(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년04월30일
(11) 등록번호 10-1974188
(24) 등록일자 2019년04월24일

(51) 국제특허분류(Int. Cl.)
*G06F 21/50* (2013.01) *G06F 21/53* (2013.01)
(21) 출원번호 10-2014-7002458
(22) 출원일자(국제) 2012년07월11일
심사청구일자 2017년06월19일
(85) 번역문제출일자 2014년01월28일
(65) 공개번호 10-2014-0054003
(43) 공개일자 2014년05월08일
(86) 국제출원번호 PCT/US2012/046243
(87) 국제공개번호 WO 2013/019369
국제공개일자 2013년02월07일
(30) 우선권주장
13/193,945 2011년07월29일 미국(US)
(56) 선행기술조사문헌
JP2008500651 A*
US20080189707 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
**마이크로소프트 테크놀로지 라이센싱, 엘엘씨**
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
**톰 스테판**
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
**콕스 제레미아**
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
*(뒷면에 계속)*
(74) 대리인
**제일특허법인(유)**

전체 청구항 수 : 총 17 항 심사관 : 문남두

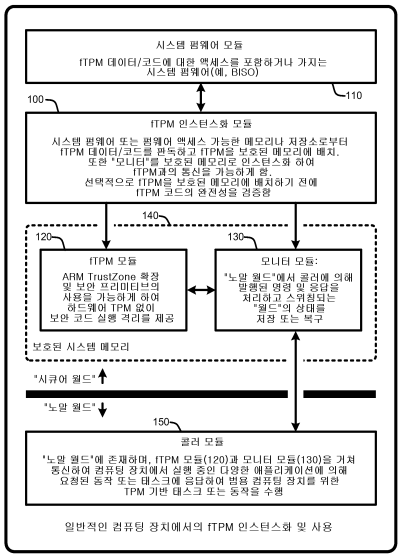
(54) 발명의 명칭 **ARM® TRUSTZONE™ 구현을 위한 펌웨어 기반 신뢰 플랫폼 모듈**

## (57) 요 약

"펌웨어 기반 TPM" 또는 "fTPM"은 보안 코드 실행이 격리되도록 하여 매우 다양한 잠재적인 보안 침해를 방지하는 것을 보증한다. 통상적인 하드웨어 기반 신뢰 플랫폼 모듈(TPM)과 달리, 전용 보안 프로세서 하드웨어 또는 실리콘을 사용하지 않고 격리가 이루어질 수 있다. 일반적으로, fTPM은 시스템 펌웨어 또는 펌웨어 액세스 가능

(뒷면에 계속)

*대 표 도* - 도1



한 메모리나 저장소로부터 fTPM을 판독하고 장치의 보호된 리드 온리(read-only) 메모리에 배치함으로써 프리 OS 부트 환경에서 먼저 인스턴스화된다. 인스턴스화 되면, fTPM은 보안 코드 실행을 보증하기 위해 실행 격리를 가능하게 한다. 더 구체적으로, fTPM이 보호 리드 온리 메모리에 배치되어 장치가 하드웨어(예를 들면, ARM® 아키텍처의 TrustZone™ 확장 및 보안 프리미티브(또는 유사한 프로세서 아키텍처))를 사용하는 것을 가능하게 하고, 이에 따라 이러한 아키텍처에 기초한 장치 가 현재의 장치에 대한 하드웨어 변경을 하지 않고, "펌웨어 기반 TPM" 내에 보안 실행 격리를 제공하는 것을 가능하게 한다.

(72) 발명자

**린슬레이 데이비드**

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

**나이스트롬 마그누스**

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

**라즈 히만슈**

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

**로빈슨 데이비드**

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

**사로이우 스테판**

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

**스피거 롭**

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

**월만 알라스테어**

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

# 명 세 서

## 청구범위

### 청구항 1

하드웨어 신뢰 플랫폼 모듈(TPM:trusted platform module) 컴포넌트 없이 컴퓨팅 장치에서 신뢰 실행 환경(TrEE:trusted execution environment)을 가능하게 하는 방법으로서,

컴퓨팅 장치의 펌웨어 컴포넌트로부터 fTPM(firmware-based TPM)을 검색하는 단계 - 상기 fTPM은 상기 컴퓨팅 장치의 하나 이상의 프로세서에 통합가능한(integral to) 보안 확장 기능에 대한 소프트웨어 기반 인터페이스를 제공함 - 와,

상기 컴퓨팅 장치의 펌웨어 컴포넌트로부터 소프트웨어 기반 모니터 모듈을 검색하는 단계와,

상기 컴퓨팅 장치 상에서 운영 체제(OS)를 부팅하기 전에 상기 fTPM 및 상기 모니터 모듈을 상기 컴퓨팅 장치의 보호된 메모리 내 시큐어 월드(Secure World) 환경으로 인스턴스화하는 단계 - 상기 하나 이상의 프로세서는 ARM(advanced RISC machine) 기반 프로세서 아키텍처를 사용하고, ARM 기반 프로세서에 통합가능한 상기 보안 확장 기능은 상기 fTPM의 인스턴스화 후에 fTPM에 의해 사용되는 TrustZone 유형의 보안 확장 및 보안 프리미티브를 포함함 - 와,

노말 월드(Normal World) 환경의 콜러(Caller) 모듈이 상기 모니터 모듈로의 시큐어 모니터 콜(Secure Monitor Call)을 통해 상기 하나 이상의 프로세서의 보안 기능에 액세스하도록 허용함으로써 상기 컴퓨팅 장치상에서 TrEE를 가능하게 하는 단계 - 상기 모니터 모듈은 이후에 상기 시큐어 모니터 콜과 관련된 명령어를 상기 시큐어 월드의 상기 fTPM으로 전달함 -

를 포함하는

방법.

### 청구항 2

제1항에 있어서,

상기 fTPM은 상기 컴퓨팅 장치에서 실행되는 하나 이상의 가상 머신에 의해 액세스 가능한

방법.

### 청구항 3

제1항에 있어서,

상기 fTPM을 상기 컴퓨팅 장치의 보호된 메모리 내 시큐어 월드 환경으로 인스턴스화하는 단계 전에 fTPM 코드 무결성이 검증되는

방법.

### 청구항 4

제1항에 있어서,

상기 콜러 모듈은, OS 부팅 전에 상기 TrEE를 하나 이상의 프리부트 애플리케이션에 노출하여 상기 애플리케이션이 상기 TrEE를 사용하여 태스크를 수행하도록 허용하는 프리 부트 애플리케이션 모듈을 포함하는

방법.

**청구항 5**

제1항에 있어서,

상기 콜러 모듈은, OS 부팅 이후에 상기 TrEE를 상기 OS에서 실행되는 하나 이상의 애플리케이션에 노출하여 상기 애플리케이션이 상기 TrEE를 사용하여 태스크를 수행하도록 허용하는 TPM 드라이버 모듈을 포함하는

방법.

**청구항 6**

제1항에 있어서,

상기 컴퓨팅 장치의 펌웨어 컴포넌트는 상기 펌웨어를 상기 fTPM을 포함하는 소프트웨어로 업데이트함으로써 상기 fTPM을 수신하는

방법.

**청구항 7**

제1항에 있어서,

상기 콜러 모듈과 상기 모니터 모듈 사이의 통신은 동기식인

방법.

**청구항 8**

제1항에 있어서,

상기 콜러 모듈과 상기 모니터 모듈 사이의 통신은 비동기식인

방법.

**청구항 9**

하드웨어 신뢰 플랫폼 모듈(TPM) 컴포넌트 없이 컴퓨팅 장치상에서 신뢰 컴퓨팅 환경을 구현하는 시스템으로서,

fTPM(firmware-based TPM)이 저장된 컴퓨팅 장치의 비휘발성 메모리 컴포넌트 - 상기 fTPM은 상기 컴퓨팅 장치의 하나 이상의 프로세서에 통합가능한 보안 확장 기능에 대한 소프트웨어 기반 인터페이스를 제공하고, 상기 비휘발성 메모리 컴포넌트는 소프트웨어 기반 모니터 모듈을 더 포함함 - 와,

상기 비휘발성 메모리 컴포넌트로부터 상기 fTPM 및 상기 모니터 모듈을 판독하고 상기 fTPM 및 상기 모니터 모듈을 상기 컴퓨팅 장치의 보호된 메모리 내의 시큐어 월드 환경으로 인스턴스화하는 장치를 포함하되,

상기 하나 이상의 프로세서는 ARM(advanced RISC machine) 기반 프로세서 아키텍처를 사용하고, ARM 기반 프로세서에 통합가능한 상기 보안 확장 기능은 상기 fTPM의 인스턴스화 후에 fTPM에 의해 사용되는 TrustZone 유형의 보안 확장 및 보안 프리미티브를 포함하며,

상기 장치는 또한, 노말 월드 환경의 콜러 모듈이 상기 모니터 모듈로의 시큐어 모니터 콜을 통해 상기 하나 이상의 프로세서의 보안 기능에 액세스하도록 허용함으로써 상기 컴퓨팅 장치 상에서 상기 신뢰 컴퓨팅 환경을 가능하게 하고, 상기 모니터 모듈은 이후에 상기 시큐어 모니터 콜에 관련된 명령어를 상기 시큐어 월드의 상기 fTPM으로 전달하는

시스템.

**청구항 10**

제9항에 있어서,

상기 콜러 모듈은, OS 부팅 전에 상기 신뢰 컴퓨팅 환경을 하나 이상의 프리부트 애플리케이션에 노출하여 상기 애플리케이션이 상기 신뢰 컴퓨팅 환경을 이용하여 태스크를 수행하도록 허용하는 프리 부트 애플리케이션 모듈을 포함하는

시스템.

**청구항 11**

제9항에 있어서,

상기 콜러 모듈은, OS 부팅 이후에 상기 신뢰 컴퓨팅 환경을 상기 OS에서 실행되는 하나 이상의 애플리케이션에 노출하여 상기 애플리케이션이 상기 신뢰 컴퓨팅 환경을 이용하여 태스크를 수행하도록 허용하는 TPM 드라이버 모듈을 포함하는

시스템.

**청구항 12**

제9항에 있어서,

상기 콜러 모듈과 모니터 모듈 사이의 동기 통신 및 비동기 통신 모두를 가능하게 하는 장치를 더 포함하는

시스템.

**청구항 13**

하드웨어 신뢰 플랫폼 모듈(TPM) 컴포넌트 없이 컴퓨팅 장치를 이용하여 신뢰 컴퓨팅 환경을 구현하는 컴퓨터 실행가능 명령어가 저장된 컴퓨터 판독가능 저장 장치로서,

상기 명령어는

컴퓨팅 장치의 하나 이상의 프로세서에 통합가능한 보안 확장 기능에 대한 소프트웨어 기반 인터페이스를 제공하는 fTPM(firmware-based TPM) 및 소프트웨어 기반 모니터 모듈과,

상기 fTPM 및 상기 모니터 모듈을 상기 컴퓨팅 장치의 비휘발성 메모리 컴포넌트에 로딩하는 명령어와,

상기 비휘발성 메모리로부터 상기 fTPM 및 상기 모니터 모듈을 검색하는 명령어와,

상기 fTPM 및 상기 모니터 모듈을 상기 컴퓨팅 장치의 보호된 메모리 내의 시큐어 월드 환경으로 인스턴스화하는 명령어와,

노말 월드(Normal World) 환경의 콜러(Caller) 모듈이 상기 모니터 모듈로의 시큐어 모니터 콜을 통해 하나 이상의 프로세서의 보안 기능을 액세스하는 것을 가능하게 함으로써 상기 컴퓨팅 장치상에서 신뢰 컴퓨팅 환경을 가능하게 하는 명령어 - 상기 모니터 모듈은 이후에 시큐어 모니터 콜과 관련된 명령어를 상기 시큐어 월드 내의 상기 fTPM으로 전달함 - 를 포함하되

OS 부팅 전에, 상기 콜러 모듈은, 상기 신뢰 컴퓨팅 환경을 하나 이상의 프리부트 애플리케이션에 노출하여 상기 애플리케이션이 상기 신뢰 컴퓨팅 환경을 이용하여 태스크를 수행할 수 있게 하는 프리 부트 애플리케이션 모듈을 포함하는

컴퓨터 판독가능 저장 장치.

**청구항 14**

제13항에 있어서,

상기 컴퓨팅 장치 내의 하나 이상의 프로세서는 ARM(advanced RISC machine) 기반 프로세서 아키텍처를 사용하고, ARM 기반 프로세서에 통합가능한 상기 보안 확장 기능은 보안 확장 및 보안 프리미티브를 포함하는

컴퓨터 판독가능 저장 장치.

**청구항 15**

제13항에 있어서,

OS 부팅 이후에, 상기 콜러 모듈은 상기 신뢰 컴퓨팅 환경을 상기 OS에서 실행되는 하나 이상의 애플리케이션에 노출하여 상기 애플리케이션이 상기 신뢰 컴퓨팅 환경을 이용하여 태스크를 수행하도록 허용하는 TPM 드라이버 모듈을 포함하는

컴퓨터 판독가능 저장 장치.

**청구항 16**

제13항에 있어서,

상기 하나 이상의 프로세서는 둘 이상의 코어를 포함하고, 상기 둘 이상의 코어 각각에 대해 상기 컴퓨팅 장치의 상기 보호된 메모리 내에 개별 fTPM이 인스턴스화되는

컴퓨터 판독가능 저장 장치.

**청구항 17**

제13항에 있어서,

상기 콜러 모듈과 모니터 모듈 사이의 동기 통신 및 비동기 통신 모두를 가능하게 하는 장치를 더 포함하는

컴퓨터 판독가능 저장 장치.

## *발명의 설명*

### *기 술 분 야*

[0001] "펌웨어 기반 TPM" 또는 "fTPM"은 하드웨어(예, ARM® 아키텍처의 TrustZone™ 익스텐션 및 보안 프리머티브)를 사용하여 "펌웨어 기반 TPM" 내의 신뢰 플랫폼 모듈(TPM:Trusted Plateform Module)에 대한 보안 실행 격리를 제공하는 다양한 기법을 제공하며, "펌웨어 기반 TPM"은 ARM® 기반 프로세서 아키텍처 또는 유사한 하드웨어를 사용하여 장치 내에 구현될 수 있다.

### *배 경 기 술*

[0002] 본 발명이 속하는 분야의 기술자에게 잘 알려진 바와 같이, 통상적인 TPM은 보안 암호화 프로세서(secure crypto-processor)를 제공하는 하드웨어 장치 또는 "칩"이다. 보다 구체적으로, 전형적인 TPM 칩은 일반적으로 하드웨어 의사 난수 생성기에 추가하여, 암호 키의 보안 생성 및 이들의 사용 제한을 용이하게 한다. 또한, 전형적인 TPM 칩은 "원격 증명"과 같은 사용범위(capability) 및 봉인된 저장소를 포함한다. 원격 증명은 특정한

하드웨어 및 소프트웨어 구성의 실질적인 위조방지 해시 키(unforgeable hash key) 요약을 생성하기 위한 것이다. 이 요약의 범위는 하드웨어 및 소프트웨어 구성에 대한 평가에 관련되는 컴포넌트에 의해 결정된다. 이는 제3 자가 소프트웨어 및 하드웨어 구성이 소정의 설정 정책(set policy)과 일치하는지를 검증하는 것을 가능하게 한다. "바인딩(Binding)"은 TPM 보증 키, TPM 칩의 생성 중에 TPM 칩에 각인된 고유 RSA 키 또는 이로부터 유래된 다른 신뢰 키를 사용하여 데이터를 암호화한다. "실링(Sealing)"은 바인딩과 유사하게 데이터를 암호화하나, 추가로 데이터가 복호화 또는 "언실링"되도록 하기 위해 TPM 칩이 존재해야 하는 상태를 특정한다.

[0003] 또한 TPM 칩은 하드웨어 장치를 인증하는데 사용된다. 각각의 TPM 칩은 TPM 칩이 생성되면서 각인된 고유한 비밀 RSA 키를 가지기 때문에, 플랫폼 인증을 수행할 수 있다. 예를 들어, TPM 칩은 액세스를 원하는 시스템이 기대 또는 인증된 시스템인지를 검증하는데 사용될 수 있다. 분명히, 대응하는 보안 소프트웨어와 함께 별개의 TPM 칩을 사용하여 시스템의 하드웨어 레벨로 보안을 다운하는 것은 소프트웨어만의 솔루션보다 나은 보안성을 제공한다. 그러나, TPM 칩이 사용되는 경우에도, 일반적인 콜드 부트 공격의 경우에, 설명된 것과 같이 TPM 칩에 의해 애플리케이션에 노출되면 키는 여전히 취약하다.

[0004] 컴퓨팅 시스템을 위한 TPM을 구현하는 많은 통상적인 솔루션은 별개의 하드웨어 TPM 칩을 이러한 컴퓨팅 시스템의 마더보드나 시스템 보드에 통합하는 것을 포함한다. 불행히도, 이러한 솔루션은 여러 과제에 직면하고 있다. 예를 들어, TPM 칩을 전형적인 마더보드 디자인에 통합하는 것은 시스템당 약 1달러 내지 2달러 정도의 BOM(bill of materials) 비용 증가를 발생시킨다. 그러나, 이러한 상대적으로 낮은 장치 당 비용은 전세계에서 생산되는 컴퓨팅 장치의 엄청난 규모를 고려하면 매우 큰 합으로 늘어날 수 있다. 통상적인 TPM 칩과 자주 연관되는 또 다른 과제는 별개의 TPM은 일반적으로 에너지 효율에 관해 최적화되지 않는다는 것이고 저전력 시스템(예, 휴대용 컴퓨팅 장치, PDA, 태블릿, 넷북, 모바일 폰 등)에 대한 전력 예산(power budget)에 영향을 미친다는 것이다. 또한, BOM 제약으로 인해, 별개의 TPM 칩은 종종 소정의 사용 시나리오에 부정적인 영향을 미치거나 잠재적으로 이를 방해하는 상대적으로 느린(그리고 이에 따라 저렴한) 프로세서로 구현된다.

[0005] 결과적으로, TPM은 일반적으로 선택적인 시스템 컴포넌트인 것으로 여겨지기 때문에, 시스템에 별개의 TPM을 포함하기 위한 추가적인 비용 및 전력적 대가로 인해 제조 공정 중에 이러한 장치를 배제시키게 된다. 이에 따라 TPM은 자유로이 사용할 수 있는 것(ubiquous)이 아니며, 소프트웨어 또는 운영 체제 개발자가 광범위한 TPM 사용 시나리오에 실질적인 자원을 투입하는 것을 어렵게 한다. 광범위한 TPM 사용 시나리오에 영향을 미치는 다른 이슈는 많은 통상적인 별개의 TPM이 일부 유형의 팩터(예, 폰, PDA, 태블릿 등)와 호환되지 않는다는 것이다. 사실상, 모바일 폰 및 태블릿 형태의 컴퓨터와 같은 많은 통상적인 장치가 일반적으로 별개의 TPM을 사용하지 않으며, 일부의 경우에 폰 또는 태블릿과 같은 SoC(system-on-chip) 구동 장치에서 별개의 TPM의 사용을 지원하기 위한 적절한 배선(예, LPC 버스)을 구비하지 않을 수 있다.

## *발명의 내용*

### *해결하려는 과제*

### *과제의 해결 수단*

[0006] 본 요약은 상세한 설명에서 이하에 추가로 설명되는 개념에 대한 선택 사항을 간략한 형태로 소개하기 위해 제공된다. 본 요약은 청구된 발명의 대상의 주요 특징이나 핵심 특징을 식별하려는 것이 아니며, 청구된 발명의 대상의 범주를 정하는 데 있어 보조 내용으로 사용하려는 것도 아니다. 또한, 종래 기술에 대한 소정의 문제점이 본 명세서에 언급되거나 논의될 수 있으나, 청구된 발명의 대상은 이러한 종래 기술의 문제점의 일부 또는 전부를 해결하거나 처리할 수 있는 구현예에 국한되지 않는다.

[0007] 신뢰성(trust)은 사람 또는 사물의 무결성(integrity)에 의존한다. 장치 사용자에 대하여, 장치에서의 신뢰는 설정 정책(set policy)과 일치하는 코드만이 장치에서 실행될 수 있다는 보증(guarantee)에 의해 형성된다. 강력한 무결성 보호를 이행하고 악성 감염 및 변조를 방어하기 위해, 하드웨어 및 소프트웨어의 조합이 사용된다. 운영 체제(OS)(예, Micorosoft® Windows®)는 이전에는 이러한 플랫폼 무결성을 다양한 시스템으로 전달하기 위한 하드웨어 컴포넌트로서 TPM(Trusted Platform Module)을 사용하였다. 불행히도, TPM의 광범위한 채용은 많은 이유(예를 들면, 별개의 TPM 컴포넌트를 마더보드에 부가하는 추가적인 BOM 비용, TPM을 그러한 장치에 연결하거나 부착하기 위해 적합한 인터페이스를 제공하도록 특정한 장치를 재설계하는 비용 및 시간 등을 포함함)로

인해 저항에 부딪히게 되었다.

[0008] 일반적으로, 본 명세서에 설명된 "펌웨어 기반 TPM" 또는 "fTPM"은 실질적으로 무비용(zero-cost) "펌웨어 TPM"을 구현하기 위해 하드웨어(예를 들면, ARM® 시스템 온 칩(SoC) 플랫폼, 또는 유사한 플랫폼)를 포함시키는 것과 연관된 비용을 해결하고, 이로써 시스템의 BOM 비용을 감소시키고, 장치의 전체 전력 소모를 낮추며, 넓은 범위의 ARM® 기반 장치들에 걸친 매우 다양한 TPM 사용 시나리오를 가능하게 한다. 통상적인 기법과 달리, fTPM은 fTPM에 의해 신뢰 컴퓨팅 환경이 사용될 수 있는 컴퓨팅 환경에서 사용될 하드웨어 TPM 모듈을 요구하지 않고, ARM® 프로세서와 같은 프로세서에 통합될 보안 확장 기능에 대한 소프트웨어 인터페이스를 제공한다.

[0009] 다르게 설명하면, 통상적인 기법과 대조적으로, fTPM은 하드웨어 TPM을 사용하지 않고 하드웨어 TPM에 의해 제공되는 것과 비교할 수 있는 컴퓨팅 장치 내의 신뢰 실행 환경을 가능하게 한다. 또한, fTPM을 구현하는 소프트웨어가 많은 현존하는 컴퓨팅 장치의 펌웨어 또는 보호된 비휘발성 메모리에 업로드, 플래시 또는, 저장이나 기입될 수 있어 다시 그러한 장치에 대한 임의의 하드웨어 변경을 하지 않고 TPM 기능을 사용하는 것을 가능하게 이들 장치를 "업그레이드"한다.

[0010] 보다 구체적으로, 임의의 TPM 구현은 매우 다양한 잠재적인 보안 침해를 방지하도록 시스템에서 동작하는 모든 다른 소프트웨어로부터 자신의 코드 및 데이터의 무결성과 비밀성을 보전하는 것을 보증한다. (실리콘을 부가하는 대가로) 전용 보안 프로세서를 이용하여 또는 하드웨어 아키텍처에 의해 제공되는 상향된 실행 권한 레벨을 이용하여 격리가 구현될 수 있다. 본 명세서에 설명된 펌웨어 기반 TPM은 시스템 펌웨어 또는 펌웨어 액세스 가능한 메모리 또는 저장소로부터 fTPM을 판독하고 fTPM을 간단한 "모니터"와 함께 장치의 보호된 메모리에 배치함으로써 프리-OS 부트 환경에서 먼저 인스턴스화된다.

[0011] "보호된 메모리", "보호된 저장소" 라는 용어 및 본 명세서에 사용되는 유사한 용어는 구체적으로 노말 월드(Normal World)와 같은 신뢰되지 않는 컴포넌트에 의해 판독 또는 변경될 수 없는 저장소로서 정의된다. 노말 오퍼레이션은 보호된 저장소에 포함되는 데이터 및 기능 모두를 판독하거나 기입할 수 없다. 예를 들어, OS는 노말 월드에서 동작하고 보호된 저장소를 판독 또는 기입할 수 없으나, fTPM을 포함하는 "시큐어 월드(Secure World)"에서는 가능하다. 이러한 보호된 메모리를 설정하는 한 가지 방식은 하드웨어(예, 메모리 또는 eMMC 저장소 컨트롤러)가 시큐어 월드에 의해서만 사용하도록 하기 위해 저장소(예, TrustZone 보호 메모리 또는 Replay 보호 메모리 블록)의 영역을 분할하는 것이다. OS가 "시큐어 월드"가 아닌 "노말 월드"에서 실행 중이기 때문에, 소정의 보호된 메커니즘(예, 본 명세서에서 이하에 설명되는 fTPM으로 전달되는 SMC(Secure Monitor Call) 명령어)을 사용하지 않는 한 OS는 시큐어로 마킹된 임의의 메모리를 액세스할 수 없다.

[0012] 본 명세서에 기술된 "모니터"는 구체적으로 "노말" 월드로부터 "시큐어 월드"가 격리되도록 하면서, "노말 월드"로부터의 통신이 "시큐어 월드"에서 동작하는 fTPM에 의해 수신되는 것을 가능하게 하는 인터페이스로서 정의된다. 또한, ARM ® 기반 아키텍처 및 TrustZone™ 확장과 같은 아키텍처의 "시큐어 월드" 및 "노말 월드" 동작 모드가 본 발명이 속하는 분야의 기술자에게 잘 알려져 있으며, 본 명세서에서 상세히 설명되지 않을 것이라는 것에 주의한다. TrustZone™ 확장은 복수의 플랫폼에 걸친 공통 보안 인프라스트럭처를 제공하는 데 있어 유용하다. 그러나, 본 명세서에 기술된 fTPM은 임의의 TPM 기반 보안 아키텍처를 이용하여 동작할 수 있다는 것을 이해해야 한다. 이러한 선택적인 보안 아키텍처의 예(즉, 선택적인 신뢰 실행 환경)는 TI OMAP 기반 아키텍처, M-Shield 기반 아키텍처, x86 시스템 관리 모드(SMM) 등을 포함하나 이에 한정되는 것은 아니다.

[0013] 인스턴스화되면, 펌웨어 기반 TPM은 현재의 ARM® 기반 아키텍처 및 TrustZone™ 확장을 사용하여 코드 및 데이터의 무결성 및 비밀성(confidentiality)을 보장하기 위한 실행 격리를 가능하게 하고 펌웨어 기반 "가상 전용 보안 프로세서"를 통해 "노말 월드"에 의한 액세스로부터 암호화 동작(및 저장소)의 격리를 가능하게 한다. 다르게 설명하면 본 명세서에 설명된 fTPM은 시스템 펌웨어 (또는 다른 소스)로부터 판독되고, 보호된 메모리에 배치되며, ARM® 기반 아키텍처 및 TrustZone™ 확장 및 보안 프리미티브를 사용하여 현재의 ARM® 기반 아키텍처 내에 그리고 이에 따라 그러한 아키텍처에 기초한 장치 내에 현재 장치에 대한 하드웨어 변경을 요하지 않고 구현될 수 있는 "펌웨어 기반 TPM" 내의 보안 실행 격리를 제공한다.

[0014] 결과적으로 펌웨어 기반 TPM의 한가지 이점은 실제로 TPM 칩 또는 다른 하드웨어를 필요로 하지 않고 TPM에 의해 요구되는 실행 격리를 구현하기 위해 펌웨어 기반 TPM 현재의 ARM® TrustZone™ 확장을 사용한다는 것이다. 대조적으로, 다양한 통상적인 시스템은 별개의 TPM을 사용하거나 SoC에 전용 보안 프로세서를 부가하였다. 불행히도, 이러한 접근법 모두는 시스템에 대해 추가 비용을 발생시킨다. 그러나, TrustZone™은 하이 엔드 ARM® SoC 로드맵의 광범위한 세트(broad set)에 대해 거의 유비쿼터스적인 특징을 보이나, 여전히 폭넓게 사용되지 않고 있다. 사용되는 경우에도, TrustZone™ 보안 지불, 모바일 뱅킹, DRM 등에 대한 적합한, 수직적(virtical)

솔루션으로 주로 사용되고 있다. 따라서, 현재의 장치의 펌웨어에 TPM을 구현하기 위해 펌웨어 기반 TPM가 TrustZone™을 사용하는 것은 현재의 장치에 대한 하드웨어 변경 없이 그러한 장치에 현저한 가치를 부여한다. 이와 같이, 복수의 SoC 플랫폼에 걸친 TPM 유비퀴티(ubiquity)가 본 명세서에 설명된 펌웨어 기반 TPM에 의해 가능하다.

[0015] 전술한 요약을 고려하면, 본 명세서에 설명된 펌웨어 기반 TPM은 ARM® 아키텍처의 TrustZone™ 확장 및 보안 프리미티브와 같은 하드웨어를 사용하여 현재의 ARM® 기반 아키텍처 내에서 그리고 이에 따라 그러한 아키텍처에 기초한 장치 내에서 구현될 수 있는 "펌웨어 기반 TPM" 내의 보안 실행 격리를 제공하기 위한 다양한 기법을 제공하는 것이 분명하다. 전술한 효과에 더하여, 펌웨어 기반 TPM의 다른 이점이 첨부된 도면과 함께 고려되는 경우에 이하에 이어지는 상세한 설명으로부터 명확하게 이해될 것이다.

## *도면의 간단한 설명*

[0016] 청구된 발명의 대상의 구체적인 특징, 측면 및 효과가 다음의 설명, 첨부된 청구범위 및 첨부된 도면을 참조하여 이해될 수 있을 것이다.

도 1은 본 명세서에 설명된, 일반적인 컴퓨팅 장치의 보호된 메모리에 "펌웨어 기반 TPM"을 인스턴스화하기 위한 제너럴 아키텍처 및 동작 흐름도를 나타낸다.

도 2는 본 명세서에 설명된, 프리-OS 부트 환경에서 "펌웨어 기반 TPM"을 사용하기 위한 제너럴 아키텍처 및 동작 흐름도를 나타낸다.

도 3은 본 명세서에 설명된, 시스템 부트에 뒤이어 OS 환경에서 "펌웨어 기반 TPM"을 사용하기 위한 제너럴 아키텍처 및 동작 흐름도를 나타낸다.

도 4는 본 명세서에 설명된, "펌웨어 기반 TPM"의 전형적인 동기식 동작의 예를 나타내는 흐름도를 제공한다.

도 5는 본 명세서에 설명된, "펌웨어 기반 TPM"의 전형적인 비동기식 동작의 예를 나타내는 흐름도를 제공한다.

도 6은 본 명세서 설명된, 펌웨어 기반 TPM의 다양한 실시예를 구현하는데 사용하기 위한 간략화된 컴퓨팅 및 I/O 기능을 가진 간략화된 범용 컴퓨팅 장치를 나타내는 포괄적인 시스템 도면이다.

## *발명을 실시하기 위한 구체적인 내용*

[0017] 청구된 발명의 대상의 실시예에 대한 다음의 설명에서, 본 명세서의 일부이며 청구된 발명의 대상이 실행될 수 있는 특정한 실시예를 예시로서 도시하는 첨부된 도면이 참조된다. 청구된 발명의 대상의 범주를 벗어나지 않는 범위에서 다른 실시예가 이용될 수 있고 구조적 변경이 이루어질 수 있다는 것을 이해할 수 있을 것이다.

[0018] 1.0 서문

[0019] 일반적으로, 통상적인 하드웨어 TPM에서와 같이, 본 명세서에 설명된 "펌웨어 기반 TPM" 또는 "fTPM"은 시스템에서 실행 중인 모든 다른 소프트웨어로부터 자신의 코드 및 데이터의 무결성 및 비밀성을 보전하여 매우 다양한 잠재적인 보안 침해를 막고, 매우 다양한 보안 애플리케이션(예, 암호화 애플리케이션, 보안 난수 생성, 디스크/파일 암호화, 패스워드 인증 등)을 가능하게 하는 것을 보증한다. (실리콘을 부가하는 대가로) 전용 보안 프로세서를 이용하여 또는 하드웨어 아키텍처에 의해 제공되는 상향된 실행 권한 레벨을 이용하여 격리가 구현될 수 있다. 효과적으로, 본 명세서에 설명된 펌웨어 기반 TPM은 하드웨어 TPM과 동일한 보안 코드 실행을 가능하게 하는 데 통상적인 하드웨어 TPM의 물리적 하드웨어를 필요로 하지 않는다.

[0020] 다르게 설명하면, 통상적인 기법과 대조적으로, fTPM은 ARM® 프로세서와 같은 프로세서에 통합되는 보안 확장 기능에 대한 소프트웨어 기반 인터페이스를 제공하여, 하드웨어 TPM을 사용하지 않고도 하드웨어 TPM에 의해 제공되는 것에 비견할 수 있는 컴퓨팅 장치의 신뢰 실행 환경(이는 또한 신뢰 컴퓨팅 환경이라고도 함)을 가능하게 한다. 또한, fTPM을 구현하는 소프트웨어가 업로드 또는 플래시되거나 또는 많은 현재의 컴퓨팅 장치의 펌웨어 또는 보호된 비휘발성 메모리에 저장 되거나 기입되어, 이러한 장치에 대한 임의의 하드웨어 변경을 요하지 않고 TPM 기능의 사용을 가능하게 하도록 이러한 장치를 "업그레이드"할 수 있다.

[0021] "보호된 메모리", "보호된 저장소" 라는 용어 및 본 명세서에 사용되는 유사한 용어는 구체적으로 노말 월드

(Normal World)와 같은 신뢰되지 않는 컴포넌트에 의해 판독 또는 변경될 수 없는 저장소로서 정의된다는 것에 유의한다. 정상 동작은 보호된 저장소에 포함되는 데이터 및 기능 모두를 판독하거나 기입할 수 없다. 예를 들어, OS는 노말 월드에서 동작하고, 보호된 저장소를 판독 또는 기입할 수 없으나, fTPM을 포함하는 "시큐어 월드(Secure World)"에서는 가능하다. 이러한 보호된 메모리를 설정하는 한 가지 방식은 시큐어 월드에 의해서만 사용되도록 하드웨어(예, 메모리 또는 eMMC 저장소 컨트롤러)가 저장소(예, TrustZone 보호 메모리 또는 Replay 보호 메모리 블록)의 영역을 분할하는 것이다. OS가 "시큐어 월드"가 아닌 "노말 월드"에서 실행 중이기 때문에, 소정의 보호된 메커니즘(예, 본 명세서에서 상세히 설명되는 fTPM으로 전달되는 SMC(Secure Monitor Call) 명령어)을 사용하지 않는 한 OS는 시큐어로 마킹되는 임의의 메모리를 액세스할 수 없다.

[0022] 예를 들어, 다양한 실시예에서, fTPM을 구현하는 소프트웨어가 전형적인 BIOS 또는 펌웨어 업데이트에 간단히 포함될 수 있어 리부트 시에 그와 같은 장치에 TPM 기능(capability)을 즉시 제공할 수 있다. 시스템 펌웨어 및/또는 BIOS를 업데이트하기 위한 다양한 프로세스 및 기법이 본 발명이 속하는 분야의 기술자에게 잘 알려져 있으며 본 명세서에 상세히 설명되지 않을 것이라는 점에 유의한다. 결과적으로, 설명을 위해, 다음의 논의에서 fTPM을 구현하는 소프트웨어가 컴퓨팅 장치(이 장치에서, fTPM이 TPM 기능(functionality)을 인에이블하기 위해 사용됨)의 펌웨어에 이미 제공되었다고 가정할 것이다.

[0023] 구체적으로, fTPM은, 시스템 펌웨어 또는 펌웨어 액세스 가능한 메모리 또는 저장소로부터 fTPM을 판독하고 fTPM을 장치의 보호된 메모리에 배치함으로써 프리 운영체제(OS) 부트 환경에서 먼저 인스턴스화된다. 다양한 실시예에서, 프리 OS 부트 환경(또는 펌웨어)은 코드가 보호된 메모리에 배치되도록 허용하기 전에 fTPM 코드의 무결성을 자동으로 검증하여(예를 들면, fTPM 코드의 "서명(signature)"을 검증함으로써) 부정하게 변경되지 않았다는 것을 보증한다는 점에 유의한다. 또한, fTPM은 OS 부트에 뒤이어 보호된 메모리에 로딩되거나 인스턴스화될 수 있으나, 프리-OS 부트 환경에서 fTPM을 인스턴스화함으로써 전체 시스템 보안성을 보증하는 것이 더 쉽다는 점에 유의한다. 나아가, fTPM은 비보호 메모리에 로딩되거나 인스턴스화될 수 있으나, 이러한 경우에 일반적으로 보안성에 대한 보장이 없을 것이다. 시스템 부트 이전 또는 이후에 또는 시스템 부트 중에 데이터(예를 들면, 본 명세서에 설명된 예에서 fTPM)를 보호된 메모리에 기입하기 위한 다양한 프로세스가 본 발명이 속하는 분야의 기술자에게 잘 알려져 있으며 본 명세서에서 설명되지 않을 것이다.

[0024] 인스턴스화 되면, 펌웨어 기반 TPM은 이어서, ARM® 기반 아키텍처와 같은 현재의 하드웨어 및 TrustZone™ 확장(또는 유사한 기법)을 사용하여 펌웨어 기반 "가상 전용 보안 프로세서"를 통해 코드 실행의 보안성을 보증하기 위한 실행 격리를 가능하게 한다. 다르게 설명하면, 본 명세서 설명된 펌웨어 기반 TPM이 보호된 메모리에 배치되고, 펌웨어 기반 TPM은 ARM® 아키텍처의 TrustZone™ 확장 및 보안 프리미티브(또는 유사한 기법)로 하여금 현재의 ARM® 기반 아키텍처 내에 그리고 이에 따라 이러한 아키텍처에 기초한 장치 내에 구현될 수 있는 "펌웨어 기반 TPM" 내에 보안 실행 격리를 제공하도록 한다. TrustZone™ 확장은 이들이 복수의 플랫폼에 걸친 공통 보안 인프라스트럭처를 제공한다는 점에서 유용하다는 것에 유의한다. 또한, 설명을 위해, fTPM이 ARM® 아키텍처의 TrustZone™ 확장 및 보안 프리미티브를 사용하여 구현되는 내용으로 포괄적으로 설명될 것이라는 점에 유의해야 한다. 그러나, 본 명세서 설명된 fTPM은 매우 다양한 TPM 기반 보안 아키텍처를 이용하여 동작할 수 있다는 것을 이해해야 한다.

[0025] 효과적으로, 이러한 보안 실행 격리는 현재의 장치에 대한 하드웨어 변경을 필요로 하지 않고 또한 물리적 하드웨어 TPM을 필요로 하지 않고, fTPM에 의해 제공된다. 결과적으로, fTPM은 실질적으로 TPM 칩이나 다른 하드웨어를 요구하지 않고도 매우 다양한 장치 내에서 보다 쉽고 저렴하게 구현된다. 또한, fTPM은 모든 TPM 표준과 완전히 호환된다. 이와 같이, fTPM은 보통 하드웨어 TPM을 필요로 하는 임의의 구현예에서 사용될 수 있다. 또한, fTPM을 이용하는 소프트웨어 또는 하드웨어의 관점에서, fTPM은 하드웨어 기반 TPM과 구분할 수 없다.

[0026] 나아가, fTPM은 인스턴스화 중에 보호된 메모리에 간단히 기입되기 때문에 fTPM의 복수의 사본 또는 버전이 보호된 메모리의 구분된 영역에 기입될 수 있고 이로써 각각의 구분된 프로세서, 코-프로세서, 멀티 프로세서 또는 멀티 CPU 시스템의 CPU, 및 다른 별개의 이종 또는 비대칭 프로세서 아키텍처마다 구분된 TPM 가용범위(capabilty)를 가능하게 한다는 점에 유의해야 한다. SoC 및 다른 새로운 시스템 아키텍처가 동일한 코어의 전통적인 컬렉션과 다른 가용범위를 가진 추가 코어를 부가하고, fTPM은 이러한 장치 및 하드웨어와 함께 동작할 수 있다는 점에 유의한다. 마찬가지로, 단일 시스템 내에서 실행되는 가상 환경(예, 가상 머신)의 경우에, 이러한 가용범위는 구분된 그리고 격리된 TPM 가용범위가 각각의 가상 환경에 제공되는 것을 가능하게 한다. 본 발명이 속하는 분야의 기술자가 잘 이해할 수 있는 바와 같이, 가상 머신(VM)은 물리적 머신과 같은, 프로그램을 실행하는 머신(즉, 컴퓨터)의 소프트웨어 구현예이다. 가상 머신은 일반적으로 이들의 용도 및 임의의 실제 머신에 대한 일치성 정도에 기초하여 두 개의 주요 카테고리로 나뉜다. 구체적으로, 시스템 가상 머신은 완전한

OS의 실행을 지원하는 완전한 시스템 플랫폼을 제공하는 반면, 프로세스 가상 머신은 단일 프로그램을 실행하도록 디자인되고, 이는 단일 프로세스를 지원하는 것을 의미한다.

[0027] 설명을 목적으로, 다음의 논의는 전반적으로 단일 프로세서 시스템 내의 fTPM의 단일 인스턴스에 초점을 맞출 것이라는 점에 유의한다. 그러나, 전술한 논의의 관점에서, fTPM의 복수의 인스턴스가 멀티 프로세서 또는 멀티 코어 시스템, 다른 이종 또는 비대칭 프로세서 아키텍처 및 하나 이상의 가상 환경(예, 가상 머신(VM)) 상에서 실행되는 시스템 내에 구현될 수 있다는 것을 이해해야 한다. 또한, 다음의 설명은 TrustZoneTM "시큐어 월드" 및 "노말 월드" 동작 모드를 지칭한다는 점에 유의한다. 이러한 동작 모드는 본 발명이 속하는 분야의 기술자에게 알 알려져 있으며, 본 명세서에서 상세히 설명되지 않을 것이다.

[0028] 1.1 **시스템 개관:**

[0029] 전술한 바와 같이, "펌웨어 기반 TPM" 또는 "fTPM"은 ARM® 아키텍처의 TrustZone™ 확장 및 보안 프리미티브를 사용하여 현재의 장치에 대한 하드웨어 변경을 필요로 하지 않고 현재의 ARM® 기반 아키텍처(또는 유사한 기법) 및 이에 따라 이러한 아키텍처에 기반한 장치 내에서 구현될 수 있는 펌웨어 기반 TPM 내의 보안 실행 격리를 제공하는 다양한 기법을 제공한다. 다음의 설명에서 "장치"라는 용어는 대체로 일반적인 컴퓨팅 장치(퍼스널 컴퓨터, 서버 컴퓨터, 핸드 헬드 컴퓨팅 장치, 랩톱 또는 모바일 컴퓨터, 통신 장치(예, 폰 또는 PDA), 멀티 프로세서 시스템, 마이크로프로세서 기반 시스템, 셋톱 박스, 프로그램 가능한 가전 기기, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 오디오 또는 비디오 미디어 플레이어 등을 포함하나 이에 한정되는 것은 아님)(장치 내에 fTPM이 인스턴스화되어 있어, 그 장치에 TPM 가용범위를 제공함)를 지칭한다는 것에 유의한다.

[0030] 위에 요약된 프로세스는 도 1의 포괄적인 시스템 도면에 도시된다. 구체적으로 도 1의 시스템 도면은 본 명세서에 설명된 장치 내에 fTPM의 다양한 실시예를 구현하는 프로그램 모듈 사이의 상호관계를 나타낸다. 또한, 도 1의 시스템 도면은 fTPM의 다양한 실시예에 대한 상위 수준의 도면을 나타내나, 이러한 도면은 본 명세서 전체에 설명되는 바와 같이 fTPM의 모든 가능한 실시예에 대한 완전한 또는 완성된 실례(illustration)를 제공하려는 것이 아니다.

[0031] 일반적으로, 도 1에 도시된 바와 같이, 펌웨어 기반 TPM에 의해 인에이블되는 프로세스는, fTPM 인스턴스화 모듈(100)을 사용하여 fTPM 데이터/코드(즉, fTPM의 실행가능한 소프트웨어 실시예)를 fTPM이 인테이블 될 장치의 시스템 펌웨어(110) 또는 펌웨어 액세스 가능한 메모리(또는 저장소)로부터 판독함으로써 동작을 시작한다. fTPM 인스턴스화 모듈(100)은 이어서 fTPM을 시스템 메모리(140)의 보호된 메모리 위치에 로딩하거나 배치하여 장치 내에서 fTPM의 인스턴스화를 가능하게 한다. 또한, 다양한 시스템에서, fTPM 인스턴스화 모듈(100)은 코드가 보호된 메모리에 배치되도록 허용하기 전에 fTPM 코드의 무결성을 자동으로 검증하여(예를 들면, fTPM 코드의 "서명(signature)"을 검증함으로써) 부정하게 변경되지 않았다는 것을 보증한다는 점에 유의한다. 또한, fTPM 인스턴스화 모듈(100)은 fTPM과의 통신을 허용하기 위해 보호된 시스템 메모리(140)로 "모니터"(즉, "모니터 모듈 130")을 인스턴스화 한다. 본 명세서에 설명된 "모니터"는 "노말 월드"로부터 "시큐어 월드"를 격리시킨 상태를 유지하면서 "노말 월드"로부터의 통신이 "시큐어 월드"에서 동작하는 fTPM에 의해 수신되는 것을 가능하게 하는 인터페이스로서 구체적으로 정의된다.

[0032] 일반적으로, 모니터 모듈(130)은 "노말 월드"에서 콜러(즉, 콜러 모듈(150))에 의해 발행된 커맨드 및 응답을 처리하고, 스위칭되는 "월드"의 상태를 저장 또는 복구한다. 더 구체적으로, 모니터(즉, 모니터 모듈(130))는 콜러 모듈(150)로부터의 커맨드 또는 요청(즉, 요청된 "동작")을 가로챈다. 이후에 모니터 모듈(130)은 fTPM 모듈(120)로 이러한 커맨드 또는 요청을 전달하고, 이는 이어서 자신의 시큐어 월드 환경에서 그 동작을 실행한다. fTPM 모듈(120)은 이후에 실행된 동작으로부터의 응답을 공유 메모리에 기입하고, 노말 월드 환경에서 콜러(즉, 콜러 모듈(150))로 다시 시스템을 복구하는 모니터 모듈(130)로 복귀한다. 마지막으로, 콜러 모듈(150)(또는 다른 애플리케이션)은 공유 메모리로부터 fTPM 응답을 검색한다.

[0033] 다르게 설명하면, 대체로 콜러 모듈(150)은 "노말 월드"에 존재하고 "시큐어 월드"의 모니터 모듈(130)을 통해 fTPM 모듈(120)과 통신하고, fTPM 모듈은 컴퓨팅 장치의 OS(또는 프리 OS 부트 환경)에서 실행되는 다양한 애플리케이션에 의해 콜러 모듈(150)을 통해 요청되는 태스크 또는 동작에 응답하여 일반적인 컴퓨팅 장치를 위한 TPM 기반 태스크 또는 동작을 수행한다.

[0034] 2.0 펌웨어 기반 TPM의 동작 세부사항

[0035] 전술한 프로그램 모듈은 펌웨어 기반 TPM의 다양한 실시예를 구현하기 위해 사용된다. 위에 요약한 바와 같이, 펌웨어 기반 TPM은 ARM®과 같은 하드웨어 아키텍처의 TrustZone™ 확장 및 보안 프리미티브를 사용하여 현재의 장치에 대한 하드웨어 변경을 필요로 하지 않고 현재의 ARM® 기반 아키텍처 내에서 그리고 이에 따라 이러한 아키텍처에 기반한 장치 내에서 구현될 수 있는 "펌웨어 기반 TPM" 내에 보안 실행 격리를 제공하는 다양한 기법을 제공한다.

[0036] 다음의 섹션은 펌웨어 기반 TPM의 다양한 실시예의 동작 및 도 1을 참조하여 섹션 1에 설명된 프로그램 모듈을 구현하는 예시적인 방법에 대한 상세한 설명을 제공한다. 구체적으로, 다음의 섹션은 펌웨어 기반 TPM의 다양한 실시예(fTPM에 대한 아키텍처 개관, fTPM을 이용한 시스템 초기화, "동작 컨텍스트에 의존하는 "콜러", 동기식 동작 및 비동기식 동작, 및 일반적인 컴퓨팅 장치 내에서 ARM® TrustZone™ 인에이블 fTPM을 구현하는 것을 포함함)에 대한 예시적이고 동작가능한 세부사항을 제공한다.

[0037] 2.1 fTPM에 대한 아키텍처 개요

[0038] 전술한 바와 같이, 펌웨어 기반 TPM 기반 프로세스는 ARM®과 같은 하드웨어 아키텍처의 TrustZone™ 확장 및 보안 프리미티브를 사용하여 현재의 ARM® 기반 아키텍처 및 이에 따라 이러한 아키텍처에 기반한 장치 내에서 구현될 수 있는 "펌웨어 기반 TPM" (본 명세서에서 "fTPM"이라고도 함)내에서 TPM에 의해 요구되는 실행 격리를 제공하는 다양한 기법을 제공한다.

[0039] 보다 구체적으로, fTPM은 임의의 ARM® 기반 SoC 플랫폼에서 이용가능한 ARM® 아키텍처 TrustZone™ 확장을 이용함으로써 현재의 하드웨어 내에 펌웨어 TPM을 제공한다. 본 발명이 속하는 분야의 기술자에게 잘 알려진 바와 같이, TrustZone™ 은 정상 실행 환경(노말 월드라 함)에서 실행되는 시스템(예, Windows® 운영체제(OS) 및 UEFI 프리 부트 환경) 내의 다른 컴포넌트로부터 강하게 격리되는 보안 실행 환경("시큐어 월드"라고 함)을 제공한다. UEFI(Unified Extensible Firmware Interface)는 운영체제 및 플랫폼 펌웨어 사이의 소프트웨어 인터페이스를 정의하는 공지의 사양(specification)이다. 각각의 TrustZone™ 인에이블 fTPM 인스턴스는 하드웨어 격리 메커니즘을 제공하는 최소 TCB(minimal Trusted Computing Base)와, 펌웨어로부터의 특정한 fTPM을 보호된 시스템 메모리에 부트스트랩으로 삽입하는(bootstraps) 보안 부트 로더를 포함하고(전술한 도 1의 설명 참조), 이로써 통상적인 하드웨어 기반 TPM과 비견할 수 있으나 통상적인 하드웨어 기반 TPM에 필요한 추가적인 하드웨어에 대한 추가 비용 또는 복잡성이 없이 보호된 환경을 제공한다.

[0040] 2.2 fTPM을 이용한 시스템 초기화

[0041] 일반적으로, 도 1을 참조하여 설명한 바와 같이, 시스템 초기화 중에, 프로세서가 노말 월드 동작 모드로 전환하기 전에, 플랫폼의 fTPM 구현예가 시스템 펌웨어(예, BIOS 또는 다른 시스템 펌웨어)로부터 장치의 보호된 메모리로 설치되고 이와 함께 TrustZone™ 시큐어 월드로 간단한 모니터가 설치된다. 도 1을 참조하여 전술한 바와 같이, 간단한 모니터(즉, 모니터 모듈(130))는 두 개의 태스크를 수행한다:

[0042] 1) 노말 월드에서 콜러에 의해 발행된 커맨드/응답 처리

[0043] (대부분의 시나리오에서, 콜러는 이하에 추가로 상세히 설명되는 부트 펌웨어 또는 운영체제 드라이버일 수 있음에 유의한다)

[0044] 2) 스위치되는 월드의 상태를 저장 또는 복구.

[0045] 일반적으로, 노말 월드에서 실행되는 콜러와 시큐어 월드에서 실행되는 fTPM 사이의 통신 인터페이스는 동기식 또는 비동기식 공유 메모리 기반 인터페이스이다. 콜러는 SMC(Secure Monitor Call) 명령어를 사용하여 모니터로 진입하고, 동기식 및 비동기식 I/O 모두가 다양한 실시예에서 지원된다. SMC 명령어는 ARM® TrustZone™ 기법과 연관된 잘 알려진 명령어 유형이고, 그러한 이유로, SMC 명령어는 본 명에서에서 상세히 설명되지 않을 것이다.

[0046] 2.3 동작 컨텍스트에 따른 "콜러"

[0047] 도 2 및 도 3에 도시된 바와 같이, 펌웨어 기반 TPM은 현재 동작 컨텍스트가 프리 OS 부트 환경인지 아니면 OS 환경인지 여부에 따라 상이한 "콜러"를 사용한다. 다음의 논의는 fTPM이 전술한 바와 같이, 장치의 보호된 메모리에 이미 로딩되거나 인스턴스화된 것으로 가정한다.

[0048] 예를 들어, 도 2에 도시된 바와 같이, 테스트되는 실시예에서, UEFI 프리 OS 부트 환경 내에서, 콜러(200)는 UEFI fTPM 드라이버이다. 그러나, UEFI fTPM 드라이버의 사용은 fTPM의 필수조건이 아니며, UEFI fTPM 드라이버의 사용은 단지 시큐어 월드에서 fTPM 모듈(120)과의 프리 OS 통신을 개시하기 위한 하나의 방법이라는 것을 이해해야 한다. 따라서, UEFI fTPM 드라이버를 사용하는 것으로 가정하면, 콜러(200)의 UEFI fTPM 드라이버는 TrEE(Trusted Execution Environment) UEFI 프로토콜을 프리 부트 애플리케이션(예, Windows® 부트 매니저 및 다른 Windows® OS 로더)에 노출한다. TrEE UEFI 프로토콜은 본 발명이 속하는 분야의 기술자에게 잘 알려져 있으며, 본 명세서에서 상세하게 설명되지 않을 것이다. 또한, 본 명세서에 설명된 fTPM은 Windows® 형태의 운영체제를 이용하는 용도에 한정되지 않으며, 또한 다른 운영체제(예, LINUX, UNIX, iOS, OS X, 크롬, 안드로이드 등)가 본 명세서에 설명된 fTPM을 이용하여 동작할 수 있다는 점에 주의한다.

[0049] 일반적으로, 도 2에 도시된 것과 같이, 프리 OS 부트 환경에서, 다양한 TPM 기반 기능을 장치에 제공하도록 보호된 시스템 메모리에 로딩되는 fTPM 모듈(120)과 통신하기 위해 콜러 모듈(200)(예, UEFI)이 사용된다. 콜러 모듈(200)("노말 월드"의 비보호 환경에 존재함) 및 fTPM 모듈(120)("시큐어 월드"의 보호 환경에 존재함) 사이의 통신이 모니터 모듈(130)을 사용하여 fTPM 모듈(120)과 콜러 모듈(200) 사이의 통신을 가로채고 패스하는 SMC(Secure Monitor Call)을 사용하여 수행된다. 전술한 바와 같이, fTPM 모듈(120)의 초기화 중에, 모니터 모듈(130)이 시스템 펌웨어 또는 다른 펌웨어 액세스 가능한 메모리 또는 저장소로부터 시스템 보호 메모리로 설치된다.

[0050] 더욱 구체적으로, 다양한 실시예의, 프리 OS 부트 환경에서, UEFI 기반 구현예를 사용하는 경우에, 콜러 모듈(200)은 하나 이상의 프리 부트 애플리케이션(210) 및, fTPM 드라이버 모듈(230)과 통신 연결되는 인터페이스 모듈(220)을 포함한다. 도 2에 도시된 예시적인 실시예에서, 인터페이스 모듈(220)은 프리 부트 애플리케이션과 fTPM 드라이버 모듈(230) 사이의 통신을 가능하게 하는 통상적인 TrEE EFI 인터페이스를 사용하여 구현된다. 또한, 도 2에 도시된 예시적인 실시예에서, fTPM 드라이버 모듈(230)은 전술한 바와 같이, SMC 명령어(250)를 통해 콜러 모듈(200)과 모니터 모듈(130) 사이의 동기식 또는 비동기식 통신을 가능하게 하는 TrustZone™ 통신 모듈(240)을 추가로 포함하는 EFI 인터페이스를 사용하여 구현된다. 또한, 프리 OS 부트 환경에서 활성화된 fTPM이 시스템 부트 뒤에 OS 환경에서 액세스되는 공유 메모리에 정보를 기입함으로써 OS 환경으로 그 정보를 전달할 수 있다는 것에 유의해야 한다.

[0051] 대조적으로, 도 3에 도시된 바와 같은 OS 환경 내에서, 콜러 모듈(300)은, 다양한 실시예에서 ACPI(Advanced Configuration and Power Interface)를 사용하여 선택적으로 구현되는 새롭게 정의된 TrEE 인터페이스 모듈(320)을 포함한다. Windows® 기반 OS를 가정하면, TrEE 인터페이스 모듈(320)은 모든 커널 서비스에 대해 fTPM에 대한 액세스를 제공한다(애플리케이션이 하드웨어 TPM의 TPM 기능을 액세스하는 것을 허가하도록 디자인된 Windows® OS의 공지의 "TPM.sys" 장치 드라이버(130)를 포함함). 콜러 모듈(300)과 모니터 모듈(130) 사이의 통신이 SMC 명령어(250)를 통해 TrustZone™ 통신 모듈(330)에 의해 인에이블된다.

[0052] **2.4 동기식 동작:**

[0053] 도 4에 도시된 것과 같이, 전형적인 동기식 동작 시나리오에서, 다음의 단계가 순차적으로 수행된다:

[0054] 1) 콜러가 공유 메모리에 커맨드를 기입(400)

[0055] 2) 콜러가 SMC 명령어를 실행하여 모니터에 진입(410)

[0056] 3) 모니터가 시큐어 월드에서 fTPM 인스턴스로 커맨드를 전달(415)

[0057] 4) fTPM이 시큐어 월드에서 동작을 실행(420)(프로세서가 둘 이상의 코어 또는 서브 프로세서를 가지는 경우에, 이러한 코어 또는 서브 프로세서 중 하나 이상이 다른 동작이나 태스크에서 자유로운 다른 코어를 남겨두면서 TPM 동작에 구체적으로 특정될 수 있다는 점에 유의)

[0058] 5) fTPM이 공유 메모리에 응답을 기입하고 모니터로 복귀(425)

[0059] 6) 모니터가 노말 월드에서 시스템을 다시 콜러로 복구(430)

[0060] 7) 콜러(또는 다른 애플리케이션)가 공유 메모리로부터의 fTPM 응답을 검색(435).

[0061] **2.5 비동기식 동작:**

[0062] 도 5에 도시된 바와 같이, 전형적인 비동기식 동작 시나리오에서, 다음의 단계가 순차적으로 수행된다:

[0063] 1) 콜러가 공유 메모리에 커맨드를 기입(500)

[0064] 2) 콜러가 SMC 명령어를 실행하여 모니터에 진입(510)

[0065] 3) 모니터가 시큐어 월드에서 fTPM 인스턴스로 커맨드를 전달(520)

[0066] 4) fTPM이 시큐어 월드에서 동작의 실행을 시작하나 동작의 완료 전에 모니터로 다시 복귀하고, 모니터는 콜러로 다시 복귀함(이는 장기간의 암호화 동작에서 콜러의 CPU 시간의 부족을 방지하는데 필요함)(530)

[0067] 5) 동작이 완료되었는지를 알기 위해 확인하고(540), 완료되지 않았으며, 콜러가 요청된 동작이 완료될 때까지 시큐어 월드 내에서의 실행을 위한 fTPM 인스턴스 CPU 사이클을 제공하기 위해 단계 2 내지 4를 반복함(프로세서가 둘 이상의 코어 또는 서브 프로세서를 가지는 경우에, 이러한 코어 또는 서브 프로세서 중 하나 이상이 다른 동작이나 태스크에서 자유로운 다른 코어를 남겨두면서 TPM 동작에 구체적으로 특정될 수 있다는 점에 유의)

[0068] 6) fTPM이 공유 메모리에 응답을 기입하고 모니터로 복귀(550)

[0069] 7) 모니터가 노말 월드에서 시스템을 다시 콜러로 복구(560)

[0070] 8) 콜러(또는 다른 애플리케이션)가 공유 메모리로부터의 fTPM 응답을 검색(570).

[0071] **2.6 ARM® TrustZone™ 인에이블 fTPM의 구현:**

[0072] ARM® TrustZone™ 인에이블 fTPM을 구현하는 것은 다음을 포함하는, 프로세서 내의 하드웨어 프리미티브를 이용한다(제한은 아님):

[0073] 1) 암호화 알고리즘을 위한 하드웨어 가속(예, AES, RSA, SHA-x 등),

[0074] 2) 하드웨어 기반 난수 생성(RNG),

[0075] 3) 보안사항(secrets)을 저장하기 위해 바람직하게 격리되는 비휘발성 저장소 액세스.

[0076] 효과적으로, 많은 통상적인 ARM® SoC 및 유사한 프로세서는 이미 하드웨어 프리미티브로서 포함되는 이러한 특징을 가진다. 따라서, 본 명세서에 설명된 펌웨어 기반 TPM은 통상적인 ARM® SoC 또는 다른 보안 인에이블 프로세서가 fTPM을 이용하여 구성되는 것을 가능하게 하고, 이로써 OS가 초기화된 후에, 실질적으로 장치에 대한 추가 BOM 비용이 없는 완전히 기능적인 TPM에 대한 액세스를 가진다. 이는 ARM® SoC 플랫폼과 같은 하드웨어가 다음을 포함하는 다양한 태스크를 수행하는 것을 가능하게 한다(제한은 아님):

[0077] 1) Windows® 특성(예, Bitlocker®), 가상 스마트카드, 신중한 부팅(Measured Boot) 등을 자연적으로 지원

[0078] 2) ARM® 기반 장치에서의 보다 나은 전력 효율 대 분리된 TPM 솔루션의 전달

[0079] 3) 하드웨어 TPM의 통합이 더 이상 필요하지 않음으로 인한 전체 시스템 BOM 비용 및 장치 디자인 복잡성 감소

[0080] 4) 통상적인 구분된 TPM을 사용하는 장치에 대해 향상된 효율성을 주는 전력 민감 장치에서 다양한 새로운 TPM 사용 시나리오를 가능하게 함.

[0081] **3.0 예시적인 동작 환경**

[0082] 본 명세서에 설명된 fTPM은 수많은 유형의 범용 또는 전용 컴퓨팅 시스템 환경 또는 구성 내에서 동작가능하다. 도 6은 본 명세서에 설명된 fTPM의 다양한 실시예 및 구성요소가 구현될 수 있는 범용 컴퓨터 시스템의 간략한 예를 도시한다. 도 6의 파선 또는 단속선에 의해 표현된 임의의 박스는 간략화된 컴퓨팅 장치의 선택적인 구현예를 나타내고, 이하에 설명되는 이러한 선택적인 구현예 중 임의의 또는 모든 예가 본 명세서 전체에 걸쳐 설명된 다른 선택적인 구현예와 함께 사용될 수 있다는 점에 유의해야 한다.

[0083] 예를 들어, 도 6은 간략한 컴퓨팅 장치(600)를 도시하는 일반적인 시스템 도면을 나타낸다. 이러한 컴퓨팅 장치는 일반적으로 적어도 일부의 최소 컴퓨팅 기능을 가지는 장치(퍼스널 컴퓨터, 서버 컴퓨터, 핸드 헬드 컴퓨팅 장치, 랩톱 또는 모바일 컴퓨터, 통신 장치(예, 휴대폰 및 PDA), 멀티프로세서 시스템, 마이크로프로세서 기반 시스템, 셋톱 박스, 프로그램 가능한 가전기기, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 오디오 또는 비디오 미디어 플레이어 등을 포함하나 이에 제한되는 것은 아님)에서 찾아볼 수 있다.

[0084] 장치가 fTPM을 구현하는 것을 가능하게 하기 위해, 장치는 시스템 펌웨어(625)(또는 다른 펌웨어 액세스 가능한 메모리 또는 저장소(이로부터 fTPM이 보호된 시스템 메모리(620)로 인스턴스화됨))와 함께 기본 컴퓨팅 동작을 가능하게 하는 데 충분한 컴퓨팅 기능 및 시스템 메모리(620)를 구비해야 한다. 구체적으로, 도 6에 도시된 것과 같이, 컴퓨팅 기능은 포괄적으로 하나 이상의 프로세싱 유닛(들)(610)에 의해 도시되고, 또한 하나 이상의 GPU(615)(이들 중 하나 또는 모두가 시스템 메모리(620)와 통신 연결됨)를 포함할 수 있다. 범용 컴퓨팅 장치(600)의 그러한 프로세싱 유닛(들)(610)이 마이크로프로세서(가령, DSP, VLIW, 또는 다른 마이크로 컨트롤러)로 특화될 수 있고, 또는 하나 이상의 프로세싱 코어를 가진 통상적인 CPU(멀티 코어 CPU의 특화된 GPU 기반 코어를 포함함)일 수 있다.

[0085] 또한, 도 6의 간략화된 컴퓨팅 장치는 다른 컴포넌트(예를 들면, 통신 인터페이스(630))를 포함할 수도 있다. 또한, 도 6의 간략화된 컴퓨팅 장치는 하나 이상의 통상적인 컴퓨터 입력 장치(640)(예, 포인팅 장치, 키보드, 오디오 입력 장치, 비디오 입력 장치, 햅틱 입력 장치, 유선 또는 무선 데이터 전송을 수신하는 장치 등)를 포함할 수 있다. 또한, 도 6의 간략화된 컴퓨팅 장치는 다른 선택적 컴포넌트(예를 들면, 하나 이상의 통상적인 컴퓨터 출력 장치(650)(예, 디스플레이 장치(들)(655), 오디오 출력 장치, 비디오 출력 장치, 유선 또는 무선 데이터 전송 장치 등))를 포함할 수 있다. 범용 컴퓨터를 위한 전형적인 통신 인터페이스(630), 입력 장치(640), 출력 장치(650) 및 저장 장치(660)가 본 발명이 속하는 분야의 기술자에게 잘 알려져 있으며 본 명세서에 상세히 설명되지 않을 것이라는 점에 주의한다.

[0086] 도 6의 간략화된 컴퓨팅 장치는 또한 다양한 컴퓨터 판독가능 매체를 포함할 수 있다. 컴퓨터 판독가능 매체는 저장 장치(660)를 통해 컴퓨팅 장치(600)에 의해 액세스될 수 있는 임의의 이용가능 매체일 수 있으며, 컴퓨터 판독가능 또는 컴퓨터 실행가능 명령어, 데이터 구조, 애플리케이션, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 제거가능(670) 및/또는 제거가능하지 않은(680) 휘발성 매체 및 비휘발성 매체 모두를 포함한다. 예로서(제한이 아님), 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있다. 컴퓨터 저장 매체는 DVD, CD, 플로피 디스크, 테이프 드라이브, 하드 드라이브, 광학 드라이브, 솔리드 스테이트 메모리 드라이브, RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기법, 자기 카세트, 자기 테이프, 자기 디스크 저장소, 또는 다른 자기 저장 장치, 또는 원하는 정보를 저장하는 데 사용될 수 있고 하나 이상의 컴퓨팅 장치에 의해 액세스될 수 있는 임의의 다른 장치를 포함하나, 이에 한정되는 것은 아니다.

[0087] 컴퓨터 판독가능 또는 컴퓨터 실행가능 명령어, 데이터 구조, 애플리케이션, 프로그램 모듈 등과 같은 정보의 저장소는 또한 하나 이상의 모듈화된 데이터 신호 또는 반송파를 인코딩하기 위한 다양한 전술한 통신 매체 중 임의의 매체, 또는 다른 전송 메커니즘 또는 통신 프로토콜을 사용하여 획득될 수 있고, 임의의 유선 또는 무선 정보 전달 매커니즘을 포함한다. "모듈화된 데이터 신호" 또는 "반송파"라는 용어는 일반적으로 신호 내에 정보를 인코딩하는 방식으로 설정 또는 변경되는 신호의 특성 중 하나 이상을 가진 신호를 지칭한다. 예를 들어, 통신 매체는 하나 이상의 모듈화된 데이터 신호를 운반하는 유선 네트워크나 직접 무선 연결과 같은 유선 매체와, 음향, RF, 적외선, 레이저 및 다른 하나 이상의 모듈화된 데이터 신호 또는 반송파를 송신 및/또는 수신하는 다른 무선 매체와 같은 무선 매체를 포함한다. 또한, 전술한 임의의 매체의 조합은 통신 매체의 범주 내에 포함되어야 한다.

[0088] 또한, 본 명세서에 설명된 fTPM의 다양한 실시예의 일부 또는 전부를 구현하는 애플리케이션, 소프트웨어, 프로그램 및/또는 컴퓨터 프로그램 제품, 또는 이들의 일부가 컴퓨터 또는 머신 판독가능 매체 또는 저장장치 및 컴퓨터 실행가능 명령어나 다른 데이터 구조 형식의 통신 매체의 임의의 바람직한 조합으로 저장, 수신, 송신 또는 판독될 수 있다.

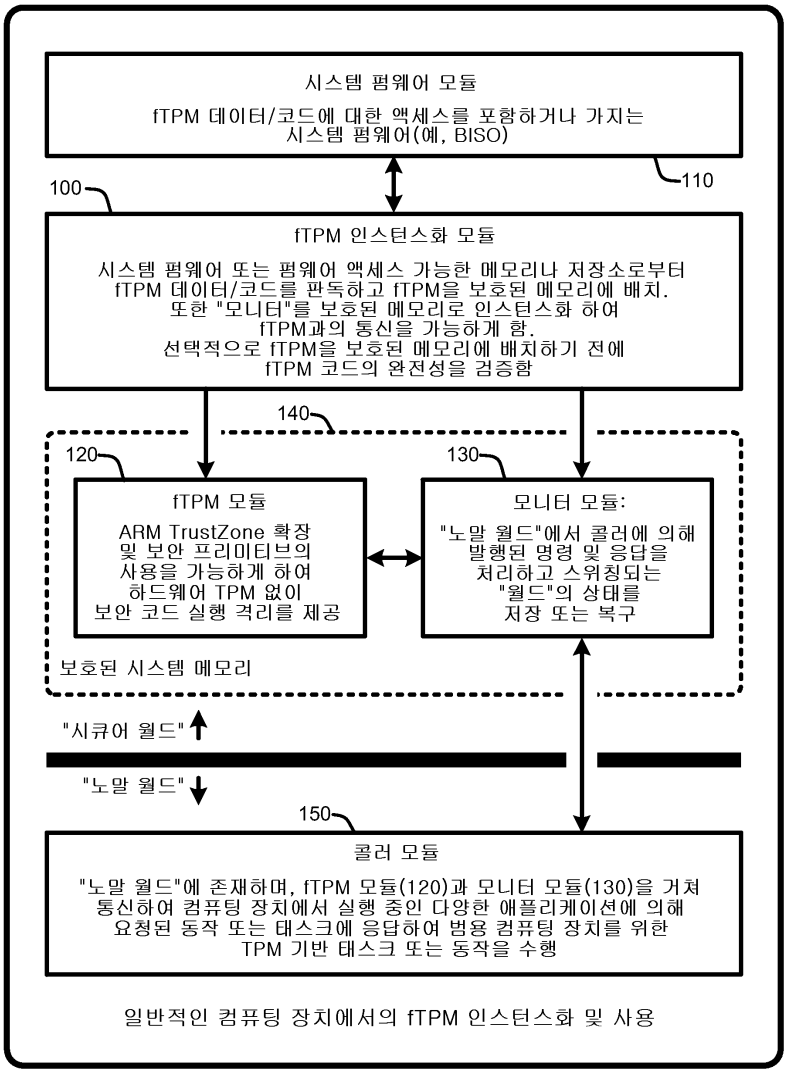
[0089] 마지막으로, 본 명세서에 설명된 fTPM은 컴퓨팅 장치에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행가능 명령어에 대한 일반적인 내용으로 추가 설명될 수 있다. 일반적으로, 프로그램 모듈은 루틴, 프로그램, 객체, 컴포넌트, 데이터 구조 등(이는 특정 태스크를 수행하거나 특정한 추상 데이터 유형을 구현함)을 포함한다. 또한, 본 명세서에 기술된 실시예는 하나 이상의 원격 처리 장치에 의해 태스크가 수행되는 분산형 컴퓨팅 환경에서 또는 하나 이상의 통신 네트워크를 통해 연결되는 하나 이상의 장치의 클라우드 내에서 실현될 수 있다. 분

산형 컴퓨팅 환경에서, 프로그램 모듈은 매체 저장 장치를 포함하는 로컬 및 원격 컴퓨터 저장 매체 모두에 배치될 수 있다. 또 다르게는, 전술한 명령어가 하드웨어 로직 회로(이는 프로세서를 포함하거나 포함하지 않을 수 있음)로서 부분적으로 또는 전체적으로 구현될 수 있다.

[0090] 펌웨어 기반 TPM에 대한 전술한 내용은 예시 및 설명을 목적으로 제시되었다. 이는 청구된 발명의 대상을 정확히 개시된 형태로 한정하거나 제한하려는 것이 아니다. 많은 변경 및 변형이 전술한 내용을 참조하여 이루어질 수 있다. 또한, 전술한 선택적인 실시예의 임의의 또는 모든 예가 펌웨어 기반 TPM에 대한 추가적인 혼합 실시예를 형성하는데 바람직한 임의의 조합으로 사용될 수 있다. 본 발명의 범주는 이러한 상세한 설명에 의해 제한되지 않으며 이하에 첨부된 청구범위에 의한다.

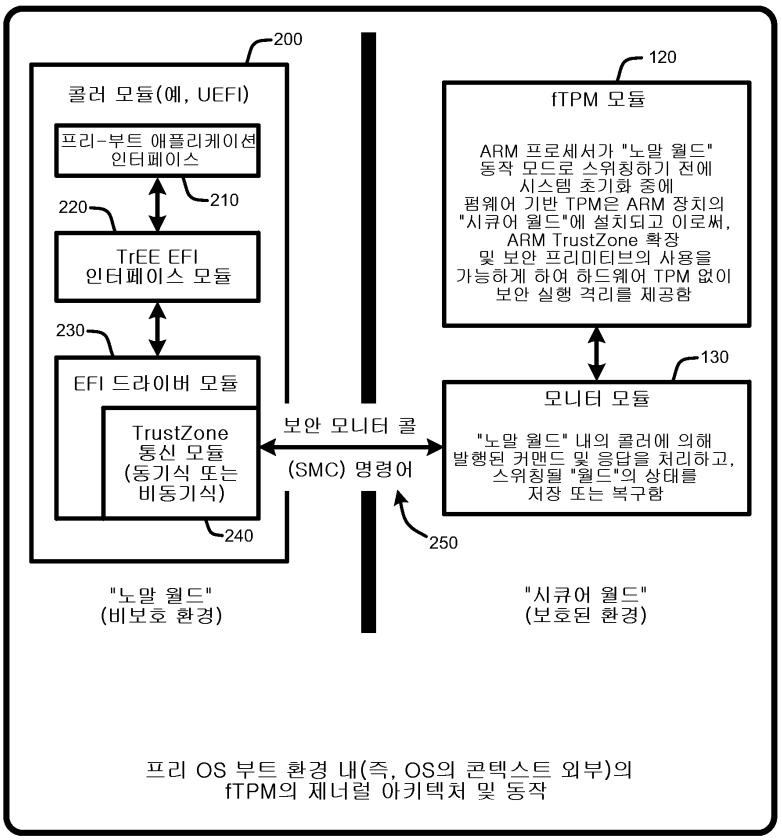
## 도면

### 도면1



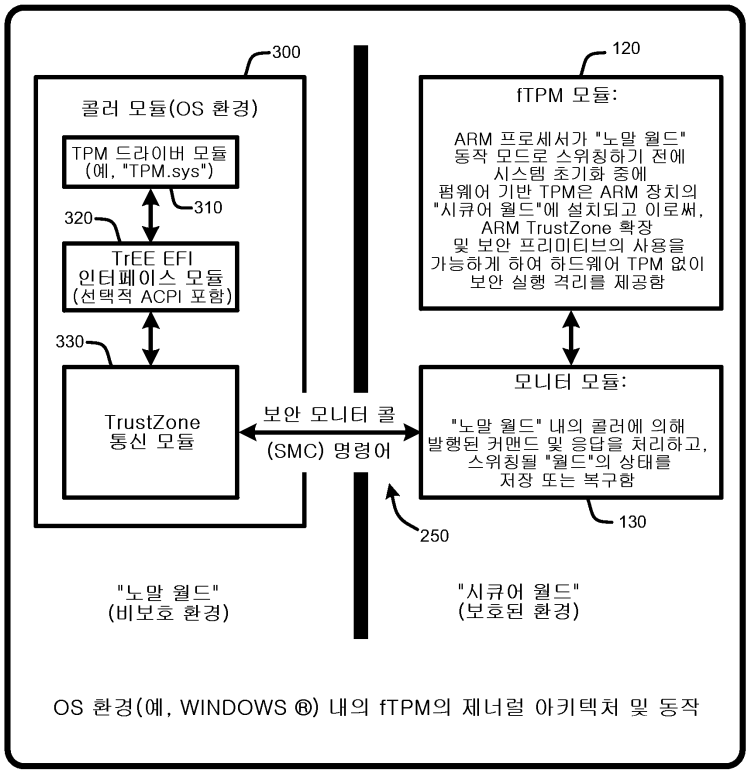

일반적인 컴퓨팅 장치에서의 fTPM 인스턴스화 및 사용

도면2



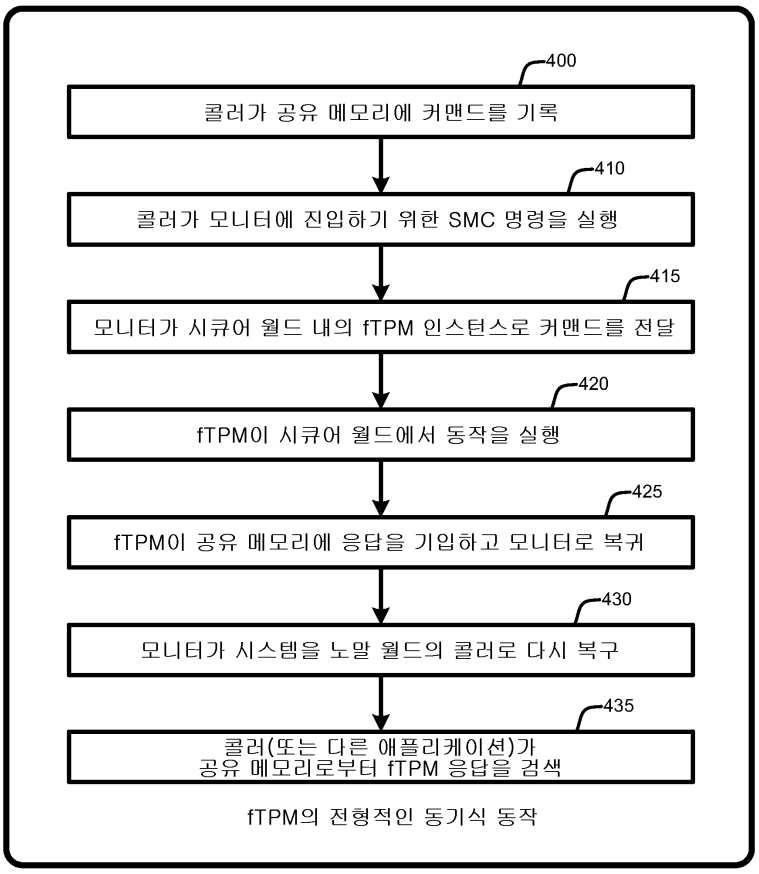

프리 OS 부트 환경 내(즉, OS의 콘텍스트 외부)의 fTPM의 제너럴 아키텍처 및 동작

도면3



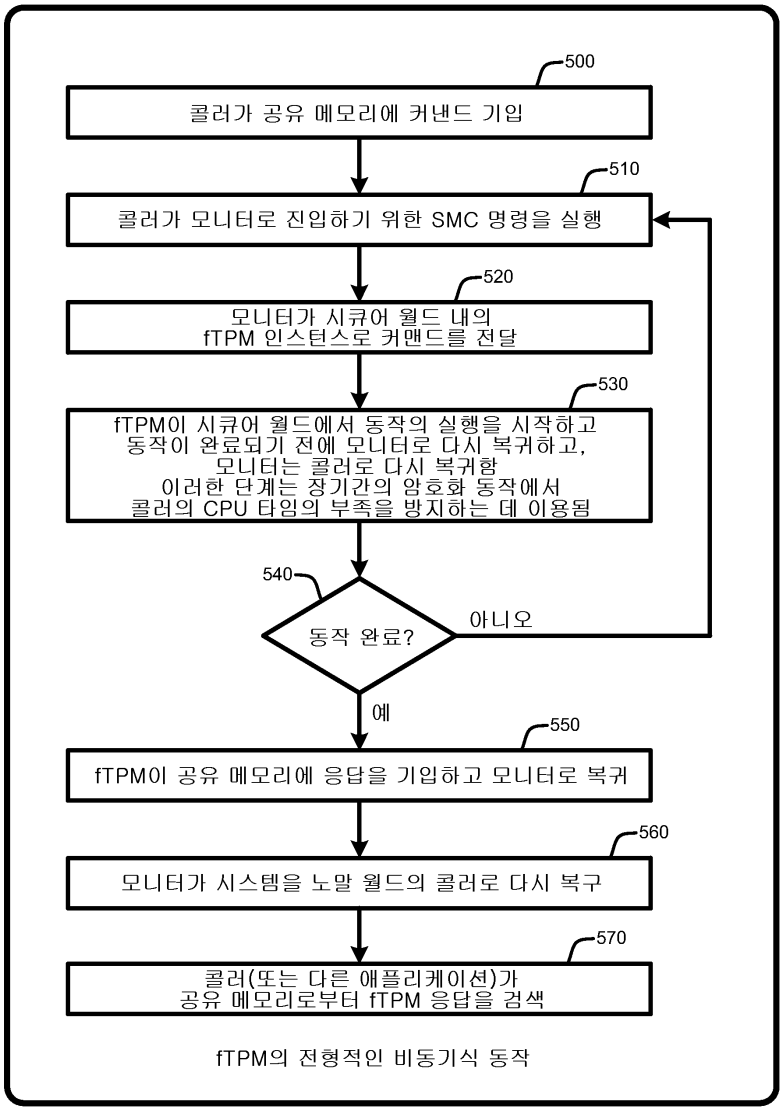

OS 환경(예, WINDOWS ®) 내의 fTPM의 제너럴 아키텍처 및 동작

도면4



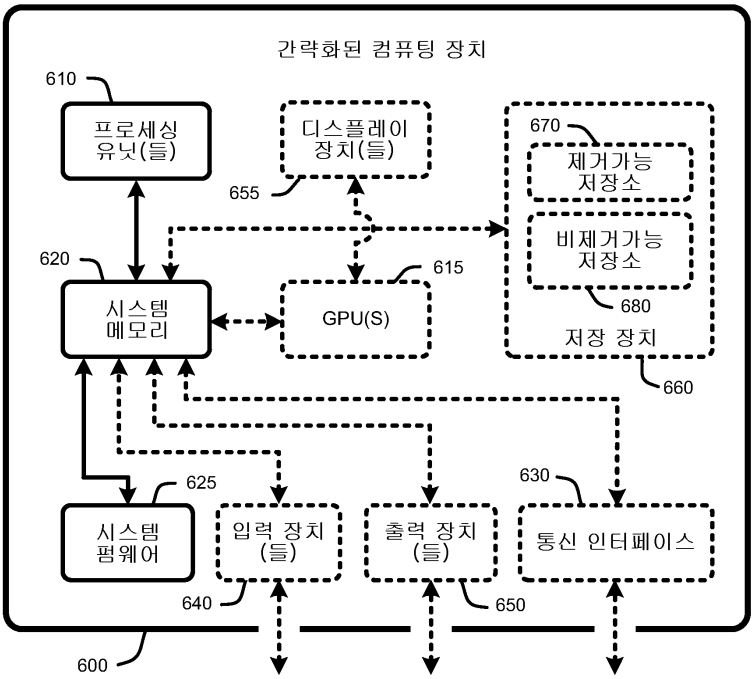

fTPM의 전형적인 동기식 동작

도면5




fTPM의 전형적인 비동기식 동작

도면6


간략화된 컴퓨팅 장치
610
프로세싱 유닛(들)
디스플레이 장치(들)
655
670
제거가능 저장소
비제거가능 저장소
680
저장 장치
660
620
시스템 메모리
615
GPU(S)
625
시스템 펌웨어
입력 장치(들)
640
출력 장치(들)
650
630
통신 인터페이스
600