(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0104780
(43) 공개일자 2024년07월05일

(51) 국제특허분류(Int. Cl.)
G06F 21/53 (2013.01) *G06F 21/55* (2013.01)
G06F 21/57 (2013.01)
(52) CPC특허분류
G06F 21/53 (2013.01)
G06F 21/55 (2013.01)
(21) 출원번호 10-2022-0187368
(22) 출원일자 2022년12월28일
심사청구일자 2022년12월28일

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
장진수
대전광역시 유성구 대학로 99, 충남대학교 공대5호관 620호
(74) 대리인
이은철, 김재문

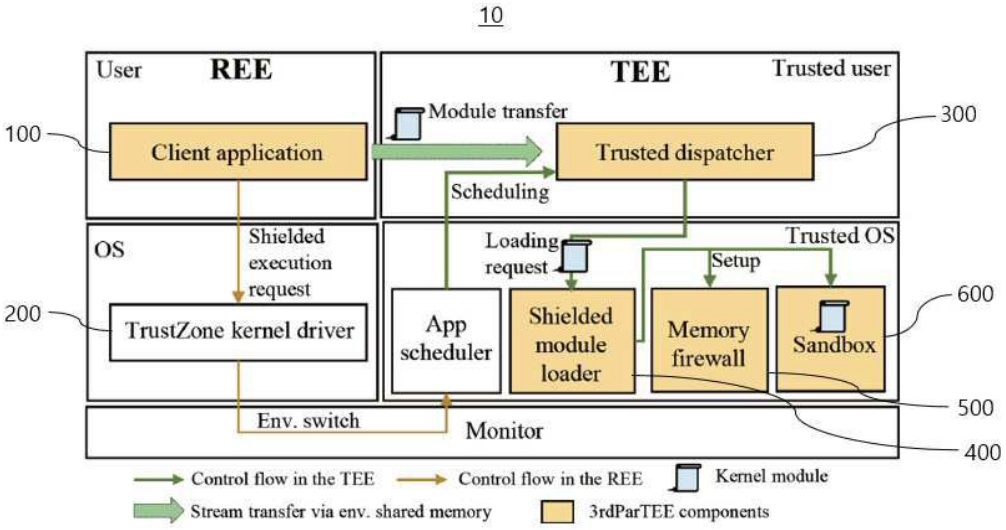
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 **모바일 기기의 동적커널모듈 보호 방법 및 이를 이용한 시스템**

(57) 요 약

본 발명은, 일반 실행 환경에서 차폐 커널 모듈의 실행을 트리거하기 위해 차폐된 커널 모듈에 관한 정보를 신뢰 실행 환경에 대한 압축 메시지로 생성하는 클라이언트 애플리케이션; 상기 클라이언트 애플리케이션에서 생성된 압축 메시지를 복사하여 신뢰 실행 환경으로 전송하는 TrustZone 커널 드라이버; 신뢰 실행 환경에서 차폐된 커널 모듈을 초기화하고 실행하기 위해, 압축 메시지를 수신하고 상기 차폐된 커널 모듈의 무결성을 확인하며 하기 차폐된 모듈 로더를 호출하는 신뢰할 수 있는 디스패처; 및 상기 디스패처의 호출에 의해 상기 차폐된 커널 모듈을 실행하는 차폐된 모듈 로더;를 포함하는 것을 일 특징으로 한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 21/57 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711152816
과제번호 2020-0-01840-003
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보보호핵심원천기술개발(정진기금)
연구과제명 스마트폰의 내부데이터 접근 및 보호 기술 분석
기 여 율 1/1
과제수행기관명 서울대학교 산학협력단
연구기간 2022.01.01 ~ 2022.12.31

명 세 서

청구범위

청구항 1

일반 실행 환경에서 차폐 커널 모듈의 실행을 트리거하기 위해 차폐된 커널 모듈에 관한 정보를 신뢰 실행 환경에 대한 압축 메시지로 생성하는 클라이언트 애플리케이션;

상기 클라이언트 애플리케이션에서 생성된 압축 메시지를 복사하여 신뢰 실행 환경으로 전송하는 TrustZone 커널 드라이버;

신뢰 실행 환경에서 차폐된 커널 모듈을 초기화하고 실행하기 위해, 압축 메시지를 수신하고 상기 차폐된 커널 모듈의 무결성을 확인하며 하기 차폐된 모듈 로더를 호출하는 신뢰할 수 있는 디스패처; 및

상기 신뢰할 수 있는 디스패처의 호출에 의해 상기 차폐된 커널 모듈을 실행하는 차폐된 모듈 로더;를

포함하는 모바일 기기의 동적커널모듈 보호 시스템.

청구항 2

제 1 항에 있어서,

상기 클라이언트 애플리케이션은,

차폐된 커널 모듈의 이미지를 파일 스트림으로 로드하고, 로드된 커널 모듈의 시작 주소, 모듈 크기, 진입점 및 서명된 해시를 신뢰 실행 환경의 API를 사용하여 신뢰 실행 환경에 대한 메시지로 압축하며, 프로세서 간 인터럽트(inter-processor interrupt, IPI)를 호출하여 다른 코어를 오프(off)시키는 것을 특징으로 하는 모바일 기기의 동적커널모듈 보호 시스템.

청구항 3

제 1 항에 있어서,

상기 클라이언트 애플리케이션은,

신뢰 실행 환경에 압축 메시지가 전달될 수 있도록 상기 TrustZone 커널 드라이버를 오픈(open)시키는 것을 특징으로 하는 모바일 기기의 동적커널모듈 보호 시스템.

청구항 4

제 1 항에 있어서,

상기 디스패처는,

수신한 압축 메시지를 구문 분석하여 상기 차폐된 커널 모듈의 주소와 크기를 획득하고, 신뢰 실행 환경의 샌드박스 영역에 메모리를 할당하고 획득한 차폐된 커널 모듈의 크기 정보를 기반으로 상기 차폐된 커널 모듈의 이미지를 메모리가 할당된 샌드박스 영역에 복사하는 것을 특징으로 하는 모바일 기기의 동적커널모듈 보호 시스템.

청구항 5

일반 실행 환경에서 사용자 애플리케이션이 커널 모듈을 신뢰 실행 환경으로 전송하고 차폐 실행을 요청하는

(a)단계;

신뢰 실행 환경에서 신뢰할 수 있는 디스패처가 상기 커널 모듈의 무결성을 확인하고 차폐된 모듈 로더를 호출하는 (b)단계; 및

상기 차폐된 모듈 로더가 상기 커널 모듈을 안전하게 실행하고 신뢰 실행 환경의 보안을 유지하도록 메모리 방화벽과 샌드박스를 구성하는 (c)단계;를

포함하는 모바일 기기의 동적커널모듈 보호 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 모바일 기기의 동적커널모듈 보호 방법 및 이를 이용한 시스템에 관한 것으로서, 특히 운영체제가 공격자로부터 장악당한 상황에서도 안전하게 실행될 수 있는 모바일 기기의 동적커널모듈 보호 방법 및 이를 이용한 시스템에 관한 것이다.

배 경 기 술

[0003] 사물 인터넷(Internet of Things, IoT) 산업은 계속해서 성장하고 있고, IoT 장치의 수는 2050년까지 240억 개로 증가할 것으로 예상된다. 또한, 가정 내 엔터테인먼트, 중요 인프라 및 산업을 포함한 일상생활을 위한 IoT 장치가 계속해서 등장하고 있다. 불행하게도 이러한 상황은 IoT 장치를 공격자의 매력적인 표적이 될 수 있고, IoT 장치와 관련하여 보안이 빠른 배포 및 생산성보다 우선순위가 낮은 경향이 있으므로 IoT 세계에 더 많은 공격자를 끌어들일 수 있다.

[0004] 신뢰 실행 환경(Trusted Execution Environment, TEE)은 일반 실행 환경(Rich Execution Environment, REE)에서 실행 컨텍스트(Execution context)를 격리할 수 있어, IoT 장치를 보호하기 위한 유망한 기술이다. 즉, 일반 실행 환경의 운영체제가 손상된 경우에도 암호화 작업 및 펌웨어 복구와 같은 보안에 중요한 논리 및 작업을 효과적으로 격리할 수 있다.

[0005] ARM TrustZone은 소형 IoT 장치에서 서버에 이르는 ARM 아키텍처 기반 플랫폼에서 사용할 수 있는 신뢰 실행 환경 기술 중 하나이다. 클라우드 서버용 가상화 신뢰 실행 환경, 모바일 장치용 일회용 암호, 보안 광고 등 TrustZone을 사용하는 다양한 보안 애플리케이션이 제안되고 있다. 특히, TrustZone을 활용하여 운영체제의 권한을 박탈하고 신뢰 실행 환경에서 중요한 작업의 에뮬레이션을 시행하여 운영체제의 불변성을 보장하는 방식으로 운영체제 커널을 보호한다. TrustZone은 모바일 및 서버 환경에서의 적용 외에도 IoT 분야에서도 주목을 받고 있다.

[0006] TrustZone을 사용하면 시스템에서 신뢰 실행 환경의 이점을 얻을 수 있지만 TrustZone에 대한 타사 응용 프로그램 개발자 및 IoT 서비스 공급자의 액세스 가능성은 제한된다. 이는 신뢰 실행 환경이 SoC(System on a Chip) 제조업체 또는 신뢰 실행 환경 소프트웨어 플랫폼 제공업체에 속하기 때문이다. 신뢰 실행 환경은 일반적으로 멤버십을 기반으로 지원되는 유료 서비스이다. 신뢰 실행 환경 클라이언트는 특정 신뢰 실행 환경 공급업체에서 제공하는 신뢰 실행 환경 API(Application Programming Interfaces)를 사용하여 신뢰할 수 있는 애플리케이션(Trusted Application, TA)을 개발해야 한다. 또한, 이러한 TA는 악용 가능한 버그가 있는 새 TA의 배포를 방지하기 위해 신뢰 실행 환경 공급업체에서 엄격하게 확인한다. 이러한 신뢰 실행 환경 기능은 새롭거나 업데이트된 신뢰할 수 있는 IoT 소프트웨어의 개발 및 배포에 필요한 시간을 지연시키고 IoT 서비스의 출시 기간을 지연시킬 수 있다. 따라서, 이것은 타사 IoT 서비스 공급자가 TrustZone 사용을 꺼리는 주된 이유일 수 있다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 한국등록특허 제10-1894926호

발명의 내용

해결하려는 과제

[0009] 본 발명은 ARM과 같은 모바일 프로세서에서 제공하는 보안기술(TrustZone)을 활용하여 생성된 신뢰 실행 환경에 커널 모듈을 동적으로 적재함으로써 해당 모듈을 공격자로부터 보호하면서 안전하게 실행시키고자 한다.

[0010] 또한, 본 발명은 기존 신뢰 실행 환경을 확장시키지 않으며 신뢰 실행 환경에 새로운 공격 표면을 도입하지 않고 타사 소프트웨어의 안전한 실행을 보장하고자 한다.

[0011] 또한, 본 발명은 신뢰할 수 있는 애플리케이션을 개발하기 위해 별도의 API 사용을 최소화하고자 한다.

과제의 해결 수단

[0013] 상기 목적을 달성하기 위하여 본 발명은, 일반 실행 환경에서 차폐 커널 모듈의 실행을 트리거하기 위해 차폐된 커널 모듈에 관한 정보를 신뢰 실행 환경에 대한 압축 메시지로 생성하는 클라이언트 애플리케이션; 상기 클라이언트 애플리케이션에서 생성된 압축 메시지를 복사하여 신뢰 실행 환경으로 전송하는 TrustZone 커널 드라이버; 신뢰 실행 환경에서 차폐된 커널 모듈을 초기화하고 실행하기 위해, 압축 메시지를 수신하고 상기 차폐된 커널 모듈의 무결성을 확인하며 하기 차폐된 모듈 로더를 호출하는 신뢰할 수 있는 디스패처; 및 상기 디스패처의 호출에 의해 상기 차폐된 커널 모듈을 실행하는 차폐된 모듈 로더;를 포함하는 것을 일 특징으로 한다.

[0014] 바람직하게는, 상기 클라이언트 애플리케이션은, 차폐된 커널 모듈의 이미지를 파일 스트림으로 로드하고, 로드된 커널 모듈의 시작 주소, 모듈 크기, 진입점 및 서명된 해시를 신뢰 실행 환경의 API를 사용하여 신뢰 실행 환경에 대한 메시지로 압축하며, 프로세서 간 인터럽트(inter-processor interrupt, IPI)를 호출하여 다른 코어를 오프시킬 수 있다.

[0015] 바람직하게는, 상기 클라이언트 애플리케이션은, 신뢰 실행 환경에 압축 메시지가 전달될 수 있도록 상기 TrustZone 커널 드라이버를 오픈(open)시킬 수 있다.

[0016] 바람직하게는, 상기 디스패처는, 수신한 압축 메시지를 구문 분석하여 상기 차폐된 커널 모듈의 주소와 크기를 획득하고, 신뢰 실행 환경의 샌드박스 영역에 메모리를 할당하고 획득한 차폐된 커널 모듈의 크기 정보를 기반으로 상기 차폐된 커널 모듈의 이미지를 메모리가 할당된 샌드박스 영역에 복사할 수 있다.

[0017] 또한 본 발명은, 일반 실행 환경에서 사용자 애플리케이션이 커널 모듈을 신뢰 실행 환경으로 전송하고 차폐 실행을 요청하는 (a)단계; 신뢰 실행 환경에서 신뢰할 수 있는 디스패처가 상기 커널 모듈의 무결성을 확인하고 차폐된 모듈 로더를 호출하는 (b)단계; 및 상기 차폐된 모듈 로더가 상기 커널 모듈을 안전하게 실행하고 신뢰 실행 환경의 보안을 유지하도록 메모리 방화벽과 샌드박스를 구성하는 (c)단계;를 포함하는 것을 다른 특징으로 한다.

발명의 효과

[0019] 본 발명은 사용자 수준 응용 프로그램 보호에 중점이 있다.

[0020] 또한, 본 발명은 타사 IoT 서비스 공급자가 안전한 OS 권한 작업을 보장받을 수 있는 환경을 제공해 줄 수 있다는 이점이 있다.

[0021] 또한, 본 발명은 신뢰할 수 없는 일반 실행 환경의 OS에서 LKM과 같은 권한 있는 소프트웨어를 격리하여 사용자 애플리케이션 보호에 국한된 이전 TrustZone 기반 작업과 차별화된다.

[0022] 또한, 본 발명은 기존의 커널 모듈 개발 관행에 따라 보호 소프트웨어를 만들 수 있는 환경을 제공하여, 신뢰 실행 환경 전용 API를 사용할 필요가 없으므로 개발자의 개발 노력이 줄일 수 있다는 이점이 있다.

[0023] 또한, 본 발명은 신뢰 실행 환경의 권한이 필요한 다양한 긴급 장치 관리 작업은 IoT 서비스 공급자가 신뢰 실행 환경 공급업체의 개입을 기다리지 않고 즉석에서 수행할 수 있으므로 IoT 장치의 시장 출시 요구 사항을 방해할 수 있다는 문제를 해결할 수 있다.

도면의 간단한 설명

[0025] 도 1은 본 발명의 배경기술로 TrustZone을 가진 ARM 아키텍처를 나타낸다.

도 2는 본 발명의 실시예에 따른 모바일 기기의 동적커널모듈 보호 시스템의 구성도를 나타낸다.

도 3은 본 발명의 실시예에 따른 메모리 변환 설정을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하, 첨부된 도면들에 기재된 내용들을 참조하여 본 발명을 상세히 설명한다. 다만, 본 발명이 예시적 실시 예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일 참조부호는 실질적으로 동일한 기능을 수행하는 부재를 나타낸다.

[0027] 본 발명의 목적 및 효과는 하기의 설명에 의해서 자연스럽게 이해되거나 보다 분명해 질 수 있으며, 하기의 기재만으로 본 발명의 목적 및 효과가 제한되는 것은 아니다. 또한, 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이, 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다.

[0028] 도 1은 본 발명의 배경기술로 TrustZone을 가진 ARM 아키텍처를 나타낸다. 도 1을 참조하면, ARM TrustZone은 ARM 프로세서의 보안 확장으로, TrustZone 기술을 통해 시스템을 일반 실행 환경과 신뢰 실행 환경의 두 가지 실행 환경으로 나눌 수 있다. Linux 및 Android 기반과 같은 일반 소프트웨어 스택은 일반 실행 환경에서 호스팅된다. 반면에 신뢰 실행 환경은 암호화 서비스 및 DRM(Digital Rights Management)과 같은 보안이 중요한 애플리케이션을 보호한다.

[0029] 이러한 환경 간의 격리를 보장하기 위해 다양한 하드웨어 구성요소가 지원되는데, ARM 프로세서는 서로 다른 보안 상태(보안 및 비보안 상태)를 정의한다. 비보안 상태에서는 사용자, 커널 및 하이퍼바이저 모드가 지원된다. 보안 상태에서는 사용자, 커널 및 모니터 모드를 사용할 수 있다. 특히 모니터 모드와 관련하여 실행 중인 소프트웨어는 각 환경의 컨텍스트를 저장하고 복원하여 환경 간 전환을 관리한다. 프로세서의 보안 상태 외에도 각 환경에 맞게 DRAM을 분할하기 위해 TZASC가 제공된다. TZASC를 사용하여 프로세서의 보안 상태에 따라 여러 물리적 메모리 영역과 해당 액세스 권한을 정의할 수 있다. 예를 들어, 물리적 메모리의 특정 범위는 신뢰 실행 환경(보안 상태)에서는 액세스를 허용하지만, 일반 실행 환경(비보안 상태)에서는 액세스를 허용하지 않음으로써 신뢰 실행 환경의 영역으로 구성할 수 있다. 보안 상태 및 메모리 액세스 제어 외에도 TrustZone은 디스플레이 및 키패드와 같은 입/출력(IO) 주변 장치도 보호한다. TZPC(TrustZone Protection Controller)는 이러한 주변 장치를 신뢰 실행 환경에 동적으로 연결하여 안전한 IO 채널을 생성할 수 있도록 하여 손상된 일반 실행 환경으로 인한 간섭을 방지할 수 있다.

[0030] ARM 프로세서는 가상 주소를 물리적 주소로 변환하는 가상 메모리 시스템을 지원한다. 이러한 변환은 TTBR(Translation Table Base Registers)을 사용하여 기본 주소가 구성된 페이지 테이블을 기반으로 수행된다. 일반적으로 TTBR0 및 TTBR1의 두 가지 아키텍처 지원 TTBR이 있다. 64비트 시스템에서 TTBR0 및 TTBR1은 일반적으로 각각 사용자 및 커널 공간에 대한 페이지 테이블 기본 주소를 포함한다. 32비트 시스템에서는 작은 주소 범위, 즉 4GB로 인해 하나의 페이지 테이블만 사용할 수 있다. 예를 들어 사용자 및 커널 공간을 포함하는 32비트 Linux 기반 주소 공간의 경우 TTBCR(Translation Table Base Control Register)의 설정에 따라 TTBR 중 하나가 활용되는지 또는 두 TTBR이 모두 활용되는지 여부가 결정된다.

[0031] 짧은 설명자 형식을 사용하는 경우 TTBCR의 N 플래그는 TTBR0 및 TTBR1을 사용하여 가리키는 페이지 테이블을 통해 수행되는 주소 변환의 크기를 결정한다. 예를 들어 N값이 0b000이면 TTBR0만 사용됩니다. 그렇지 않으면 TTBR0과 TTBR1이 모두 활용되고 각 TTBR을 사용하여 변환된 주소 범위는 N값으로 결정된다. 예를 들어 N값을 0b001로 설정하면 가상 메모리의 2GB 미만과 2GB 초과 주소는 각각 TTBR0과 TTBR1을 사용하여 변환된다. TTBR 및 TTBCR은 프로세서의 비보안 및 보안 상태를 위한 뱅크 레지스터이다. 따라서 TTBR 및 TTBCR은 일반 실행 환경과 신뢰 실행 환경에서 독립적으로 구성될 수 있다.

[0032] 도 2는 본 발명의 실시예에 따른 모바일 기기의 동적커널모듈 보호 시스템(10)의 구성도를 나타낸다. 도 2를 참조하면, 모바일 기기의 동적커널모듈 보호 시스템(10)은 클라이언트 애플리케이션(100), TrustZone 커널 드라이버(200), 신뢰할 수 있는 디스패처(300), 차폐된 모듈 로더(400), 메모리 방화벽(500), 및 샌드박스(600)를 포

함할 수 있다.

[0033] 모바일 기기의 동적커널모듈 보호 시스템(10)은 유저 어플리케이션을 신뢰 실행 환경에 적재하는 기존의 시스템과는 달리 커널 모듈을 신뢰 실행 환경에 동적으로 적재하여 실행할 수 있다. 동적커널모듈 보호 시스템(10)은 ARM과 같은 모바일 프로세서에서 제공하는 보안기술(TrustZone)을 활용하여 생성된 신뢰 실행 환경에 커널 모듈을 동적으로 적재함으로써 해당 모듈을 공격자로부터 보호하면서 안전하게 실행할 수 있다. 동적커널모듈 보호 시스템(10)은 페이지 테이블을 신뢰 실행 환경 내에 동적으로 로딩된 커널 모듈이 공유하여 사용하도록 하여, 커널 모듈이 운영체제와 격리된 영역인 신뢰 실행 환경에서 실행됨으로써 메모리나 파일 시스템 접근 시의 성능 저하가 발생하는 것을 방지할 수 있다.

[0034] 동적커널모듈 보호 시스템(10)은 신뢰 실행 환경의 단점을 최소화하기 위해 제3자가 TrustZone 기반의 신뢰 실행 환경을 쉽게 사용할 수 있도록 해주는 신뢰 실행 환경 플랫폼을 제공할 수 있다. 동적커널모듈 보호 시스템(10)은 신뢰 실행 환경을 사용하여 LKM(Loadable Kernel Module)과 같은 커널 권한 소프트웨어를 보호할 수 있다.

[0035] 신뢰 실행 환경 기반 커널 무결성 모니터링에 의해 무결성이 보호되는 OS에 대한 모든 업데이트는 일반적으로 신뢰 실행 환경 공급업체가 업데이트를 확인하고 OS 이미지에 적용해야 한다. 그러나, 동적커널모듈 보호 시스템(10)은 신뢰 실행 환경 공급업체가 확인을 완료하고 변경 사항을 OS 이미지에 적용할 때까지 기다리지 않고 제3자가 메모리의 OS 버그를 즉시 수정하는 것과 같은 긴급 장치 관리 작업을 수행할 수 있는 환경을 제공할 수 있다.

[0036] 동적커널모듈 보호 시스템(10)은 일반 실행 환경 구성요소와 신뢰 실행 환경 구성요소로 구성될 수 있다. 일반 실행 환경 구성요소는 클라이언트 애플리케이션(100)과 TrustZone 커널 드라이버(200)를 포함할 수 있다. 신뢰 실행 환경 구성요소는 신뢰할 수 있는 디스패처(300), 차폐된 모듈 로더(400), 메모리 방화벽(500), 및 샌드박스(600)를 포함할 수 있다.

[0037] 일반 실행 환경은 신뢰 실행 환경의 구성요소와 통신하여 타사에서 제공하는 커널 모듈의 보안 실행 요청을 보내는 간단한 로더가 구현될 수 있다. 신뢰 실행 환경의 구성요소는 일반 실행 환경의 요청을 처리하기 위해 먼저 비대칭 암호화 알고리즘을 기반으로 수신된 커널 모듈의 서명과 무결성을 확인할 수 있다. 수신된 커널 모듈은 신뢰 실행 환경의 보호 아래 실행될 수 있다.

[0038] 동적커널모듈 보호 시스템(10)은 신뢰 실행 영역의 메모리 영역에 대한 필터링 액세스를 가능하게 하는 TZASC(TrustZone Address Space Controller)를 활용하여 커널 모듈의 실행을 보호할 수 있다. 동적커널모듈 보호 시스템(10)은 하드웨어 디버깅 감시점을 설정하여 차폐된 커널 모듈에 의한 신뢰 실행 환경의 메모리 영역에 대한 모든 액세스를 모니터링하여, 일반 실행 환경의 OS 대신 신뢰 실행 환경의 메모리를 잘못 배치하는 등 기존 신뢰 실행 환경의 플랫폼 보안에 영향을 미칠 수 있는 프로그래밍 오류를 고려할 수 있다.

[0039] 동적커널모듈 보호 시스템(10)은 신뢰할 수 있는 신뢰 실행 환경의 클라이언트에 의해서만 사용하도록 하기 위해 DRM과 같은 모바일 장치 서비스를 보호하는 데 널리 사용되는 공개키 인프라(Public Key Infrastructure, PKI)에 의존할 수 있다. 즉, 동적커널모듈 보호 시스템(10)을 이용하는 클라이언트는 자신의 공개 키를 신뢰 실행 환경의 공급업체에 등록해야 한다. 신뢰 실행 환경의 공급업체는 공개키의 해시를 신뢰 실행 환경에 저장한다. 클라이언트가 제공한 커널 모듈을 실행하기 위해, 모듈 개발자는 먼저 등록된 공개키 쌍인 개인키를 사용하여 모듈의 해시에 서명한다. 동적커널모듈 보호 시스템(10)은 커널 모듈이 로드되기 전에 먼저 신뢰 실행 환경에 저장된 해시를 사용하여 공개키의 무결성을 확인할 수 있다. 이어서, 동적커널모듈 보호 시스템(10)은 공개키를 사용하여 새로 계산된 해시와 복호화된 해시를 비교하여 커널 모듈의 무결성을 확인할 수 있다.

[0040] 동적커널모듈 보호 시스템(10)은 IoT 서비스 공급자(신뢰 실행 환경의 클라이언트)가 필요에 따라 커널 모듈을 배포하고 신뢰 실행 환경의 보안을 방해하지 않고 TrustZone 보호 아래에서 안전하게 실행되게 할 수 있다. 동적커널모듈 보호 시스템(10)은 응용 프로그램의 보안에 중요한 논리의 확인 및 분리가 필요한 TA를 구축할 때 개발자의 노력을 최소화시킬 수 있다. 동적커널모듈 보호 시스템(10)은 특정한 신뢰 실행 환경의 공급업체에서 제공하는 API를 사용하지 않고 구현될 수 있다.

[0041] 개발자는 IoT 서비스 및 장치 유지 관리를 위해 차폐된 커널 모듈로 실행되는 커널 모듈을 생성할 수 있으므로 동적커널모듈 보호 시스템(10)의 이점을 누릴 수 있다. 예를 들어, 커널 모듈은 메모리의 커널 텍스트 영역을 긴급하게 패치하여 전체 커널 이미지를 교체하고 장치를 재부팅하지 않고도 버그를 수정할 수 있다. 일반적으로 신뢰 실행 환경 기반의 메모리 보호 솔루션이 제공하는 보호로 인해 텍스트 영역이 불변으로 강제되는 조건 아

래에서 이 패치 작업은 보호된 메모리를 업데이트하기 위해 신뢰 실행 환경의 공급업체 개입이 필요할 수 있다.

[0042] 동적커널모듈 보호 시스템(10)은 신뢰 실행 환경의 권한으로 클라이언트 모듈을 실행할 수 있어서 신뢰 실행 환경의 공급업체 개입이 필요하지 않다. 또한, 동적커널모듈 보호 시스템(10)에서 커널 모듈은 실행 중에 신뢰할 수 없는 OS로부터 격리되고 보호될 수 있다. 따라서, 동적커널모듈 보호 시스템(10)은 포렌식 분석을 위한 일반 실행 환경의 메모리 덤프와 같은 메모리 검사 작업을 공격자의 간섭 없이 안정적으로 수행할 수 있다.

[0043] 일반 실행 환경과 신뢰 실행 환경은 서로 다른 API를 사용하므로, 동적커널모듈 보호 시스템(10)은 개발자의 개발 노력과 개발의 복잡성을 최소화하는데 기여할 수 있다. 이하에서는 동적커널모듈 보호 시스템(10)의 일반 실행 환경의 구성요소를 설명한다.

[0044] 차폐된 커널 모듈은 커널 데이터 구조 검사와 같은 작업을 허용하는 일반 Linux 커널 기호를 사용하여 개발될 수 있다. 차폐된 커널 모듈은 독립적이며 사용자 프로세스와 상호 작용하지 않고 실행될 수 있다. 차폐된 커널 모듈은 copy_from_user 및 copy_to_user와 같은 커널 API를 사용하지 않을 수 있다. 차폐된 커널 모듈은 신뢰 실행 영역에서 무결성 검증되는데, 이를 위해 SHA256을 기반으로 모듈의 해시를 얻고 개발자의 개인 키를 사용하여 서명될 수 있다. 서명된 해시는 차폐된 커널 모듈의 실행 요청과 함께 신뢰 실행 환경으로 전달될 수 있다.

[0045] 클라이언트 애플리케이션(100)은 일반 실행 환경에서 차폐 커널 모듈의 실행을 트리거하기 위해 차폐된 커널 모듈에 관한 정보를 신뢰 실행 환경에 대한 압축 메시지로 생성할 수 있다. 클라이언트 애플리케이션(100)은 차폐된 커널 모듈 이미지의 경로, 진입점(함수 이름) 및 서명된 해시의 세 가지 입력 인수를 처리할 수 있다.

[0046] 클라이언트 애플리케이션(100)은 차폐된 커널 모듈의 이미지를 파일 스트림으로 로드하고, 로드된 커널 모듈의 시작 주소, 모듈 크기, 진입점 및 서명된 해시를 신뢰 실행 환경의 API를 사용하여 신뢰 실행 환경에 대한 메시지로 압축하며, 프로세서 간 인터럽트(inter-processor interrupt, IPI)를 호출하여 다른 코어를 오프(off) 시킬 수 있다.

[0047] 클라이언트 애플리케이션(100)은 다른 코어를 오프(off)함으로써 여러 코어 간의 경쟁 조건을 악용하는 TOCTTOU(time-of-check-totime-of-use) 공격을 방지할 수 있다. 특히, 클라이언트 애플리케이션(100)은 잠재적으로 악의적인 코어가 공격자의 발자국을 숨기는 것을 방지하여, 일반 실행 환경의 메모리 검사와 같은 커널 모듈의 작업을 안정적으로 수행할 수 있다.

[0048] 클라이언트 애플리케이션(100)은 신뢰 실행 환경에 압축 메시지가 전달될 수 있도록 상기 TrustZone 커널 드라이버를 오픈(open)시킬 수 있다.

[0049] TrustZone 커널 드라이버(200)는 클라이언트 애플리케이션에서 생성된 압축 메시지를 복사하여 신뢰 실행 환경으로 전송할 수 있다. TrustZone 커널 드라이버(200)는 일반 실행 환경에 상주하므로 손상될 수 있다. 따라서, TrustZone 커널 드라이버(200)는 차폐된 커널 모듈의 실행 요청, 즉 서비스 거부(DoS) 공격을 무시할 수 있다.

[0050] 신뢰 실행 환경은 차폐된 커널 모듈의 실행 요청을 처리할 수 있다. 신뢰 실행 환경에서는 차폐된 커널 모듈의 초기화, 메모리 방화벽 구성, 샌드박스(600) 활성화 및 모듈 디스패칭이 수행될 수 있다. 이하에서는 동적커널모듈 보호 시스템(10)의 신뢰 실행 환경의 구성요소를 설명한다.

[0051] 신뢰할 수 있는 디스패처(300)는 신뢰 실행 환경에서 차폐된 커널 모듈을 초기화하고 실행하기 위해, 압축 메시지를 수신하고 상기 차폐된 커널 모듈의 무결성을 확인하며 하기 차폐된 모듈 로더(400)를 호출할 수 있다. 신뢰할 수 있는 디스패처(300)는 일반 실행 환경의 클라이언트 애플리케이션(100)에 대응되는 구성이다.

[0052] 신뢰할 수 있는 디스패처(300)는 복사된 차폐된 커널 모듈의 해시를 클라이언트 애플리케이션(100)의 서명된 해시를 사용하여 확인할 수 있다. 신뢰할 수 있는 디스패처(300)는 해시를 확인한 후 차폐된 커널 모듈을 초기화하고 실행하기 위해 차폐된 모듈 로더를 호출할 수 있다. 신뢰할 수 있는 디스패처(300)는 샌드박스(600) 및 진입점에 있는 차폐 커널 모듈의 주를 차폐 모듈 로더로 전달할 수 있다.

[0053] 신뢰할 수 있는 디스패처(300)는 수신한 압축 메시지를 구문 분석하여 상기 차폐된 커널 모듈의 주소와 크기를 획득하고, 신뢰 실행 환경의 샌드박스(600) 영역에 메모리를 할당하고 획득한 차폐된 커널 모듈의 크기 정보를 기반으로 상기 차폐된 커널 모듈의 이미지를 메모리가 할당된 샌드박스(600) 영역에 복사할 수 있다.

[0054] 차폐된 모듈 로더(400)는 신뢰할 수 있는 디스패처(300)의 호출에 의해 차폐된 커널 모듈을 실행할 수 있다. 차폐된 모듈 로더(400)는 ELF 로더의 변형 버전으로 차폐된 커널 모듈을 적절하게 실행하기 위해 ELF 헤더 파싱,

커널 심볼 연결, 각 메모리 세그먼트 로딩 등의 일반적인 동작을 수행할 수 있다.

[0055] 차폐된 커널 모듈이 신뢰 실행 환경에서 실행되기 때문에 공격자는 모듈 실행에 직접 영향을 미칠 수 없으나, 공격자는 적시에 대상 검사 개체의 상태를 조작하여 차폐된 커널 모듈의 실행 결과에 여전히 영향을 미칠 수 있다. 이 문제를 해결하기 위해 클라이언트 애플리케이션(100)은 차폐된 커널 모듈을 실행하는 코아를 제외한 모든 코어를 일시적으로 중지할 수 있다. 그러나 다른 코어를 중지하기 위한 IPI는 공격자가 무시할 수 있는 커널 API의 실행에 의존하기 때문에 이 접근 방식은 위협을 안정적으로 방지할 수 없다.

[0056] 메모리 방화벽(500)은 악성 코어가 차폐된 커널 모듈의 결과를 방해하지 않도록 TZASC를 활용할 수 있다. TZASC는 CPU의 보안 상태를 기반으로 물리적 메모리 영역에서 액세스 제어를 수행할 수 있다. 메모리 방화벽(500)은 일반 실행 환경에 할당된 전체 물리적 영역이 비보안 상태의 코어에서 액세스할 수 없도록 차폐된 커널 모듈을 실행하기 전에 TZASC를 동적으로 재구성할 수 있다. 메모리 방화벽(500)은 공격자가 IPI 중지 요청을 차단하더라도 비보안 상태에서 실행되는 악성 코어가 차폐된 커널 모듈 및 커널 API에 액세스할 수 없도록 할 수 있다. 차폐된 커널 드라이버를 실행하는 코어는 보안 상태에 있기 때문에 여전히 전체 물리적 메모리에 액세스할 수 있으므로 모듈을 원활하게 실행할 수 있다. TZASC의 이러한 동적 구성은 일반 실행 환경 및 신뢰 실행 환경에 대한 물리적 메모리 영역이 부팅 시 이미 정의되고 구성되므로 영역을 재정의할 필요가 없다. 메모리 방화벽(500)은 TZASC의 REGION_ID_ACCESS_<n> 레지스터를 재설정하여 비보안 상태의 코어에서 일반 실행 환경의 영역에 대한 읽기 및 쓰기 액세스를 차단할 수 있다.

[0057] 차폐된 드라이버의 취약성 또는 프로그래밍 오류로 인한 신뢰 실행 환경에 대한 의도하지 않은 액세스는 공격자가 이러한 측면을 악용하여 신뢰 실행 환경을 손상시킬 수 있다. 예를 들어 공격자는 차폐된 커널 모듈을 악용하여 신뢰 실행 환경의 내부 작동을 학습할 수 있다. 신뢰 실행 환경의 OS와 펌웨어 이미지는 일반적으로 암호화된 바이너리로 제공되기 때문에 정적 분석을 통해서는 일반적으로 수행할 수 없다. 그러나 런타임 시 신뢰 실행 환경에서 복호화되기 때문에 공격자는 차폐된 커널 모듈 권한을 남용하여 신뢰 실행 환경의 유익한 정보를 유추할 수 있다.

[0058] 샌드박스(600)는 신뢰 실행 환경 내에 적재된 커널 모듈이 취약점을 수반할 경우, 공격자는 해당 취약점을 공격하여 신뢰 실행 환경 전체를 장악할 수 있는 문제를 해결하기 위해 신뢰 실행 환경 내에서 동적으로 로딩된 커널 모듈이 실행되는 영역을 분리(격리)할 수 있다.

[0059] 샌드박스(600)는 감시점 기반 메모리 격리 기술로 구축될 수 있다. 샌드박스(600)를 구축하는 방법은 다음과 같다. 신뢰 실행 환경의 메모리 일부를 샌드박스 영역으로 예약한다. 다음으로 차폐된 커널 모듈의 초기화 단계에서 예약된 영역은 모듈 할당에 활용된다. 샌드박스(600)에서 모듈을 실행하기 전에 다른 신뢰 실행 환경의 소프트웨어가 상주하는 샌드박스 외부의 모든 액세스를 모니터링하도록 감시점을 구성한다.

[0060] 차폐된 커널 모듈이 실행을 완료하면 신뢰 실행 환경의 영역에 다시 액세스할 수 있도록 감시점이 재구성된다. 감시점 구성 지침은 샌드박스(600)에서 액세스할 수 있기 때문에 공격자는 이를 악용하여 구성을 조작하려고 할 수 있다. 이를 방지하기 위해 구성의 정확성을 확인하는 검증 로직을 통해 감시점 구성 지침이 준수되는지 확인한다.

[0061] 감시점 기반 보호에도 불구하고 샌드박스(600) 영역 외부는 여전히 실행 가능하다. 따라서 공격자는 유용한 신뢰 실행 환경의 라이브러리를 활용하고 성공적인 공격을 달성하기 위해 제어 흐름을 샌드박스 처리되지 않은 영역으로 전환하려고 시도할 수 있으나, 신뢰 실행 환경의 내부 레이아웃을 동적으로 또는 정적으로 분석할 수 없어 매우 어려울 것이다. 특히 암호화된 신뢰 실행 환경의 소프트웨어는 정적 분석을 방지한다. 또한, 감시점 기반 샌드박스(600)는 신뢰 실행 환경의 레이아웃에서 런타임 정보 유출을 방해한다. 결과적으로 공격자는 신뢰 실행 환경에서 유용한 코드 청크의 위치를 정확히 찾아낼 수 없다.

[0062] 본 발명의 다른 실시예인 모바일 기기의 동적커널모듈 보호 방법은 일반 실행 환경에서 사용자 애플리케이션이 커널 모듈을 신뢰 실행 환경으로 전송하고 차폐 실행을 요청하는 (a)단계, 신뢰 실행 환경에서 신뢰할 수 있는 디스패처가 커널 모듈의 무결성을 확인하고 차폐된 모듈 로더를 호출하는 (b)단계, 및 차폐된 모듈 로더가 커널 모듈을 안전하게 실행하고 신뢰 실행 환경의 보안을 유지하도록 메모리 방화벽과 샌드박스를 구성하는 (c)단계를 포함할 수 있다.

[0063] 이상에서 대표적인 실시예를 통하여 본 발명을 상세하게 설명하였으나, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 상술한 실시예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리 범위는 설명한 실시예에 국한되어 정해져서는 안 되며, 후술하는

특허청구범위뿐만 아니라 특허청구범위와 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태에 의하여 정해져야 한다.

부호의 설명

[0065] 10 : 모바일 기기의 동적커널모듈 보호 시스템

100 : 클라이언트 애플리케이션

200 : TrustZone 커널 드라이버

300 : 신뢰할 수 있는 디스패처

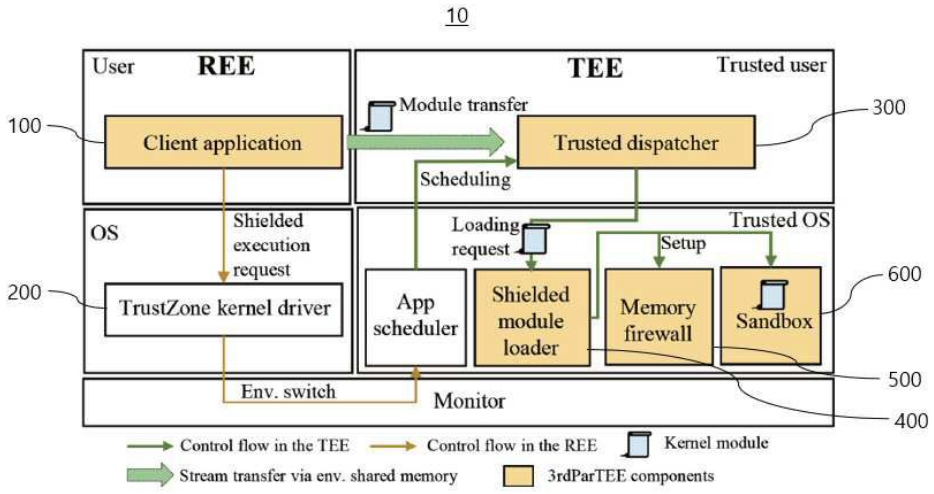
400 : 차폐된 모듈 로더

500 : 메모리 방화벽

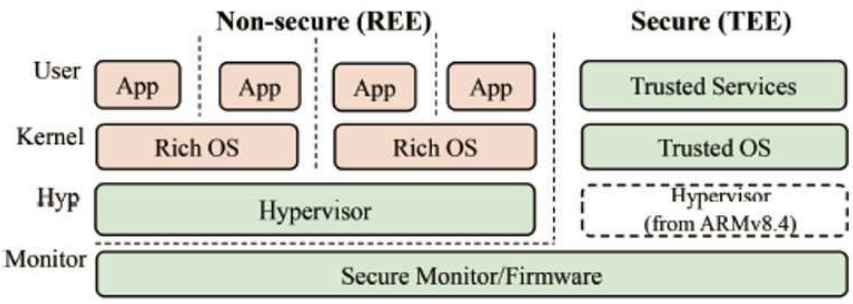
600 : 샌드박스

도면

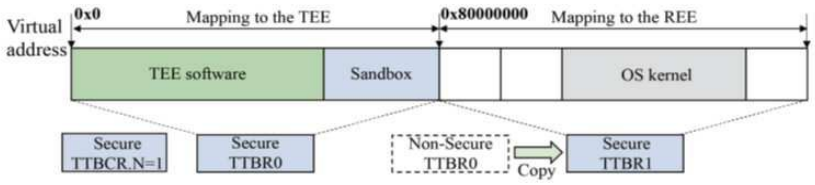
도면1



도면2



도면3


Virtual address
0x0
Mapping to the TEE
0x80000000
Mapping to the REE
TEE software
Sandbox
OS kernel
Secure TTBCR.N=1
Secure TTBR0
Non-Secure TTBR0
Copy
Secure TTBR1