



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0092596
(43) 공개일자 2018년08월20일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) H04L 9/30 (2006.01)
H04L 9/32 (2006.01)
(52) CPC특허분류
G06F 21/575 (2013.01)
H04L 9/30 (2013.01)
(21) 출원번호 10-2017-0018490
(22) 출원일자 2017년02월10일
심사청구일자 2017년02월10일

(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
조진성
경기도 수원시 영통구 봉영로 1620, 101동 1801호 (영통동, 대우 월드마크 영통)
김병선
경기도 용인시 기흥구 덕영대로 1732 경희대학교 (서천동)
(뒷면에 계속)
(74) 대리인
특허법인도담

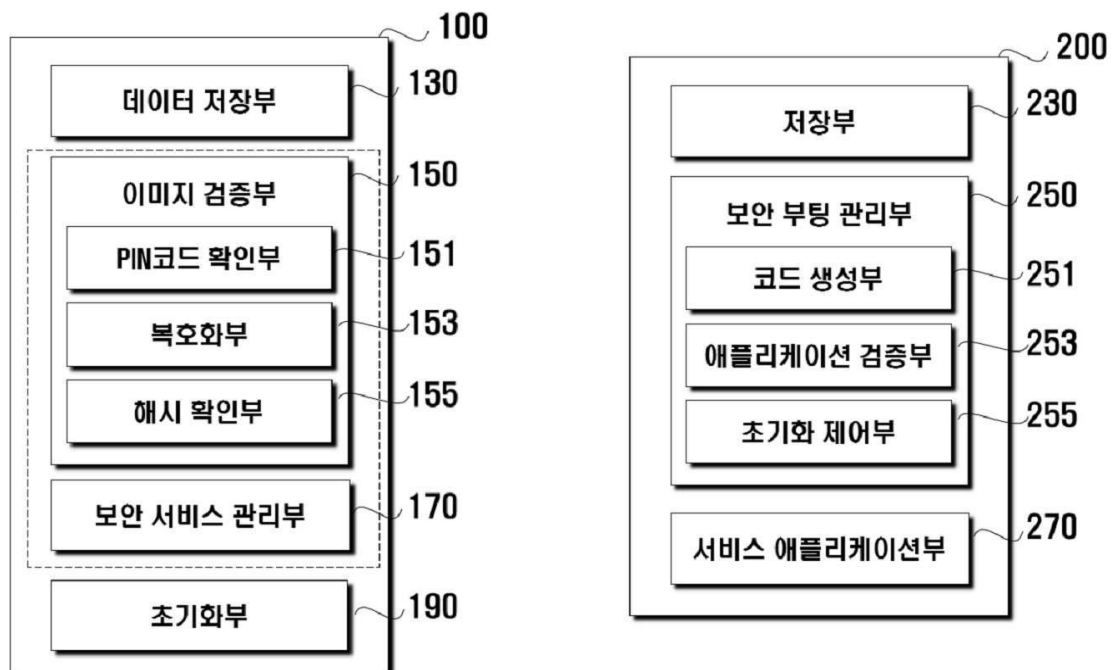
전체 청구항 수 : 총 19 항

(54) 발명의 명칭 임베디드 보안 모듈을 이용한 IoT 디바이스 플랫폼의 보안 부팅 방법 및 시스템

(57) 요약

본 발명은 임베디드 보안 모듈을 이용한 IoT(Internet of Things, 사물 인터넷) 디바이스 플랫폼의 보안 부팅 방법 및 시스템에 관한 것이다. 본 발명은 임베디드 보안 모듈을 이용한 IoT 디바이스 플랫폼의 보안 부팅 방법에 있어서, 상기 보안 모듈의 초기화가 완료되면, 상기 IoT 디바이스의 호스트 프로세서가 애플리케이션 이미지 바 (뒷면에 계속)

대표도 - 도1



이너리의 해시를 추출하는 단계, 상기 호스트 프로세서가 상기 해시를 이용하여 검증 요청을 생성하는 단계, 상기 호스트 프로세서가 상기 검증 요청을 상기 보안 모듈로 전송하면, 상기 보안 모듈이 상기 검증 요청에서 PIN 코드 및 상기 해시를 추출하는 단계, 상기 보안 모듈은 기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 상기 검증 요청에서 추출한 해시의 일치 여부를 이용하여 상기 이미지 바이너리를 검증하는 단계, 상기 보안 모듈은 상기 검증 결과를 이용하여 상기 검증 요청에 대응하는 응답 신호를 생성하고, 상기 응답 신호를 상기 호스트 프로세서에 전송하는 단계를 포함하는 것을 일 특징으로 한다. 본 발명에 의하면, IoT 디바이스 플랫폼 내에서 펌웨어 교체 공격이 이루어지는 경우 중요 데이터와 프라이버시 정보를 더욱 안전하게 보호할 수 있다.

(52) CPC특허분류

H04L 9/3226 (2013.01)

H04L 9/3236 (2013.01)

(72) 발명자

이기웅

경기도 군포시 수리산로 203번길 14 (산본동)

송영철

울산광역시 남구 대공원로141번길 8, 103동 1305호
(옥동, 대공원롯데인벤스가아파트)

윤중철

경기도 화성시 삼성전자로 1-1 DSR Security 제품
개발팀 (반월동)

조성희

경기도 용인시 수지구 성북2로 86, 203동 1104호
(성북동, 성동마을LG빌리지2차아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호 20161655

부처명 -

연구관리전문기관 경희대학교 산학협력단

연구사업명 IOT 디바이스 보안 요소 기술 및 취약점 분석 연구

연구과제명 경희대-삼성전자 반도체 산학협력위원회 연구과제

기 여 율 1/1

주관기관 삼성전자 주식회사

연구기간 2015.09.01 ~ 2020.08.31

명세서

청구범위

청구항 1

임베디드 보안 모듈을 이용한 IoT 디바이스 플랫폼의 보안 부팅 방법에 있어서,

상기 보안 모듈의 초기화가 완료되면, 상기 IoT 디바이스의 호스트 프로세서가 애플리케이션 이미지 바이너리의 해시를 추출하는 단계;

상기 호스트 프로세서가 상기 해시를 이용하여 검증 요청을 생성하는 단계;

상기 호스트 프로세서가 상기 검증 요청을 상기 보안 모듈로 전송하면, 상기 보안 모듈이 상기 검증 요청에서 PIN코드 및 상기 해시를 추출하는 단계;

상기 보안 모듈은 기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 상기 검증 요청에서 추출한 해시의 일치 여부를 이용하여 상기 이미지 바이너리를 검증하는 단계;

상기 보안 모듈은 상기 검증 결과를 이용하여 상기 검증 요청에 대응하는 응답 신호를 생성하고, 상기 응답 신호를 상기 호스트 프로세서에 전송하는 단계를 포함하는 보안 부팅 방법.

청구항 2

제1항에 있어서,

상기 보안 모듈의 초기화는

상기 호스트 프로세서가 상기 보안 모듈에 리셋 신호를 전송하는 단계;

상기 보안 모듈은 상기 리셋 신호를 수신하면 상기 보안 모듈의 운영 체제 및 구성 요소를 초기화하는 단계;

초기화가 완료되면, 상기 보안 모듈은 상기 호스트 프로세서에 ATR(Answer to Reset) 메시지를 전송하는 단계;

상기 호스트 프로세서는 상기 ATR 메시지가 수신되지 않으면 상기 보안 모듈에 리셋 신호를 재전송하고, 상기 ATR 메시지를 수신하면 초기화를 완료하는 단계;

를 포함하는 방법으로 수행되는 보안 부팅 방법.

청구항 3

제1항에 있어서,

상기 검증 요청 생성 단계는

기 설정된 PIN코드 생성 알고리즘을 이용하여 PIN코드를 생성하는 단계;

상기 생성된 PIN코드와 상기 해시를 포함하는 검증 요청을 생성하는 단계를 포함하는 보안 부팅 방법.

청구항 4

제3항에 있어서

상기 PIN코드 생성 알고리즘은 상기 이미지 바이너리에 포함된 것을 특징으로 하는 보안 부팅 방법.

청구항 5

제1항에 있어서,

상기 검증 단계는

기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드가 일치하는지 판단하는 단계;

상기 검증 요청에서 추출한 PIN코드가 상기 기 저장된 PIN코드와 일치하면 기 저장된 검증 해시를 복호화하고, 일치하지 않으면 PIN코드 에러 메시지를 생성하는 단계;

상기 검증 해시와 상기 검증 요청에서 추출한 해시가 일치하는지 판단하는 단계;

상기 검증 요청에서 추출한 해시가 상기 검증 해시와 일치하지 않으면, 해시 에러 메시지를 생성하고, 일치하면 검증 성공 메시지를 생성하는 단계를 포함하는 보안 부팅 방법.

청구항 6

제5항에 있어서,

상기 기 저장된 검증 해시를 복호화하는 단계는

기 설정된 비대칭키 알고리즘에 따라 공개키를 이용하여 상기 보안 모듈에 저장된 디지털 서명을 복호화하는 단계를 포함하는 보안 부팅 방법.

청구항 7

제1항에 있어서,

상기 검증 단계에서 검증 성공 메시지가 생성된 경우 상기 보안 모듈은,

보안 서비스 애플릿을 잠금 상태에서 해제 상태로 전환하는 단계;

상기 보안 서비스 애플릿의 PIN 코드를 상기 기 저장된 PIN코드와 동일하게 설정하는 단계를 더 포함하는 보안 부팅 방법.

청구항 8

제1항에 있어서,

상기 검증 단계에서 PIN코드 에러 메시지 또는 해시 에러 메시지가 생성된 경우,

상기 응답 신호를 수신한 호스트 프로세서는 부트 프로세스를 차단하는 단계를 더 포함하는 보안 부팅 방법.

청구항 9

제1항에 있어서, 상기 검증 요청과 상기 응답 신호는 APDU 프로토콜을 따르는 것을 특징으로 하는 보안 부팅 방법.

청구항 10

임베디드 보안 모듈을 이용하여 IoT 디바이스 플랫폼의 보안 부팅을 수행하는 상기 IoT 디바이스의 호스트 프로세서에 있어서,

기 설정된 PIN코드 생성 알고리즘 및 애플리케이션 이미지 바이너리를 저장하는 저장부;

부트 프로세스를 관리하는 보안 부팅 관리부;

상기 부트 프로세스가 종료되면 PIN코드를 이용하여 상기 보안 모듈과 통신하며, 기 설정된 IoT 서비스를 제공하는 서비스 애플리케이션부를 포함하며,

상기 보안 부팅 관리부는,

상기 PIN코드 생성 알고리즘을 이용하여 PIN코드를 생성하는 코드 생성부;

상기 애플리케이션 이미지 바이너리의 해시를 추출하고, 상기 생성된 PIN코드와 상기 해시를 이용하여 검증 요청을 생성하여 상기 보안 모듈로 전송하며, 상기 검증 요청에 대응하는 응답 신호를 수신하고, 상기 응답 신호에 따라 부트 프로세스를 처리하는 애플리케이션 검증부;

상기 보안 모듈을 초기화하기 위한 리셋 신호를 전송하고, 상기 보안 모듈로부터 ATR(Answer to Reset) 메시지를 수신하는 초기화 제어부를 포함하는 호스트 프로세서.

청구항 11

제10항에 있어서,

상기 초기화 제어부는

상기 보안 모듈로부터 ATR 메시지가 수신되지 않으면, 상기 보안 모듈에 리셋 신호를 재전송하고, 상기 ATR 메시지를 수신하면 초기화를 완료하는 호스트 프로세서.

청구항 12

제10항에 있어서,

상기 PIN코드 생성 알고리즘은 상기 이미지 바이너리에 포함된 것을 특징으로 하는 호스트 프로세서.

청구항 13

제10항에 있어서,

상기 애플리케이션 검증부는

상기 응답 신호에 검증 성공 메시지가 포함되어 있으면, 상기 부트 프로세스를 정상 종료하고, 상기 응답 신호에 PIN코드 에러 메시지 또는 해시 에러 메시지가 포함되어 있으면 부트 프로세스를 차단하는 호스트 프로세서.

청구항 14

제10항에 있어서,

상기 검증 요청과 상기 응답 신호는 APDU 프로토콜을 따르는 것을 특징으로 하는 호스트 프로세서.

청구항 15

IoT 디바이스 플랫폼의 보안 부팅을 지원하는 임베디드 보안 모듈에 있어서,

비대칭방식으로 암호화된 검증 해시를 포함하는 디지털 서명, 상기 디지털 서명의 복호화에 사용되는 공개키를 저장하는 데이터 저장부;

상기 IoT 디바이스의 호스트 프로세서로부터 검증 요청을 수신하면, 상기 검증 요청에서 PIN코드와 해시를 추출하고, 기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 상기 검증 요청에서 추출한 해시의 일치 여부를 이용하여 상기 호스트 프로세서의 애플리케이션 이미지 바이너리를 검증하는 이미지 검증부;

상기 이미지 검증부의 검증 결과에 따라 보안 서비스 애플릿을 잠금 상태에서 해제 상태로 전환하는 보안 서비스 관리부를 포함하는 임베디드 보안 모듈.

청구항 16

제15항에 있어서,

상기 호스트 프로세서로부터 리셋 신호를 수신하면, 운영 체제 및 구성 요소를 초기화하고 상기 호스트 프로세서에 ATR(Answer to Reset) 메시지를 전송하는 초기화부를 더 포함하는 임베디드 보안 모듈.

청구항 17

제15항에 있어서,

상기 이미지 검증부는

기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드가 일치하는지 판단하여, 상기 검증 요청에서 추출한 PIN코드가 상기 기 저장된 PIN코드와 일치하지 않으면 PIN코드 에러 메시지를 생성하는 PIN코드 확인부;

상기 판단 결과 상기 검증 요청에서 추출한 PIN코드가 상기 기 저장된 PIN코드와 일치하면 상기 공개키를 이용하여 상기 디지털 서명을 복호화하여 상기 검증 해시를 추출하는 복호화부;

상기 검증 해시와 상기 검증 요청에서 추출한 해시가 일치하는지 판단하여, 상기 검증 요청에서 추출한 해시가 상기 검증 해시와 일치하지 않으면, 해시 에러 메시지를 생성하고, 일치하면 검증 성공 메시지를 생성하는 해시 확인부를 포함하는 임베디드 보안 모듈.

청구항 18

제15항에 있어서,

상기 보안 서비스 관리부는

상기 이미지 검증부에서 검증 성공 메시지가 생성되면, 상기 보안 서비스 애플릿을 잠금 상태에서 해제 상태로 전환하고, 보안 서비스 애플릿의 PIN 코드를 상기 기 저장된 PIN코드와 동일하게 설정하는 임베디드 보안 모듈.

청구항 19

임베디드 보안 모듈을 이용한 IoT 디바이스 플랫폼의 보안 부팅 시스템에 있어서,

상기 보안 모듈의 초기화가 완료되면, 애플리케이션 이미지 바이너리의 해시를 추출하고, 상기 해시를 이용하여 검증 요청을 생성하여, 상기 검증 요청을 상기 보안 모듈로 전송하는 호스트 프로세서;

상기 검증 요청에서 PIN코드 및 상기 해시를 추출하고, 기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 상기 검증 요청에서 추출한 해시의 일치 여부를 이용하여 상기 이미지 바이너리를 검증하고, 상기 검증 결과를 이용하여 상기 검증 요청에 대응하는 응답 신호를 생성하고, 상기 응답 신호를 상기 호스트 프로세서에 전송하는 보안 모듈을 포함하는 보안 부팅 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 보안 부팅 방법 및 시스템에 관한 것으로, 보다 구체적으로 임베디드 보안 모듈을 이용한 IoT(Internet of Things, 사물 인터넷) 디바이스 플랫폼의 보안 부팅 방법 및 시스템에 관한 것이다.

배경 기술

- [0003] 범용(COTS; Commercial Off-The-Shelf) IoT 디바이스는 사용자 또는 서비스 목적에 따라 다양한 기능을 구현할 수 있는 오픈 소스(Open source)기반의 하드웨어 플랫폼을 의미한다. 현재 IoT 시장에는 라즈베리파이(Raspberry), 아두이노(Arduino), 비글 본 블랙(Beagle Bone Black), 인텔 에디슨(Intel Edison) 등이 대표적인 COTS IoT 디바이스 플랫폼으로 자리잡고 있다. 이러한 디바이스 플랫폼은 가전, 자동차, 스마트시티 등과 같은 다양한 서비스 분야에 폭넓게 활용될 수 있도록 저전력(저성능), 가격 제약, 소형화 특징을 갖는다.
- [0004] 최근 다양한 분야의 IoT 서비스가 확대됨에 따라 IoT 디바이스 수는 지속적으로 증가하고 있으며, 그로 인해 디바이스 보안에 대한 우려도 높아지고 있다. 현재 출시되고 있는 대부분의 IoT 디바이스들은 그 자체에 오픈소스 소프트웨어를 내장하는 방식이어서 보안에 취약하다. 또한, 현존하는 다양한 보안 솔루션들은 고성능 PC 환경을 대상으로 개발된 것이기 때문에 상대적으로 저성능, 저전력 특징을 갖는 IoT 디바이스에 적용하기에는 많은 제약이 있다. 이러한 한계로 인해 IoT 디바이스를 대상으로 한 보안 공격은 점차 다양해지고 있으며, 그 수도 증가하고 있다.
- [0005] IoT 디바이스를 대상으로 한 다양한 공격 기술 중, 펌웨어 교체 공격(Firmware Replacement Attack)은 디바이스 플랫폼 내 플래쉬 메모리에 저장된 펌웨어를 악의적인 펌웨어로 교체/변조하는 공격 방법이다. 펌웨어 교체 공격을 받은 디바이스는 디바이스 부트 단계에서 정상 펌웨어가 아닌 교체/변조된 펌웨어를 실행하게 되며, 이를 통해 공격자는 불법적으로 시스템 제어권을 획득할 수 있다. 공격자는 디바이스 내부에 저장된 중요 데이터(암호화 키)에 접근하고, 사용자의 프라이버시를 침해하거나 타 디바이스와 연동된 서비스에 악영향을 미칠 수 있다.
- [0006] 펌웨어 교체 공격을 방지하기 위하여 사용되고 있는 신뢰 부트(Secure Boot) 기술로는 Root of Trust(RoT) 기술과 단계별 부트 컴포넌트의 무결성 검증을 수행하는 Trust of Chain 기술이 있다.
- [0007] IoT 디바이스는 일반적으로 롬 바이오스(ROM BIOS: ROM Basic Input Output System), 부트로더(Bootloader), 애플리케이션(Application)을 포함하는데, RoT는 롬 바이오스에서 구현되며, Trust of Chain 기술은 부트로더와 같은 신뢰 부트를 실행하는 다음 단계의 컴포넌트에서 사용된다. 신뢰 부트 과정에 대한 자세한 설명은 다음과 같다.
- [0008] 롬 바이오스 RoT - 디바이스에 전원이 입력되면 먼저 롬 바이오스가 구동된다. 디바이스 플랫폼 제조 과정에서 프로그래밍된 롬 바이오스에는 기본적으로 제어권을 넘길 컴포넌트(i.e., 부트로더)의 무결성 검증에 필요한 고유값(i.e., 일반적으로 해쉬)을 포함하고 있다. 롬 바이오스는 이 고유값과 다음 수행될 컴포넌트의 이미지 바이너리로부터 측정된 고유값(i.e., 해쉬)을 비교한다. 두 고유값이 일치한다면 다음 컴포넌트에 제어권을 전달한다. 만약 그렇지 않은 경우, 롬 바이오스는 다음 컴포넌트가 변조되었음을 판단하고 디바이스 부트 진행 과정을 중단한다.
- [0009] 부트로더의 애플리케이션 무결성 검증 (Trust of Chain) - 기본적으로 부트로더는 다음 제어권을 넘겨받을 애플리케이션 이미지 바이너리에 대한 고유값을 포함한다. 이 고유값은 롬 바이오스의 부트로더 검증 과정과 동일하게 다음 애플리케이션 이미지 바이너리의 변조 여부를 확인하기 위해 사용된다. 만약, 부트로더가 소유한 고유값이 다음에 수행될 애플리케이션 이미지 바이너리의 고유값과 일치하지 않는다면 부트로더는 부트 과정을 중단한다. 반대로, 두 값이 일치한다면 제어권을 서비스 애플리케이션에 전달한다.
- [0010] 이처럼 신뢰 디바이스의 근원이 되는 RoT는 Trust of Chain에 대한 신뢰성을 보장하기 위한 필수 요소 기술이다. 하지만, RoT를 수행하는 디바이스는 지정된 부트로더와 애플리케이션만을 사용해야 하기 때문에 디바이스의 확장성을 제한하는 문제가 있다. 이러한 이유로 대부분의 디바이스 플랫폼 제조사들은 롬 바이오스의 RoT 지원 여부를 고려하고 있지 않다. RoT가 지원되지 않는 IoT 디바이스는 부트로더 및 서비스 애플리케이션의 무결성 보장 방법이 별도로 마련되어 있지 않기 때문에 기본적으로 신뢰할 수 없다. 다시 말해, 롬 바이오스의 무결성 검증 과정이 제외된 서비스 애플리케이션은 언제든지 공격자에 의해 교체/변조될 수 있다.
- [0011] 따라서 IoT 디바이스를 대상으로 애플리케이션 이미지 바이너리를 변조/교체하여 악의적 목적으로 활용하는 것을 차단할 수 있는 기술이 필요하다.

발명의 내용

해결하려는 과제

- [0013] 본 발명은 기술한 문제를 해결하기 위한 것으로, IoT 디바이스 플랫폼 내에서 펌웨어 교체 공격이 이루어지는 경우 중요 데이터와 프라이버시 정보를 보호할 수 있는 방법 및 시스템을 제공하는 것을 일 목적으로 한다.
- [0014] 또한, 본 발명은 디바이스의 프로세서와 보안 애플릿 통신을 수행하는 별도의 임베디드 보안 모듈을 이용함으로써, 디바이스의 부트 과정에서 발생할 수 있는 컴포넌트의 변조 사실을 보다 강력하게 검증할 수 있는 보안 부팅 방법 및 시스템을 제공하는 것을 다른 목적으로 한다.
- [0015] 또한, 본 발명은 디바이스 컴포넌트의 종류와 상관없이 롬 바이오스의 무결성을 검증함으로써, 확장성과 보안성이 개선된 보안 부팅 방법 및 시스템을 제공하는 것을 다른 목적으로 한다.

과제의 해결 수단

- [0017] 이러한 목적을 달성하기 위한 본 발명은 임베디드 보안 모듈을 이용한 IoT 디바이스 플랫폼의 보안 부팅 방법에 있어서, 상기 보안 모듈의 초기화가 완료되면, 상기 IoT 디바이스의 호스트 프로세서가 애플리케이션 이미지 바이너리의 해시를 추출하는 단계, 상기 호스트 프로세서가 상기 해시를 이용하여 검증 요청을 생성하는 단계, 상기 호스트 프로세서가 상기 검증 요청을 상기 보안 모듈로 전송하면, 상기 보안 모듈이 상기 검증 요청에서 PIN코드 및 상기 해시를 추출하는 단계, 상기 보안 모듈은 기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 상기 검증 요청에서 추출한 해시의 일치 여부를 이용하여 상기 이미지 바이너리를 검증하는 단계, 상기 보안 모듈은 상기 검증 결과를 이용하여 상기 검증 요청에 대응하는 응답 신호를 생성하고, 상기 응답 신호를 상기 호스트 프로세서에 전송하는 단계를 포함하는 것을 일 특징으로 한다.
- [0018] 또한, 본 발명은 임베디드 보안 모듈을 이용하여 IoT 디바이스 플랫폼의 보안 부팅을 수행하는 상기 IoT 디바이스의 호스트 프로세서에 있어서, 기 설정된 PIN코드 생성 알고리즘 및 애플리케이션 이미지 바이너리를 저장하는 저장부, 부트 프로세스를 관리하는 보안 부팅 관리부, 상기 부트 프로세스가 종료되면 PIN코드를 이용하여 상기 보안 모듈과 통신하며, 기 설정된 IoT 서비스를 제공하는 서비스 애플리케이션부를 포함하며, 상기 보안 부팅 관리부는 상기 PIN코드 생성 알고리즘을 이용하여 PIN코드를 생성하는 코드 생성부, 상기 애플리케이션 이미지 바이너리의 해시를 추출하고, 상기 생성된 PIN코드와 상기 해시를 이용하여 검증 요청을 생성하여 상기 보안 모듈로 전송하며, 상기 검증 요청에 대응하는 응답 신호를 수신하고, 상기 응답 신호에 따라 부트 프로세스를 처리하는 애플리케이션 검증부, 상기 보안 모듈을 초기화하기 위한 리셋 신호를 전송하고, 상기 보안 모듈로부터 ATR(Answer to Reset) 메시지를 수신하는 초기화 제어부를 포함하는 것을 일 특징으로 한다.
- [0019] 또한, 본 발명은 IoT 디바이스 플랫폼의 보안 부팅을 지원하는 임베디드 보안 모듈에 있어서, 비대칭방식으로 암호화된 검증 해시를 포함하는 디지털 서명, 상기 디지털 서명의 복호화에 사용되는 공개키를 저장하는 데이터 저장부, 상기 IoT 디바이스의 호스트 프로세서로부터 검증 요청을 수신하면, 상기 검증 요청에서 PIN코드와 해시를 추출하고, 기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 상기 검증 요청에서 추출한 해시의 일치 여부를 이용하여 상기 호스트 프로세서의 애플리케이션 이미지 바이너리를 검증하는 이미지 검증부, 상기 이미지 검증부의 검증 결과에 따라 보안 서비스 애플릿을 잠금 상태에서 해제 상태로 전환하는 보안 서비스 관리부를 포함하는 것을 일 특징으로 한다.
- [0020] 또한, 본 발명은 임베디드 보안 모듈을 이용한 IoT 디바이스 플랫폼의 보안 부팅 시스템에 있어서, 상기 보안 모듈의 초기화가 완료되면, 애플리케이션 이미지 바이너리의 해시를 추출하고, 상기 해시를 이용하여 검증 요청을 생성하여, 상기 검증 요청을 상기 보안 모듈로 전송하는 호스트 프로세서, 상기 검증 요청에서 PIN코드 및 상기 해시를 추출하고, 기 저장된 PIN코드와 상기 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 상기 검증 요청에서 추출한 해시의 일치 여부를 이용하여 상기 이미지 바이너리를 검증하고, 상기 검증 결과를 이용하여 상기 검증 요청에 대응하는 응답 신호를 생성하고, 상기 응답 신호를 상기 호스트 프로세서에 전송하는 보안 모듈을 포함하는 것을 일 특징으로 한다.

발명의 효과

- [0022] 기술한 바와 같은 본 발명에 의하면, IoT 디바이스 플랫폼 내에서 펌웨어 교체 공격이 이루어지는 경우 중요 테

이터와 프라이버시 정보를 더욱 안전하게 보호할 수 있다.

[0023] 또한, 본 발명은 디바이스의 프로세서와 보안 애플릿 통신을 수행하는 별도의 임베디드 보안 모듈을 이용함으로써, 디바이스의 부트 과정에서 발생할 수 있는 컴포넌트의 변조 사실을 보다 강력하게 검증할 수 있다.

[0024] 또한, 본 발명은 디바이스 컴포넌트의 종류와 상관없이 롬 바이오스의 무결성을 검증함으로써 확장성과 보안성이 개선된 보안 부팅 방법을 제공할 수 있다.

도면의 간단한 설명

- [0026] 도 1은 본 발명의 일 실시 예에 의한 보안 부팅 시스템의 구성을 도시한 도면,
 도 2는 본 발명의 일 실시 예에 의한 보안 부팅 방법을 설명하기 위한 도면,
 도 3은 본 발명의 일 실시 예에 의한 보안 모듈 초기화 방법을 설명하기 위한 도면,
 도 4는 본 발명의 일 실시 예에 의한 검증 요청 생성 방법을 설명하기 위한 도면,
 도 5는 본 발명의 일 실시 예에 의한 이미지 바이너리 검증 방법을 설명하기 위한 도면,
 도 6은 본 발명의 일 실시 예에 의한 검증 성공 후 보안 모듈의 동작을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0027] 전술한 목적, 특징 및 장점은 첨부된 도면을 참조하여 상세하게 후술되며, 이에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 상세한 설명을 생략한다.

[0028] 도면에서 동일한 참조부호는 동일 또는 유사한 구성요소를 가리키는 것으로 사용되며, 명세서 및 특허청구의 범위 기재된 모든 조합은 임의의 방식으로 조합될 수 있다. 그리고 다른 식으로 규정하지 않는 한, 단수에 대한 언급은 하나 이상을 포함할 수 있고, 단수 표현에 대한 언급은 또한 복수 표현을 포함할 수 있음이 이해되어야 한다.

[0029] 본 명세서에서 사용되는 용어는 단지 특정 예시적 실시 예들을 설명할 목적을 가지고 있으며 한정할 의도로 사용되는 것이 아니다. 본 명세서에서 사용된 바와 같은 단수적 표현들은 또한, 해당 문장에서 명확하게 달리 표시하지 않는 한, 복수의 의미를 포함하도록 의도될 수 있다. 용어 "및/또는," "그리고/또는"은 그 관련되어 나열되는 항목들의 모든 조합들 및 어느 하나를 포함한다. 용어 "포함한다", "포함하는", "포함하고 있는", "구비하는", "갖는", "가지고 있는" 등은 내포적 의미를 갖는바, 이에 따라 이러한 용어들은 그 기재된 특징, 정수, 단계, 동작, 요소, 및/또는 컴포넌트를 특정하며, 하나 이상의 다른 특징, 정수, 단계, 동작, 요소, 컴포넌트, 및/또는 이들의 그룹의 존재 혹은 추가를 배제하지 않는다. 본 명세서에서 설명되는 방법의 단계들, 프로세스들, 동작들은, 구체적으로 그 수행 순서가 확정되는 경우가 아니라면, 이들의 수행을 논의된 혹은 예시된 그러한 특정 순서로 반드시 해야 하는 것으로 해석해서는 안 된다. 추가적인 혹은 대안적인 단계들이 사용될 수 있음을 또한 이해해야 한다.

[0030] 또한, 각각의 구성요소는 각각 하드웨어 프로세서로 구현될 수 있고, 위 구성요소들이 통합되어 하나의 하드웨어 프로세서로 구현될 수 있으며, 또는 위 구성요소들이 서로 조합되어 복수 개의 하드웨어 프로세서로 구현될 수도 있다.

[0031] 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시 예를 상세히 설명하기로 한다.

[0032] 도 1은 본 발명의 일 실시 예에 의한 보안 부팅 시스템의 구성을 도시한 도면이다. 도 1을 참조하면, 본 발명의 일 실시 예에 의한 보안 부팅 시스템(10)은 임베디드 보안 모듈(100)과 호스트 프로세서(200)를 포함한다.

[0033] 임베디드 보안 모듈(100)은 각종 보안 기능을 하드웨어적으로 구현한 모듈로, 임베디드 보안 칩(eSE:embedded Secure Element)으로도 불리며, 본 명세서에서는 설명의 편의를 위하여 '보안 모듈(100)'로 명명한다.

[0034] 보안 모듈(100)은 기본적으로 디바이스 인증을 제공하고, 저장된 개인 정보 및 데이터를 보호하기 위한 암호화 키를 생성하며, 생성된 키는 보안 모듈 내부에 안전하게 저장되어 소프트웨어 기반 악성 프로그램으로부터 디바

이스를 보호하는 기능을 제공한다. 또한 보안 모듈(100) 내부에는 JavaCard OS와 같은 별도의 운영체제가 포함되어 있어, 보안 모듈(100)은 기존 시스템과 통합이 용이하고 시스템의 요구사항에 맞춰 보안 모듈(100) 내부에 개별적인 보안 솔루션을 프로그래밍 할 수 있다는 장점이 있다.

- [0035] 보안 모듈(100)은 검증 요청에서 PIN코드 및 해시를 추출하고, 기 저장된 PIN코드와 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 검증 요청에서 추출한 해시의 일치 여부를 이용하여 이미지 바이너리를 검증하고, 검증 결과를 이용하여 검증 요청에 대응하는 응답 신호를 생성하고, 응답 신호를 호스트 프로세서에 전송하는 모듈로, 데이터 저장부(130), 이미지 검증부(150), 보안 서비스 관리부(170), 초기화부(190)를 포함할 수 있다.
- [0036] 보안 모듈(100)은 크게 애플릿 영역, 데이터 영역, 기타 영역으로 구분될 수 있는데, 애플릿 영역은 보안 부팅 기술의 구현을 위한 Boot Component Verifier(BCV) 애플릿과 호스트 프로세서(200)가 제공하는 서비스 애플리케이션의 다양한 보안 서비스를 지원하기 위한 보안 서비스 애플릿(미도시)으로 구분할 수 있다.
- [0037] 보안 서비스 애플릿은 Custom 애플릿으로, 보안 모듈(100)에 전원이 인가되면 자동적으로 잠금 상태를 유지하며, 초기 PIN코드는 되지 않을 수 있다.
- [0038] 도 1의 이미지 검증부(150)와 보안 서비스 관리부(170)는 BCV 애플릿으로 분류할 수 있으며, 도 1의 데이터 저장부(130)는 데이터 영역으로 분류될 수 있다. 보안 모듈(100)의 운영 체제(예-JavaCard OS), JavaCard Stack, APDU Buffer, JavaCard Transient heap, Transaction buffer 등은 기타 영역으로 분류될 수 있다. 이하에서는 도면에 도시된 구성 요소를 중심으로 보안 모듈(100)의 구성을 설명한다.
- [0039] 데이터 저장부(130)는 비대칭방식으로 암호화된 검증 해시를 포함하는 디지털 서명, 디지털 서명의 복호화에 사용되는 공개키를 저장한다. 데이터 저장부(130)는 각종 중요 데이터를 안전하게 보관하기 위한 용도로 사용되며, 데이터 저장부(130)에 저장된 데이터는 전원 입력이 차단되어도 손실되지 않는다.
- [0040] 이미지 검증부(150)는 IoT 디바이스의 호스트 프로세서로부터 검증 요청을 수신하면, 검증 요청에서 PIN코드와 해시를 추출하고, 기 저장된 PIN코드와 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 검증 요청에서 추출한 해시의 일치 여부를 이용하여 호스트 프로세서(200)의 애플리케이션 이미지 바이너리를 검증하며, 보다 구체적으로 PIN코드 확인부(151), 복호화부(153), 해시 확인부(155)를 포함할 수 있다.
- [0041] PIN코드 확인부(151)는 기 저장된 PIN코드와 검증 요청에서 추출한 PIN코드가 일치하는지 판단하여, 검증 요청에서 추출한 PIN코드가 기 저장된 PIN코드와 일치하지 않으면 PIN코드 에러 메시지를 생성할 수 있다. PIN코드 확인부(151)는 운영 체제에 PIN코드 에러 메시지를 반환하며, 운영 체제는 PIN코드 에러 메시지를 이용하여 응답 APDU를 생성할 수 있다.
- [0042] PIN코드는 보안 모듈(100)의 각 애플릿에 접근하기 위한 식별 정보로, 보안 모듈에 기 저장된 PIN코드는 제조 단계에서 개발자에 의해 미리 설정될 수 있다. 보안 모듈(100)의 PIN코드는 보안 모듈(100)이 완전히 재설정(reprogramming)되는 경우를 제외하고는 별도로 추출할 수 없는 정보일 수 있다. 보안 모듈(100)이 재설정되는 경우, 보안 모듈(100)에 저장된 모든 정보는 손실되며, 호스트 프로세서(200)는 재설정 이전의 보안 모듈(100)의 기능을 사용할 수 없도록 구현되는 것이 바람직하다. 이는 보안 모듈(100)이 재설정되면, 호스트 프로세서(200)가 BCV 애플릿과 초기에 잠금 상태로 설정된 보안 서비스 애플릿을 사용할 수 없음을 의미하며, 보안 서비스 애플릿과 연관된 IoT 서비스에 접근할 수 없음을 의미한다. 따라서 본 발명의 일 실시 예에 따르면, 디바이스 내의 애플리케이션 펌웨어가 변조되었을 경우 디바이스는 부팅이 차단될 뿐 아니라 서비스 및 사용자 보안 정보를 저장하고 있는 보안 모듈(100)의 기능을 정상적으로 활용할 수 없어, 기존의 IoT 서비스에 접근할 수 없게 된다.
- [0043] 복호화부(153)는 판단 결과 검증 요청에서 추출한 PIN코드가 기 저장된 PIN코드와 일치하면 공개키를 이용하여 디지털 서명을 복호화하여 검증 해시를 추출한다. 디지털 서명은 공개키와 검증해시를 이용해 생성된 것일 수 있다. 검증해시는 비대칭방식으로 암호화될 수도 있으나, 대칭방식으로 암호화될 수 있으나, 만일 비대칭방식으로 암호화된 경우 복호화부(153)는 데이터 저장부(130)에 저장된 공개키를 이용하여 디지털 서명을 복호화하여 검증 해시를 추출할 수 있다.
- [0044] 해시 확인부(155)는 검증 해시와 검증 요청에서 추출한 해시가 일치하는지 판단하여, 검증 요청에서 추출한 해시가 검증 해시와 일치하지 않으면, 해시 에러 메시지를 생성하고, 일치하면 검증 성공 메시지를 생성한다.
- [0045] 해시 확인부(155)는 저장부(130)에 저장된 검증 해시와 호스트 프로세서(200)에서 추출된 해시를 비교함으로써,

해시의 변조/교체 여부를 확인할 수 있다. 비교 결과에 따라 해시 확인부(155)가 생성(또는 반환)한 해시 에러 메시지 또는 검증 성공 메시지는 응답 신호로 생성되어 호스트 프로세서(200)에 전송된다.

- [0046] 보안 서비스 관리부(170)는 이미지 검증부의 검증 결과에 따라 보안 서비스 애플릿을 잠금 상태에서 해제 상태로 전환할 수 있다. 예를 들어, 이미지 검증부(150)에서 검증 성공 메시지가 생성되면, 보안 서비스 관리부(170)는 보안 서비스 애플릿을 잠금 상태에서 해제 상태로 전환하고 보안 서비스 애플릿의 PIN 코드를 기 저장된 PIN코드와 동일하게 설정할 수 있다. 보안 서비스 애플릿이 해제 상태로 전환하면 호스트 프로세서(200)의 서비스 애플리케이션 요청을 처리할 수 있다.
- [0047] 만약 이미지 검증부(150)의 PIN코드 및 해시 확인 결과 PIN코드가 일치하지 않거나 해시가 일치하지 않으면, 보안 서비스 애플릿은 초기 상태인 잠금 상태를 그대로 유지할 것이다. 잠금 상태의 보안 서비스 애플릿은 호스트 프로세서(200)로부터 어떠한 요청도 수신 및/또는 응답할 수 없다.
- [0048] 초기화부(190)는 호스트 프로세서로부터 리셋 신호를 수신하면, 운영 체제 및 구성 요소를 초기화하고 호스트 프로세서에 ATR(Answer to Reset) 메시지를 전송할 수 있다.
- [0049] 호스트 프로세서(200)는 보안 모듈(100)의 초기화가 완료되면 애플리케이션 이미지 바이너리의 해시를 추출하고, 해시를 이용하여 검증 요청을 생성하여, 검증 요청을 보안 모듈로 전송하며, 저장부(230), 보안 부팅 관리부(250), 서비스 애플리케이션부(270)를 포함할 수 있다.
- [0050] 호스트 프로세서(200)의 저장부(230)는 플래시 메모리 내에 기 설정된 PIN코드 생성 알고리즘 및 애플리케이션 이미지 바이너리를 저장할 수 있다.
- [0051] 보안 부팅 관리부(250)와 서비스 애플리케이션부(270)는 호스트 프로세서(200)의 플래시 메모리 내 애플리케이션 영역(Application section)으로 구분될 수 있다.
- [0052] 보안 부팅 관리부(250)는 부트 프로세스를 관리하며, 코드 생성부(251), 애플리케이션 검증부(253), 초기화 제어부(255)를 포함할 수 있으며, 각 구성 요소는 다음과 같은 기능을 수행할 수 있다.
- [0053] 코드 생성부(251, Service Code Generator)는 PIN코드 생성 알고리즘을 이용하여 PIN코드를 생성할 수 있다. PIN코드는 각 서비스 기능(Service functions)의 실행에 있어서 보안 모듈(100)과의 통신에 사용되는 일종의 패스워드로, PIN코드 생성 알고리즘은 이미지 바이너리에 포함된 것일 수 있으며, 알고리즘의 소스코드는 외부에 노출되지 않도록 보안 개발자에 의해 비공개 형태로 관리되는 것이 바람직하다.
- [0054] 호스트 프로세서(200)는 PIN코드의 코드 값 자체를 저장하지 않고 보안 개발자만 알고 있는 PIN코드 생성 알고리즘으로 저장하고 있다가, 필요한 경우에만 PIN코드 생성 알고리즘을 이용하여 PIN코드를 생성한다. 따라서 호스트 프로세서(200)는 공격자가 플래시 메모리 내의 특정 메모리 번지에 접근하여 PIN코드를 획득하는 것을 차단할 수 있다.
- [0055] 애플리케이션 검증부(253, Application checker)는 애플리케이션 이미지 바이너리의 해시를 추출하고, 생성된 PIN코드와 해시를 이용하여 검증 요청을 생성하여 보안 모듈로 전송하며, 검증 요청에 대응하는 응답 신호를 수신하고, 응답 신호에 따라 부트 프로세스를 처리한다.
- [0056] 애플리케이션 검증부(253)는 보안 모듈(100)의 초기화가 완료되면 플래시 메모리에 기록된 애플리케이션 이미지 바이너리의 해시를 추출하는데, 이 때 해시 알고리즘으로는 SHA256이 사용될 수 있으며, 해시 결과는 32바이트의 크기를 갖도록 추출될 수 있다.
- [0057] 추출된 해시의 변조/교체 여부에 대한 검증은 보안 모듈(100)에서 이루어질 수 있다. 따라서 전술한 바와 같이 애플리케이션 검증부(253)는 추출된 해시를 포함하는 검증 요청을 생성하여 보안 모듈(100)로 전송하고 이에 대해 응답 신호가 수신되면, 이를 이용하여 부트 프로세스를 처리한다.
- [0058] 예를 들어, 애플리케이션 검증부(253)는 응답 신호에 검증 성공 메시지가 포함되어 있으면 부트 프로세스를 정상 종료하고, 응답 신호에 PIN코드 에러 메시지 또는 해시 에러 메시지가 포함되어 있으면 부트 프로세스를 차단할 수 있다.
- [0059] 애플리케이션 검증부(253)가 해시 검증을 위하여 보안 모듈(100)과 통신함에 있어서, 검증 요청과 응답 신호는 APDU(Application Protocol Data Unit) 프로토콜을 따르는 것일 수 있다. APDU 프로토콜은 국제표준에서 규정하고 있는 프로토콜로, 크게 명령 APDU와 응답 APDU로 나눌 수 있다.

[0060] 예를 들어, 검증 요청은 명령 APDU(Command APDU)의 형태로 보안 모듈(100)에 전송될 수 있으며, 응답 신호는 응답 APDU(Response APDU)의 형태로 호스트 프로세서(200)에 수신될 수 있다. 본 발명의 일 실시 예에 따르면 세가지 타입의 응답 APDU가 사용될 수 있다.

[0061] 먼저, 본 발명의 실시 예에 따른 검증 요청은 [표 1]과 같은 구조를 가질 수 있다.

표 1

Command APDU					
Header				Payload	
CLA	INS	P1	P2	Lc	Data Field
0x80	0x13	0x00	0x00	0x24	02 02 03 03 8C 8D 56 4F 87 3F A1 88 AC 07 86 0A 7B D2 F3 BC E0 99 FA 6E C8 75 C0 E0

[0065] 표 1을 참조하면, CLA(Class of instruction)은 APDU의 명령 유형을 의미하며 “0x80”의 16진수 고정 값을 가질 수 있다. INS(Instruction code)는 수행할 명령을 나타내는 코드로 “0x13”의 16진수 고정 값을 가질 수 있다. P1(Instruction Parameter 1) 및 P2(Instruction Parameter 2)는 수행할 명령의 옵션을 선택하는 매개 변수를 의미하며, Lc(Length of command data field)는 데이터 필드의 길이를 나타낸다. 본 영역에는 실제 바이트 길이의 16진수 값이 삽입될 수 있다.

[0066] Data Field는 실제 데이터가 삽입되는 사용자 데이터 영역으로, 본 발명의 일 실시 예에 따른 해시와 PIN코드는 명령 APDU의 사용자 데이터 영역(Payload)에 포함될 수 있다.

[0067] 예를 들어, 데이터 영역에는 4바이트의 PIN코드와 32바이트의 해시가 포함될 수 있다. 위 예시에서, PIN코드는 (02 02 03 03)이며 해시는 (8C 8D 56 4F 87 3F A1 88 AC 07 86 0A 7B D2 F3 BC E0 99 FA 6E C8 75 C0 E0)이다. 위 예시에서 데이터 필드의 길이는 36바이트이므로, Lc 필드에는 36을 16진수로 환산한 0x24가 입력된 것으로 이해할 수 있다.

[0068] 한편, 검증 요청에 대응하여 반환되는 세 가지 타입의 응답 신호는 아래 [표 2] 내지 [표 4]와 같은 구조를 가질 수 있다.

[0069] 본 발명의 일 실시 예에 의하여 정의된 응답 APDU의 SW1(Status Word 1)은 명령 수행 결과를 나타내며 1바이트 크기를 가질 수 있다. SW1에는 Success(“0x90”), Error(“0x6A”)와 Warning(“0x63”)으로 정의된 값이 포함될 수 있다. SW2(Status Word 2) 역시 명령 수행 결과를 나타내며, 오류 타입을 구분하도록 정의될 수 있다. 예를 들어, PIN코드 에러 메시지를 “0x71”, 해시 에러 메시지를 “0x70”로 정의할 수 있으며, 이러한 오류 타입은 응답 APDU의 SW2에 포함될 수 있다.

[0070] 따라서, 상술한 예시에 의하면, [표 2]는 검증 성공 메시지(success)를 포함하는 응답 신호, [표 3]은 해시 에러 메시지(Hash error)를 포함하는 응답 신호, [표 4]는 PIN코드 에러 메시지(PIN code error)를 포함하는 응답 신호에 해당한다.

표 2

Response APDU	
SW1	SW2
0x90	0x00

표 3

Response APDU	
SW1	SW2
0x6A	0x70

표 4

Response APDU	
SW1	SW2
0x6A	0x71

[0075]

[0077]

초기화 제어부(255)는 보안 모듈(100)을 초기화하기 위한 리셋 신호를 전송하고, 보안 모듈로부터 ATR(Answer to Reset) 메시지를 수신할 수 있다. 초기화 제어부(255)는 보안 모듈로부터 ATR 메시지가 수신되지 않으면, 보안 모듈에 리셋 신호를 재전송하고, ATR 메시지를 수신하면 초기화를 완료할 수 있다. 초기화 제어부(255)는 ATR 메시지가 수신되지 않는 경우뿐 아니라, 수신된 ATR 메시지가 정상적인 신호가 아닌 경우에도, 보안 모듈(100)에 리셋 신호를 반복적으로 전송할 수 있다. 초기화가 정상적으로 완료되면, 호스트 프로세서(200)는 플래시 메모리에 저장된 애플리케이션 이미지 바이너리의 해시를 추출 및 검증하기 위한 애플리케이션 검증부(253)가 동작하도록 제어한다.

[0078]

서비스 애플리케이션부(270)는 IoT 서비스를 제공하기 위한 기능들(Security functions)를 포함하고 있으며, 서비스 애플리케이션부(270)는 IoT 서비스 제공과 관련된 사용자 인증, 데이터 암호화 등의 기능을 구현하기 위하여, 부트 프로세스(Boot Process)가 정상적으로 종료되면, 보안 모듈(100)의 보안 서비스 애플릿(Security Service Applet)과 통신을 수행하며 애플리케이션 서비스를 실행할 수 있다.

[0079]

이하에서는 도 2 내지 도 6을 참조하여, 본 발명의 일 실시 예에 의한 보안 부팅 방법을 설명한다. 보안 부팅 방법에 관한 설명에서 전술한 보안 부팅 시스템과 중복되는 세부 실시 예는 생략될 수 있음에 유의한다.

[0080]

도 2를 참조하면, 먼저 호스트 프로세서(200)는 보안 모듈을 초기화할 수 있다(S100). 단계 100의 일 실시 예를 나타낸 도 3을 참조하면, IoT 디바이스 플랫폼에 전원이 인가되면, 호스트 프로세서가 보안 모듈에 리셋 신호를 전송하고 이에 대한 응답을 기다릴 수 있다(S110). 리셋 신호를 수신한 보안 모듈은 보안 모듈의 운영 체제 및 구성 요소를 초기화하고(S130), 초기화가 완료되면 호스트 프로세서에 ATR(Answer to Reset) 메시지를 전송할 수 있다(S150). 호스트 프로세서(200)는 ATR 메시지가 수신되지 않으면 보안 모듈에 리셋 신호를 재전송하고(S110), ATR 메시지를 수신하면 초기화를 완료할 수 있다(S190).

[0081]

보안 모듈의 초기화가 완료되면, 호스트 프로세서(200)는 애플리케이션 이미지 바이너리의 해시를 추출할 수 있다(S200).

[0082]

다음으로 호스트 프로세서(200)는 해시를 이용하여 검증 요청을 생성하여(S300), 보안 모듈에 전송할 수 있다(S390).

[0083]

단계 300에서 호스트 프로세서(200)는 도 4에 도시된 바와 같이 기 설정된 PIN코드 생성 알고리즘을 이용하여 PIN코드를 생성하고(S330), 생성된 PIN코드와 해시를 포함하는 검증 요청을 생성할 수 있다(S350). 이는 공격자가 PIN코드가 저장된 특정 메모리 번지에 접근하여 PIN코드를 획득하는 것을 차단하기 위함이다. PIN코드 생성 알고리즘은 이미지 바이너리에 포함되어 있으므로, 읽기만 가능하며 수정 불가능한 것일 수 있다.

[0084]

호스트 프로세서가 검증 요청을 보안 모듈로 전송하면, 보안 모듈은 검증 요청에서 PIN코드 및 해시를 추출할 수 있다(S400). 그리고 기 저장된 PIN코드와 검증 요청에서 추출한 PIN코드의 일치 여부 및/또는 기 저장된 검증 해시와 검증 요청에서 추출한 해시의 일치 여부를 이용하여 이미지 바이너리를 검증할 수 있다(S500).

[0085]

단계 500에서 보안 모듈은 도 5에 도시된 바와 같이 기 저장된 PIN코드와 검증 요청에서 추출한 PIN코드가 일치하는지 판단하고(S510), 단계 510에서의 판단 결과 검증 요청에서 추출한 PIN코드가 기 저장된 PIN코드와 일치하면 기 저장된 검증 해시를 복호화할 수 있다(S530). 검증 해시의 복호화는 검증 해시가 비대칭 방식으로 암호화된 경우 기 설정된 비대칭키 알고리즘에 따라 공개키를 이용하여 보안 모듈에 저장된 디지털 서명을 복호화하는 방법으로 수행될 수 있다.

[0086]

만약, 검증 요청에서 추출한 PIN코드가 기 저장된 PIN코드와 일치하지 않으면, 보안 모듈은 PIN코드 에러 메시

지를 생성할 수 있다(S575).

- [0087] 단계 530에서 검증 해시가 복호화된 경우, 보안 모듈은 검증 해시와 검증 요청에서 추출한 해시가 일치하는지를 판단할 수 있다(S550). 단계 550의 판단 결과 검증 요청에서 추출한 해시가 검증 해시와 일치하지 않으면, 보안 모듈은 해시 에러 메시지를 생성할 수 있다(S573). 만약 단계 550에서의 판단 결과 검증 해시와 검증 요청에서 추출한 해시가 일치하면, 보안 모듈은 검증 성공 메시지를 생성할 수 있다(S570).
- [0088] 단계 570, 573, 575에서 생성된 검증 결과는 운영 체제로 반환되어 APDU 프로토콜을 따르는 응답 신호 생성에 사용될 수 있다. 즉, 보안 모듈은 검증 결과를 이용하여 검증 요청에 대응하는 응답 신호를 생성하고(S600), 응답 신호를 호스트 프로세서에 전송할 수 있다(S690). 호스트 프로세서(200)는 응답 신호를 수신하면, 수신된 응답 신호에 따라 부트 프로세스를 처리할 수 있다(S700).
- [0089] 한편, 단계 500에서 검증 성공 메시지가 생성된 경우(S570) 보안 모듈은 도 6에 도시된 바와 같이 보안 서비스 애플릿을 잠금 상태에서 해제 상태로 전환하고(S593), 보안 서비스 애플릿의 PIN 코드를 기 저장된 PIN코드와 동일하게 설정할 수 있다(S597). 검증 성공 메시지가 포함된 응답 신호를 수신한 호스트 프로세서(200)는 응답 신호를 분석하여 부트 프로세스를 정상 종료할 것이다(S700). 부트 프로세스를 정상 종료한 호스트 프로세서(200)는 단계 300에서 생성한 PIN코드를 이용하여 보안 모듈의 보안 서비스 애플릿과 통신할 수 있다.
- [0090] 만약 단계 500에서 PIN코드 에러 메시지(S575) 또는 해시 에러 메시지(S575)가 생성된 경우, 위 메시지가 포함된 응답 신호를 수신한 호스트 프로세서(200)는 응답 신호를 분석하여 부트 프로세스를 차단할 것이다. 따라서 애플리케이션 이미지 바이너리가 변조 또는 교체된 경우, IoT 디바이스는 부트 프로세스의 차단을 통해 애플리케이션 이미지에 포함된 서비스 애플리케이션의 실행을 차단할 뿐 아니라, 임베디드 보안 모듈(보안 칩)을 사용할 수 없도록 조치함으로써 외부 공격에 대응하여 중요 정보들을 안전하게 보호할 수 있다.
- [0091] 본 명세서에서 설명되는 방법들은 하나 이상의 프로세서들에 의해 실행되는 하나 이상의 컴퓨터 프로그램들에 의해 구현될 수 있다. 컴퓨터 프로그램들은 비-일시적인 유형의 컴퓨터 판독가능 매체에 저장되는 프로세서-실행가능 명령들을 포함한다. 컴퓨터 프로그램들은 또한 저장된 데이터를 포함할 수 있다. 비-일시적인 유형의 컴퓨터 판독가능 매체(non-transitory tangible computer readable medium)의 비한정적 예들은 비휘발성 메모리 시스템, 자기 저장소 및 광학 저장소이다.
- [0092] 앞서 설명된 기법들의 특정 실시형태들은 알고리즘 형태로 본 명세서에서 설명되는 처리 단계들 및 명령들을 포함한다. 앞서 설명된 처리 단계들 및 명령들은 소프트웨어, 펌웨어, 혹은 하드웨어로 구현될 수 있고, 소프트웨어로 구현되는 경우 실시간 네트워크 오퍼레이팅 시스템(real time network operating system)들에서 사용되는 다른 플랫폼들 상에 상주하도록 다운로드 될 수 있고 이로부터 동작될 수 있음에 유의해야만 한다.
- [0093] 본 발명은 또한 본 명세서에서의 동작들을 수행하기 위한 장치와 관련된다. 이러한 장치는 원하는 목적을 위해 특별히 구성될 수 있거나, 또는 컴퓨터에 의해 액세스될 수 있는 컴퓨터 판독가능 매체 상에 저장된 컴퓨터 프로그램에 의해 선택적으로 활성화되거나 재구성되는 범용 컴퓨터를 포함할 수 있다. 이러한 컴퓨터 프로그램은 유형의 컴퓨터 판독가능 저장 매체에 저장될 수 있는바, 이러한 유형의 컴퓨터 판독가능 저장 매체는, 예를 들어, 플로피 디스크들, 광학 디스크들, CD-ROM들, 자기-광학 디스크들(magnetic-optical disks), 판독-전용 메모리(Read-Only Memory, ROM)들, 랜덤 액세스 메모리(Random Access Memory, RAM)들, EPROM들, EEPROM들, 자기 혹은 광학 카드들, 애플리케이션 특정 집적 회로(ASIC)들을 포함하는 임의 타입의 디스크, 또는 전자 명령들을 저장하기에 적합하고 그 각각이 컴퓨터 시스템 버스에 결합되는 임의 타입의 매체들이 있지만 이러한 것으로만 한정되는 것은 아니다. 더욱이, 본 명세서에서 지칭되는 컴퓨터들은 단일 프로세서를 포함할 수 있거나, 또는 컴퓨팅 능력 증진을 위해 복수의 프로세서 설계를 사용하는 아키텍처들일 수 있다.
- [0094] 본 명세서에 제시되는 알고리즘들 및 동작들은 본질적으로 임의의 특정 컴퓨터 혹은 다른 장치들과 관련되지 않는다. 다양한 범용 시스템들이 또한, 본 명세서에서의 가르침에 따른 프로그램들과 함께 사용될 수 있고, 또는 원하는 방법의 단계들을 수행하기 위해 더 특수하게 설계된 장치들을 구성하는 것이 편리한 것으로 판명될 수 있다. 다양한 이러한 시스템들을 위해 그 요구되는 구조는 그 등가적 변형물들과 함께 본 발명의 기술 분야에서 숙련된 자들에게 명백할 것이다. 추가적으로, 본 개시내용은 임의의 특정 프로그래밍 언어와 관련되어 설명되는 것이 아니다. 다양한 프로그래밍 언어가 본 명세서에서 설명되는 바와 같은 본 개시내용의 가르침들을 구현하기 위해 사용될 수 있고, 특정 언어에 대한 임의의 언급은 본 발명의 실시예 및 최상의 모드를 설명하기 위한 것임을 이해해야 한다.
- [0095] 본 명세서에서 생략된 일부 실시 예는 그 실시 주체가 동일한 경우 동일하게 적용 가능하다. 또한, 전술한 본

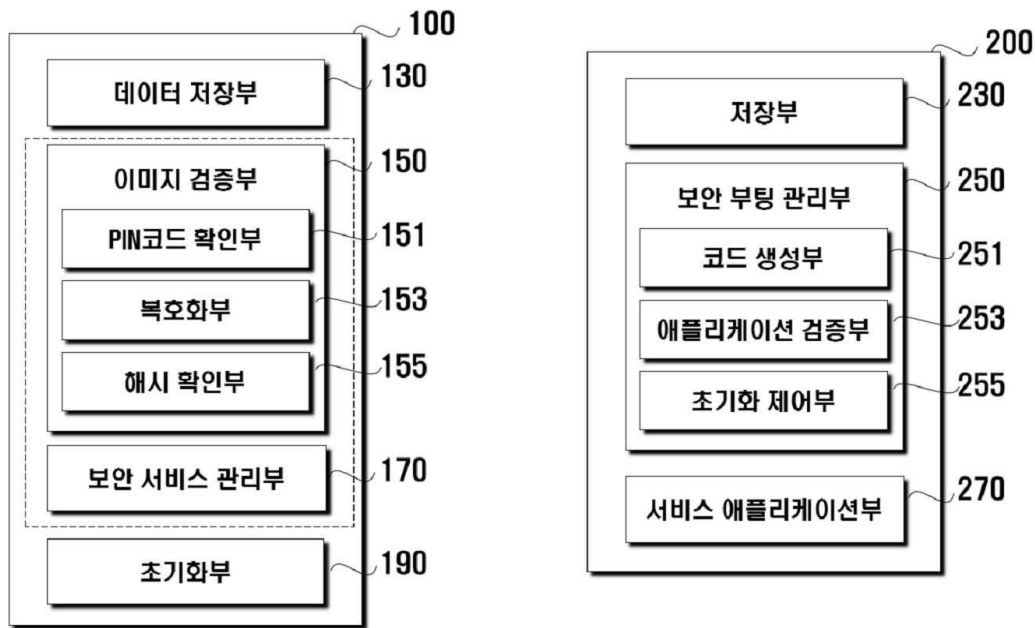
발명은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 있어 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하므로 전술한 실시 예 및 첨부된 도면에 의해 한정되는 것이 아니다.

부호의 설명

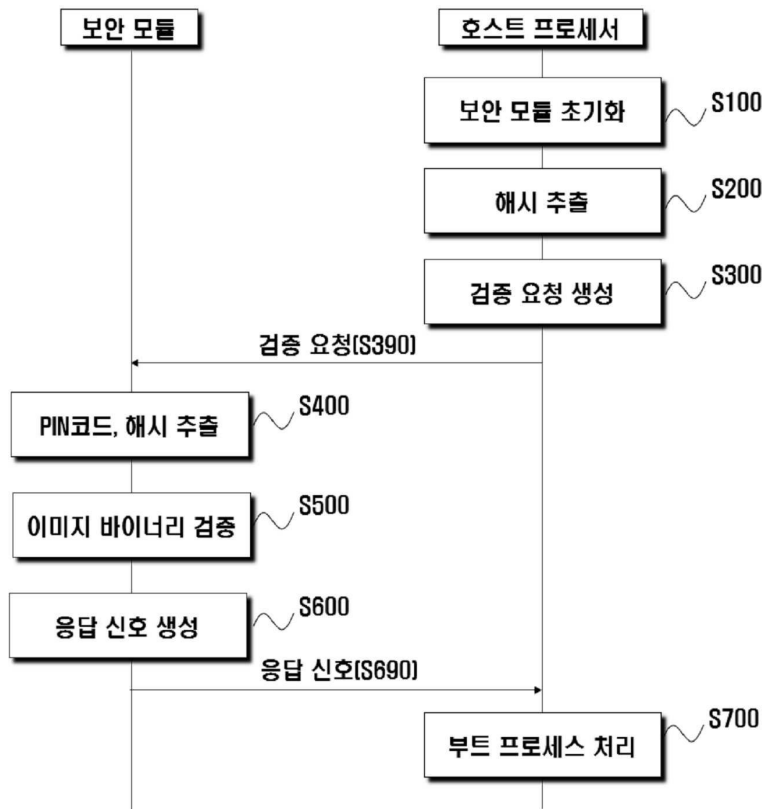
100: 임베디드 보안 모듈(보안 칩)
200: IoT 디바이스의 호스트 프로세서

도면

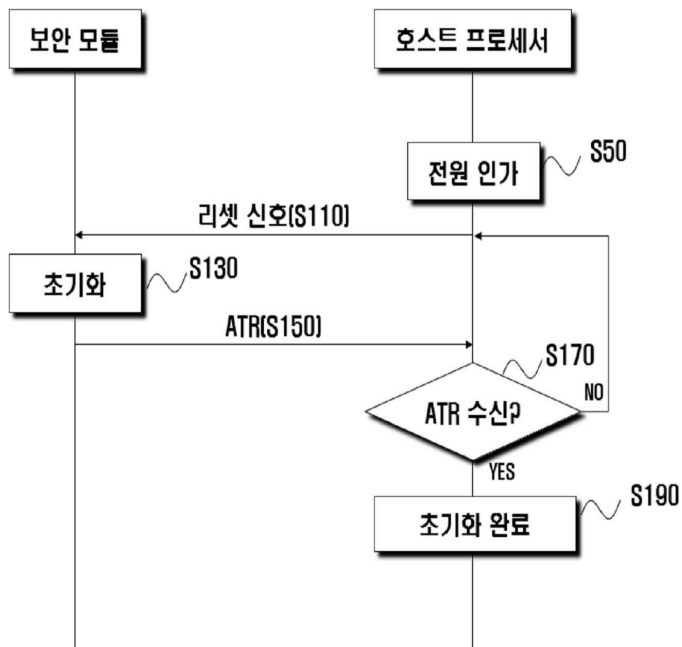
도면1



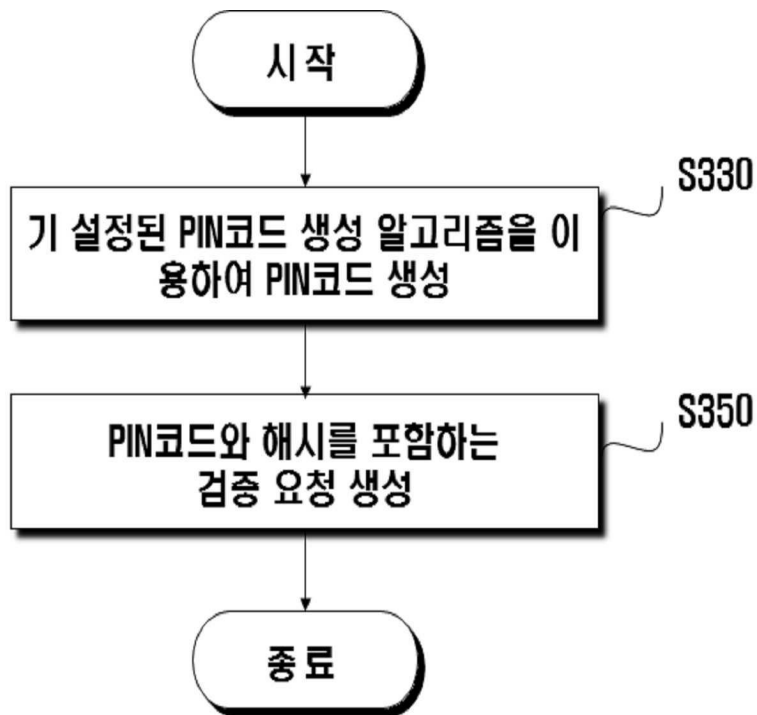
도면2



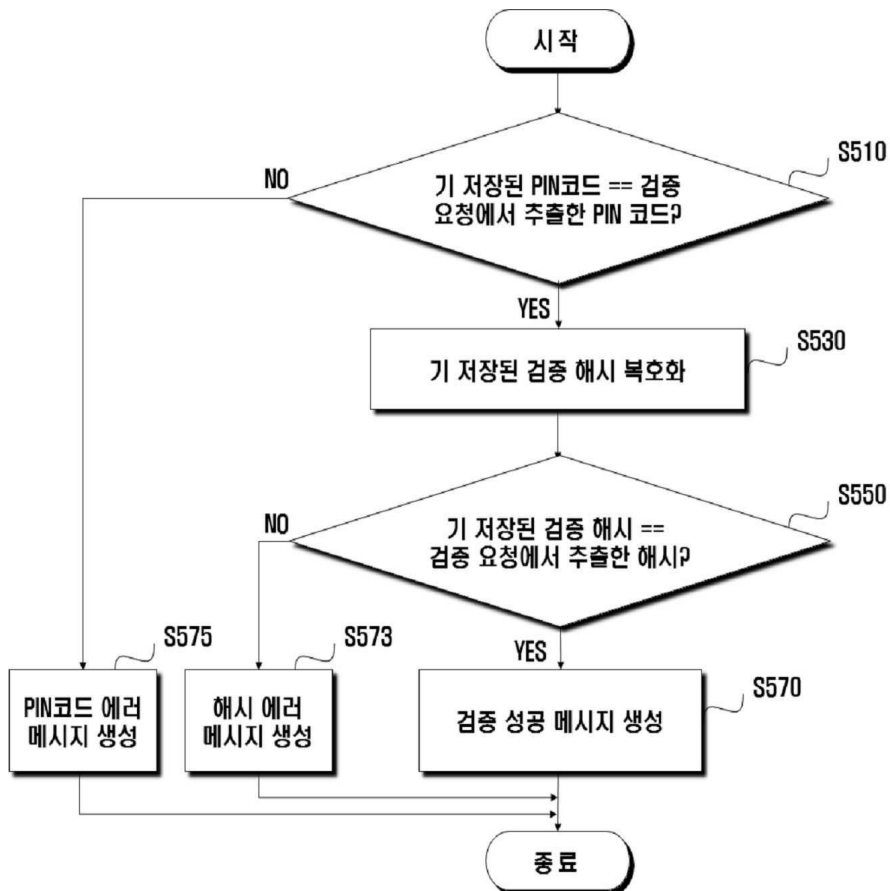
도면3



도면4



도면5



도면6

