(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.
G06F 9/24 (2006.01)
G06F 9/445 (2006.01)

(11) 공개번호 10-2006-0047897
(43) 공개일자 2006년05월18일

(21) 출원번호 10-2005-0040209
(22) 출원일자 2005년05월13일

(30) 우선권주장 10/882,134 2004년06월30일 미국(US)

(71) 출원인 마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이

(72) 발명자 윌먼, 브라이언 마크
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
잉글랜드, 폴
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
래이, 케네스 디.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
헌터, 제이미
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
맥마이클, 로니 딘
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
라살, 데렉 노만
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
자코메, 피에르
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
팔리, 마크 엘리엇
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
쿠리엔, 데크딸라칼 바루기스
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
크로스, 데이비드 비.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내

(74) 대리인 주성민
백만기
이중희

심사청구 : 없음

(54) 상태 검증을 사용하여 보호된 오퍼레이팅 시스템 부팅을위한 시스템 및 방법

요약

악성 컴포넌트가 오퍼레이팅 시스템과 함께 로딩되는 것을 방지하여, 부적절한 환경 하에서 시스템 키의 누설을 방지하는 보호된 오퍼레이팅 시스템 부팅에 대한 메커니즘이 제공된다. 기계 구동 프로시저의 일부분이 발생한 후, 오퍼레이팅 시스템 로더가 실행되고, 로더가 검증되고, 올바른 기계 상태가 존재하는 것으로 검증되고/검증되거나 생성된다. 로더가 합법적인 로더인 것으로 검증되고, 로더가 실행되는 기계 상태가 올바른 것으로 검증되면, 로더의 미래의 동작은 시스템 키의 누설을 야기할 수 있는 악성 컴포넌트의 로딩으로부터 보호하는 것으로 알려진다. 로더의 동작이 시스템 키에 대해 안전한 것으로 알려지면, 검증기는 그 시스템 키를 봉인해제할 수 있고 그것을 로더에 제공한다.

대표도

도 4

색인어

오퍼레이팅 시스템 부팅, 오퍼레이팅 시스템 로더, 로딩

명세서

도면의 간단한 설명

도 1은 본 발명 양태가 구현될 수 있는 예시적인 컴퓨팅 환경의 블록도.

도 2는 그 올바른 동작이 시스템 키에 의존하는 프로세스를 채용하는 시스템의 블록도.

도 3은 암호화가 시스템 키에 의존하게 함으로써, 암호화된 데이터를 인증되지 않은 복호화로부터 보호하는 암호화된 파일 시스템의 블록도.

도 4는 본 발명의 양태에 따른 검증을 갖는 예시적인 부팅 프로세스의 흐름도.

도 5는 본 발명의 양태에 따른 예시적인 검증기의 블록도.

도 6은 본 발명의 양태에 따라 시스템 키를 보호하는 예시적인 프로세스의 흐름도.

<도면의 주요 부호에 대한 설명>

141 : 하드 디스크

202 : 프로세스

204 : 키

206 : 키 보호 메커니즘

302 : 파일 암호화 시스템(EFS)

304 : 파일

306 : 암호화된 파일

308 : 암호화된 파일

310 : 컨텐츠 키

312 : 키 생성기 모듈

402 : BIOS

404 : 옵션 ROM

406 : MBR

410 : OS 로더

412 : 단계 1

414 : 단계 2

418 : 로그온 프로그램

416 : HAL, OS 커널

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 일반적으로 컴퓨팅 분야에 관한 것이다. 특히, 본 발명은 시스템이 공지의 안전 상태로부터 진행하는 것을 보장하는 메커니즘을 제공하고, 이 메커니즘은 시스템의 올바른 행위에 대한 충분한 보증을 제공하는 방식으로 시스템을 부팅하는 데 사용될 수 있다. 이 올바른 행위의 보증은 하나 이상의 키가 부적절한 환경 하에서 배포되는 것을 방지할 수 있다.

컴퓨터 보안은 종종, 소프트웨어 컴포넌트의 행위를 예측할 수 있는 것에 의존한다. 일반적으로, 시스템의 보안은, 그 행위가 이해되고 공지의 양호 상태로부터 발생한 공지의 프로그램이 예측가능한 방식으로 작용할 것이라는 전제로부터 일어날 수 있다. 반대로, 컴퓨터 시스템이 설계자의 계획 밖의 방식으로 동작하게 하는 것을 포함할 수 있는 보안의 방해는, 일반적으로 공지의 프로그램을 대체 또는 변경하거나 그 행위가 이해되지 않는 상태에서 그것을 실행함으로써, 실현될 수 있다. 따라서, 컴퓨팅 환경에 보안을 제공하는 일 양태는, 공지의 프로그램이 사용되고 그것이 공지의 양호 상태로부터 발생한 것을 검증하는 것을 포함한다.

행위의 예측가능성이 특히 중요한 하나의 영역은, 오퍼레이팅 시스템 및 그 컴포넌트를 로딩할 때이다. 오퍼레이팅 시스템 자체가 자신의 행위에 관하여 임의의 신뢰가치성 레벨을 제공하도록 설계되어 있더라도, 오퍼레이팅 시스템을 공격으로부터 보호하는 인프라구조가 아직 설립되지 않았을 수 있기 때문에(또는, 설립되는 과정 중일 수 있기 때문에), 그러한 오퍼레이팅 시스템이 로딩되기 전의 시간은 시스템이 공격에 특히 취약해지는 시간이다. 따라서, 오퍼레이팅 시스템이 예측가능한 방법으로 로딩하는 것을 보장하는 것은 오퍼레이팅 시스템을 특정 부류의 공격으로부터 보호하는 데 중요하다.

오퍼레이팅 시스템의 안전하지 않은 로딩으로부터 일어날 수 있는 하나의 유형의 보안 침해는, 특정 제한된 기능을 가능하게 하는 키(또는 키들)의 보호에 관련된다. 예를 들어, 마이크로소프트 윈도우즈 오퍼레이팅 시스템은, 프로세스의 올바른

수행이 SYSKEY의 사용가능성에 의존하게 함으로써 여러 프로세스를 보호하는 데 사용되는 시스템 키 또는 "SYSKEY"를 채용하지만, 이것으로 제한되는 것은 아니다. 예를 들어, 암호화된 형태로 오퍼레이팅 시스템에 의해 저장된 개인 정보를 복호화하는 데 필요한 키는 SYSKEY로부터 유도될 수 있다.

전통적으로, 제한된 동작을 수행하는 데 필요한 키는 로그온 프로시저에 의해 보호된다. 일반적으로, 사용자는 시스템 사용을 개시하기 전에, (예를 들어, 사용자 이름/패스워드 조합과 같은 올바른 로그온 자격 증명을 제공함으로써) 스스로를 올바르게 인증해야 한다. 키의 사용은 사용자가 올바르게 인증할 경우에만 가능해지고, 시스템은 사용자가 적절한 로그온에 실패한 것으로 결정하기 전에 사용자에게 제한된 시도 횟수(예를 들어, 세번)만을 허용할 것이다. 로그온 시도의 횟수에 대한 이러한 유형의 제한은, 비인증된 사용자가 예를 들어, 도난된 랩탑 컴퓨터의 경우에 패스워드를 알아맞추는 강제 공격을 사용함으로써 보호된 기능을 사용하는 것을 방지한다. 그러나, 로그온 프로시저를 사용하여 키로의 액세스를 보호하는 것은, 오퍼레이팅 시스템 로더가 올바른 로그온 프로그램으로 오퍼레이팅 시스템을 올바르게 로딩했고, 실행될 수 있는 악성 코드에 의한 키의 사용이 가능해질 수 없었음을 가정한다. 악성 로더가 대신에 사용되어 악성 로그온 프로그램이 오퍼레이팅 시스템과 함께 로딩되게 하는 경우, 키의 사용이 가능해질 수도 있고 심지어 올바른 자격증명이 입력되지 않고서 키가 누설될 수도 있다. 오퍼레이팅 시스템의 로딩은 보안 침해의 기회를 제공하기 때문에, 그러한 상황에서의 키의 보호는, 오퍼레이팅 시스템의 로딩이 올바르게 발생하는 것이 검증될 수 있는 환경 하에서 오퍼레이팅 시스템의 로딩이 일어나기를 요구한다.

발명이 이루고자 하는 기술적 과제

오퍼레이팅 시스템 로드 프로세스의 보안을 검증할 때 일어나는 하나의 문제는, 적법한 오퍼레이팅 시스템 로드가 다수의 상이한 프로그램들(예를 들면, 시스템 부팅 프로시저 동안 실행하는 pre-OS 프로그램인 수많은 상이한 "옵션 ROM")을 호출할 수 있고, 오퍼레이팅 시스템 로드의 일부분으로서 수행될 수 있는 다수의 상이한 프로시저들이 있다는 것이다. 따라서, 로드 동안 거의 무수한 수의 상이한 적법한 기계 상태들이 있으며, 그러한 모든 상태를 식별하고 그 기계가 공지의 양호 상태에 있음을 검증하는 것은 실행 불가능한 태스크인 것으로 판명될 수 있다. 그러나, 로드 프로시저의 모든 부분이 보안 의미를 갖는 것은 아니다. 그 보안을 평가하려는 임의의 시도 없이 로드가 진행되게 한 다음, 키의 배포와 같은 보안-관련 기능에 영향을 줄 수 있는 임의의 프로시저를 개시하기 전에 공지의 양호 상태로 환경을 설정하는 것이 더 효율적일 수 있다. 더 일반적으로, 임의의 시스템은 보안 의미가 발생하게 하는 임의의 동작을 허용하기 전에, 공지의 양호 상태로 설정될 수 있는 한, 임의의 유형의 보안 평가 없이 임의의 시간동안 실행하도록 허용될 수 있다.

상술된 관점에서, 종래의 기술분야의 취약점을 극복하는 메커니즘이 필요하다.

발명의 구성 및 작용

요약

본 발명은, 로드가 올바르게 수행되는 것이 검증될 수 있는 환경 하에서의 오퍼레이팅 시스템의 로딩을 제공한다. 기계가 개시될 때, 초기 개시 프로시저(예를 들면, BIOS, 옵션 ROM, 마스터 부팅 레코드, 부팅 섹터 등)가 수행된다. 이 초기 프로시저가 수행된 후, 오퍼레이팅 시스템 로더가 개시되고 다양한 예비 태스크를 수행할 수 있다. 오퍼레이팅 시스템 로더가 개시되고 그러한 예비 태스크를 수행한 후, 오퍼레이팅 시스템 로더의 검증이 수행된다.

이 검증은 로더 자체 또는 로더의 일부분에 대한 검증 테스트(예를 들면, 검사합, 또는 로더의 아이덴티티와 정확도를 평가하기 위한 다른 테스트)를 수행할 뿐만 아니라, 현재 기계 상태를 평가한다(또는, 기계가 공지의 양호 상태에 부합하게 함). 로더(또는, 관련 일부분)가 올바른 것으로 알려지고 기계 상태가 로더가 올바르게 동작하는 것으로 이전에 검증된 것이면, 로더의 행위는 예측될 수 있다. 따라서, 올바른 기계 상태 하의 올바른 로더 동작은 보호된 기능(예를 들어, 시스템 키와 같은 암호 키)을 가능하게 하는 데이터가 부적절한 환경 하에서 배포되게 하는 컴포넌트를 로딩하지 않을 것임이 보증될 수 있다.

바람직하게는, 검증은 고-신뢰도 환경(high-assurance environment)에서 실행되는 검증기에 의해 수행된다. 높은 보증 환경은, 그 안에서 수행되는 프로세스가 예상되는 방법으로 실행될 것이라는 상대적으로 높은 수준의 보증을 제공하는 것이 가능한 환경이다. 따라서, 검증기가 올바르게 동작할 것이라는 보증은, 검증기가 고-신뢰도 환경에서 실행되는 프로세스에 의해 검증된다는 사실로부터, 및 고-신뢰도 환경 내의 프로세스가 올바르게 수행될 것이라는 기저 사실로부터 유도된다. 또는, 적어도, 고-신뢰도 환경이 그러한 환경 내에서 동작하는 프로세스의 올바른 동작에 간섭(간섭 허용)하지 않을 것이라는 보증 레벨이 있다. 그래도, 사람은 고-신뢰도 환경 내에서 프로세스를 구현하는 프로그램이 수행이 예상되는 방식으로 올바르게 수행될 것이라는 별도의 신뢰 기반을 가져야 한다. 고-신뢰도 환경은, 데이터가 특정 오브젝트에 봉인될

수 있는 저장 장치로서, 고-신뢰도 환경이 봉인된 데이터가 (고-신뢰도 환경에 의해 검증된 것과 같이) 데이터가 봉인된 오브젝트 외의 임의의 오브젝트에 배포되지 않는 방식으로 제어하는 봉인된 저장 장치를 제공할 수 있다. 검증기는 이 봉인된 저장 장치를 사용하여 키(예를 들어, SYSKEY)를 그 자체에 봉인할 수 있고, 임의의 표준 하에서 환경(예를 들어, 기계 상태)이 만족스러울 때 올바른 오브젝트 이외의 키를 봉인해제하는 것을 거부할 수 있다.

본 발명의 다른 특징이 이하에서 설명된다.

상술된 요약 뿐만 아니라 다음의 바람직한 실시예의 상세한 설명은 첨부된 도면과 함께 읽을 때 더 잘 이해된다. 본 발명을 설명하기 위하여, 도면에서 본 발명의 예시적인 구성을 도시하지만, 본 발명은 개시된 특정 방법 및 수단으로 한정되지 않는다.

개요

오퍼레이팅 시스템 하에서 일어날 수 있는 몇몇 프로세스는 자신의 올바른 동작을 위해 하나 이상의 키에 의존한다. 키로의 액세스는, 사용자가 사용자 이름/패스워드 조합과 같은 올바른 자격증명을 제공하지 않으면 키(들)의 사용을 가능하게 하는 것을 거부하는 로그온 프로그램과 같은 인증 프로그램에 의해 제어될 수 있다. 따라서, 올바른 자격 증명의 부재로 키의 사용을 가능하게 하는 것에 대한 로그온 프로그램의 거부에 의해, 몇몇 프로세스(예를 들어, 암호화된 파일의 복호화)는 패스워드를 모르는 사용자에 대해 중단(또는, 완전히 차단)될 수 있다. 로그온 프로그램은 키로의 액세스에 문호가 되는 데 효율적일 수 있지만, 오퍼레이팅 시스템 로더는, 키를 로그온 프로그램에 의해 부과된 인증 규칙의 외부로 배포하는 상이한 컴포넌트를 로딩하는 것으로 착각할 수 있다. 따라서, 키가 이러한 방식으로 배포될 때, 키를 보호하는 것은 오퍼레이팅 시스템 로딩 프로세스를 보호하는 것을 요구한다. 본 발명은 로딩 프로세스를 보호하는 데 사용될 수 있는 메커니즘을 제공한다.

예시적인 컴퓨팅 배열

도 1은 본 발명의 양태가 구현될 수 있는 예시적인 컴퓨팅 환경을 도시한다. 컴퓨팅 시스템 환경(100)은 적합한 컴퓨팅 환경의 일례일 뿐이며, 본 발명의 사용 또는 기능의 범주에 대한 어떤 제한도 제시하려는 것이 아니다. 컴퓨팅 환경(100)은 예시적인 오퍼레이팅 환경(100)에서 도시되는 임의의 하나의 컴포넌트 또는 그 조합에 대해 어떤 의존성 또는 요구사항을 갖는 것으로 해석되어서는 안된다.

본 발명은 다수의 다른 범용 또는 특수 목적 컴퓨팅 시스템 환경 또는 구성과 함께 동작할 수 있다. 본 발명과 함께 사용하는 데 적합할 수 있는 잘 알려진 컴퓨팅 시스템, 환경 및/또는 구성의 예로는, 개인용 컴퓨터, 서버 컴퓨터, 핸드-헬드 또는 랩탑 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 컴퓨터, 셋탑 박스, 프로그램가능한 전자 기기, 네트워크 PC, 미니 컴퓨터, 메인프레임 컴퓨터, 내장형 시스템, 상기 시스템 또는 장치들 중 임의의 것을 포함하는 분산 컴퓨팅 환경 등이 있지만, 이것으로 제한되지 않는다.

본 발명은 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행가능 명령어의 일반적 문맥에서 설명될 수 있다. 일반적으로, 컴퓨터 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 타입을 구현하는 루틴, 프로그램, 오브젝트, 컴포넌트, 데이터 구조 등을 포함한다. 본 발명은 또한, 통신 네트워크 또는 다른 데이터 전송 매체를 통해 링크되어 있는 원격 프로세싱 장치에 의해 태스크가 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈 및 다른 데이터는 메모리 저장 장치를 포함하는 로컬 및 원격 컴퓨터 저장 매체 둘 다에 위치될 수 있다.

도 1을 참조하면, 본 발명을 구현하기 위한 예시적인 시스템은 컴퓨터(110) 형태의 범용 컴퓨팅 장치를 포함한다. 컴퓨터(110)의 컴포넌트는 프로세싱 유닛(120), 시스템 메모리(130), 및 시스템 메모리를 프로세싱 유닛(120)에 연결시키는 것을 포함하여 다양한 시스템 컴포넌트들을 연결시키는 시스템 버스(121)를 포함할 수 있지만, 이것으로 제한되지 않는다. 프로세싱 유닛(120)은 멀티-스레드 프로세서 상에서 지원되는 것과 같은 복수의 논리 프로세싱 유닛을 나타낼 수 있다. 시스템 버스(121)는 메모리 버스 또는 메모리 제어기, 주변장치 버스, 및 다양한 버스 아키텍처 중 임의의 것을 사용하는 로컬 버스를 포함하는 몇몇 유형의 버스 구조들 중 임의의 것일 수 있다. 예를 들면, 그러한 아키텍처로는 ISA(Industry Standard Architecture) 버스, MCA(Micro Channel Architecture) 버스, EISA(Enhanced ISA) 버스, VESA(Video Electronics Standards Association) 로컬 버스, 및 PCI(Peripheral Component Interconnect) 버스(메자닌 버스로도 알려짐)가 있지만, 이것으로 제한되지 않는다. 시스템 버스(121)는 또한, 통신 장치들 중에서 점-대-점 접속, 파이버 채널 스위치 등으로서 구현될 수 있다.

컴퓨터(110)는 일반적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터 판독가능 매체는 컴퓨터(110)에 의해 액세스될 수 있는 임의의 사용가능한 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체 둘 다를 포함한다. 예를 들어, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있지만, 이것으로 제한되지 않는다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체 둘 다를 포함한다. 컴퓨터 저장 매체로는 RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 다른 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 다른 자기 저장 장치, 또는 요구된 정보를 저장하는 데 사용될 수 있고 컴퓨터(110)에 의해 액세스될 수 있는 임의의 다른 매체가 있지만, 이것으로 제한되지 않는다. 통신 매체는 일반적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 다른 데이터를 반송파 또는 다른 전송 메커니즘과 같은 변조된 데이터 신호로 구현하고, 임의의 정보 전달 매체를 포함한다. "변조된 데이터 신호"라는 용어는 신호 내의 정보를 인코딩하는 방식으로 설정 또는 변경된 특징들 중 하나 이상을 갖는 신호를 의미한다. 예를 들어, 통신 매체는 유선 네트워크 또는 직접-유선 접속과 같은 유선 매체, 및 음향, RF, 적외선 및 다른 무선 매체와 같은 무선 매체를 포함하지만, 이것으로 제한되지 않는다. 상술된 것들 중 임의의 것의 조합도 컴퓨터 판독가능 매체의 범주 내에 포함되어야 한다.

시스템 메모리(130)는 판독 전용 메모리(ROM)(131) 및 랜덤 액세스 메모리(RAM)(132)와 같은 휘발성 및/또는 비휘발성 메모리 형태의 컴퓨터 저장 매체를 포함한다. 시작 동안과 같은 컴퓨터(110) 내의 구성요소들 사이에서 정보를 전송하는 것을 돕는 기본 루틴을 포함하는 기본 입/출력 시스템(133)(BIOS)은 일반적으로, ROM(131)에 저장된다. RAM(132)은 일반적으로 프로세싱 유닛(120)에 의해 즉시 액세스가능하고/액세스가능하거나 프로세싱 유닛(120) 상에서 현재 동작 중인 데이터 및/또는 프로그램 모듈을 포함한다. 예를 들어, 도 1은 오퍼레이팅 시스템(134), 어플리케이션 프로그램(135), 다른 프로그램 모듈(136) 및 프로그램 데이터(137)를 도시하지만, 이것으로 제한되지 않는다.

컴퓨터(110)는 또한, 다른 분리형/비분리형, 휘발성/비휘발성 컴퓨터 저장 매체를 포함할 수 있다. 예를 들어, 도 1은 비분리형, 비휘발성 자기 매체로부터 판독하고 그곳에 기록하는 하드 디스크 드라이브(141), 분리형, 비휘발성 자기 디스크(152)로부터 판독하고 그곳에 기록하는 자기 디스크 드라이브(151), 및 CD-ROM 또는 다른 광 매체와 같은 분리형 비휘발성 광 디스크(156)로부터 판독하고 그곳에 기록하는 광 디스크 드라이브(155)를 도시하지만, 이것으로 제한되지 않는다. 예시적인 오퍼레이팅 환경에서 사용될 수 있는 다른 분리형/비분리형, 휘발성/비휘발성 컴퓨터 저장 매체는 자기 테이프 카세트, 플래시 메모리 카드, DVD(digital versatile disk), 디지털 비디오 테이프, 고체 상태 RAM, 고체 상태 ROM 등을 포함하지만, 이것으로 제한되지 않는다. 하드 디스크 드라이브(141)는 일반적으로 인터페이스(140)와 같은 비분리형 메모리 인터페이스를 통해 시스템 버스(121)에 접속되고, 자기 디스크 드라이브(151) 및 광 디스크 드라이브(155)는 일반적으로 인터페이스(150)와 같은 분리형 메모리 인터페이스에 의해 시스템 버스(121)에 접속된다.

상술되고 도 1에 도시된 드라이브 및 그에 관련된 컴퓨터 저장 매체는 컴퓨터(110)에 대한 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 및 다른 데이터의 저장을 제공한다. 도 1에서, 예를 들면, 하드 디스크 드라이브(141)는 오퍼레이팅 시스템(144), 어플리케이션 프로그램(145), 다른 프로그램 모듈(146) 및 프로그램 데이터(147)를 저장하는 것으로 도시된다. 이 컴포넌트들은 오퍼레이팅 시스템(134), 어플리케이션 프로그램(135), 다른 프로그램 모듈(136) 및 프로그램 데이터(137)와 동일하거나 다를 수 있음을 유념한다. 오퍼레이팅 시스템(144), 어플리케이션 프로그램(145), 다른 프로그램 모듈(146) 및 프로그램 데이터(147)는 적어도, 그들이 복사본이라는 것을 나타내기 위하여, 여기에서 상이한 번호들을 부여받는다. 사용자는 키보드(162), 및 주로 마우스, 트랙볼 또는 터치 패드로 지칭되는 포인팅 장치(161)와 같은 입력 장치를 통해 컴퓨터(110)에 명령 및 정보를 입력할 수 있다. 다른 입력 장치(도시되지 않음)로는, 마이크로폰, 조이스틱, 게임 패드, 위성 접시, 스캐너 등이 있다. 이러한 입력 장치 및 다른 입력 장치는 종종, 시스템 버스에 연결되어 있는 사용자 입력 인터페이스(160)를 통해 프로세싱 유닛(120)에 접속되지만, 병렬 포트, 게임 포트 또는 USB(universal serial bus)와 같은 다른 인터페이스 및 버스 구조에 의해 접속될 수 있다. 모니터(191) 또는 다른 유형의 디스플레이 장치는 또한, 비디오 인터페이스(190)와 같은 인터페이스를 통해 시스템 버스(121)에 접속된다. 모니터 외에, 컴퓨터는 또한, 출력 주변 장치 인터페이스(195)를 통해 접속될 수 있는 스피커(197) 및 프린터(196)와 같은 다른 주변 출력 장치를 포함할 수 있다.

컴퓨터(110)는 원격 컴퓨터(180)와 같은 하나 이상의 원격 컴퓨터로의 논리 접속을 이용하여 네트워크 환경에서 동작할 수 있다. 원격 컴퓨터(180)는 개인용 컴퓨터, 서버, 라우터, 네트워크 PC, 피어 장치 또는 다른 공통 네트워크 노드일 수 있으며, 도 1에서는 단일 메모리 저장 장치(181)가 도시되어 있지만, 일반적으로, 컴퓨터(110)에 관련하여 상술된 구성요소들 다수 또는 전부를 포함한다. 도 1에 나타난 논리 접속은 근거리 네트워크(LAN; 171) 및 광역 네트워크(WAN; 173)을 포함하지만, 다른 네트워크도 포함할 수 있다. 그러한 네트워크 환경은 사무실, 기업형 컴퓨터 네트워크, 인트라넷 및 인터넷에서 흔하다.

LAN 네트워크 환경에서 사용될 때, 컴퓨터(110)는 네트워크 인터페이스 또는 어댑터(170)를 통해 LAN(171)에 접속된다. WAN 네트워크 환경에서 사용될 때, 컴퓨터(110)는 일반적으로, 인터넷과 같은 WAN(173)을 통한 통신을 설립하기 위한 모뎀(172) 또는 다른 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(172)은 사용자 입력 인터페이스(160) 또는 다른 적절한 메커니즘을 통해 시스템 버스(121)에 접속될 수 있다. 네트워크 환경에서, 컴퓨터(110)에 관련하여 도시된 프로그램 모듈 또는 그 일부는 원격 메모리 저장 장치에 저장될 수 있다. 예를 들어, 도 1은 메모리 장치(181)에 상주하는 원격 어플리케이션 프로그램(185)을 도시하지만, 이것으로 제한되지 않는다. 도시된 네트워크 접속은 예시적인 것이며 컴퓨터들 간의 통신 링크를 설립하는 다른 수단이 사용될 수 있음을 이해할 것이다.

키에 의해 보호된 액션

컴퓨팅 환경은 그 환경 내에서 일어나는 특정 프로세스가 올바른 동작을 위해 의존하는 키를 채용할 수 있다. 마이크로소프트 윈도우즈 오퍼레이팅 시스템에 의해 사용되는 시스템 키 또는 "SYSKEY"는 그러한 키의 일례이며, 이 예는 제한적인 것이 아니다. 바람직한 실시예에서, 프로세스가 의존하는 키는 유일한 것이며, 암호식으로 랜덤한 플랫폼별 키이다. 즉, 2개의 기계가 주어지면, 2개의 기계는 상이한 키를 가질 것이다. 따라서, 그러한 키에 의존적인 프로세스는, 적어도 플랫폼의 키가 그 플랫폼 외부에서는 사용불가능하다는 것을 보장하는 효과적인 메커니즘이 채용된다는 점에서, 한 플랫폼에서 다른 플랫폼으로 이식할 수 없다.

도 2는 키에 의존적인 프로세스가 실행되는 예시적인 시스템을 도시한다. 프로세스(202)는 올바르게 동작하기 위해 키(204)에 의존한다. 프로세스(202)는 프로세스의 전통적인 개념 즉, 오퍼레이팅 시스템에 의해 관리되고 주소 공간을 지정받을 수 있는 실행 유닛으로 제한되지 않고, 오히려, 더 일반적으로, 컴퓨터 상에서 수행될 수 있는 임의의 동작 또는 일련의 동작을 의미하는 것을 유념해야 한다. 또한, 이 예는 암호 키에 의존하는 프로세스를 나타내고 있지만, 여기에 사용되는 용어 "프로세스"는 암호 동작을 수행하는 프로세스로 한정되지 않는다는 것을 유념해야 한다.

도 2에 도시된 바와 같이, 키(204)가 프로세스(202)로의 입력으로서 사용가능하면, 프로세스(202)는 올바르게 동작한다. 반면, 키(204)가 프로세스(202)로의 입력으로서 사용가능하지 않으면, 프로세스(202)는 올바르게 동작하지 않는다. 키 보호 메커니즘(206)은 키(204)로의 액세스를 제어한다. 즉, 메커니즘(206)은 관련 보안 조건이 충족되는지의 여부에 따라 프로세스(202)에 키(204)를 제공하거나 제공하지 않는다. 예를 들어, 사용자는 메커니즘(206)이 키(204)의 사용을 가능하게 하기 전에, 로그인하여 올바른 패스워드를 제공해야 한다.

프로세스(202)가 올바르게 동작하는 것을 방지하기 위하여 키(204)의 거부가 때때로 요구되는 결과가 된다는 것을 유념해야 한다. 예를 들어, 파일의 암호화/복호화는 키(204)에 의해 보호되는 프로세스(202)의 일례이다. 파일의 올바른 복호화는 키(204)로의 액세스에 의존할 수 있다. 사용자가 올바르게 로그인하여 스스로를 인증할 수 없는 경우, 사용자가 로그인할 수 없는 것은 (예를 들어, 도난된 랩탑의 경우) 의도된 사용자가 아닌 사람에 의해 컴퓨터가 동작되는 것을 나타낼 수 있기 때문에, 파일의 복호화가 진행되지 않는 것이 바람직할 수 있다. 따라서, 키 보호 메커니즘(206)은 키(204)로의 액세스를 충족되어야 하는 관련 보안 조건에 의존하게 할 수 있고, 그러한 보안 조건이 충족되지 않을 경우, 키(204)의 거부를 사용하여 중단되어야 하는 프로세스를 중단시킬 수 있다. 이러한 방식으로 프로세스를 중단시키는 메커니즘(206)의 능력은 키를 거부하는 행위가 프로세스를 중단시킬 수 있게 하는 의존성이기 때문에, 이 프로세스가 자신의 올바른 동작을 위해 요구하는 키(204)에 의존한다.

도 3은 키(204)에 의존하는 프로세스의 특정 (비제한적인) 예를 도시한다. 도 3의 예에서, 예시적인 프로세스는 파일을 암호화된 형태로 저장하고 그 암호화된 파일을 복호화하는 파일 암호 시스템(Encrypting File System; EFS)(302)이다. 그러한 파일 암호화의 한 목적은, 랩탑 컴퓨터가 도난당한 경우 랩탑 컴퓨터 상의 데이터를 절도자에 의해 발견되는 것으로부터 보호하는 것임을 유념해야 한다. 저장을 위해 파일(304)이 생성되면, 파일은 EFS(302)에 제공된다. EFS(302)는 파일(304)을 암호화하고 파일(304)을 하드 디스크(141)에 저장되는 암호화된 파일(306)로 변환한다. 암호화된 파일(306)을 검색하라는 요구가 있을 때, EFS(302)는 암호화된 파일(306)을 검색하고, 그것을 복호화하여 복호화된 파일(308)을 생성한다. 도 3은 명료성을 위해, 이 파일의 두개의 인스턴스를 따로 도시하지만, 실제로, 복호화된 파일(308)의 실체는 원본 파일(304)의 실체와 동일하다. 즉, 파일(304)은 원본 파일이고, 복호화된 파일(308)은 EFS(302)에 의해 암호화되고 저장되고 검색되고 복호화된 후의 동일한 파일이다.

EFS(302)로의 하나의 입력은 컨텐츠 키(310)임을 유념한다. 컨텐츠 키(310)는 바람직하게, 암호화 방법으로의 입력으로서 기능하는 대칭 키다. 컨텐츠 키(310)는 암호화된 파일(306)을 생성하기 위해 파일(304)을 암호화하는 데 사용되고, 또한, 복호화된 파일(308)을 생성하기 위해 암호화된 파일(306)을 복호화하는 데 사용된다. 임의의 쉽게 검색가능한 장소에 키(310)를 저장하는 것은 무효한 데이터를 보호하는 EFS(302)의 능력을 신속히 렌더링할 수 있다. 컨텐츠 키가 하드 디스

크 상에서 쉽게 사용가능해지거나 랩탑의 몇몇 알려진 특징(예를 들어, 프로세서 일련 번호)으로부터 쉽게 유도되면, 절도자가 쉽게 키를 찾아 복호화할 수 있기 때문에, 파일이 암호화된 형태로 저장되었든지 저장되지 않았든지 그 차이가 없다. 따라서, 진짜 소유자의 협력으로만 행해질 수 있는 임의의 방법으로 키를 유도함으로써 그 키를 보호하는 것이 바람직하다. 키를 보호하는 한가지 방법은, 키(204)를 입력으로서 수신하고 컨텐츠 키(310)를 키(204)의 함수로서 유도하는 키 생성기 모듈(312)을 사용하는 것이다. 따라서, 키(204)가 적절히 안전한 환경 하에서만 제공될 것이라는 점에서, 컨텐츠 키(310)가 적절한 환경 하에서만 유도가능할 것이라는 것도 사실이다. 다시 말하면, 컨텐츠 키(310)의 유도가 키(204)의 사용가능성에 의존하게 함으로써, 키(204)에 제공되는 보호는 어느 것이든지 컨텐츠 키(310)로 확장될 수 있다. 예를 들면, 키(204)를 제공하는 것이 사용자가 올바른 패스워드를 제공함으로써 로그인하는 것을 요구하면, 사용자가 올바르게 로그인하지 않는 경우, 컨텐츠 키(310)는 사용불가능해질 것임을 보장할 수 있다.

따라서, 다른 프로세스는 올바른 보안 컨텍스트(예를 들어, 올바른 패스워드를 제공하는 적합하게 로그인된 사용자)가 존재하는 경우에만 배포되는 키(204)에 의존할 수 있기 때문에, 키(204)를 잘못된 환경 하에서 배포되는 것으로부터 보호하는 것이 중요하다. 이하에 설명되는 바와 같이, 오용으로 귀착될 수 있는 방식으로 기계가 키(204)를 배포하게 하는 한가지 방법은, 오퍼레이팅 시스템의 악성 컴포넌트가 올바른 컴포넌트를 대체할 수 있는{올바른 컴포넌트가 키(204)를 보호할 수 있는 것으로 가정함} 불안한 환경에서 기계를 부팅하는 것이다. 따라서, 키(204)가 배포될 수 있기 전에 공지의 (안전한) 환경에서 기계가 부팅되는 것을 보장하는 메커니즘이 이하에서 설명된다.

상태 검증을 갖는 부팅 프로세스

도 4는 부팅 프로시저에서 사용되는 이벤트의 일반적인 시퀀스를 도시한다.

초기에, 기계가 전원이 켜진다. 일반적인 기계는 전원이 켜진 시각에 임의의 고정 주소에서 명령어를 실행하기 시작하도록 구성된다. 일반적으로 이 주소에 포함된 명령어는 "BIOS"(402) 또는 "기본 입/출력 시스템"으로서 알려져 있다. BIOS(402)의 실행의 끝에서, BIOS(402)는 "옵션 ROM"(404)이라고 불리는 작은 프로그램의 실행을 시작한다. 옵션 ROM은 기계에 대한 하드웨어 패스워드를 설정하는 것, 또는 몇몇 오퍼레이팅 시스템 중 어느 것이 부팅되어야 할지를 선택하는 것과 같은 매우 기본적인 초기-부팅 함수를 수행하는 프로그램이다. 옵션 ROM(404)이 실행된 후, 기계는 마스터 부트 레코드(Master Boot Record; MBR)(406)를 로딩하도록 지시된다.

MBR(406)은 실행가능한 프로그램이다. 일반적으로, MBR(406)은 컴퓨터의 하드 디스크의 제1 섹터 상에 상주하고, 파티션 테이블에서 추가 부팅을 위해 어느 파티션을 사용할지를 검색함으로써 시작된다(예를 들어, 디스크는 상이한 오퍼레이팅 시스템들과 함께 사용하기 위해 파티션될 수 있고, 각각의 오퍼레이팅 시스템은 상이한 부팅 프로시저를 요구할 수 있음). 올바른 파티션이 검색된 후, MBR(406)은 그 파티션에 관련된 부팅 섹터(408)에 제어를 전달한다. 부팅 섹터는 최종적으로 오퍼레이팅 시스템을 로딩할 오퍼레이팅 시스템 로더(410)를 로딩하는 프로세스를 개시한다. MBR(406)은 도 4에서 단지, 그러한 컴포넌트가 어떻게 예시적인 부팅 프로세스에 맞는지를 보여주기 위해 도시되며, 본 발명은 MBR(406)을 사용하는 부팅 프로시저로 제한되지 않는다.

오퍼레이팅 시스템 로더(410)가 실행되는 시간 동안, 로더 및 기계 상태가 검증(450)된다. 이 문맥에서, "기계"는 물리적인 기계 또는 가상 기계를 의미할 수 있다. 검증은 고-신뢰도 환경에서 동작하는 신뢰된 소프트웨어 컴포넌트에 의해 수행되고(이것의 일 실시예는 이하에서 더 상세히 설명됨), 따라서, 로더(410)의 검증이 올바르게 행해지는 보증/신뢰가치성의 임의의 정도가 존재한다. 본질적으로, 오퍼레이팅 시스템 로더(410)가 기계의 제어를 갖는 동안 기계 상태의 관련 양태를 검증함으로써, 및 로더(410)를 검증함으로써, 오퍼레이팅 시스템 컴포넌트{예를 들어, 하드웨어 추상 계층, 커널, 드라이버 등(416)}의 불법 또는 악성 버전이 로그온 프로그램(418)이 실행될 때까지 나중에 로딩되지 않을 것이라는 보증의 임의의 레벨을 제공하는 것이 가능하다. 오퍼레이팅 시스템이 로딩된 후 행할 첫번째 것들 중 하나는 키(204)로의 액세스에 문호가 되는 로그온 프로그램(418)을 실행하는 것이기 때문에, 악성 컴포넌트가 로그온 프로그램(418)이 실행될 때까지 로딩되는 것을 방지하는 것이 중요하고, 악성 컴포넌트가 로딩되면, 이 컴포넌트는 로그온 프로그램(418)이 부적절한 환경 하에서 키(204)를 배포하게 되는 방식으로 오작용하게 하고, 이에 의해, 상술된 키(304)의 보호에 의존하는 모든 컴포넌트의 보안을 손상시킬 수 있다. 따라서, 키(204)의 보호는 오퍼레이팅 시스템 로더(410)가 실행되는 시각으로부터 로그온 프로그램이 완료될 때까지 기계의 상태에 대한 엄격한 제어를 발휘함으로써 달성될 수 있다.

오퍼레이팅 시스템 로더(410)가 실행되는 시간 동안 검증(450)이 일어난다. 검증은 로더 및 기계 상태를 검증하는 것을 포함하고, 또한, 기계 상태를 공지의 양호 상태로 설정하는 것도 포함할 수 있다. 기계 상태를 검증하고 설정하는 기본 아이디어는 그 기계를, 기계가 그 상태에 있는 동안 로더가 실행되면 로더는 임의의 악성 컴포넌트를 로딩하지 않고, 실행되지 않으면, 보안 침해가 되는 방식으로 행동하는 상태에 놓는 것이다. 검증(450)은 로더 코드가 사실, 이전에 올바르게 행동하는 것으로 검증된 코드임을 보장하며, 또한, 기계가 이 공지의 코드가 올바르게 행동하게 될 상태에 있음을 (기계가 이미

그 상태에 있음을 검증함으로써, 또는 그 기계를 그 상태에 놓음으로써) 보장한다. 이 기술은 기계의 기존의 (안전한) 상태에 영향을 주어, 기계의 미래 상태도 안전할 것임을 보장할 수 있고, 이 영향은 오퍼레이팅 시스템 로더(410)의 행동이 공지되고 이해될 뿐만 아니라 엄격하게 제한된다는 사실에 의해 가능해진다는 것을 이해할 것이다. 이 관측에 기초하여, 다수의 상이한 소스들로부터의 매우 다양한 코드, 및 그러한 코드의 실행의 결과로서 기계가 놓일 수 있는 상태의 방대한 폭은 불가능하지 않다면, 이 다양한 상이한 코드 모듈들 전부를 실행하는 동안 기계의 행위을 결정하는 것을 어렵게하기 때문에, 검증(450)이 너무 이른 시간에{예를 들어, 옵션 ROM(404) 또는 MBR(406)이 실행되는 시간에} 일어나서는 안된다는 것을 이해할 것이다. 따라서, 기계가 로더(410)가 실행되는 시간에 합법적인 상태에 놓일 수 있는 한, 로더(410)의 실행 전에 기계가 어느 상태에 있는지를 고려하지 않는 것이 바람직하다.

바람직한 실시예에서, 로더(410)의 실행은 2개의 단계 즉, 단계 1(412) 및 단계 2(414)로 나뉘어진다. 바람직하게, 단계 2에 진입하기 전에, 단계 2를 구현하는 코드가 검증(450)되고, 검증(450)을 수행하는 신뢰된 코드가 잘 정의된 엔트리 지점에서 단계 2로 점프한다. 검증을 수행하는 "신뢰된 코드"는 고-신뢰도 환경에서 실행되는 상술된 프로그램이다. 본 기술분야에 숙련된 기술자는 "신뢰된다"는 것이 절대적인 확실성을 의미하는 것이 아니라, 단지, 코드가 자신의 일을 올바르게 수행할 것을 가정할 임의의 기반이 있다는 것을 의미함을 이해할 것이다. 프로그램의 행위는 그것이 실행되는 환경에 의해 영향을 받을 수 있기 때문에, 고-신뢰도 환경에서 신뢰된 코드를 실행하는 것은 신뢰된 코드가 올바르게 동작할 것이을 의미한다. 즉, (1) 신뢰된 코드는 임의의 기대에 부합하는 환경에서 자신의 기능을 올바르게 수행하는 것으로 신뢰될 수 있고, (2) 고-신뢰도 환경은 그러한 기대에 부합하는 환경을 올바르게 제공하는 것으로 신뢰될 수 있다. 단계 2는 단계 1로부터 수신한 임의의 정보를 (임의의 정의된 표준에 따라) 검증할 수 있다. 단계 1과 단계 2 간의 분할 선은, 보안-관련 태스크가 올바르게 수행되지 않게 할 수 있는 이벤트가 운용중에 설정되기 전에 관련 프로그램 및 상태가 임의의 지점에서 검증될 수 있는 한, 임의의 보안 검증 없이도, 및 {키(204)의 배포와 같은} 그러한 임의의 보안-관련 태스크의 임의의 결과를 갖지 않고서도 수행될 수 있는 부팅 프로시저의 몇몇 양태가 있을 수 있다는 사실을 반영한다. 또한, 부팅 프로세스에서 상태를 너무 빨리 검증하려는 것에는 문제가 있을 수 있다. 합법적인 부팅 프로시저가 취할 수 있는 실제 실행 경로는 매우 가변적이어서, 그러한 가변적인 환경 하에서 유효하거나 무효한 기계 상태들 간의 차이를 정의하는 것이 어려워진다. 기계가 통과하는 임의의 상태가 유효한지를 결정하는 것을 시도하지 않고서, 단순히 이 가변적 상태들 중 임의의 것을 통해 로드가 진행되게 하는 것은 이치에 맞는다. 따라서, 단계 1은 임의의 검증 없이 진행하는 로드의 일부분을 나타낸다. 단계 1의 끝에서, 로더 및 기계 상태가 검증되고, 단계 2가 시작된다. 그러한 분할 선의 가정은, 기계에 관한 모든 관련 요소들이 검증되고, 기계가 보안 표준지점으로부터 수용불가능한 상태에 기계를 놓았던 임의의 이전의 액션의 효과를 반드시 무효화하는 공지의 양호 상태에 놓이는 특정 지점까지, 임의의 보안 제한 없이 프로세스가 실행할 수 있다는 것이다. 단계 1과 단계 2 간의 분할 선이 그려져 있는 로드 프로세서 내의 정확한 지점은 환경(예를 들어, 로더 코드가 어떻게 생겼는가? 로드의 일부로서 어떤 태스크가 행해져야 하는가?)에 매우 특정적이며, 일종의 트레이드오프(tradeoff)를 나타낸다. 한편, 분할 선은, 미래의 액션을 위해 합법적인 로드 행위의 변화량이 충분히 작아서 합법적인 행위가 불법적인 행위로부터 확실히 구별될 수 있게 하도록, 충분히 나중이어야 한다. 상술된 바와 같이, 이른 로드 프로시저에서는, 다수의 옵션 ROM 및 다른 변수가 가능한 실행 경로의 수를 매우 크게 하여, 불법적인 행위로부터 합법적인 행위를 구별하는 것이 어려워진다. 한편, 분할 선은, 키가 부적절하게 배포되게 하는 것과 같이, 보안에 영향을 줄 수 있는 임의의 이벤트(예를 들어, 로그온 프로그램의 로딩)를 진행하도록, 충분히 일찍이어야 한다. 일반적으로, 단계 1과 단계 2 간의 분할 선은, 사람이 얼마동안 "미가공"(또는, "공개" 또는 "비검증") 상태의 임의의 시스템을 실행하고, 행위(예를 들어, 고-신뢰도 환경)를 검증하는 데 필요한 컴포넌트를 시작하고, 그러한 컴포넌트를 사용하여 기계의 현재 상태를 검증(또는, 기계를 공지의 양호 상태로 시행)하게 하는데, 이 때, 나중에 보안에 영향을 줄 수 있는 무언가(예를 들어, 키의 배포)를 행하는 프로세스가 진행될 수 있다.

검증(450)

검증(450)은 본질적으로, 오퍼레이팅 시스템 로더(410)가 공지의 신뢰된 프로그램임을 검증하고(또는 특히, 바람직한 실시예에서, 로더의 단계 2를 검증), 검증이 실행되는 기계가 공지의 양호 상태에 있음을 보장하는 행위이다. 따라서, 검증은 2개의 부분 즉, (1)로더 프로그램(또는, 로더 프로그램의 단계 2)을 검사하여 그것이 보통 생각하는 공지의 신뢰된 프로그램임을 보장하는 부분, 및 (2)로더가 올바르게 행위하는 것으로 알려진 상황 하에서, 관련 기계 상태를 공지의 양호 상태로 변경하는 부분을 갖는다. 수행 (1) 및 (2)의 전제는 공지의 기계 상태에서 동작하는 공지의 프로그램은 공지의 방법으로 동작할 것이라는 것이다.

검증의 제1 부분 즉, 로더 검사는 다양한 방법으로 수행될 수 있다. 가장 단순한 예로, 실행중인 실제 프로그램이 비교될 수 있는 (디지털 서명을 생성하기 위해 암호식으로 서명되거나, 임의의 속일 수 없는 방법으로 저장될 수 있는) 공지의 프로그램의 해쉬가 있다. 전체 프로그램에 대해 해쉬를 취하는 문제는, 합법적인 프로그램의 상이한 인스턴스들이 약간의 상이한 이미지들을 가질 수 있어서, 공지의 프로그램 및 실행중인 프로그램 인스턴스 사이의 완전한 아이덴티티를 요구하는 방식으로 계산되는 해쉬가 너무 제한적일 수 있다는 것이다. 바람직하게, 검증은, 프로그램이 그렇게 기대되는 것임을 보장하

지만 지나치게 제한적이지 않은 방법으로 수행된다(예를 들어, 검증 프로세스는 수리되어야 할 것으로 알려진 프로그램 컴포넌트의 해시들을 비교할 수 있고, 시간이 흐름에 따라 변할 수 있는 프로그램 컴포넌트에 대한 다른 테스트를 수행할 수 있음). 검증을 수행하는 컴포넌트는 검증되는 프로그램의 세부사항에 맞추어 만들어져야 한다. 본 발명은 프로그램을 검증하기 위한 임의의 특정 기술로 제한되지 않는다.

검증의 제2 부분 즉, 기계 상태 검증/설정은 바람직하게, 상태의 관련 "소스" 전부를 공지의 값으로 설정함으로써 수행된다. 일반적으로, 오퍼레이팅 프로그램에 영향을 줄 수 있는 관련 상태는 3개의 소스 즉, CPU, 칩셋 및 메모리로부터 온다. 따라서, 이 아이템들은 검증 시에, 공지의 양호 상태에 놓일 수 있다. 예를 들어, CPU는 공지의 상태 예를 들면, 프로그램 카운터가 공지의 위치를 가리키고, 모든 데이터 레지스터가 0으로 설정되고, 로더 프로그램이 저장되어 있는 메모리 외의 모든 메모리가 0으로 설정되는 링 0에 놓일 수 있다. 로더가 이 상태의 기계와 함께 실행될 때 올바르게 동작하는 것으로 검증되면, 로더를 검사하는 것과 기계를 공지의 양호 상태에 놓는 것의 결합은 로그온 프로그램(418)이 실행될 때까지의 올바른 행위를 보장해야 한다. 상술된 바와 같이, 로그온 프로그램(418)의 실행까지의 올바른 행위에 대한 (유일한 이점이 아니더라도) 이점들 중 하나는, 부적절한 환경 하에서 키(204)가 배포되게 하는 악성 컴포넌트가 로딩되지 않을 것이라는 것이다.

검증 프로세스가 악성 컴포넌트로부터 오퍼레이팅 시스템의 로드를 보호하기 위해서는, 검증 프로세스 자체가 올바르게 수행된다는 충분한 보장이 있어야 한다는 것을 명심한다. 검증 프로세스의 정확도는 검증기를 높은-보증 컴퓨팅 환경에서 실행하는 신뢰된 에이전트로서 구현함으로써 보장될 수 있다. 예를 들어, 제한된 집합의 기능을 수행하지만, 그러한 오퍼레이팅 시스템이 본 명세서에 따라 수행될 높은 수준의 보증을 제공하는 작은 오퍼레이팅 시스템이 있을 수 있다. 그러한 오퍼레이팅 시스템은 단일 기계 상에서 다른 오퍼레이팅 시스템과 함께 실행될 수 있고, 시스템 상의 다른 (덜 안전한) 환경으로부터 그러한 높은 보증 오퍼레이팅 시스템을 고립하는 것은 하이퍼바이저 또는 가상 기계 모니터와 같은 감독 컴포넌트에 의해 시행될 수 있다. 일 실시예에서, 하이퍼바이저는, 하이퍼바이저가 서로 임의의 상호 고립의 상태를 유지하고 오퍼레이팅 시스템이 실행될 수 있는 환경인 파티션을 시행한다. 또한, 높은 보증 컴포넌트는 플랫폼별 암호 키를 적용하고 철저하게 보호하는 하드웨어 모듈과 같은 신뢰 루트로의 배제적인 액세스를 가질 수 있다. 검증기는, 고-신뢰도 환경 자체가 공격에 저항하는 것으로 신뢰될 수 있다는 점에서, 검증기가 올바르게 동작하게 하는 고-신뢰도 환경의 외부로부터 검증기가 공격받지 않는다는 보장을 제공하는 그러한 높은 보증 환경에서 실행하는 프로그램(또는, "에이전트")일 수 있다. 또한, 상술된 바와 같이, 고-신뢰도 환경은 봉인된 저장장치(즉, 몇몇 데이터를 저장하고, 그 데이터를 데이터가 봉인된 특정 오브젝트에게만 배포하는 능력)를 제공할 수 있고, 검증기는 이 봉인된 저장장치를 사용하여 배포될 임의의 키를 저장할 수 있다. 검증기는 고-신뢰도 환경의 필수 부분일 수 있지만, 또한, 고-신뢰도 환경에서 실행되는 별도의 컴포넌트일 수도 있음을 유념한다.

도 5는 상술된 프로시저에 따라 검증을 수행하는 검증기(550)의 예를 도시한다. 검증기(550)는 정확도 및/또는 임의의 집합의 공지의 표준과의 호환성에 대해 로더(410)를 평가하는 능력을 갖는다. 또한, 검증기(550)는 기계 상태(502)를 평가하고/평가하거나 그것에 영향을 주는 능력을 갖는다. 이 능력의 결합을 이용하여, 검증기(550)는 로더(410)가 실행되기로 기대되는 로더임을 보증함으로써, 및 로더가 실행되는 기계가 로더(410)가 올바르게 동작하는 것으로 알려진 상태에 있음을 보장함으로써, 로더(410)가 올바르게 동작할 것을 보장할 수 있다.

구조적으로, 검증기(550)는 2개의 컴포넌트 즉, 일반 부분(504)과 로더-특정 부분(506)을 포함할 수 있다. "일반 부분"(504)은 매우 다양한 검증기들(또는 모든 검증기)에 공통인 코드를 포함한다. 로더-특정 부분(506)은 특정 로더(410)의 검증에 특히 관련된 코드, 즉, 로더(410){또는 로더(410)의 단계 2}의 올바른 인스턴스가 어떻게 생겼는지를 이해하고, 로더(410)가 이 이해에 부합하는 것을 보장하기 위해 테스트를 수행하는 코드이다. 따라서, 로더-특정 부분(506)은 일반 부분(504)과 결합되어 완전한 검증기를 형성할 수 있다.

검증기(550)는 궁극적으로 키(204)가 로더(이것은 상술된 바와 같이, 로그온 프로그램에 의해 허용된 방법에 따라 오퍼레이팅 시스템에 의해 사용되기 위해, 나중에 그것을 오퍼레이팅 시스템에 제공할 것임)에게 제공될지를 결정하는 컴포넌트이다. 검증기(550)는 봉인된 저장 장치(508)에서 키(204)를 검증기(550)에 봉인함으로써 키(204)를 보호할 수 있다. 봉인된 저장 장치(508)는 높은-보증 컴퓨팅 환경의 특징일 수 있다. 보인된 저장 장치(508)는 고-신뢰도 환경에서 실행중인 컴포넌트가 임의의 데이터를 자신들에게 봉인하는 것을 허용하여, 다른 컴포넌트가 그 데이터를 검색할 수 없도록 할 수 있다. 예를 들어, 검증기(550)의 로더-특정 부분(506)은 키(204)를 자신에게 봉인할 수 있다. 봉인된 저장 장치(508)는 로더-특정 부분(506) 이외의 임의의 컴포넌트가 키(204)를 봉인해제하는 것을 방지할 것이다. 따라서, 키(204)는 오직 로더-특정 부분(506)으로부터 획득될 수 있고, 로더-특정 부분(506)은 로더-특정 부분(506)이 키(204)가 부적절하게 배포되게 하지 않을 것임을 믿는 로더에게만 키(204)를 배포할 것이기 때문에, 키(204)가 보호된다. 또 다른 실시예에서, 키는 검증기 및 로더(410)의 단계 2를 포함하는 컴포넌트의 결합에 기초하여 봉인된다. 그러한 실시예에서, 공지의 올바른 단계 2

로더에 대한 이진(binary)은 봉인에 들어가는 메트릭의 일부분이므로, 봉인된 저장 장치 메커니즘 자체는 본질적으로, 키가 봉인해제될 수 있기 전에 올바른 단계 2 로더의 존재를 보장하기 때문에, 검증기(506)의 로더-특정 부분이 제거될 수 있다.

SYSKEY를 보호하는 예시적인 프로세스

도 6은 상술된 본 발명의 양태에 따라 키를 보호하는 예시적인 프로세스를 도시한다. 기계가 시작된 후의 임의의 지점에서, 오퍼레이팅 시스템 로더가 실행된다(602). 오퍼레이팅 시스템 로더의 실행 동안, 로더 및 기계 상태가 검증되고(604), 이에 의해, 상술된 이유로, 로더가 예측가능한 방식으로 동작할 것임을 보증한다. 로더 및 기계 상태가 검증된 후, 로더는 오퍼레이팅 시스템을 로딩하는 한편, 악성 컴포넌가 로딩되는 것을 방지하는 데 사용된다(606). 상술된 바와 같이, 로더 및 기계 상태의 검증은 로더의 미래 동작이 악성 컴포넌트의 로딩으로 귀착하지 않는 것으로 알려지는 것을 의미한다. 오퍼레이팅 시스템이 로딩된 후, 키로의 액세스에 문호가 되는 로그온 프로그램이 실행된다.

다른 예시적인 실시예

본 발명의 메커니즘은 단지 오퍼레이팅 시스템 로드를 수행하기 위해 사용되는 것이 아니라, 일반적으로, 시스템이 미가공(미검증된) 상태의 몇몇 기능을 수행하게 하는 동시에, 검증이 특정 기능을 시작하도록 요구하는 데 사용될 수 있다. 예를 들어, 컴퓨터는 임의의 유형의 검증 없이 수행될 수 있는 임의의 기능을 수행할 수 있지만(예를 들어, 컴퓨터는 무선으로서 동작할 수 있음), 컴퓨터가 더 민감한 기능(예를 들면, 하드 디스크로부터 파일을 판독하는 것)을 수행할 수 있기 전에 검증이 수행되어야 한다. 일반적으로, 기계는 기기와 일반 PC 둘 다로서 보일 수 있는데, 그 기기 부분은 검증/로그온을 요구하지 않지만, 일반 PC는 요구한다.

또한, 검증기는 특정 집합의 키를 로그온 프로그램에 배포하는/배포하기 않는 것으로 제한되지 않고, 일반적으로, 특정 집합의 키를 특정 소프트웨어 스택에 배포하도록 구성될 수 있다. 예를 들어, 제1 키를 획득할 수 있는 하나의 소프트웨어 스택이 있지만, 검증기는 제2 키를 "더 엄격히 검증된" 소프트웨어 스택에만 배포할 수 있다.

또한, 키는 반드시, 로그온 프로시저를 통해 배포될 필요가 없고, 오히려 임의의 검증 프로시저를 통해 제공될 수 있다. 예를 들어, DVD 플레이어 이진은 검증기가 플레이어가 올바른 이진임을 결정한 후에 로그온을 요구하지 않고서 DVD 기능 키를 획득할 수 있다.

또한, 몇몇 프로그램은 검증 없이 임의의 형태로 작업할 수 있지만, 특정 기능을 노출하도록 허용되기 전에 일종의 검증을 요구한다. 예를 들어, 전화 어플리케이션은 로그온이 요구되지 않은 채 시작할 수 있지만, 어플리케이션의 암호 기능이 작동하도록 허용하는 키가 배포될 수 있기 전에 로그온을 요구할 수 있다.

상술한 예는 단순히 설명을 위해 제공된 것이며, 본 발명의 제한으로서 해석되어서는 안된다. 본 발명은 다양한 실시예를 참조하여 설명되었지만, 여기에서 사용된 단어는 제한적인 것이 아니라, 설명 및 예시적인 단어임을 이해해야 한다. 또한, 본 발명은 여기에서 특정 수단, 자료 및 실시예를 참조하여 설명되었지만, 본 발명은 여기에 개시된 특정사항으로 제한되도록 의도되지 않고, 오히려 본 발명은 첨부된 청구범위의 범주 내에 있는 것과 기능적으로 동등한 모든 구조, 방법 및 사용으로 확장된다. 본 명세서의 가르침의 이점을 가지고, 본 기술분야에 숙련된 기술자는 본 발명의 양태에서, 본 발명의 범주 및 취지를 벗어나지 않고서 다수의 수정 및 변경을 행할 수 있다.

발명의 효과

본 발명은, 로드가 올바르게 수행되는 것이 검증될 수 있는 환경 하에서의 오퍼레이팅 시스템의 로딩을 제공한다.

(57) 청구의 범위

청구항 1.

컴퓨터 실행가능 명령어들로 인코딩된 컴퓨터-판독가능 매체로서,

상기 컴퓨터 실행가능 명령어들은,

오퍼레이팅 시스템 로더를 시작하는 단계;

상기 로더의 아이덴티티 또는 정확도를 검증하는 단계;

상기 오퍼레이팅 시스템 로더가 실행되는 기계가 공지의 상태에 있음을 보장하는 단계;

상기 로더의 상기 아이덴티티 또는 정확도가 검증되고, 상기 오퍼레이팅 시스템이 실행되는 상기 기계가 공지의 상태에 있는 경우,

상기 로더에 키를 제공하는 단계; 및

상기 로더가 오퍼레이팅 시스템을 로딩하게 하는 단계

를 포함하는 방법을 수행하는 컴퓨터-판독가능 매체.

청구항 2.

제1항에 있어서,

상기 방법은 상기 키로의 액세스에 문호(gate)가 되는 로그온 프로그램을 실행하는 단계를 더 포함하는 컴퓨터-판독가능 매체.

청구항 3.

제2항에 있어서,

상기 로그온 프로그램은 사용자가 상기 키로의 액세스를 부여하는 상기 로그온 프로그램에 대한 조건으로서 자격증명을 제공하는 것을 보장하는 컴퓨터-판독가능 매체.

청구항 4.

제1항에 있어서,

상기 키는 암호화된 오퍼레이팅 시스템 파티션을 복호화하는 데 사용가능한 컴퓨터-판독가능 매체.

청구항 5.

제1항에 있어서,

상기 로더의 아이덴티티 또는 정확도를 검증하는 단계, 및 상기 기계가 공지의 상태에 있음을 보장하는 단계는,

기본 입력 출력 시스템;

옵션 ROM;

마스터 부팅 레코드; 및

부팅 섹터

가 모두 실행된 후에 수행되는 컴퓨터-판독가능 매체.

청구항 6.

제5항에 있어서,

상기 로더의 아이덴티티 또는 정확도를 검증하는 단계, 및 상기 기계가 공지의 상태에 있음을 보장하는 단계는, 상기 오퍼레이팅 시스템 로더의 일부분이 실행한 후에 수행되는 컴퓨터-판독가능 매체.

청구항 7.

제6항에 있어서,

상기 검증 단계는 오퍼레이팅 시스템의 전부 또는 거의 전부가 시작된 후에, 또는 임의의 수의 파티션이 시작된 후에 수행되고,

상기 파티션 각각은, 다른 파티션으로부터의 임의의 정도의 고립이 하이퍼바이저에 의해 유지되는 환경을 포함하는 컴퓨터-판독가능 매체.

청구항 8.

제1항에 있어서,

상기 키는 상기 검증 단계 및 보장 단계를 수행하는 검증기에 봉인되고,

상기 방법은 상기 검증기가 상기 키를 봉인해제하는 단계를 더 포함하는 컴퓨터-판독가능 매체.

청구항 9.

제8항에 있어서,

상기 키는 적어도 상기 검증기에 봉인되고, 상기 검증기는 상기 로더의 적어도 일부분을 검증하는 컴퓨터-판독가능 매체.

청구항 10.

제8항에 있어서,

상기 키는 상기 검증기 및 상기 로더의 적어도 일부분에 봉인되는 컴퓨터-판독가능 매체.

청구항 11.

제1항에 있어서,

상기 기계는 물리적 기계를 포함하는 컴퓨터-판독가능 매체.

청구항 12.

제1항에 있어서,

상기 기계는 가상 기계를 포함하는 컴퓨터-판독가능 매체.

청구항 13.

제12항에 있어서,

상기 키는 상기 가상 기계의 아키텍처가 변경되지 않았거나 유효한 경우에만 누설되는 컴퓨터-판독가능 매체.

청구항 14.

제1항에 있어서,

상기 기계가 공지의 상태에 있음을 보장하는 단계는, 상기 기계의 현재 상태를 평가하고 상기 현재 상태를 상기 공지의 상태와 비교하는 단계를 포함하는 컴퓨터-판독가능 매체.

청구항 15.

제1항에 있어서,

상기 기계가 공지의 상태에 있음을 보장하는 단계는, 상기 기계의 현재 상태를 상기 공지의 상태와 일관되게 설정하는 단계를 포함하는 컴퓨터-판독가능 매체.

청구항 16.

제1항에 있어서,

상기 로더는 전체 오퍼레이팅 시스템 또는 오퍼레이팅 시스템의 전체 인스턴스인 컴퓨터-판독가능 매체.

청구항 17.

오퍼레이팅 시스템의 부팅의 신뢰도에 관한 보증을 제공하는 환경 하에서, 상기 부팅을 수행하기 위한 시스템으로서,

상기 오퍼레이팅 시스템을 로딩할 오퍼레이팅 시스템 로더의 정확도 또는 아이덴티티를 평가하고, 상기 오퍼레이팅 시스템 로더가 동작할 기계의 상태를 더 평가하고, 상기 오퍼레이팅 시스템 로더의 상기 정확도 또는 아이덴티티가 검증되는지의 여부에 따라 상기 오퍼레이팅 시스템 로더가 상기 오퍼레이팅 시스템의 로딩을 진행하는 것을 허용하거나 허용하지 않고, 상기 오퍼레이팅 시스템 로더가 진행하도록 허용되기 전에 상기 기계를 공지의 상태로 하는 검증기

를 포함하는 시스템.

청구항 18.

제17항에 있어서,

상기 검증기에게 키가 봉인되고,

상기 오퍼레이팅 시스템 로더가 진행하도록 허용되는 경우, 상기 검증기는 상기 키를 봉인해제하고, 상기 키를 상기 오퍼레이팅 시스템 로더에 제공하는 시스템.

청구항 19.

제17항에 있어서,

상기 검증기는 상기 오퍼레이팅 시스템 로더가 동작을 시작한 후지만 상기 오퍼레이팅 시스템 로더가 커널 또는 장치 드라이버를 개시하기 전에, 상기 오퍼레이팅 시스템 로더의 상기 정확도 또는 아이덴티티를 평가하고, 상기 기계의 상태를 평가하는 시스템.

청구항 20.

제17항에 있어서,

상기 검증기는,

한 클래스의 검증기들에게 공통인 일반 부분; 및

상기 오퍼레이팅 시스템 로더에 특정적이고, 상기 검증기가 상이한 오퍼레이팅 시스템 로더를 검증하는 데 사용되는 경우 상이한 부분으로 대체가능한 특정 부분

을 포함하는 시스템.

청구항 21.

제17항에 있어서,

상기 기계는 물리적 기계를 포함하는 시스템.

청구항 22.

제17항에 있어서,

상기 기계는 가상 기계를 포함하는 시스템.

청구항 23.

오퍼레이팅 시스템을 부팅하는 방법으로서,

기본 입출력 시스템, 옵션 ROM, 마스터 부팅 레코드 및 부팅 섹터를 실행하는 단계;

오퍼레이팅 시스템 로더를 시작하는 단계;

상기 오퍼레이팅 시스템 로더를 검증하는 단계;

상기 오퍼레이팅 시스템 로더가 실행되는 기계의 상태를 검증하는 단계; 및

상기 오퍼레이팅 시스템 로더 및 상기 기계의 상태가 유효한 것으로 결정되는 경우, 상기 오퍼레이팅 시스템 로더가 오퍼레이팅 시스템을 로딩하게 하는 단계

를 포함하는 방법.

청구항 24.

제23항에 있어서,

상기 오퍼레이팅 시스템 하에서 적어도 하나의 기능을 올바르게 수행하는 데 필요한 키는, 상기 오퍼레이팅 시스템 로더를 검증하는 단계 및 상기 기계 상태를 검증하는 단계를 수행하는 검증기에 봉인되고,

상기 방법은, 상기 오퍼레이팅 시스템 로더 및 상기 기계 상태가 유효한 경우, 상기 키를 봉인해제하고 상기 키를 상기 오퍼레이팅 시스템 로더에 제공하는 단계를 더 포함하는 방법.

청구항 25.

제23항에 있어서,

상기 오퍼레이팅 시스템이 로딩된 후에, 상기 키로의 액세스에 문호가 되는 로그온 프로그램을 실행하는 단계를 더 포함하는 방법.

청구항 26.

제23항에 있어서,

상기 로그온 프로그램은 사용자가 인증 프로시저를 성공적으로 완료하는지의 여부에 따라 상기 오퍼레이팅 시스템의 컴포넌트가 상기 키를 사용하게 하거나 사용하지 못하게 하는 방법.

청구항 27.

제23항에 있어서,

상기 오퍼레이팅 시스템 로더를 검증하는 단계 및 상기 기계 상태를 검증하는 단계는 상기 오퍼레이팅 시스템 로더가 적어도 하나의 액션을 수행한 후에 수행되는 방법.

청구항 28.

제27항에 있어서,

상기 오퍼레이팅 시스템 로더를 검증하는 단계 및 상기 기계 상태를 검증하는 단계는, 상기 기계 상태의 재설정에 의해 상기 로더가 상기 오퍼레이팅 시스템을 올바르게 로딩하도록 기능하는 것이 방지되기 전의 시점에서 수행되는 방법.

청구항 29.

제23항에 있어서,

상기 기계는 물리적 기계를 포함하는 방법.

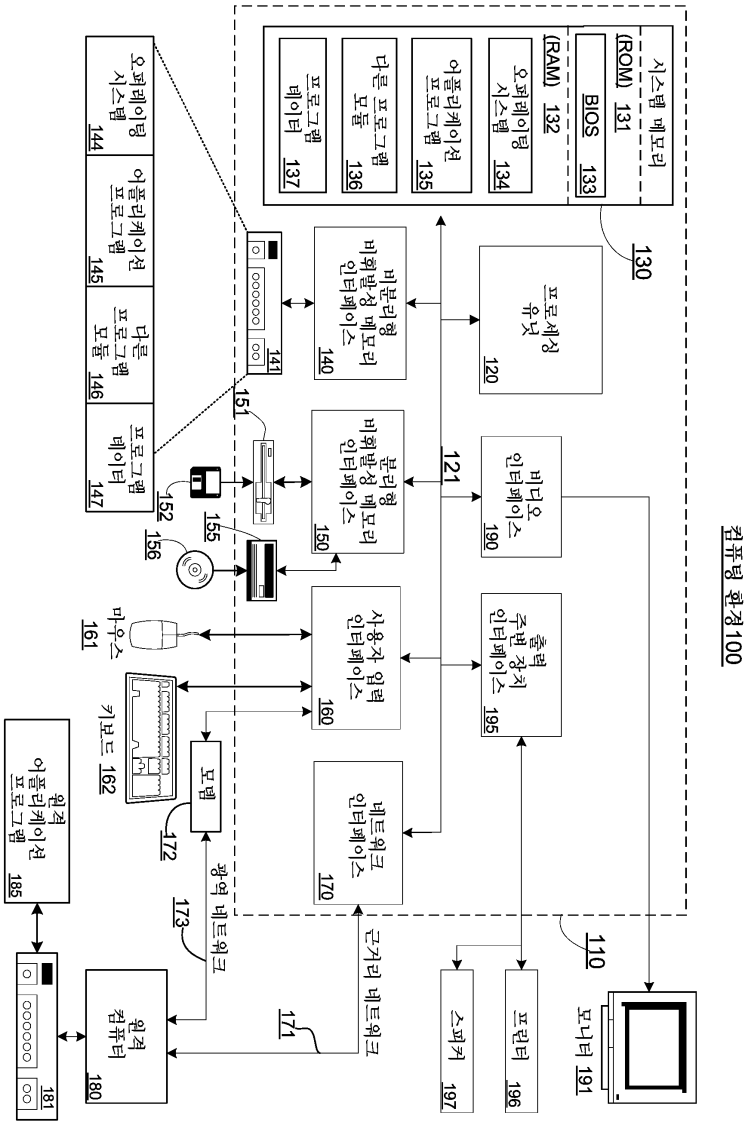
청구항 30.

제23항에 있어서,

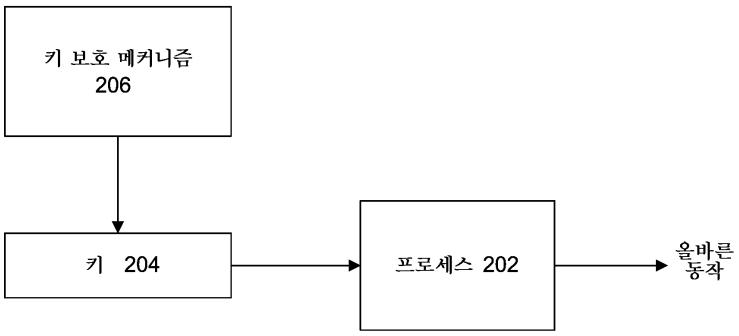
상기 기계는 가상 기계를 포함하는 방법.

도면

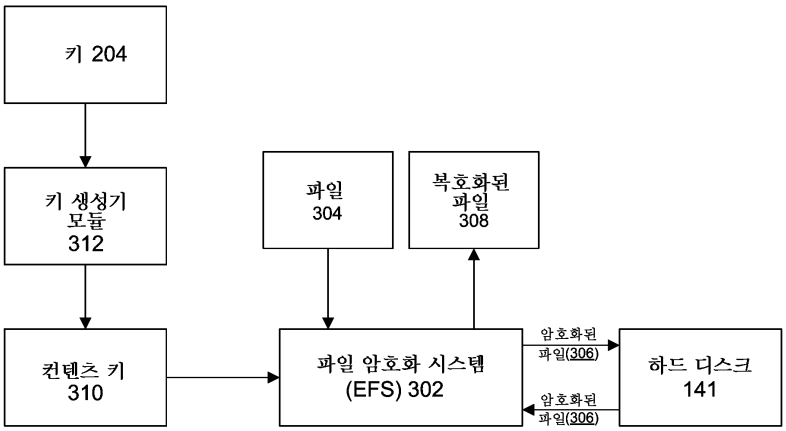
도면1



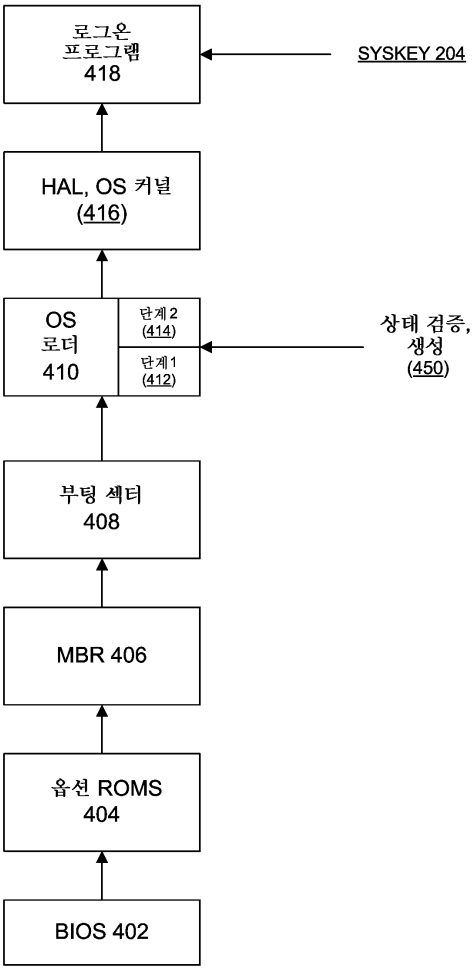
도면2


키 보호 메커니즘
206
키 204
프로세스 202
올바른
동작

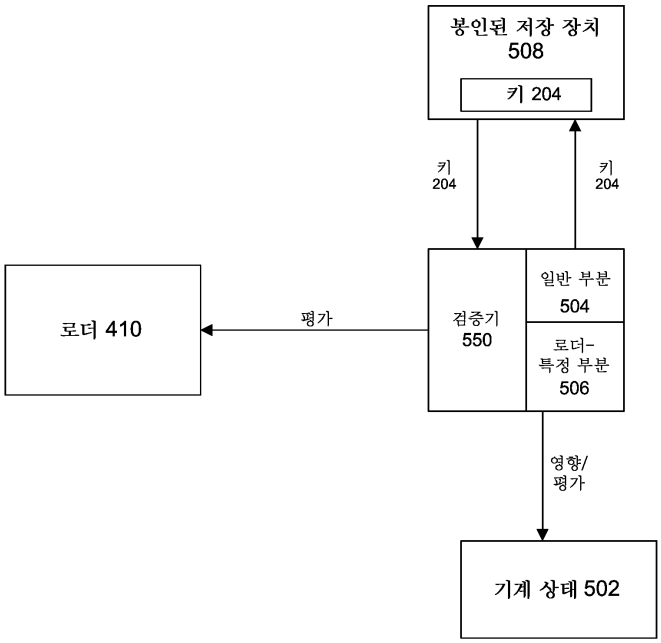
도면3


키 204
키 생성기
모듈
312
파일
304
복호화된
파일
308
컨텐츠 키
310
파일 암호화 시스템
(EFS) 302
암호화된
파일(306)
암호화된
파일(306)
하드 디스크
141

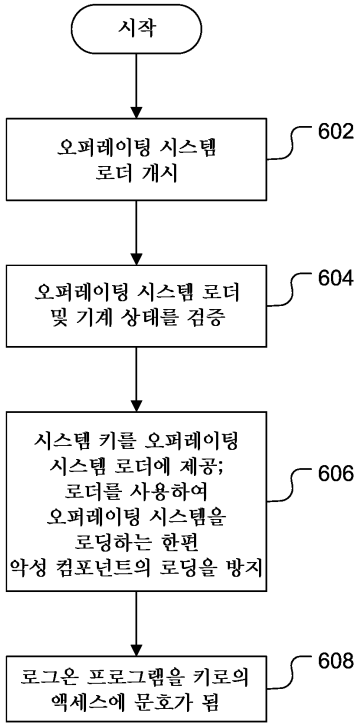
도면4


로그온
프로그램
418
SYSKEY 204
HAL, OS 커널
(416)
OS
로더
410
단계2
(414)
단계1
(412)
상태 검증,
생성
(450)
부팅 섹터
408
MBR 406
옵션 ROMS
404
BIOS 402

도면5


봉인된 저장 장치
508
키 204
키
204
키
204
로더 410
평가
검증기
550
일반 부분
504
로더-
특정 부분
506
영향/
평가
기계 상태 502

도면6


시작
오퍼레이팅 시스템
로더 개시
602
오퍼레이팅 시스템 로더
및 기계 상태를 검증
604
시스템 키를 오퍼레이팅
시스템 로더에 제공;
로더를 사용하여
오퍼레이팅 시스템을
로딩하는 한편
악성 컴포넌트의 로딩을 방지
606
로그온 프로그램을 키로의
액세스에 문호가 됨
608