KIPS Tr. Comp. and Comm. Sys.
Vol.7, No.7 pp.183~194 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2018.7.7.183

Analysis of IoT Open-Platform Cryptographic Technology and Security Requirements

Choi Jung-In† · Oh Yoon-Seok†† · Kim Do-won††† · Choi Eun Young†††† · Seo Seung-Hyun†††††

ABSTRACT

With the rapid development of IoT(Internet of Things) technology, various convenient services such as smart home and smart city have been realized. However, IoT devices in unmanned environments are exposed to various security threats including eavesdropping and data forgery, information leakage due to unauthorized access. To build a secure IoT environment, it is necessary to use proper cryptographic technologies to IoT devices. But, it is impossible to apply the technologies applied in the existing IT environment, due to the limited resources of the IoT devices. In this paper, we survey the classification of IoT devices according to the performance and analyze the security requirements for IoT devices. Also we survey and analyze the use of cryptographic technologies in the current status of IoT open standard platform such as AllJoyn, oneM2M, IoTivity. Based on the research of cryptographic usage, we examine whether each platform satisfies security requirements. Each IoT open platform provides cryptographic technology for supporting security services such as confidentiality, integrity, authentication an authorization. However, resource constrained IoT devices such as blood pressure monitoring sensors are difficult to apply existing cryptographic techniques. Thus, it is necessary to study cryptographic technologies for power-limited and resource constrained IoT devices in unattended environments.

Keywords : IoT Device Class, IoT Security Requirements, IoT Platform, Cryptography

IoT 오픈 플랫폼 암호기술 현황 및 보안 요구사항 분석

최 정 인† · 오 윤 석†† · 김 도 원††† · 최 은 영†††† · 서 승 현†††††

요 약

IoT 기술의 급격한 발전으로 스마트홈이나 스마트 시티와 같은 다양한 편리한 서비스들이 실현되었다. 그러나 무인 환경에서의 IoT 기기는 도청 및 데이터 위조, 무단 액세스로 인한 정보 누출 등 다양한 보안 위협에 노출되어 있다. 안전한 IoT 환경을 구축하려면 IoT 기기에 적절한 암호화 기술을 사용해야 한다. 그러나 IoT 기기의 제한된 자원으로 인해 기존 IT 환경에 적용된 암호화 기술을 그대로 적용하는 것은 불가능하다. 본 논문에서는 성능에 따라 IoT 디바이스의 분류를 조사하고 IoT 디바이스의 보안 요구 사항을 분석한다. 또한 AllJoyn, oneM2M, IoTivity와 같은 IoT 개방형 표준 플랫폼의 현재 암호화 기술의 사용 현황을 조사하고 분석한다. 암호화 기술 사용 현황에 대한 연구를 기반으로 각 플랫폼이 보안 요구사항을 만족하는지 확인한다. 각 IoT 개방형 플랫폼은 기밀성, 무결성, 인증 및 인증과 같은 보안 서비스를 지원하기위한 암호화 기술을 제공한다. 하지만 혈압 모니터링 센서와 같은 자원이 제한된 IoT 장치는 기존의 암호화 기법을 적용하기가 어렵다. 따라서 무인 환경에서 전력 제한 및 자원 제약을 받는 IoT 장치에 대한 암호화 기술을 연구 할 필요가 있다.

키워드 : IoT 기기 등급, IoT 보안 요구사항, IoT 플랫폼, 암호기술

1. 서 론

Machina 리서치 기관에서 세계 사물인터넷 시장은 2022년까지 연평균 21.8% 성장률을 보이며 1조2000억 달러 규모까지 성장할 것이라 전망하였다[1]. 이처럼 현재 IoT는 수직성장하며 급격하게 발전하였지만 개인정보 침해나 보안 위협의 위험성은 크게 고려하지 않았으며 기술 발전 위주로 연구 및 개발되어 왔다.

IoT 환경의 모든 기기들은 인터넷에 연결되어 있으며 개방된 장소에 설치되어 비인가적인 접근으로 인한 정보유출 및 파손의 위험이 있다. 또한 IoT 오픈 플랫폼은 이기종 단말 및 유·무선 네트워크 간에 연동을 기반으로 한다. 그리하여 보안이 취약한 통신구간에서의 도청 및 데이터 위변조 등과 같은 다양한 형태의 보안위협에 노출되고 있다. 네트워크를

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2017-0-00267).
† 준 회 원 : 부산대학교 SW교육센터 강의전담
†† 준 회 원 : 한양대학교 전자공학과 석사과정
††† 비 회 원 : 한국인터넷진흥원 주임연구원
†††† 정 회 원 : 한국인터넷진흥원 책임연구원
††††† 종신회원 : 한양대학교 전자공학과 부교수
Manuscript Received : February 26, 2018
First Revision : April 10, 2018
Accepted : April 15, 2018
* Corresponding Author : Seung-Hyun Seo(seosh77@hanyang.ac.kr)

통한 보안 위협은 개인정보 침해 문제로 연결된다. IP카메라가 해킹되어 인터넷에 연결된 IP카메라가 전 세계에 생중계되었으며 IP카메라에서 모르는 익명의 목소리가 흘러나오는 등의 사례가 쏟아져 나오고 있다[2]. 이미 IoT 기기를 통한 보안 위협은 현실화되었으며 이는 IoT 서비스가 보편화될수록 더 크게 증가할 것으로 예상된다[3]. Gartner에서는 2020년 전 세계 사이버공격의 25%는 IoT 기기를 대상으로 발생할 것으로 예측하고 있다[4].

하지만 이러한 보안위협에도 네트워크에 연결 가능한 IoT 기기의 70%가 수집된 정보를 암호화되지 않은 상태로 클라우드 컴퓨팅 시스템이나 로컬 네트워크에 전송한다. 또한 IoT 기기의 60%는 보안에 취약한 웹 인터페이스를 적용하고 있다. 소프트웨어 업데이트 시에도 60%가 암호화를 사용 하지 않는 등 암호화나 사용자 접근 권한 등에 있어 취약점을 갖고 있다[5]. IoT 기기들은 경량으로 제한된 자원(메모리, 전력 등)으로 구성되어 있기 때문에 기존 IT 환경에서의 다양한 보안 기술을 그대로 적용할 수 없다.

본 논문에서는 보안 표준을 기반으로 하여 개발된 주요 IoT 오픈 플랫폼의 암호기술 현황을 분석한다. IoT 플랫폼을 주도해나가는 표준 단체들이 늘어남에 따라 다양한 IoT 오픈 플랫폼이 개발되고 연구되고 있다. 대표적인 IoT 오픈 플랫폼으로는 AllJoyn, oneM2M, IoTivity가 있다. 먼저 IoT 기기의 등급 분류와 보안 요구사항에 대해 알아보고 현재 활용되고 있는 다양한 IoT 오픈 플랫폼의 암호기술 현황을 분석한다. 대표적인 IoT 오픈 플랫폼들에 내장되어 있거나 제공 중인 암호기술을 조사하고 나아가 앞서 분석한 IoT 플랫폼들의 암호기술을 비교 분석한다.

2. IoT 기기 등급 분류 및 보안 요구사항

2.1 IoT 기기 등급 분류

국내 TTA 표준의 “TTAK.KO-12.0298 사물인터넷 기기 등급 분류 및 보안 요구사항”에 따르면, IoT 기기는 기기의 리소스와 프로세싱 능력과 같은 역량에 따라 등급 0~3으로 구분될 수 있다[6].

등급 0은 메모리 및 프로세싱 능력에 제약이 많은 초소형, 초경량, 초절전 센서와 같은 기기이다. 이 기기들은 통신에 필요한 리소스를 가지고 있지 않아 안전한 방법으로 직접 인터넷 통신을 하지 못한다. 게이트웨이, 프락시와 같은 기기의 도움으로 인터넷 통신에 참여할 수 있어 관리 목적으로 Keep Alive 시그널에 응답하거나 상태정보를 보내는 정도만 가능하다. 일반적으로 안전하게 관리되지 못하기 때문에 사전에 아주 작은 용량의 설정 파일을 통해 관리한다.

등급 1은 리소스 및 프로세싱 능력에 제한을 가지고 있는 기기이다. 기존의 통신 프로토콜 스택을 사용하기는 힘들지만 CoAP(Constrained Application Protocol)과 같이 IoT 기기를 위해 설계된 프로토콜 스택은 사용할 수 있다. 대표적인 기기로는 8/16 bit 프로세서를 기반으로 하는 혈당측정기와 같은 의료헬스 기기, 온도조절기와 같은 스마트홈 기기 등이 있다. 게이트웨이의 도움 없이 통신이 가능하며, 통신 네트워크에서 요구하는 보안 기능 지원이 가능하다.

등급 2는 리소스 제약이 거의 없으며 기존의 통신 프로토콜 스택을 사용할 수 있는 기기이다. 대표적으로는 32 bit 프로세서를 기반으로 하는 IP 카메라나 스마트 미터기 등이 있다. 제약을 거의 받지 않지만 등급 2 기기에서 경량화된 프로토콜 스택을 사용하는 이유는 개발 단가를 낮추고, 다른 기기와의 상호 운용성을 증가시키기 위함이다.

등급 3은 스마트폰이나 태블릿 같은 기기이다. 대부분 변경이나 수정을 요구하지 않고 기존의 프로토콜을 그대로 사용할 수 있다. 이 기기들은 리소스 및 프로세싱 능력에 별다른 제약이 존재하지 않으나, 전원 공급에 대한 제약은 여전히 존재한다. Table 1은 IoT 기기 등급 별 CPU 클럭, 데이터 크기, 코드 크기와 사용 운영체제를 정리한 것이다.

Table 1. Classification according to Capabilities of IoT

	CPU clock	Data size	Code size	OS
Class 0	<< 10MHz	<< 10KB	<< 100KB	Firmware
Class 1	~ 100MHz	~ 50KB	~ 250KB	TinyOS, Contiki, RIOT, NanoQplus
Class 2	~ 500MHz	~ 250KB	~ 1MB	Embedded Linux
Class 3	>> 1GHz	>> 1MB	>> 5MB	Android, iOS, Tizen

IoT 기기에서 주로 사용하는 대표적인 경량 프로세서로는 Atmega128(8bit), MSP430(16bit), ARM-Cortex A(32bit)가 있으며 Table 2에서 이들의 특성을 비교하였다[7].

Table 2. Comparison of IoT Device Processor [7]

	Atmega128	MSP430	ARM-Cortex A
Data width	8bit	16bit	32bit
General-purpose registers	32	12/16	8/13/16
Number of instructions	61	27	56
Core area	6140GE	4913GE	-
Applications	Arduino, Micaz	TelosB	Beagle, Odroid
IoT device	Smart Plug, Smart light bulb	Clothing, Healthcare, Smart shoes	Automatic temperature regulator, Home gateway, Smart watch

Atmegal28과 MSP430은 IoT 디바이스 등급 1이며 ARM-Cortex A는 등급 2이다. 등급 0은 저전력 센서와 같은 기기이며 등급 3은 경량 암호기술을 사용하지 않아도 되는 높은 성능을 가진 기기들이다.

2.2 IoT 기기 보안 위협에 따른 요구사항

IoT 환경의 모든 기기들은 비인가적인 접근으로 인한 정보 유출 및 파손의 위험이 있다. 또한 IoT 오픈 플랫폼에 따른 이기종 단말 및 유·무선 네트워크 간 연동으로 인해 보안이 취약한 통신구간에서의 도청 및 데이터 위변조 등 기존 IT 환경보다 다양한 형태의 보안위협에 노출되고 있다. 다양한 보안 위협으로 정보 보안의 3대 요소인 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)이 침해될 가능성이 높아지고 있다[8]. 이 외에도 인증 및 인가 관련 필요성도 높아지고 있다. 보안 위협으로부터 IoT 기기를 안전하게 지키기 위한 보안 요구사항은 다음과 같다.

1) 기밀성 관련 보안 요구사항

IoT 기기 간 전송되는 메시지는 불법적인 스니핑(sniffing) 또는 도청 방지를 위해 암호화된 형태로 전송되어야 한다. 정보유출 방지를 위해 웜, 바이러스와 같은 악성코드 감염이나 외부 해킹 공격을 탐지하고 방어할 수 있는 기능을 제공해야 한다. 정보유출 방지를 위해 개인정보 및 암호 키와 같은 중요 데이터를 암호화하여 안전하게 처리 및 저장 관리하여야 한다. 물리적 공격으로부터 안전성과 신뢰성을 보장하기 위해 부당 변경 방지(Tamper Resistance) 기능을 제공해야 한다. 기기 복제 방지를 위해 기기 고유의 식별정보가 외부로 유출되거나 변경되지 않도록 안전하게 처리 및 관리해야 한다.

2) 무결성 관련 보안 요구사항

IoT 기기는 데이터 위변조 방지를 위해 데이터 무결성 검증 기능을 제공해야 한다. 또한 펌웨어, 운영체제와 같은 시스템 레벨의 기기 플랫폼 무결성 검증 기능을 제공해야 한다. 더불어 부팅 시 소프트웨어에 대한 무결성 검증, 인가된 소프트웨어만이 로드되고 실행되어 기기의 신뢰성을 보장하는 시큐어 부팅(Secure Booting) 기능을 제공해야 한다.

3) 가용성 관련 보안 요구사항

IoT 기기는 사용자, 시스템, 보안 이벤트에 대한 적절한 로그 기능을 제공해야 한다. 물리적 제거/파괴 및 비정상적인 설치 시도 방지를 위해 주기적인 Keep Alive 메시지 전송 또는 기기 상태 정보 전송 기능을 제공해야 한다. 외부 공격자의 지속적인 접속 시도 및 서비스 요청에 대한 서비스 거부 공격과 같은 외부 공격에 대응할 수 있는 기능을 제공해야 한다. 도난이나 분실, 설치 및 폐기 등에 적절히 대응하기 위한 보안 모니터링 및 보안 관리 기능이 제공되어야 한다. 안전한 소프트웨어 업데이트 및 보안 패치 기능을 제공해야 한다. 다양한 형태의 기기에 쉽고 적절한 보안 정책을 안전하게 설정할 수 있는 기능을 제공해야 한다. 소프트웨어 오류나 악성코드 감염에 의한 오동작 시에도 해당 모듈 분리 및 제거, 접근 권한 제한 등의 기능을 통해 소프트웨어 안전성을 보장해야 한다.

4) 인증/인가 관련 보안 요구사항

IoT 기기는 비인가된 사용자의 접근을 차단하기 위해 사용자 인증 기능을 제공해야 한다. 불법적인 기기의 접근을 차단하기 위해 기기 인증 기능을 제공하고 안전한 비밀번호를 설정하고, 주기적으로 업데이트를 제공해야 한다. 안전하고 자율적인 통신 환경 구축을 위해 기기 간 상호 인증 기능을 제공해야 한다. 정보유출 방지 및 프라이버시 보호를 위해 소유권 제어와 같은 권한 제어 및 설정 기능을 제공해야 한다. 불법적인 사용자 및 기기의 접근을 차단하는 접근 제어 기능을 제공해야 한다. 기기 복제, 변경, 도용을 방지하기 위한 기기 고유 식별 정보 검증 기능을 제공해야 한다.

3. AllJoyn 플랫폼 암호기술 현황 분석

AllJoyn 플랫폼[9]은 퀄컴사(Qualcomm)가 스마트홈에서 기기 간에 미디어 콘텐츠를 공유하는 기술인 DLNA (Digital Living Network Alliance)을 개선하여 2011년에 개발한 오픈 소스이다. IoT의 모든 사물이 상호작용할 수 있도록 IoT 제어 오픈 소스 플랫폼을 목표로 하여 개발되었다. 플랫폼 관리를 위하여 퀄컴사를 중심으로 100여개의 국내외 기업들이 멤버로 참여하여 AllSeen Alliance 표준 단체를 조직하였다. 하지만, 현재 AllSeen Alliance는 OCF로 합병되어 AllJoyn 플랫폼을 관리하고 있다[10].

AllJoyn 플랫폼은 물리 계층, 라우터 계층, 클라이언트 라이브러리 계층, 서비스 프레임워크 계층, 어플리케이션 계층으로 구성되어있다. 근거리 기반의 기기 간 P2P(Peer-To-Peer) 통신이며, 중계서버를 사용하지 않고, 기기 간의 통신을 사용한다. Bluetooth나 Wi-Fi 등의 물리적인 통신방식 위에 소프트웨어 프레임워크로 개발된 통신을 이용하기 때문에, 하드웨어에 의존적이지 않다. AllJoyn은 기기 간의 세션 연결을 위해서 RMI(Remote Method Invocation)방식의 D-Bus를 이용한다. AllJoyn 프레임워크는 로컬 네트워크에서 실행된다[11, 12].

AllJoyn 플랫폼은 기본적으로 Windows, iOS/OS X, 리눅스 등의 운영체제와 안드로이드와 같은 스마트폰 운영체제를 지원한다. 또한 IoT 기기에 내장되어있는 Thin-Linux, Thin-Windows와 아두이노를 지원한다.

3.1 AllJoyn 보안 구조

AllJoyn은 IoT 환경에서의 보안 성능 강화를 위해, 장치 간의 인증 및 암호화된 데이터 통신을 위한 보안 프레임워크를 제공한다[13]. 단대단 (end-to-end) 애플리케이션 레벨의 보안성을 제공하며, 인증과 데이터의 암호화는 애플리케이션 영역에서 수행한다. AllJoyn은 통신보안을 위해서 TLS 1.2버전을 사용한다. Fig. 1은 Security 2.0 Architecture를 보여준다. IoT 시스템을 보안기능을 구축을 할 때 시스템을 어떻게 설계하고 구성하는지와 기기들 간의 인증 및 키 교환 방식을 설명하고 있다[14]. Fig. 1은 AllJoyn에서 공개한 자료[14]를 재구성하였다.

Security 2.0의 기능은 Core 라이브러리의 일부이다. 응용 프로그램에 대해 정의된 접근제어목록(ACLs)기반으로 응용

프로그램 수행 방법을 정의할 수 있다. 보안 매니저(SM)는 사용자에 키 관리와 사용 권한 규칙을 구축하게 도와주는 서비스다. 보안 매니저는 개발자가 정의한 응용프로그램 매니페스트 템플릿을 통하여 최종 사용자가 승인 할 수 있는 접근제어목록으로 구성된 매니페스트를 작성한다. Security 2.0 권한 모듈은 기기 간의 암호화 메시지 외에도 접근 자격 증명 및 접근 제어 목록의 데이터베이스를 관리한다.

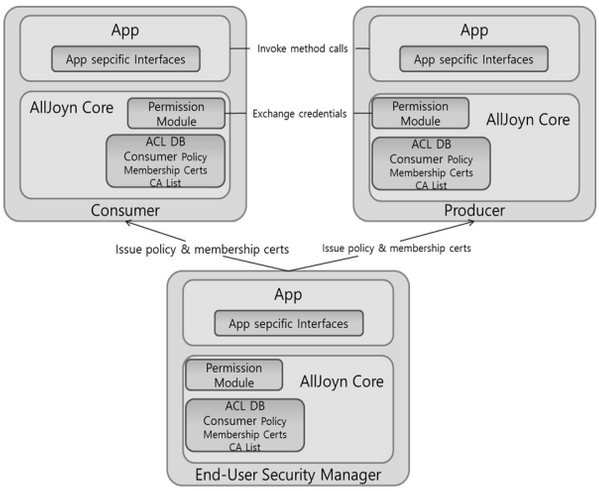


Fig. 1. AllJoyn Security 2.0 Architecture

보안 채널 형성은 3단계로 이루어져있다. 첫 번째, GUID (Globally Unique IDentifier) 교환에서는 그룹마다 정해진 GUID를 비교하여 키 공유가 가능한 기기 인지 확인하다. 두 번째, 키 교환에서는 상대 기기를 인증하고 마스터 키를 생성하고 키 저장소에 저장한다. 세 번째, 세션키 생성에서는 넌스(nonce) 와 마스터 키를 통해 세션키를 생성한다.

1) 암호키 생성을 위한 키 교환

AllJoyn은 키 교환을 위한 보안 채널로는 TLS 1.2, 키 교환 방식으로는 ECDHE(Elliptic curve Diffie - Hellman Ephemeral)를 사용한다. ECDHE는 통신보안을 위해 통신하는 메시지 암호화에 필요한 비밀키를 생성한다. 키 교환 시 인증방법은 NULL, PSK, ECDSA 방법을 사용한다. NULL은 기기 간 별도의 인증절차가 없기 때문에, 중간자 공격에 취약하다. PSK (PreShared Key) 방법은 기기끼리 사전에 공유된 키를 가지고 인증하는 방식으로 사전 공유된 키인 PSK의 길이가 길고 높은 엔트로피 일 경우 안전하다. ECDSA(Elliptic Curve Digital Signature Algorithm) 방법은 기기 간에 전자 서명을 통하여 상대 기기를 인증하는 방식으로, 상대 기기의 인증서 유효성을 검사한다. 키 교환 프로토콜을 구성하는 알고리즘 ECDHE는 NIST SP800-56A에서 정의한 알고리즘을 이용하고, ECDSA는 FIPS 186-4, ANSI X9.62에서 정한 알고리즘을 이용한다. 타원곡선 파라미터로는 256bit NIST curve P256을 사용한다.

2) 인증서

인증서는 표준 X.509 v3 인증서(RFC 5280)를 사용한다. 인증서에서 사용하는 서명은 타원 곡선 파라미터 P256을 사용한 ECDSA이고, 암호학적 해쉬 알고리즘은 SHA256을 이용하며, PEM 인증서 인코딩을 지원한다. 특정 EKU (Extended Key Usage)들은 인증서 유형을 나타낸다. 인증서 폐지는 구현되지 않았다.

3) 데이터 전송에서의 암호화

암호화는 RFC 3610에 나와 있는 AES-128-CCM (Counter with CBC- MAC) 모드를 사용한다. AllJoyn 플랫폼은 타 IoT 오픈 플랫폼과 다르게 128bit 암호화만을 사용하고 있다. 인증 태그 역시 128bit 암호화를 이용한다.

4) 해쉬 및 키 생성

해쉬 함수는 SHA256을 사용한다. 단 SRP는 SHA-1을 이용한다. ECDSA와 인증서 내에서도 SHA256를 이용한다. 키 유도함수는 TLS PRF(RFC 5246, Section5)를 이용한다.

5) 보안 레벨

128bit 보안 수준을 충족해야 한다. 128bit의 암호화 키와 인증 태그를 가져야 하고 256bit 해쉬 digest 와 타원 곡선 파라미터를 설정해야한다. 난수 발생기 역시 같은 수준에 보안 수준을 가지고 있어야 한다.

3.2 AllJoyn 보안을 사용할 때 고려사항

Security 1.0 앱은 반드시 지원하는 인증 메커니즘을 선택하고, 어떻게 인증을 부트트래핑하고 비밀키을 제공할지를 결정해야한다. Security 2.0 앱은 보안 매니저를 이용해야 하며, 보안매니저와 처음 연결 시 NULL 또는 PSK 인증을 선택해야한다. 인증서의 유효 검사를 위해 Security 1.0 앱은 callback으로 인증서 체인을 검증해야 하며 Security 2.0 앱은 부트트래핑 후 인증서를 이용한다. 보안매니저는 관리자 인증서에 필요한 것들을 설치한다. 보안 키 저장소에 관련하여 코어 라이브러리는 암호 키들을 키 저장소에 보관한다. AllJoyn은 플랫폼이기 때문에 보호 장치에 대한 내용이 부족하다. 그리하여 안전한 저장을 위해 하드웨어기반의 방법을 활용할 수 있다.

표준 클라이언트 앱은 암호기능을 구현할 필요가 없다. 인증된 안전한 채널은 코어 라이브러리를 통해 구현 할 수 있다. 플랫폼에서 제공하는 암호구현기능을(CNG, OpenSSL) 사용할 수 있다. 의존성이 없는 built-in 옵션도 제공한다. 암호는 엔트로피 소스가 필요하다. 특히 경량 디바이스에서 엔트로피 소스 입력은 중요하다. DRBG구현은 엔트로피가 높은 seed를 필요로 한다. Built-in 구현은 부채널 분석 공격에 대한 보호 방법을 포함한다. 예를 들어, 실행시간과 메모리 접근 패턴이 비밀 키와 무관하게 함으로써 부채널 분석 공격에 대응한다.

4. oneM2M 플랫폼 암호기술 현황 분석

oneM2M은 2012년 7월, 글로벌 IoT 서비스 플랫폼 표준 개발을 위해 ETSI(유럽), TIA(미국), ATIS(북미), ARIB(일본), TTC(일본), CCSA(중국), TTA(한국) 7개의 세계 주요 표준화 단체가 공동으로 설립한 M2M공통 플랫폼 글로벌 표준개발 협력체이다[15]. 2016년에 인도의 표준화기구인 TSDSI가 가입되어 현재 8개의 표준화 기구가 활동하고 있다. M2M은 Machine to Machine의 약자로서 무선 통신을 이용한 기기간의 통신을 의미한다. oneM2M은 다양한 서비스의 요구사항을 만족시킬 수 있는 IoT 공통 플랫폼을 정의하고, 타 IoT 플랫폼과의 상호운영을 위한 방법 및 기능을 표준화하였다. 다양한 어플리케이션 간의 호환을 위한 인터페이스를 정의하고 수평적인 플랫폼을 구성하였다[16]. 스마트홈, 스마트 카, 에너지, 헬스케어, 엔터프라이즈, 공공 서비스와 같은 7개 IoT 산업 분야의 서비스 제공을 위한 요구사항을 도출하고, 핵심 기능(데이터수집 및 보고 기능, 기기의 원격 제어, 연결성 유지, 보안 및 프라이버시)과 인터페이스를 정의하였다.

oneM2M의 기본 구조는 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[17, 18].

4.1 OneM2M 보안구조

OneM2M은 통신 보안을 위해 TLS 1.2버전과 DTLS 1.2버전을 모두 선택하고 있다. DTLS(Datagram Transport Layer Security)는 전송 계층의 TCP 프로토콜에 보안성을 제공해주는 TLS(Transport Layer Security)프로토콜을 기반으로 하여 암호화된 데이터 그램을 전송할 수 있도록 해주는 보안 프로토콜이다. 이는 UDP(User Datagram Protocol)을 위한 프로토콜로 UDP를 기반으로 한 API는 TLS, DTLS를 사용하여 통신 보안 기능을 제공할 수 있다. DTLS는 IoT에 보안성을 추가해줄 수 있는 프로토콜로 제안된다.

Fig. 2는 oneM2M의 보안구조를 보여준다[19]. Fig. 2는 oneM2M에서 공개한 자료[19]를 재구성하였다.

보안 기능 계층(Security Fuctions Layer)은 참조 포인트인 Mac와 Mcc를 통해 이루어지는 보안기능 집합이 존재한다. 보안기능 집합은 6개의 보안기능으로 나뉘며, 식별과 인증, 권한부여, ID관리, 보안 연계, 센서티브 데이터 처리, 보안 관리로 구성되어있다[20].

인증은 식별과정에서 제공되는 자격 증명에 관련한 검증을 하는 과정을 나타낸다. 식별 과정은 인증의 목적에 따라 다른데 리소스 접근의 경우 AE와 CSE가 로컬 CSE의 등록이 되어 있는지 확인을 한다. 예를 들어 인증서 기반으로 한 인증 메커니즘은 디지털 서명을 검증을 하고 대칭키 기반 인증 메커니즘은 메시지 인증 기반를 사용한다. 권한 부여는 인증된 기관에게 서비스 및 데이터 접근을 허가하는 역할을 하며, 다수의 접근의 제어정책을 평가할 필요가 있다. 인증 평가 과정(ACL, RBAC)은 인증된 객체에 가입하고, 보호 지원과 관련된 접근 통제 정책에 따라 서비스 가입 지원을 활용한다. ID 관리에서 식별자들이 보안 환경에 저장된 상태라면, oneM2M 식별 또는 식별에 필요한 독립 개체들에 ID를 제공한다. ID 증명에 관련한 어떠한 역할도 부여 받지 않고 독립적으로 사용된다. 보안 연계는 통신 링크에 연결된 두 노드사이에 물리적 관계이다. 안전한 세션 설정과 사물 보안을 통해 안전한 연결을 하는 것이 목적이다. 센서티브 데이터 처리는 어플리케이션 계층에게 세 가지의 특정 센서티브 기능을 제공한다. 첫째로 안전하게 데이터를 저장하는 기능을 제공하고, 두 번째로 암호화 기능을 지원하고, 마지막으로 최초의 키를 부트스트래핑하기 위한 방법을 지원한다. 보안 관리는 보안환경(독립적인 하드웨어 모듈, 신뢰 할 수 있는 통합 실행환경 또는 소프트웨어 보호)의 독립적 환경이다. 규격에는 명시 되어 있지 않지만 Table 3과 같이 다양한 보안 기능인 보호 단계(protection level)을 제공하고 있다[21].

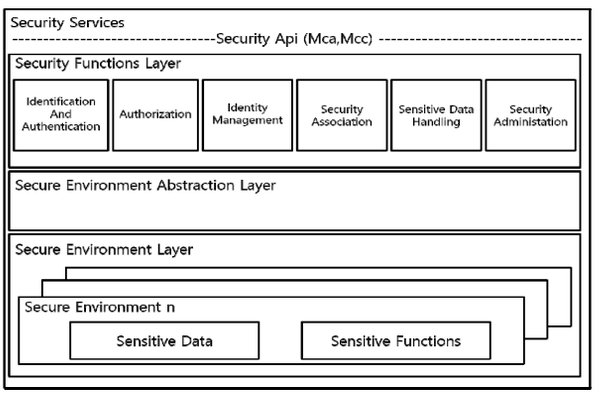


Fig. 2. OneM2M Security Architecture

Table 3. Provided Protection Level[21]

Level	Description
Level 0	Non defense state, the method does not perform any protection.
Level 1	Steps to defend against passive attacks, protects against attacks such as receipt of eavesdropping information.
Level 2	Steps to defend against remote attacks, prevent attacks and viruses or malicious code from being installed to view information or steal them illegally. Protected by local software attacks.
Level 3	Protect your computer from local hardware attacks

보호 환경 계층(Secure Environment Layer)은 여러 개의 보호 환경을 포함하는데, 각각의 보호 환경들은 센서티브 데이터와 센서티브 기능을 포함한다. 이중 센서티브 데이터는 SE capability, 보안 키, 로컬 인증(local credentials), 식별정보, 구독정보를 포함한다. 센서티브 기능에서는 데이터 암·복호화를 포함하고 있다[22].

1) 암호키 생성을 위한 키 교환

oneM2M은 TLS 1.2버전과 DTLS 1.2버전의 키 교환 알고리즘을 사용한다. 둘 다 ECDHE를 사용한다. PSK TLS에서 PSK TLS_PSK_WITH_AES_128_CBC_SHA256을 사용하고 DTLS에서의 PSK는 TLS_PSK_WITH_AES_128_CC M_8을 사용한다. ECDSA TLS에서 ECDSA는 TLS_ ECDHE_ECDSA_WITH_AES_128_CBC_SHA256을 사용하고 DTLS에서의 ECDSA는 TLS_ECDHE_ECDSA_WITH_ AES_128_CCM_8을 사용한다. 또한, Fig. 2의 보안 구조에서 Security Functions Layer에서는 MAF(m2m authentication function) 기반 SAEF(security association establishment framework)으로 인증을 수행한다. MAF 기반 SAEF 방식은 통신하는 개체들이 아닌 제 3자인 서비스 제공자(3rd party domain)에 의해 M2M 개체 A와 B, MAF의 3자간 인증을 수행한다.

2) 인증서

oneM2M의 인증서는 X.509(IETF RFC 5280)를 준수해야 한다. 인증서는 ECDSA 서명알고리즘을 사용한다. 인증서에는 NIST P256 커브와 id-ec공개키의 알고리즘을 나타내는 Subject 공개키 정보를 포함해야 한다. 공개키는 압축되지 않아야 하며 해쉬 알고리즘은 SHA-256(SHA-256-120, SHA-256128, IETF RFC 6920의 SHA256)을 사용한다. 전자서명도 포함하고 있어야 한다.

5. IoTivity 플랫폼 암호기술 현황 분석

OCF(Open Connectivity Foundation)는 IoT 오픈 플랫폼을 개발하는 글로벌 표준 단체로 IoTivity 오픈 플랫폼을 개발하였으며 IoT 환경에서 신뢰성 있는 상호운용성을 보장하기 위해 커넥티비티 프레임워크 규격을 개발하였다. 더불어 시험인증을 통해 검증하는 서비스를 제공한다. OCF는 IoT 기기들을 연결하여 로컬 안 밖에서 제어가 가능하도록 규격을 제공한다. 향후에는 타 IoT 플랫폼과도 연동 가능하도록 기술을 개발 중이다. OCF에 삼성, Intel, LG, Microsoft, Cisco, Qualcomm, Cablelabs, Electrolux, Haier, Cannon 등 약 400여개의 기업과 연구 단체, 제조사 등 다양한 종류의 기업들이 참여하고 있다[23].

IoTivity 플랫폼 계층은 전송 계층, 프레임워크 계층, 프로파일 계층으로 구성되어 있다. 전송 계층은 Bluetooth, Wi-Fi, ZigBee 등 IoT에서 활용되는 다양한 네트워크 연결기술에 대한 부분으로 확장 가능하다. 프로파일 계층은 IoT 응용 분야를 의미한다. 프레임워크 계층은 다양한 IoT 어플리케이션들의 자원 발견, 데이터 전송, 기기 관리, 데이터 관리 등의 기능을 지원한다. RESTful(representational state transfer ful) 아키텍처 모델을 기반으로 하고, 모든 사물을 자원으로 표현하고 CRUDN(Create, Read, Update, Delete and Notify)을 제공한다. 데몬 없이 CoAP(Internet Engineering Task Force Constrained Application Protocol) 기반으로 설계되어 저사양, 저전력 기기 지원이 용이하다[24].

현재 IoTivity는 우분투, 윈도우, 타이젠, 안드로이드, IOS를 지원하고 오픈 소스 하드웨어 플랫폼의 경우 현재 아두이노, 에디슨을 지원한다.

5.1 IoTivity 보안 구조

OCF 보안 구조의 목표는 자원 보호 및 이를 지원하는 하드웨어 및 소프트웨어를 보호하는 것이다. OCF 관점에서 기기는 OCF 명세를 따르는 논리적 개체다. 서버는 자원을 제어하고 보안 정책에 따라 클라이언트 역할을 하는 기기에 해당 자원에 대한 접근 권한을 제공한다. OCF에서는 서로 다른 암호화 기능을 사용하는 다양한 기기들 간의 네트워크 구성을 허용하기 위해 OBT(Onboarding Tool)라는 신뢰 가능한 초기 설정 툴을 사용하여 기존 사물 인터넷 네트워크 또는 신규 네트워크를 구성하는 방법을 설명하고 있다. 또한 OCF의 리소스 지향 아키텍처에서 동작하는 보안 기능들이 필요한 보안 리소스들과 데이터 송수신 과정에서 리소스들을 보호하는 메커니즘을 기술하고 기기에 있는 리소스들을 관리하는 접근 제어 메커니즘과 기기들 간 보안 통신 메커니즘을 정의한다. 다양한 운영체제나 플랫폼 위에 OCF 소프트웨어를 탑재하기 위한 보안 요구사항들 또한 Security 표준 규격에서 제공한다. Table 4는 OCF의 보안기능과 이에 대한 설명을 정리한 것이다[25.].

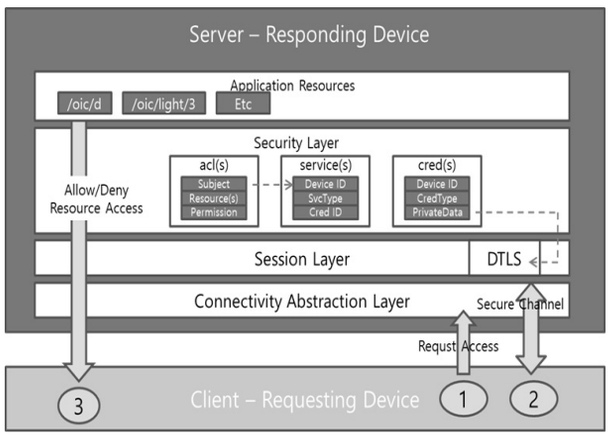


Fig. 3. IoTivity Security Architecture

Fig. 3은 IoTivity 보안 구조를 나타낸다[25]. Request Access에서 클라이언트는 서버에 네트워크 연결을 설정한다. 보안채널을 사용해 메시지를 교환하는 경우 각 장치의 oic.sec.cred 리소스는 상호 인증 및 적용 가능한 경우 인증서 유효성 검사에 사용되는 자격 증명을 보유한다. 또한 보안 채널을 통해 수신 된 메시지는 기기 UUID와 연결한다. 기기의 UUID는 다른 장치에서 받은 인증한다. 대칭키 인증을 위해 기기의 UUID는 oic.sec.cred 자원에 구성되고 장치 콘텍스트와 플랫폼 사이에 바인딩이 있어야한다. 허용된 역할이 있을 경우에는 역할 인증서에 제공되며 역할 인증서는 클라이언트가 서버에 구성한다. 보안채널을 사용하지 않고 메시지를 교환하는 경우에는 보안되지 않은 채널을 통해 서버에서 받은 요청은 익명으로 처리된다. 또한 기기의 UUID 또는 roleid와 연결되지 않는다. Allow/Deny Resource Access에서 요청이 전송되면 액세스 제어 목록을 참조하여 기준과 일치하는 항목을 찾는다. 요청한 리소스가 ACE의 리소스 참조와 일치하면 요청된 작업은 ACE의 “사용 권한”에 의해 허용한다. 요청이 보안 채널을 통해 전송되는 경우에 클라이언트는 요청에 ‘role’옵션을 포함시킨다. 그리하여 명시적으로 특정 roleid를 지정하거나 ‘role’옵션을 포함하지 않음으로써 클라이언트와 관련된 모든 roleid를 암시적으로 지정한다. 요청이 보안되지 않은 채널을 통해 수신된 경우 서버는 요청을 익명으로 처리하며 기기의 UUID 또는 roleid는 요청과 관련되지 않는다. 보안 채널을 통해 요청이 수신되면 서버는 기기 UUID를 연결하고 클라이언트의 허용된 역할 ID와 일치시켜 명시적으로 선언된 역할 ID의 유효성을 검사하거나 클라이언트의 유효한 모든 역할 ID를 암시적으로 지정한다.

Table 4. OCF Security Features and Descriptions

OCF Security Features	Description
Access Control	Provides a method to manage access control of resources using ACL (Access Control List) and ACE (Access Control Entry)
Onboarding and Security Provisioning	Provides initial setup method using OBT (Onboarding Tool) and initial entry method of OCF device of existing IoT network
Bootstrap process and Security bootstrapping	Provides a way to protect the bootstrap process and bootstrap
Secure Resource Manager	Secure Resource Manager (SRM) plays a key role in providing security. It consists of Resource Manager (RM), Policy Engine (PE), and Persistent Storage Interface (PSI), and includes functions such as resource management, policy enforcement and secure repository management
Security Credential Management	Provides a way to protect data during identification and communication between OCF devices using public / private keys
Device Authentication	Provides a way for a server to authenticate access to a client
Message Integrity and Confidentiality	Provides the use of security mechanisms that provide confidentiality and integrity to protect messages from attacks such as eavesdropping, tampering, or message repetition that can occur in communication between a server and a client.
Security Resources	Define security resources needed to provide security functions

Fig. 4는 IoTivity 보안 구조도를 보여준다. IoTivity 기기는 서비스를 사용하는 OIC Client, 서비스를 제공하는 OIC Server, 서비스 중계 역할을 수행하는 OIC Intermediaries로 구성된다[26]. Fig. 3과 Fig. 4는 IoTivity에서 공개한 자료[25, 26]를 기반으로 한 보안 구조도이다.

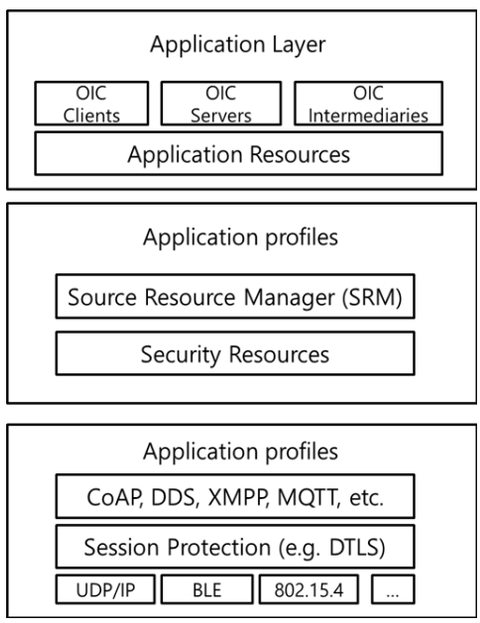


Fig. 4. IoTivity Security Structure

IoTivity 기기는 Application Resources를 가지고 있으며, 각 애플리케이션에 대한 Application profiles이 정의된다. 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 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 연결

을 성립하고, Connectivity abstraction 계층은 추상화를 통해 다양한 연결 옵션을 제공한다. OIC 기기를 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 암호 키를 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, 자격증명에 대한 각각의 오브젝트를 정의한다. 이 때, ACL 오브젝트는 Subject, Resource, Permission을 포함하며, 서비스 오브젝트는 Device ID, Svc Type, CredID를 포함하고, 자격 증명 오브젝트는 Device ID, Cred Type, Private Data로 구성된다. 정의된 세 개의 오브젝트를 연결하기 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[27].

1) 데이터 보안

자원을 안전하게 저장하기 위해 하드웨어 보안 방법이나 데이터 암호화 방법을 사용한다. 하드웨어 보안 저장장치로 반도체기반 비휘발성 메모리 등을 사용하며 비밀키, 개인키, 권한 등의 데이터를 저장하고 비인가된 접근으로부터 이를 보호하기 위해 데이터 암호화 방법을 사용한다. 전송 데이터 보호를 위해 DTLS를 이용할 수 없는 경우, 자원 계층 에서 JSON 웹 암호화 (JWE) 및 JSON 웹 서명(JWS)과 같은 메커니즘을 통해 전송 데이터의 전체 페이로드 보호를 제공한다. DTLS를 이용하는 경우, 전송계층에서 전송되는 데이터의 패킷 보호를 제공한다. 예를 들어 페이로드 전체의 무결성이 필요하면 패킷을 전송 계층으로 전달하기 전 별도의 서명 메커니즘이 이미 마련되어 있어야 한다. Table 5는 소프트웨어 보안을 위해 사용되는 알고리즘을 정리한 것이다.

Table 5. Algorithm for Software Security

Key Exchange	Non-repudiation	Integrity	Confidentiality	Managing Permissions
ECDH ECDHE RSA	ECDSA	SHA256	AES_128 AES_256	RBAC SBAC ACL SVR

Table 6. Provided Cryptographic Key

Method	Cryptographic key
Just Works Method	• TLS_ECDH_ANON_WITH_AES_128_CBC_SHA256 • TLS_ECDH_ANON_WITH_AES_256_CBC_SHA256
Random PIN Method	• TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA256
Certificate Method	• TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 • TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8 • TLS_ECDHE_ECDSA_WITH_AES_128_CCM • TLS_ECDHE_ECDSA_WITH_AES_256_CCM
Symmetric Keys	• TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA256 • TLS_PSK_WITH_AES_128_CCM_8 • TLS_PSK_WITH_AES_256_CCM_8 • TLS_PSK_WITH_AES_128_CCM • TLS_PSK_WITH_AES_256_CCM
Asymmetric Credentials	• TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 • TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8 • TLS_ECDHE_ECDSA_WITH_AES_128_CCM • TLS_ECDHE_ECDSA_WITH_AES_256_CCM

2) 메시지의 무결성과 기밀성

보안 통신을 위해 DTLS [RFC 6347]와 TLS v1.2 [RFC 5246]를 지원한다. 사용이 허용된 암호는 컨텍스트에 따라 다르다. 모든 CCM 기반 암호는 인증을 위해 HMAC-SHA-256를 사용한다. Table 6은 제공하는 암호키를 정리한 것이다.

6. IoT 플랫폼 암호기술 비교 분석

본 절에서는 IoT 플랫폼 암호기술을 비교 분석한다. AllJoyn 플랫폼을 제공하는 AllSeen Alliance은 OCF와 합병하여 현재 IoTivity의 형태로 제공이 된다. 하지만 AllJoyn 플랫폼은 현재까지 IoT 기기에 많이 사용된 대중화된 플랫폼으로 IoTivity와 다른 암호기술을 제공하기 때문에 본 논문에서는 플랫폼을 각각 비교하였다. IoT 기기 보안 위협에 따른 보안 요구사항은 기밀성(confidentiality), 무결성(integrity), 가용성(availability)과 인증(authentication), 인가(authorization)이다. AllJoyn 플랫폼은 기밀성, 무결성과 인증, 인가 요구사항을 위한 암호기술을 제공한다. 데이터 전송 시 AES-128-CCM 알고리즘을 사용하여 암호화하여 전송하며 Security 2.0 권한 모듈에서 접근 자격 증명 및 접근 제어 목록 데이터베이스를 관리한다. oneM2M은 Fig. 2. security function layer 내부의 6개 보안 기능 계층을 통해 기밀성, 무결성, 인증, 인가를 제공한다. oneM2M에서 제공하는

Table 7. Cryptographic Technology for Each Platform

Requirements	Methods	AllJoyn	oneM2M	IoTivity
Confidentiality	cryptographic algorithm	AES128-CCM	AES192/256	AES128/256
	key exchange method for data encryption	ECDHE	ECDHE	ECDH ECDHE RSA
Integrity	cryptographic hash function	SHA256	SHA256	SHA256
Authorization	access control mechanism	ACL	ACL, RBAC, ABAC	ACL, RBAC, SBAC, SVR
Authentication	Device Authentication	NULL, PSK, ECDSA with X.509 (TLS 1.2)	PSK, ECDSA with X.509(TLS 1.2, DTLS 1.2), MAF based SAEF	ECDSA with X.509 (TLS 1.2, DTLS 1.2)
	Message Authentication	AES-CCM (generates 8bit MAC)	HMAC-SHA-256 HMAC-SHA-512	HMAC-SHA-256

인증방법은 개인 서명키와 인증서, 공개키 기반 방법, 미리 공유된 대칭키(PSK: Pre-Shared Key) 기반 방법, MAF (M2m Authentication Function) 기반 SAEF(Security Association Establishment framework) 방법이 있다. 이러한 인증 메커니즘을 통해 인증과정을 수행하고 센서티브 데이터 처리 계층에서 데이터를 AES-CCM 암호알고리즘을 이용해서 암·복호화 한다. IoTivity도 기밀성, 무결성, 인증, 인가를 제공한다. 데이터 전송 시 AES192/256 알고리즘을 사용하여 암호화하고 접근 제어 목록을 참조하여 사용 권한을 제공하며 기기 ID의 자격 증명에 따라 인증한다.

가용성은 IoT 기기의 보안 모니터링과 관리를 위해 소프트웨어 업데이트 및 보안 패치 기능을 제공해야함을 의미한다. 이는 소프트웨어의 안전성을 보장하는 것으로 플랫폼을 활용하여 개발하는 개발자가 제공하여야 한다. 아래 Table 7은 AllJoyn, oneM2M, IoTivity 플랫폼을 IoT 기기의 보안요구사항에 따라 비교한 결과를 정리한 것이다.

Table 7에서 기밀성 항목을 보면 세 플랫폼 모두 AES 기반 암호 알고리즘을 사용하지만 제공되는 키 길이가 다르다. AllJoyn은 128bit AES 암호알고리즘만 사용하지만oneM2M과 IoTivity는 256bit AES 암호알고리즘도 사용할 수 있어 좀 더 높은 보안을 제공할 수 있는 것으로 보여진다. 데이터 암호를 위한 암호 키 생성을 위해서, AllJoyn과 oneM2M은 매 통신세션 마다 새로운 암호 키를 생성할 수 있는 ECDHE (Elliptic-Curve Diffie - Hellman Ephemeral)를 사용한다. IoTivity는 ECDHE 뿐 아니라 고정된 키를 생성하는 ECDH (Elliptic-Curve Diffie - Hellman)와 RSA암호 기반 키전송 기법도 함께 제공한다. 무결성은 세 플랫폼 모두 암호학적 해쉬 함수 인 SHA-256을 제공한다.

메시지 인증에서 AllJoyn은 대칭키 암호 알고리즘인 AES-CCM을 사용하여 인증하는데 oneM2M과 Iotiviy는 암호학적 해쉬 함수 SHA 기반의 HMAC(Hash-based Message Authentication Code)을 사용한다. 그중에서 oneM2M은 SHA-512 기반의 HMAC을 사용할 수 있도록 메시지 인증방법을 제공한다. IoT 기기 인증을 위해 AllJoyn과 oneM2M은 기기 설정 단계나 등록 단계에서 공유한 비밀키 PSK 기반 인증 또는 X.509 인증서를 활용한 ECDSA 서명 알고리즘을 제공한다. oneM2M은 추가로 M2M 고유의 3자 서비스 제공자를 경유한 인증 방법인 MAF 기반 SAEF를 사용한다. IoTivity는 X.509 인증서를 활용한 ECDSA 서명 알고리즘을 사용한다.

인가 항목은 시스템 보안측면에서 권한이 있는 사용자에게 시스템 접근을 통제하기 위한 암호 알고리즘을 나타낸다. SBAC(Subject-Based Access Control)는 접근제어를 위해 인증을 요청한 대상의 신원을 지정하고 RBAC(Role- Based Access Control)은 접근 요청한 개체의 역할을 지정한다. ABAC(Attribute-based access control)은 속성을 결합하는 방식을 사용하여 사용자에게 접근제어권한을 부여한다. SVR (Secure Virtual Resources)은 데이터베이스에 표준 형식으로 정의되며 보안 처리를 위해 SRM에 제공된다. AllJoyn은 ACL만 제공되며 oneM2M과 IoTivity는 RBAC, ABAC, SBAC과 같은 접근제어 알고리즘을 제공한다.

비교 분석한 세 플랫폼 전부 기밀성, 무결성, 인증 및 인가 요구사항을 만족하며 가용성은 각 플랫폼들의 보안 가이드라인 문서에 소프트웨어 개발자에게 요청하는 보안 요구사항으로 작성되어 있다. 암호 키 생성을 위한 키 교환 알고리즘인 ECDH/ECDHE, 또는 RSA 기반 암호키 전송방식, 인증서 기반 ECDSA 서명알고리즘을 활용한 인증 방식 등이 현재 IoT 오픈 플랫폼에서 제공되고 있다. 하지만 경량 IoT 기기 중 0

등급은 현재 제공되는 암호 알고리즘을 사용할 수 없으며 1등급은 일부 기기에서 최적화 구현을 한 암호 알고리즘을 사용할 수 있다. 0등급은 통신 기능이 없는 기기로 주로 센서의 역할을 한다. 0등급 기기들은 다른 기기에 부착되어 간접적으로 통신 네트워크와 물리적 사물을 연결시켜 데이터를 전송하기 때문에 부착된 기기에 암호 기술을 적용할 수 있다. 하지만 0등급에 해당하는 기기 중 혈압계와 같은 제품은 사용자의 중요한 데이터를 내장하며 다른 기기에 부착하여 사용하는 것이 아니다. 그러므로 일부 0등급 IoT 기기는 기기 자체에 암호 기술을 제공하는 하드웨어를 부착하는 방식을 사용해야 한다. IoT 기기는 기기의 메모리, 통신 사양에 따라 등급이 구분되어지기 때문에 이에 따른 암호 기술이 요구된다.

7. 그 외 IoT 플랫폼 암호기술 현황 분석

앞서 조사한 플랫폼 외 대표적인 IoT 플랫폼은 Android Things와 Smart Things, Thread가 있다. Thread는 IP 기반 무선 네트워킹 프로토콜로 보안성이 높고 저전력 무선 네트워크로 가정용 생활 제품들을 연결하는 솔루션 개발을 목표로 한다[28]. 가정용 제품 사용 촉진하고 보안 및 상호 운용성을 보장하기 위해 엄격한 제품 인증을 제공한다. AES 암호알고리즘을 사용하여 다른 무선 프로토콜이 가지고 있는 문제점을 보안한다. 205개 이상의 기기를 하나의 네트워크를 통해 연결할 수 있고 네트워크와 어플리케이션 계층에서 보안을 제공한다. 인증된 기기만 네트워크에 연결되게 하며 은행-클래스와 공개 키 암호 방식을 지원한다.

임베디드 기기나 IoT 유스 케이스에 특화되어 있는 Android Things은 안드로이드 플랫폼이 확장된 것으로 IoT 기기 개발에 안드로이드 프레임워크를 적용할 수 있다[29]. 모든 안드로이드 프레임워크를 사용할 수 있어 개발자들이 이미 알고 있는 안드로이드 API의 재사용이 가능하다[30].

SmartThings는 보다 개방적이고 수용적인 스마트 가전제품을 위해 삼성전자에서 인수한 IoT 플랫폼이다[31]. 스마트 홈 기기를 위한 개방형 플랫폼으로 허브 역할을 하는 모바일 어플리케이션 하나로 많은 플랫폼과 호환이 가능하다.

Thread는 네트워킹 프로토콜에 가까우며 Android Things와 Smart Things는 플랫폼 내 암호기술에 관련하여 공개된 정보가 많지 않아 본 분석에서 제외하였다.

8. 결 론

IoT 환경에서 IoT 기기들은 사용자에게 서비스를 제공하기 위해 실시간으로 다양한 데이터를 송수신한다. 이러한 환경에서 보안은 중요한 이슈이다.

본 논문에서는 암호기술 현황을 분석하기에 앞서 IoT 기기의 등급을 조사하고 보안 위협에 따른 보안 요구사항을 분석하였다. 암호기술 현황 분석을 위해 현재 사용 중인 AllJoyn, oneM2M, IoTivity 등과 같은 IoT 오픈 플랫폼들의 암호기술 현황을 분석하였다. oneM2M은 주로 IoT 서버-클라이언트 환경 구성에 집중한 모습을 보이며, AllJoyn과 IoTivity는 기기 자체 개발에 집중하였다. 각 플랫폼은 기밀성, 무결성, 인증/인가에 대한 보안요구사항을 제공한다. 가용성의 소프트웨어 안정성 부분은 플랫폼이 아닌 소프트웨어 개발 시 고려해야할 사항이다. 하지만 보안 패치와 보안 정책 설정, 로그 기능에 관련한 부분은 정보가 공개되지 않아 제공 여부를 알 수 없다. 또한 각 플랫폼들은 IoT 기기 등급이 높은 경우에만 적용이 가능하다. IoT 오픈 플랫폼과 표준화 단체들을 중심으로 경량 디바이스에 관련한 암호기술 연구는 활발히 진행되고 있지만 등급 0인 IoT 기기에 대한 암호기술 연구는 미미한 상태이다. 기존의 암호기술은 IoT 경량 디바이스에 적용하기 어려우며 기기와 환경의 다양한 요소가 고려되어야 한다.

IoT 기기의 암호기술은 IoT의 발전과 사용량에 비해 상대적으로 미흡한 상태이다. IoT 기기의 등급과 성능, 통신 수단에 따른 다양한 암호기술은 향후 많은 연구가 이루어져야 할 분야라고 생각된다.

References

[1] Machina research report, https://machinaresearch.com/

[2] IoT Small Smart Home Appliance Security Guide, The Korea Internet & Security Agency(KISA), 2016.12.

[3] Jeong-Yong Eom, "Security technology for Home IoT / connected appliances," *The Journal of The Korean Institute of Communication Sciences*, Vol.34, No.10, pp.10-16, 2017.

[4] Gartner, "Predicts 2016: Security for the Internet of Things."

[5] IoT Security Survey Results (SANS Institute data).

[6] Telecommunications Technology Association Standardization Committe, "TTAK.KO-12.0298 IoT device class classification and security requirement," Telecommunications Technology Association(TTA), 2016.12.27.

[7] The Korea Internet & Security Agency (KISA), "Guide to Using Cryptography Authentication Technology in Internet of Things(IoT) Environment," KISA, 2016.04.11.

[8] Internet of Things Forum, "IoT device class classification and security requirement," Internet of Things Forum, 2015.12.01.

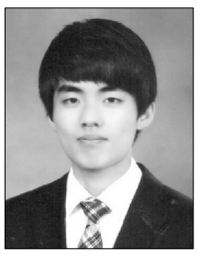
[9] Allseen Alliance [Internet], https://allseenalliance.org/, Linux Foundation.

[10] O. Tomanek and L. Kencl, "Security and privacy of using AllJoyn IoT framework at home and beyond," *Intelligent Green Building and Smart Grid (IGBSG), 2016 2nd International Conference on. IEEE*, 2016.

[11] A. Alliance, “Alljoyn framework. Linux Foundation Collaborative Projects,” https://allseenalliance.org/framework, 2016.
[12] M. Villari, A. Celesti, M. Fazio, and A. Puliafito, “Alljoyn lambda: An architecture for the management of smart environments in iot,” In *Smart Computing Workshops (SMARTCOMP Workshops), 2014 International Conference on (pp. 9-14). IEEE*, 2014.11.
[13] AllSeen Alliance Security [Internet], https://allseenalliance.org/frame work/documentation/learn/core/system-description/alljoyn-security.
[14] AllJoyn Security 2.0 Feature: High-level Design [Internet], https:// identity.allseenalliance.org/developers/learn/core/system-description, 2016.
[15] oneM2M [Internet], http://www.oneM2M.org.
[16] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, “Toward a standardized common M2M service layer platform: Introduction to oneM2M,” *IEEE Wireless Communications*, Vol.21, No.3, pp,20-26, 2014.
[17] 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, 2015.
[18] oneM2M Function Architecture, “TTAT.MM-TS.0001 v2.10.0 oneM2M - Functional Architecture”, oneM2M, 2016.08.
[19] oneM2M Security, “TTAT.MM-TS.0003 v. 2.4.1 oneM2M - Security Solutions,” oneM2M, 2016.08.
[20] oneM2M Service Layer Core Protocol Specification, “TTAT.MM-TS.0004 v. 2.7.1 oneM2M - Service Layer Core Protocol Specification,” oneM2M, 2016.08.
[21] oneM2M Authorization Architecture and Access Control Policy, “TTAT.MM-TR.0016 v. 2.0.0 oneM2M - Authorization Architecture and Access Control Policy,” oneM2M, 2016.08.
[22] oneM2M Security, “TTAT.MM-TR.0008 v. 2.0.0 oneM2M - Security,” oneM2M, 2016.08.
[23] Open Conectivity Foundation, [Internet], https://openconnectivity.org/
[24] A. Subash, “IoTivity - Connecting Things in IoT,” TIZEN Development Summit, 2015.
[25] OCF Security, “OCF Security Specification, v.1,” OCF, 2017.6.
[26] Iotivity Security, [Internet], https://wiki.iotivity.org/iotivity_security.
[27] K. Ashwini, L. Chul, S. Randeep, S. Sandeep, and S. WooChul, “IoTivity Provisioning Manager Design Specification v0.1d,” The Open Interconnect Consortium (OIC), 2015.
[28] Threadgroup, [Internet], Threadgroup.org
[29] Dave Smith, “Just Android Things,” realm, 2017.07.03.
[30] Android Things SDK overview [Internet], https://developer.android.co m/things/sdk/index.html
[31] SmartThings Developer Documentation [Internet], http://docs.smartthi ngs.com/en/latest/architecture/

최 정 인
http://orcid.org/0000-0003-2959-2268
e-mail : peach0206@hanyang.ac.kr
2010년 가천대학교 컴퓨터미디어학과 (공학사)
2012년 이화여자대학교 컴퓨터공학과 (공학석사)
2017년 이화여자대학교 컴퓨터공학과(공학박사)
2017년~2018년 한양대학교 공학기술연구소 전자공학과 박사후연구원
2018년~현 재 부산대학교 SW교육센터 강의전담
관심분야 : Sensor, IoT, Authentication

오 윤 석
https://orcid.org/0000-0002-4941-1447
e-mail : ashbringer@hanyang.ac.kr
2017년 고려대학교 정보수학과(이학사)
2017년~현 재 한양대학교 전자공학과 석사과정
관심분야 : IoT Security & Privacy



김 도 원
https://orcid.org/0000-0001-8524-524X
e-mail : dowonkim@kisa.or.kr
2009년 국민대학교 수학과(이학사)
2015년 국민대학교 금융정보보안학과 (이학석사)
2017년~현 재 한국인터넷진흥원 주임연구원
관심분야 : 대칭키 암호, 디지털 포렌식, 암호키 관리, 랜섬웨어

최 은 영
https://orcid.org/0000-0002-8904-3669
e-mail : bluecey@kisa.or.kr
2001년 고려대학교 수학과(이학사)
2003년 고려대학교 정보보호대학원 (공학석사)
2009년 고려대학교 정보보호대학원 (공학박사)
2017년~현 재 한국인터넷진흥원 책임연구원
관심분야 : 암호이론, 정보보호, 암호키 관리, 랜섬웨어

서 승 현

http://orcid.org/0000-0002-1150-7080

e-mail : seosh77@hanyang.ac.kr

2000년 이화여자대학교 수학과(이학사)

2002년 이화여자대학교 컴퓨터학과 (공학석사)

2006년 이화여자대학교 컴퓨터학과 (공학박사)

2006년~2010년 금융보안연구원 주임연구원

2010년~2012년 한국인터넷진흥원 선임연구원

2014년~2015년 고려대학교 정보보호대학원 BK21+ 사업단 연구교수

2015년~2016년 고려대학교 수학과 조교수

2017년~현 재 한양대학교 전자공학과 부교수

관심분야 : IoT/CPS Security, Blockchain Security and Application, Cryptographic Protocols