을 암호화하여, 사용자의 ID와 HA의 인증서를 함께 Device에 전송한다.

(2)에서 Device는 사용자로부터 받은 데이터와 사용자와 Device간의 인증 검증을 위한 Nonce값을 Device와 FA간의 대칭키를 사용하여 암호화하여, Device의 ID와 함께 FA에 전송한다. FA에서는 ID를 이용하여 Device와의 대칭키를 찾고, 이를 이용하여 전송 받은 데이터를 복호화 하게 되고, 여기서 HA의 인증서와 Device의 Nonce값을 알게 된다. HA의 인증서를 인증기관(CA)를 통해 검증 받고, (3)에서 HA의 공개키를 이용하여 나머지 정보와 자신의 인증서, 그리고 자신의 Nonce값을 암호화하여 HA에 전송한다.

HA에서는 전송 받은 사용자의 ID를 이용하여 사용자와의 대칭키를 찾고, 그 키를 이용하여 전송 받은 데이터를 복호화 한다. 여기서 복호화 한 데이터중의 ID와 평문으로 받은 ID를 비교하여 무결성을 보장받고, ID에 대한 Password를 검증한다. 이러한 절차가 완료되면 (4)를 통해서 FA와 User로부터 받은 각각의 Nonce값에 1을 더하여 각각과 공유하고 있는 키를 이용하여 암호화하여 전송한다.

(5)와 (6)에서 사용자와 Device는 각각 자신이 보냈던 Nonce값과 돌아온 Nonce+1값을 비교하여 중간에서의 데이터의 변조가 없었다는 것과 자신과 각각의 Agent간의 인증이 이루어졌다는 것을 묵시적으로 확신하게 된다.

4) 서비스의 이용

사용자는 자신의 존재를 FA나 HA에 의해 자신이 현재 위치하고 있는 곳에서 인증 받게 된다. 이때, 사용자가 현재 위치하고 있는 Space내의 서비스를 이용하고 싶은 경우, 해당 Space내의 Agent를 통하여 권한을 부여받고 사용하게 되므로 추가적인 인증절차는 필요가 없다. 그러나 현재의 Space내의 서비스가 아닌 제3의 Space에 존재하는 디바이스에서의 서비스를 받기 원한다면 현재의 Space내의 Agnet와 제3의 Space의 Agent와의 인증 절차를 통해 추가적인 인증이 요구되어진다.

IV. 비교 분석

표 1: 시스템 비교

	Tinv SESAME	Vigil	제안한 시스템
디바이스의 경량화	component based (DCL이용)	없음	대칭키 사용, 인증을 Agent에 위임
상호인증	패스워드를 이용한 사용자인증	없음	모든 개체간
확장성	제한된 공간에서의 인증	인증서의 공유에 따른 확장의 제한	확장이 용이
특기사항		권한의 위임	

표 1에서 보는 것과 같이 제안한 시스템에서의 특징을 보면 다음과 같다.

1. 디바이스의 경량화

디바이스에서의 인증을 검증하기 위하여 Agent와의 대칭키를 사용함으로써 공개키를 사용하는 인증서를 이용한 방법보다 좀더 간단한 모듈로 구현이 가능하다.

2. 상호인증

기존의 시스템에서 제공할 수 없었던 디바이스와 사용자간의 상호 인증과 경로상의 모든 개체간의 인증이 보장된다.

3. 확장성

사용자 및 포터블 디바이스의 인증정보가 자신의 HA에만 저장, 관리되기 때문에 사용자 및 디바이스의 이동 및 네트워크의 확장이 용이하다.

V. 결론

기존의 연구들에서는 디바이스의 경량화를 위하여 모듈을 동적으로 적재시키거나, 권한의 위임을 통해 사용자 인증을 대신하고 있다. 그러나 근본적으로 Pervasive Computing환경에서는 사용자, 디바이스, 서버들이 서로 비 신뢰관계에 있을 경우가 많기 때문에 서로간의 상호인증이 중요하다. 여기서 제안한 인증기법은 제한된 디바이스의 자원을 이용하여 사용자, 디바이스, 서버들이 각각 서로를 인증하고, 필요하다면 암호화 통신을 이용하여 보다 안전한 통신이 이루어질 수 있다. 또한 사용자의 이동성을 보장하고 사용자 인증에 필요한 정보들이 분산되어 관리됨에도 불구하고 중복되어 저장되지 않기 때문에, 관리상의 편리성도 제공된다.

아직까지도 Pervasive Computing환경은 이상적인 환경에 대한 개념에 불과할 지도 모르지만 기존의 환경과의 공통점에서 점점 확장시켜 나간다면 머지 않은 장래에 현실화될 수 있을 것이다.

참고문헌

[1] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, pp. 10-17, Aug. 2001.

[2] A.C. Huang, B. Ling, S. Ponnekanti, and A. Fox. "Pervasive Computing: What Is It Good For?", *Workshop on Mobile Data Management*, pp. 84-91, 1999.

[3] Swarup Acharya, "Application and Infrastructure Challenges in Pervasive Computing", *NSF Workshop on Context-Aware Mobile Database Management*, Jan. 2002.

[4] J.Al-Muhtadi, M.Anand, M.Mickunas, R.Campbell, "Secure Smart Home using Jini and UIUC SESAME", UIUCDCS-R-99-2142, Dec. 1999.

[5] Lalana Kagal, Jeffrey Undercoffer, Filip Perich, Anupam Joshi, and Tim Finin. "A Security Architecture Based on Trust Management for Pervasive Computing Systems.", *In Proceedings of Grace Hopper Celebration of Women in Computing* 2002.

[6] J. Linn, "Generic Security Service Applica- tion Program Interface Version 2", RFC 2078, Jan. 1997.