# Secure Pi: COTS IoT 디바이스 보안 플랫폼

김병선○ 조진성
경희대학교 컴퓨터공학과
ykbs0903@khu.ac.kr, chojs@khu.ac.kr

# Secure Pi: COTS IoT Device Security Platform

Byoungseon Kim○ Jinsung Cho
Department of Computer Engineering
KyungHee University

**요 약**

IoT 는 다양한 기술들의 통합, 유무선 네트워크 인프라, 각종 디바이스들을 상호 연동함으로써 측정된 정보를 다양한 응용에 활용할 수 있는 기술을 의미한다. 하지만, 이러한 기술의 통합과 상호 연동으로 인해 IoT 보안 취약점들이 추가적으로 발생하고 있으며, 대부분의 보안 취약점은 디바이스 수준에서 발견되고 있다. 안전한 IoT 서비스를 제공받기 위해서는 디바이스 보안의 지속적인 취약점 분석과 개선이 중요하며, 본 논문에서는 이러한 보안 문제를 해결하기 위한 기반 조건으로 IoT 표준을 통한 대표적인 보안 취약점과 대책을 분석하였다. 또한, COTS IoT 디바이스로 활용될 수 있는 Raspberry Pi와 TPM을 연동하여 하드웨어 보안 플랫폼 프로토타입을 제작하였고, 각 보안 대책을 만족시킬 수 있는 디바이스 핵심 요소기술을 제시한다.

## 1. 서 론

과거 Ubiquitous Sensor Network (USN), Cyber Physical System (CPS) 시대에는 다양한 디바이스들이 공중망이 아닌 개별 네트워크 대역을 통한 단방향 전송 방식으로 운영되었기 때문에 서버 보안을 제외한 기타 구성요소 보안에 대한 인식은 크게 강조되지 않았다.

하지만 IoT (Internet of Things) 시대에 들어서면서 다양한 기술들의 통합, 유무선 네트워크 인프라, 디바이스 간 상호 연동은 기존 보안 취약점뿐만 아닌 다양한 취약점들을 발생시키는 원인이 되었다. 플랫폼 개방에 따른 공격 방법의 다양화, 각 디바이스에 특화된 취약점, 디바이스 간 연결 부분의 취약점, 보안 취약성이 서로 연결됨으로써 새롭게 등장하는 보안 취약점 등이 대표적인 예라고 볼 수 있다 [1]. 이러한 취약점에 대응하기 위해 IoT 보안은 구성요소에 따라 디바이스 보안, 네트워크 보안, 서비스 플랫폼 보안으로 구분되어 각 분야마다 다양한 연구가 진행되고 있다.

한편 IoT 서비스를 제공하는 디바이스는 운영체제와 응용 소프트웨어의 무결성, 파일 시스템 일관성 등의 다양한 측면에서 보안과 정보보호가 고려되어 설계되어야 한다. 하지만, 펌웨어 패치의 어려움, 제로데이 공격, DoS 공격, 디바이스의 집단적 기능 불능 등과 같은 피해 발생과 대부분의 보안 취약점이 디바이스 수준에서 끊임없이 발견되고 있다는 점에서 지속적인 취약점 분석과 보안 강화가 필요하다 [2].

IoT의 궁극적 목표는 다양한 디바이스로부터 수집된 데이터를 기반으로 이를 융합하여 다양한 응용 분야에 활용하는 것이고, 디바이스로부터 수집되는 정보가 방대하기 때문에 이에 대한 적절한 대응책이 제공되지 않는다면 개인정보 유출, 신분 도용, 서비스 거부 등의 심각한 보안 사고가 발생할 것이다.

이러한 IoT 디바이스 보안과 정보 보호를 위해 본 논문에서는 IoT 표준을 통한 보안 취약점과 대책을 분석하였다. 또한, 범용 COTS IoT 디바이스인 Raspberry Pi 와 TPM을 연동한 하드웨어 보안 플랫폼 프로토타입을 개발하였으며, 이를 기반으로 IoT 디바이스에 보안 특성을 제공할 수 있는 핵심 요소기술을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 IoT 보안 표준을 통해 IoT 취약점과 이에 대한 대책을 살펴본다. 3장에서는 IoT 디바이스 보안을 실현하기 위한 핵심 요소기술에 대해 설명하며, 4장에서는 향후 연구를 기술하고 결론을 맺는다.

## 2. IoT 보안 취약점과 대책

IoT 환경의 서비스 플랫폼 표준 규격을 만들기 위해 세계 주요 표준 제정 기관들은 2012년 7월 oneM2M 파트너십 프로젝트를 출범하였다. 본 표준화 조직은 6개의 Working Group (WG)을 구성하여 각 기술 분야의 표준화를 담당하고 있으며, 2014년 8월, 표준 후보(Release 1) 발표를 거쳐 현재 Release 2를 준비하고 있다. 이 중 보안 분야와 관련된 WG SEC은 TR-0008 (Technical Report)를 통해 [표 1]와 같은 보안 취약점 및 대책을 나타내고 있다 [3].

oneM2M 에서는 IoT 시스템을 크게 사용자/관리자가 존재하는 Application Domain, 공통 서비스를 제공하는 M2M 서버로 구성된 Infrastructure Domain, 각종 M2M 디바이스 및 M2M 게이트웨이를 포함하는 Field Domain 으로 구분하고 있다. 여기서, [표 1]에 열거한 보안 대책들 중 6개 항목 (공유 자산 인벤토리, 민감도 평가, 위험 평가, 컨텍스트 인

"이 논문은 교육부 및 한국연구재단의 기초연구사업 (NRF-2013R1A1A2059741)과 삼성전자 DS의 지원으로 수행된 연구결과임."

| 보안 취약점/위협 | 요구기능 (대책) |
|---|---|
| [위협 1] M2M 장치/게이트웨이에 저장된 장기 서비스 키 노출 | **- M2M 장치/게이트웨이의 장기 서비스 키 저장 용도의 침입 방지 저장소**<br>**- M2M 서비스 키로부터 유도되는 키** |
| [위협 2] M2M 장치/게이트웨이에 저장된 장기 서비스 키 삭제 | **- M2M 장치/게이트웨이의 장기 서비스 키 저장 용도의 침입 방지 저장소**<br>**- HSM과 M2M 장치/게이트웨이 간의 물리적/논리적 연동**<br>**-장기 서비스 키를 접근하는 과정에서의 강력한 인증** |
| [위협 3] 저장된 장기 서비스 키의 교체 | **- M2M 장치/게이트웨이의 장기 서비스 키 저장 용도의 침입 방지 저장소**<br>**- 장기 서비스 키를 접근하는 과정에서의 강력한 인증** |
| [위협 4] M2M 기반 구조에 저장된 장기 서비스 키 노출 | **- M2M 기반구조 장비의 장기 서비스 키 저장용 시큐어 스토리지**<br>**- HSM/서버 HSM에 저장된 접근 불가 속성의 서비스 키**<br>**- M2M 서비스 키로부터 유도되는 키** |
| [위협 5] M2M 기반 구조의 장비에 저장된 장기 서비스 키의 삭제 | **- M2M 기반구조 장비의, 장기 서비스 키 저장용 시큐어 스토리지**<br>**- 장기 서비스 키를 접근하는 과정에서의 강력한 인증** |
| [위협 6] M2M 장치나 M2M 게이트웨이에 저장된 민감한 데이터 노출 | **- M2M 장치/게이트웨이의 민감한 기능에 대한 안전한 실행** |
| [위협 7] 개체간 M2M 서비스 계층 메시지의 도청 | **- 보안 연계, 상호 인증, 기밀성 사용**<br>**- 중간자 공격에 대한 검증된 방어** |
| [위협 8] 개체간 M2M 서비스 계층 메시지의 교체 | **- 보안 연계, 상호 인증, 기밀성 사용**<br>**- 중간자 공격에 대한 검증된 방어**<br>**- 서비스 계층에 사용하는 세션키의 제한된 수명** |
| [위협 9] 개체간 M2M 서비스 계층 메시지의 리플레이 | **- 리플레이 방지**<br>**- M2M 서비스 키로부터 유도되는 키** |
| [위협 10] M2M 장치/게이트웨이에 설치된, 인가받지 않거나 손상된 응용 또는 소프트웨어 | **- 일관성 검증**<br>**- 정책 기반의 행동** |
| [위협 11] M2M 시스템의 상호 의존성에 대한 위협과 그에 따른 영향 | - 공유 자산 인벤토리<br>- 민감도, 위험 평가 |
| [위협 12] M2M 보안 상황인지 | - 컨텍스트 인벤토리와 민감도 평가<br>- 위험 평가 |
| [위협 13] 도청, 중간자 공격 | **- 안전한 통신 채널** |
| [위협 14] 독립적인 보안 요소를 통한 키 이전 | **- HSM과 M2M 장치/게이트웨이 간의 물리적/논리적 연동** |
| [위협 15] 버퍼 오버플로우 | - 시큐어코딩 |
| [위협 16] 삽입공격 | **- 신뢰할 수 없는 데이터 삽입 방지** |
| [위협17] 세션 관리와 부적절 인증 | **- 보안 제어** |
| [위협18] 보안 설정 오류 | - 응용 프로그램의 깔끔한 구조 |
| [위협19] 안전하지 않은 암호화 저장 | **- 표준 알고리즘 사용** |
| [위협20] 유효하지 않은 입력 데이터 | **- 권한을 사용하여 저장소 보호** |
| [위협21] 크로스 스크립팅 | **- 화이트리스트** |

<표 1 - TR-0008-Security 의 IoT 보안 취약점 및 대책>

벤토리와 민감도에 대한 평가, 응용 프로그램의 깔끔한 구조, 시큐어코딩)은 Application Domain, Infrastructure Domain 보안 영역에 해당되어 개발 습관 및 시스템 운영상의 보안 대책과 관련성이 높다. 즉, 상기 대책들은 M2M 디바이스의 보안을 직접적으로 책임질 수 없고, 소프트웨어의 안전한 코드 개발 습관과 보안 진단 및 컨설팅을 통한 IoT 보안 관제 프로세스의 정립이 필요하다. 나머지 보안 대책은 IoT 디바이스 수준에서 지원함에 따라 디바이스의 취약점 보완과 높은 보안성을 제공할 수 있을 것이다.

## 3. IoT 디바이스의 보안을 위한 요소기술

본 장에서는 M2M 게이트웨이를 포함한 M2M 디바이스 보안을 책임질 수 있는 핵심 요소기술과 이를 통해 보장될 수 있는 보안 대책에 대해 설명한다.

### 3.1. Secure Key Storage & Management

본 기술은 저장된 데이터의 암호화/복호화를 수행하기 위한 암호화 키 기밀성과 무결성을 제공한다. 이 암호화 키는 신뢰할 수 있는 하드웨어 (HSM; Hardware Security Module)에 의해 관리되어 [4], 사용자나 다른 소프트웨어에 의해 유출/교체될 수 없어 소프트웨어 기반의 다양한 공격 방법들을 원천적으로 차단할 수 있다.

본 연구에서는 HSM으로서 TPM 칩을 채택하여 해당 기술을 구현하였고, 이후에 설명할 요소기술들과도 연동 작업을 수행하고 있다. 대상 디바이스는 COTS IoT 디바이스로 널리 활용될 수 있는 Raspberry Pi를 기반으로 하였다.

・ 대응 가능한 취약점/위협 : 1, 2, 3, 4, 5, 14

### 3.2. Secure Boot

본 기술은 디바이스가 부트 될 때, 부트로더를 포함한 각종 부트 소프트웨어와 운영체제 구동까지의 무결성, 일관성을 검증하는 기술이다 [5]. 각 부트 단계마다 미리 생성한 서명(Measured Signature)과 각 부트 과정에서 생성한 서명이 서로 일치하지 않을 경우, 부팅을 중단함으로써 손상 및 검증되지 않은 소프트웨어의 실행을 방지한다.

・ 대응 가능한 취약점/위협 : 6, 10

### 3.3. Secure Firmware Update

본 기술은 펌웨어 불법 교체, 수정을 방지하기 위해 업데이트 이미지 검증 및 기존 이미지를 새로운 업데이트 이미지로 안전하게 교체하는 기능을 제공한다. 상호 인증 실패, 새로운 펌웨어 이미지나 업데이트 모듈이 검증되지 않았을 경우, 메모리 접근을 차단하여 검증되지 않은 펌웨어 업데이트 이미지가 메모리에 로드되는 것을

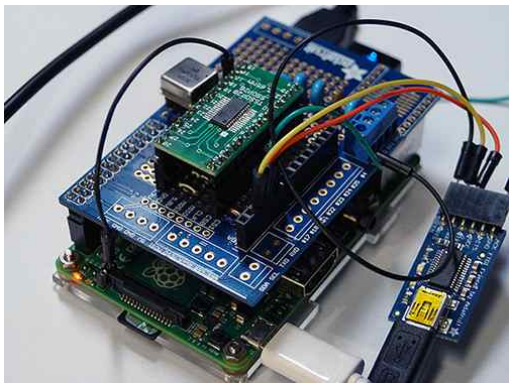
방지한다.
・ 대응 가능한 취약점/위협 : 6, 10

3.4. Device Attestation
본 기술은 디바이스가 오염/변경되지 않았다는 것을 입증하기 위한 기기간 상호 인증 기술을 의미하며, 많은 보안 프로토콜과 응용에서 필수적으로 사용되어야 할 보안 요소기술이다. 디바이스 인증의 구현은 디바이스 인증 프로토콜 (DAP; Device Attestation Protocol)을 사용한 소프트웨어 기반 인증 방식과 한층 더 강화된 보안을 제공할 수 있는 TPM, ARM Trustzone, Intel SGX (Software Guard eXtentions) 등을 활용한 하드웨어 기반 디바이스 인증 방식으로 구분될 수 있다.
・대응 가능한 취약점/위협 : 7, 8

3.5. Secure Communication
통신 부분에 있어서 디바이스 공격 방법은 날마다 발전하고 있으며 지능화되고 있다. 본 기술은 디바이스 간 통신에서 다른 개체의 개입을 방지하고 안전한 데이터 교환을 위한 통신 기술을 의미한다. 대표적인 Open Software로는 SSL(Secure Socket Layer) / TLS(Transport Layer Security) 프로토콜 기반의 OpenSSL이 있다. 본 프로토콜은 기본적으로 데이터 암호화와 데이터 무결성, PKI(Public Key Infrastructure) framework 기반의 서버/클라이언트 인증 등의 다양한 보안 기능을 제공하고 있다. 하지만, Heartbleed, BEAST, 등의 다양한 취약점이 계속적으로 보고되고 있어 꾸준한 취약점 분석과 개선이 필요하다.
・대응 가능한 취약점/위협 : 7, 8, 9, 13

3.6. Mandatory Access Control (MAC)
본 기술은 어떤 주체가 특정 객체에 접근 시 보안 레벨에 기초해 낮은 수준의 주체가 높은 수준의 객체에 접근하는 것을 제한하는 기술이다. 이러한 기술은 어플리케이션을 샌드박스/컨테이너 안에서 실행되게 함으로서 강제적 접근 제어를 구현할 수 있다. 이 기술은 SELinux(Security Enhanced Linux), SMACK (Simple Mandatory Access Control Kernel) 등과 같은 LSM (Linux Security Module)를 통해 지원될 수 있다.
・대응 가능한 취약점/위협 : 10, 17, 20, 21

3.7. File System Integrity
본 기술은 파일 시스템 내 파일들이 임의로 수정/오염되는 것을 방지하기 위해 파일 시스템 내 파일들의 무결성, 일관성을 검증하기 위한 기술이다. 이러한 기술을 지원하기 위한 대표적인 Linux 모듈은 IMA (Integrity Measurement Architecture) / EVM(Extented Verification Module)이 있다. IMA/EVM의 핵심 기능은 모든 파일들의 Hash를 Kernel Resident List에 보유하고, 미리 생성해 둔 Measurement 들과의 Local validation 을 통해 무결성을 검사한다. 또한, TPM이 지원될 경우, TPM PCR(Platform Configuration Register) 를 확장함으로서 각 파일들의 Hash와 TPM Signature를 통해 원격지의 Measurement 리스트와의 무결성 검사에 활용될 수 있다.



<그림 1 - Raspberry Pi와 TPM을 연동한 보안 플랫폼 프로토타입>

・대응 가능한 취약점/위협 : 10, 16

3.8. File System Encryption
본 기술은 파일 시스템 수준의 기밀성을 보장하기 위해 디스크 내 기밀 파일을 암호화/복호화 하는 기능을 제공한다. 대표적인 Linux 기반 Software로는 eCryptFS (Enterprise Cryptographic Filesystem)가 있다. 본 패키지는 기존 파일 시스템 위에 존재하는 가상 암호화 파일 시스템을 중심으로 Keyring 서비스, 각종 Key 관리, 암호화 API, 등의 다양한 기능을 제공한다. TPM과 같은 HSM을 지원함으로써 키 관리 측면에서 보안을 향상시킬 수 있다.
・대응 가능한 취약점/위협 : 6, 19

## 4. 결론 및 향후 계획
본 논문에서는 IoT 디바이스 보안을 위해 oneM2M TR-0008 표준을 통한 보안 취약점 및 대책을 분석하였고, 이러한 각 대책들을 만족시킬 수 있는 디바이스 보안 요소기술을 제시하였다.
이러한 기술들을 구현하기 위해 현재 [그림 1]과 같이 Raspberry Pi 와 TPM을 연동한 하드웨어 프로토타입을 제작하였고, 각 요소기술에 관련된 오픈소스 적용 및 추가 개발을 진행하고 있는 상태이다. 향후 개발된 요소기술의 문제점, 취약점 및 성능을 분석하고 개선점을 제시할 계획이다.

참 고 문 헌
[1] 김기영, “oneM2M 사물인터넷 서비스 플랫폼 표준화 현황 “, TTA 저널, vol.155, pp.34-44, 2014
[2] J. Pescatore, G. Shpantzer, “Securing the Internet of Things Survey”, SANS Institute, 2014
[3] oneM2M partners, “TR-0008 1.0.0: Analysis of Security Solutions for the oneM2M System”, oneM2M Specification (Release 1), 2014
[4] Trusted Computing Group, “Trusted Platform Module Library Part 1: Architecture “, TPM 2.0 Library Specification, 2014
[5] H. Lohr, A.-R. Sadeghi, and M. Winandy. “Patterns for secure boot and secure storage in computer systems”, International Conference on Availability, Reliability and Security, pp.569-573, 2010