



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0020626
(43) 공개일자 2020년02월26일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) G06F 21/60 (2013.01)
H04L 9/08 (2006.01)
(52) CPC특허분류
G06F 21/57 (2013.01)
G06F 21/602 (2013.01)
(21) 출원번호 10-2019-0100041
(22) 출원일자 2019년08월16일
심사청구일자 2019년08월16일
(30) 우선권주장
1020180095297 2018년08월16일 대한민국(KR)

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

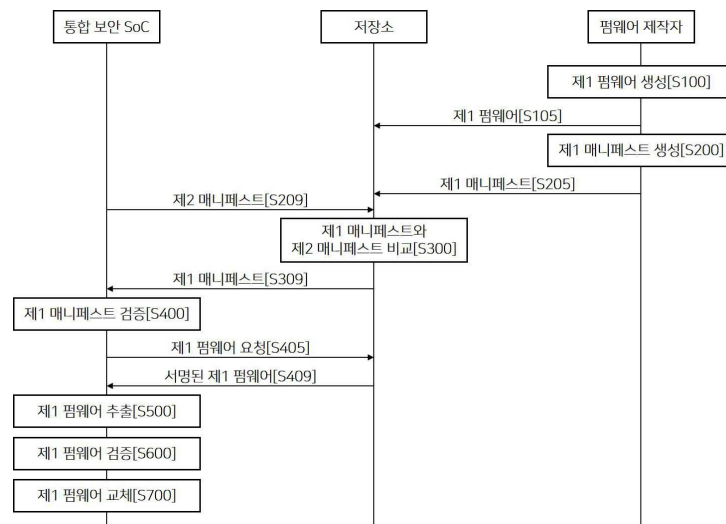
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 통합 보안 SoC를 이용한 IoT 디바이스의 안전한 펌웨어 업데이트 방법

(57) 요약

본 발명은 통합 보안 SoC의 펌웨어 업데이트 방법에 관한 것으로, 통합 보안 SoC 기반의 IoT 디바이스 보안 플랫폼을 활용하여 IoT 디바이스의 펌웨어를 안전하게 업데이트하는 것을 일 목적으로 한다. 이를 달성하기 위한 본 발명은 펌웨어 저장소로부터 제1 매니페스트 파일을 수신하는 단계, 상기 제1 매니페스트를 검증하고, 검증이 완료되면 상기 펌웨어 저장소에 상기 제1 매니페스트에 대응하는 제1 펌웨어의 요청 신호를 전송하는 단계, 상기 펌웨어 저장소로부터 제1 펌웨어의 서명 값을 수신하고, 상기 제1 펌웨어의 서명 값에서 상기 제1 펌웨어를 추출하는 단계, 상기 제1 펌웨어와 상기 제1 매니페스트 파일을 비교하여 상기 제1 펌웨어를 검증하는 단계, 상기 제1 펌웨어가 검증되면, 현재 저장된 제2 펌웨어를 상기 제1 펌웨어로 교체하는 단계를 포함하는 것을 특징으로 한다.

대표도 - 도2



(52) CPC특허분류

H04L 9/0861 (2013.01)

(72) 발명자

이기영

경기도 성남시 중원구 도촌남로 22, 113동 404호
(도촌동, 휴먼시아섬마을1단지아파트)

김병선

경기도 용인시 기흥구 덕영대로 1732, 전자정보대학 348호(서천동)

이 발명을 지원한 국가연구개발사업

과제고유번호 NRF-2017R1D1A1B04035914

부처명 교육부

연구관리전문기관 한국연구재단 (학술진흥)

연구사업명 이공분야기초연구사업/이공학개인지초연구지원사업

연구과제명 IoT 디바이스 신뢰 플랫폼 개발 및 오픈소스 SW로의 공개(3/4)

기 여 율 1/1

주관기관 경희대학교(국제캠퍼스)

연구기간 2019.03.01 ~ 2020.02.28

명세서

청구범위

청구항 1

통합 보안 SoC가 펌웨어를 업데이트하는 방법에 있어서,

펌웨어 저장소로부터 제1 매니페스트 파일을 수신하는 단계;

상기 제1 매니페스트를 검증하고, 검증이 완료되면 상기 펌웨어 저장소에 상기 제1 매니페스트에 대응하는 제1 펌웨어의 요청 신호를 전송하는 단계;

상기 펌웨어 저장소로부터 제1 펌웨어의 서명 값을 수신하고, 상기 제1 펌웨어의 서명 값에서 상기 제1 펌웨어를 추출하는 단계;

상기 제1 펌웨어와 상기 제1 매니페스트 파일을 비교하여 상기 제1 펌웨어를 검증하는 단계;

상기 제1 펌웨어가 검증되면, 현재 저장된 제2 펌웨어를 상기 제1 펌웨어로 교체하는 단계를 포함하는 펌웨어 업데이트 방법.

청구항 2

제1항에 있어서,

상기 통합 보안 SoC는 펌웨어 제작자로부터 생성된 공개키 및 개인키 중 상기 공개키를 수신하는 단계를 더 포함하는 것을 특징으로 하는 펌웨어 업데이트 방법.

청구항 3

제2항에 있어서,

상기 제1 매니페스트는 펌웨어 제작자로부터 생성된 것으로, 상기 제1 펌웨어의 버전 정보, 상기 개인키로 서명된 제1 펌웨어 해시의 서명 값, 또는 상기 개인키로 서명된 제1 매니페스트 해시의 서명 값 중 적어도 하나를 더 포함하는 것을 특징으로 하는 펌웨어 업데이트 방법.

청구항 4

제3항에 있어서,

상기 제1 매니페스트를 검증하는 단계 및 상기 제1 펌웨어의 서명 값에서 상기 제1 펌웨어를 추출하는 단계는 상기 공개키를 이용하는 것을 특징으로 하는 펌웨어 업데이트 방법.

청구항 5

제1항에 있어서,

상기 펌웨어 저장소는 통합 보안 SoC로부터 현재 저장된 제2 매니페스트를 수신하고, 펌웨어 제작자가 새로 생성한 상기 제1 매니페스트와 상기 제2 매니페스트를 비교하여 상기 제1 매니페스트와 상기 제2 매니페스트가 서로 상이하면 보안 SoC에 상기 제1 매니페스트를 전송하는 것을 특징으로 하는 펌웨어 업데이트 방법.

청구항 6

펌웨어를 업데이트하기 위한 통합 보안 SoC에 있어서,

펌웨어 저장소와 제1 펌웨어와 관련된 민감 데이터를 송수신하는 통신부;

상기 제1 펌웨어에 대응하는 제1 매니페스트를 검증하는 제1 검증부;

제1 펌웨어 해시의 서명 값에서 상기 제1 펌웨어를 추출하고, 상기 제1 펌웨어와 상기 제1 매니페스트를 비교하여 상기 제1 펌웨어를 검증하는 제2 검증부;

상기 제1 펌웨어가 검증되면, 현재 저장된 제2 펌웨어를 상기 제1 펌웨어로 교체하는 펌웨어 교체부를 포함하는 통합 보안 SoC.

청구항 7

제6항에 있어서,

상기 통신부는 펌웨어 제작자로부터 생성된 공개키 및 개인키 중 상기 공개키를 더 수신하는 것을 특징으로 하는 통합 보안 SoC.

청구항 8

제7항에 있어서,

상기 제1 매니페스트는 펌웨어 제작자로부터 생성된 것으로 상기 제1 펌웨어의 버전 정보, 상기 개인키로 서명된 제1 펌웨어 해시의 서명 값, 또는 상기 개인키로 서명된 제1 매니페스트 해시의 서명 값 중 적어도 하나를 더 포함하는 것을 특징으로 하는 통합 보안 SoC.

청구항 9

제7항에 있어서,

상기 제1 검증부는 상기 제1 펌웨어 해시의 서명 값을 상기 공개키로 복호화하고,

상기 제2 검증부는 상기 제1 펌웨어의 서명 값을 상기 공개키로 복호화하는 것을 특징으로 하는 통합 보안 SoC.

청구항 10

제6항에 있어서,

상기 펌웨어 저장소는 통합 보안 SoC로부터 현재 저장된 제2 매니페스트를 수신하고, 펌웨어 제작자가 새로 생성한 상기 제1 매니페스트와 상기 제2 매니페스트를 비교하여 상기 제1 매니페스트와 상기 제2 매니페스트가 서로 상이하면 보안 SoC에 상기 제1 매니페스트를 전송하는 것을 특징으로 하는 통합 보안 SoC.

발명의 설명

기술 분야

[0001] 본 발명은 통합 보안 SoC를 이용한 IoT 디바이스의 안전한 펌웨어 업데이트 방법에 관한 것으로, 보다 자세하게는 IoT 디바이스의 펌웨어 바이너리가 변조 또는 교체되어 악의적 목적으로 활용되는 것을 차단하기 위한 통합 보안 SoC를 이용한 IoT 디바이스의 안전한 펌웨어 업데이트 방법에 관한 것이다.

배경 기술

[0002] 기술과 인터넷의 발전으로 다양한 임베디드 플랫폼과 센서의 개발이 진행되었고, 그에 따라 IoT(사물 인터넷, Internet of Things) 사회가 도래하였다. 여러 단체와 기업은 IoT 서비스를 위한 오픈소스 기반의 하드웨어 플

랫폼을 범용 IoT 디바이스로 개발하였다. 대표적인 범용 IoT 디바이스 플랫폼은 라즈베리 파이, 아두이노, 비글 본 블랙, 인텔 에디슨 등이 있으며, 사용자 또는 서비스의 목적에 따라 폭넓게 활용될 수 있도록 저전력, 가격 제약, 소형화의 특징을 갖는다.

[0003] 최근에는 이를 기반으로 다양한 IoT 서비스가 출시되고 있으나 대다수 기업은 IoT 디바이스의 취약점과 그에 따른 보안 기술의 필요성을 인식하지 못하는 경우가 대부분이다. 그로 인해 IoT 디바이스의 보안에 대한 우려가 높아지고 있으나, 성능에 제약이 있는 IoT 디바이스는 이를 적용하는 것에 문제가 있다.

[0004] 이를 해결하기 위하여 국가 표준 단체인 oneM2M에서는 IoT 디바이스를 위한 보안 솔루션으로 하드웨어 보안 모듈을 제시하였으나, 하드웨어 보안 모듈이 단독으로 동작함에 따라 호스트 프로세서와 데이터를 주고받는 과정에서 데이터가 노출될 수 있으며, 또한 호스트 프로세서와 하드웨어 보안 모듈 사이에 악의적인 디바이스를 추가하여 전송된 데이터를 가로채는 하드웨어 트로이 공격에 당할 수 있다는 치명적인 단점이 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 전술한 문제점을 해결하기 위한 것으로서, 통합 보안 SoC 기반의 IoT 디바이스 보안 플랫폼을 활용하여 IoT 디바이스의 펌웨어를 안전하게 업데이트하는 것을 일 목적으로 한다.

[0007] 또한 본 발명은 통합 보안 SoC의 암호화 기능과 안전한 펌웨어 저장소를 사용함으로써 펌웨어 매니페스트와 펌웨어의 변조 여부를 확인하는 것을 일 목적으로 한다.

과제의 해결 수단

[0008] 이러한 목적을 달성하기 위한 본 발명은 통합 보안 SoC가 펌웨어를 업데이트하는 방법에 있어서, 펌웨어 저장소로부터 제1 매니페스트 파일을 수신하는 단계, 상기 제1 매니페스트를 검증하고, 검증이 완료되면 상기 펌웨어 저장소에 상기 제1 매니페스트에 대응하는 제1 펌웨어의 요청 신호를 전송하는 단계, 상기 펌웨어 저장소로부터 제1 펌웨어의 서명 값을 수신하고, 상기 제1 펌웨어의 서명 값에서 상기 제1 펌웨어를 추출하는 단계, 상기 제1 펌웨어와 상기 제1 매니페스트 파일을 비교하여 상기 제1 펌웨어를 검증하는 단계, 상기 제1 펌웨어가 검증되면, 현재 저장된 제2 펌웨어를 상기 제1 펌웨어로 교체하는 단계를 포함하는 것을 일 특징으로 한다.

[0009] 또는 상기 통합 보안 SoC는 펌웨어 제작자로부터 생성된 공개키 및 개인키 중 상기 공개키를 수신하는 단계를 더 포함하는 것을 일 특징으로 한다.

[0010] 나아가 상기 제1 매니페스트는 펌웨어 제작자로부터 생성된 것으로, 상기 제1 펌웨어의 버전 정보, 상기 개인키로 서명된 제1 펌웨어 해시의 서명 값, 또는 상기 개인키로 서명된 제1 매니페스트 해시의 서명 값 중 적어도 하나를 더 포함하는 것을 일 특징으로 한다.

[0011] 또한 상기 제1 매니페스트를 검증하는 단계 및 상기 제1 펌웨어의 서명 값에서 상기 제1 펌웨어를 추출하는 단계는 상기 공개키를 이용하는 것을 일 특징으로 한다.

[0012] 나아가 상기 펌웨어 저장소는 통합 보안 SoC로부터 현재 저장된 제2 매니페스트를 수신하고, 펌웨어 제작자가 새로 생성한 상기 제1 매니페스트와 상기 제2 매니페스트를 비교하여 상기 제1 매니페스트와 상기 제2 매니페스트가 서로 상이하면 보안 SoC에 상기 제1 매니페스트를 전송하는 것을 일 특징으로 한다.

[0013] 또한 본 발명은 펌웨어를 업데이트하기 위한 통합 보안 SoC에 있어서, 펌웨어 저장소와 제1 펌웨어와 관련된 민감 데이터를 송수신하는 통신부, 상기 제1 펌웨어에 대응하는 제1 매니페스트를 검증하는 제1 검증부, 제1 펌웨어 해시의 서명 값에서 상기 제1 펌웨어를 추출하고, 상기 제1 펌웨어와 상기 제1 매니페스트를 비교하여 상기 제1 펌웨어를 검증하는 제2 검증부, 상기 제1 펌웨어가 검증되면, 현재 저장된 제2 펌웨어를 상기 제1 펌웨어로 교체하는 펌웨어 교체부를 포함하는 것을 일 특징으로 한다.

[0014] 나아가 상기 통신부는 펌웨어 제작자로부터 생성된 공개키 및 개인키 중 상기 공개키를 더 수신하는 것을 일 특징으로 한다.

[0015] 또한 상기 제1 매니페스트는 펌웨어 제작자로부터 생성된 것으로 상기 제1 펌웨어의 버전 정보, 상기 개인키로 서명된 제1 펌웨어 해시의 서명 값, 또는 상기 개인키로 서명된 제1 매니페스트 해시의 서명 값 중 적어도 하나

를 더 포함하는 것을 일 특징으로 한다.

[0016] 나아가 상기 제1 검증부는 상기 제1 펌웨어 해시의 서명 값을 상기 공개키로 복호화하고, 상기 제2 검증부는 상기 제1 펌웨어의 서명 값을 상기 공개키로 복호화하는 것을 일 특징으로 한다.

[0017] 또한 상기 펌웨어 저장소는 통합 보안 SoC로부터 현재 저장된 제2 매니페스트를 수신하고, 펌웨어 제작자가 새로 생성한 상기 제1 매니페스트와 상기 제2 매니페스트를 비교하여 상기 제1 매니페스트와 상기 제2 매니페스트가 서로 상이하면 보안 SoC에 상기 제1 매니페스트를 전송하는 것을 일 특징으로 한다.

발명의 효과

[0018] 진술한 바와 같은 본 발명에 의하면, 통합 보안 SoC 기반의 IoT 디바이스 보안 플랫폼을 활용하여 IoT 디바이스의 펌웨어를 안전하게 업데이트할 수 있다.

[0019] 또한 본 발명은 통합 보안 SoC의 암호화 기능과 안전한 펌웨어 저장소를 사용함으로써 펌웨어 매니페스트와 펌웨어의 변조 여부를 확인할 수 있다.

도면의 간단한 설명

[0020] 도 1은 본 발명의 일 실시 예에 의한 디바이스 제조 시 통합 보안 SoC에 민감 데이터를 주입하는 단계를 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시 예에 의한 통합 보안 SoC의 펌웨어 업데이트 방법을 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시 예에 의한 통합 보안 SoC에 저장되는 데이터를 설명하기 위한 도면이다.

도 4는 펌웨어 업데이터 기반 펌웨어 교체 공격이 발생한 경우에 펌웨어 업데이트를 중단하는 과정을 나타낸 순서도이다.

도 5는 펌웨어 롤백 공격이 발생한 경우에 펌웨어 업데이트를 중단하는 과정을 나타낸 순서도이다.

도 6은 본 발명의 일 실시 예에 의한 통합 보안 SoC의 구성을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0021] 진술한 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.

[0022] 도면에서 동일한 참조부호는 동일 또는 유사한 구성요소를 가리키는 것으로 사용되며, 명세서 및 특허청구의 범위 기재된 모든 조항은 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.

[0023] 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 갖는 바, 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.

[0024] 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.

- [0025] 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시 예를 상세히 설명하기로 한다.
- [0027] 일반적으로 범용 IoT 디바이스는, 부트로더가 주변 방치를 통해 새로운 펌웨어를 받기 위한 수신 대기 상태가 되어 새로운 펌웨어를 수신함으로써 펌웨어를 업데이트한다. 이때 부트로더는 플래시 메모리에 저장된 이전 펌웨어 파일을 삭제하고 새로운 펌웨어를 저장하며, 시스템 메모리에 새로운 펌웨어를 로드한다. 이러한 펌웨어 업데이트 과정은 펌웨어 업데이트를 하는 주체와 펌웨어 파일에 대한 검증을 할 수 없기 때문에 공격자가 펌웨어를 손쉽게 변경하여 시스템 제어권을 획득할 수 있게 된다. 또한 취약점이 존재하는 이전 버전의 펌웨어를 사용하면 공격자는 취약점을 공격하여 서비스 운영에 악영향을 미칠 수 있다.
- [0028] 이를 해결하기 위한 본 발명의 통합 보안 SoC는 IoT 디바이스의 부트로더 및 펌웨어 바이너리가 변조 또는 교체되어 악의적 목적으로 활용되는 것을 차단할 수 있다. 통합 보안 SoC는 물리적 복제 방지 기술(PUF, Physical Unclonable Function)을 통해 디바이스 고유 식별키를 제공함으로써 IoT 디바이스의 불법 복제와 수정을 차단할 수 있으며, IoT 디바이스 제조 과정을 제외한 전 과정에서 접근 불가능한 암호화 펌웨어 저장소에 민감 데이터를 저장하여 안전하게 보호할 수 있고 또한 하드웨어로 구현된 대칭/비대칭 보안 가속기를 사용하여 암호화키를 생성할 수 있다.
- [0029] 먼저 도 1을 이용하여 디바이스 제조 시 통합 보안 SoC에 민감 데이터인 암호화 키, 매니페스트 파일, 펌웨어를 전송하는 방법을 설명한다. 매니페스트 파일은 컴퓨팅에서 집합의 일부 또는 논리 정연한 단위의 파일들의 그룹을 위한 메타데이터를 포함한 파일로, 본 발명의 매니페스트 파일은 펌웨어의 버전 정보, 펌웨어 해시의 서명 값, 매니페스트 해시의 서명 값을 포함할 수 있다.
- [0030] 펌웨어 제작자는 통합 보안 SoC에 직접적으로 데이터 주입을 할 수 없기 때문에 제조사를 통하여 펌웨어 관련 데이터를 통합 보안 SoC에 전송할 수 있다. 먼저 IoT 디바이스를 제조하는 제조사는 IoT 디바이스가 제조될 때마다 고유 식별 키를 생성하여(S10) 통합 보안 SoC에 전송할 수 있다(S15). 이후 펌웨어 제작자는 한 쌍의 비대칭키인 공개키와 개인키를 생성(S20)하여 그 중 공개키를 제조사에 전송(S25)하면, 펌웨어 제작자로부터 공개키를 수신한 제조사는 통합 보안 SoC에 공개키를 전송(S29)할 수 있다. 펌웨어 제작자는 IoT 디바이스에 펌웨어를 전송하기 위하여 먼저 매니페스트 파일을 생성(S30)하여 제조사에 전송(S35)하면, 펌웨어 제작자로부터 매니페스트 파일을 수신한 제조사는 통합 보안 SoC에 매니페스트 파일을 전송(S39)할 수 있다. 펌웨어 제작자는 최종적으로 펌웨어를 생성(S40)하여 제조사에 전송(S45)하면 제조사는 통합 보안 SoC에 펌웨어를 전송(S49)할 수 있다.
- [0031] 이하에서는 도 2를 참조하여 IoT 디바이스에 설치된 펌웨어를 업데이트하는 방법을 설명한다. 도 2를 참조하면 통합 보안 SoC의 펌웨어 업데이트는 통합 보안 SoC, 펌웨어 저장소, 펌웨어 제작자가 참여할 수 있다. 펌웨어 제작자는 새로운 펌웨어인 제1 펌웨어를 생성할 수 있다(S100).
- [0032] 펌웨어 제작자는 제1 펌웨어를 생성하면 제1 펌웨어의 서명 값을 펌웨어 저장소에 전송(S105)할 수 있다. 펌웨어 저장소는 펌웨어 제작사로부터 수신한 제1 펌웨어를 저장할 수 있다. 보다 구체적으로 펌웨어 제작자는 제1 펌웨어를 먼저 생성한 개인키로 암호화하여 서명 값을 생성할 수 있다. 본 발명에서 사용되는 전자 서명은 인증서 형태로 발급되는 자신만의 디지털 인감으로 어떤 사람이 데이터를 생성했다는 것을 증명하는 방법이다. 전자 서명은 주로 서명하고자 하는 데이터의 해시 값을 개인키로 암호화하여 상대방에 전송하고, 상대방은 공개키를 이용하여 서명 값을 복호화하여 해시 값을 획득하는 과정을 통해 데이터의 무결성을 검증할 수 있다.
- [0033] 제1 펌웨어의 생성을 완료한 펌웨어 제작자는 제1 펌웨어에 대응하는 제1 매니페스트 파일을 생성할 수 있다(S200). 제1 매니페스트 파일은 제1 펌웨어의 버전 정보, 제1 펌웨어 해시의 서명 값, 제1 매니페스트 해시의 서명 값을 포함할 수 있다. 펌웨어 제작자는 제1 펌웨어와 제1 매니페스트의 서명 값을 생성하기 위하여 먼저, 제1 펌웨어와 제1 매니페스트에 SHA 알고리즘을 적용하여 제1 펌웨어와 제1 매니페스트의 해시 값을 추출할 수 있다. SHA 알고리즘은 임의의 길이의 데이터를 입력 값으로 하여 고정된 길이의 해시 값으로 출력하는 것으로, 데이터의 오류나 변조를 탐지할 수 있는 무결성을 제공할 수 있다. SHA 알고리즘은 입력 값의 최대 길이를 256 바이트로 제한하고 있으나, 복수 개의 명령을 연결하여 실행함으로써 256 바이트보다 큰 데이터를 입력 값으로 사용할 수 있다.
- [0034] 펌웨어 제작자는 제1 펌웨어와 제1 매니페스트의 해시 값을 추출하면, 제1 펌웨어의 해시 값과 제1 매니페스트의 해시 값을 개인키로 각각 암호화하여 서명 값을 생성할 수 있다. 펌웨어 제작자는 생성된 제1 펌웨어 해시의 서명 값과 제1 매니페스트 해시의 서명 값을 이용하여 제1 매니페스트를 생성할 수 있다.

- [0035] 펌웨어 제작자는 제1 매니페스트 파일을 생성하면 제1 매니페스트 파일을 펌웨어 저장소에 전송(S205)할 수 있다. 펌웨어 저장소는 펌웨어 제작사로부터 수신한 제1 매니페스트 파일을 저장할 수 있다.
- [0036] 통합 보안 SoC는 펌웨어 업데이트를 수행하기 위하여 펌웨어 저장소에 현재 사용 중인 제2 펌웨어에 대응하는 제2 매니페스트 파일을 전송할 수 있다(S209). 펌웨어 저장소는 통합 보안 SoC로부터 수신한 제2 매니페스트 파일과 펌웨어 제작사로부터 수신하여 저장한 제1 매니페스트 파일을 비교할 수 있다(S300). 펌웨어 저장소는 제1 매니페스트 파일과 제2 매니페스트 파일이 동일하지 않다고 판단하면, 제1 매니페스트 파일과 제2 매니페스트 파일 중 최신의 매니페스트 파일을 통합 보안 SoC에 전송할 수 있다. 펌웨어 저장소는 제1 매니페스트 파일과 제2 매니페스트 파일이 서로 상이할 경우, 최신의 매니페스트 파일을 식별하기 위하여 매니페스트 파일에 저장된 펌웨어의 버전 정보를 이용할 수 있으며, 매니페스트 파일의 생성 시간 등을 더 이용할 수도 있다. 일반적으로 펌웨어 저장소에 저장된 제1 매니페스트 파일은 통합 보안 SoC의 제2 매니페스트 파일보다 최신의 것이거나 동일한 것으로, 제1 매니페스트 파일과 제2 매니페스트 파일이 동일하지 않은 경우 펌웨어 저장소는 통합 보안 SoC에 제1 매니페스트 파일을 전송할 수 있다(S309).
- [0037] 통합 보안 SoC는 펌웨어 저장소로부터 수신한 제1 매니페스트 파일을 검증할 수 있다(S400). 통합 보안 SoC는 제1 매니페스트 파일에 포함된 제1 매니페스트 해시의 서명 값을 이용하여 제1 매니페스트 파일을 검증할 수 있다. 통합 보안 SoC는 제1 매니페스트 해시의 서명 값을 사전에 수신한 펌웨어 제작자의 공개키를 이용하여 복호화하여 제1 매니페스트의 해시 값을 추출할 수 있다. 통합 보안 SoC는 제1 매니페스트 파일에 포함된 제1 매니페스트에 SHA 알고리즘을 적용하여 해시 값을 생성하고, 생성된 해시 값과 제1 매니페스트 해시의 서명 값에서 추출한 제1 매니페스트의 해시 값을 비교할 수 있다. 이 때 두 해시 값이 동일하면 통합 보안 SoC는 제1 매니페스트 파일이 무결하다고 판단할 수 있다.
- [0038] 통합 보안 SoC는 제1 매니페스트 파일의 검증에 실패하면 펌웨어 업데이트를 중단하고, 그렇지 않으면 펌웨어 저장소에 제1 펌웨어에 대한 요청 신호를 전송할 수 있다(S405).
- [0039] 통합 보안 SoC로부터 제1 펌웨어에 대한 요청 신호를 수신한 펌웨어 저장소는 저장된 제1 펌웨어의 서명 값을 통합 보안 SoC에 전송할 수 있다(S409).
- [0040] 통합 보안 SoC는 수신한 제1 펌웨어의 서명 값에서 제1 펌웨어를 추출할 수 있다(S500). 통합 보안 SoC는 제1 펌웨어의 서명 값을 사전에 수신한 펌웨어 제작자의 공개키를 이용하여 복호화 함으로써 제1 펌웨어를 추출할 수 있다.
- [0041] 통합 보안 SoC는 추출된 제1 펌웨어를 검증할 수 있다(S600). 보다 구체적으로 설명하면 통합 보안 SoC는 추출된 제1 펌웨어와 제1 매니페스트 파일을 비교함으로써 제1 펌웨어를 검증할 수 있다. 통합 보안 SoC는 제1 매니페스트 파일에 포함된 제1 펌웨어의 버전 정보 및 제1 펌웨어 해시의 서명 값과 제1 펌웨어의 서명 값에서 추출된 제1 펌웨어를 비교할 수 있다. 통합 보안 SoC는 제1 펌웨어의 서명 값에서 추출한 제1 펌웨어의 해시 값과 제1 매니페스트 파일에 포함된 제1 펌웨어 해시의 서명 값에서 추출된 제1 펌웨어 해시 값을 서로 비교하여 제1 펌웨어를 검증할 수 있다. 통합 보안 SoC는 버전 정보와 펌웨어가 일치하면 제1 펌웨어가 정상이라고 판단하고 제1 펌웨어를 임시 저장할 수 있다. 통합 보안 SoC는 제1 펌웨어의 검증 과정을 통해 공격자가 펌웨어 저장소를 탈취하여 펌웨어를 변조한 경우를 식별하여 펌웨어 업데이트를 중단할 수 있다.
- [0042] 예를 들어 설명하면, 도 4는 펌웨어 업데이터 기반 펌웨어 교체 공격이 발생한 경우에 펌웨어 업데이트를 중단하는 과정을 나타낸 순서도이고, 도 5는 펌웨어 롤백 공격이 발생한 경우에 펌웨어 업데이트를 중단하는 과정을 나타낸 순서도이다. 도 4를 참조하면, 공격자가 펌웨어 저장소를 탈취하여 펌웨어를 변조시킨 경우 통합 보안 SoC는 펌웨어 저장소로부터 수신한 펌웨어의 서명 값을 이용한 펌웨어의 검증에 실패하여 펌웨어 업데이트를 중단할 수 있다. 도 5를 참조하면, 공격자가 이전 버전의 펌웨어를 이용하여 업데이트를 수행하고자 하면 통합 보안 SoC는 이전 버전의 펌웨어의 서명 값에서 추출한 펌웨어의 버전 정보에 대한 비교 검증에 실패하여 펌웨어 업데이트를 중단할 수 있다.
- [0043] 통합 보안 SoC는 제1 펌웨어에 대한 검증이 완료되면, 제1 매니페스트 파일에 포함된 제1 펌웨어의 버전 정보와 제1 펌웨어의 서명 값을 메모리에 저장할 수 있다. 이때 통합 보안 SoC는 제1 펌웨어의 버전 정보를 IoT 디바이스의 고유 식별키를 이용하여 암호화하여 저장할 수 있다. 이 때 사용되는 암호화 알고리즘은 AES-CBC128일 수 있다.
- [0044] 통합 보안 SoC의 메모리의 구성은 도 3을 참조하면, 플래시 메모리(Flash Memory)와 eFuse로 구분될 수 있다. 플래시 메모리는 시스템 부트 컴포넌트가 저장된 Public Section과 IoT 디바이스 인증 및 암호화에 사용되는 데

이터가 저장된 Private Section로 더 구분될 수 있다. Public Section에는 IoT 서비스 어플리케이션 수행을 위한 펌웨어가 저장되어 있으며, Private Section은 펌웨어 제작자의 공개키, 펌웨어 해시의 서명 값, 암호화된 펌웨어의 버전 정보가 저장되어 있다. eFuse는 OTP(One-Time Programmable)의 특징을 갖는 비휘발성 메모리로, IoT 디바이스의 제조 단계에서 IoT 디바이스의 식별 ID, 플랫폼 키, 디바이스 고유 식별키, 사용자 정의 키가 주입되어 저장될 수 있다. eFuse에 저장된 디바이스 고유 식별키는 AES 하드웨어 모듈을 통한 AES-CBC(Cipher Block Chaining) 128 알고리즘의 대칭 키로 사용되며, 데이터의 각 블록을 암호화 또는 복호화할 때 사용 가능하다.

[0045] 통합 보안 SoC는 상기 과정이 모두 완료되면 IoT 디바이스의 재부팅을 진행하고 현재 사용 중인 펌웨어를 제1 펌웨어로 교체할 수 있다(S700).

[0046] 이하에서는 도 6를 이용하여 본 발명의 일 실시 예에 의한 통합 보안 SoC의 구성을 설명한다. 통합 보안 SoC의 구성에 관한 설명에 있어서 전술한 통합 보안 SoC의 펌웨어 업데이트 방법과 중복되는 세부 실시 예는 생략될 수 있다. 도 6을 참조하면, 통합 보안 SoC는 통신부(100)와 제어부(200)를 포함할 수 있고, 나아가 제어부(200)는 제1 검증부(210), 제2 검증부(230), 그리고 펌웨어 교체부(250)를 포함할 수 있다.

[0047] 통신부(100)는 펌웨어 저장소와 제1 펌웨어와 관련된 민감 데이터를 송수신할 수 있다. 보다 구체적으로 통신부(100)는 펌웨어를 업데이트할 때 현재 사용 중인 제2 펌웨어에 대한 제2 매니페스트 파일을 펌웨어 저장소에 전송할 수 있으며, 펌웨어 저장소로부터 제1 매니페스트 파일을 수신할 수 있다. 이때 펌웨어 저장소는 저장된 가장 최근의 제1 매니페스트 파일과 통합 보안 SoC로부터 수신한 제2 매니페스트 파일을 비교하여 서로 상이하다고 판단하면 통합 보안 SoC에 제1 매니페스트 파일을 전송할 수 있다. 또한 통신부(100)는 통합 보안 SoC가 제1 매니페스트 파일에 대한 검증을 완료하면 펌웨어 저장소에 제1 매니페스트 파일에 대응하는 제1 펌웨어에 대한 요청 신호를 전송하고, 펌웨어 저장소로부터 제1 펌웨어의 서명 값을 수신할 수 있다.

[0048] 제1 검증부(210)는 통신부(100)에서 수신한 제1 매니페스트 파일을 검증할 수 있다. 제1 검증부(210)는 제1 매니페스트 파일에 포함된 제1 매니페스트 해시의 서명 값을 이용하여 제1 매니페스트 파일을 검증할 수 있는데, 보다 구체적으로 제1 검증부(210)는 제1 매니페스트 해시의 서명 값을 사전에 수신한 펌웨어 제작자의 공개키를 이용하여 복호화함으로써 제1 매니페스트의 해시를 추출할 수 있다. 이 후 제1 검증부(210)는 제1 매니페스트에 SHA 알고리즘을 적용하여 해시 값을 생성하고, 생성된 해시 값과 추출된 해시 값을 비교하여 제1 매니페스트 파일을 검증할 수 있다.

[0049] 제2 검증부(230)는 제1 펌웨어의 서명 값에서 제1 펌웨어를 추출하고, 제1 펌웨어와 제1 매니페스트를 비교하여 제1 펌웨어를 검증할 수 있다. 제2 검증부(230)는 제1 펌웨어의 서명 값을 사전에 수신한 펌웨어 제작자의 공개키를 이용하여 복호화함으로써 제1 펌웨어를 추출하고, 추출된 제1 펌웨어와 제1 매니페스트 파일에 포함된 제1 펌웨어의 버전 정보와 제1 펌웨어 해시의 서명 값을 비교하여 제1 펌웨어를 검증할 수 있다. 제2 검증부(230)는 제1 펌웨어의 검증이 완료되면 제1 펌웨어를 임시 저장할 수 있다.

[0050] 펌웨어 교체부(250)는 제1 펌웨어가 검증되면, 제1 매니페스트 파일에 포함된 제1 펌웨어의 버전 정보와 제1 펌웨어의 서명 값을 메모리에 저장할 수 있다. 이후 펌웨어 교체부(250)는 현재 저장된 제2 펌웨어를 제1 펌웨어로 교체할 수 있다.

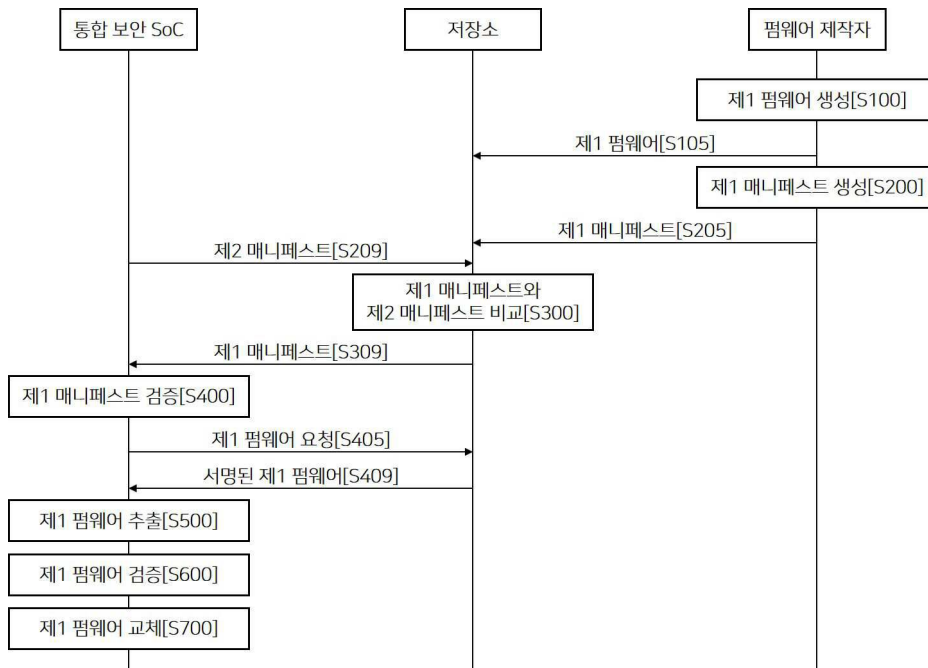
[0051] 본 명세서와 도면에 개시된 본 발명의 실시 예들은 본 발명의 기술 내용을 쉽게 설명하고 본 발명의 이해를 돕기 위해 특정 예를 제시한 것뿐이며, 본 발명의 범위를 한정하고자 하는 것은 아니다. 여기에 개시된 실시 예들 이외에도 본 발명의 기술적 사상에 바탕을 둔 다른 변형 예들이 실시 가능하다는 것은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 자명한 것이다.

도면

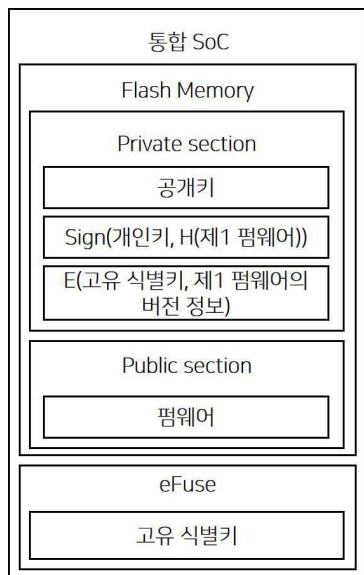
도면1



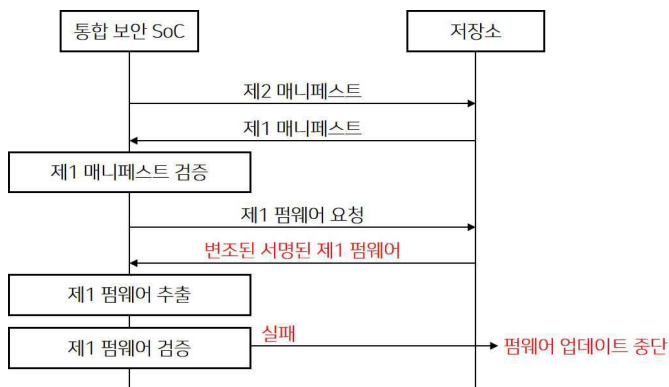
도면2



도면3



도면4



도면5



도면6

