사물인터넷 정보 보호 기술 동향

최 종 석 · 김 호 원

부산대학교

Ⅰ. 개 요

다양한 이기종 통신 방법과 센싱 빅데이터를 활용한 서비스 개발에 대한 관심이 높아지면서 사물인터넷 기반의 서비스 개발 및 제공에 대한 연구[1]~[4]가 활발하게 이루어지고 있다. 사물인터넷은 디바이스, 통신, 플랫폼, 서비스 계층으로 나누어진다. 디바이스 계층은 센서와 같은 디바이스를 통해서 데이터를 수집하는 역할을 수행하며, 통신계층은 수집된 데이터를 플랫폼으로 전달하고, 데이터에 대한 신뢰성을 보장해주기 위한 기능을 제공한다. 플랫폼은 데이터, 서비스, 사용자와 같은 리소스를 관리하고, 다양한 서비스에서 활용가능한 데이터를 제공해주기 위한 인터페이스를 가진다. 서비스 계층은 플랫폼에서 제공해주는 데이터를 기반으로 하여 사용자 중심의 사물인터넷 서비스를 제공한다.

사물인터넷 분야에서 주로 사용되고 있는 주요 플랫폼은 크게 oneM2M[5], AllJoyn[6], IoTivity[7], LWM2M[8]이다. oneM2M은 oneM2M Alliance에서 개발하였으며, M2M 통신을 하나로 통합하는 표준을 제시하기 위한 노력을 하였다. 이에 따라 oneM2M을 이용한 공통 M2M 서비스 계층을 제공하기 위한 연구[9]~[11]를 활발하게 진행하고 있다. AllJoyn은 AllSeen Alliance에서 개발하였으며, P2P 통신을 이용하여 통신에 제약을 받지 않고, 다양한 서비스를 개발하기 위한 많은 연구[12],[13]를 진행하고 있다. 그 외에도 IoTivity와 LWM2M 플랫폼에 기반한 많은 연구[14],[15]들이 진행되고 있지만, 사물인터넷 표준 플랫폼의 보안 기술에 대한 분석 및 연구는 미흡하다.

사물인터넷 플랫폼은 데이터, 서비스, 사용자 리소스를 관리할 뿐만 아니라, 플랫폼에서 제공되는 리소스의 신뢰성을 높이고, 개인정보를 보호하기 위한 플랫폼 보안 기능을 제공해준다. 특히 IoT-A에서 제안하는 사물인터넷 보안기능인 인증, 인가, 식별 등은 사물인터넷 보안을 위한 주요 보안 기능에 포함되며, 각 플랫폼에 맞는 보안기능을 정의할 필요가 있다.

본 논문에서는 oneM2M[1], AllJoyn[2], IoTivity[3], LWM2M[4]과 같은 주요 사물인터넷 플랫폼의 보안 기술을 검토하고, 사물인터넷 플랫폼 보안기능을 보안주체, 인증, 인가, 키 관리 측면에서 비교분석한다. oneM2M을 제외한 타 플랫폼은 인증 및 키 관리 측면에서 대칭키를 플랫폼 기능을 위해서 명시적으로 사용하지 않아서 대칭키 사용에 따른 이슈를 최소화할 수 있는 반면, 경량 디바이스에 대한 보안 이슈를 고려하여야 한다. oneM2M은 인증을 위해서 대칭키, 인증서, GBA (Generic Bootstrapping Architecture) 기반의 인증을 제공하며, 대칭키 사용에 따른 키 관리를 위해서 원격 보안 준비 프레임워크(Remote Security Provisioning Framework)를 통해서 원격으로 키를 배치할 수 있는 기능을 제공한다. 인가 측면에서는 oneM2M만 ACL, RBAC, ABAC을 모두 고려하고 있으며, 타 플랫폼은 ACL만 고려하고 있다.

Ⅱ. oneM2M 보안 기술

본 장에서는 oneM2M의 개요와 보안기술에 대해서 살펴본다.

2-1 oneM2M 개요

본 절에서는 oneM2M의 전체 아키텍쳐를 살펴본다. oneM2M은 다양한 서비스의 요구사항을 만족시킬 수 있는 사물인터넷에서의 공통 플랫폼을 정의하고, 타 플랫폼과의 상호동작(Internetworking)을 표준화하였다. 다양한 응용 간의 호환을 위한 인터페이스를 정의하여 종래의 수직적인 형태의 사물인터넷 플랫폼에서 벗어나 수평적인 플랫폼을 구성하여 사물인터넷 플랫폼의 파편화 방지, 개발 및 운용 비용을 감소할 수 있다. 스마트 홈, 스마트 카, 에너지, 헬스케어, 엔터프라이즈, 공공 서비스와 같은 7개 산업 분야의 use case를 반영하여 요구사항을 도출하고, 핵심 기능(데이터수집 및 보고 기능, 기기의 원격 제어, 연결성 유지, 보안 및 프라이

버시 기능 등)과 인터페이스를 정의하였다.

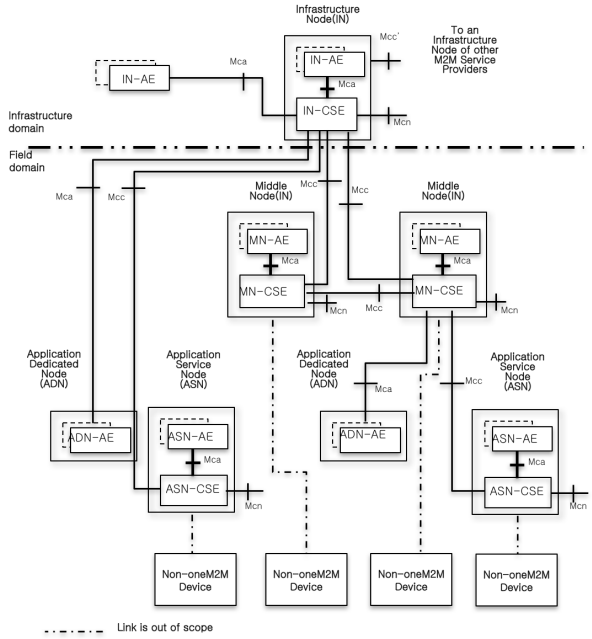
oneM2M의 개체는 User/End-User, application service provider, M2M service provider, network operator로 구성된다. User/End-User는 M2M 솔루션을 사용하는 개인 또는 기업을 의미하며, application service providers는 M2M 서비스를 제공하는 제공 주체를 의미한다. M2M service provider는 application service provider에게 M2M 공통 서비스를 제공하는 주체이며, network operator는 M2M service provider에게 네트워크를 제공하는 주체이다.

oneM2M은 여러 개의 노드(Node) 연결되어 하나의 인프라를 형성하며, 하나의 노드는 AE(Application Entity)와 CSE (Common Service Entity), NSE(Network Service Entity)로 구성된다. 기능적인 관점에서 AE는 M2M 서비스를 제공하기 위한 애플리케이션 기능 로직을 담당하며, CSE는 AE를 위한 12개의 공통 서비스기능을 제공한다. NSE는 CSE에게 네트워크 장치 관리 및 서비스 등을 제공하고, 각각의 개체(entity)는 참조점(reference point)를 통해서 상호동작한다. 이때, 참조점은 CSE와 AE, CSE 간의 연결을 의미하며, 실제 통신을 위한 바인딩 프로토콜(binding protocol)에 매핑되어 통신을 수행한다. [그림 1]은 oneM2M의 전체 구조를 보여준다. [그림 1]에서 Mca는 CSE-AE간의 통신, Mcc는 CSE-CSE간의 통신, Mcn은CSE와 NSE 간의 통신, Mcc'는 다른 Infrastructure Domain CSE와의 통신을 나타낸다.

CSE는 Registration, Discovery, Security, Group Management, Data Management, Subscription & Notification, Device Management, Application and Service LayerManagement, Communication Management/Delivery Handling, Location, Network Service Exposure/Triggering, Charging&Accounting과 같은 공통 서비스 기능(common service function)을 제공하며, ROA(Resource-Oriented Architecture)에 기반하여 CRUDN(Create, Retrieve, Update, Delete, Notify) 연산을 12개의 공통 서비스 기능에 대해 제공한다.

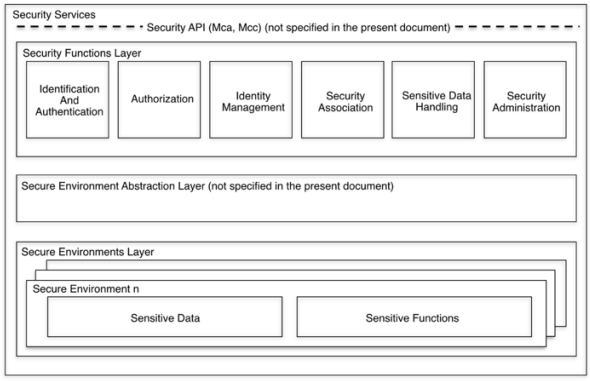
2-2 보안기술

oneM2M의 보안 아키텍처는 Security Function Layer, Secure Environment Abstraction Layer, Secure Environments Layer로 총 3개의 계층으로 구성된다. Security Functions Layer는 one-



[그림 1] oneM2M 전체 구조도

M2M 3개의 보안 계층 중에 가장 중요한 계층으로 식별(identifiction) 및 인증(authentication), 인가(authorization), 식별자 관리(identity management), 보안 연관(security association), 민감 데이터 관리(sensitive data handling), 보안 관리(security administration) 기능들을 제공한다. Secure Environment Abstraction Layer는 Security Function Layer에서 제공하는 기능을 만족하기 위한 핵심 알고리즘을 제공해주며, 암호알고리즘 등이 이에 포함될 수 있다. Secure Environments Layer는 민감 데이터



[그림 2] oneM2M 보안 구조도

등에 접근하기 위해서 추가적으로 필요한 요소를 정의하며, 스마트카드, 유심(USIM)카드 등이 해당 계층에 포함된다.

식별 및 인증을 위해 oneM2M은 AE, CSE, NSE 등에 각각 식별자를 부여하고, 이에 대한 유효성을 판단한다. oneM2M에서 사용하는 식별자는 AE-ID, CSE-ID, App-ID, M2M-Node-ID, M2M-Sub-ID, M2M-Request-ID, M2M-Ext-ID, UNetwork-ID, Trigger-Recipient-ID, M2M-Sev-ID, SRole-ID, M2M-Service-Profile-ID 등이 있으며, 이는 각각의 사용용도와 특성에 따라서 글로벌 또는 로컬 유일성(uniqueness)을 가진다.

원격 보안 준비 프레임워크(Remote Security Provisioning Framework: RSPF)는 oneM2M 엔터티 간의 식별 및 인증을 위한 사전 보안 정보(security credential)의 원격 배포를 위한 프레임워크로 사전 분배된 대칭키 기반 프레임워크(Pre-Provisioning Symmetric Enrolee Key: RSPF), 인증서 기반 프레임워크(Certificate-Based RSPF), GBA 기반 프레임워크(GBA-Based RSPF)가 있다. 각각의 구조는 개체에 대한 인증방법에 따라서 나눠지며 MEF(M2M Enrolment Function)에 등록을 통해서 사전 보안 정보가 배포되고, 사전 분배된 대칭키 기반의 원격 보안 준비 프레임워크는 개체의 식별 및 인증 정보로 사전에 분배된 대칭키를 사용하며, (D)TLS-PSK 핸드쉐이크를 통해서 상호 인증 및 보안 채널을 성립한다. 인증서 기반의 원격 보안 준비 프레임워크는 개체의 식별 및 인증정보로 비밀키로 서명된 X.509 인증서를 사용하며, GBA 기반의 원격 보안 준비 프레임워크는 이동 통신의 3GPP 또는 3GPP2의 대칭키를 사용하여 인증을 수행하여 개체 식별 및 인증과정을 대체하고, (D)TLS를 통해 다른 개체와의 상호인증을 수행한다.

보안 연관 성립 구조는 원격 보안 준비 프레임워크를 통해서 분배된 키 또는 인증서를 이용하여 두 개체간의 상호 인증을 수행한다. 공통적으로 자격 증명 설정(credential configuration), 연관 설정(association configuration), 연관 보안 핸드쉐이크(association security handshake) 단계로 구성된다. 자격 증명 설정 단계는 각 구조에 맞는 자격 증명 요소를 설정하고, 자격 증명 요소로는 사전분배된 대칭키, 인증서, 마스터 자격 증명(Master Credential)이 존재하며, 이를 통해 개체간의 상호 인증을 수행한다. 연관 설정 단계는 식별 및 상호 인증에 필요한 엔터티의 식별자를 설정하고, 인증서 기반의 보안 연관 성립 구조에서는 검증과정에서 필요한 인증서 이름, RoT(Root of Trust)의 정보가 설정한다. 연관 보안 핸드쉐이크는 개체간의 식별 및 인증하고, 보안 채널을 성립한다.

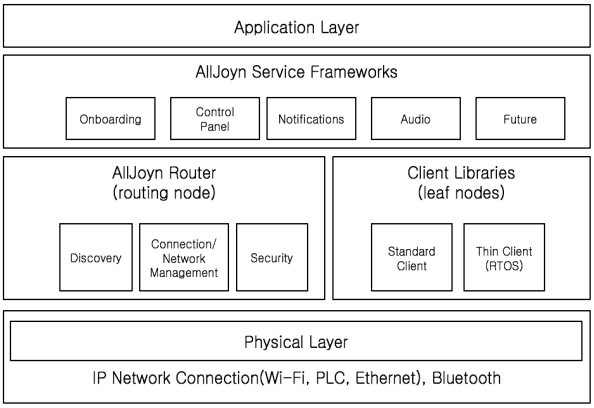
인가 기능은 인증된 개체(entity)가 리소스에 접근할 때, PEP, PDP, PRP, PIP 간의 요청 및 응답을 통해 접근 여부를 결정한다. PEP(Policy Enforcement Point)는 oneM2M 리소스에 대한 접근 요청을 받고, 접근 결정에 따라 리소스에 연결하도록 하는 역할을 수행한다. PDP(Policy Decision Point)는 PRP, PIP와의 요청/응답을 통해 접근 요청에 해당되는 인가 정책들과 속성들을 가져오고, 접근 제어 리스트(access control list), 역할 기반 접근 제어(role-based access control), 속성 기반 접근 제어(attribute-based access control) 등의 접근 제어정책을 이용하여 리소스에 대한 접근 여부를 결정하는 역할을 수행하며, 접근 결정 결과를 PEP에게 전송한다. PRP(Policy Retrieval Point)는 접근 제어 정책을 관리하며, PDP의 접근 제어 정책요청에 대한 응답(policy response)를 전달하고, PIP(Policy Information Point)는 인가 정책을 평가하는데 필요한 속성들을 관리하며, PDP의 요청에 따른 응답을 전송한다.

Ⅲ. AllJoyn 보안 기술

본 장에서는 AllJoyn에 대한 개요 및 보안 기술에 대해서 살펴본다.

3-1 AllJoyn 개요

[그림 3]은 AllJoyn의 전체 구조도를 보여준다. AllJoyn은 Application Layer, AllJoyn Service Frameworks, AllJoyn Router, Client Libraries, Physical Layer로 구성된다. Physical Layer는 Wi-Fi, PLC, Ethernet 등의 IP기반 통신과 Bluetooth 통신 모듈 물리적인 인터페이스를 가지는 계층이다. 하나의 AllJoyn 노드는 다수의 AllJoyn 앱과 하나의 AllJoyn 라우터로 구성되는데, AllJoyn 앱을 개발하기 위한 기능을 제공하는 것이 Client Libraries 계층이며, AllJoyn 라우터가 다른 AllJoyn 라우터와 통신을 하기 위해 필요한 기능을 제공하는 것인 AllJoyn Router 계층이다. AllJoyn Service Frameworks는 실제 AllJoyn 앱이 사용하기 위한 Onboarding, Control Panel, Notifications, Audio 등의 기능을 제공해준다. Application Layer는 AllJoyn



[그림 3] AllJoyn 전체 구조도

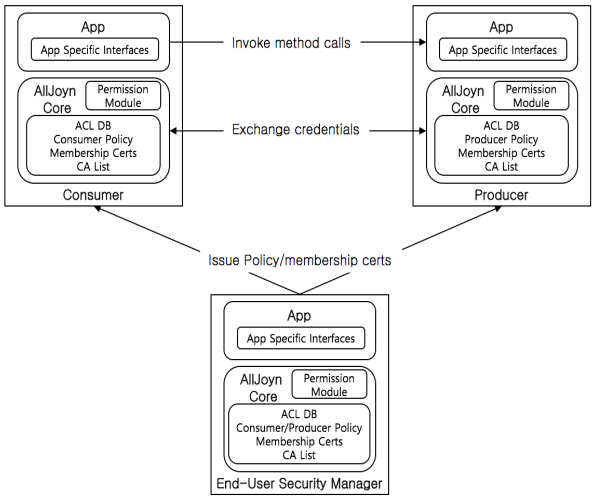
Service Frameworks 계층에서 제공되는 기능들을 이용하여 생성된 AllJoyn 앱들이 구동하는 계층이다.

AllJoyn은 근거리 기반의 기기 간 P2P(Peer-To-Peer) 통신이며, 중계서버를 사용하지 않고, 디바이스 간의 통신을 사용한다. Bluetooth나 Wi-Fi 등의 물리적인 통신방식 위에 소프트웨어 프레임워크로 개발된 통신을 이용하기 때문에, 하드웨어에 의존적이지 않다. AllJoyn은 디바이스 간의 세션 연결을 위해서 RMI(Remote Method Invocation)방식의 D-Bus를 이용한다.

3-2 보안기술

[그림 4]는 AllJoyn 보안 구조도를 보여준다. AllJoyn Security 시스템에서는 서비스를 제공하는 생산자(producer), 서비스를 실제 사용하는 소비자(consumer), 보안 기술을 실행하는 Security Manager로 구성된다. 허가모듈(permission module)은 접근제어와 관련된 데이터베이스와 오브젝트 및 인터페이스에 대한 접근권한을 관리한다. 접근 제어는 기본적으로 접근 제어 리스트(Access Control List: ACL)를 사용하며, ACL DB에 저장된다.

인터페이스 레벨에서의 보안을 지원하여 인증 및 암호화를 사용하도록 응용 프로그램 인터페이스에 태그를 지정한다. Security Manager는 Key management 및 Permission Rule 설정을 지원하는 서비스로서, 개발자에 의해 정의된 애플리케이션 메니페스트 템플릿을 통해 애플리케이션과 상호작용을 할 수 있는 최종 사용자의 권한 부여를 할 수 있도록 ACL



[그림 4] AllJoyn 보안 구조도

(Access Control Lists)로 이루어진 메니페스트를 생성한다. 인증 GUID는 인증을 위해 애플리케이션에 할당된 GUID를 의미하는데, 키 저장소에 저장되고 애플리케이션의 long-term ID를 제공한다. 일반적으로 GUID는 하나의 애플리케이션과 연결되어 관련 애플리케이션의 그룹이 하나의 키 저장소를 공유할 시에 동일한 인증 GUID를 공유한다. 특히 인증 GUID는 저장 및 관련 애플리케이션에 대한 인증 및 암호화 키에 엑세스하는 매핑 키로써 사용된다.

마스터키(master secret)는 인증된 디바이스 애플리케이션 간에 공유되는 키로써, 각 디바이스 애플리케이션은 동일한 마스터키 독립적으로 생성한다. 암호화 키는 두 개의 피어 애플리케이션 간의 지점 간 데이터 트래픽을 암호화하는데 사용되며, 모든 애플리케이션의 연결에 대한 세션키가 별도로 생성된다. AllJoyn에서는 X.509 인증서를 기반으로 ID 인증서(identity cert), 멤버십 인증서(membership cert)를 구성된다. ID 인증서는 디바이스가 가지고 있는 GUID와 공개키에 대한 인증서이며, 멤버십 인증서는 보안 그룹 멤버십(security group membership)에 대한 인증서이다.

Ⅳ. IoTivity 보안 기술

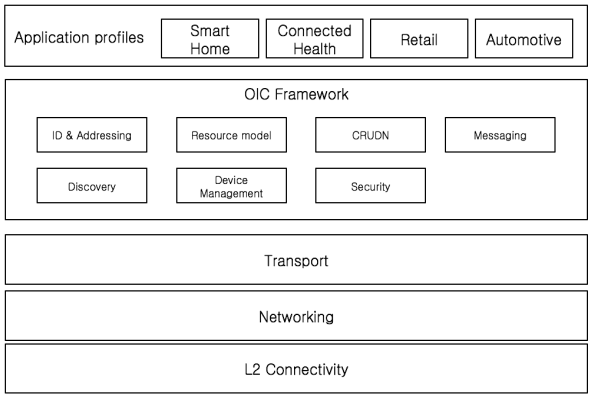
본 장에서는 IoTivity의 개요 및 보안기술에 대해서 살펴본다.

4-1 IoTivity 개요

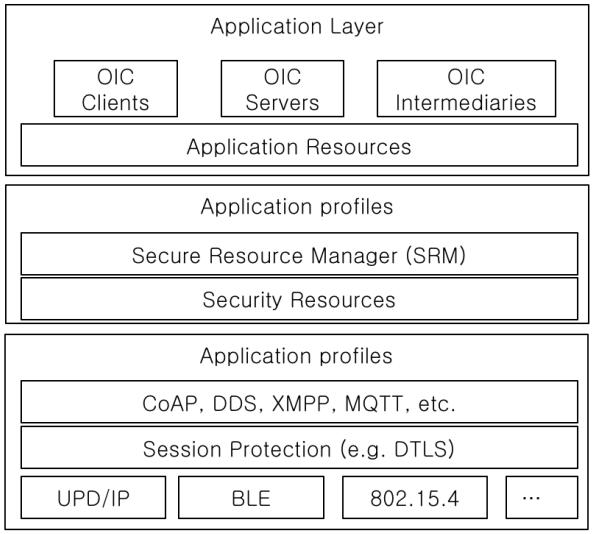
[그림 5]는 IoTivity 전체 구조도를 보여준다. IoTivity는 Application profiles, OIC Framework, Transport, Networking, L2 Connectivity 계층으로 구성된다. Application profiles 계층은 Smart Home, Connected Health, Retail, Automotive 등의 다양한 응용애플리케이션이 수행된다. OIC Framework는 Application profiles 계층에서 수행되는 애플리케이션이 요구하는 기능을 제공해주는 계층으로 ID& Addressing, Resource model, CRUDN, Messaging, Discovery, Device Management, Security 기능을 제공한다. Transport 계층은 특정 QoS(Quality of Service) constraints를 가지는 단대단(end-to-end) 전송기능을 제공한다. Networking 계층은 인터넷과 같은 네트워크상에서 디바이스 간의 데이터 교환 기능을 제공하며, L2 Connectivity 계층은 물리 계층과 데이터 링크 계층 간의 연결을 제공한다.

4-2 보안기술

[그림 6]은 IoTivity 보안 구조도를 보여준다. IoTivity 디바이스는 서비스를 사용하는 OIC Client, 서비스를 제공하는 OIC Server, 서비스 중계 역할을 수행하는 OIC Intermediaries로 구성된다. 각각의 디바이스는 Application Resources를 가지고 있으며, 각 애플리케이션에 대한 Application profiles이 정의된다. 이 때, Application profiles에는 접근제어를 수행하는 SRM(Secure Resource Manager)과 Security Resources 가 있으며, 보안 채널을 형성하기 위한 Session Protection 등이 포함될 수 있는데, 이는 각각의 애플리케이션 특성마다 특화된



[그림 5] IoTivity 전체 구조도



[그림 6] IoTivity 보안 구조도

Application profiles로 정의된다.

OIC Client는 OIC Resource에 대한 접근 요청 Action을 수행하며, Resource 에 대한 접근 제어는 OIC Server의 접근 제어 모델에 따라 수행된다. OIC Client는 Resource를 소유하고 있는 OIC Server와 네트워크 연결을 수성립하고, Connectivity abstraction 계층은 추상화를 통해 다양한 연결 옵션을 제공한다. OIC 디바이스를 식별하기 위해서 IoTivity에서는 Device ID를 사용한다. 네트워크 주소는 Device ID로 맵핑되며, 네트워크 주소를 통해 연결이 성립한다.

IoTivity에서 보안 정책은 Device ID를 이용하여 기술되는데, (D)TLS를 이용하여 보안 채널을 생성하고, 로컬플랫폼에 저장되어 있는 암호키를 이용하여 상호 인증 및 보안 통신을 수행한다. OIC Client가 Resource에 접근하기 위해서 OIC Server는 OIC Client에 대한 식별 및 인증을 수행하고, SRM은 접근제어모델에 따라 접근 제어를 수행한다. SRM이 접근제어를 수행하기 위해서 Security Resource에 정의된 모델을 참조하는데, Security Resource를 정의하기 위해서 ACL, 서비스(Service), 자격증명(Credential)에 대한 각각의 오브젝트를 정의한다. 이 때, ACL 오브젝트는 Subject, Resource, Permission을 포함하며, 서비스 오브젝트는 Device ID, Svc Type, CredID를 포함하고, 자격증명 오브젝트는 Device ID, Cred Type, Private Data로 구성된다. 정의된 세 개의 오브젝트를 연결하

기 위해서 ACL의 subject와 서비스 오브젝트의 Device ID를 연결하고, 서비스의 Cred ID와 자격증명 오브젝트의 Device ID를 연결한다. Session Protection을 위해서는 위에서 정의된 자격증명 오브젝트의 Private Data를 (D)TLS와 연계하여 상호인증 및 보안채널을 형성한다.

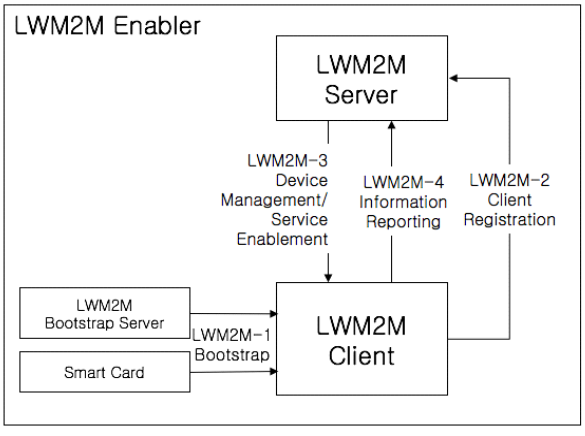
Ⅴ. LWM2M 보안 기술

본 장에서는 LWM2M의 개요 및 보안기술에 대해서 살펴본다.

5-1 LWM2M 개요

[그림 7]은 LWM2M의 전체 구조도를 보여준다. LWM2M은 LWM2M Bootstrap 서버, 스마트카드, LWM2M 서버/클라이언트로 구성된다. LWM2M Enabler는 LWM2M 서버와 LWM2M 클라이언트 요소를 기술한다.

LWM2M을 사용하기 위해서 최초에는 선택적으로 Bootstrap단계를 수행할 수 있다. LWM2M의 부트스트랩 단계는 LWM2M 서버의 일부 정보 또는 상호 인증 등을 간소화하기 위한 파라미터를 사전에 LWM2M 클라이언트에 기술하는 과정이다. LWM2M Bootstrap 서버와 스마트카드를 이용해 LWM2M 부트스트랩을 수행한다. LWM2M 클라이언트는 LWM2M 서버에 클라이언트 등록을 수행하고, LWM2M 서버는 LWM2M 클라이언트를 관리하기 위해서 Device Management, Service Enablement 기능을 제공한다. 또한 LWM2M 클라이언트는 LWM2M 서버의 리소스 관리를 돕기 위해서 Information Reporting 기능을 제공한다.



[그림 7] LWM2M 전체 구조도

LWM2M에서 M2M User가 M2M 서비스 제공자에게 서비스를 제공받고, M2M 서비스 제공자가 LWM2M 서버와 M2M 응용서비스를 제공한다. LWM2M 서버는 네트워크를 통해서 LWM2M 클라이언트에 접근할 수 있다. 또한 LWM2M 서버를 M2M 서비스 제공자가 아닌 네트워크 서비스 제공자가 LWM2M 서버를 운영하고, LWM2M 서버의 인터페이스를 M2M 응용서비스에 제공하고, LWM2M 서버/클라이언트 간의 통신을 수행한다.

5-2 보안기술

LWM2M 기반의 리소스에 대한 접근제어는 ACO(Access Control Object)에 의해서 결정된다. ACO는 LWM2M 서버가 LWM2M 클라이언트의 객체(Object)에 접근할 때, 적절한 권한을 가졌는지를 확인하기 위해서 사용하며, LWM2M에서 ACO를 기술하기 위해서 XML 스키마를 이용한다. [그림 8]은 ACO의 주요 스키마를 보여준다. Resource 스키마로 구성되며, Operations 태그는 Read/Write/Execution에 대한 권한설정을 할 수 있고, Range Enumeration을 통해서 설정 가능한 범위를 기술한다.

LWM2M 기반의 리소스에서 ACO를 작성할 때, [그림 8]의 주요 스키마 이외에도 Object ID, Item ID 등을 포함하고, Resources 스키마에 다수의 ItemID를 포함하여 하나의 Object 밑에 있는 Item에 대한 접근제어객체(Access Control Object)를 생성한다. 실질적으로, LWM2M 서버가 LWM2M 클라이언트에게 LWM2M 기반의 리소스를 요청하면, 정의된 접근제어객체에서 LWM2M 서버가 Object/Item에 대한 권한이 있

```
<xs:element name="Resources">
  <xs:element name="Name" type="xs:string"/>
  <xs:element name="Operations"/>
  <xs:element name="MultipleInstances"/>
  <xs:element name="Mandatory"/>
  <xs:element name="Type"/>
  <xs:element name="RangeEnumeration" type="xs:string"/>
  <xs:element name="Units" type="xs:string"/>
  <xs:element name="Description" type="xs:string"/>
</xs:element>
```

[그림 8] LWM2M 보안 스키마

는 지를 확인하고, 기술된 접근제어객체의 Operation이 R/W/RW/E 인지에 따라서 LWM2M 서버에 접근권한을 할당한다.

Ⅵ. 사물인터넷 보안 기술 비교분석

본 장에서는 oneM2M, AllJoyn, IoTivity, LWM2M의 보안기술에 대해서 비교분석한다. <표 1>은 주요 사물인터넷 플랫폼 보안기술을 비교한 것을 보여준다.

oneM2M은 CSE가 보안을 제공하며, ACL/RBAC/ABAC에 대한 인가가 정의되어 있으며, 개체인증을 위해 대칭키/인증서/MAF 기반의 SAEF(Security Association Establishment Framework)를 사용하고, 대칭키 관리를 위해 RSPF(Remote Security Provisioning Framework)를 사용한다. AllJoyn은 AllJoyn Core Library에서 제공되는 보안기능을 이용하여 AllJoyn Application이 보안 기능을 제공한다. Policy 기반의 ACL을 통해서 인가기능을 수행하며, 인증서 기반의 사용자 인증을 수행하며, 대칭키는 Security Manager와 Application 간에 사용될 수 있지만, 키 관리는 제공하지 않는다. IoTivity는 Resource Layer의 SRM이 제공하는 보안기능을 통해서 IoTivity 서버/클라이언트 각각 보안기능을 제공한다. ACL 기반의 인가기능을 제공하며, 개체인증을 위해서 DTLS에 의존하고, 키 관리는 제공하지 않는다. LWM2M은 LWM2M 클라이언트가 보안을 제공하며, XML 스키마 기반으로 ACL을 작성하여 LWM2M 서버에 대한 인가기능을 수행하고, LWM2M 서버에 대한 인증은 DTLS에 의존하며, LWM2M의 내부 대칭키를 사용하지 않기 때문에 키 관리 기능을 제공하지 않는다.

사물인터넷 플랫폼에서 oneM2M이 인가, 인증, 키 관리 측면에서 높은 보안성을 제공하고 있으며, AllJoyn은 ACL 기반의 접근제어와 인증서 기반의 인증만을 제공한다. 따라서 AllJoyn은 인증을 위해서 인증서를 사용하여야 하며, 이는 경량 디바이스를 지원할 때, 기능의 제약성이 생길 수 있다. IoTivity 및 LWM2M과 같이 (D)TLS에 의존하여 인증기능 제공하고 있으며, 이는 잠재적 위협요소로 작용할 수 있다.

<표 1> 사물인터넷 플랫폼 보안기술 비교

	보안개체	인가	인증	키 관리
oneM2M	CSE	ACL/RBAC/ABAC	SAEF	RSPF
AllJoyn	Application	ACL	인증서	×
IoTivity	Server/client	ACL	DTLS	×
LWM2M	Client	ACL	DTLS	×

Ⅶ. 결 론

웨어러블 디바이스와 스마트폰의 보급에 따라 개인당 디바이스 보유량이 증가하였다. 이에 따라 다양한 이기종 디바이스에서 생성 및 수집되는 데이터를 이용한 사용자 중심의 서비스를 개발하는데 관심이 높아지고 있다. 이기종 디바이스를 서로 연결하고, 이기종 데이터를 이용하여 새로운 서비스를 생성하는 것이 가장 적합한 사물인터넷 분야에 대한 관심이 높아졌으며, 특히 데이터/사용자/서비스에 대한 관리 및 분석을 주요 역할로 하는 사물인터넷 플랫폼에 중요성을 더욱 높아졌다. 그러나 사용자 중심의 서비스를 제공하기 위해서는 사물인터넷 플랫폼이 사용자와 아주 밀접한 프라이버시 데이터를 관리해야만 하며, 이러한 보안기능에 대한 연구가 활발하게 진행되고 있다

본 논문에서는 사물인터넷 분야의 주요 플랫폼으로 사용되고 있는 oneM2M, AllJoyn, IoTivity, LWM2M 플랫폼의 보안기술에 대해서 살펴보고, 각각의 플랫폼의 보안기술을 인증/인가/키 관리 측면에서 비교분석하였다. 인증 관점에서는 oneM2M은 보안 연관 구조(Security Association Establishment Framework)를 이용하여 플랫폼에 맞게 다양한 인증 방법을 제공한다. AllJoyn은 인증서 기반의 인증을 제공하며, IoTivity와 LWM2M은 DTLS에 의존한 인증만을 제공하고 있다. 인가 측면에서는 oneM2M은 ACL/RBAC/ABAC의 다양한 접근제어모델을 권장하고 있으며, 나머지 플랫폼은 ACL만을 채택하였다. 키 관리 측면에서는 oneM2M을 제외한 타 플랫폼은 대칭키를 플랫폼 기능을 위해서 명시적으로 사용하지 않으며, 따라서 키 관리 기능을 제공하지 않는다. oneM2M은 원격 보안 준비 프레임워크(Remote Security Provisioning Framework)를 통해서 대칭키를 원격으로 관리할 수 있다.

참 고 문 헌

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. "Internet of things (IoT): A vision, architectural elements, and future directions", *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, Sep. 2013.

[2] L. Atzori, A. Iera, and G. Morabito. "The internet of things: A survey", *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, Oct. 2010.

[3] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton. "Smart objects as building blocks for the internet of things", *IEEE Internet Computing*, vol. 14, no. 1, pp. 44-51, Jan. 2010.

[4] G. B. Flavio, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things", In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13-16, Aug. 2012.

[5] OneM2M Alliance, "Onem2m: Standards for m2m and the Internet of Things", 2014.

[6] Allseen. Alliance, "AllJoyn Framework.", 2015.

[7] OIC, "IoTivity 1.1.1", 2016.

[8] L. Tian, "Lightweight m2m (oma lwm2m)", OMA Device Management Working Group (OMA DM WG), Open Mobile Alliance, 2012.

[9] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. S. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M", *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20-26, Jun. 2014.

[10] M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira, "Toward semantic interoperability in oneM2M architecture", *IEEE Communications Magazine*, vol. 53, no. 12, pp. 35-41, Dec. 2015.

[11] S. Husain, A. Kunz, J. S. Song, and T. Koshimizu, "Interworking architecture between oneM2M service layer and underlying networks", *In 2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 636-642, Dec. 2014.

[12] M. Villari, A. Celesti, M. Fazio, and A. Puliafito. "Alljoyn lambda: An architecture for the management of smart environments in iot", *InSmart Computing Workshops (SMARTCOMP Workshops), 2014 International Conference on*, pp. 9-14, Nov. 2014.

[13] Y. Wang, L. Wei, Q. Jin, and J. Ma. "AllJoyn based direct proximity service development: Overview and prototype", In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pp. 634-641, Dec. 2014.

[14] C.-H. Lee, Y. H. Lai, "Design and implementation of a universal smart energy management gateway based on the Internet of Things platform", In *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 67-68, Jan. 2016.

[15] S. Rao, D. Chendanda, C. Deshpande, and V. Lakkundi, "Implementing LWM2M in constrained IoT devices", In *Wireless Sensors (ICWiSe), 2015 IEEE Conference on*, pp. 52-57, Aug. 2015.

≡ 필자소개 ≡

최 종 석

2011년 2월: 동명대학교 정보보호학과 (공학사)
2013년 2월: 부산대학교 전기전자컴퓨터공학과 (공학석사)
2013년～현재: 부산대학교 전기전자컴퓨터공학과 박사과정
[주 관심분야] IoT, 스마트그리드 보안, RFID/USN 보안, PKC 암호, 임베디드 시스템 보안

김 호 원

1993년 2월: 경북대학교 전자공학과 (공학사)
1995년 2월: 포항공과대학교 전자전기공학과 (공학석사)
1999년 2월: 포항공과대학교 전자전기공학과 (공학박사)
2008년～현재: 부산대학교 정보컴퓨터공학부 부교수
[주 관심분야] IoT, 데이터마이닝, 스마트그리드 보안, RFID/USN 보안, PKC 암호, VLSI, 임베디드 시스템 보안