KIPS Tr. Software and Data Eng.
Vol.4, No.9 pp.337~346 pISSN: 2287-5905

http://dx.doi.org/10.3745/KTSDE.2015.4.9.337

The Access Control Platform of the IoT Service Using the CapSG

Jin-Bo Kim† · Deresa Jang†† · Mi-Sun Kim††† · Jae-Hyun Seo††††

ABSTRACT

There is great need for efficient user rights management method to provide a flexible service on variety protocols, domains, applications of IoT environments. In this paper, we propose a IoT service platform with CapSG to provide efficient access control for IoT various services of the environment. CapSG uses a token including authentication and access rights to perform authentication and access control service entity providing services. In addition, the generated token for service management, delegation, revocation, and provides a function such as denied. Also, it provides functions such as generation, delegation, disposal and rejection for service token management. In this paper, it provides the flexibility and efficiency of the access control for various services require of the IoT because of it is available to access control specific domain service by using the token group for each domain and is designed to access control using specific service token of tokens group.

Keywords : IoT, Capability Access Control, Security, Access Control

CapSG를 이용한 IoT 서비스 접근제어 플랫폼

김 진 보† · 장 데 레 사†† · 김 미 선††† · 서 재 현††††

요 약

사물인터넷(Internet of Things, IoT) 환경의 다양한 프로토콜, 도메인, 애플리케이션 위에서 유연한 서비스 제공을 위한 효율적인 사용자 권한 관리 방법이 필요하다. 본 논문은 IoT 환경의 다양한 서비스에 대한 효율적인 접근제어를 제공하기 위하여 CapSG(Capability Service Gateway)를 이용한 IoT 서비스 플랫폼을 제안하였다. CapSG는 인증과 접근 권한을 포함하는 토큰을 사용하여, 서비스 주체에 대한 인증 및 접근제어를 수행하여 서비스를 제공한다. 또한, 서비스 토큰 관리를 위한 생성, 위임, 폐기, 거절 등의 기능을 제공한다. 본 논문은 각 도메인에 대한 토큰 그룹을 사용함으로써 도메인별 서비스 접근제어가 가능하며, 토큰 그룹 내의 특정 서비스 토큰을 이용한 접근제어 수행도 가능하도록 설계하여 IoT의 다양한 서비스 요구에 대한 접근제어의 유연성과 효율성을 제공한다.

키워드 : 사물인터넷, 케이퍼빌리티 접근제어, 보안, 접근제어

1. 서 론

IoT는 글로벌 네트워크 인프라를 기반으로 하는 새로운 인터넷 기반의 정보 아키텍처로 RFID(Radio Frequency Identification) 태그와 리더기, NFC(Near Field Communication) 장치, 임베디드 센서/액추에이터 노드와 같은 디바이스 기술의 활성화를 가져왔다[1]. 그러나 IoT 기술의 광범위한 적용은 파생 가능한 위험을 내포하고 있으며, 실제로 기존의 인터넷에 비해 훨씬 더 널리 정보 보안 위험을 분산할 수 있다[2]. 다양한 무선 통신 장치뿐만 아니라, 사람, 사물, 데이터와 같은 모든 객체가 인터넷과 연결되기 때문에 각 객체 간의 프라이버시 및 보안의 문제가 중요한 영향을 미친다. 따라서, 공격, 데이터 인증, 접근제어 및 프라이버시에 대한 구조적인 탄력성을 보장할 수 있는 방법이 수립될 필요가 있다[3].

Capability 기반 접근제어는 주체가 권한 리스트를 가지고 있으며, 자신이 갖고 있는 Capability를 서비스 제공자에게 제시하면, 서비스 제공자는 Capability를 확인하여 인가한다. 반복적인 서비스 요청이 발생할 경우 ACL 기반 시스템에서는 반복적으로 인증 프로세스가 진행되지만, Capability 기반 접근제어는 기발행된 Capability를 통해 반복 작업을

※ 본 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2014R1A2A1A11053774).
† 준 회 원 : 목포대학교 정보보호기술학협동과정 박사과정
†† 준 회 원 : 목포대학교 정보보호기술학협동과정 석사과정
††† 정 회 원 : 목포대학교 정보보호학과 초빙교수
†††† 종신회원 : 목포대학교 정보보호학과 교수
Manuscript Received : July 23, 2015
First Revision : August 26, 2015
Accepted : August 27, 2015
* Corresponding Author : Jae-Hyun, Seo(jhseo@mokop.ac.kr)

최소화할 수 있다. 따라서 ACL 기반 시스템에 비해 워크플로우가 가볍다고 할 수 있다[4].

본 논문에서는 IoT에 포함된 다양한 리소스들, 서비스들을 관리하고 사용자 및 서비스 주체에 대해 안전하고 신뢰할 수 있는 서비스를 제공하기 위한 서비스 접근제어 플랫폼을 제안한다. 제안된 IoT 서비스 접근제어 플랫폼은 사용자 및 서비스 주체의 서비스 요청에 대한 중계자 역할을 수행하는 CapSG(Capability Service Gateway)를 이용하여 다양한 리소스, 서비스에 대한 신뢰할 수 있는 접근을 수행한다. CapSG는 인증과 접근 권한을 포함하는 CaC(Certificate and Capability) 토큰을 사용하여, 서비스 주체에 대한 인증 및 접근제어를 수행한다. 또한, 서비스 토큰 관리를 위한 생성, 위임, 폐기, 거절 등의 기능을 제공한다.

제안한 IoT 서비스 접근제어 플랫폼에서 서비스는 도메인 영역 내에 포함되며, 도메인에 대한 토큰 그룹을 사용함으로써 도메인별 서비스 접근제어가 가능하다. 또한, 토큰 그룹 내의 특정 서비스 토큰을 이용한 접근제어 수행도 가능하도록 설계하여 IoT의 다양한 서비스 요구에 대한 접근제어의 유연성과 효율성을 제공한다. 인증 및 접근제어는 CapSG에 의해 수행되며, 도메인별 서비스 접근제어를 수행하기 때문에 IoT 환경의 다양한 IoT 장치 관리 및 확장성이 용이하다.

본 논문은 다음과 같이 구성되어있다. 2절에서는 IoT에서의 보안 및 프라이버시 관련 연구에 대한 내용을 기술하고, 3절에서는 CapSG를 이용한 IoT 기반 서비스 접근제어 플랫폼을 제시하며, 인증 및 접근제어를 위한 CaC 토큰 구조를 설명한다. 4절에서는 제안한 시스템의 구현 및 테스트 결과를 보여주고, 마지막으로 5절에서는 결론과 미래 연구방향에 대해 제시한다.

2. 관련 연구

IoT의 주요 이슈는 각 장치의 신뢰성과 프라이버시 및 보안을 보장하면서, 적응성과 자율성을 가지고 연결된 전체 장치 간의 상호운용성을 가능하게 하는 것이다. IoT의 보안과 프라이버시 문제는 기존 인터넷 환경과의 차별성을 고려하여 접근하여야 한다. 첫째, IoT는 기존 인터넷 환경과 달리 짧은 시간 동안 상호작용이 일어나며, 동일한 요청이 자주, 자발적으로 수행될 수 있다. 둘째, IoT에서 자원/서비스/오퍼레이션/데이터 등에 대한 분석 및 인가는 같은 요청에 대해서도 고정적이지 않고, 주변의 상황에 따라서 바뀔 수 있다. 따라서, IoT와 같이 개방되고, 광범위한 컴퓨팅 환경에서는 확장성의 문제, 장치들의 관리의 문제, 유연성 있고 쉬운 권한 위임의 문제를 고려한 접근제어 기법이 필요하다[5-6].

접근제어는 주체가 정책에 따라 객체의 작업을 수행할 수 있는지 여부를 나타내는 것으로, 자원에 대한 인가되지 않은 접근을 감시한다. 접근 요청에 대한 이용자를 식별하며, 접근요청이 정당한 것인지를 확인하여 기록하고 보안정책(security policy)에 따라 접근 승인 또는 거부함으로써 비인가자로부터의 불법적인 자원접근 및 파괴를 예방한다. 즉 접근제어는 각 자원에 대한 기밀성, 무결성, 가용성 및 합법적인 이용과 같은 정보보호 서비스에 권한부여를 위한 수단이 된다[6-7]. 접근제어 기법으로는 접근제어 리스트(Access Control List, ACL), 역할기반 접근제어(Role Based Access Control, RBAC), 속성기반 접근제어(Attribute Based Access Control, ABAC) 등이 있으며, 최근에는 IoT를 위한 접근제어 기법으로 Capability 기반 접근제어, 위치-시간 기반 접근제어에 대한 연구가 진행되고 있다[5, 6, 9, 10, 11].

Capability 기반 접근제어는 주체가 권한 리스트를 가지고 있으며, 자신이 갖고 있는 Capability를 서비스 제공자에게 제시하면, 서비스 제공자는 Capability를 확인하여 인가한다. 반복적인 서비스 요청이 발생할 경우 ACL 기반 시스템에서는 반복적으로 인증 프로세스가 진행되지만, Capability 기반 접근제어는 기발행된 Capability를 통해 반복 작업을 최소화할 수 있다. 따라서 ACL 기반 시스템에 비해 워크플로우가 가볍다고 할 수 있다[4].

Mark S. Miller[12]는 전통적인 접근제어 기법에서 사용하는 접근제어 리스트 시스템과 Capability 기반 시스템의 차이를 설명하며, Capability 기반의 접근제어 시스템의 장점을 강조하고 있다. 이 연구에서는 Capability에 대한 오해를 3가지로 제시하고 있는데, 첫 번째, 접근제어 리스트와 Capability 리스트는 형식적으로 동일하며, 두 번째, Capability는 제한성을 제공하지 않고, 권한 취소가 불가능하다. 그러나 실제로 Capability 기반 접근제어는 Lampson의 접근 행렬에 기초하였기 때문에 ACL과 형식적으로 유사한 면이 있으나, 권한 표현에 있어서 다른 관점에서 접근하고 있다. 또한, Capability 기반 접근제어는 위임을 제공하지만 전파의 경계를 명확하게 제시하고 있으며 위임된 권한은 취소가 가능하다. 이 연구에서는 3가지 오해에 근거하여 기존의 ACL 기반 모델과 Capability 기반 시스템들의 비교를 통해 Capability 기반 시스템들의 보안성을 기술하였다.

S. Gusmeroli[4, 9]는 IoT의 접근제어를 위해 Capability 기반 접근제어 기법을 제안하였으며 이를 CapBAC로 명명하였다. 이 연구는 최소 권한 원칙과 권한 위임 기능을 부여하여 주체에게 자신의 서비스 및 정보에 대한 접근제어 사용자 정의를 지원한다. 접근 권한은 주체가 가지고 있으고, 주체가 가진 권한은 위임될 수 있으며, 이를 통해 위임받은 주체는 위임받은 정도의 권한으로 리소스에 접근할 수 있다. 또한, Capability는 폐기될 수 있으며, 정보의 세분화를 통해 권한의 동적 적응성을 제공한다. 이 연구에서 Capability는 SAML/XACML 표기에 따라 기술한다.

Bumki Lee[5]는 Capability 토큰 기반의 접근제어를 구현한 연구로 라즈베리파이와 LED센서로 구성된 테스트베드에서 Capability 토큰을 이용한 제어시스템을 구축하였다. 이 연구에서 Capability 토큰은 위임, 재위임, 폐기될 수 있다. 그러나 토큰의 위임 거절에 대한 부분은 고려하지 않고 있으며, 단일 서비스에 대한 구현에 국한되어있다.

Table 1은 기존 관련 연구와 본 논문의 차이점에 대해 정리한 표이다. 기존의 Capability 기반 접근제어 시스템[6,

10]이 모든 서비스에 대한 토큰을 각각 부여하고, 이에 대한 위임, 폐기를 수행할 수 있었던 것과 달리 본 논문은 도메인 영역에 해당하는 모든 서비스에 대해 각각의 서비스 토큰을 발행하고 이를 그룹으로 관리할 수 있으며, 토큰 그룹에 대한 위임 및 폐기가 가능하다. 기존의 연구에서는 권한 위임의 경우, 위임받는 자의 위임 거절에 대한 논의가 없었으나, 본 논문에서는 권한을 위임받는 주체가 무조건적으로 위임을 받는 것이 아니라, 경우에 따라 위임을 거절할 수 있는 기능을 제공한다.

Table 1. Difference between Existing Related Researches and this Paper

Property	CapBAC[10]	Capability Token[6]	CapSG
Right's subject	user	user	user
Delegation	yes	yes	yes
Revocation	yes	yes	yes
Delegation Reject	-	-	yes
Group Delegation	no	no	yes
Group Revocation	no	no	yes

3. 본 론

본 논문은 IoT 환경의 다양한 서비스에 대한 효율적인 접근제어를 제공하기 위하여 CapSG를 이용한 IoT 서비스 접근제어 플랫폼을 제안하였다. 이를 위하여 Fig. 1과 같이 온도, 습도, 조도, 카메라 등 장치노드로 구성된 서비스 도메인 영역, 각 도메인 영역에 대한 데이터 수집 및 전송을 처리하는 리소스 게이트웨이 영역, 리소스 영역을 통해 수집된 데이터 정보를 제공할 수 있는 IoT 서비스 영역으로 구성한다.

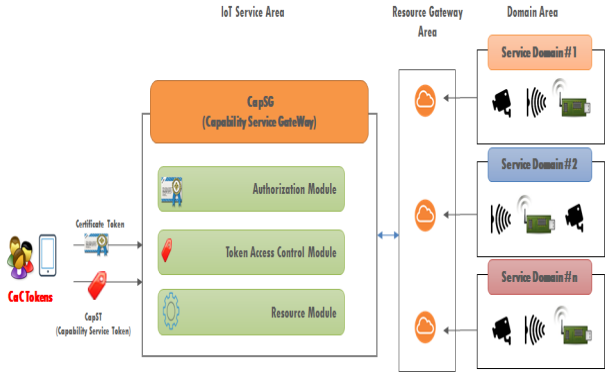

Fig. 1. Architecture of IoT Service Access Control Platform

IoT 서비스 접근제어 플랫폼에서 서비스의 주체는 사용자, 사물(Things) 및 디바이스가 될 수 있으며, 서비스 요청을 위한 토큰을 제시하여 IoT 내 서비스를 제공받을 수 있다. IoT의 모든 디바이스는 서비스 주체이면서 동시에 서비스 객체가 될 수 있으며, 제안한 플랫폼에서 서비스 도메인 내의 각 장치들은 서비스 객체이면서 동시에 서비스 도메인 내의 타 장치에 대한 서비스 주체가 될 수도 있다.

도메인 영역의 각 서비스 도메인은 같은 리소스 게이트웨이를 통하는 센서, 컨트롤러 및 카메라 등 서비스를 제공하는 객체들의 그룹이며, 리소스 게이트웨이 영역의 리소스 게이트웨이가 해당 서비스 도메인의 데이터를 수집하여 전송한다.

CapSG는 리소스 서비스를 요청하는 서비스 주체에 대해 인증 및 접근제어를 위한 CaC 토큰을 발행하고 관리한다. CaC 토큰을 사용하여 서비스 주체에 대한 인증 기능을 수행하고, 해당 서비스에 대한 접근제어를 수행한다. CaC 토큰은 사용자에 대한 인증 정보와 리소스 서비스 정보로 구성된 XML 문서 형식이며 CapSG의 인증 절차를 거쳐 리소스 서비스에 접근한다.

제안한 시스템에서는 CapSG가 서비스 접근을 위한 인증 및 접근제어를 중계하므로, 서비스 도메인 영역에 대한 확장성을 제공할 수 있다. 제안한 IoT 서비스 플랫폼에는 멀티 리소스 게이트웨이, 멀티 서비스 도메인으로 구성할 수 있고, 다수의 서비스 요청자가 존재한다. 서비스 요청자는 CapSG가 발행한 CaC 토큰을 사용하여 서비스에 접근할 수 있으며, CapSG를 통해 모든 구성요소들은 확장 가능하다.

3.1 CapSG

제안한 시스템에서 CapSG는 사용자 또는 서비스 주체의 요청에 대해 인증, 접근제어 서비스를 제공하는 중계자 역할을 수행한다. Fig. 2와 같이 CapSG(Capability Service Gateway)는 요청 핸들러(Ruquest Handler), 서비스 핸들러(Service Handler) 및 리소스 핸들러(Resource Handler)로 구성된다.

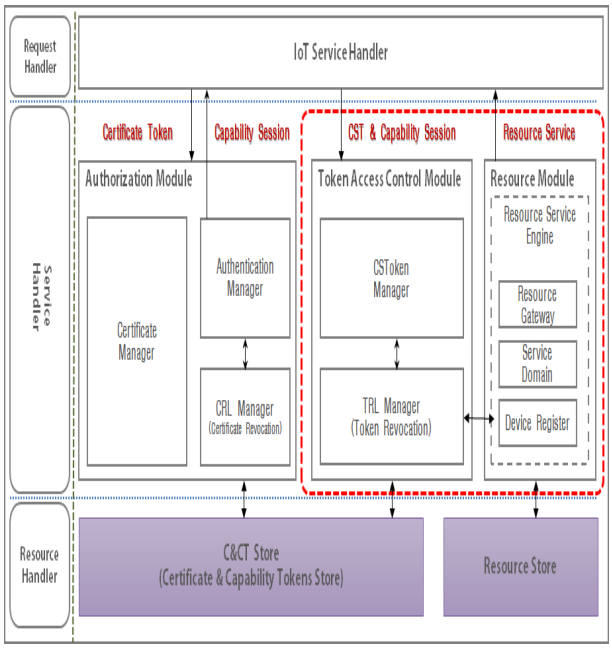

Fig. 2. Structure of CapSG

요청 핸들러는 사용자 및 서비스 주체자의 인터페이스 처

리 영역으로, 사용자 인증과 서비스 요청에 대한 분할 처리 역할을 수행한다.

서비스 핸들러는 CapSG의 중심 부분으로 인가 모듈, 토큰 접근제어 모듈, 리소스 모듈로 나누어 기능을 수행하며, 본 논문에서는 서비스 핸들러 중 토큰 접근제어 모듈과 리소스 모듈에 대한 부분에 대해 구현 및 테스트를 수행하였다. 서비스 핸들러는 요청 핸들러로부터 받은 요청에 의해 실질적인 인증 처리 및 접근제어 처리를 수행하며, 사용자 인증과 세션 생성, 인증서 취소 목록(Certificate Revocation List, CRL) 및 토큰 취소 목록(Token Revocation List, TRL)의 유효성 검증을 처리한다.

리소스 핸들러는 인증 토큰과 서비스 토큰 정보을 저장, 관리하고 장치노드에서 수집된 데이터를 분석하여 제공할 수 있는 리소스 서비스를 정의한다.

1) 인가 모듈(Authorization Module)

인가 모듈은 인증 토큰을 생성하고, 이를 이용하여 장치 및 사용자에 대한 인증을 수행하는 모듈이다. 인가 모듈에서는 인증 토큰 및 인증 토큰 취소 목록을 관리하며, 인증 절차를 수행한다. 인증 모듈은 인증 토큰 정보를 이용하여 CRL 유효성 검증을 통해 주체와 세션을 생성하고 유지하는 동안 인증 절차는 생략한다.

a) 인증서 관리자(Certificate Manager)

CapSG를 통해 서비스를 제공받고자 하는 사용자는 인증 모듈을 통해 인증서를 발급받아야 하며, 발급된 인증서의 개인키가 유출되거나, 갱신 또는 유효기간 종료 시 인증서 발급에 대한 정보를 관리한다.

b) 인증 관리자(Authentication Manager)

서비스 접근 사용자의 개인키를 이용하여 인증 절차를 처리하고, 인증처리 과정에서 CRL을 이용한 인증서 유효성 검증을 수행한다. 인증서에 대한 유효성 검증이 완료되어 인증된 사용자에 대해서 CapSession를 생성하고 생성된 세션에 대한 세션 인스턴스를 관리한다.

c) 인증서 취소 목록 관리자(CRL Manager)

인증서 관리자에서 발행한 인증서에 대해 서비스 탈퇴, 비밀키 손상 및 유출이 의심될 경우 폐기 절차를 거쳐 CRL에 등록하고 인증서 유효성 검증 서비스를 제공한다.

2) 토큰 접근제어 모듈(Token Access Control Module)

토큰 접근제어 모듈은 인가된 주체가 요청한 토큰에 대해 TRL 검증을 거쳐 위임, 폐기, 거부 토큰 정보를 갱신하고 요청에 대한 리소스 인터페이스를 관리한다. 인증 세션 생성 후 메인 토큰 정보를 이용하여 서비스 토큰에 대한 위임, 폐기, 거부 상태를 갱신하고, 유효성 검증 결과에 따른 서비스 토큰 요청 정보를 리소스 모듈에 전달한다.

a) CS토큰 관리자(Capability Service Token Manager)

요청 서비스 토큰에 대한 기간 만료 및 폐기 요청에 따른 서비스 토큰 상태 정보를 변경하고, 토큰 폐기 시 위임한 토큰 정보를 추적하여 해당 토큰 정보를 토큰 취소 목록 리스트에 전달한다. 토큰 접근제어 모듈을 통해 전달받은 서비스 토큰에 대한 유효성 검증을 수행한다.

b) 토큰 취소 목록 관리자(Token Revocation List Manager)

요청 서비스 토큰에 대한 목록을 관리하고 신뢰성 확보를 위해 정상 토큰 목록을 서비스 토큰의 시그니처를 이용하여 관리한다. 사용자에 의한 서비스 토큰 폐기는 사용자가 요청한 시점을 기준으로 정상 토큰 목록에서 관련 토큰 정보를 삭제 처리하고 유효기간이 만료된 토큰의 경우 이벤트 스케줄러를 통해 토큰 정보를 갱신한다.

3) 리소스 모듈(Resource Module)

도메인별 장치노드 등록과 서비스 도메인 및 리소스 게이트웨이를 관리하며 노드에서 수집된 데이터를 분석한 CapAIS(Capability Analysis Information Service)를 구성한다. 도메인 영역의 노드 장치에서 수집된 데이터를 기반으로 사용자가 이용할 수 있는 서비스를 생성하고 노드 장치의 접근제어 서비스를 관리하며, 요청 토큰에 대한 서비스 응답을 처리한다.

a) Resource Service Engine

서비스 토큰의 만료 기간과 접근 횟수에 대한 접근 제한을 하고자 토큰 취소 목록 리스트에 대한 유효성 검증을 수행하며, 장치노드의 제어 및 장치노드의 환경설정을 제어하는 API를 관리한다.

3.2 CaC(Certificate and Capability) 토큰 구조

제안한 시스템에서 CaC 토큰은 인증 및 서비스 접근제어를 위해 사용하며, Fig. 3과 같이 장치 및 사용자 인증을 위한 인증서 토큰과 서비스 접근 권한을 가진 Capability 토큰으로 구성된다. 인증서 토큰은 서비스 주체에 대한 인증 정보를 가지며, Capability 토큰은 도메인 영역에 대한 권리정보를 가진다. Capability 토큰은 도메인 서비스를 그룹화할 수 있는 메인 토큰 정보와 각 도메인 서비스 내의 서비스 각각에 대한 서비스 토큰으로 구성된다. CaC 토큰은 인가를 위한 인증서 토큰과 리소스 접근을 위한 서비스 토큰 정보를 XML로 표현한다.

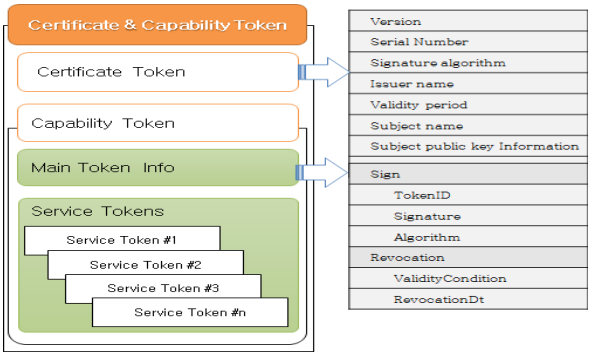

Fig. 3. Structure of CaC Token

인증서 토큰은 PKI에서 사용하는 표준 인증서 형식인

X.509 구조를 기반으로 구조화하였다. 인증서 토큰의 등록 및 CRL은 인가모듈을 통해 관리된다.

인증서 토큰이 만료 또는 폐기되는 경우에는 해당 인증서의 주체에게 발행된 모든 서비스 토큰들도 만료 또는 폐기되어야 한다. 따라서, 인증서 토큰의 만료 또는 폐기의 경우에는 메인 토큰 정보의 토큰 ID, 시그니처 데이터를 참조하여 해당 CaC 토큰에서 위임한 리소스 서비스 토큰들을 TRL에 등록하고 향후 위임된 리소스 서비스 토큰 접근 시 토큰 취소 목록 리스트를 참조하여 서비스 요청을 거부한다. 인증서 토큰은 CapSG에 접속하는 인증 과정에만 사용되며, 리소스 서비스 토큰 위임 시 인증서의 정보를 다른 사용자에게 전달하지 않고 리소스 서비스 토큰만 위임한다.

메인 토큰 정보(Main Token Info) 부분은 도메인 서비스 그룹의 정보를 나타내는 부분으로 그룹 토큰에 대한 기본 정보 및 그룹 단위 토큰에 대한 폐기 정보를 포함한다. 토큰ID, 토큰 구분자 서명, 암호화 알고리즘 정보를 포함한 메인 토큰 정보의 Sign 요소는 사용자가 CapSG를 통해 인증서 발급 또는 갱신 시 사용자가 정의한 토큰ID와 알고리즘을 이용한 시그니처를 생성하고, 토큰 폐기 정보를 포함한 Revocation 요소는 그룹 토큰의 상태 및 유효기간 정보를 관리한다. 메인 토큰도 폐기 및 갱신될 수 있으며 폐기 및 갱신이 이루어질 경우 해당 토큰ID로 위임한 모든 서비스 토큰 정보는 서비스 토큰의 서명과 리소스 정보 요소를 참조하여 TRL에 등록되고 관리된다. 메인 토큰 정보를 통해서 그룹 토큰 위임 및 폐기 기능이 가능하다.

Service Tokens	
Resource Token : ServiceID	Capability Service Token ID
Sign	Service Token-Signing Information
Owner	· Service Token Owner
Signature	· Service Token Signature
Algorithm	· Encrypted Algorithm Type
Resource	Resource Information
ResourceID	· Define by the resource administrator
Resource Rights	· Access rules to the resource service
AccessDt	· Setting the access period of resource service
Status	Status Information of the Service Token
Condition	· Status of the Service Token
RevocationDt	· Validity period of the service token
Delegate	Delegation information of service token
Delegable	· Whether the re-delegation token
DepthMaxCnt	· Delegation of times

Fig. 4. Structure of Capability Service Token

CaC 토큰은 메인 토큰 정보 하부에 서비스 토큰 정보를 갖고 있기 때문에 이를 통해 서비스 토큰들을 그룹화할 수 있으며, 이에 대한 토큰 위임 및 폐기를 수행할 수 있다. 하나의 도메인 영역이 제공하는 모든 서비스 토큰들을 위임하거나 폐기하고자 할 경우에 메인 토큰을 폐기하거나 위임한다. 하나의 CaC 토큰에 모든 서비스 토큰 정보를 저장하고 있어, 리소스 서비스의 수가 늘어날수록 토큰의 수는 많아질 수 있으나, 실질적으로 통신 시에는 전체 토큰 정보를 통한 접근제어를 수행하지 않고, 해당 서비스에 대한 토큰 정보만을 검증하여 접근제어를 수행한다. 또한 CaC 토큰에 인증서 토큰을 포함하므로, 한 번의 인증서 토큰 제시로 세션이 유지되는 동안에는 주체에 대한 인증 과정에 대한 처리가 감소한다.

Fig. 4는 Capability 서비스 토큰에 대한 구조를 보여주며, Capability 서비스 토큰은 서비스 ID 서비스 토큰 서명 정보, 리소스 정보, 토큰의 상태, 토큰의 위임 정보를 갖는다.

서비스 토큰은 사용자가 자체적으로 생성할 수 없으며, 서비스 관리자가 등록한 리소스 서비스 정보에 접근할 수 있는 서비스 토큰을 위임받아야 한다. 서비스 토큰 위임 시 사용자는 위임하고자 하는 리소스 토큰의 서비스 ID 정보를 확인하고 해당 서비스 토큰을 위임한다. 서비스 토큰을 위임받은 사용자는 자신의 메인 토큰 정보로 서비스 토큰 정보를 갱신하고 위임 서비스 토큰에 대하여 승인, 거부 의사는 서비스 토큰의 상태 정보 요소를 참조하여 결정한다.

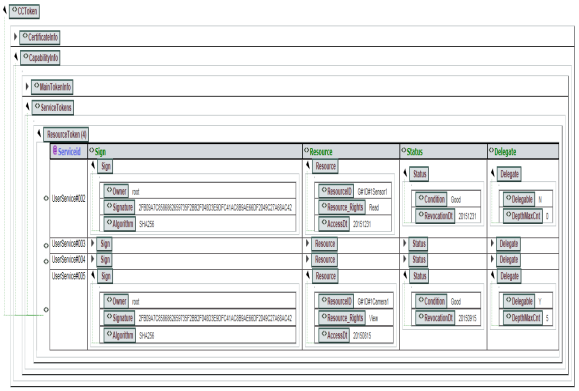

Fig. 5. Example of CaC Token

Fig. 5는 XML로 생성된 CaC Token을 보여주고 있다. 현재 Service Tokens 하부에 4개의 Resource Token이 생성되어있음을 알 수 있다.

3.3 서비스 토큰 관리

1) 서비스 토큰 생성

장치노드 정보와 노드 제어 기능을 제공할 수 있는 리소스 서비스에 대한 관리는 리소스 관리자에 의해 처리된다. 리소스 관리자는 장치노드로부터 수집된 데이터 또는 노드 제어를 위한 서비스를 정의하고 서비스에 대한 리소스 서비스 토큰을 생성한다. 리소스에 대한 접근 권한을 가진 서비스 토큰은 리소스 관리자에 의해 생성되며, 리소스 생성자가 주체에게 위임할 수 있다.

Fig. 6은 서비스 토큰의 생성과정을 보이며 처리 시나리오는 다음과 같다.

a) 리소스 관리자는 장치노드로부터 수집된 데이터 또는 노드 제어를 위한 서비스를 정의하고 Mr.Kim에게 위임할 서비스 토큰에 대해 재위임 여부, 토큰의 유효기간 정보를 지정한다.

b) 인증과정을 거친 Mr. Kim는 관리자에 의해 설정된 리소스 서비스 토큰 정보를 갱신한다.

c) 주체는 발급된 Resource Service #2 서비스 토큰을 사용하여 해당 서비스를 요청한다.

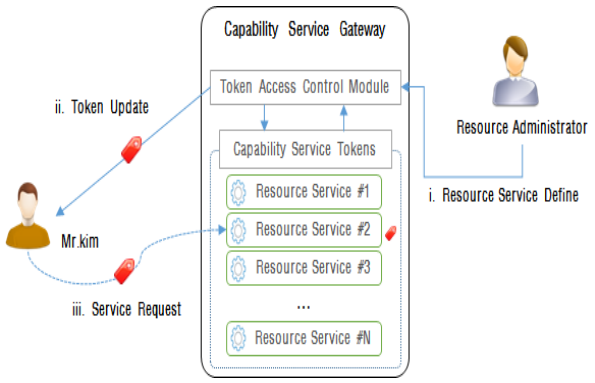

Fig. 6. CSToken Create Process

2) 서비스 토큰 위임 및 재위임

Fig. 7은 서비스 토큰의 위임 과정을 보이고 있으며, 서비스 토큰을 요청한 주체는 리소스 서비스 토큰을 부여받은 후 해당 토큰을 다른 주체에게 위임할 수 있다. 토큰은 재위임 가능 여부와 허가된 위임의 최댓값에 의해 재위임할 수 있다. 재위임 불가한 서비스 토큰을 위임받은 주체가 리소스 서비스에 접근할 경우 해당 서비스 토큰은 취소 목록 리스트에 등록되고 리소스 서비스에 접근할 수 없다.

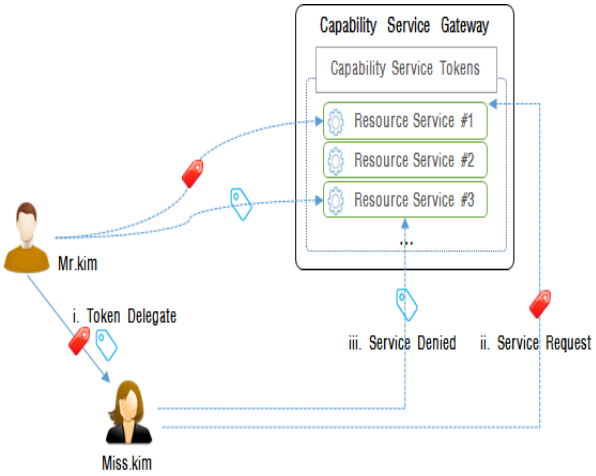

Fig. 7. CSToken Delegation Process

토큰 위임 및 재위임 시나리오는 다음과 같다.

a) Mr.Kim은 Resource Service #1과 Resource Service #3에 대한 토큰을 Miss.Kim에게 위임하였다. Resource Service #1은 위임 가능 토큰이고, Resource Service #3은 재위임 불가 토큰이다.

b) Miss.Kim은 위임받은 Resource Service #1 토큰을 가지고 서비스를 요청하여, 이에 대한 서비스를 받는다.

c) Miss.Kim이 Resource Service #3 토큰을 가지고 서비스를 요청할 경우, 이 서비스는 거부된다.

3) 서비스 토큰 폐기

Fig. 8은 서비스 토큰 폐기 처리 과정을 보이고 있으며, 서비스 토큰을 위임한 주체가 해당 서비스 토큰을 폐기하면 폐기 요청된 서비스 토큰은 TRL에 등록되고 TRL Manager를 통해 폐기 요청 토큰을 소유한 주체의 서비스 토큰 정보를 갱신한다. 폐기 서비스 토큰을 통해 서비스 요청 시 서비스는 거부된다.

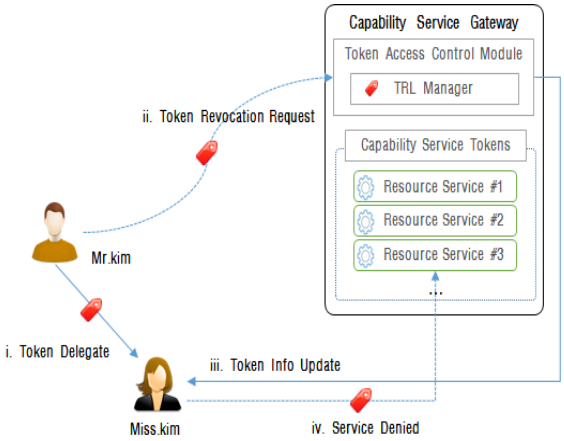

Fig. 8. CSToken Revocation Process

토큰 폐기에 대한 시나리오는 다음과 같다.

a) Mr.Kim은 Resource Service #3에 대한 토큰을 Miss.Kim에게 위임한다.

b) Mr.Kim은 Resource Service #3에 대한 토큰의 위조 가능성을 감지하고, 이에 대한 폐기를 요청한다.

c) 폐기 요청은 TRL 관리자에 의해 처리되어, 토큰 정보가 갱신된다.

d) Miss.Kim이 Resource Service #3 토큰을 가지고 서비스를 요청할 경우, 이 서비스는 거부된다.

3) 서비스 토큰 위임 거절

Fig. 9는 서비스 토큰 위임 거절 처리 과정을 보이고 있다. 주체는 서비스 토큰의 위임을 통해 다른 주체에게 리소스 서비스에 접근할 수 있는 권한을 부여하기도 하지만 필요 이상의 권한을 부여할 경우 위임받은 주체는 위임 토큰에 대해 위임 거절을 할 수 있다. 위임이 거절될 경우 해당 토큰 정보는 토큰 취소 목록 리스트에 등록되고 위임 거절 정보는 위임 주체에게 통보된다.

토큰 위임 거절에 대한 시나리오는 다음과 같다.

a) Mr.Kim은 서비스 토큰을 Miss.Kim에게 위임한다.

b) Miss.Kim은 서비스 토큰의 위임을 거절한다.

c) 위임 거절 정보는 토큰 접근제어 모듈을 통해 Mr.Kim에게 통보된다.

본 논문에서 각각의 서비스 토큰은 생성, 위임, 폐기, 거절될 수 있으며, 만일 CaC 토큰의 소유자가 도메인 영역 내의 서비스 토큰 모두를 위임하거나, 폐기하고자 할 경우에는 CaC 토큰의 메인 토큰 정보를 통해 위임, 폐기할 수 있다.

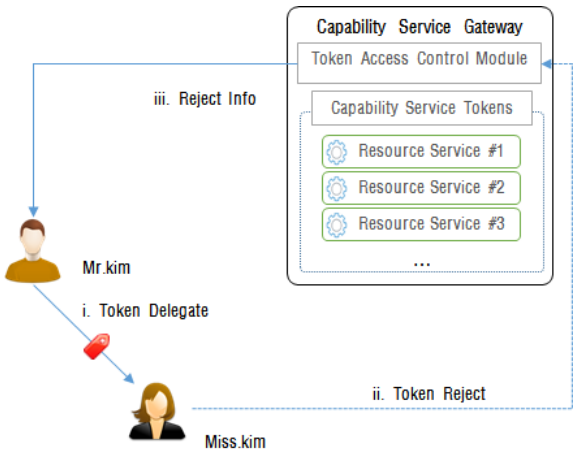

Fig. 9. CSToken Reject Process

3.4 CapSG를 이용한 서비스 요청 시나리오

Fig. 10은 본 논문에서 제안한 플랫폼을 이용한 서비스 요청 시나리오에 대한 흐름을 보이고 있다. 주체가 CaC 토큰의 인증 토큰과 서비스 접근 토큰을 이용하여 리소스 서비스에 접근하는 과정이다. 전체 흐름에 대해 3개의 과정으로 구분하여 보여주고 있으며, [A]는 인증 토큰을 이용한 서비스 요청 및 유효성 확인 과정, [B]는 인증 과정이 종료된 후 세션이 형성된 후, 세션 정보를 가지고 토큰에 대한 정보를 갱신하는 과정, [C]는 요청 서비스에 해당하는 서비스 토큰을 사용하여 서비스에 접근하는 과정이다.

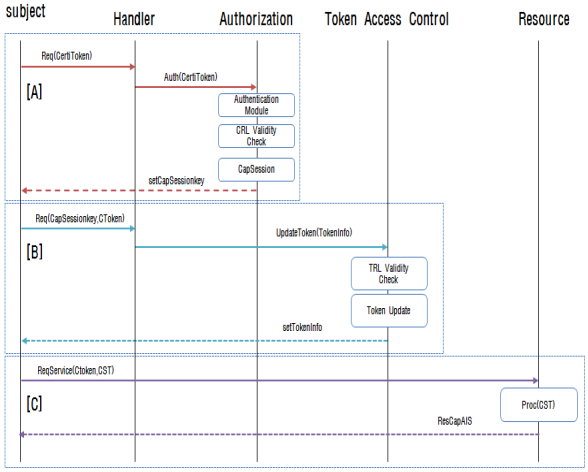

Fig. 10. Service Request Process

[A] 영역의 처리과정은 CaC Token의 인증서 토큰을 이용한 인증과 CapSession 생성 과정으로 다음과 같다.

a) 주체는 CapSG의 요청 핸들러에 개인 비밀키가 포함된 인증토큰 정보를 전송한다.
b) 요청 핸들러는 사용자의 인증 요청에 의해 인증 메소드(Auth)를 이용하여 인증 토큰 정보를 인증 모듈로 전달한다.
c) 인증 모듈은 인증 토큰의 구조와 인증 토큰의 정보로 인증 절차를 거치고 인증서 취소 목록 리스트의 유효성 검증을 거친다.
d) 인증 절차 완료 후 인증 모듈은 CapSession를 통해 세션키를 생성하고 주체에게 재전송한다.

[B] 영역은 인증 토큰 정보를 이용하여 인증 절차를 완료한 사용자가 세션키와 메인 토큰 정보를 이용하여 리소스 서비스 토큰의 정보를 갱신하는 과정이다.

a) 주체는 인증 모듈의 CapSession에서 생성된 세션키와 메인 토큰 정보를 요청 핸들러에 전송한다.
b) 요청 핸들러는 세션키를 통해 연결 상태를 점검하고 주체의 메인 토큰 정보를 Token Access Control Module에 UpdateToken 메소드를 통해 전송한다.
c) Token Access Control Module는 메인토큰의 유효성 검증 후 폐기 또는 유효기간이 만료는 메인 토큰의 경우 주체가 보유한 리소스 서비스 토큰 및 위임한 서비스 토큰을 토큰 취소 리스트에 등록하여 리소스 서비스에 접근할 수 없도록 하며, 위임받은 서비스 토큰이 존재할 경우 setTokeninfo 메소드를 통해 주체에게 리소스 서비스 토큰 정보를 갱신한다.

[C] 영역은 인증을 통한 세션 연결과 리소스 서비스 토큰 정보 갱신이 완료된 상태로 주체는 소유한 리소스 서비스 토큰들 중 요청하고자 하는 리소스 서비스 토큰을 선택하여 리소스 모듈에 요청하는 것이다.

a) 주체는 메인 토큰과 요청하고자 하는 리소스 서비스 토큰을 요청 핸들러에 요청한다.
b) 요청 핸들러는 주체 요청을 리소스 모듈로 전송하여 리소스 서비스 정보를 제공한다.

본 논문에서 주체는 서비스 요청을 위해 [A]-[C]와 같은 인증 및 접근제어 과정을 거치게 되는데 한 번 인증이 된 후 인증 세션이 완료된 후에는 [A], [B] 과정은 생략하고, 원하는 리소스 서비스 토큰을 선택하여 서비스 요청을 하여 서비스를 제공받을 수 있다. 한 번의 인증서 토큰 제시로 세션이 유지되는 동안에는 주체에 대한 인증 과정에 대한 처리가 감소한다.

4. 구현 및 테스트

본 논문에서는 IoT 서비스의 접근제어 플랫폼을 제안하였으며, 제안 아키텍처의 검증을 위하여 전체 구성 모듈에서 토큰 접근제어 모듈과 리소스 모듈을 구현하고 테스트하였다.

구현된 모듈의 검증을 위해 각 모듈에서 정의한 기능이 명확히 수행되는지를 테스트하였으며, 리소스 모듈의 경우는 테스트 환경에 포함된 각 리소스에 대한 정보 등록 및 관리 여부를 테스트하였다. 토큰 접근제어 모듈은 토큰을

이용한 접근제어 수행여부를 테스트하기 위하여 첫째, 각 토큰에 대한 정보 관리 기능 테스트, 둘째, 토큰에 대한 위임 기능 테스트, 그리고 마지막으로 토큰을 사용한 서비스 요청 수행 테스트 항목으로 나누어 진행하였다.

4.1 구현 환경

CapSG 테스트 환경 구축을 위해 조도, 습도, 온도 센서와 영상 정보 취득을 위한 IP카메라를 설치하고 센싱 데이터 처리를 위한 스토리지를 구축하였다. IP카메라의 영상정보는 NVR(Network Video Recorder)을 활용한 웹서비스를 구성하였으며, 구성된 리소스 서비스는 WEB과 CS(Client and Server) 버전으로 분리하여 제공되도록 하였다.

구현을 위한 개발 환경은 Table 2와 같다.

Table 2. System Development Tools

Component	Develoment Tool
WEB Server	Apache 2.x
WAS	Tomcat 1.7.x
DBMS	Oracle 10g
CS(Win Form)	Visual Studio, C#
Web Form	Eclipse, JAVA
DAO	iBATIS

4.2 리소스 관리

Fig. 11은 리소스 관리를 위한 페이지로 리소스 관리자는 Resource Module을 통해 장치노드 등록 및 서비스 도메인에 따른 리소스 게이트웨이를 관리한다. Capability Resource Manager에서는 ServiceID와 Device Serial를 사용하여 시그니처를 생성하고 센싱 데이터 분류, 장치노드의 컨트롤 서비스 타입에 따라 분류한다.

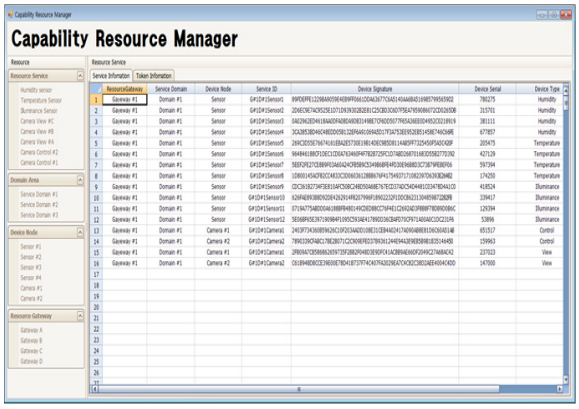

Fig. 11. Resource Manager

4.3 토큰 정보 관리

Fig. 12는 토큰 정보 관리 화면으로 리소스 서비스 정보와 보유한 서비스 토큰의 요소정보를 확인하고, Camera View #A 서비스 토큰을 이용하여 1번 카메라 영상에 접근하여 영상 제공 서비스를 확인한 것이다. 주체는 서비스에 접근하고자 인증서를 통해 로그인 후 CapSession을 이용하여 리소스 서비스에 접근한다. 접근한 주체는 자신이 보유한 리소스 서비스 정보를 확인하고, 서비스 토큰을 이용하여 리소스 서비스에 접근한다.

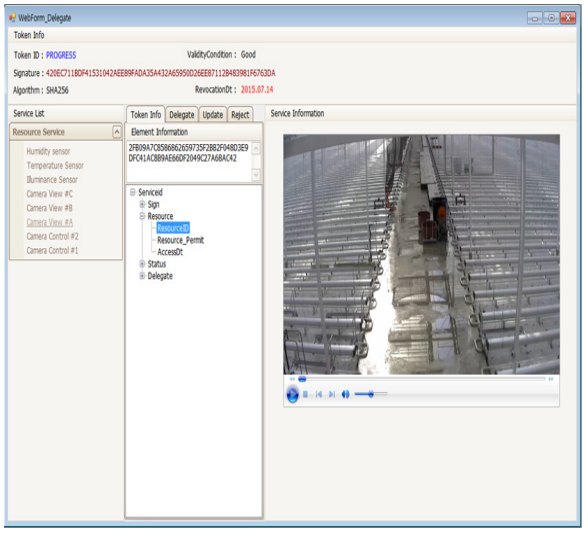

Fig. 12. Token Information of Camera View Service

4.4 토큰 위임 관리

Fig. 13은 토큰에 대한 위임 관리 페이지로 주체는 자신이 보유한 리소스 서비스를 다른 주체에게 위임할 수 있다. 위임 시 주체는 위임 주체의 메인 토큰 정보를 조회하여 리소스 서비스 토큰의 재위임 권한 및 유효기간을 설정할 수 있다.

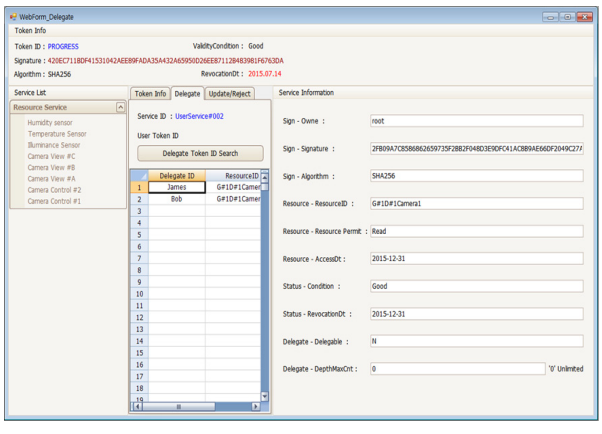

Fig. 13. Delegation of Capability Service Token

4.5 서비스 요청 관리 로그

Fig. 14는 서비스 요청이 이루어졌을 때 시스템의 로그 정보를 보여준다. 주체의 접속 정보와 요청 메시지, 리소스 서비스 상태, 센싱 데이터의 실시간 데이터를 확인하고 센

싱 데이터를 이용하여 제공할 수 있는 서비스를 분석할 수 있도록 메시지 모니터링을 통해 실시간으로 확인 가능하다.

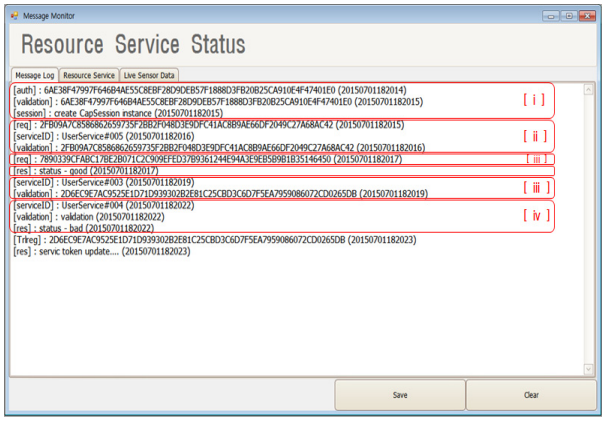

Fig. 14. Monitoring of Service Request

본 실험에서는 한 번의 인증과 세 개의 서비스에 대한 요청을 수행하였으며, 로그 정보에서 [i]는 주체에 대한 인증 처리 과정을 보이고 있다. 인증서 정보에 대한 검증을 거치면, CapSession 인스턴스가 생성된다. [ii], [iii], [iv]는 서비스 요청 및 검증 과정, 요청 결과를 보이고 있으며, 세션이 유지되는 동안에는 인증 과정은 생략되고 있음을 알 수 있다.

본 논문에서는, 실험 결과에서 본 바와 같이 주체가 서비스 요청을 위해 인증을 수행하고, 인증 세션이 연결된 후에는 인증 과정을 생략하고, 원하는 리소스 서비스 토큰을 선택하여 서비스 요청을 하여 서비스를 제공받을 수 있다. 한 번의 인증서 토큰 제시로 세션이 유지되는 동안에는 주체에 대한 인증 과정에 대한 처리가 감소한다.

5. 결 론

본 논문은 IoT 환경의 다양한 서비스에 대한 효율적인 접근제어를 제공하기 위하여 Capability Service Gateway (CapSG)를 이용한 IoT 서비스 접근제어 플랫폼을 제안하였다. 제안한 IoT 서비스 접근제어 플랫폼은 현재도 계속 개발 진행 중이며, 시스템의 검증을 위하여 본 논문에서는 IoT 서비스 접근제어 플랫폼의 한 요소인 CapSG에서 토큰 접근제어 및 리소스 모듈을 구현하고 테스트하였다.

CapSG는 인증과 접근 권한을 포함하는 토큰을 사용하여, 서비스 주체에 대한 인증 및 접근제어를 수행하여 서비스를 제공한다. 또한, 서비스 토큰 관리를 위한 생성, 위임, 폐기, 거절 등의 기능을 제공한다.

각 도메인에 대한 토큰 그룹을 사용함으로써 도메인별 서비스 접근제어가 가능하며, 토큰 그룹 내의 특정 서비스 토큰을 이용한 접근제어 수행도 가능하도록 설계하여 IoT의 다양한 서비스 요구에 대한 접근제어의 유연성과 효율성을 제공한다. 인증 및 접근제어는 CapSG에 의해 수행되며, 도메인별 서비스 접근제어를 수행하기 때문에 IoT 환경의 다양한 IoT 장치 관리 및 확장성이 용이하다.

토큰은 인증정보 및 접근제어를 포함하고 있으며, 토큰 내에 인증 정보 및 접근제어가 포함되어있으므로, 서비스 요청 시 매번 장치 및 사용자에 대한 인증 절차 없이 한 번의 인증으로 세션을 유지하여, 해당 서비스에 대한 서비스 토큰만으로 접근제어를 수행할 수 있다. 이는 기존 시스템에 비해 인증에 대한 처리 과정을 단축하며, 서비스 요청 시 서비스 토큰만을 제시함으로써 통신되는 메시지의 양을 감소할 수 있다.

본 연구에서는 인증 및 접근제어를 위해 인증 토큰 및 서비스 토큰을 사용하고 있으며, 이에 대한 폐기 목록을 유지하여 토큰에 대한 신뢰성을 제공하고자 하였다. 현재는 플랫폼 내에서 자체 CRL과 TRL을 유지하고 있지만, 추후 연구를 통해 CRL 및 TRL의 신뢰성을 보장할 수 있는 제3의 인증기관(trusted third party)에 대한 문제가 개선되어야 한다. 또한, 현재는 하나의 도메인 영역에 대하여 접근제어 수행을 진행하였으나, 추후 리소스 관리를 통한 멀티 도메인 관리 기법에 대한 연구도 진행할 예정이다.

제안한 아키텍처의 전체 모듈 중 토큰 접근제어 모듈과 리소스 모듈만이 구현되어있어, 전체 시스템에 대한 타당성 검증 대신에 각 구현 모듈에 대한 테스트를 통해 제안한 기능들이 정확하게 수행되고 있음을 검증하였으며, 추후 제안한 IoT 서비스 플랫폼의 개발이 완료되는 시점에서 검증 모델을 선정하여 제안한 시스템에 대한 검증을 수행할 것이다.

References

[1] L Sancheza, L Muñoza, Jose Antonio Galachea, P Sotresa, J R. Santanaa, V Gutierreza, R Ramdhanyb, A Gluhakc, S Krcod, E Theodoridise, and D Pfistererf, “SmartSantander: IoT experimentation over a smart city testbed,” *Computer Networks*, Vol.61, pp.217-238, 2014.

[2] L Atzoria, A Ierab, and G Morabito, “The internet of things: A survey,” *Computer Networks*, Vol. 4, Issue.15, pp.2787-2805, 2010.

[3] Rolf H. Weber, “Internet of Things - New security and privacy challenges,” *Computer Law & Security Review*, Vol. 26, Issue.1, pp.23-30, 2010.

[4] S. Gusmeroli, S. Piccione, and D. Rotondi, “IoT access control issues: a capability based approach,” *IMIS-2012*, pp.787-792, 2012.

[5] Bum-Ki Lee, Mi-Sun Kim, and Jae-Hyun Seo, “Design and Implementation of The Capability Token based Access Control System in the Internet of Things,” *Journal of The Korea Institute of Information Security & Cryptology*, Vol.25, No.2, pp.439-448, 2015.

[6] Romuald Thion, “Access Control Models,” in *Cyber Warfare and Cyber Terrorism,* Hershey, pp.318-326, 2008.

[7] Pierangela Samarati and Sabrina De Capitani di Vimercati, “Access Control: Policies, Models, and Mechanisms,” in *Foundations of Security Analysis and Design*, pp.137-196, 2001.
[8] Chao Lee, Yunchuan Guo, and Lihua Yin, “A Location Temporal based Access Control Model for IoTs,” *AASRI Procedia*, Vol.5, pp.15-20, 2013.
[9] Sergio Gusmeroli, Salvatore Piccione, and Domenico Rotondi, “A capability-based security approach to manage access control in the Internet of Things,” *Mathematical and Computer Modelling*, Vol.58, pp.1189-1205, 2013.
[10] Jos´e L. Hern´andez-Ramos, Antonio J. Jara, Leandro Mar´ın, and Antonio F. Skarmeta1, “Distributed Capability-based Access Control for the Internet of Things,” *Journal of Internet Services and Information Security*, Vol.3, Num.3/4, pp.1-16, 2013.
[11] L. Fang, D. Gannon, and F. Siebenlist, “XPOLA－an extensible capability based authorization infrastructure for grids,” *4th Annual PKI R&D Workshop*, pp.30-40, 2005.
[12] Mark S. Miller, Ka-Ping Yee, and J. Shapiro, “Capability Myths Demolished,” Systems Research Laboratory, Johns Hopkins University, *Tech.Report SRL 2003-02*, 2003.

김 진 보
e-mail : progress97@mokpo.ac.kr
2003년 목포대학교 멀티미디어학과(학사)
2007년 목포대학교 정보보호기술학협동과정(석사)
2008년～현 재 목포대학교 정보보호기술학협동과정 박사과정
관심분야 : 정보보호, 웹서비스 보안, 빅데이터, 프로그래밍 언어

장데레사
e-mail : jdrs0405@mokpo.ac.kr
2015년 목포대학교 정보보호학과(학사)
2015년～현 재 목포대학교 정보보호기술학협동과정 석사과정
관심분야 : 네트워크 보안, 프로그래밍 언어, 모바일 네트워크 보안

김 미 선
e-mail : misun@mokpo.ac.kr
1996년 목포대학교 컴퓨터공학과(학사)
2000년 목포대학교 컴퓨터공학과(석사)
2012년～현 재 목포대학교 정보보호학과 초빙교수
관심분야 : 정보보호, 프로그래밍 언어, 컴퓨터 네트워크, 모바일 시스템 보안

서 재 현
e-mail : jhseo@mokpo.ac.kr
1985년 전남대학교 계산통계학과(학사)
1988년 중앙대학교 전자계산학과(석사)
1996년 전남대학교 전산통계학과(박사)
1996년～현 재 목포대학교 정보보호학과 교수
관심분야 : 정보보호, 시스템 및 네트워크보안, 컴퓨터 네크워크, 모바일 네트워크 보안