커뮤니티 컴퓨팅 보안

김순동[O] 심윤주 김현숙 최동순 조위덕
아주대학교 유비쿼터스시스템 연구센터
{sdkim[O], bluepond, virtus78, gooday, chowd}@ajou.ac.kr

Security for Community Computing

Soondong Kim[O], Yunju Shim, Dongsoon Choi, Hyeonsook Kim, Weduk Cho
Center of excellence for Ubiquitous System

요 약

유비쿼터스 컴퓨팅 패러다임이 제시된 이래로 세계적으로 유비쿼터스 컴퓨팅에 대한 연구가 활발히 진행되고 있다. 그러나 아직까지 유비쿼터스 컴퓨팅은 현실세계에 적용하기에는 많은 기술적 · 비기술적 장벽들이 많이 존재한다. 보안과 프라이버시에 관한 사회적인 우려는 유비쿼터스 컴퓨팅 기술의 실제 환경에 적용하는데 큰 걸림돌로 작용한다. 따라서 진행되고 있는 많은 유비쿼터스 연구들은 각각의 프로젝트에 적합한 보안 및 프라이버시 보호를 위한 기술들을 개발하여 적용하고 있는 실정이다. 유비쿼터스 시스템 연구센터는 유비쿼터스 컴퓨팅을 구현하기 위해 커뮤니티 컴퓨팅을 제안하였다. 커뮤니티 컴퓨팅에서 발생할 수 있는 보안 문제점들을 해결하기 위해 커뮤니티 스킨을 구현하였다. 커뮤니티 스킨은 커뮤니티 구성원들을 커뮤니티 외부 개체들과 구분 짓고, 외부의 침입으로부터 보호해 줄 수 있다. 커뮤니티 스킨은 암호학적 알고리즘을 적용함으로써 구현될 수 있다. 본고에서는 보안 서비스를 4가지 유형으로 분류하고 이를 토대로 커뮤니티가 보안 협약을 통한 커뮤니티 스킨을 형성 하도록 하였다.

1. 서 론

유비쿼터스 컴퓨팅[1] 연구가 전세계적으로 활발히 진행되고 있다. 그러나 아직 유비쿼터스 컴퓨팅이 실제 환경에 적용되기에는 많은 장벽들이 존재한다. 눈앞에서 컴퓨터의 실체가 사라지지만 어디에서나 존재하는 컴퓨터들로 인해 인간의 모든 행동이나 데이터는 언제 어디서나 관찰(captured) 되고 수집(collected)되며 저장(stored)될 수 있다. 이러한 가능성의 존재로 인해 많은 사람들은 유비쿼터스 컴퓨팅에 대해 부정적인 시각으로 바라보게 되며 이러한 현상을 해결하는 것이 유비쿼터스 컴퓨팅을 실제 환경에 적용하는데 시급한 과제일 것이다.

유비쿼터스 컴퓨팅 연구들은 아직까지 실험적인 성격을 많이 띄고 있으며, 센싱 기술, 장치소형화 기술 및 RFID등 유비쿼터스 기술들은 특정 도메인에 국한되어 연구되고 있다. 본 유비쿼터스 시스템 연구센터에서는 범용적인 유비쿼터스 서비스를 좀더 쉽고 효율적으로 개발 할 수 있도록 커뮤니티 컴퓨팅을 제안하였다[2]. PICO[3] 혹은 RCSM[4] 같은 기존의 프로젝트에서도 커뮤니티 컴퓨팅이라는 개념을 서비스 제공 차원에서 사용하였지만 유비쿼터스 시스템 연구센터에서는 서비스의 효율적인 개발을 돕고 재사용률을 높일 수 있는 방향으로 연구되었다. 커뮤니티 컴퓨팅 모델을 정의하고 이를 기반으로 커뮤니티 매니지먼트 시스템을 개발하였으며 동적 서비스 검색(dynamic service discovery)과 결합(binding)을 통해 서비스 개발 단계와 실행단계를 분리하였다. 따라서 이러한 커뮤니티 컴퓨팅의 특성을 반영하는 새로운 보안 메커니즘이 필요하게 되었으며 본 논문은 커뮤니티 매니지먼트 시스템 보안 이슈들과 그를 해결하기 위한 보안 메커니즘을 제시하였다.

2. 커뮤니티 매니지먼트 시스템

2.1 커뮤니티 개념

커뮤니티는 일반적으로 " 물리적으로 닫힌 공간에 존재하는 사람들의 집합" 혹은 " 동일한 관심사를 가진 사람들의 모임" 등으로 정의된다[5,6,7]. 커뮤니티 매니지먼트 시스템에서는 커뮤니티를 동일한 목적을 달성하기 위한 서비스들의 집합으로 정의하였다.

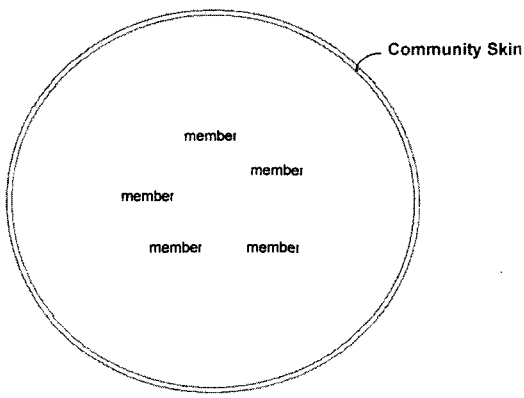

그림 1 커뮤니티 개념모델

커뮤니티의 개념모델은 그림 1과 같다. 커뮤니티는 커뮤니티의 목적을 달성하기 위해 협력해야 하는 구성원들과 커뮤니티가 제공해야 하는 기본 서비스들, 그리고 커뮤니티 스킨으로 구성된다. 커뮤니티 스킨은 커뮤니티 구성원들과 커뮤니티 외 개체간을 분리하며 커뮤니티 내 구성원들을 외부로부터 보호한다.

2.2 커뮤니티 생명주기

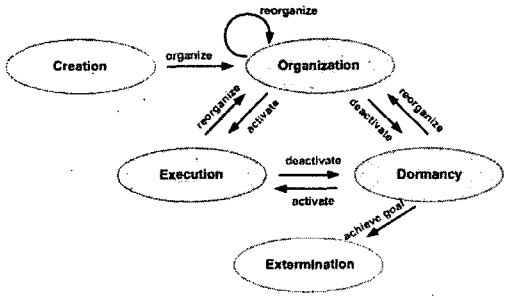

그림 2 커뮤니티 생명주기

그림2는 커뮤니티의 생명주기를 보여준다. 생성단계는 커뮤니티 서비스 개발자가 서비스를 구성하는 단계이다. 이 단계에서 개발자는 커뮤니티 서비스의 목적을 결정하고 해당 목적을 달성하기 위해서 협업 해야 하는 서비스들을 추상화 하여 기술한다. 조직단계에서는 추상화 수준으로 기술된 서비스들을 실제 환경의 서비스로 결속한다. 이 단계에서 실제 서비스들이 커뮤니티에 구성원으로 등록된다. 구성단계가 완료되고 특성 상황이 발생하면 커뮤니티는 실행단계가 된다. 실행단계는 커뮤니티에 묶이게 된 서비스들이 실제로 서비스를 수행하는 단계이다. 커뮤니티가 수행 중이거나 조직단계 일 때 특성 상황이 발생할 경우 휴면단계로 넘어갈 수 있다. 이 단계는 다시 활성상황이 될 때까지 대기한다. 마지막으로 커뮤니티의 목적이 달성되면 커뮤니티는 소멸된다.

3. 커뮤니티 보안 이슈

3장에서는 커뮤니티에서 발생할 수 있는 보안 이슈들에 대해서 살펴본다. 커뮤니티 컴퓨팅과 유비쿼터스 컴퓨팅 환경을 고려하여 필요한 보안 이슈들을 정리하고 커뮤니티 매니지먼트 시스템에 필요한 보안 메커니즘들을 정리하였다.

3.1 커뮤니티 구성원 보안 수준과 유비쿼터스 컴퓨팅 환경의 제한된 컴퓨팅 자원

커뮤니티 스킨을 구현함에 있어서 커뮤니티내의 컨텍스트의 민감성에 따른 커뮤니티 보안 수준을 고려하여야 한다. 정보를 공유하는 구성원들은 해당 정보의 수준에 부합하는 보안 서비스를 제공하여야 하며, 수준에 미달되는 컴퓨팅 개체는 커뮤니티에 가입되어서는 안 된다. 커뮤니티 구성원들은 공유하는 컨텍스트의 흐름에 따라 몇 개의 그룹으로 구분될 수 있다. 각 그룹에 속해 있는 구성원들은 그룹 내 컨텍스트의 보안 요구사항을 지원할 수 있어야 한다. 커뮤니티 내의 모든 컨텍스트의 보안 요구사항을 만족하는 보안 정책을 사용할 경우 민감하지 않은 컨텍스트를 유통하는데 있어 불필요한 부하가 생기게 되어 효율적이지 않다.

유비쿼터스 환경은 컴퓨팅 능력이 매우 낮은 장치들에서부터 PC급 혹은 더 낮은 컴퓨팅 장치들까지 다양한 컴퓨팅 장치들로 구성될 수 있다. 낮은 컴퓨팅 장치들을 좀 더 효율적으로 사용하기 위해서는 불필요한 커뮤니티에 참가하는데 있어 제약을 최소화 하여야 한다.

3.2 탐색과 보안협약

유비쿼터스 지능공간에서는 많은 컴퓨팅 장치들과 그들의 서비스들이 산재해 있다. 많은 경우에서 특정 서비스는 여러 컴퓨팅 장치에서 공통으로 제공되기도 하며 또 어떤 경우에는 하나의 장치에서 여러 서비스를 제공하기도 한다. 예를 들면 ' display' 기능은 컴퓨터에서도 가능하며 LCD TV에서도 가능하다. 2장에서 언급하였듯이, 생성단계에서는 실제 서비스가 구동될 환경에 독립적으로 서비스를 추상화 수준에서 기술한다. 본고에서는 이러한 서비스를 메타 서비스(meta-serivce)라고 부른다. 메타 서비스는 서비스의 유형을 나타내며 ' text display' , ' alarm sound' 등으로 표현된다. 추상 수준으로 기술된 각 구성원들은 제약사항(constraints)를 가지고 있다. 동일한 여러 후보 서비스들 중에 제약사항을 만족하는 서비스만이 커뮤니티 구성원으로 선택될 수 있다. 예를 들어 제약사항이 ' distance:closest' 일 경우, 후보 서비스들 중 가장 가까운 서비스가 선택되게 된다. 메타 서비스들은 각각이 보안서비스 요구사항을 제약사항으로 가지게 된다. 이것은 서비스가 다루게 되는 컨텍스트의 민감성에 기반하여 설정된다. 예를 들어 자신의 비밀번호가 전송 될때는 기밀서비스(Confidentiality)가 제약사항이 되며, 금융결제 같은 트랙잭션에 대해서는 ' digital sign & non-repudiation' 이 필요하게 된다. 각각의 메타 서비스에 설정된 보안 제약사항을 만족할 수 있는 개체만이 커뮤니티 구성원으로 선택되어야 하며 보안 협약 과정을 통해서 요구사항 만족여부를 판단할 수 있어야 한다. 각 후보 서비스 개체들은 서로 다른 보안 알고리즘들을 가지고 있으며 보안 협약 과정을 수행함으로써 서로간의 데이터 교환에 사용할 보안 알고리즘 및 비밀 키 등으로 교환하여야 한다.

4. 구현

4.1 보안 서비스 유형 구분

앞서 언급한 바와 같이 요구되는 보안 서비스는 해당 서비스의 특징들에 의해서 결정된다. 본 논문에서는 커뮤니티 생성 단계에서 커뮤니티를 기술하는 개발자가 직접 보안서비스 요구사항을 기술하도록 하였다. 요구되는 보안 서비스는 다음과 같이 구분될 수 있다.

- None
- Message Integrity (ex. MD2, MD5, SHA1)
- Confidentiality (ex. DES, tripleDES, AES)
- Digital Signature (ex. MD5withRSA, SHA1withDSA)

어떤 메타서비스는 보안 서비스가 요구되지 않을 수도 있지만 금융서비스 같은 민감한 프로세싱은 디지털 서명 등 보다 강력한 보안 서비스가 필요하다. 메타 서비스의 제약사항에 보안 제약사항으로 위에 언급된 4가지 보안 서비스들을 기술할 수 있으며 각각을 조합하여 기술할 수 있다. 예를 들어 " Message Integrity & Confidentiality " 혹은 " Confidentiality & Digital Signature" 로 기술하여 적용 가능하도록 하였다.

4.2 커뮤니티 구성과 보안 제약

커뮤니티 구성단계에서 커뮤니티는 생성단계에서 작성된 커뮤니티 기술과 서비스 환경에 적용되어 있는 실제 서비스들을 기반으로 커뮤니티를 구성하게 된다. 많은 후보 서비스들 중에 메타서비스가 요구하는 제약사항을 만족시키는 개체만이 커뮤니티 구성원이 될 수 있다. 각각의 개체는 자신의 보안 능력을 커뮤니티 매니지먼트 시스템에 전달하여 자신이 커뮤니티 구성원이 될 수 있는지 없는지를 전달받게 된다. 예를 들어 서비스 개체 A는 가용한 알고리즘이 ' MD2' , MD5' 밖에 없다고 가정하면 메타서비스 M의 보안 제약사항이 ' Confidentiality' 라면 A는 커뮤니티 구성원으로 선택되지 않는다. 하지만 M의 보안 제약사항이 ' Integrity' 라면 A는 커뮤니티 구성원으로 채택될 수 있게 된다.

4.3 가정

4.3.1 키 분배

보안 알고리즘의 선택과 더불어 가장 민감한 사안이 키의 안전한 분배이다. 컴퓨팅 능력이 낮은 디바이스들은 안전한 키의 분배에 많은 제약이 따른다. 이를 극복하기 위해서 다음과 같은 가정이 필요하다.

- ✓ 가정1. 신뢰할 수 있는 키 서버가 존재한다.
- ✓ 가정2. 각각의 컴퓨팅 장치들은 사전에 키 서버와 키를 교환하고 안전한 채널을 통해 키를 분배 받을 수 있다.

4.3.2 어뎁터 적용

어뎁터는 커뮤니티 매니지먼트 시스템의 프래임워크이다. 이 프래임워크를 사용하여 응용프로그램 개발자는 메시지 전달이나 통신채널 확립, 서비스 검색 등의 일들을 고민하지 않아도 된다. 서비스를 검색하는 과정에서 발생하게 되는 보안 협약과정등도 이에 포함된다. 본 논문에서 나타나는 모든 서비스들은 어뎁터를 사용하도록 개발된다고 가정한다.

4.4 보안협약 과정

그림 3는 보안 협약 과정을 간단히 나타내고 있다. 커뮤니티 매니지먼트 시스템과 응용프로그램이 서로간의 보안 알고리즘을 확립하면 별도의 키분배 서버를 통하여 키를 전달받고 이를 확인한 후 안전한 채널을 형성하게 된다. 각각의 어뎁터는 키서버에 자신의 식별자, 상대방 식별자, 선택된 알고리즘, 선택된 키 길이를 전달하고 키서버는 서로에게 같은 요청이 오면 키를 생성하여 전달한다.

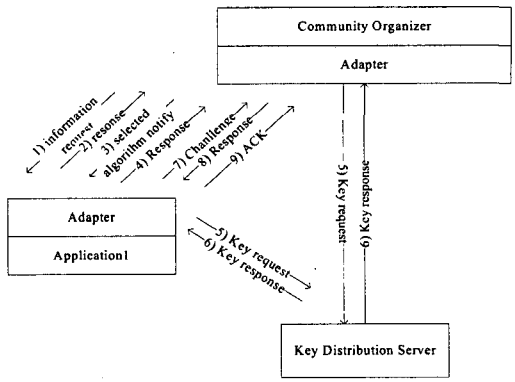

그림 3 보안협약 과정

5. 결론

본 논문에서는 유비쿼터스 환경에 적합하게 제안된 커뮤니티 매니지먼트 시스템에 대해서 간략히 소개하고 해당 시스템에 필요한 보안 메커니즘을 제안하였다. 커뮤니티 생명주기에 따른 특징에 맞게 메타 서비스 보안 제약사항을 설정하고 그 타입을 정의 하였으며 실행단계에서 적절한 보안 채널을 형성하는 프로세스를 제안하여 커뮤니티 구성원들이 안전하게 서비스를 수행하는 시스템을 구현하였다. 현재 커뮤니티 스킨은 외부로부터 커뮤니티 내부의 구성원들을 보호하는 형식으로 제안되었으며 이를 위한 최소한의 요구사항을 만족하도록 구성하였다. 외부로부터 좀더 고도화된 추측(inference)등의 공격 등에 대한 대비가 필요할 것으로 예상되며 커뮤니티 내부의 구성원들간의 프라이버시 보호를 위한 고민이 필요하다.

참고문헌

[1] M. Weiser, The computer for the 21st century. Human-computer interaction:toward the year, 2000, p933.940, 1995.
[2] 김현숙, 최동순, 조위덕, 지속적인 협업 서비스를 지원하는 커뮤니티 시스템, HCI2006 학회, 1권 287~292p, 2006
[3] Mohan Kumar, et al., PICO: A Middleware Framework for Pervasive Computing, IEEE Pervasive Computing, Vol.2, No.3, July/September 2003
[4] S. S. Yau and F. Karim, A Context-Sensitive Middleware-based Approach to Dynamically Integrating Mobile Devices into Computational Infrastructures, Journal of Parallel and Distributed Computing, vol. 64(2), February 2004, p.301-317
[5] J. Michael Yohe, Community Comuputing and the Computing Community, Proc. of ACM SIGUCCS Conference on User Services, October 1994
[6] J. Sawamoto, K. Mutoh, H. Tsuji, and H. Koizumi, Evaluation of Multi-Agent Model for Community Formation in Network Society, Proc. of International Conference on Advanced Information Networking and Application, 2004
[7] T. Nishimura, H. Yamaki, T. Komura, and T. Ishida, " Community Viewer: Visualizing Community Formation on Personal Digital Assistants, " ACM SIGAPP Applied Computing Review, Vol.6, No.1, February 1998, pp.13-18

※ 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업의 지원에 의한 것임