情 報 保 護 學 會 誌
第 16 卷 第 2 號, 2006. 4

상황 인식 보안에 관한 고찰

정 목 동*

요 약

본 고에서는 상황 인식(Context-Awareness)의 의미와 관련 기술을 소개하고, 상황 인식에 바탕을 두고 있는 상황 인식 보안(Context-Aware Security) 기술 동향을 살펴본 다음 그 응용 예를 알아본다. 상황 인식에 관한 연구는 국내외적으로 상당한 연구가 진행 되어 왔지만, 상황 인식 기술을 컴퓨터 보안에 응용하는 상황 인식 보안 연구는 최근에야 시작 되었기 때문에 본 고를 준비하는데 어려움이 있었다.

Ⅰ. 서 론

컨텍스트(context)에 관한 정의는 정의하는 사람에 따라, 자신의 주관적 판단에 따라 조금씩 달라진다. 컨텍스트에 대한 용어를 처음으로 사용한 것으로 되어 있는 Shildt and Theimer[3]는컨텍스트의 정의를 사용 장소나 주변의 사람이나 사물의 집합, 또는 시간이 지남에 따른 변화 등을 일컫는다고 컨텍스트를 정의하고있다. Dey[4]는 이를 좀더 일반화 하여 엔티티(entity)의 상태를 특징 지울 수 있는 어떤 정보도 모두 컨텍스트로 정의하고 있다. 여기서 엔티티는 사람이나, 사용자와 애플리케이션 간의 의사 소통에 관계되는 사물 등이 될 수 있다고 규정하고 있다.

본 고에서는 상황 인식(Context-Awareness)의 의미와 관련 기술을 소개하고, 상황 인식 기술을 정보보호에 응용하는 상황 인식 보안(Context-Aware Security) 기술 동향을 살펴본 다음 그 응용 예를 알아본다. 상황 인식에 관한 연구는 마크 와이저의 유비쿼터스 컴퓨팅 개념[1,2]이 제시된 이래 국내외적으로 상당한 연구가 진행 되어 왔지만, 상황 인식 기술을 컴퓨터 보안에 응용하는 상황 인식 보안 연구는 비교적 최근에야 시작되었다.

본 고의 구성은 다음과 같다. 2절에서 컨텍스트(Context)와 상황 인식 컴퓨팅(Context-Aware Computing)에 대해서 고찰하고, 3절에서는 상황인식 컴퓨팅 기술을 컴퓨터 보안에 응용하고 있는 상황 인식 보안(Context-Aware Security)의 핵심 기술에 대해서 살펴본다. 4절에서는 상황 인식 보안의 사례를 들고, 5절에서 결론을 내린다.

Ⅱ. 상황 인식 컴퓨팅과 유비쿼터스 컴퓨팅

2.1 컨텍스트(Context) 및 상황 인식 컴퓨팅 (Context-Aware Computing)

Shildt and Theimer[3]는 컨텍스트의 정의를 사용 장소나 주변의 사람이나 사물의 집합, 또는 시간이 지남에 따른 변화 등을 일컫는다고 컨텍스트를 정의하고 있다. 한편 Dey[4]는 이를 좀더 일반화 하여 엔티티(entity)의 상태를 특징 지울 수 있는 어떤 정보도 모두 컨텍스트로 규정하고 있다. 여기서 엔티티는 사람이나, 사용자와 애플리케이션 간의 의사소통에 관계되는 사물 등이 될 수 있다고 정의하고 있다. 상황 인식 컴퓨팅에 관한 최초의 연구는 일반적으로 1989년과 1992년 사이에 있었던 Olivetti의 Active Badge system으로 알려져 있다. 이 시스템은 건물에 있는 직원에게 자동으로 전화를 걸 수 있도록 위치 정보를 알려주는 시스템인데, 배지를 부착하고 있는 직원들의 위치 정보가 주기적으로 센터로 보내진다.[5]

상황 인식 컴퓨팅(context-aware computing)은 넓은 의미에서 유비쿼터스 컴퓨팅 기술에 속할 수 있으나 상황인식 컴퓨팅은 실세계의 상황 특징을 표현하는 정보 기술에서 시작한다는 점에서 유비쿼터스 컴퓨팅 기술과 차이를 보인다고 하지만[20], 유비쿼터스 환

* 부경대학교 전자컴퓨터정보통신공학부 (mdchung@pknu.ac.kr)

경이 구축되기 위해서는 상황인식 기술 없이는 불가능하므로 상황인식 기술은 유비쿼터스 컴퓨팅 기술의 핵심 기술에 속한다고 하겠다. 유비쿼터스 컴퓨팅(ubiquitous computing)은 사용자 및 환경을 인식하고(context-aware), 언제 어디서나 어떤 디바이스를 통해서도 끊이지 않는(seamless) 서비스를 제공하는 것을 말하는데, 한 예로서 병원이라는 공간에 있는 다양한 사용자에 따라 서로 다른 수준의 접근제어가 제공되어야 한다는 것을 들 수 있다. 즉, 중요 기밀정보에 대해서는 인가된 사용자라도 언제, 어디서나 정보의 접근이 허용되어서는 곤란하며, 인가된 사용자에게만 접근이 허용되어야 하고 이를 사용자와 주변의 상황 정보를 이용하여 제어할 수 있어야 한다[18].

2.2 유비쿼터스(Ubiquitous) 컴퓨팅과 퍼베이시브(Pervasive) 컴퓨팅

마크 와이저에 의하면 가장 심오한 기술은 숨어버리는(disappear) 기술[1,2]이라고 언급하면서, 사람을 포함한 현실 공간에 존재하는 모든 대상물들을 상호 연결해서 사용자에게 필요한 정보나 서비스를 즉시 제공할 수 있는 컴퓨팅 환경을 유비쿼터스 컴퓨팅으로 정의하고 있다. 인간을 컴퓨터의 세계로 들어가게 하는 현재의 기술보다 인간의 환경에 보다 적합한 컴퓨터는 마치 숲 속을 가볍게 산책 하는 것 같이 컴퓨터를 사용하게 되는 것 이라고 마크 와이저는 제시하고 있다.

유비쿼터스 응용의 가장 큰 특징 중 하나는 지능형 에이전트를 이용한 개인화 된 서비스(personalized service)를 제공할 수 있다는 것이다. 이를 위해 지능형 사용자 에이전트는 사용자의 현재 상황을 정확하게 파악하고 필요로 하는 서비스를 지능적으로 제공한다. 상황 인식을 위한 상황 관리 기술, 이벤트 전달을 위한 이벤트 서비스 기술, 응용의 효율적인 수행을 위한 브로커 서비스 기술 등이 있다[20].

Berkeley 대학의 ICEBERG 프로젝트[6]는 인터넷을 기반으로 다양한 플랫폼들과 서비스를 적절하게 선별적으로 제공하되 이용자의 위치나 접속 네트워크의 범위 뿐만아니라 이용자의 개인 취향이나 요구에 적응화된 맞춤형 서비스를 제공하는 것을 목표로 하고 있다. ICEBERG 아키텍처는 다음과 같은 서비스를 목표로 하고 있다.

- Any-to-any 통신 서비스: 모든 종류의 기기들 사이의 효과적인 통신을 의미한다.
- 개인 이동 서비스 (Personal mobility services) : 기기가 아닌 사람이 통신 단말(end point)이 된다는 의미이다.
- 맞춤형 통신 서비스 (Communication service customization): 최종 사용자에게 통신 서비스를 사용자의 선호도에 따라서 맞춤형으로 선택할 수 있도록 해준다는 의미이다.
- 사용자 행위-기반 서비스(User activity-driven services): 사용자의 위치에만 의존하는 현재의 컨텍스트 인식에서 사용자의 행위 컨텍스트를 지원함으로써 컨텍스트를 일반화 한다.

유비쿼터스 컴퓨팅은 수천의 디바이스와 센서들로써 구성되어 있으며, 컴퓨팅 자원과 물리적인 공간이 끊이지 않는 결합이 가능하도록 하고, 사용자들에게 편리하고 다(多)-정보(information-rich) 스마트 공간(smart space) 환경을 제공한다. 그러나 이런 유비쿼터스 컴퓨팅은 당연히 보안과 개인 정보 보호 문제를 야기하게 된다. 그렇지만 기존의 인증과 접근제어 문제는 수동적인 사용자의 개입을 요구하는데[8] 이는 사용자의 개입을 최소화해야 하는 non-intrusive 유비쿼터스 컴퓨팅 정신을 훼손하게 된다. 따라서 유비쿼터스 컴퓨팅 보안은 사람의 개입을 최소로 하는 정보보호 기술이 필요하게 된다.

유비쿼터스 기술은 대체로 다음과 같이 분류되고 있다[19].

- 기반 기술 : 언제 어디서나 편리하고 안전하게 컴퓨터를 이용할 수 있는 기술
  - 사용자 인증 기술이나 보안 기술
  - 상황 인식 기술
- 하드웨어 기술 : 사물과 환경속에 내장하여 인간 생활을 인간의 생활을 지원하고 개선해 줄 수 있는 기술
  - 비가시적인 입출력장치 기술,
  - SoC 기술
  - 저전력 기술
  - 고집적화 기술
  - 내장형 기술
- 통신 기술 : 센서네트워크를관리할 수 있는 라우팅 기술
  - 센서네트워크 기술
  - 근거리 무선통신 기술
  - 장치접속 기술
- 애플리케이션 개발 기술
  - 인공지능기술 : 학습, 계획, 판단

- 응용서비스 제공 기술 : Java, Wap, XML 등을 이용한 P2P 기술 바탕

유비쿼터스 컴퓨팅과 퍼베이시브 컴퓨팅(pervasive computing)은 혼용할 수 있는 용어로 외국에서는 퍼베이시브 컴퓨팅(pervasive computing) 용어를 많이 쓰는 경향이 있다. 퍼베이시브 컴퓨팅(pervasive computing)은 모바일 컴퓨팅의 상위 개념으로서, 모바일 컴퓨팅의 이동성에 호환성(interoperability), scalability, smartness, 비가시성(invisibility) 등의 특성을 추가로 요구하고 있으며, 사용자가 컴퓨팅을 원하는 경우 끊이지 않는 접근이 가능하게 한다. 또한 퍼베이시브 컴퓨팅은 우리의 생활을 단순하게 해주고, 컴퓨팅의 특성을 바꿔준다. 가까운 미래에 가정에서의 정보 처리나 통신 등의 작업이 네트워크 기능이 잘 갖춰진 지능적인 장비에 의해서 사람이 인지하지 못한 가운데 일어날 가능성이 있다.

우리는 아직도 컴퓨터를 어떤 일을 하기 위해서 들어가게 되는 가상의 환경으로 정의하고 있다[7]. 그러나 퍼베이시브 컴퓨팅(pervasive computing)의 개념은 가상적인 환경과는 차이가 난다. Saha는 퍼베이시브 컴퓨팅의 개념을 다음과 같이 언급하고 있다[7].

- 디바이스는 사용자가 반드시 다루어주어야 하는 소프트웨어의 저장소가 아니라 애플리케이션-데이터 공간으로 들어가는 관문(portal)이다. 애플리케이션은 디바이스의 능력을 활용하기 위해서 작성된 소프트웨어가 아니라 특정 태스크를 수행하는 수단이다. 그리고 컴퓨팅 환경은 소프트웨어를 저장하고 실행하는 가상적인 환경이 아니라 풍부한 정보를 가지고 있는 물리적인 공간으로 보고 있다.

〔표 1〕은 퍼베이시브 컴퓨팅의 연구 동향을 보여주고 있다[7].

Ⅲ. 상황 인식 보안 (Context-Aware Security) 핵심 기술

3.1 상황 인식 컴퓨팅 (Context-Aware Computing) 기술

상황 인식 컴퓨팅 (Context-Aware Computing)이란 유비쿼터스 환경에서 임의의 애플리케이션이 사용자의 환경요소에 대한 상황 정보(Contextaul Infomation)[1,4] (물리적 상황 : 사용자의 위치, 디바이스의 종류 및 성능, 정서적 상황 : 사용자의 선호도, 역사적 상황 :현재 시간 등)를 감지하여, 사용자가 이를 이용할 수 있도록 해주는 컴퓨팅 패러다임이라 정의 할 수 있다. 즉, 환경요소의 상황정보인 컨텍스트(Context)[1,4]와 사용자가 입력한 정보를 결합하여 사용자가 처한 상황에 맞게 사용자가 원하는 형태로 조정하여 사용자에게 적절한 서비스를 제공하는 것을 말한다.

유비쿼터스 환경에서는 음악, 영화와 같은 다양한 멀티미디어 컨텐츠 그리고 휴대폰에서 PDA, 고성능 PC에 이르기까지 유무선망을 기반으로 한 다양한 자원들이 공존하므로 도처에 편재된 컴퓨팅 인프라를

〔표 1〕 퍼베이시브 컴퓨팅 연구 동향

프로젝트 명	연구 기관	기 능
Aura	CMU	착용(wearable), handheld, 데스크탑, 인프라구조 컴퓨터
Endeavour	Berkeley 대학	Planet-scale, self-organizing, 적응적"정보 유틸리티"
Oxygen	MIT	컴퓨터는 산소같이 어디서도 가용. 8개의 환경 가용 기술에 바탕
Portolano	Washington 대학	비가시적인, 의도 기반(intent-based) 컴퓨팅 구현, 사용자의 의도 파악
Sentient Computing	AT&T Lab., Cambridge, UK	센서와 자원의 상태 자료를 연계 시키는 사용자 인터페이스 연구
Cooltown	HP	웹 기술, 무선 네트워크, 휴대 장비등의 기술을 확장해서 이동 사용자, 물리적 엔티티와 e-서비스 사이의 가상의 다리 구축
EasyLiving	MS	지능 환경을 위한 아키텍처와 관련 기술, 미들웨어, 지리 모델링, 인식 기술, 서비스 표현 기술 등
WebSphere Everyplace	IBM	WebSphere 소프트웨어 플랫폼을 확장해주는 미들웨어와 애플리케이션에 초점

통해 사람과 컴퓨팅 기기 및 환경이 서로 상호작용하여 다양한 자원에 대한 환경 정보인 컨텍스트(context, Context-Awareness)[1,4]를 생성 및 통합하고 통합된 정보를 바탕으로 적응적으로 보안 서비스를 제공하는 기법이 요구된다.

3.2 MAUT (Multi-Attribute Utility Theory) 기술

다양한 자원의 서로 다른 특성을 통합하여 계량적인 수치로 표현하는 방법론 중에 유틸리티 이론이 있다. MAUT[14]는 다중변수에 대한 의사결정 문제(decision problem)에서 유틸리티(utility)를 통한 정략적인 의사결정 방법이다. 유틸리티 분석(utility analysis)은 의사결정자(decision maker)가 원하는 제비뽑기(lottery)의 결과를 분석해주는 분야로서 의사 결정자는 이들 결과에 대한 개인의 선호도(preference)를 유틸리티 수(utility number)로 표현한다.

유틸리티는 0과 1사이의 상대적인 값으로서 가장 선호하지 않는 결과 유틸리티를 $u(x^0)=0$, 가장 선호하는 결과 유틸리티를 $u(x^*)=1$로 나타낸다. 그리고 결과에 대한 유틸리티 수의 대입은 의사결정자의 최적행동의 기준이 되는 기대 유틸리티(expected utility)를 최대화 시켜주는 쪽으로 이루어진다.

특히 MAUT를 e-commerce에 사용한 예로 Pmart (*Pukyong*-mart)[15]에서는 가격에만 의한 기존의 협상을 개선하기 위하여 상품의 특성, 보장 기간, 서비스정책 등 다양한 조건에 대해 MAUT를 이용하여 협상을 수행하는 에이전트 중재에 의한 전자 상거래 프레임워크이고, 독일의 "Stiftung Warentest"[16]는 MAUT를 이용해 상품을 평가하는 방법으로 소비자에게 널리 알려져 있다.

사용자를 포함한 다양한 자원들의 환경 요소들이 혼재하게 되는 유비쿼터스 환경에서 다양한 변수에 기반하여 사용자의 선호도를 결정 하기 위해서는 자원마다 다른 환경 상황 정보 유틸리티를 기반으로 MAUT이론을 적용하는 것이 효율적일 수 있다.

3.3 RBAC (Role-Based Access Control) 기술

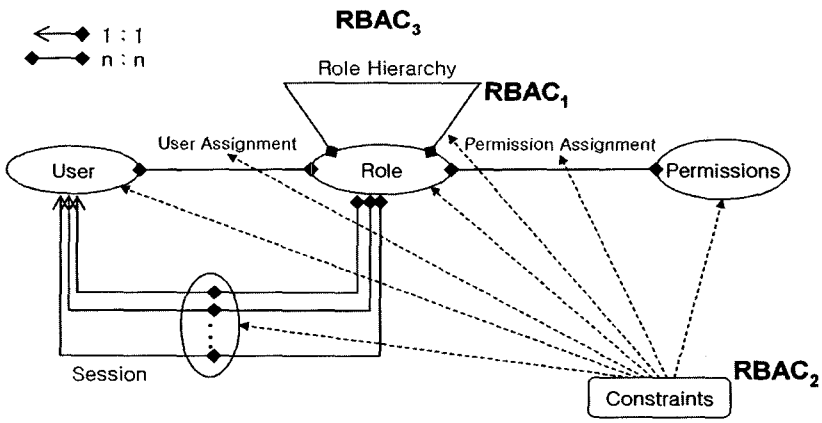
Ravo S. Sandhu에 의해 제안된 RBAC[12,13]는 사용자의 역할에 기반을 둔 접근 통제 방법으로 조직 내의 사용자 허가 할당에 있어서 복잡성과 비용, 잠재적인 실수를 줄이기 위한 강력한 기법을 제공한다. 또한 조직 수준에서 보안 관리를 증진시키기 위해서 사용자 식별자 수준이 아닌 추상화 수준으로 제공하므로 권한 관리를 매우 단순화 시켜주고, 특정한 보안 정책을 구현하는데 있어서 유연성을 제공하는 장점이 있다.

혼재된 자원의 환경 요소가 동적으로 변화하는 유비쿼터스 환경에서 다양한 자원의 보안 설정을 개별적으로 설정하고 관리하는 것은 복잡하고 많은 부담이 따른다. 따라서 RBAC 모델을 적용하면 권한 관리를 단순화 시켜주고 보안정책의 설정과 변경이 쉽고 변경된 보안 정책이 잘 반영 될 수 있는 유동적인 보안 관리 시스템을 제공 할 수 있다.

[그림 1]은 RBAC의 개념적인 모델인 $RBAC_1$, $RBAC_2$, $RBAC_3$을 보여준다.

3.4 GRBAC (Generalized Role-Based Access Control) 기술

지금까지 보안은 정적인 것으로 인식되어서 접근 제어도 컨텍스트에 따라 변경되지 않았으며 동적인



[그림 1] RBAC 모델 [12]

환경의 변화를 반영할 수 없었다. 이를 해결하기 위해서는 상황 인식 컴퓨팅을 보안 서비스에 도입하여야 한다. 상황 인식 컴퓨팅을 위한 보안 서비스는 자연스럽고, 사용자의 개입을 최소로 하는 non-intrusive 형태의 것이어야 한다. 따라서 더 이상 보안을 위한 사용자 세션이 시간에 무관하게 동일한 인증 기법과 접근 기법을 사용한다고 가정 할 수 없다. 예를 들면, 상황 인식 인증 기법은 풍부한 정보로 이루어져 있는 디지털 홈(digital home, aware home)과 같은 환경에서는 굉장히 복잡하다. 인증 정책은 여러 요소에 의해서 제약을 받는데 사용자 종류와 사용자의 현재 위치 등의 정보에 따라서 많이 달라진다.

기존의 RBAC는 역할에만 기반을 두고 상황 정보를 사용하지 못했기 때문에 다음과 같은 한계를 가지게 되었다[12, 13, 18].

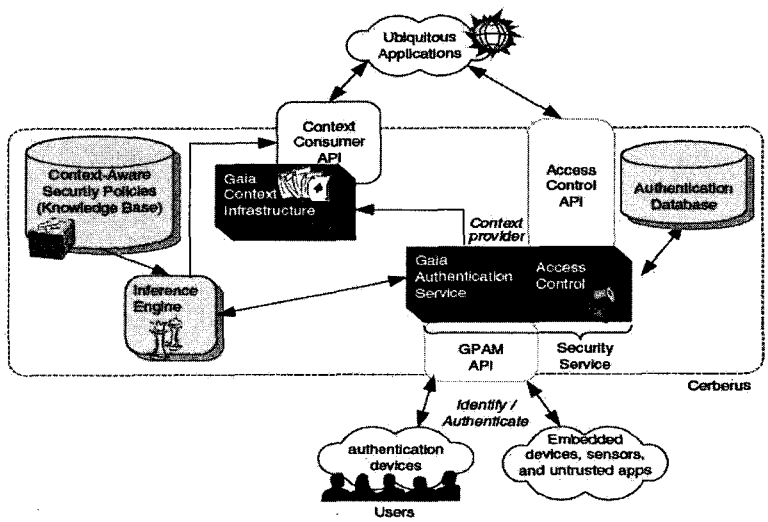
- 개별 사용자들이 멤버로서 요구되는 역할에 근거하여 이루어짐.
- 접근 권한은 역할이름에 따라 그룹화됨.
- 역할이 사용자 업무 책임과 권한에 따라 부과됨으로써 자원의 이용에 제약

이를 해결하기 위하여 GRBAC (Generalized Role-Based Access Control)[9,10] 개념을 고려하게 되었는데, 이는 기존의 역할기반 접근 제어가 시간에 따른 접근제어 등과 같이 상황에 근거한 접근제어를 수행할 수 없었던 문제점을 해결하고 있다. GRBAC는 상황에 근거한 접근제어를 수행하기 위하여, 접근제어 결정에 사용자 역할 (subject role), 객체 역할(object role), 환경 역할 (environment role)을 추가하여 기존 RBAC 모델을 확장하였다. GRBAC 모델은 기존 RBAC 모델보다 상황 인식 애플리케이션의 접근 정책을 나타내는데 강력하고 융통성 있는 방법을 제공하고 있다[11]. 또한 디지털 홈 등을 실제 사용하게 될 사용자들은 컴퓨터나 보안 관련 초보자일 가능성이 크므로 여기에 사용될 애플리케이션에서 보안 정책을 쉽게 정의하고 사용할 수 있는 기능은 매우 중요하다. 이런 면에서 다양한 역할에 바탕을 둔 GRBAC 모델을 이용한 보안 서비스는 유비쿼터스 컴퓨팅 환경의 보안 서비스를 구현하는데 유용한 방법이 될 수 있다.

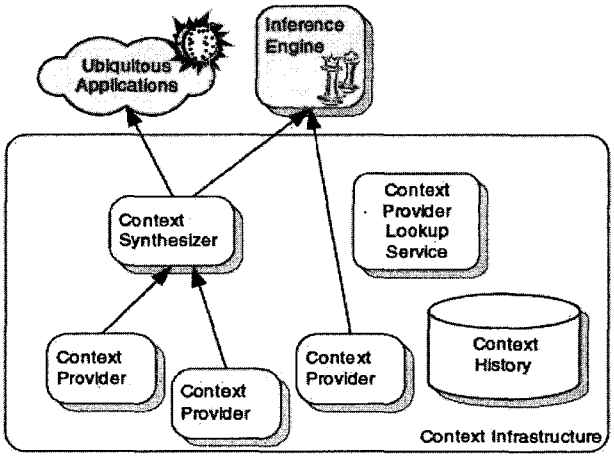
Ⅳ. 상황 인식 보안 사례

4.1 Gaia[8]

Gaia 프로젝트는 Illinois 대학에서 수행하고 있는 프로젝트로서 일반적인 계산 환경을 정의하고 이를 이용해서 물리적 환경과 유비쿼터스 컴퓨팅 디바이스를 프로그램 가능한 컴퓨팅 및 통신 시스템으로 통합하고 있다[8]. 보안과 상황 인식 기술은 어떤 스마트 공간에서도 중요한 2가지 핵심 서비스인데, Gaia의 Cerberus는 인식(identification), 인증(authentication), 상황인식과 추론등의 핵심 기능을 담당하고 있다. [그림 2]는 Gaia의 Cerberus 전체 구조를 보여주고 있다.



[그림 2] Gaia의 Cerberus 전체 구조[8]



〔그림 3〕 Gaia의 컨텍스트 인프라 구조[8]

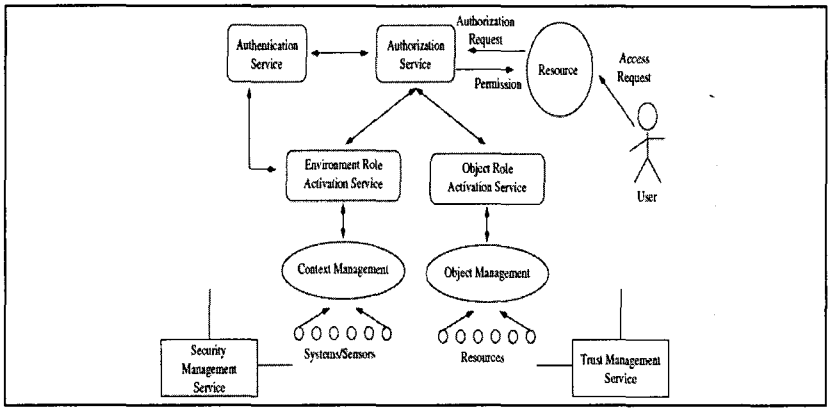
Cerberus는 4개의 컴포넌트로 구성되어 있는데 1) 보안 서비스, 2) 컨텍스트 인프라 구조, 3) 보안 정책 지식 베이스, 4) 추론 엔진 등이다. Gaia의 컨텍스트 인프라 구조는 〔그림 3〕과 같다.

〔그림 3〕에서 각 모듈의 기능은 다음과 같다.

- Context Provider
  - 센서나 데이터 소스로부터 컨텍스트 정보를 수집하여 애플리케이션에 제공
- Context Synthesizer
  - Context Provider에서 수집한 컨텍스트 정보를 상위개념의 상황정보로 추론하고 추상화하여 애플리케이션에 제공
- Context Provider Lookup Service
  - 컨텍스트 정보를 제공하는 Context Provider를 알려줌
- Context History
  - 이전의 컨텍스트 정보 이력들을 기록하고 있는 데이터베이스
- Context Consumer
  - 컨텍스트 정보를 사용하는 애플리케이션 서비스

4.2 CASA (Context-Aware Security Architecture)[11]

Georgia Tech에서는 상황 인식 기술을 보안 서비스에 제공하기위하여 CASA (Context-Aware Security Architecture)를 제안하고 있다. 〔그림 4〕는 CASA 시스템의 논리적인 컴포넌트의 전체적인 구조를 보여주고 있는데, 다-정보 컴퓨팅 환경에서 동작하면서 애플리케이션에 보안 서비스를 제공해주고 있다. 소프트웨어 컴포넌트 개념을 제공해주고 있



〔그림 4〕 CASA 시스템의 논리적 개념도[11]

```
<GRBAC TABLES>
    <POLICY>
        <SROLE> Child </SROLE>
        <OROLE> Dangerous Appliance </OROLE>
        <ACTION> ALL </ACTION>
        <EROLE> Working Hours </EROLE>
        <PERMS> Deny </PERMS>
    </POLICY>
</GRBAC TABLES>
```

〔그림 5〕 XML로 표현한 정책의 예

```
ERole    ::= < L_Exp > | < L_Exp >< L_Exp >< L_Opr >
< L_Exp > ::= True| False | < M_Exp >< M_Exp >< C_Opr >
< M_Exp > ::= < Meta > |< Meta >< Meta >< M_Opr >
 < Meta > ::= < Constants > | < Environment_Variable>
< L_Opr > ::= AND | OR | NOT
< C_Opr > ::= = | <> | > | >= | < | <=
< M_Opr > ::= + | - | * | /
```

〔그림 6〕 환경 역할의 BNF 형태 정의

으며 J2SE SDK를 이용하여 구현되고 있다. GRBAC 정책을 정의하기 위해서 GPDL (Generalized Policy Definition Language) 언어를 사용하고 있다.

〔그림 5〕는 XML로 표현한 정책의 예인데, 정해진 환경 조건 하에서 *child* 사용자가 위험한 물건에 접근하는 경우에 *deny* 되는 경우를 보여주고 있다[11].

〔그림 6〕은 BNF 형태로 표기한 환경 역할의 정의이다[11].

〔그림 7〕은 환경 역할의 예로서 환경 역할 *Party*에 관한 형식 정의를 보여주고 있다[11].

```
Party      ::= L_Exp1 L_Exp2 AND
L_Exp1     ::= M_Exp11 10 >=
L_Exp2     ::= M_Exp21 50 >=
M_Exp11    ::= Person_LivingRoom
M_Exp21    ::= NoiseLevel_LivingRoom_DB
```

〔그림 7〕 환경 역할 *Party*의 예

〔그림 7〕의 의미는 (**Person_LivingRoom** 〉= 10 AND **NoiseLevel_LivingRoom_DB** 〉= 50)와 같다. ERole (Environment role :환경 역할)은 기 정의된 ERole로 부터 상속 받을 수 있다. 가령 ERole *Party*와 ERole *Weekend*가 이미 정의 되어 있다면 새로운 ERole인 *Weekend_Party*는 *Party*와 *Weekend*의 논리적 AND를 취함으로써 정의 될 수 있다.

Authorization Service는 다음의 튜플로 표기 된다.

〈SRole, ORole, Action, ERole, Perms〉

여기서 SRole, ORole, Action, ERole, Perms은 사용자 역할(subject role), 객체 역할 (object role), 행위, 환경 역할(environment role), 허용 여부를 각각 의미한다.

환경 역할 행위 서비스(Environment Role Action Service)는 특정 환경 변수에 따른 조건에 의해서 시스

템 상태, 역할의 활성화/비활성화에 관한 정보를 가진다.

인증 서비스는[11] 푸쉬 모델(push model)과 풀 모델(pull model) 2가지로 구분이 되는데, 푸쉬 모델은 사용자가 시스템에 자신을 인증할 수 있는 비밀 정보를 보내면 저장된 정보에 따라서 인증이 진행된다는 점에서 기존의 인증 모델과 유사하다. 풀 모델은 인증이 필요한 순간에만 인증 과정이 진행 된다는 점에서 기존의 모델과 다르다. 인증 관리 서비스는 Context Management Services SNMP(Simple Network Management Protocol)와 CTTK(Context Toolkit)에 바탕을 두고 있다. 인증과 암호화를 위해서 CA 기반 PKI를 구축하고 있다.

V. 결 론

본 고에서는 상황 인식(Context-Awareness)의 의미와 관련 기술을 소개하였고, 상황 인식 기술을 보안에 응용한 상황 인식 보안(Context-Aware Security) 기술 동향을 살펴본 다음 그 응용 예를 알아보았다. 상황 인식에 관한 연구는 마크 와이저의 유비쿼터스 컴퓨팅 개념이 제시된 이래 국내외적으로 상당한 연구가 진행 되어 왔지만, 상황 인식 기술을 컴퓨터 보안에 응용하는 상황 인식 보안 연구는 최근에야 시작 되었기 때문에 본 고를 준비하는데 어려움이 있었다.

유비쿼터스 컴퓨팅이나 퍼베이시브 컴퓨팅을 구축할수록 눈에 드러나지 않는 숨는 기술이 일반화 될 터인데, 이런 환경일수록 적절한환경 변수에 대한 컨텍스트 정보를 컴퓨터 보안에 응용하는상황 인식 보안 연구는 더욱 중요해 질 것이고, 보안 개념 없는 유비쿼터스 컴퓨팅이나 퍼베이시브 컴퓨팅은 생각할 수 없는 만큼 이 부분에 대한 깊이 있는 연구가 필요하다 하겠다.

참 고 문 헌

[1] M.Weiser, Ubiquitous Computing, http://www.ubiq.com/hypertext/weiser/

[2] M.Weiser, "The Computer for the Twenty-First Century," *Scientific American*, pp. 94-10, September 1991

[3] B.Schildt and M.Theimer, "Diseminating Active Map Information to Mobile Hosts," *IEEE Networks*, 8(5), 1994, pp.22-32.

[4] A.K.Dey, *Providing Architectural Support for Building Context-Aware Applications*, PhD Thesis, College of Computing, Georgia Institute of Technology, 2000.

[5] G.K.Mostefaoui, et al., "Context-Aware Computing: A Guide for the Pervasive Computing Community," In *Proc. of the IEEE/ACS International Conference on Pervasive Services (ICPS'04)*, 2004.

[6] http://iceberg.cs.berkley.edu/

[7] D.Saha and A. Mukherjce, "Pervasive Computing: A Paradigm for the 21st Century," IEEE Computer, March 2003, pp.25-31.

[8] J.A. Muhatadi, et al., "Cerberus: A Context-Aware Security Scheme for Smart Spaces," In *Proc. of the First International Conference on Pervasive Computing and Communications (PerCom'03)*, 2002.

[9] M.J.Convington, et al., "Generalized Role-Based Access Control for Securing Future Applications," In Proc of 23rd National Information Systems Security Conference(NISSC), Baltimore, Oct.2000, pp.115-125.

[10] M.J. Moyer and M.Ahamad, "Generalized Role-Based Access Control," In proc of IEEE Int'l Conf. on Distributed Computing Systems(ICDSC2001), Mesa, April 2001, pp.391-398.

[11] M.J.Covington, et al., "A Context-Aware Security Architecture for Emerging Applications," In *Proc. of the 18th Annual Computer Security Applications Conferences (ACSAC'02)*, 2002., pp. 249-258.

[12] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E.Youman., "Role-based access control models," IEEE Computer, Vol. 29, No. 2, February 1996, pp. 38- 47.

[13] http://csrc.nist.gov/rbac/NIST.

[14] R.L.Keeney and H.Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley & Sons, New York, NY, 1976.

[15] M. Chung and V. Honavar, "A Negotiation Model in Agent-mediated Electronic Commerce," In *Proc. of the IEEE International Symposium on Multimedia Software Engineering*, Taipei, Dec. 2000, pp. 403-410.
[16] http://www.warentest.de/
[17] 김효남, 박용, "유비쿼터스 컴퓨팅환경에서 상황인식 미들웨어 설계," 한국컴퓨터 정보학회 논문지, 10권, 5호, 2005, pp.115-122.
[18] 남승좌, 박석, "유비쿼터스 컴퓨팅 환경의 역할 기반 접근제어에서 발생하는 상황충돌," 정보보호학회논문지, 15권 2호, 2005, pp.37-52.
[19] 심춘보, 신용원, "유비쿼터스 컴퓨팅환경에서 상황인식을 지원하는 컨텍스트 미들웨어 개발," 한국지능정보시스템학회논문지, 11권, 1호, 2005, pp.53-63.
[20] 임신영, 허재두, "상황인식 컴퓨팅 응용 기술 동향," 전자통신동향분석, 19권, 5호, 2004, pp.31-40.

〈著 者 紹 介〉

정 목 동(Chung, Mokdong)
정회원

1981년 2월 경북대학교 컴퓨터공학과 공학사
1983년 2월 서울대학교 컴퓨터공학과 공학석사
1990년 8월 서울대학교 컴퓨터공학과 공학박사
1984년~1985년 금성반도체(주) 연구소 연구원
1985년~1996년 부산외국어대학교 컴퓨터공학과, 부교수
1999년~2000년 미국 Iowa State University, 방문 교수
1996년~현재 부경대학교 컴퓨터공학과, 교수
관심 분야: Computer Security for Application, RFID Security, 물류 IT 기술, Intelligent Agent