# 사물인터넷 서비스 접근제어를 위한 리소스 서비스 관리 모델 구현

## Resource management service model implemented for the Internet of Things services access control

김진보*, 김미선**, 서재현**

(Jin-Bo Kim*, Mi-Sun Kim**, Jae-Hyun Seo**)

**요 약**

사물인터넷 기술을 이용한 서비스 형태는 셀 수 없이 많으며, 현재도 여러 기관에서 다양한 서비스를 제공하기 위해 새로운 기술이나 프로토콜을 만들어 내고 있다. 본 논문은 사물인터넷 서비스 접근제어를 위한 시스템에서 효율적인 서비스 제공을 위한 리소스 서비스 모델을 설계하고 구현한다.

사용자가 접근하고자 하는 리소스 서비스를 LCRS(Left Child-Right Sibling) 트리를 이용하여 모델링하고, 리소스 서비스 토큰을 이용하여 서비스에 대한 접근 제어를 수행할 수 있다.

■ 중심어 : 사물 인터넷, 접근 제어, 리소스 서비스, 서비스 토큰

**Abstract**

Many countless services form using the Internet technology of things, in order to provide a variety of services in the current also many institutions, have created a new technology and protocols. In this paper, we design the resource service model for the efficient service provided by the system for controlling access to the Internet services of the things, to implement.

The resources of the service that the user tries to access modeled using LCRS (Left Child-Right Sibling) tree, by utilizing the service token resource, it is possible to perform access control to services.

■ keywords : IoT, Access Control, Resource Service, Service Token

## Ⅰ. 서 론

웹 서비스 기술의 발전과 소프트웨어 플랫폼 기술, 클라우드 기술, 유·무선 통신 기술, 센서 기술 및 스마트 기기의 급속한 발전으로 인해 사물인터넷은 산업분야에서 재조명 받고 있다. 기존 인간 중심의 통신에서 모든 사물이 통신의 주체로 참여하는 시대가 본격화 되고 있는 것이다.

사물인터넷 기술을 이용한 서비스 형태는 셀 수 없이 많으며, 현재도 여러 기관에서 다양한 서비스를 제공하기 위해 새로운 기술이나 프로토콜을 만들어 내고 있다[3].

기존 연구[1]에서는 IoT 접근 제어 시스템에서 사용자가 리소스 서비스 접근을 요청하고 권한을 획득하는 데 있어서 리소스 서비스의 관리를 효율적으로 수행하기 위하여 리소스 서비스 모델을 제안하였다[2].

본 논문은 사물 간 연결을 통해 수집된 다양한 데이터를 기반으로 사물인터넷 서비스에 대한 관리 방안를 LCRS 리소스 모델 관리 방안을 구현하고, 리소스 서비스에 대한 접근제어를 위해 서비스 요청자의 리소스 서비스 토큰과 리소스 관리자가 등록한 리소스 서비스 정보를 이용한 리소스 서비스 URI 보안을 통해 서비스 접근제어 서비스 관리 모델을 구현하였다.

## Ⅱ. 본 론

### 1. 관련연구

#### 가. 사물 인터넷 서비스

* 학생회원, 목포대학교 정보보호기술학협동과정

** 정회원, 목포대학교 정보보호학과

이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2014R1A2A1A11053774).

접수일자 : 2016년 07월 11일

수정일자 : 2016년 08월 16일

게재확정일 : 2016년 09월 21일

교신저자 : 서재현 e-mail : jhseo@mokpo.ac.kr

서비스 관점의 사물인터넷이란 “센서가 탑재된 다양한 디바이스들을 통해 의미 있는 센싱 정보들이 지속적으로 수집되고, 직간접적으로 연결된 유무선 네트워크를 통해 빠르고 안정적으로 정보들이 송·수신되어 클라우딩 컴퓨팅 환경 및 빅 데이터 분석을 통해 자동화된 지능형 서비스가 제공되는 것”이다.

전 세계 사물인터넷 시장의 규모는 연평균 약44%의 고성장을 통해 2020년 6,000억 달러에 이를 것으로 시장 조사 기관인 비즈니스 인사이더는 전망하고 있다[9]. 특히 소프트웨어 및 서비스 시장의 비중이 높을 것으로 예측하고 있으며, 가트너는 2020년 전 세계 사물인터넷 시장 규모를 3,280억 달러로 그중 2,620억 달러 규모로 전체 매출의 80%가 서비스에서 발생할 것으로 전망하고 있다[11].

사물인터넷의 다양한 서비스는 여러 가지 센싱 장치를 이용하여 데이터를 수집하고, 분석한 결과를 이용하여 각각의 다양한 서비스를 제공할 수 있는 오픈 플랫폼 서비스를 나타내고 있다. 국내에서는 SKT의 스마트 팜 서비스, KT의 스마트 홈 서비스, LG U+의 지능형 차량 관제 서비스 등 점차적으로 개인에 직접적으로 영향을 미치는 B2C 서비스가 확산되고 있다. 이러한 서비스가 확산됨에 따라 기존의 센서와 같은 하드웨어는 사물인터넷 서비스가 활성화되기 위한 기반 기술로서 반드시 필요하지만, 부가가치 창출을 위해서 집중해야 할 핵심 기술은 소프트웨어 및 서비스라 할 수 있다[4].

### 나. 시스템 접근제어

시스템 접근제어 정책은 접근제어에 대한 객체기반과 주체기반으로 분류된다. 객체기반은 객체 관점에서 접근이 허용된 주체들에 대해 접근 권한을 테이블 구조로 기술하여 이를 기반으로 하는 접근제어이고, 주체기반은 주체가 접근하고자 하는 객체의 티켓을 부여 받아 이를 기반으로 하는 접근제어 정책이다[7].

사물인터넷 환경에서 접근제어 문제는 기존 인터넷 환경과의 차별성을 고려하여 접근하여야 한다. 첫째, 사물인터넷은 기존 인터넷 환경과 달리 짧은 시간동안 상호 작용이 일어나고 동일한 요청이 자주, 자발적으로 수행될 수 있다. 둘째, 사물인터넷에서 자원과 서비스에 대한 분석 및 인가는 매번 같은 요청에 대해서도 고정적이지 않고, 주변의 상황에 따라서 바뀔 수 있다. 따라서 개방되고 광범위한 컴퓨팅 환경에서의 확장성 문제와 장치들의 관리 문제 및 유연성 있는 권한 위임의 문제를 고려한 접근제어 기법이 필요하다[12][13].

#### (1) 강제적 접근제어

강제적 접근제어(Mandatory Access Control)는 객체에 포함된 정보의 비밀성과 이러한 비밀정보에 대하여 주체가 갖는 정형화된 권한에 근거하여 객체에 대한 접근을 제한하는 방법이다[4]. 임의적 접근제어 정책에 비해 객체의 소유자에 의하여 변경할 수 없는 주체와 객체간의 접근제어 관계를 정의하며 주체가 객체를 판독하고 그 내용을 다른 객체에게 복사하는 경우에 원래의 객체에 내포된 강제적 접근제어 제약사항이 복사된 객체에 전달된다. 강제적 접근제어 정책은 모든 주체 및 객체에 대하여 일정하며, 특정 주체 대 객체 단위로 접근 제한을 설정할 수 없다. 임의적 접근제어 기법의 보안 문제를 보완하기 위해서는 서비스 제공자 대신에 공통 플랫폼을 통한 사물인터넷 시스템 관리자와 같은 최고권한자가 서비스 사용자에 대한 권한을 설정해야 한다.

#### (2) 임의적 접근제어

임의적 접근제어(Discretionary Access Control)는 주체 또는 그들이 소속되어 있는 그룹들의 신분에 근거하여 객체에 대한 접근을 제한하는 방법이다. 접근제어는 객체의 소유자에 의해 임의적으로 이루어지므로 어떠한 접근 허가를 가지고 있는 한 주체는 임의의 다른 주체에게 자신의 권한을 넘겨줄 수 있다.

사물인터넷 서비스 관점에서 보면, 임의적 접근제어는 서비스 제공자가 서비스 사용자에 대한 권한을 직접 부여하는 방식이다.

#### (3) 역할기반 접근제어

시스템이 대규모화되고 다양해지면서 조직들은 그 조직 특성에 적합한 보안 정책을 필요로 하게 되었고, 보안 정책의 일관성 유지 및 보안 정책의 변경을 실제 시스템에 적용하기 위한 비용이 높아졌다.

임의적 접근제어와 강제적 접근제어는 규칙 수준에서 접근제어 서비스를 제공하기 때문에 위와 같은 요구를 반영하기 어렵다. 이를 해결하기 위해 역할 기반 접근제어(Role Based Access Control) 모델은 이를 해결하기 위해 특정 사용자를 관련된 모든 객체와 직접적으로 연관시키기 보다는 그와 관련된 객체들과 관계를 맺고 있는 역할과의 관계를 설정함으로써 접근제어를 관리하는데 있어서 보다 효율적인 보안 정책이다.

역할기반 접근제어는 비임의적 접근제어로 분류되며, 서비스 사용자에게 각각 역할을 부여하고 특정 역할마다 서비스에 대한 접근권한을 부여할 수 있는 방식으로 접근제어 작업을 단순화할 수 있다.

#### (4) 속성기반 접근제어

속성기반 접근제어(Attribute Based Access Control)는 사용자의 인증을 위해 사용자의 권한이 아닌 속성에 따라 접근제어를 수행하는 방법이다. 각 객체에 직접적인 접근 권한을 표현하는 속성 기반 접근제어 방식으로, 접근제어를 할 객체 내의 각 엘리먼트에 접근 권한에 대한 실제 값 또는 정보를 저장한다. 이러한 속성기반 접근제어는 수행 속도 관점에서는 효율적이며, 저장 공간 관점에서는 비효율적이다.

#### (5) 자격기반 접근제어

역할을 기반으로 접근제어를 수행하는 역할기반 접근제어의 경우 초기에 제어 규칙을 관리하기 위한 비용은 적게 든다. 하지만 많은 디바이스에 적용될 경우 규칙의 폭발적인 증가를 수용하기 어려운 단점이 있다.

속성기반 접근제어는 사용자의 속성을 바로 사용함으로써 규칙을 잘 다룰 수 있지만, 다수의 디바이스가 연동되는 사물인터넷 환경에서는 속성들을 동일하게 일치시켜야 하는 어려움이 있다.

속성 및 역할기반 접근제어 두 방법 모두 최소한의 권한만을 체크해 리소스를 허가하지 않으면 모든 권한을 검토한 후 재허가 해야 하는 단점이 있다. 또한 권한을 위임하는 것이 어려운 점 등 접근제어에 대한 운영이 유동적이지 못하다. 이러한 문제는 디바이스의 수가 증가 할수록 더 심각해진다.

표 3. 접근제어 기법 비교 분석

| 속성 | CapBAC[7] | Capability Token[14] | CapSG [1] |
|---|---|---|---|
| 위임 주체 | user | user | user |
| 위임 | yes | yes | yes |
| 폐기 | yes | yes | yes |
| 위임거절 | – | – | yes |
| 그룹 위임 | no | no | yes |
| 그룹 폐기 | no | no | yes |

S. Gusmeroli[8]는 사물인터넷 시스템의 접근제어를 위해 자격기반 접근제어 기법(Capability Based Access Control)을 제안하고, 이를 CapBAC으로 명명하였다. 이 연구는 최소 권한 원칙과 권한 위임 기능을 부여하여 주체에게 자신의 서비스 및 정보에 대한 접근제어를 관리할 수 있도록 하였다.

표 3은 기존 관련 연구와 본 논문의 서비스 접근제어에 대한 차이점을 정리한 것이다[1]. 본 논문의 서비스 접근제어는 접근권한은 주체가 가지고 있고 주체의 권한은 위임 할 수 있다. 이를 통해 주체는 위임 받은 정도의 권한으로 리소스에 접근할 수 있다. 또한 접근 권한은 폐기될 수 있으며, 정보의 세분화를 통해 권한의 동적 적응성을 제공한다. 이 연구에서 자격기반 접근제어 표기는 SAML/XACML 로 기술하였다.

## 2. 서비스 탐색 기법

사물인터넷 환경에서 서비스 탐색 방법으로 CoAP-RD와 DNS-SD 기법으로 기술이 개발 중에 있다. DNS-SD는 서비스 탐색을 위한 방법으로 사용되며, CoAP의 서비스 탐색 시 상호 보완적 방법으로 활용된다. CoAP은 CoAP 자원의 위치와 식별 방법으로 CoAP URI 형식으로 표현한다[10].

CoAP-RD와 DNS-SD 프로토콜은 사물인터넷 환경에서 서비스 탐색과 자원탐색으로 구분되며, 프로토콜의 종류에 따라 서비스 및 자원 정의를 다르게 사용된다. CoAP는 서비스의 프로토콜, 호스트, 포트 등의 정보 집합으로 정의된다. 자원의 정의는 REST 기반의 상호작용을 위한 포인트로 표현 되고 coap://[2001::11]:5683/sensor/temp로 정의된다. DNS-SD는 서비스를 서비스의 서브타입과 프로토콜 타입 등으로 정의된다.

CoAP에서는 분산형 CoAP자원 탐색 기법과 자원디렉터리 기반의 중앙형 CoAP 자원 탐색 기법 등이 사용된다. 분산형 CoAP 자원 탐색 기법은 자원디렉터리와 같은 외부 요소를 이용하지 않고 디바이스 스스로 자원 탐색을 수행하는 방법이며 이에 반해 자원디렉터리 기반 서비스 및 자원 탐색 기법은 모든 자원을 자원디렉터리 내에서 관리하고 자원에 대한 문의를 자원디렉터리 내에서 처리하는 중앙 관리 방식이다[15].

## 3. 사물인터넷 서비스 접근제어 시스템

### 가. IoT 서비스 게이트웨이

사물인터넷 서비스 접근제어 시스템은 사물인터넷 환경에서 디바이스 장치로부터 수집된 데이터 분석을 통해 리소스와 리소스 서비스를 정의하였다. 또한 주체 인증을 위한 인증서와 서비스 인가를 위한 C&C 서비스 토큰 관리를 위해 IoT서비스 게이트웨이를 설계하고 구현하였다[1].


그림 1. IoT 서비스 시스템 구성도

그림 1은 센싱 디바이스를 이용한 장치노드로 구성된 서비스 도메인 영역과 각 도메인 영역에 대한 데이터 수집 및 전송을 처리하는 리소스 게이트웨이 영역, 리소스게이트웨이 영역을 통해 수집된 데이터 정보를 제공할 수 있는 서비스 영역으로 구성된 IoT 서비스 접근제어 시스템 구성도이다[2].

서비스 영역에서는 주체의 서비스 요청에 대한 인증서를 발급하고 요청 서비스 토큰에 대한 위임, 폐기, 위임거부 와 같은 접근제어 기능을 수행한다. 서비스 도메인 영역은 서비스 도메인별 리소스 게이트웨이에서 센서, 컨트롤러, 영상 등 서비스를 제공하는 객체의 정보를 관리한다. 리소스 게이트웨이 영역은 리소스 게이트웨이가 해당 서비스 도메인의 데이터를 수집하여 IoT 서비스 게이트웨이 저장장소로 전송한다. IoT 서비스 게이트웨이는 주체의 C&C 서비스 토큰관리와 리소스 서비스 접근에 대한 인터페이스 역할을 하고, LCRS 서비스 관리 모델을 통해 서비스 도메인 영역에 대한 확장이 가능하다.

### 나. 리소스 모듈

본 논문의 IoT 서비스 접근제어 시스템에서는 논리적 또는 물리적 공간으로 구분되는 서비스도메인영역을 분류하고 각 서비스도메인에서 발생하는 디바이스 원시 데이터를 리소스게이

트웨이영역을 통해 중앙 데이터베이스에 저장한다.

저장된 데이터를 분석하여 새로운 서비스를 생성하고 이러한 서비스에 대한 서비스 모델을 관리하는 역할을 리소스 모듈에서 처리한다.


그림 2. 리소스 모듈

그림 2의 리소스 서비스 엔진은 서비스도메인관리, 리소스게이트웨이관리, 리소스서비스 관리로 나뉜다.

서비스도메인관리는 센서 디바이스 및 사물을 물리적 공간으로 그룹화하여 관리한다. 그룹화 된 도메인은 그룹별로 관리하는 리소스 게이트웨이를 통해 데이터를 전송한다. 리소스 게이트웨이는 각각의 서비스도메인영역에서 발생하는 사물의 원시 데이터를 수집하고 처리하며, 데이터를 저장하는 인터페이스 역할을 수행하고 관리한다. 리소스서비스는 수집된 디바이스 데이터를 기반으로 분석 서비스 또는 디바이스 제어 서비스를 생성하고 목록화 하여 서비스별 모델을 관리한다. 또한 서비스 접근에 따른 API를 제공한다.

### 다. 서비스 토큰

C&C 서비스 토큰은 사물인터넷 서비스 접근제어 시스템에서 주체의 인증을 위한 인증서정보와 개인키 정보 그리고 리소스 서비스 접근을 위한 리소스 목록 리스트로 구성된다.

사용자는 IoT 서비스 게이트웨이에서 인가를 위한 인증서와 리소스 서비스 토큰을 이용하여 서비스를 제공받을 수 있다.

리소스 서비스 토큰은 주체에 의해 생성 될 수 없으며, 서비스 관리자의 의해 정의된 리소스 서비스에 대한 접근 허가 시 서비스 토큰이 생성되고 이에 대한 위임을 받을 수 있다. 서비스 토큰 위임 시 사용자는 위임하고자 하는 리소스 토큰의 서비스ID 정보를 확인하고 해당 서비스 토큰을 위임한다.

## 4. 사물인터넷 리소스 관리 모델 설계

본 논문은 기존 연구[1]에서 구현한 IoT 접근 제어 시스템에서 사용자가 리소스 서비스 접근을 요청하고 권한을 획득하는데 있어서 리소스 서비스의 관리를 효율적으로 수행하기 위하여 리소스 서비스에 대한 구조를 정의하고, 리소스 매핑 테이블을 이용한 리소스 URI 관리 방안을 제시하였다.

### 가. 사물인터넷 리소스

사물인터넷 리소스는 접근제어 시스템 구성을 위한 노드장치, 데이터, 주체에게 편리성을 제공하는 서비스의 집합체이다.

본 논문에서 노드는 도메인, 서버, 센싱 디바이스 및 게이트웨이에서 발생한 신호를 디지털화여 기록한 것으로, 데이터를 통해 분석한 정보와 디바이스 제어 기능을 주체가 이용할 수 있도록 구성하는 것을 리소스 서비스로 정의한다.

### 나. 사물인터넷 리소스 서비스 구조 정의

리소스 서비스는 사물인터넷 서비스를 구성하는 요소로 서비스 사용자, 응용프로그램, 네트워크 관리, 디바이스 관리 등 리소스의 디바이스를 이용한 제어 메소드 또는 서비스 메소드이다.

계층으로 구조화된 서비스는 파일시스템 접근제어로 기법으로 관리되며, 신규 디바이스가 추가 될 경우 서비스구분 및 서브-카테고리 서비스 분류 디렉토리와 서비스 파일 생성 절차를 거치고 접근 권한 설정이 필요하다. 또한 서비스 분류 변경 시 기존 디렉토리를 변경함에 따라 접근권한이 재정의해야 하는 문제점이 있다.

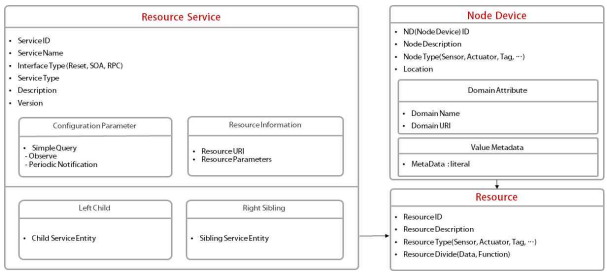


그림 3. LCRS 리소스 구조

그림 3은 기존의 M2M으로 연결된 디바이스의 계층구조 서비스 모델의 단점을 보완하고자 본 논문에서 제안한 LCRS 서비스 모델링 구조이다. 본 논문은 디렉토리 형태로 구조화된 리소스 서비스 접근에 대한 효율적 관리를 위해 LCRS 트리 구조를 사용하여 서비스 모델을 제안하고 CoAP의 디렉토리 구조 표현된 URI 정보를 C&C 서비스 토큰과 서비스 URI 매핑테이블을 이용하여 서비스 접근 정보를 암호화를 통해 보안성을 높이고자 하였다. 본 논문의 리소스 서비스는 주체의 서비스 토큰과 매핑 되고, 신규 디바이스 추가 시 디바이스 정보 및 제어에 필요한 리소스 서비스는 리소스 관리자가 정의하고 관리한다.

그림 3에 정의된 리소스 서비스 구조 모델링의 각 요소인 리소스 서비스, 리소스, 노드 디바이스는 다음과 같은 기능을 수행한다.

#### (1) 리소스

서비스도메인을 구성하는 영역으로 노드 디바이스와 노드 디바이스로부터 생성된 원형 데이터, 원형 데이터를 분석하여 주체에게 유용한 정보를 제공할 수 있도록 하는 리소스 서비스의 집합체이다.

#### (2) 노드 디바이스

노드 디바이스는 리소스를 구성하기 위한 자원으로 조도, 습도, 풍향, 풍속, 온도 등 센싱 데이터를 생성하는 장치이다. 노드 장치는 노드 장치의 구분을 위한 NDID(Node Device IDentity), 임베디드 센서/액츄에이터 노드와 같은 노드장치 유형, 노드의 위치정보와 그룹화 되는 도메인 정보를 참조한다.

(3) 리소스 서비스

사물인터넷 환경에서 리소스 서비스는 노드 장치로부터 발생한 원형 데이터를 분석하여 정보를 제공하는 서비스메소드(sM)와 노드 장치의 기능 제어를 위한 제어메소드(cM) 집합이다.

- Control Method : { $cM_1$, $cM_2$, $cM_3$, $cM_4$, ... , cMn }
- Service Method : { $sM_1$, $sM_2$, $sM_3$, $sM_4$, ... , sMn }
- Node Device Method Group : ndMGn{ cMn + sMn }

이러한 노드장치의 서비스메소드와 제어메소드로 조합된 노드 디바이스 메소드 그룹은 다른 노드장치의 메소드 그룹과 결합하여 리소스서비스를 생성한다.

- RSn : { $ndMS_1$, $ndMS_2$, $ndMS_3$, $ndMS_4$, ... , ndMSn }

리소스 서비스는 주체의 요구에 맞게 재구성 할 수 있고 일부 서비스 메소드가 삭제 될 수 있으며, 재 그룹화 되면서 새로운 리소스 서비스가 될 수 있다.

리소스 서비스는 최소 한 개 이상의 ndMS로 구성되며, 서비스 관리자에 의해 정의되고 관리된다.

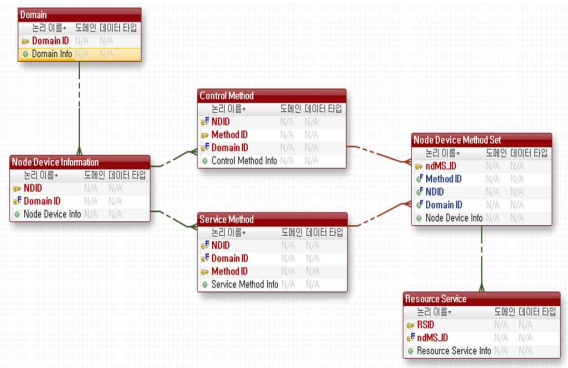


그림 4. LCRS 서비스 ERD

그림 4는 LCRS 서비스 관리 모델을 ERD로 표현한 것으로 노드 장치에 대한 컨트롤 메소드와 서비스 메소로를 분리시켜 노드 디바이스 메소스 그룹을 만들 수 있도록 테이블 관계를 설정하였다.

### 다. 사물인터넷 LCRS 리소스 모델

그림 5는 본 논문에서 구축한 테스트베드 환경을 LCRS 리소스 서비스 모델로 표현한 것으로 LCRS 리소스 서비스 모델은 서비스 분류 과정을 통해 새로운 융합 서비스를 생성할 수 있다[10]. 디렉토리 구조 형태의 서비스 모델 관리 형태는 최하단에 서비스를 위치시켰으나, LCRS 서비스 관리 모델은 각 서비스 모델 노드에 리소스를 연결하여 해당 리소스 서비스에 각각의 독립된 서비스가 이루어지도록 하였다.

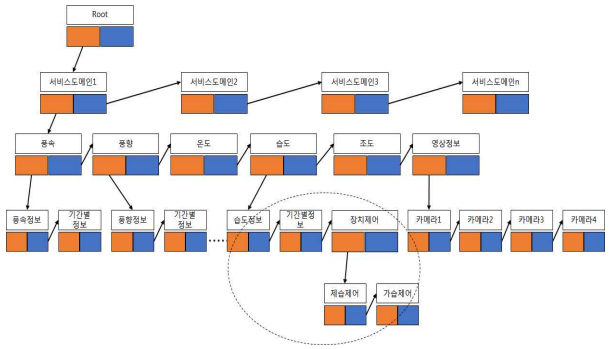


그림 5. LCRS 리소스 재구성

기 정의된 리소스 모델에서, 제어 서비스를 습도환경 정보 서비스 하단 영역으로 복사하면 습도환경 정보를 취득할 수 있는 서비스에서 장치 제어에 대한 서비스 권한을 부여 받아 처리 할 수 있으며, 도메인 영역의 리소스를 재배치를 하고자 할 경우 해당 리소스 서비스를 변경하고자 하는 서비스도메인으로 이동시키면 리소스 서비스에 대한 재분류가 가능하다. 독립된 장치 제어 리소스 서비스를 습도 디바이스와 관련된 서비스 분류로 이동한 것으로 리소스 서비스 구성요소의 자식 서비스 엔티티와 형제 서비스 엔티티의 연결 정보를 변경하여 리소스 서비스에 대해 재정의 할 수 있다.

### 라. 리소스 매핑 테이블을 이용한 리소스 URI 관리

사물인터넷 환경에서 각 노드 장치와의 서비스 통신을 위해 REST 개념을 적용하여 서비스를 제공하고 있다.

REST는 통신 규약이나 표준 또는 스펙이 아니라 분산 하이퍼미디어 시스템을 위한 HTTP 같은 형식으로 네트워크에서 클라이언트와 서버 사이의 통신 방식으로 서비스 제공 형식은 “/groups/groupid/groupid/member/sensor” 와 같이 계층적 구조로 구성된다[5][10].

REST의 장점은 통신을 위한 별도의 서비스를 설치하지 않아도 되고 특정 언어에 귀속되지 않게 서비스를 구성 할 수 있다. 하지만 설계 및 구성에 표준이 없어 각 서비스별로 아키텍처가 달라질 수 있고, 이것에 따를 서비스 구현 방안이 달라져 시스템의 복잡도가 높아질 수 있다[7].

REST을 기반으로 한 유일한 URI을 가지는 서비스는 사물인터넷과 같은 다양한 서비스가 존재하는 환경에서 효율적 구성이 아니다. 또한 계층적으로 구성된 URI는 직관적으로 어떤 기능을 제공하는지 예측이 가능하다. 이러한 서비스의 URI 정보는 비 인가된 주체로부터 공격 대상이 될 수 있는 위험요소가 된다.

## 5. 리소스 서비스 구현

본 논문에서는 IoT 서비스의 접근제어 플랫폼의 테스트베드에서 리소스 서비스 관리 모델을 테스트 하였다.

구현된 기술의 검증을 위해 테스트 환경에 포함된 각 리소스에 대한 리소스 서비스 등록 및 관리 여부를 테스트하였다.

### 가. 시스템 환경

본 논문에서 구축한 온도, 습도, 조도, 풍향, 풍속 센서 디바이스와 4대의 IP 카메라 및 디바이스 제어 컨트롤을 설치한 IoT 서비스 게이트웨이 기반의 사물인터넷 서비스 테스트베드를 구성하였다.

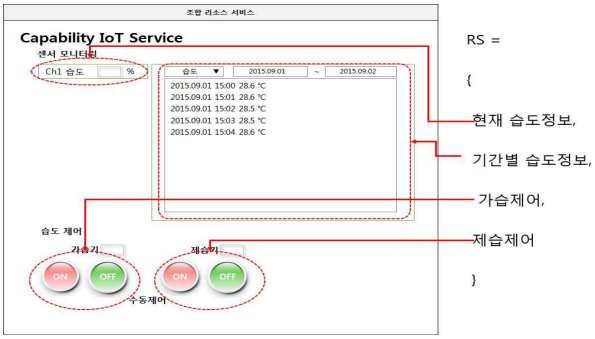


그림 6. 리소스 서비스 구성 예

그림 6은 테스트베드 환경에서 구현된 리소스 서비스의 예를 보이고 있으며, 습도 센싱 정보를 모니터 하는 서비스로 현재 습도 상태 값을 제공하는 ndMS와 일정기간 동안의 온도 정보를 전달하는 ndMS 메소드 셋 조합을 통해 생성된 리소스 서비스이다.

### 나. 리소스 서비스 생성 및 관리 시나리오

그림 7은 리소스 서비스의 생성 및 접근 과정을 보이고 있으며, 리소스관리자가 리소스 서비스를 생성하고, 이에 대해 서비스 주체가 리소스 URI 매핑 테이블 정보를 이용하여 리소스 서비스에 접근하는 시나리오는 다음과 같다.

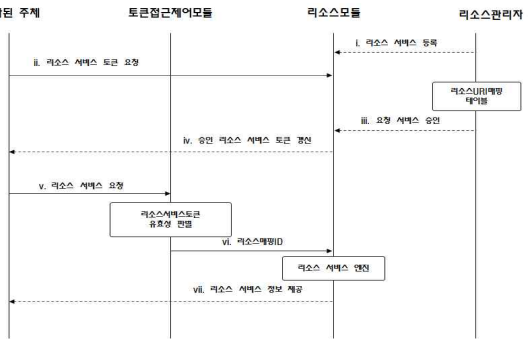


그림 7. 리소스 매핑 테이블을 이용한 서비스 요청

i. 리소스 서비스 관리자는 노드 디바이스 제어 서비스 또는 수집된 데이터를 활용하여 정보를 제공하는 리소스 서비스 만든다. 리소스 서비스에 ID를 발급하여 서비스 목록에 등록한다.

ii. 인가된 주체는 리소스 관리자 등록한 리소스 서비스 목록에서 이용하고자 하는 서비스에 대해 접속 할 수 있는 리소스 서비스 토큰을 요청한다.

iii. 리소스 관리자는 주체가 요구한 리소스 서비스에 대해 승인한다. 승인 과정에서 요청 주체의 암호화 알고리즘 정보를 이용하여 리소스 서비스 ID 값을 인증 개인키로 암호화 하여 리소스 서비스 매핑테이블에 정보를 저장한다.

iv. 토큰접근제어 모듈은 리소스 관리자가 승인 한 리소스 서비스 토큰 정보를 갱신하여 인가된 주체에게 전달한다.

v. 리소스 관리자가 접근을 허용한 리소스 서비스 토큰을 가지고 서비스 주체는 토큰접근제어모듈에 인증서정보와 암호화된 리소스 서비스 정보를 전송한다.

vi. 토큰접근제어 모듈은 주체의 요청 리소스 서비스 토큰의 리소스 영역 정보를 확인하여 요청서비스 분석, 공개키의 위변조 확인과 유효기간, 서비스 토큰 폐기목록을 참조하여 유효성 판별 후 주체 서비스 토큰 정보를 갱신하다. 마지막으로 주체의 개인키를 이용하여 암호화된 리소스 서비스 정보를 복호화 한 후 URI 매핑 테이블 통해 최종 리소스 서비스 URI를 리소스 모듈로 전달한다.

vii. 리소스 서비스 엔진은 해당 서비스 URI 정보를 통해 서비스를 주체에게 제공한다.

### 다. 리소스 서비스 구현

```
. RS : 리소스 서비스 객체
 LOOP
  RSid = RS.createid();
  IF RS,Exists(RSid) THEN
   Continue;
  ELSE
   RS.Store_ResourceServiceList(RSid,ResourceInfo);
  END IF:
END LOOP;
```

그림 8. 리소스 서비스 등록

```
. RS : 리소스 서비스 객체
. Cert : 인증서 객체
. RSList : Resource Service List Object

  RSList = RS.getResource_ServiceList();
  RSi = RSList.getId(i);
  KeyPare = Cert.getKeyPare();
  RS.RequestRS(RSi,KeyPare);
```

그림 9. 리소스 서비스 목록 갱신

```
. RS : 리소스 서비스 객체

RSObject[] = RS.getRequestRSList()

 LOOP
  IF RSObject[i] != EOF THEN
   IF RS.Agree(RSObject[i]) THEN
    _RSid = RSObject[i].getRSid()
    _Keypare = RSObject[i].getKeypare()
     LOOP
      _EncryptRS = RS.setEncrypt(_RSid,_Keypare )
      IF RS.ExistsMatrix(EncryptRS) THEN
       Continue;
      ELSE
       RS.Store_ResourceMatrix(_RSid,_EncryptRS)
      END IF
     END LOOP
   END IF
  END IF
 END LOOP
RETURN RSObject
```

그림 10. 요청 서비스에 대한 승인과 등록

```
. RS : 리소스 서비스 객체
. CnCToken : C&C 서비스 토큰 객체
. ServiceToken : 토큰접근제어모듈의 서비스 토큰 객체

RS.requestRS(RSToken,CnCToken.Cert)
 IF ServiceToken.classify(RSToken) THEN
  IF ServiceToken.validity(RSToken) THEN
   RSToken = ServiceToken.updateRSToken(RSToken)
   Keypare = CnCToken.getKeypare()
   DecryptRS = RS.Decrypt(Keypare,RS.EncryptRS)
   IF RS.MatrixMapping(RS.getRSMatrix(DecryptRS)) THEN
    RSuri = RS.getRSuri(RSid);
    RETURN RSToken.Service;
   IF END
  IF END
 IF END
```

그림 11. 리소스 서비스 요청과 토큰 갱신

그림 8, 그림 9, 그림 10, 그림 11은 리소스 서비스 요청 i, ii, iii, iv 단계로 리소스 관리자가 등록한 서비스를 주체가 요청하고, 서비스 접근 권한을 획득하는 과정을 기술한 것이다.

표 4. 사물인터넷 리소스 서비스

| 서비스명 | 설명 |
|---|---|
| Live Humidity | 실시간 습도 조회 서비스 |
| Live Illumination | 실시간 조도 조회 서비스 |
| Live Temperature | 실시간 온도 조회 서비스 |
| Live WindCustom | 실시간 풍속 조회 서비스 |
| Live WindInfo | 실시간 풍향/풍속 조회 서비스 |
| Live WindDirection | 실시간 풍향 조회 서비스 |
| PowerDevice | 가습기, 제습기 전원 제어서비스 |
| Now Humidity | 단일 습도 조회 서비스 |
| Now Illumination | 단일 조도 조회 서비스 |
| Now Temperature | 단일 온도 조회 서비스 |
| Now WindCustom | 단일 풍속 조회 서비스 |

표 4는 본 논문에서 구현한 리소스 서비스에 대한 설명으로 한 개의 도메인 영역에 12개의 리소스 서비스로 구성하였다.

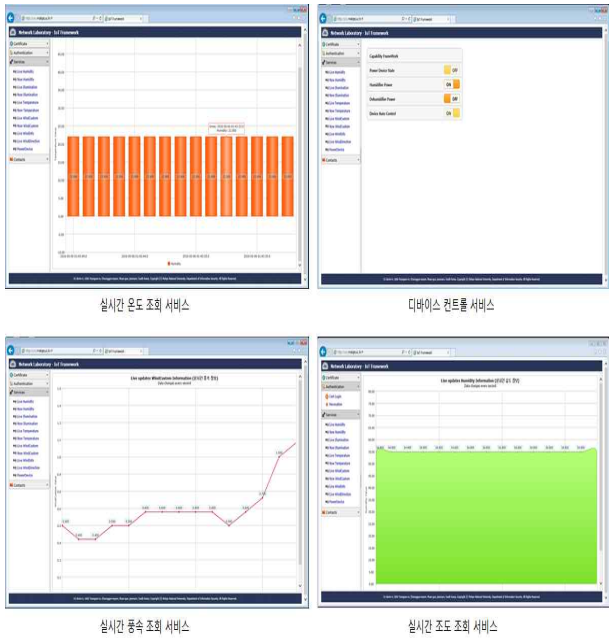


그림 12. RIA 기반의 리소스 서비스

그림 12는 리소스 서비스를 JQuery와 데이터베이스 연결 관리는 스프링 프레임워크 기반 iBATIS를 이용하여 구현한 것으로 센서 디바이스의 정보를 확인할 수 있는 서비스와 제어 컨트롤을 이용한 디바이스 제어 서비스를 나타내고 있다. 리소스 서비스는 IoT 서비스 게이트웨이의 리소스 모듈에서 서비스 토큰의 URI 정보를 참조하여 서비스를 제공하므로 주체의 플랫폼에 영향을 받지 않고 서비스 할 수 있다.

## III. 결 론

사물인터넷의 편리성을 제공하기 위한 서비스가 개별적이고 독자적인 방식으로 개발 될 경우 인터넷의 장점인 다양한 서비스로의 확장이 어렵고, 다른 기기 및 서비스와의 연동됨에 따라 보안 취약점이 발생하고 있다. 서비스에 대한 편리성과 효율성만을 강조한 나머지 서비스 공격에 대한 보안 안전성 확보가 어려운 것이다.

본 논문은 무분별하게 관리되는 사물인터넷 서비스를 효율적으로 관리하고, 사물인터넷 서비스에 대한 접근제어를 위해 리소스 서비스 토큰을 설계 및 구현하였다. 구현한 시스템에서 LCRS 리소스 서비스 모델은 서비스 분류 과정을 통해 새로운 융합 서비스를 생성이 용이하고, 각 서비스 모델 노드에 리소스를 연결하여 해당 리소스 서비스에 각각의 독립된 서비스가 이루어질 수 있다. 또한 LCRS 리소스 모델 관리는 도메인별 서비스에 대한 접근 제어를 수행하기 때문에 사물인터넷 장치들에 종속되지 않아 서비스에 대한 관리 및 확장성을 갖추고 있다.

구현된 시스템은 단일 도메인 환경에서 센싱 데이터를 이용하여 사물인터넷 서비스를 구축한 것으로 차후 멀티 도메인 기반에서 리소스 서비스 관리 방안과 멀티 플랫폼에서 리소스 서비스 접근 방안에 대한 연구를 진행할 예정이다.

## References


[1] 김진보,장데레사,김미선,서재현, “CapSG를 이용한 IoT 서비스 접근제어 플랫폼”, 정보처리학회논문지. 제4권 제9호, 337~346, 2015.

[2] 장데레사,김진보,김미선,서재현,“IoT 서비스 접근제어를 위한 리소스 서비스 관리 모델 연구”, 한국정보처리학회, 제22권 제2호, 664~667, 2015.

[3] 최성찬,류민우,진남,김재호, “사물인터넷 플랫폼 및 서비스 동향”, 한국통신학회지, 20~27, 2014.

[4] 유재학,이형규,권순현,김선진,방효찬, “다양한 IoT 환경을 고려한 IoT 통합 플랫폼 기술 동향”, 10~22, 2015.

[5] Andrew J. Rettig and Sumit Khanna and Richard A. Beck, “Open source REST services for environmental sensor networking”. Applied Geography, 294~300, 2015.

[6] Internet of Things Architecture, IoT-A, Project Deliverable D4.3.

[7] JoseL. Hernandez-Ramos & M. VictoriaMoreno & Jorge BernalBernabe & DanGarciaCarrillo & Antonio F.Skarmeta. "SAFIR: Secure access framework for IoT-enabled services on smart buildings". Journal of Computerand System Sciences, 81, 1452~1463, 2015.

[8] Luis Sanchez, Luis Munoz, Jose Antonio Galache, Pablo Sotres, Juan R. Santana, Veronica Gutierrez and Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis and Dennis Pfisterer. "SmartSantander: IoT experimentation over a smart city testbed". Computer Networks, 61, 217~238, 2014.

[9] 임유경. “서비스 관점의 IoT를 말한다”, http://blog.lgcns.com/758, 2015.

[10] REST, http://rest.elkstein.org

[11] Gartner, Http://www.gartner.com/newsroom/id/3114217

[12] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic and Marimuthu Palaniswami. “Internet of Things (IoT): A vision, architectural elements, and future directions”. Future Generation Computer Systems, 29, 1645~1660, 2013.

[13] Jos´e L. Hern´andez-Ramos, Antonio J. Jara, Leandro Mar´ın and Antonio F. Skarmeta. "Distributed Capability-based Access Control for the Internet of Things". Journal of Internet Services and Information Security, Vol 3, No 3/4, 1~16, 2013.

[14] 이범기,김미선,서재현, “IoT에서 Capability 토큰 기반 접근제어 시스템 설계 및 구현”, 정보보호학회논문지, 제25권 제2호, 439~448, 2015.

[15] 윤주상,최영환, “IoT서비스 탐색 기술”, 정보와통신 열린강좌, 32(12(별책2호)), 20~26, 2015


## 저 자 소 개


**김진보(정회원)**

2003년 국립목포대학교 멀티미디어학과 학사 졸업.
2007년 국립목포대학교 정보보호기술학협동과정 석사 졸업.
2016년 국립목포대학교 정보보호기술학협동과정 박사 졸업.

<주관심분야 : 웹서비스 보안, 네트워크 보안, 빅데이터, 프로그래밍 언어>

**김미선(정회원)**

1996년 국립목포대학교 컴퓨터공학과 학사 졸업.
2000년 국립목포대학교 컴퓨터공학과 석사 졸업.
2007년 국립목포대학교 컴퓨터공학과 박사 졸업.
2012년~현재 국립목포대학교 정보보호학과 초빙교수

<주관심분야 : 정보보호, 프로그래밍 언어, 컴퓨터 네트워크, 모바일 시스템 보안>

**서재현(정회원)**

1985년 전남대학교 계산통계학과 학사 졸업.
1988년 중앙대학교 전자계산학과 석사 졸업.
1996년 전남대학교 전산통계학과 박사 졸업.
1996년~현재 국립목포대학교 정보보호학과 교수

<주관심분야 : 정보보호, 시스템 및 네트워크보안, 컴퓨터 네크워크, 모바일 네트워크 보안>