https://doi.org/10.5392/JKCA.2018.18.04.164

IoT 보안 요구사항 및 보안 운영체제 기반 기술 분석

Technologies Analysis based on IoT Security Requirements and Secure Operating System

고재용, 이상길, 김진우, 이철훈
충남대학교 컴퓨터공학과

Jae-Yong Ko(bbxiix@cnu.ac.kr), Sang-Gil Lee(sk0137@cnu.ac.kr),
Jin-Woo Kim(jinu@cnu.ac.kr), Cheol-Hoon Lee(clee@cnu.ac.kr)

요약

IoT 디바이스와 관련한 시장이 확대됨에 따라 이와 관련한 보안 침해 사고의 규모가 상당할 것으로 예측된다. 이에 따라 국내 정보보호와 관련한 법제 정비를 위한 움직임 또한 활발해지고 있으며, 강화된 정보통신망 법이 시행되었다. IoT 관련 침해 사고는 재정적 피해뿐만 아니라 인명 사고로도 이어질 수 있기 때문에, IoT 디바이스 보안에 큰 관심이 집중되고 있다. 본 논문에서는 IoT 디바이스가 갖춰야 할 필수 보안 기능을 법제적 관점과 기술적 관점을 통해 시사점을 제시하고, 이와 관련한 기술들을 분석한다. 이는 Start-up 개발자나 IoT 디바이스 설계자에게 참고자료로 활용될 수 있다.

■ 중심어 : | 사물 인터넷 | 보안 운영체제 | 시스템 보안 | 보안 요구사항 |

Abstract

As the market for IoT devices grows, it is expected that the scale of malware attack will be considerable. Accordingly, the improvement of related legislation has been actively promoted, the recently strengthened Information and Communication Network Act was enforced. Because IoT related accidents can lead to not only financial damages but also human accidents, IoT device Security has been attracted a great deal of attention. In this paper, IoT devices provide essential security functions through legal and technical perspectives, and analyze related technologies. This can be used to a reference for the Start-up developer and IoT device designer.

■ keyword : | IoT | Secure Os | System Security | Security Requirements |

I. 서 론

초연결시대가 도래함에 따라, 클라우드 및 커넥티드 기기의 발전은 다양한 IoT 기기들이 개발될 수 있는 토대가 되고 있다. IoT 기기들은 대부분 임베디드 기기로서 가전제품부터, 항공·우주, 자동차, 의료 및 산업용 시장까지 폭 넓게 적용된다. 하지만 이러한 임베디드 기기는 오작동 시 생명이나 자산의 심각한 피해로 이어질 수 있기 때문에, 그 피해와 위험성을 염두에 두고 설계되어야 한다. 최근 IoT 기기들이 대거 출시되면서 IoT 기기들을 대상으로 하는 보안 침해 사례들이 발생하고 있으며 시장 규모가 커지면서 보안 침해 사고의

* 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임
(No. NRF-2017R1D1A1B03034775)
접수일자 : 2018년 02월 07일
수정일자 : 2018년 03월 12일
심사완료일 : 2018년 03월 13일
교신저자 : 이철훈, e-mail : clee@cnu.ac.kr

규모 또한 커지는 추세이다. 산업연구원의 조사에 따르면, 이러한 보안 침해에 의한 피해 규모는 2020년 약 18조 원에 이를 것으로 예측된다[1].

기업이나 개발자는 보안 침해 공격이 발생했을 경우 발생한 피해에 대해서 가장 큰 책임을 지게 된다. 한 예로서, 최근 방송통신위원회는 숙박 앱 '여기 어때'를 운영하는 '위드이노베이션'에 대해 과징금 3억100만 원, 과태료 2500만 원, 책임자 징계권고 등의 징계 조치를 내렸으며, 각 피해자로부터 손해배상 소송이 진행되고 있다. 기업에 대해 엄격하고 세밀한 개인정보 관리가 요구되었지만, 접근통제나 접속기록보존, 암호화, 유효기간제 등 각종 기술적·관리적 보호조치의 미흡이 개인정보 유출의 결정적인 원인으로 밝혀졌다. 해킹 공격이 발생하여 피해가 발생했을지라도, 기업이 개인정보를 보호하기 위한 의무를 소홀히 하였을 경우 그 피해에 대해 배상을 해야 한다는 것을 알린 사건이다.

이처럼 방대한 개인정보를 다루는 기업들에는 보안사고를 막기 위한 많은 노력이 요구되고 있다. 특히 피해 금액의 최대 3배를 배상하도록 하는 징벌적 손해배상제도 및 법정 손해배상제도가 도입되면서 개인정보 유출 피해자는 피해액을 입증하지 않아도 손해배상을 받을 수 있고, 고의나 중·과실로 개인정보를 유출한 기관이나 업체는 가중된 책임을 물게 된다. 그렇기 때문에, 앞으로 기업들은 보안사고가 발생했을 경우에 발생되는 모든 재정적 손실이나, 보안 솔루션을 도입하면서 발생되는 비용에 대해 더욱더 심사숙고하지 않을 수 없게 되었다. 특히 IoT 관련한 Start-up이나 개발자들은 이러한 보안 사고가 발생하였을 경우 충분한 재정적 뒷받침이 없기에 그 피해를 감당하기에 어려울 수 있으며, 정보보호 관련 법률 강화로 인해 발생되는 사회적·경제적 비용이 발생할 수 있다. 이러한 점은 궁극적으로 4차 혁명의 신 성장 동력으로 주목되는 IoT 관련 산업 성장을 저해할 수 있는 요소가 될 수 있다.

본 논문에서는 법제적 측면과 보안 기능 및 요구사항을 고려하여 IoT 핵심 보안 기술을 선정하였으며, 설계 및 제작단계에서 필수적으로 적용해야 할 기능에 대해 살펴본다.

본 논문의 구성은 다음과 같다. 2장에서는 IoT 기기와 관련한 법제적 요구사항을 살펴보고 3장에서는 IoT 개발 환경과 취약점, 전문가들이 제시한 보안 요구사항 및 보안 인증 점검 항목들을 고려하여 기능적 요구사항을 살펴본다. 4장에서는 강조된 법제적·기능적 요구사항을 토대로 IoT 보안 운영체제 기반 기술을 정리하였으며, 그 중에서도 시스템 레벨 보안의 중요성과 관련 기능들을 설명한 뒤, 마지막으로 5장에서 결론을 맺는다.

II. IoT 보안 관련 법제적 요구사항 분석

IoT 기기에 관련한 법제적 요구사항이란, IoT 보안을 위해 '반드시 해야 하는 것'을 의미한다. 본 장에서는 IoT 보안과 관련하여 IoT 기기가 반드시 지키거나 갖춰야 할 사항에 대해 알아본다.

1. IoT 관련 현행 법률

먼저 IoT 기기를 규제하는 단일 법률이 구체적으로 마련되지 않은 현시점에서는 IoT 기기는 정보 관련 법률들에 의해 적용을 받게 된다.

IoT 기기를 개발하는 업체나 개발자는 보안 사고를 막기 위해 개인정보보호법 제29조의 '안전조치의무'와 정보통신망 이용촉진 및 정보보호 등에 관한 법률(이하 정보통신망법)에 따른 '개인정보의 보호조치'와 같은 보안조치 의무를 갖춰야 한다.

정보통신망 법 제28조에 명시된 기술적·관리적 조치에는 내부관리계획의 수립·시행, 접근 통제장치의 설치·운영, 접속기록의 위·변조를 위한 조치, 개인정보에 대한 암호화 조치, 백신 소프트웨어의 설치·운영, 기타 안전성 확보를 위한 보호 등이 있다.

"개인정보의 기술적·관리적 보호 조치 기준"은 정보통신망 법 제28조를 따르며, "개인정보의 안전성 확보 조치"는 개인정보 보호법 제29조를 따르게 되는데, 각 조치 기준의 세부 항목들은 유사한 구분을 통해 세부 내용을 규정하고 있다. 다음은 "개인정보의 기술적·관리적 보호 조치 기준"과 "개인정보의 안전성 확보 조치"에서 IoT 기기와 연관이 깊은 항목을 나타낸 것이다.

- 내부 관리 정책
- 개인정보 책임자의 지정, 자격에 대한 사항
- 개인정보 관리 및 책임자, 취급자의 역할과 책임
- 내부관리 계획의 수립과 승인
- 기술적·관리적 보호조치 이행 여부의 내부 점검
- 개인정보 피해가 발생했을 경우의 대응절차
- 접근 권한 관리
- 접근 권한의 차등 부여
- 접근 권한의 변경이나 말소에 대한 기록 보관
- 접근 통제 시스템
- 업무망과 외부망의 분리
- 방화벽, IPS, IDS등의 설치
- 개인정보 암호화
- 비밀번호 일방향 암호화
- 개인정보 송·수신 시 암호화
- 비밀번호 작성 규칙 최소 8자리 이상
- 접속 기록 관리
- 접속기록 최소 6개월 이상 보관 관리
- 보안 프로그램
- 보안 프로그램 설치 및 운영
- 보안 프로그램의 자동업데이트 및 일 1회 업데이트 기능을 통한 최신 상태 유지
- 물리적 통제
- 출입 통제 절차
- 저장매체, 서류의 안전한 보관 및 입·반출 통제

이러한 기준들은 개인정보보호를 위한 최소한의 의무이며, 이외에도 IoT 기기의 특성에 따라 '국가정보화 기본법', '정보통신기반 보호법', '위치 정보의 이용 등에 관한 법률' 등을 참고할 수 있다.

IoT 기기가 법제적인 의무사항을 충족하는 데 필요한 보안 기능을 다음 [그림 1]과 같이 제안한다.

2. 현재 시행 법률의 한계 및 현황

IoT 기기에 해당 법률에 따른 보안 기능들을 적용하기에는 몇 가지 문제점이 남아있다. 먼저 IoT 관련 정보보호의 피해 규모와 심각성이 두드러지게 클 것으로 예상함에도 불구하고 단일화 법률이 없다는 점은 구체

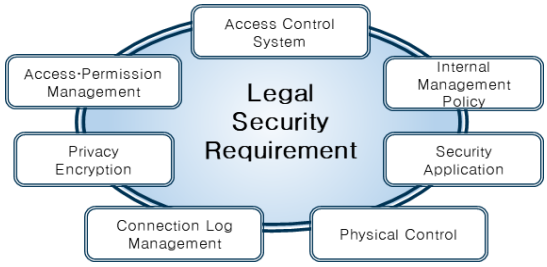


그림 1. 현행 법률을 충족하는 보안 요구 사항

적인 의무 및 책임에 대한 명확한 기준을 제시하지 못하고 있다는 점으로 지적된다[2]. 이는 보안의 수준이나, 강제성을 띠는 의무사항이 없기에 기업마다 그 수준이 상이하고 법제적 해석에도 문제가 될 수 있다. 현재에는 IoT 기기가 정보 관련 법률들에 의해 적용되지만 이마저도 개별 법규의 성격상 수범 주체가 다르고 보호 방법에 차이가 있다는 점이 지적되고 있다[2]. IT 장비에 대하여 ISMS와 CC 인증제도와 같은 정보보호 관련 인증제도가 있으나 법정 임의 인증제로서 보안 의무에 대해 법적인 강제성을 띠고 있지는 않다[3].

이에 따라 IoT 기기의 보안성 확보를 위해 법제적 개선방안들이 연구되고 있다[2][3]. IoT 관련한 법 조항은 점차 세부적으로 발의되고 있으며, 보안 요구사항에 따른 구체적인 기준들이 제시되고 있다[4]. 한편, 국내 온라인 리서치 조사기관 '두잇서베이[5]'에 따르면 "IoT 보안 관련 규제를 정부 차원에서 시급히 마련해야 한다."는 의견에 63.3%의 응답자가 공감하는 것으로 나타나 앞으로도 끊임없이 정부 차원의 규제 강화에 대한 요구가 있을 것으로 예측된다. 정부 차원의 규제 강화는 산업을 저해시킬 수 있는 요인이 되기도 한다. 그렇기 때문에, 국내외에서 발의되는 법률 조항들을 분석하고 국내 업계 상황에 맞게 적용하여 개인과 기업의 경제적 자산을 보호해야 한다. 이를 위해 IoT와 관련한 법률 조항들을 개선하여 점차 규제해 나아가는 것이 필요하다.

만약 기업이 의무사항을 지키지 않아 피해가 발생할 경우에는 시스템을 구축한 주체에게 큰 책임이 뒤따르기에, IoT 기기를 제작하는 기업이나 개발자는 이러한 보안 의무를 반드시 지켜야 한다.

III. IoT 보안 관련 기능적 요구사항 분석

'IoT 보안 관련 기능적 요구사항'이 의미하는 바는 IoT 기기 보안에 대해 개발자나 설계자가 '할 수 있는 것'을 말한다. 본 장에서는 IoT 제작환경과 취약성, IoT 공통 보안 가이드 및 보안 인증 점검 사항들을 참고하여 IoT 기기에 적용할 수 있는 기능들을 살펴본다.

1. IoT 기기관련 현황 및 취약성 분석

지금까지 IoT 기기와 관련하여 많은 취약점들이 발견되고 관련된 해킹 공격 시범 사례나 실제 피해 사례들이 보고되고 있다. 이러한 공격들은 악성코드로 인한 감염, 데이터 및 기기 탈취, 관리자 권한 획득, 펌웨어 및 운영체제에 대한 불법적 침입 또는 접근을 통해 이뤄지며 개인정보의 유출이나 시스템 오작동, 또는 악성코드로 인한 DDoS와 같은 2차 피해를 유발한다. 현재 국내·외 기업들에 대한 가장 빈번히 발생하는 주요 공격방법들은 악성코드 및 랜섬웨어 공격으로 알려져 있다[6].

이러한 보안 취약점들은 IoT 기기 제작 환경과 연관이 깊다. Eclipse IoT Working Group의 IoT Developer Survey 2016은 현재 IoT 기기에 가장 많이 사용되고 있는 OS가 임베디드 리눅스로 73.1%의 압도적인 비율을 차지하고 있으며 앞으로도 IoT 기기 특성상 모든 산업군에서 이용성이 편리하고 비용이 저렴하다는 장점이 있어 앞으로도 많이 활용될 것으로 예상하였다[7].

이에 따라 임베디드 리눅스 관련 악성코드는 IoT 기기에 대한 가장 빈번한 공격 방법으로 사용될 소지가 크다. 현재 대표적인 임베디드 리눅스 악성코드로는 Aidra, Gafgyt, Kaiten, Mirai 등이 있으며, 대부분 변종 악성코드가 생성되어 전파되고 있다. 시장조사기관 가트너는 IoT 기기의 도입 대수가 2020년에 200억대를 초과할 것으로 예상하고 있으며[8], 이러한 기기들이 해킹되어 DDoS에 사용된다면 피해는 예측하기 힘들 정도로 커질 수 있다.

특히 Rootkit, Bootkit 형태의 악성코드는 위험하다. 시스템의 루트 권한 획득 후 프로세스 은닉 및 우회를 하므로 탐지가 어렵고, 접근 권한 및 보안 설정 등을 변경하거나 스니핑을 통해 개인정보를 유출할 수 있다. 가장 중요한 문제는 악성 프로그램을 삭제하더라도 부팅 시 재생성 되는 기능이 있어 사용자는 제조사의 도움 없이 악성코드를 삭제하기 굉장히 어려울 수 있다. 이러한 유형의 공격은 각종 악성 2차 피해를 유발하는 원인이 되며 IoT 기기에서는 가장 치명적인 공격이다.

또한, 최근 많이 발생하고 있는 랜섬웨어는 초창기 특정 파일 암호화부터 현재에는 MBR(Master Boot Record)까지 암호화하는 형태로 발전하였으며, 앞으로는 운영체제나 파일 시스템과 같은 중요 영역들을 암호화하는 방식으로 변화할 것으로 예측된다. 이러한 랜섬웨어가 IoT 기기에 공격할 경우 시스템 오작동으로 인해 사용 불능이 될 수 있으며, 이에 관련한 개인정보 침해뿐만 아니라 개인의 생명까지도 위태롭게 할 수 있다.

이러한 Rootkit 이나 Bootkit 과 같은 형태의 공격에는 Secure Boot와 같이 무결성을 검증하는 과정이 효과적인 방어 방법으로 사용된다[9][10]. 또한, 대부분의 악성코드는 루트 권한탈취가 목적이기에 강력한 접근 권한 제어 기술을 사용하는 것이 효과적이다. 그리고 암호화 및 인증 기능으로 기밀 정보를 보호하고, 무엇보다 시스템을 신뢰 된 데이터와 프로세스만으로 구성될 수 있도록 유지하는 것이 중요하다. IoT 기기의 특성상 다양한 분야에서 공격이 발생할 수 있고, 피해의 심각성과 규모가 클 것으로 예상되기에 개인정보보호와 안전에 대한 고려가 필수적이다.

2. IoT 기기의 보안 요구사항 및 인증 점검 항목

2.1 보안 인증 관련 기능 요구 사항

아직은 IoT 기기와 관련하여 명확하게 법률에 명시된 사항이나 의무 조치 기준이 없기 때문에 제품의 안전성과 보안성을 위해 각 보안 관련 인증이나 IoT 기기의 보안요구사항을 참고할 수 있다.

[표 1]은 ISO/IEC 27001, TCSEC, PIMS, CC 등과 같이 다양한 국내·외 보안 인증 관련 연구 분석을 통해 도출된 결과로 IoT 기기의 공통 주요 보안 인증 점검 항목들을 나타낸다[11].

표 1. IoT 기기의 공통 주요 보안 인증 점검 항목

Class	Check list
Confidentiality	– Lightweight encryption – Anti-malware – Access and permission control – Device authentication – Identification verification – IoT Standard based design
Integrity	– Data, Platform, S/W Integrity authentication and verification design – Secure boot
Availability	– Access control design – Closed firewall
Authentication Authorization	– User authentication and password setting – Mutual Authentication between devices – Ownership control – Identification verification

각 점검항목은 기밀성, 무결성, 가용성, 인증·인가로 구분되며 IoT 기기에서 적용할 수 있는 기능들을 나타낸다. IoT 기기의 보안 기능이 각 점검 항목을 따를 때 주요 기능으로는 암호화, 사용자 및 기기 인증, 접근 권한 제어, 무결성 측정·검증, 백신 프로그램의 설치 등이 있다. 그러므로 IoT 기기가 고수준의 보안인증을 획득하기 위해서는 주요 인증 점검 항목에 포함된 기능들이 구현되어 있어야 한다.

2.2 IoT 기기 등급에 따른 보안 기능 요구 사항

TTA 표준인 IoT 기기 등급 분류 및 보안 요구사항[12]에 따르면, IoT 기기는 프로세싱 능력에 따라 등급 0~3의 기기들로 분류될 수 있다. 일반적으로 개인정보를 취급 및 가공하는 프로세스를 갖추고 있는 기기는 1~3등급의 기기로 분류된다.

1등급의 기기는 TinyOS, NonoQplus와 같은 경량 운영체제를 탑재하며, 메모리나 리소스 사용에 제한이 있지만, 개인정보를 저장 및 가공하기 때문에 보안 요구사항에 대한 최소한의 기준이 된다.

2등급 기기는 Embedded Linux를 탑재할 수 있는 IoT 기기를 의미하며 리소스에 대한 제한이 거의 없어 1등급에서 요구되는 요구사항을 바탕으로 더 세부적인 기술들을 추가할 수 있다.

3등급의 기기는 Android, iOS를 탑재하고 전력을 제외한 모든 리소스의 제한이 없는 고급 장비를 의미하며, 사용자와의 상호작용이 자주 발생하여 다양한 보안 기술들이 적용될 것으로 예상한다.

TTA 표준에 따라, 이러한 기기들에 공통으로 요구되는 기능으로는 통신 및 데이터 암호화, 무결성 검증, 사용자 및 기기 인증, 권한 및 접근 제어, 상태 정보 전송 및 식별 정보 관리, 비밀번호 관리를 들 수 있다. 이러한 요구사항들은 중요 정보들을 다루는 기기들에서 사용하는 필수적인 보안 기술로 볼 수 있다.

2.3 IoT 공통 보안 가이드에서의 보안 요구사항

다음은 한국인터넷진흥원에서 제안한 IoT 공통 보안 가이드[13] 중 설계·개발 단계를 나타낸 것이다.

- 설계·개발 단계의 IoT 공통 보안 가이드 항목
  - IoT 기기의 특성을 고려한 보안 서비스의 경량화
  - IoT 서비스 운영환경에 적합한 접근 권한 관리 및 인증, 종단 간 통신보안 데이터 암호화
  - 소프트웨어 보안 기술과 하드웨어 보안 기술의 적용 검토 및 안전성이 검증된 보안 기술 적용
  - IoT 제품 및 서비스에서 수집하는 민감 정보(개인정보 등) 보호를 위해 암호화, 비식별화, 접근관리 등의 방안 제공
  - IoT 서비스 제공자는 수집하는 민감 정보의 이용목적 및 기간 등을 포함한 운영체제 가시화 및 사용자에 투명성 보장
  - 소스코드 구현단계부터 내재될 수 있는 보안 취약점을 사전에 예방하기 위해 시큐어 코딩 적용
  - IoT 제품·서비스 개발에 사용된 다양한 S/W에 대해 보안 취약점 점검 수행 및 보안패치 방안
  - 펌웨어·코드 암호화, 실행코드 영역제어, 역공학 방지 기법 등 다양한 하드웨어 보안 기법 적용

2.4 IoT 기술적 보안 요구사항

IoT 관련 기술안내서는 법제적인 강제성을 띠고 있지는 않지만 IoT 제품 설계·개발 단계에서 참고하는 데 유용하다[13]. 대부분의 취약점은 설계·개발 단계에서 보안성이 고려되지 않아 발생하는 경우가 대부분이기에 이러한 요구 사항은 필수적으로 지키는 것이 권장된

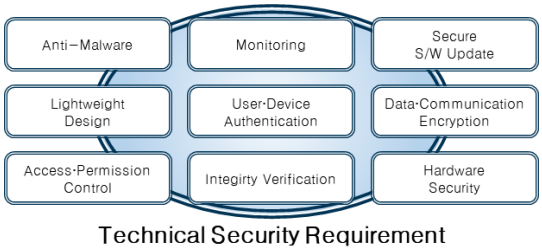


그림 2. 기술적 보안 요구사항

다. 보안 인증 점검항목, IoT 공통 보안 가이드라인, 개발환경 및 취약점들을 고려하였을 때. 필수적으로 적용해야 할 것으로 제안되는 기술적 보안 요구사항은 [그림 2]와 같다.

IV. IoT 보안 운영체제 기반 기술

1. IoT 보안 프레임워크 기반 기술 및 기능

법제적인 의무사항은 규제가 산업을 저해할 수 있다는 측면이 있어 현행법상 기준에는 적용 범위가 넓은 경향이 있다. 반면에 기능적인 요구사항은 IoT 기기에 맞게 적용 가능한 보안 기능들의 세밀한 기준을 제시하지만 강제성을 띠고 있지는 않다. 그렇기 때문에 이러한 요구사항들을 참고하여 IoT 보안 프레임워크 기반 기술들을 선정하고 해당 기술들을 사용하여 IoT 기기 내에 보안성을 확보해야한다.

[표 2]의 각 항목은 “개인정보의 기술적·관리적 보호 지치 기준”과 “개인정보의 안전성 확보 조치 기준”을 참고하였으며, 각 행은 도출된 법제적 요구사항을, 각 열은 도출된 기술적 요구사항들을 나타낸다. 이는 IoT 기기들에 적용할 수 있는 기술적 요구사항들과 이에 해당되는 법제적 요구사항의 상관관계를 나타낸다. 예를 들어 앞서 도출된 기술적 요구사항 중 하나인 백신 프로그램(Anti-Malware) 항목은 “개인 정보의 기술적·관리적 보호조치” 제7조와 “개인정보의 안전성 확보 조치 기준”의 제9조를 만족한다. 그렇기 때문에 백신 프로그램 기술은 기술적 요구사항과 법제적 요구사항을 만족하기에 IoT 환경에 필요한 기술로 판단할 수 있다. 하지만 경량화 설계(Lightweight Design) 기술적 요구사항은 법제적 요구사항을 만족하는 항목이 없다. 현재 법률상 IoT 기기에 대한 단일 법률이나 이를 언급한 항목이 존재하지 않아 필수적으로 포함되진 않았지만, IoT 기기에서 경량화에 대한 요구가 끊임없이 있기 때문에 개발자는 구현과정에서 이에 대해 고려할 수 있다. 또한 법제적 요구사항에서의 물리적 보안은 특정 정보를 저장하기 위한 물리적 보관 장소나 출입통제 절차를 의미하지만, IoT 기기에 대해 적용했을 때 개인정보를 하드웨어로 구성된 저장소 내부에 격리 및 보관할 수 있다는 특징이 있어 [표 2]와 같이 법률적 요구사항과 기술 요구사항의 상관관계를 나타내었다.

표 2. 법제적 요구사항과 기술적 요구사항의 상관관계

Technical Requirement \ Legal Requirement	Security Application	Connection Lag Management	Privacy Encryption	Access Permission Management	Access Control System	Internal Management Policy	Physical Control
Monitoring		O			O		
Anti-Malware	O					O	
Secure S/W update	O		O			O	
Lightweight Design							
File · Communication Encryption			O		O	O	
Users · Devices Authentication			O		O	O	
Access·Permission Control		O		O	O	O	
Integrity Verification			O		O	O	
Hardware Security Module							O

표 3. IoT 보안 프레임워크 기반 기술 및 기능

Security Technology	Security Framework Functions
Monitoring	Anomaly Detection
Anti-Malware	Recovery
Secure update	Encryption·Authentication
File · Communication Encryption	Encryption·Authentication
Users · Devices Authentication / Authorization	Encryption·Authentication
System Audit Log	Audit Log
Mandatory Access Control	Access·Permission Control
File Integrity Measurement and Verification	Integrity Measurement and Verification
Secure Boot / Trusted Boot	Integrity Measurement and Verification
Hardware Security Module	Hardware Security Module

[표 3]은 [표 2]를 근거로 하여 도출된 보안 프레임워크 기반 기술 및 기능을 나타낸다. 보안 프레임워크 기술은 기술적 요구사항과 법제적 요구사항을 만족하는 기술을 의미하며, Secure Boot/Trusted Boot나 System Audit Log와 같은 중요 기술이 추가되어 도출되었다. IoT 기기는 접근 권한관리 기술을 통해 불법적인 접근을 차단하고 권한을 분산하여 루트 권한의 탈취가 주목적인 악성코드를 봉쇄해야 한다. 또한 모니터링 기술을 사용하여 책임 추적을 위해 모든 불법적인 접근과 프로세스를 탐지해야 한다. 이를 위해 시스템 레벨에서의 감시 로그 기술과 사용자영역에서의 악성코드 모니터링 기술이 있다. 또한, 개인정보를 안전하게 보호하기 위해서는 파일 및 파일시스템, 통신 채널을 암호화하는 기술이 필요하며. 기기 및 사용자 인증 기술을 통해 비인가된 사용자나 기기의 접근을 차단해야 한다. 이러한 암호화 및 인증기술을 사용하여 펌웨어나 OS, 소프트웨어 등을 안전하게 원격 업데이트하는 기술이 가능해진다. 또한 데이터의 위·변조를 막기 위한 데이터 무결성 검증 기술이 제공되어야 하며, 검증된 소프트웨어만을 수행하도록 하는 부팅과정에서의 보안 기술은 펌웨어부터 운영체제, 소프트웨어까지 안전한 환경을 조성하는 데 중요한 역할을 하게 된다. 그중에서도 암호화 기술은 데이터의 기밀성과 무결성을 제공하기 위해 중요한 기술이며 다양한 기능에 활용되기 때문에 암호화 키 관리가 가장 중요하다. 하드웨어 보안 모듈은 이러한 암호화키를 보관하는데 필수적인 기술이며 이를 통해 각종 보안 기능의 신뢰성을 높일 수 있다.

보안 프레임워크 기반 기능은 크게 접근 권한 제어, 암호화, 인증, 무결성 검증, 감시 로그, 하드웨어 보안 모듈, 이상 탐지, 복구 프로그램 등으로 구분할 수 있다. [그림 3]은 [표 3]을 토대로 IoT 기기에 적용될 수 있는 신뢰성 있는 컴퓨팅 환경을 위한 보안 프레임워크 기반 기술들을 나타낸 그림이다.

2. IoT 보안 운영체제 기반 기술

개발자나 사용자 입장에서는 기기에 설치된 보안 소프트웨어만으로 모든 의무를 다했다고 생각하는 경우가 종종 있다. 그러나 소프트웨어 레벨에서의 보안은

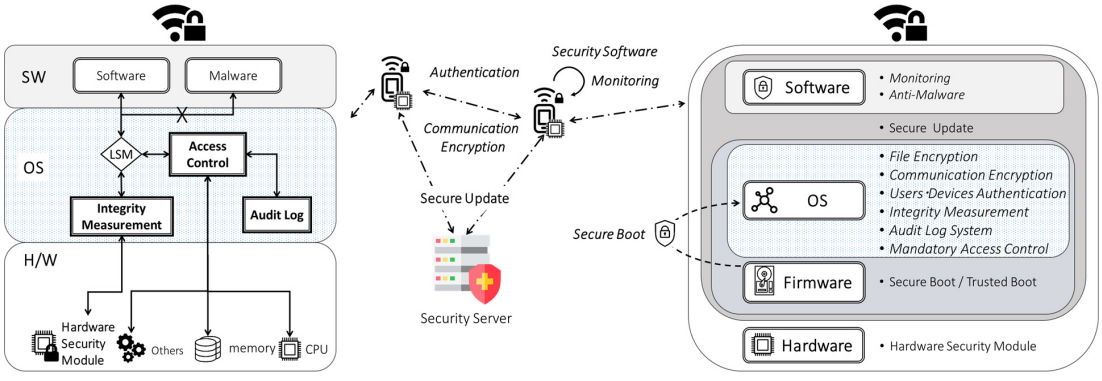


그림 3. IoT Security framework based technologies for Trusted Computing

몇몇 단점이 있기에 소프트웨어 레벨에서의 보안 솔루션만으로 보안 의무 조치를 다 했다고 볼 수 없다. 근본적으로, 소프트웨어 레벨에서의 보안은 알려지지 않은 취약점에 대한 제로 데이 공격이나 서비스 거부 공격에 대해서는 대처하기 힘들다는 한계가 있다. 만약 이러한 공격들을 통해 루트 권한이 탈취될 경우 기기 간 통신을 통해 순식간에 악성코드가 전염될 것이며, 피해가 기하급수적으로 커질 수 있다. 알려진 악성코드에 수동적으로 대응하는 방식의 소프트웨어 레벨에서의 보안만으로는 IoT 기기를 보호하기에는 역부족이며, IoT 기기에는 보다 근본적으로 악성코드를 대비할 시스템 레벨에서의 강력한 보안 수단이 필요하다.

보안 운영체제 기술은 이에 대한 대안이 될 수 있다. 보안 운영체제는 운영체제상의 보안성 결함으로 인해 발생할 수 있는 각종 취약점으로부터 시스템을 보호하기 위해 운영체제 내에 부가적으로 보안 기능을 향상시킨 보안 커널을 의미한다. 따라서 보안 운영체제는 사용자에 대한 인증, 접근 통제, 암호, 무결성 인증, 감사 등의 기능을 수행하며 시스템 레벨에서의 보안 수행을 통해 안전한 시스템 환경을 제공한다. 더불어 엄격하게 물리적으로 분리된 영역에 중요 정보를 보관함으로써 공격자가 중요정보를 탈취하거나 침입의 흔적을 제거하는 것을 방지한다.

[그림 4]는 [표 3]에서 시스템 레벨 보안관점에서 도출한 보안 운영체제 기반 기술을 나타낸 것이다. IoT 기기의 보안성과 안전성, 그리고 법제적 의무를 위해 언급된 기능들은 반드시 적용되어야 한다. 각 기능이 속한 영역은 적용될 수 있는 컴포넌트의 범위를 나타낸다. 예를 들어 암호화나 인증은 소프트웨어, OS, 펌웨어 영역에 걸쳐 적용될 수 있으며 강제적 접근 제어나 감시 로그 기술은 OS 레벨에서 적용 가능하다는 것을 나타낸다.

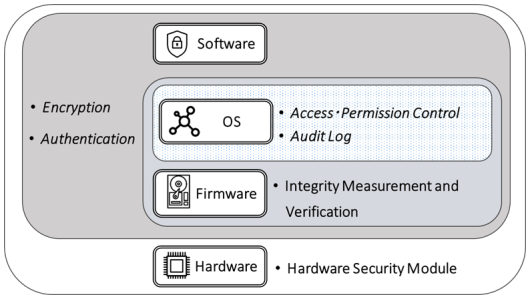


그림 4. IoT Secure-OS based technologies for Trusted Computing

그 밖에도, IoT 기기 등급 1~3으로 분류되는 보안 요구사항들은 대부분 시스템 레벨에서의 보안 운영체제 기반 기능들로 적용할 수 있으며 기존에 제공되는 공개 소프트웨어와 같은 보안 기술들을 활용하거나 이미 정의된 보안 기술이 존재하기에 경제성·보안성·기술성 측면에서 유리하다. 이럴 경우 응용 레벨에서의 보안 기술인 'Anomaly Detection'이나 'Recovery'와 관련된 백신 프로그램을 기업이나 개발자의 경제적 사정을 고려하여 적용할 수 있게 된다.

다음의 세부 항들은 IoT 보안운영체제를 구현하기 위해 도출된 기술들에 대한 설명을 포함한다.

2.1 Hardware Security Module

하드웨어 보안 모듈은 PC, 서버, 임베디드 기기 등 별도의 보안을 위한 칩을 만들어 암호화를 위한 공개키, 개인키, 비밀번호, 전자서명 등을 넣어 보관하고 칩을 물리적으로 훔치지 않는 한 해킹이 불가능하도록 하는 것을 목표로 하는 하드웨어 칩 기술이다.

본 논문에서 제시하는 시스템 레벨 보안 프레임워크 기능 중 유일하게 추가적인 비용이 들어가는 기술이지만, 하드웨어 보안 모듈 기능을 통해 무결성 인증이나 암호화와 같이 관련된 보안 기술의 신뢰성을 증가시킬 수 있고, 다양하게 응용할 수 있기에 필수적으로 권장된다.

대표적인 하드웨어 보안 모듈은 TCG(Trusted Computing Group)에서 제안한 TPM(Trusted Platform Module)이며 현재 20억대 이상의 기기에 적용되어 있다[14]. TPM의 주요 목적은 키를 안전하게 보관하는 것이기에 IoT 기기 보안과 관련하여 서명, 인증, 암호화 등에 폭 넓게 사용된다. 하드웨어 칩 내부에서 개인 키를 발생하고, 특정 한 시점에 칩 밖으로 전송될 키들을 암호화하는 것에 의해서 TPM은 악의적인 소프트웨어가 키에 절대 접근할 수 없다는 것을 보장한다. 심지어 개인 키의 소유자도 칩의 외부에서 칩 내부의 암호화

되지 않은 상태의 키를 확인할 수 없기 때문에 어떠한 이유에서도 파밍이나 피싱에 개인 키를 제공할 수 없다. TPM내 레지스터에 Sealing된 값은 내부적으로 값이 변하지 않아야 Unsealing될 수 있기 때문에 무결성 검증에 활용된다. TCG에서는 TSS(TCG Software Stack)을 통해 공개 라이브러리를 제공하며, 사용자는 이를 통해 서명키 생성, 서명과 검증, 데이터 해시 및 Sealing, 공개키 생성, 의사난수 생성 등을 사용할 수 있다[15]. TPM이 사용될 경우 물리적인 하드웨어 보안, 펌웨어, OS, 소프트웨어 레벨까지 보안을 수행할 수 있다.

2.2 Integrity Measurement and Verification

무결성 측정 및 검증 기술은 IoT 기기 내의 안전한 데이터 사용과 프로세스 실행을 위해 중요하다. 대표적으로는 파일 무결성 측정 기능이나 부팅과정에서의 신뢰성을 보장하는 안전부팅(Secure Boot)[16] 및 신뢰부팅(Trusted Boot)[17] 등이 있다.

파일 무결성 측정 기능은 각 파일의 무결성 보장을 위해 해시값을 측정하고 이에 대해 리스트에 보관한 뒤, 리스트 안에 저장된 값과 접근될 파일의 측정값을 비교하여 무결성을 보장한다. 이러한 기능은 중요 파일에 대한 무결성을 보장함으로써 시스템 운용 중에 트로이 목마와 같은 악성코드로부터 개인정보를 보호하기 위해 필수적인 기능이다. 대표적인 공개 소프트웨어로는 IMA(Integrity Measurement Architecture)가 있다[18]. Linux 2.6버전 이후로 공식적으로 채택된 IMA는 실행 파일의 무결성을 측정하며 LSM(Linux Security Module)로 접근 명령이 발생할 경우 이를 후킹하여 실행되기 전 무결성 측정·검증을 수행한다.

Secure Boot는 IoT 기기 환경의 무결성을 위해 반드시 필요하다. IoT 기기는 임베디드 기기 특성상 전력 문제가 있어, 부팅이 잦을 수 있고 Rootkit이나 Bootkit과 같은 유형의 악성코드가 삽입되어 있다면 삭제하더라도 재 생성되어 무결성을 보장할 수 없다. Secure Boot는 설계 및 제조단계에서 제공하는 펌웨어 및 운영체제의 무결성을 보장하며, 오염되지 않은 데이터와 프로세스만을 실행시킬 수 있게 환경을 구축한다. Secure Boot 는 전원 인가 후 펌웨어의 가장 초기 코드 모듈인 CRTM(Core Root of Trust Module)를 시작으로 부트로더, OS, 시스템 파일 및 드라이버 등 각 단계별 신뢰체인을 형성해 나아가는 방식을 따르며 무결성 검증 실패가 발생할 경우에는 초기 상태로 돌아가거나 펌웨어 업데이트를 통해 무결성을 회복한 뒤 다음 부팅 과정을 수행하는 방식으로 진행된다. IoT 기기에서 참고할 수 있는 공개소프트웨어로는 Verified U-Boot가 있다. Verified U-Boot는 펌웨어가 공개되지 않은 시스템에서 사용할 수 있는 Secure Boot 방법 중 하나로, U-Boot가 사용가능할 경우 손쉽게 Secure Boot 기능을 구현할 수 있다. Verified U-Boot는 [그림 5]과 같이 개인키로 서명된 FIT(Flat Image Tree) 이미지를 U-boot dtb 파일 내 포함된 공개키로 검증한 뒤 성공적일 경우 커널을 로드하는 U-Boot 부트로더를 사용한다[19].

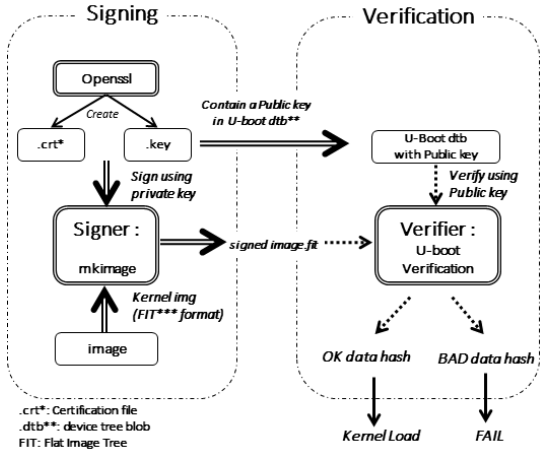


그림 5. Verified U-Boot

TPM 기술은 Secure Boot와 함께 사용될 수 있으며, 측정된 해시 값을 저장하거나 암호화키를 저장하는데 사용하여 부팅과정의 신뢰성을 높일 수 있다. 펌웨어가 공개되어 있을 경우 TSS 라이브러리를 추가하여 키 값이나 해시 값을 TPM칩에 Extend 함으로써 검증을 수행할 수 있다. Secure Boot는 부팅중간에 무결성 검증 실패가 발생할 경우 부팅과정을 중단하기에 특정 경우에서는 이로 인해 문제가 발생할 수도 있다. 이렇게 부팅을 중단하면 안 되는 경우에는 Trusted Boot를 사용

할 수 있다. Trusted Boot의 일반적인 과정은 Secure Boot와 흡사하지만, 차이점은 부팅을 중단하지 않는다는 것이다. 각 부팅 단계에서 검증된 해시값을 측정만 하고 부팅 완료 후 검증 프로세스를 수행하여 측정된 값들을 원격 서버나 파일시스템에 존재하는 리스트와 비교하여 검증을 수행한다. 그렇기 때문에 TPM과 같은 하드웨어 보안 칩이 필수적으로 요구된다.

2.3 Access·Permission Control

대중적으로 사용되는 접근제어 모델은 임의적인 접근제어[20]로 정보에 대한 사용을 요청하는 사용자의 신원에 근거를 두고 접근 허가를 결정하는 방식이다. 하지만 이러한 방식은 사용에는 편리하지만 시스템의 전반적인 보안 관리에는 용이하지 않다는 단점을 지닌다.

반면에 강제적 접근제어(Mandatory Access Control)는 정보와 사용자에게 보안 등급을 부여하고 세밀한 정책을 통해 접근제어를 수행한다. IoT 기기의 시스템 레벨에서 강제적 접근제어 정책을 사용할 경우, 응용 수준의 보안 제품이 막지 못하거나 탐지할 수 없는 공격을 근본적으로 차단할 수 있으며 내부 사용자에 의한 공격이나 실수 등에서도 대응할 수 있다는 장점이 있다. 무엇보다 루트권한의 탈취에도 권한이 분산되어 있기에 피해를 최소화할 수 있다.

강제적 접근제어는 크게 DTE(Domain Type Enforcement), MLS(Multi-level Security), RBAC (Role-based Access Control)로 구성되어 있다.

DTE은 모든 주체와 자원에 대하여 지정된 타입으로 Labeling 한다[21]. 강제적 접근 제어에서 사용하는 정책은 타입을 기준으로 생성되고 시스템 내 발생하는 모든 명령들은 정책을 기준으로 관리되기에 정책 기반 접근제어로도 볼 수 있다. Type Labeling은 TCSEC B1 기준이상이 되기 위한 최소한의 조치이며 일반 리눅스는 TCSEC C1 기준을 만족한다고 알려져 있다.

MLS는 다중 레벨 기반 보안으로 수학적으로 증명된 BLP(Bell-LaPadula)모델을 통해 각 주체나 자원에 대한 기밀성의 등급을 부여하여 접근을 제한하는 방법을 말한다[22]. 이를 통해 중요자원에 대한 차별적인 접근 프로시저를 수행하게 하는 것이 가능하다. MLS 또한 주체와 객체에 대한 구분을 통해 정책화한 뒤 수행을 하기에 Type Labeling과 밀접한 연관을 맺는다.

RBAC는 주체의 역할에 따라서 권한을 차등 부여하게 되는 기술을 일컫는다[23]. 대부분 역할은 특정 기업이나 단체의 규정이나 규칙에 맞게 생성하며 각 사용자들을 이 역할에 따라서 권한을 각기 다르게 제공한다. RBAC를 사용할 경우 정보통신망 법 제28조에 명시된 기술적·관리적 의무 조치의 내부관리 계획의 수립·시행과 접근 통제장치의 설치·운영을 IoT 기기에 반영할 수 있다.

리눅스 2.6버전으로 공식적으로 채택된 SELinux는 DTE, MLS, RBAC와 같은 강제적 접근 제어를 기반으로 리눅스에 강력한 시스템 레벨 보안 기능을 제공하며 SELinux 정책을 수정 및 적용 할 경우 시스템에 맞게 보안 기능을 설정할 수 있다[24].

2.4 Audit Log

감시 로그 시스템(Audit Log System)은 시스템 내에서 발생하는 모든 접근들을 감시하고 기록한다. 이러한 기술은 시스템 내에서 마치 블랙박스와 같은 역할을 수행한다. “개인 정보의 기술적·관리적 보호 조치 기준”과 “개인정보의 안전성 확보 조치”에 명시된 “접속기록의 위·변조를 위한 조치”를 위해서, IoT 기기는 반드시 감시 로그를 실행해야하며, 이는 책임 추적 및 원인 규명을 위해 중요하다. 시스템 레벨에서 감시 로그 시스템은 기본적으로 OS 레벨에 위치하여 각종 시스템 콜에 대한 정보들을 기록한다. 특히 SELinux에서는 강제적 접근제어와 함께 OS 레벨에서 발생하는 불법적인 접근들에 대한 로그를 기록한다. SELinux를 적용할 경우 강제적 접근 제어와 함께 감시 로그 시스템을 사용할 수 있다.

2.5 Encryption

법률에 명시된 것처럼 암호화 기술은 개인정보를 보호하기 위한 핵심 기술이다. 또한 암호화 기술은 [그림 2]에서 제시된 것처럼 다양한 IoT 기기 컴포넌트에 활용된다. 암호화 기술은 신뢰성 있는 시스템 구축을 위해 필수적인 기술이며 안전한 업데이트 기능부터, 무결

성 검증, 파일이나 통신 채널의 암호화, 인증, 서명 등 다양하게 활용된다.

IoT 기기에서 사용할 수 있는 대표적인 암호화 관련 공개 소프트웨어로는 OpenSSL이 있다[25]. OpenSSl은 SSL(Secure Socket Layer)· TLS(Transport Layer Security)프로토콜 기반이며 데이터 암호화 및 데이터 암호화 통신, 서명 등에 사용할 수 있다. 예를 들어 OpenSSL을 활용할 경우 서버와 클라이언트가 동일한 암호화 기법을 선택한 뒤, 인증서를 사용하여 서버를 인증하고 공개키 방식의 암호화를 통해 세션 키를 안전하게 교환 한 뒤 인증이 완료되면 통신하는 방법을 통해 암호화 데이터 통신기술을 구현할 수 있다.

파일시스템 암호화 기능은 중요 파일들을 암호화하고 관리하기에 적합하다. 대표적으로 eCryptFS는 커널에 대한 변경 없이 VFS(Virtual File System) 위에 지정된 디렉토리에 암호화 파일을 마운트하여 관리한다[26]. 이때 키를 TPM과 같은 하드웨어 보안 칩을 사용하여 관리할 경우 시스템의 신뢰성을 높일 수 있다.

암호화 기능은 기밀성이나 무결성을 위해 중요한 기능이지만 암호화 강도에 따라 그 안전성이 달라질 수 있다. IoT 기기는 경량화 및 저 전력 환경이기에 암호화 알고리즘의 강도를 적용하는데 있어서 성능 상 걸림돌이 될 수 있다. 한국인터넷진흥원에서 제공하는 국산 경량암호화알고리즘을 사용할 경우 IoT 기기 환경에 맞는 강도의 보안성을 유지하면서 성능을 향상할 수 있다. 대표적으로 HIGHT는 초경량 블록암호 알고리즘으로 128비트 마스터키, 64비트 평문으로부터 64비트 암호문을 출력하며, 간단한 알고리즘 설계와 안전성, 효율성을 고려하여 제작되었다[27]. LEA(Lightweight Encryption Algorithm)은 경량 환경에서 기밀성을 제공하는 128비트 블록암호알고리즘이다[27]. 이러한 알고리즘들을 사용할 경우 사용비용에 대한 부담 없이 IoT 기기 환경에 맞게 암호화 기능을 구현할 수 있다.

2.6 Authentication

인증 기능은 데이터의 무결성을 보장하기 위한 기술이다. 인증 기능은 암호화 기능과 함께 다양한 범위에 걸쳐 사용되며 특히 사용자 및 기기인증, 안전한 업데이트 등에 사용된다. 대표적인 인증 시스템에는 생체인식시스템과 OTP(One-Time Password)등이 있다. 이러한 인증시스템에서는 해시함수를 사용하게 되는데 이러한 해시 함수는 사용자 및 메시지 정보 변조방지를 위한 핵심 기술로 사용된다.

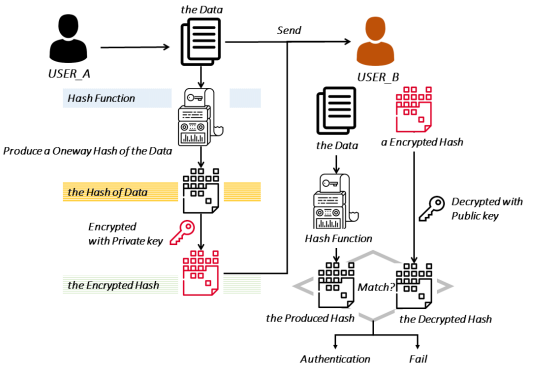


그림 6. Authentication using the hash function

[그림 6]은 이러한 해시함수를 통한 인증과정의 한 예를 나타낸다. 송신자의 데이터는 일방향 해시함수를 통해 데이터의 해시를 생성하고 이를 송신자의 개인키로 서명한다. 이후 수신자는 데이터의 원본과 서명된 해시를 수신하게 되며, 공통된 해시함수를 통해 원본에서 해시값을 추출하고, 송신자의 공유키로 복호화한 해시값과 비교하여 일치여부를 확인한다. 일치할 경우 송신자의 데이터임을 인증할 수 있게되는 것이며, 부인을 방지할 수 있다. 보안운영체제에서 해시함수는 다양한 인증의 용도로 사용될 수 있다. 그 이유는 해시함수는 파일이나 메시지의 무결성을 검증하는 용도로 사용되며 전자 서명, 메시지 인증코드, 암호키 유도, 의사난수 생성 등에 있어서도 핵심적이기 때문이다. IoT 기기 내 안전한 환경을 구축하기 위해서 사용할 수 있는 공개 라이브러리는 한국인터넷진흥원에서 제공하는 LSH (Lightweight Secure Hash)[27]이나 TPM에서 제공하는 SHA가 있다. LSH는 다양한 환경에서 적용될 수 있고 국제 표준인 SHA2·SHA3에 비해 2배 이상의 성능을 제공한다. TPM이 지원될 경우에는 TPM에서 제공하는 TSS 라이브러리를 사용하여 해시함수 기능을 사

용할 수 있다[15].

3. 오픈소스를 활용한 IoT 보안 플랫폼

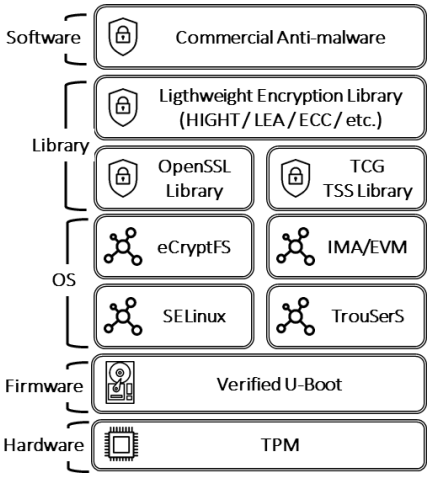


그림 7. IoT Security Platform using Open Source

앞서 제시한 IoT 기기 관련 보안 프레임워크 기술들은 오픈소스를 활용하여 구현할 수 있으며, [그림 7]은 본 논문에서 제시한 필수 기술들을 오픈소스 플랫폼 형태로 나타낸 그림이다. 해당 기술들은 누구나 무료로 접근할 수 있기에 IoT 기기를 제작하기 위해 리눅스 시스템을 사용하는 모든 Start-up 기업이나 개인 개발자들은 이를 참고할 수 있다. IoT 기기가 [그림 7]과 같은 오픈소스를 활용하여 보안 기술들을 갖추고 이에 해당하는 보안 기능들을 수행한다면 기기복제, 기기 위·변조, 펌웨어 추출, 통신 내용 탈취 및 위·변조, 악성코드 감염에 대비할 수 있다. 먼저 무결성 검증과 관련된 IMA/EVM, Verified U-Boot, TPM 등을 활용할 경우, 위·변조에 효과적으로 대응할 수 있으며, 무결한 시스템 내부 환경을 구축할 수 있다. 특히 TPM과 같이 생성한 해시값을 하드웨어적으로 암호화하여 저장하고, 검증 시에 복호화하여 비교하는 과정은 물리적인 탈취에도 중요 정보에 대한 키를 보관하고 있기에 공격자는 함부로 복호화를 수행할 수 없다. 또한, eCryptFS를 사용할 경우, 특정 디렉토리에 저장하는 파일들을 암호화하여 보관하는 것이 가능해진다. 이렇게 될 경우 각 파일의 해시값을 TPM을 통해 저장하여 무결성이 인증될 경우 복호화하게 하는 것도 가능하다. SELinux와 같은 강력한 접근제어 시스템을 사용할 경우, 정책으로 권한을 분리하여 최소권한을 수행하기 때문에 정보의 유출 피해를 최소화하여 침해 후 대응이 가능하다는 장점이 있다. 이는 사용자 레벨의 프로그램으로는 할 수 없는 시스템 레벨의 접근 권한 제어시스템만의 큰 장점이다. 또한, 이 모든 접근기록을 시스템 로그로 관리하기에 침해 후 책임추적을 위한 방법으로 효과적으로 접근제어 관련 정보가 활용될 수 있다. 이는 명확한 실행 주체와 사용되는 자원 객체들에 대한 정보가 명시되기에 특정 문제가 발생한 지점과 관련한 정확한 정보를 획득할 수 있다. 또한, 역할 기반의 접근제어로 허용되지 않은 인가로 IoT 기기에 원격 접근할 경우 모든 명령이 제한되기 때문에 SELinux 정책이 바뀌지 않는다면 IoT 기기를 안전하게 보호할 수 있게 된다. 사용자는 오픈소스로 구성된 IoT 플랫폼을 사용할 경우 경제적 이익을 취하며 시스템 레벨의 강력한 보안을 수행할 수 있게 된다.

V. 결론

현재 국내 IoT 시장은 초기단계로 압도적인 사업자가 없어 시장 선점을 위한 치열한 경쟁이 벌어지고 있다. 특히 신생 기업이나 개발자는 경제적인 이유로 보안을 소홀히 하는 경우가 많고 사업자간 다양한 방식으로 개발이 진행되기 때문에 이러한 차이점에서 발생하는 취약점들을 예방해야한다.

따라서 본 논문에서는 법제적인 관점에서 지켜야할 의무들과 IT 전문가들이 고려한 IoT 보안환경 요구사항들을 정리하여 시스템 레벨의 IoT 보안 운영체제 기반 기술들을 강조하였다. 시스템 레벨에서의 보안기술은 근본적으로 시스템 무결성을 유지하는 것을 원칙으로 하여 설계·제조 단계에서부터 기업에서 제공하는 인증된 시스템에 대한 악의적인 변경을 차단하는 것을 목표로 한다. 특히 산업현장에서 사용되는 IoT 정밀 기기나, 차량, 스마트 디바이스들은 고급사양단말을 사용하며 중요정보에 대한 처리를 수행하기에 이러한 보안

운영체제 기술이 요구된다. 시스템 레벨에서의 보안 기술은 이미 존재하는 공개 소프트웨어를 활용 및 참고할 수 있기 때문에 비교적 적용이 쉽고 비용이 적게 든다는 장점이 있다. 특히 보안 운영체제 기반 기술을 사용할 경우 해킹 공격으로 인한 피해를 최소화할 수 있으며, 원인분석과 책임 추적에 도움을 줄 수 있다. 그리고 강제적 접근제어를 사용하여 집중된 권한으로 인한 부작용을 차단할 수 있으며, 무결성 검증 과정을 통해 신뢰성 있는 기기 환경을 조성할 수 있다. 특히 IoT 기기의 도난과 분실에도 중요 정보의 유출을 막을 수 있다.

향후 연구과제로는 시스템 레벨 보안 기술이 적용된 경량 보안 RTOS의 요구사항 도출, 설계 및 개발과 SKPP(Separation Kernel Protection Profile)을 만족하는 고 신뢰성 보안 RTOS 설계 및 개발이 있다.

참 고 문 헌

[1] www.kisa.or.kr/uploadfile/201412/20141230112413 0506.pdf

[2] 손승우, 박장혁, 문수미, "사물인터넷 사업자를 위한 정보보안 법률의 개선 방안 연구," LAW REVIEW, 제57권, 제1호, pp.181-215, 2016(2).

[3] 이동혁, 박남제, "IoT 기기의 보안성 확보를 위한 제도적 개선방안," 정보보호학회논문지, 제27권, 제3호, pp.607-615, 2017(1).

[4] http://www.kisa.or.kr/public/laws/lawsTrend_List.jsp

[5] http://www.dooit.co.kr/survey/report/index/182097

[6] 문해은, 곽성현, 장격익, 곽기용, *2016년 국내외 사이버 위협정보 심층분석 연구*, KISA 연구 보고서, 2016.

[7] https://iot.ieee.org/images/files/pdf/iot- developer-survey-2016-report-final.pdf

[8] http://www.gartner.com/document/2625419?ref=QuickSearch&sthkw=G00259115

[9] Hyungon Moon, Hojoon Lee, Ingoo Heo, Kihwan Kim, Yunheung Paek, and Byunghoon Kang, "Detecting and preventing kernel rootkit attacks with bus snooping," IEEE Transactions on Dependable and Secure Computing, Vol.14, No.2, pp.145-157, 2017(4).

[10] Eugene Rodionov, Alexander Matrosov, and David Harley, "Bootkits: Past, Present and Future," VB Conference, 2014.

[11] 공희경, 구혜경, 조현웅, 강지성, *IoT 디바이스 보안 인증 기반 연구*, KISA 연구보고서, 2015.

[12] http://www.tta.or.kr/data/ttas_view.jsp?totalSu=758&by=desc&order=publish_date&rn=1&pk_num=TTAK.KO-12.0298&nowSu=295

[13] https://www.kisa.or.kr/public/laws/laws3_View.jsp?cPage=1&mode=view&p_No=259&b_No=259&d_No=80&ST=&SV=

[14] https://www.trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.38.pdf

[15] https://trustedcomputinggroup.org/tcg-software-stack-tss-specification/

[16] K. Dietrich and J. Winter, “Secure boot revisited,” In Young Computer Scientists, ICYCS 2008, The 9th International Conference for IEEE., pp.2360-2365, 2008(11).

[17] https://www.trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientImplementation_1-21_1_00.pdf

[18] R. Sailer, X. Zhang, T. Jaeger, and L. Van Doorn, "Design and Implementation of a TCG-based Integrity Measurement Architecture," USENIX Security Symposium, pp.223-238, 2004(8).

[19] S. Glass, "Verified Boot in Chrome OS and how to make it work for you," Embedded Linux Conference Europe., 2013(10).

[20] Henk C. A. van Tilborg and Sushil Jajodia, *Encyclopedia of Cryptography and Security*, Springer US, 2011.

[21] L. Badger, D. F. Sterne, D. L. Sherman, K. M.

Walker, and S. A. Haghighat "Practical domain and type enforcement for UNIX," Security and Privacy, Proceedings 1995 IEEE Symposium on IEEE, pp.66-77, 1995.

[22] D. E. Bell and J. L. Leonard, "Secure computer systems: Mathematical foundations," No.MTR-2547-VOL-1, MITRE CORP BEDFORD MA, 1973.

[23] 변창우, 박석, "비밀성와 무결성을 보장하는 역할기반 접근제어모델," 정보호호학회논문지, 제15권, 제3호, pp.13-29, 2005(6).

[24] F. Mayer, K. Macmillan, and D. Caplan, *SELinux by Example*, Prentice Hall, 2006.

[25] https://www.openssl.org/

[26] https://www.ecryptfs.org/

[27] http://seed.kisa.or.kr

저 자 소 개

고 재 용(Jae-Yong Ko) 준회원

- 2016년 2월 : 충남대학교 컴퓨터공학과(공학사)
- 2018년 2월 : 충남대학교 컴퓨터공학과 석사과정 수료

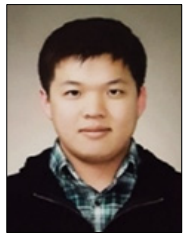
<관심분야> : 운영체제, 실시간 운영체제, 보안 운영체제, 실시간 보안 운영체제

이 상 길(Sang-Gil Lee) 정회원

- 2014년 2월 : 충남학교 컴퓨터공학과(공학사)
- 2016년 2월 : 충남학교 컴퓨터공학과(공학석사)
- 2018년 2월 : 충남학교 컴퓨터공학과 박사과정 수료

<관심분야> : 실시간 운영체제, 임베디드 시스템

김 진 우(Jin-Woo Kim) 준회원



- 2017년 8월 : 충남대학교 컴퓨터공학과(공학사)
- 2017년 9월 ~ 현재 : 충남대학교 컴퓨터공학과 석사과정 재학

<관심분야> : 운영체제, 실시간 시스템, 보안 운영체제

이 철 훈(Cheol-Hoon Lee) 정회원

- 1983년 2월 : 서울대학교 전자공학과(공학사)
- 1988년 2월 : 한국과학기술원 전기 및 전자공학과(공학석사)
- 1992년 2월 : 한국과학기술원 전기 및 전자공학과(공학박사)
- 1983년 3월 ~ 1986년 2월 : 삼성전자 컴퓨터 사업부 연구원
- 1992년 3월 ~ 1994년 2월 : 삼성전자 컴퓨터 사업부 선임연구원
- 1994년 2월~1995년 2월 : Univ. of Michigan 객원 연구원
- 1995년 5월 ~ 현재 : 충남대학교 컴퓨터공학과 교수
- 2004년 2월 ~ 2005년 2월 : Univ. of Michigan 초빙 연구원

<관심분야> : 실시간시스템, 운영체제, 고장허용 컴퓨팅, 로봇 미들웨어