(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0068201
(43) 공개일자 2009년06월25일

(51) Int. Cl.
G06F 21/00 (2006.01) *G06F 9/06* (2006.01)
(21) 출원번호 10-2009-7001576
(22) 출원일자 2009년01월23일
심사청구일자 없음
번역문제출일자 2009년01월23일
(86) 국제출원번호 PCT/US2007/079737
국제출원일자 2007년09월27일
(87) 국제공개번호 WO 2008/051679
국제공개일자 2008년05월02일
(30) 우선권주장
11/586,283 2006년10월25일 미국(US)

(71) 출원인
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
우텐, 데이비드 알.
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
홀트, 에릭
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
(뒷면에 계속)
(74) 대리인
양영준, 백만기

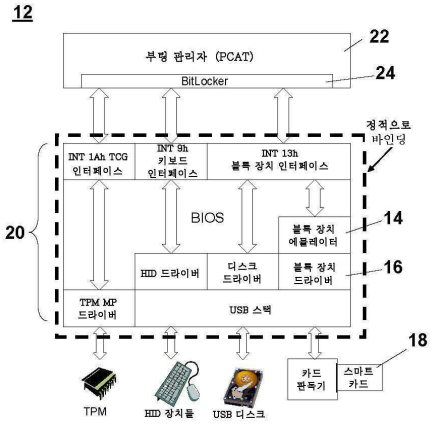
전체 청구항 수 : 총 20 항

(54) 플랫폼 인증 시스템, 블록 장치를 통해 플랫폼을 인증하기 위한 방법 및 블록 장치를 통해 플랫폼을 인증하기 위한 컴퓨터 실행가능 명령어들이 저장되어 있는 컴퓨터 판독가능 매체

(57) 요 약

시스템의 펌웨어는 스마트 카드와 같은 제2 장치들이 인증을 위해 사용될 수 있도록 구성된다. 예시적인 실시예에서, 제2 장치는 ISO 7816 명세에 따르는 CCID 스마트 카드이다. 시스템을 부팅하기 전에 스마트 카드는 시스템에 연결된 카드 판독기에 삽입된다. 펌웨어는 스마트 카드로부터의 인증 정보가 부팅 프로세스의 실행을 가능하게 하는 데 사용될 수 있도록 구성된 에뮬레이터 및 드라이버를 포함한다. 예시적인 실시예에서, 스마트 카드는 BITLOCKER™와의 사용을 위한 외부 키들을 포함한다. 제2 장치는 BIOS를 구현하는 시스템 및 EFI를 구현하는 시스템과 호환가능하다. 인증은 또한 데이터 저장소를 제공하지 않는 생체인식 장치 또는 이와 유사한 것 등을 통해 이루어질 수도 있다.

대 표 도 - 도1



(72) 발명자

톰, 스테판

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

우레케, 토니

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

슬레드즈, 댄

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

맥클버, 더글라스 엠.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

특허청구의 범위

청구항 1

플랫폼 인증 시스템(platform authenticating system)으로서,

BIOS 부분 및 EFI 부분 중 적어도 하나를 포함하고;

상기 BIOS 부분 및 EFI 부분은,

상기 시스템에 연결된 블록 장치를 검출하고, 상기 블록 장치 상의 파일을 열도록 구성된 블록 장치 에뮬레이터(block device emulator), 및

상기 블록 장치 상의 정보를 액세스하도록 구성된 블록 장치 드라이버(block device driver)

를 포함하는, 플랫폼 인증 시스템.

청구항 2

제1항에 있어서,

상기 블록 장치는 스마트 카드를 포함하는 플랫폼 인증 시스템.

청구항 3

제2항에 있어서,

상기 블록 장치는 CCID 스마트 카드를 포함하는 플랫폼 인증 시스템.

청구항 4

제1항에 있어서,

상기 블록 장치는 USB 호환성 블록 장치(USB compatible block device)를 포함하는 플랫폼 인증 시스템.

청구항 5

제1항에 있어서,

상기 블록 장치는 PCMCIA 호환성 블록 장치를 포함하는 플랫폼 인증 시스템.

청구항 6

제1항에 있어서,

상기 블록 장치 상의 상기 정보에 대한 액세스는 보호되는 플랫폼 인증 시스템.

청구항 7

제1항에 있어서,

상기 블록 장치는 생체인식 장치(biometric device)를 포함하는 플랫폼 인증 시스템.

청구항 8

제7항에 있어서,

생체인식 장치에서 액세스된 상기 정보는 상기 시스템의 부팅 프로세스의 실행을 가능하게 하는 토큰(token)을 포함하는 플랫폼 인증 시스템.

청구항 9

제7항에 있어서,

상기 블록 장치는 지문 인식기(fingerprint reader)를 포함하는 플랫폼 인증 시스템.

청구항 10

제1항에 있어서,

상기 블록 장치에서 액세스된 상기 정보는 상기 시스템의 부팅 프로세스의 실행을 가능하게 하는 키 관련 정보(keying information)를 포함하는 플랫폼 인증 시스템.

청구항 11

블록 장치를 통해 플랫폼을 인증하기 위한 방법으로서,

상기 플랫폼의 BIOS 부분 및 상기 플랫폼의 EFI 부분 중 적어도 하나를 통해 블록 장치를 검출하는 단계;

상기 검출된 블록 장치를 인증하는 단계; 및

시스템의 부팅 프로세스의 실행을 가능하게 하기 위한 정보를 상기 블록 장치로부터 검색하는 단계

를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 방법.

청구항 12

제11항에 있어서,

상기 블록 장치는 스마트 카드 및 CCID 스마트 카드 중 적어도 하나를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 방법.

청구항 13

제11항에 있어서,

상기 블록 장치는 USB 호환성 블록 장치 및 PCMCIA 호환성 블록 장치 중 적어도 하나를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 방법.

청구항 14

제11항에 있어서,

상기 블록 장치로부터 상기 정보를 검색하기 위한 인증에 대한 표시(indiciation)를 제공하는 단계를 더 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 방법.

청구항 15

제11항에 있어서,

상기 블록 장치는 생체인식 장치 및 지문 인식기 중 적어도 하나를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 방법.

청구항 16

컴퓨터 판독가능 매체로서,

플랫폼 시스템의 BIOS 부분 및 플랫폼의 EFI 부분 중 적어도 하나의 블록 장치 에뮬레이터를 통해 블록 장치를 검출하는 단계;

상기 블록 장치 에뮬레이터를 통해 상기 블록 장치 상의 파일을 여는 단계; 및

적어도 하나의 BIOS 부분 및 EFI 부분의 블록 드라이버를 통해, 상기 블록 장치 상의 정보를 액세스하는 단계

를 수행함으로써 블록 장치를 통해 플랫폼을 인증하기 위한 컴퓨터 실행가능 명령어들이 저장되어 있는 컴퓨터 판독가능 매체.

청구항 17

제16항에 있어서,

상기 블록 장치는 스마트 카드 및 CCID 스마트 카드 중 적어도 하나를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 컴퓨터 실행가능 명령어들이 저장되어 있는 컴퓨터 판독가능 매체.

청구항 18

제16항에 있어서,

상기 블록 장치는 USB 호환성 블록 장치 및 PCMCIA 호환성 블록 장치 중 적어도 하나를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 컴퓨터 실행가능 명령어들이 저장되어 있는 컴퓨터 판독가능 매체.

청구항 19

제16항에 있어서,

상기 블록 장치는 생체인식 장치 및 지문 인식기 중 적어도 하나를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 컴퓨터 실행가능 명령어들이 저장되어 있는 컴퓨터 판독가능 매체.

청구항 20

제16항에 있어서,

상기 블록 장치에서 액세스되는 상기 정보는 상기 시스템의 부팅 프로세스의 실행을 가능하게 하기 위한 키 관련 정보를 포함하는, 블록 장치를 통해 플랫폼을 인증하기 위한 컴퓨터 실행가능 명령어들이 저장되어 있는 컴퓨터 판독가능 매체.

명 세 서

기 술 분 야

<1> 본 기술분야는 일반적으로 컴퓨터 프로세싱에 관한 것이며 더 구체적으로는 보호된 파일(protected file)을 액세스하는 것에 관한 것이다.

배 경 기 술

<2> 현재의 프로세싱 시스템들은 데이터 보호를 제공한다. 예를 들어, MICROSOFT® CORPORATION의 제품인 BITLOCKER™ Drive Encryption(BITLOCKER™이라고도 함)은 다양한 WINDOWS® 운영 체제들과 사용가능한 데이터 보호 기능이다. BITLOCKER™는 다른 운영 체제를 부팅하거나 소프트웨어 해킹 도구(hacking tool)를 실행하는 사용자가, 파일 또는 시스템 보호를 좌절시키는 것 또는 보호된 드라이브에 저장된 파일들에 대한 오프라인 보기(offline viewing)를 수행하는 것을 막는다. BITLOCKER™는 사용자 데이터를 보호하고 WINDOWS® 운영 체제(예를 들어, WINDOWS® VISTA)를 실행하는 프로세서가 시스템이 오프라인인 중에 변형(tamper)되지 않았음을 보장한다.

<3> BITLOCKER™는 사용자가 PIN을 제공하거나, 키 관련 자료(keying material)를 포함하는 USB 플래시 드라이브를 삽입할 때까지 일반 부팅 프로세스를 잠금(lock)시킬 수 있다. 예를 들어, USB 플래시 장치는 부팅 프로세스 전에 프로세싱 시스템 내에 삽입될 수 있다. 부팅 프로세스 중에, USB 플래시 장치는 인식될 것이고 키 관련 자료는 그것으로부터 액세스될 수 있다. 그러나, 현재의 프로세싱 시스템들은 부팅 프로세스 중에 이러한 목적을 위해 스마트 카드 및 CCID(Integrated Circuit(s) Cards Interface Device) 스마트 카드와 같은 장치들을 인식하지 않는다.

<4> <발명의 요약>

<5> 이 발명의 요약은 아래의 실시예에 대한 상세한 설명에서 더 상세히 설명되는 개념들의 선택을 간략한 형태로 소개하기 위해 제공된다. 이 발명의 요약은 청구된 본 발명의 주요 특징 또는 본질적 특징들을 식별하기 위한 것이 아니며, 또한, 청구된 본 발명의 범위를 제한하는 데 사용되도록 하기 위한 것이 아니다.

<6> 플랫폼은 플랫폼의 펌웨어에 의해 지원되는 임의의 인증 메커니즘에 의해 통제되는(gated) 보호된 파일들로의 투명 액세스(transparent access)를 제공한다. 플랫폼은 PIN을 수용하는(accepting) 것이 가능하고, USB 플래

시 드라이브 또는 스마트 카드와 같은 다양한 장치들로부터 키들을 이용하는 것이 가능하고, 생체인식 장치(biometric device) 등과 같이 데이터 저장소를 제공하지 않는 다른 장치들과 호환가능하다. 예시적인 실시예에서, 플랫폼 펌웨어는 BITLOCKER™가 장치(예를 들어, CCID 스마트 카드)에 특별히 제공된 파일들(specially provisioned files)을 액세스하고/거나 장치의 보호된 BIOS(basic input/output system) 부분을 액세스할 수 있도록 한다.

실 시 예

<17> 플랫폼 인증을 위한 투명한 제2 요소, 또는 장치는, 예시적인 실시예에서 예를 들어, 스마트 카드, CCID 스마트 카드(Integrated Circuit(s) Cards Interface Device 스마트 카드), 생체인식 장치(예를 들어, 지문 인식기(fingerprint reader), 망막 스캐너 등), 또는 이들의 조합을 포함한다. 플랫폼의 인증을 지원하기 위한 투명한 제2 요소는 MICROSOFT® CORPORATION 제품인 BITLOCKER™ Drive Encryption(BITLOCKER™라고도 불림)에 적용되는 것으로 설명되지만 이에 제한되지 않는다. 제2 요소는 플랫폼의 사용자의 관점에서 투명하고, 따라서 사용자는 다양한 요소들 간의 차이의 세부사항들에 대한 지식을 가질 필요가 없다.

<18> BITLOCKER™는 블록 장치들에 저장되는 외부 키들을 사용하는 것이 가능하다. 블록 장치는, 예를 들어 하드 디스크 드라이브와 같이, 블록의 형태로 정보를 송신 및/또는 수신하는 장치이다. 블록 장치에 대한 액세스는 플랫폼의 펌웨어에 의해 제공된다. 플랫폼은 임의의 적절한 프로세서 또는 이와 유사한 것 등을 포함할 수 있다. 예를 들어, 적절한 플랫폼은 개인용 컴퓨터, 서버 컴퓨터, 핸드-헬드 또는 랩톱 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 시스템, 셋톱 박스, 프로그램가능한 가전 제품, 네트워크 PC, 미니 컴퓨터, 메인프레임 컴퓨터, 휴대용 장치, 예를 들어, MP3 플레이어, 워크맨 등과 같은 휴대용 뮤직 플레이어와 같은 휴대용 미디어 플레이어, 예를 들어, PDA(personal digital assistant), 휴대 전화, 휴대용 이메일 장치, 신 클라이언트(thin client), 휴대용 게임 장치 등과 같은 휴대용 컴퓨팅 장치, 예를 들어, TV, DVD 플레이어, 셋톱 박스, 모니터, 디스플레이 등과 같은 가전 제품, 예를 들어, 키오스크(kiosk), 가게에 설치된 음악 샘플링 장치, 현금 자동 입출금기(ATM), 금전 등록기 등과 같은 공용 컴퓨팅 장치, 휴대용이거나 자동차에 설치된 네비게이션 장치 및/또는 예를 들어, 주방기기, 자동차 조작 기기(예를 들어, 핸들) 등과 같은 일반적이지 않은(non-conventional) 컴퓨팅 장치, 또는 이들의 조합을 포함할 수 있지만 이에 제한되지는 않는다. 예시적인 실시예에서, 플랫폼의 펌웨어는 인증을 위해 BITLOCKER™가 스마트 카드, CCID 스마트 카드, 및/또는 생체인식 장치와 같은 제2 요소를 사용할 수 있도록 하는 능력을 제공하도록 구성된다. 따라서, BITLOCKER™는, 예를 들어, 하드 디스크와 같은 저장 장치로부터 파일들을 판독하는 것과 유사하게, 블록 장치 상의 특별히 제공된 파일들 또는 블록 장치의 보호된 BIOS 영역의 파일들을 판독한다.

<19> 여기에 적용가능한 예시적인 스마트 카드는 ISO 7816 명세에 따르는 스마트 카드이다. 이 명세는 스마트 카드의 속성 및 기능을 설명한다. 예시적인 실시예에서, 스마트 카드를 액세스하기 위해 사용되는 명령어들은, 단순 파일 시스템(simple file system)을 액세스하기 위한 기본 명령어들 및 파일 시스템 자체를 정의하는 ISO 7816 명세, 예를 들어, 섹션 ISO 7816-4에 따른다.

<20> 예시적인 실시예에서, BITLOCKER™ 키들(예를 들어, BITLOCKER™에 의해 이용되는 외부 키들)은 블록 장치의 파일 시스템에 저장된다. 블록 장치가 인증되면, 키들은 블록 장치로부터 검색되고 이어서 플랫폼의 부팅 프로세스 중에 BITLOCKER™에 의해 사용된다. 예시적인 실시예에서, 블록 장치에 대한 액세스는 판독 전용 액세스(read-only access)이다. 다른 실시예에서, 블록 장치 상의 파일들은 보호된다. 예를 들어, 블록 장치 상의 파일에 대한 액세스는 PIN, 또는 이와 유사한 것이 제공될 때까지 허용되지 않을 수 있다. 또는, 다른 예로서, 블록 장치 상의 미리 결정된 파일들만이 플랫폼에 의해 액세스가능하도록 선택된다. 따라서, 블록 장치를 잃어버리거나 도난당하면, 블록 장치 상의 정보는 보호되고, 권한이 없는 엔티티에 의해 액세스될 수 없다.

<21> 도 1은 시스템(12)의 BIOS(basic input/output system) 부분(20)을 통해 블록 장치(18)를 액세스하기 위한 예시적인 시스템(12)의 기능 블록도이다. 도 1에 도시된 바와 같이, 시스템(12)은 PCAT(PC advanced technology) 호환성 부팅 관리자(compatible boot manager)(22), BITLOCKER™ 소프트웨어(24), 및 BIOS 부분(20)을 포함한다. BIOS 부분(20)은 블록 장치 에뮬레이터 부분(14) 및 블록 장치 드라이버 부분(16)을 포함한다. 블록 장치 에뮬레이터(14)는 블록 장치(18)를 검출하고 블록 장치(18) 상의 파일을 열도록 구성된다. BIOS 부분(20)은 블록 장치(18)에 저장된 정보를 액세스하도록 구성된 블록 장치 드라이버(16)를 포함한다.

<22> BIOS 부분(20)은 시스템(12)을 포함하는 플랫폼의 부팅 프로세스를 제어하기 위한 펌웨어를 포함한다. 예시적인 실시예에서, BIOS 부분(20)은 예를 들어 신뢰된 플랫폼 모듈(trusted platform module, TPM)과 같은 신뢰된 컴퓨팅 그룹(trusted computing group, TCG)에 대한 인터페이스를 다루기 위한 인터럽트 핸들러(interrupt handler)(예를 들어, 인터럽트 1A 16진수, INT 1Ah), 예를 들어 키보드와 같은 휴먼 인터페이스 장치(human interface device, HID)에 대한 인터페이스를 다루기 위한 인터럽트 핸들러(예를 들어, INT 9h), 예를 들어 USB(universal serial bus) 호환성 장치(예를 들어, USB 디스크, USB 카드 판독기)와 같은 블록 장치 인터페이스에 대한 인터페이스를 다루기 위한 인터럽트 핸들러(예를 들어, INT 13h)를 포함한다. 구체적 인터럽트 핸들러들에 대한 도 1의 도시는 예시적이며 임의의 인터페이스가 사용될 수 있음이 강조된다.

<23> BIOS 부분(20)은 HID 및 저장 블록 장치(storage block device)들과 같은 기본 장치 및 USB 컨트롤러를 지원한다. 예시적인 실시예에서, BIOS 부분(20)은, 예를 들어, 스마트 카드(18), 또는 생체인식 장치(도 1에 도시되지 않음)와 같은 장치들과 호환가능하도록 블록 장치 에뮬레이터(14) 및 블록 장치 드라이버(16)를 포함하도록 구성된다. 예시적인 실시예에서, 블록 장치 드라이버(16)는 적어도 부분적으로, 예를 들어, CCID에 대한 명세(Specification for Integrated Circuit(s) Cards Interface Devices), 리비젼 1.1(Revision 1.1)과 같은 CCID에 대한 명세에 따라 구성되고 동작한다. CCID에 대한 명세에 설명된 바와 같이 전체 장치 클래스(Device Class)를 구현할 필요가 없고, 오히려 장치 클래스의 서브셋(subset)이 적절하다. 예를 들어, 카드 판독기와의 기본 상호작용(basic interaction) 및 스마트 카드(18)와의 통신, 예를 들어, 스마트 카드(18)에 전력을 제공하는 것, 스마트 카드(18)로부터 전력을 제거하는 것, 및 스마트 카드(18)로 및 스마트 카드(18)로부터 명령어 및 응답 APDU(Application Protocol Data Unit)들을 릴레이하는(relaying) 것을 가능하게 하기 위해 이용되는 구성요소들이 적절하다.

<24> 예시적인 실시예에서, 카드 판독기는 PCMCIA(Personal Computer Memory Card International Association, 또는 Peripheral Component Microchannel Interconnect Architecture) 인터페이스를 통해 스마트 카드(18)와 호환가능하다. 이 실시예는 휴대 컴퓨터 및 이와 유사한 것 등에 대해 이로운 사용(advantageous use)을 제공한다. 스마트 카드(18)는 카드, 전자 열쇠(key fob), 또는 임의의 적절한 구성의 형태일 수 있다.

<25> 예시적인 실시예에서, 도 2에 도시된 바와 같이, 블록 장치(18)는 지문 인식기, 망막 스캐너, 또는 이와 유사한 것 등과 같은 생체인식 장치를 포함한다. 생체인식 장치는 통상적으로 데이터 저장소를 제공하지 않는다. 블록 장치(18)가 데이터 저장소를 제공하지 않는 생체인식 장치 또는 다른 제2 요소를 포함하는 실시예에서, BIOS 부분(20)의 펌웨어는 블록 장치(18)의 인증을 수행한다. 블록 장치(18)에 대한 인증이 성공적이면, 블록 장치(18)에 의해 토큰이 제공될 수 있고, 이는 예를 들어 BITLOCKER™를 시작함으로써 부팅 프로세스를 계속하는 데 필요한 키들을 포함하는 플랫폼의 안전한 영역을 액세스하도록 사용될 수 있다. 이 실시예에서, 플랫폼의 안전한 영역에 저장된 정보는 블록 장치(18)에 대한 사전의 성공적인 인증없이 판독될 수 없다.

<26> 도 1 및 도 2에 도시된 바와 같이, 플랫폼의 BIOS 부분(20)은 정적으로 바인딩되어(statically bound) 있다. 다시 말해, 바인딩(binding), 즉, 값들을 식별자들과 관련시키는 프로세스는 플랫폼의 부팅 프로세스 또는 운영 체제가 시작되기 전에 완수된다. 따라서, 블록 장치 에뮬레이터(14) 및 블록 장치 드라이버(16)는 모든 유형의 블록 장치(18)들에 대해 고정된다.

<27> 도 3은 시스템(28)의 EFI(extensible firmware interface)(26)를 통해 블록 장치(18)에 액세스하기 위한 예시적인 시스템(28)의 기능 블록도이다. 시스템(28)은 상술된 시스템(12)과 유사하게 동작하고, 차이는 시스템(12)의 BIOS 구현과 시스템(28)의 EFI 구현이다. EFI는 BIOS를 대체하도록 이용되는 것으로 알려져 있다. BIOS와 유사하게, EFI는 운영 체제의 부팅 프로세스를 제어하고 운영 체제와 하드웨어 사이의 인터페이스를 제공하는 데 책임이 있다. EFI는 동적으로 바인딩된다(dynamically bound). 다시 말해, 바인딩은 부팅 프로세스 중에 성취된다. 따라서, 블록 장치 에뮬레이터(14) 및 블록 장치 드라이버(16)는 개별적으로 구현될 수 있고 하나의 정적 유닛에 바인딩될 필요가 없다. 블록 장치 드라이버(16)는 EFI 부팅 시간에 로드될 수 있다.

<28> 도 3에 도시된 바와 같이, 시스템(28)은 EFI 호환성 부팅 관리자(30), BITLOCKER™ 소프트웨어(24), 및 EFI 부분(26)을 포함한다. EFI 부분(26)은 블록 장치 에뮬레이터 부분(14) 및 블록 장치 드라이버 부분(16)을 포함한다. 블록 장치 에뮬레이터(14)는 블록 장치(18)를 검출하고 블록 장치(18) 상의 파일을 열도록 구성된다. EFI 부분(26)은 블록 장치(18)에 저장된 정보를 액세스하도록 구성된 블록 장치 드라이버(16)를 포함한다.

<29> EFI 부분(26)은 시스템(12)을 포함하는 플랫폼의 부팅 프로세스를 제어하기 위한 펌웨어를 포함한다. 예시적인 실시예에서, EFI 부분(26)은 예를 들어 신뢰된 플랫폼 모듈(trusted platform module, TPM)과 같은 신뢰된 컴

퓨팅 그룹(trusted computing group, TCG)에 대한 인터페이스를 다루기 위한 인터럽트 핸들러, 예를 들어 키보드와 같은 휴먼 인터페이스 장치(human interface device, HID)에 대한 인터페이스를 다루기 위한 인터럽트 핸들러, USB(universal serial bus) 호환성 장치(예를 들어, USB 디스크, USB 카드 판독기)와 같은 블록 장치 인터페이스에 대한 인터페이스를 다루기 위한 인터럽트 핸들러를 포함한다. 구체적 인터럽트 핸들러들에 대한 도 1의 도시는 예시적이며 임의의 인터페이스가 사용될 수 있음이 강조된다.

<30> EFI 부분(26)은 HID 및 저장 블록 장치들과 같은 기본 장치 및 USB 컨트롤러를 지원한다. 예시적인 실시예에서, EFI 부분(26)은 예를 들어 스마트 카드(18), 또는 생체인식 장치(도 3에 도시되지 않음)와 같은 장치들과 호환가능하도록 블록 장치 에뮬레이터(14) 및 블록 장치 드라이버(16)를 포함하도록 구성된다. 예시적인 실시예에서, 블록 장치 드라이버(16)는 적어도 부분적으로, 예를 들어, CCID에 대한 명세(Specification for Integrated Circuit(s) Cards Interface Devices), 리비젼 1.1(Revision 1.1)과 같은 CCID에 대한 명세에 따라 구성되고 동작한다. CCID에 대한 명세에 설명된 바와 같이 전체 장치 클래스를 구현할 필요가 없고, 오히려 장치 클래스의 서브셋이 적절하다. 예를 들어, 카드 판독기와의 기본 상호작용 및 스마트 카드(18)와의 통신, 예를 들어, 스마트 카드(18)에 전력을 제공하는 것, 스마트 카드(18)로부터 전력을 제거하는 것, 및 스마트 카드(18)로 및 스마트 카드(18)로부터 명령어 및 응답 APDU(Application Protocol Data Unit)들을 릴레이하는 것을 가능하게 하기 위해 이용되는 구성요소들이 적절하다.

<31> 예시적인 실시예에서, 카드 판독기는 PCMCIA(Personal Computer Memory Card International Association, 또는 Peripheral Component Microchannel Interconnect Architecture) 인터페이스를 통해 스마트 카드(18)와 호환가능하다. 이 실시예는 휴대 컴퓨터 및 이와 유사한 것 등에 대해 이로운 사용(advantageous use)을 제공한다. 스마트 카드(18)는 카드, 전자 열쇠, 또는 임의의 적절한 구성의 형태일 수 있다.

<32> 예시적인 실시예에서, 도 4에 도시된 바와 같이, 블록 장치(18)는 지문 인식기, 망막 스캐너, 또는 이와 유사한 것 등과 같은 생체인식 장치를 포함한다. 생체인식 장치는 통상적으로 데이터 저장소를 제공하지 않는다. 블록 장치(18)가 데이터 저장소를 제공하지 않는 생체인식 장치 또는 다른 제2 요소를 포함하는 실시예에서, EFI 부분(26)의 펌웨어는 블록 장치(18)에 대한 인증을 수행한다. 블록 장치(18)에 대한 인증이 성공적이면, 블록 장치(18)에 의해 토큰이 제공될 수 있고, 이는 예를 들어 BITLOCKER™를 시작함으로써 부팅 프로세스를 계속하는데 필요한 키들을 포함하는 플랫폼의 안전한 영역을 액세스하도록 사용될 수 있다. 이 실시예에서, 플랫폼의 안전한 영역에 저장된 정보는 블록 장치(18)에 대한 사전의 성공적인 인증없이 판독될 수 없다.

<33> 다른 예시적인 실시예에서, EFI는 하드 디스크, 또는 이와 유사한 것으로부터 BIOS에 부팅된다.

<34> 운영 체제가 플랫폼에 로드되기 전에 블록 장치가 인증된다. 예시적인 실시예에서, BIOS가 구현되면, 블록 장치는 BIOS 상의 인터럽트 19h가 호출되기 전에 인증된다. 다시 말해, 블록 장치는, 부팅 프로세스 중에, BIOS가 운영 체제를 로드하기 시작하기 전에 인증된다. 예시적인 실시예에서, EFI가 구현되면, 블록 장치는 EFI에서 부팅 관리자가 호출(invoke)되기 전에 인증된다. 블록 장치는 임의의 적절한 수단에 의해 인증될 수 있으며, 예를 들어, PIN 또는 이와 유사한 것의 사용을 통해 인증될 수 있다. 블록이 성공적으로 인증된 후, 플랫폼의 메모리에 가상 블록 장치가 생성된다.

<35> 플랫폼 메모리에 가상 블록 장치를 생성하기 위해, 부팅 프로세스가 실행을 시작/계속하도록 하는 정보를 포함하는 위치들에 대해 블록 장치의 저장소가 검색된다. 블록 장치 상의 저장소의 구조는 임의의 적절한 구조를 포함할 수 있고, 부팅 프로세스의 실행을 가능하게 하는 정보를 포함하는 블록 장치 상의 저장소의 위치 또는 위치들은 임의의 적절한 방식으로 식별될 수 있다.

<36> 도 5는 ISO 7816 명세에 따르는 블록 장치의 예시적인 파일 시스템(40)에 대한 도시이다. 파일 시스템(40)은 예시적이며, 임의의 적절한 파일 시스템이 블록 장치에 구현될 수 있음이 강조된다. 예시적인 실시예에서, 블록 장치의 파일 시스템(40)은 적어도 하나의 마스터 파일(master file)(32)(도 5에 MF로 도시됨), 적어도 하나의 전용 파일(dedicated file)(34)(도 5에 DF로 도시됨), 및 적어도 하나의 기본 파일(elementary file)(도 5에 EF로 도시됨)을 포함한다. 간략함을 위해 각 유형의 파일 중 하나만이 레이블된다(32, 34, 및 36). 파일 시스템(40)의 계층은 전용 파일(들) 및 기본 파일(들)이 마스터 파일(32) 하위에 있도록 되어 있다. 마스터 파일(32)은 루트 파일(root file)이다. 전용 파일은 파일 제어 정보를 포함한다. 전용 파일은, 선택적으로, 할당이 가능한 메모리에 대한 표시를 포함할 수 있다. 전용 파일은 기본 파일의 부모(parent)일 수 있고/거나 전용 파일은 기본 파일(elementary file)일 수 있다.

<37> 예시적인 실시예에서, 기본 파일은 부팅 프로세스의 실행이 가능하도록 하는 정보를 포함한다는 표시를 포함한

다. 예를 들어, 기본 파일은 BITLOCKER™에 의해 사용되는 키들을 포함한다는 표시를 포함할 수 있다. 예시적인 실시예에서, 도 5에 도시된 바와 같이, 기본 파일은 VFAT(virtual file attribute table)로서 지정되어, 자신이 부팅 프로세스의 실행을 가능하게 하는 정보를 포함한다는 것을 나타낸다. VFAT의 ID를 갖는 기본 파일에 포함된 정보의 유형의 예가 블록(38)에 도시된다. 다른 정보들 중에서, BITLOCKER™에 의해 이용가능한 키들은 LFN(Logical File Name): E5E1A420-0134-4677-88B1-855E9DC200F7.BEK, 및 LFN: AE324B14-5264-E32A-9A23-225AB6823A32.BEK로 나타난다.

<38> 도 6은 플랫폼 메모리에 가상 블록 장치를 생성하기 위한 예시적인 프로세스의 흐름도이다. 단계(40)에서, 플랫폼 내의 메모리는 할당되고 임의의 필요한 시스템 구조들은 초기화된다. 예시적인 실시예에서, 블록 장치는 부팅 프로세스의 실행 전에 플랫폼에 연결된다. 예시적인 실시예에서, 블록 장치 상의 파일 시스템은 판독 전용 FAT로서 구현되며, 플랫폼이 블록 장치로부터 부팅되는 것을 막기 위해 0과 동일한 부팅 섹터(boot sector)를 갖는다. 운영 체제를 플랫폼에 로드하기 전에, 단계(42)에서 블록 장치의 마스터 파일이 플랫폼 메모리에 로드된다. BIOS 또는 EFI와 같은 펌웨어는 단계(44)에서 (ID 번호를 나타내는) ID VFAT를 갖는 기본 파일들에 대해 블록 장치 상의 마스터 파일 및 하위 전용 파일들을 반복적으로(recursively) 검색한다. 만약 단계(46)에서 ID VFAT를 갖는 기본 파일들이 찾아지지 않으면, 단계(58)에서 사용되지 않은 리소스들이 자유롭게 되고(freed) 프로세스가 종료된다. 만약 단계(46)에서 ID VFAT를 갖는 기본 파일들이 찾아지면, 모든 이러한 기본 파일들은 단계(48)에서 플랫폼 메모리에 로드된다. ID VFAT를 갖는 기본 파일을 찾는 것은, 블록 장치가 부팅 액세스를 위해 준비되었고 가상 블록 장치의 생성이 진행될 수 있다는 표시이다.

<39> 단계(50)에서 기본 파일이 열리고 구문 분석(parse)된다. ISO 7816 명세에 따른 예시적인 실시예에서, 기본 파일은 5-바이트 식별자로 제한된다. 따라서, BITLOCKER™가 5비트보다 긴 이름을 갖는 외부 키들을 이용할 수 있도록 하기 위해 가상 블록 장치에 포함되어야 하는 모든 파일들의 카탈로그를 블록 장치에 유지한다. 예를 들어 BITLOCKER™ 키들과 같은 정보가 단계(52)에서 구문 분석된 기본 파일로부터 추출된다. 단계(54)에서, 예를 들어 BITLOCKER™ 키들과 같은 추출된 정보는 플랫폼 메모리의 가상 블록 장치의 각 적절한 파일에 추가된다. ID VFAT를 갖는 기본 파일들이 더 존재하면(단계(56)), 프로세스는 단계(50)로 진행되고 상술된 바와 같이 계속된다. ID VFAT를 갖는 기본 파일들이 더 존재하지 않으면(단계(56)), 단계(58)에서, 사용되지 않은 리소스들이 자유롭게 되고 프로세스는 종료된다. EF에서 VFAT로 식별되지 않은, 블록 장치에서 찾아진 파일들은 액세스되거나 판독되지 않고 따라서 노출되지 않는다.

<40> 가상 블록 장치는 플랫폼 메모리에 동적으로 생성된다. 가상 블록 장치는 부팅 프로세스 중에(예를 들어, 플랫폼의 전원이 켜질 때(powered up)) 한번 생성된다. 블록 장치가 플랫폼으로부터 연결이 해제(uncoupled)되더라도 가상 블록 장치의 파일들은 플랫폼 메모리에 남아있다. 그러나, 가상 블록 장치는 부팅-전(pre-boot) 시간 중에 존재하고 플랫폼 운영 체제가 로드되면 파괴된다.

<41> 가상 블록 장치의 폴더 계층은 블록 장치의 폴더들의 계층을 따른다. ID VFAT를 갖는 기본 파일들만 로드되므로 블록 장치의 원하지 않는 파일들(unwanted files)은 가상 블록 장치에 나타나지 않는다. 따라서, 실제로 데이터로 채워진(populated) 블록들을 위한 플랫폼 메모리만이 할당된다. 가상 블록 장치를 위한 플랫폼 메모리의 예시적인 할당에 대한 도시가 도 7에 도시된다. 도 7에 도시된 바와 같이, 플랫폼 메모리는 부팅 레코드 섹터(boot record sector)(60), 파일 할당 테이블 섹터(62), 백업 파일 할당 테이블 섹터(64), 루트 디렉터리(root directory)(66), 및 적어도 하나의 데이터 저장 영역 섹터(68)에 대해 할당된다. 데이터 저장 영역(68)은, 예를 들어 BITLOCKER™ 키와 같이, 부팅 프로세스가 실행되도록 하는 데 사용되는 정보를 포함할 수 있다. 예시적인 실시예에서, 각 섹터는 512 바이트를 포함한다. 플랫폼 메모리는 임의의 적절한 방식으로 할당될 수 있으며, 예를 들어, 연속적으로(contiguously) 할당될 수 있다. 예시적인 실시예에서, 단계(66)가 수행되고 다른 모든 블록들은 부팅 애플리케이션으로부터 요청될 때 "즉시(on the fly)" 컴퓨팅된다.

<42> 상술된 바와 같이, 블록 장치는 CCID 스마트 카드를 포함할 수 있다. 도 8은 CCID 스마트 카드를 액세스하기 위한 예시적인 프로세스의 흐름도이다. 도 8에 도시된 프로세스는 임의의 적절한 블록 장치에 적용될 수 있음이 강조된다. 단계(70)에서, CCID 카드 판독기가 검출되었는지가 결정된다. 상술된 바와 같이,카드 판독기가 검출되지 않으면(단계(70)), 단계(90)에서, BIOS가 구현된 경우 인터럽트 19h가 호출되고 EFI가 구현된 경우 부팅 관리자가 로드된다. 카드 판독기가 검출되면(단계(70)), 단계(72)에서, 카드 판독기에 연결된 스마트 카드가 검출되었는지가 결정된다. 스마트 카드가 검출되지 않으면(단계(72)) 프로세스는 단계(90)로 진행된다. 스마트 카드가 검출되면(단계(72)), 단계(74)에서, 스마트 카드가 ISO 7816-4 또는 임의의 다른 적절한 프로토콜

과 컴플라이언트(compliant)한지가 결정된다. 스마트 카드가 컴플라이언트 하지 않으면(단계(74)) 프로세스는 단계(90)로 진행된다. 스마트 카드가 컴플라이언트 하면(단계(74)), 단계(76)에서, 전용 부팅 파일들이 스마트 카드로부터 플랫폼 메모리에 로드된다.

<43> 단계(78)에서, DF 부팅 파일들이 예를 들어 PIN에 의해 보호되는지 결정된다. DF 부팅 파일들이 보호되지 않으면(단계(78)), 단계(88)에서, 플랫폼 메모리에서의 가상 블록 장치의 생성은 초기화된다. DF 부팅 파일들이 보호되면(단계(78)), 단계(80)에서, PIN 또는 이와 유사한 것 등에 대해 프롬프트가 렌더링된다. PIN 또는 이와 유사한 것 등이 입력되도록 하는 데 사용되는 사용자 인터페이스(UI)는 IPL(initial program load), 또는 부팅, 코드가 개시되기 전에 이루어진다. 키보드, 디스플레이, 및/또는 다른 UI 장치들에 대한 제어가 IPL 코드로 전달되면, 인증 프롬프트는 렌더링되지 않을 수 있다. 예시적인 실시예에서, 플랫폼에 전력을 제공하기 전에 스마트 카드가 카드 판독기와 연결(예를 들어, 삽입)될 것이 기대된다.

<44> PIN 또는 이와 유사한 것 등은 단계(82)에서 수신된다. 단계(84)에서, PIN 또는 이와 유사한 것 등은 스마트 카드를 잠금 해제(unlock)하기 위해 사용된다. 단계(86)에서, 스마트 카드가 잠금 해제되었는지 결정된다. 스마트 카드가 잠금 해제되었으면(단계(86)) 프로세스는 단계(88)로 진행된다. 스마트 카드가 잠금 해제되지 않았으면(단계(86)) 프로세스는 단계(80)로 진행된다.

<45> 가상 블록 장치가 성공적으로 생성된 후, IPL 코드는 가상 블록 장치를 열거(enumerate)하고 BITLOCKER™ 외부 키 또는 이와 유사한 것 등에 대한 검색에 포함시킨다. 키(들)가 부팅 관리자에 의해 로드되고 키(들)가 암호화된 불륨(volume)(들)을 성공적으로 잠금 해제하고 나면, 제어는 운영 체제 로더(operating system loader)에 전달된다. 예시적인 실시예에서, 가상 블록 장치의 플랫폼 메모리 내의 저장 위치는 보호되거나, 보존되거나, 또는 운영 체제 로더에 알려지지 않는다. 따라서, 운영 체제 파일들은 로드되고 결국 가상 블록 장치의 저장 공간에 겹쳐 쓸(overwrite) 수 있다.

<46> 플랫폼의 인증을 지원하기 위한 투명한 제2 요소의 다양한 실시예들은 컴퓨팅 장치에 실행가능하다. 도 9 및 아래의 논의는 이러한 컴퓨팅 장치가 구현될 수 있는 적절한 컴퓨팅 환경에 대한 간략한 일반적인 설명을 제공한다. 요구되지는 않지만, 플랫폼의 인증을 지원하기 위한 투명한 제2 요소의 다양한 양상들은 클라이언트 워크스테이션 또는 서버와 같은 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행가능 명령어들과 관련하여 일반적으로 설명될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형들을 구현하는 루틴, 프로그램, 객체, 컴포넌트, 데이터 구조 등을 포함한다. 또한, 투명한 제2 요소를 통한 플랫폼 인증은 핸드-헬드 장치, 멀티 프로세서 시스템, 마이크로프로세서 기반 또는 프로그램가능한 가전 제품, 네트워크 PC, 미니 컴퓨터, 메인프레임 컴퓨터, 및 이와 유사한 것 등을 포함하는 다른 컴퓨터 시스템 구성과 함께 실행될 수 있다. 또한, 투명한 제2 요소를 통한 플랫폼의 인증은 통신 네트워크를 통해 링크된 원격 프로세싱 장치들에 의해 태스크들이 수행되는 분산된 컴퓨팅 환경에서 또한 실행될 수 있다. 분산된 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치들 모두에 위치할 수 있다.

<47> 컴퓨터 시스템은 개략적으로 세 개의 컴포넌트 그룹으로 나눠질 수 있다: 하드웨어 컴포넌트, 하드웨어/소프트웨어 인터페이스 시스템 컴포넌트, 및 애플리케이션 프로그램 컴포넌트("사용자 컴포넌트" 또는 "소프트웨어 컴포넌트"라고도 불림). 컴퓨터 시스템의 다양한 실시예에서, 하드웨어 컴포넌트는, 여럿 중에서도 특히, 중앙 처리 장치(CPU)(621), 메모리(ROM(664) 및 RAM(625) 모두), 기본 입/출력 시스템(BIOS)(666), 및 키보드(640), 마우스(642), 모니터(647), 및/또는 프린터(도시되지 않음)와 같은 다양한 입력/출력(I/O) 장치를 포함할 수 있다. 하드웨어 컴포넌트는 컴퓨터 시스템을 위한 기본 물리적 인프라를 포함한다.

<48> 애플리케이션 프로그램 컴포넌트는 컴파일러, 데이터베이스 시스템, 워드 프로세서, 비즈니스 프로그램, 비디오 게임 기타 등등을 포함하는 다양한 소프트웨어 프로그램을 포함하지만 이에 제한되지는 않는다. 애플리케이션 프로그램은 문제를 해결하고, 솔루션을 제공하고, 다양한 사용자들(기계, 다른 컴퓨터 시스템, 및/또는 최종 사용자)을 위해 데이터를 프로세스하기 위해 컴퓨터 리소스들이 사용되게 하는 수단을 제공한다. 예시적인 실시예에서, 애플리케이션 프로그램은 상술된 바와 같이 투명한 제2 요소를 통한 플랫폼 인증과 관련된 기능들을 수행하며, 예를 들어 블록 장치를 검출하는 것, 블록 장치 상의 파일을 여는 것, 블록 장치 상의 파일들을 검색하는 것, 플랫폼 메모리에 가상 블록 장치를 생성하는 것, 블록 장치를 잠금 해제하기 위해 PIN 또는 이와 유사한 것 등을 이용하는 것, 및 부팅 프로세스를 시작하는 것이 있다.

<49> 하드웨어/소프트웨어 인터페이스 시스템 컴포넌트는, 대부분의 경우 셸(shell) 및 커널을 포함하는 운영 체제를 포함한다(그리고, 몇몇 실시예에서, 이것만으로 구성됨). "운영 체제"(OS)는 애플리케이션 프로그램과 컴퓨터 하드웨어 사이에서 매개물(intermediary)로서 동작하는 특별한 프로그램이다. 하드웨어/소프트웨어 인터페이스

시스템 컴포넌트는 또한 VMM(virtual machine manager), CLR(Common Language Runtime) 또는 이의 기능적 동등물(functional equivalent), JVM(Java Virtual Machine) 또는 이의 기능적 동등물, 또는 컴퓨터 시스템의 운영 체제 대신의 또는 운영 체제에 추가된 다른 이러한 소프트웨어 컴포넌트들을 포함할 수 있다. 하드웨어/소프트웨어 인터페이스 시스템의 목적은 사용자가 애플리케이션 프로그램을 실행할 수 있는 환경을 제공하는 것이다

<50> 하드웨어/소프트웨어 인터페이스 시스템은 일반적으로 스타트-업(startup) 시에 컴퓨터 시스템에 로드되고, 그 후 컴퓨터 시스템 내의 모든 애플리케이션 프로그램들을 관리한다. 애플리케이션 프로그램은 애플리케이션 프로그램 인터페이스(API)를 통해 서비스들을 요청함으로써 하드웨어/소프트웨어 인터페이스 시스템과 상호작용한다. 몇몇 애플리케이션 프로그램들은 최종 사용자들이 명령어 또는 그래픽 사용자 인터페이스(GUI)와 같은 사용자 인터페이스를 통해 하드웨어/소프트웨어 인터페이스 시스템과 상호작용할 수 있도록 한다.

<51> 하드웨어/소프트웨어 인터페이스 시스템은 통상적으로 애플리케이션을 위한 다양한 서비스들을 수행한다. 복수의 프로그램들이 동일한 시간에 실행될 수 있는 멀티태스킹 하드웨어/소프트웨어 인터페이스 시스템에서, 하드웨어/소프트웨어 인터페이스 시스템은 어느 애플리케이션들이 어떤 순서로 실행되어야 하고, 다음 순서를 위해 다른 애플리케이션으로 전환하기 전에 각 애플리케이션에 대해 얼마나 많은 시간이 허용되어야 하는지를 결정한다. 하드웨어/소프트웨어 인터페이스 시스템은 또한 복수 애플리케이션들 간에서의 내부 메모리의 공유를 관리하고, 하드 디스크, 프린터, 및 전화 접속 포트(dial-up port)와 같은 부착된 하드웨어 장치로의 또는 이로부터의 입력 및 출력을 다룬다. 하드웨어/소프트웨어 인터페이스 시스템은 또한 동작들의 상태 및 발생했을 수 있는 임의의 에러들에 관한 메시지들을 각 애플리케이션(그리고, 특정 경우에는 최종 사용자에게)에 전송한다. 하드웨어/소프트웨어 인터페이스 시스템은 또한 일괄 작업(batch job)(예를 들어, 인쇄)에 대한 관리를 오프로드(offload)하여, 시작 애플리케이션(initiating application)이 이 작업으로부터 자유로워지고 다른 프로세싱 및/또는 동작들을 다시 시작할 수 있도록 한다. 병렬 프로세싱을 제공할 수 있는 컴퓨터에서, 하드웨어/소프트웨어 인터페이스 시스템은 또한 프로그램을 나누는 것을 관리하여, 동시에 둘 이상의 프로세서에서 실행되도록 한다.

<52> 하드웨어/소프트웨어 인터페이스 시스템 셸("셸"이라 불림)은 하드웨어/소프트웨어 인터페이스 시스템에 대한 대화형(interactive) 최종 사용자 인터페이스이다. (셸은 또한 "명령 인터프리터(command interpreter)"라 불릴 수 있고, 또는 운영 체제에서는 "운영 체제 셸"이라 불릴 수 있다). 셸은 애플리케이션 프로그램 및/또는 최종 사용자들에 의해 직접 액세스가능한 하드웨어/소프트웨어 인터페이스 시스템의 외부 레이어(outer layer)이다. 셸과 반대로, 커널은 하드웨어 컴포넌트들과 직접 상호작용하는, 하드웨어/소프트웨어 인터페이스 시스템의 가장 내부의 레이어(innermost layer)이다.

<53> 도 9에 도시된 바와 같이, 예시적인 범용 컴퓨팅 시스템은 처리 장치(621), 시스템 메모리(662), 및 시스템 메모리를 포함하는 다양한 시스템 컴포넌트들을 처리 장치(621)와 연결하는 시스템 버스(623)를 포함하는 통상적인 컴퓨팅 장치(660) 또는 이와 유사한 것 등을 포함한다. 시스템 버스(623)는 메모리 버스 또는 메모리 컨트롤러, 주변 장치 버스 및 각종 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스를 비롯한 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 시스템 메모리는 판독 전용 메모리(ROM)(664) 및 랜덤 액세스 메모리(RAM)(625)를 포함한다. 시동 중과 같은 때에, 컴퓨팅 장치(660) 내의 구성요소들 사이의 정보 전송을 돕는 기본 루틴을 포함하는 기본 입/출력 시스템(BIOS)(666)은 ROM(664)에 저장되어 있다. 컴퓨팅 장치(660)는 또한 하드 디스크 드라이브(하드 디스크는 도시되지 않음)에 기록을 하거나 그로부터 판독을 하는 하드 디스크 드라이브(627), 이동식 자기 디스크(629)(예를 들어, 플로피 디스크, 이동식 저장소)에 기록을 하거나 그로부터 판독을 하는 자기 디스크 드라이브(628)(예를 들어, 플로피 드라이브), 및 CD-ROM 또는 기타 광학 매체와 같은 이동식 광학 디스크(631)에 기록을 하거나 그로부터 판독을 하는 광학 디스크 드라이브(630)를 포함할 수 있다. 하드 디스크 드라이브(627), 자기 디스크 드라이브(628), 및 광학 디스크 드라이브(630)는 각각 하드 디스크 드라이브 인터페이스(632), 자기 디스크 드라이브 인터페이스(633), 및 광학 드라이브 인터페이스(634)에 의해 시스템 버스(623)에 연결된다. 드라이브들 및 그들과 관련된 컴퓨터 판독가능 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 및 컴퓨팅 장치(660)를 위한 다른 데이터에 대한 비휘발성 저장소를 제공한다. 여기에 설명된 예시적인 환경이 하드 디스크, 이동식 자기 디스크(629), 및 이동식 광학 디스크(631)를 사용하지만, 당업자들은, 컴퓨터에 의해 액세스가능한 데이터를 저장할 수 있는 다른 유형의 컴퓨터 판독가능 매체들, 예를 들어, 자기 카세트, 플래시 메모리 카드, DVD, 베르누이 카트리지, 랜덤 액세스 메모리(RAM), 판독 전용 메모리(ROM), 및 이와 유사한 것 등이 또한 예시적인 운영 환경에서 이용될 수 있다는 것을 이해해야 한다. 마찬가지로, 예시적인 환경은 또한 열 센서, 및 보안 또는 화재 경보 시스템과 같은 여러 유형의 모니터링 장치들, 및 다른 정보의 소스들을 포함할 수 있다.

<54> 운영 체제(635), 하나 이상의 애플리케이션 프로그램(636), 기타 프로그램 모듈(637), 및 프로그램 데이터(638)를 포함하는 다수의 프로그램 모듈이 하드 디스크, 자기 디스크(629), 광학 디스크(631), ROM(664), 또는 RAM(625)에 저장될 수 있다. 사용자는 키보드(640) 및 포인팅 장치(642)(예를 들어, 마우스)와 같은 입력 장치들을 통해 컴퓨팅 장치(660)에 명령어 및 정보를 입력할 수 있다. 다른 입력 장치(도시되지 않음)로는 마이크, 조이스틱, 게임 패드, 위성 안테나, 스캐너, 또는 이와 유사한 것 등이 포함될 수 있다. 이들 및 다른 입력 장치들은 종종 시스템 버스에 연결된 직렬 포트 인터페이스(646)를 통해 처리 장치(621)에 연결되지만, 병렬 포트, 게임 포트, 또는 USB(universal serial bus)와 같은 다른 인터페이스에 의해 연결될 수 있다. 모니터(647) 또는 다른 유형의 디스플레이 장치는 또한 비디오 어댑터(648)와 같은 인터페이스를 통해 시스템 버스(623)에 연결된다. 모니터(647)에 더해, 컴퓨팅 장치들은 일반적으로 스피커 및 프린터와 같은 다른 주변 출력 장치들(도시되지 않음)을 포함한다. 도 9의 예시적인 환경은 또한 호스트 어댑터(655), SCSI(Small Computer System Interface) 버스(656), 및 SCSI 버스(656)에 연결된 외부 저장 장치(662)를 포함한다.

<55> 컴퓨팅 장치(660)는 원격 컴퓨터(649)와 같은 하나 이상의 원격 컴퓨터들로의 논리적 접속을 이용하는 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(649)는 다른 컴퓨팅 장치(예를 들어, 개인용 컴퓨터), 서버, 라우터, 네트워크 PC, 피어 장치, 또는 다른 일반 네트워크 노드일 수 있고, 도 9에는 메모리 저장 장치(650)(플로피 드라이브)만이 도시되었지만, 통상적으로 컴퓨팅 장치(660)와 관련하여 위에서 설명된 다수 또는 모든 구성 요소들을 포함한다. 도 9에 도시된 논리적 접속은 근거리 통신망(LAN)(651) 및 광역 통신망(WAN)(652)을 포함한다. 이러한 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크(enterprise wide computer network), 인트라넷, 및 인터넷에서 일반적인 것이다.

<56> LAN 네트워킹 환경에서 사용되는 경우, 컴퓨팅 장치(660)는 네트워크 인터페이스 또는 어댑터(653)를 통해 LAN(651)에 연결된다. WAN 네트워킹 환경에서 사용되는 경우, 컴퓨팅 장치(660)는 인터넷과 같은 WAN(652)을 통해 통신을 설정하기 위한 모뎀(654) 또는 기타 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(654)은 직렬 포트 인터페이스(646)를 통해 시스템 버스(623)에 접속될 수 있다. 네트워크화된 환경에서, 컴퓨팅 장치(660) 또는 그의 일부와 관련하여 기술된 프로그램 모듈은 원격 메모리 저장 장치에 저장될 수 있다. 도시된 네트워크 접속은 예시적인 것이며 이 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 이해할 것이다.

<57> 투명한 제2 요소를 통한 플랫폼 인증에 대한 수많은 실시예들이 특히 컴퓨터화된 시스템들에 대해 적절한 것으로 구상되지만, 이 명세서의 어떠한 것도 본 발명을 이러한 실시예들에 제한하기 위한 것이 아니다. 반대로, 여기에 사용된 "컴퓨터 시스템"이라는 용어는, 장치들이 본질적으로 전자적, 기계적, 논리적 또는 가상인지에 상관없이, 정보를 저장하고 프로세싱할 수 있고/거나 장치 자체의 비헤비어 또는 실행을 제어하도록 저장된 정보를 이용할 수 있는 임의의 장치 및 모든 장치를 포함하기 위한 것이다.

<58> 여기에 설명된 다양한 기술들은 하드웨어 또는 소프트웨어, 또 필요시에 이들의 조합과 관련하여 구현될 수 있다. 따라서, 투명한 제2 요소를 통한 플랫폼의 인증을 위한 방법 및 장치들, 또는 이들의 특정 양상 또는 부분들은 플로피 디스크, CD-ROM, 하드 드라이브, 또는 임의의 다른 기계-판독가능 저장 매체와 같은 실체적 매체(tangible media)에 구현된 프로그램 코드(즉, 명령어들)의 형태를 가질 수 있고, 여기서, 프로그램 코드가 컴퓨터와 같은 기계에 로드되고 실행되면 기계는 투명한 제2 요소를 통한 플랫폼의 인증을 위한 장치가 된다.

<59> 프로그램(들)은 원한다면 어셈블리 또는 기계 언어에 구현될 수 있다. 어떤 경우든, 언어는 컴파일된 또는 해석된(interpreted) 언어일 수 있고, 하드웨어 구현들과 결합될 수 있다. 투명한 제2 요소를 통한 플랫폼의 인증을 위한 방법 및 장치들은 또한 어떤 전송 매체를 통해 전송되는 프로그램 코드의 형태로 구현된 통신들을 통해 실행될 수 있으며, 예를 들어, 전기적 배선(wiring) 또는 케이블링(cabling)을 통해, 광섬유를 통해, 또는 임의의 다른 형태의 전송을 통해서 실행될 수 있고, 여기서, 프로그램 코드가 EPROM, 게이트 어레이, 프로그램가능 논리 소자(programmable logic device, PLD), 클라이언트 컴퓨터 또는 이와 유사한 것 등과 같은 기계에 의해 수신되고 로드되고 실행되면, 기계는 속성에 의해 가상 기계들을 관리하기 위한 장치가 된다. 범용 프로세서에 구현된 경우, 투명한 제2 요소를 통한 플랫폼의 기능 인증을 호출하도록 동작하는 고유한 장치를 제공하기 위해, 프로그램 코드는 프로세서와 결합한다. 또한, 투명한 제2 요소를 통한 플랫폼의 인증과 관련하여 사용된 임의의 저장 기술들은 항상 하드웨어 및 소프트웨어의 조합일 수 있다.

<60> 투명한 제2 요소를 통한 플랫폼의 인증이 다양한 도면의 예시적 실시예들과 관련하여 설명되었지만, 다른 유사한 실시예들이 사용될 수 있고, 또는 투명한 제2 요소를 통한 플랫폼의 인증을 벗어나지 않고 이에 대해 변경 및 추가가 이루어질 수 있음을 이해해야 한다. 따라서, 여기에 설명된 바와 같은 투명한 제2 요소를 통한 플랫

폼의 인증은 임의의 단일 실시예로 제한되어서는 안되며, 오히려 첨부된 청구항들에 따른 폭 및 범위로 이해되어야 한다.

도면의 간단한 설명

<7> 앞선 발명의 요약뿐 아니라 아래의 상세한 설명은 첨부된 도면과 함께 읽었을 때 더 잘 이해된다. 투명한 제2 요소(transparent second factor)를 통한 플랫폼 인증을 설명하기 위해 도면에는 예시적인 구성들이 도시되지만, 투명한 제2 요소를 통한 플랫폼 인증은 개시된 특정 방법 및 수단들에 제한되지 않는다.

<8> 도 1은 시스템의 BIOS 부분을 통해 블록 장치(block device)를 액세스하기 위한 예시적인 시스템에 대한 기능 블록도이다.

<9> 도 2는 시스템의 BIOS 부분을 통해 생체인식 장치를 액세스하기 위한 예시적인 시스템의 기능 블록도이다.

<10> 도 3은 시스템의 EFI를 통해 블록 장치를 액세스하기 위한 예시적인 시스템의 기능 블록도이다.

<11> 도 4는 시스템의 EFI를 통해 생체인식 장치를 액세스하기 위한 예시적인 시스템의 기능 블록도이다.

<12> 도 5는 ISO 7816 명세에 따르는 블록 장치의 예시적인 파일 시스템에 대한 도시이다.

<13> 도 6은 플랫폼 메모리에 가상 블록 장치를 생성하기 위한 예시적인 프로세스의 흐름도이다.

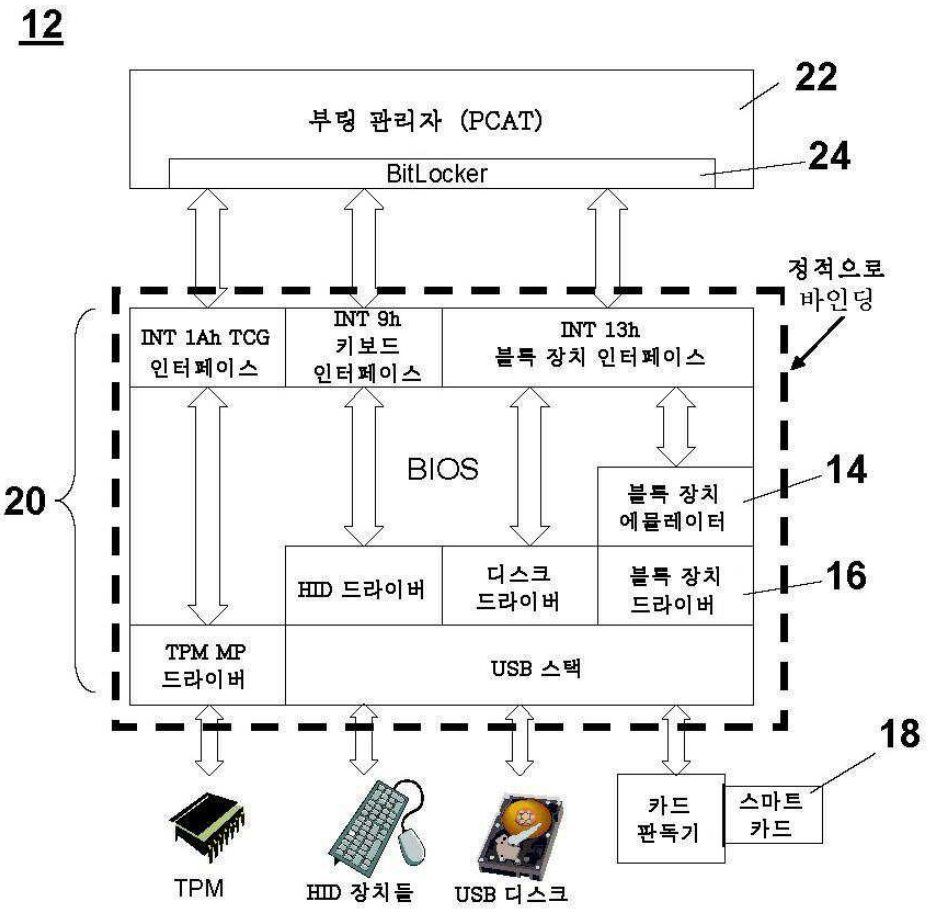
<14> 도 7은 가상 블록 장치를 위한 플랫폼 메모리의 예시적인 할당에 대한 도시이다.

<15> 도 8은 CCID 스마트 카드를 액세스하기 위한 예시적인 프로세스의 흐름도이다.

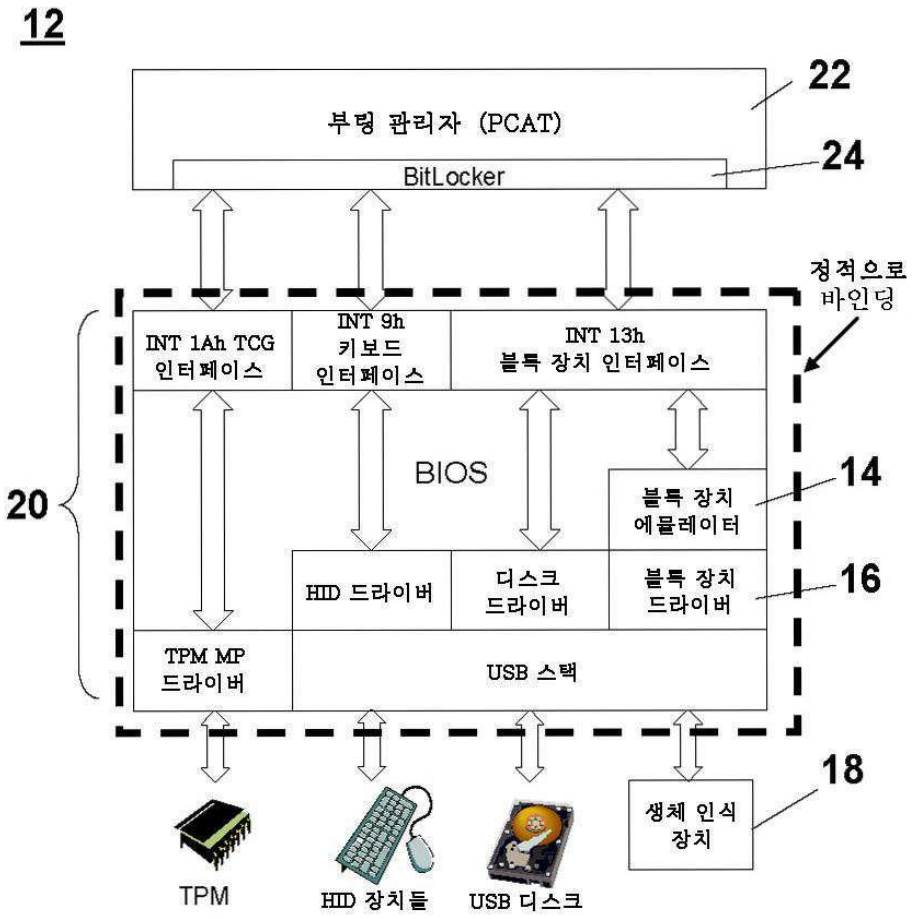
<16> 도 9는 투명한 제2 요소를 통한 플랫폼 인증이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 도시이다.

도면

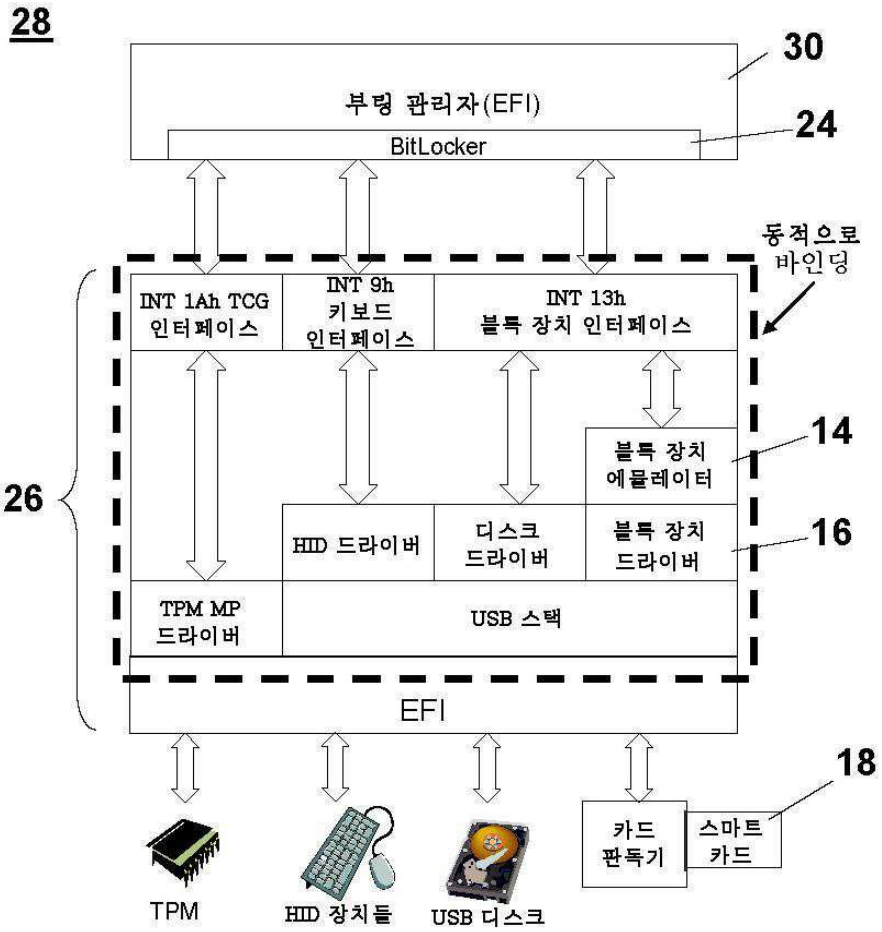
도면1



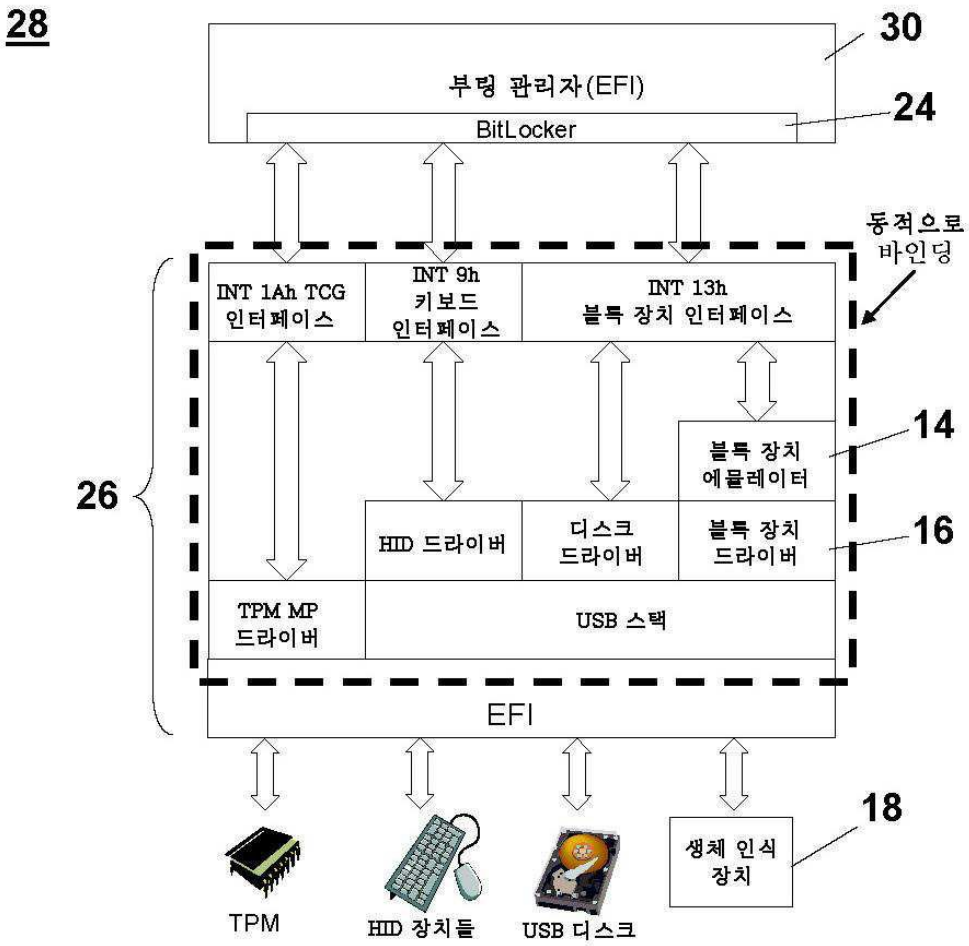
도면2


12
부팅 관리자 (PCAT)
22
BitLocker
24
정적으로
바인딩
INT 1Ah TCG
인터페이스
INT 9h
키보드
인터페이스
INT 13h
블록 장치 인터페이스
BIOS
20
블록 장치
에뮬레이터
14
HID 드라이버
디스크
드라이버
블록 장치
드라이버
16
TPM MP
드라이버
USB 스택
18
생체 인식
장치
TPM
HID 장치들
USB 디스크

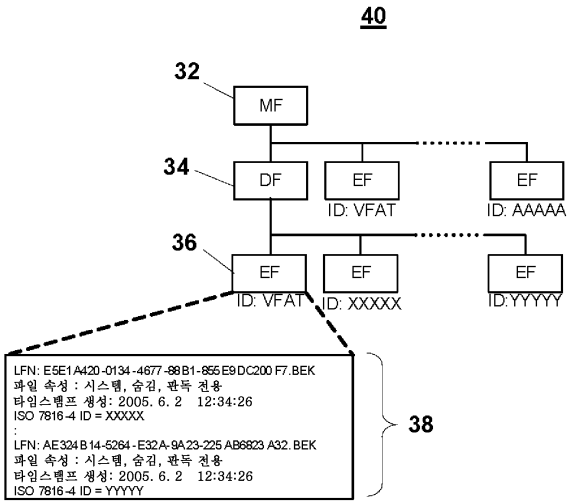
도면3


28
30
부팅 관리자(EFI)
24
BitLocker
동적으로
바인딩
INT 1Ah TCG
인터페이스
INT 9h
키보드
인터페이스
INT 13h
블록 장치 인터페이스
14
블록 장치
에뮬레이터
26
HID 드라이버
디스크
드라이버
16
블록 장치
드라이버
TPM MP
드라이버
USB 스택
EFI
18
카드
판독기
스마트
카드
TPM
HID 장치들
USB 디스크

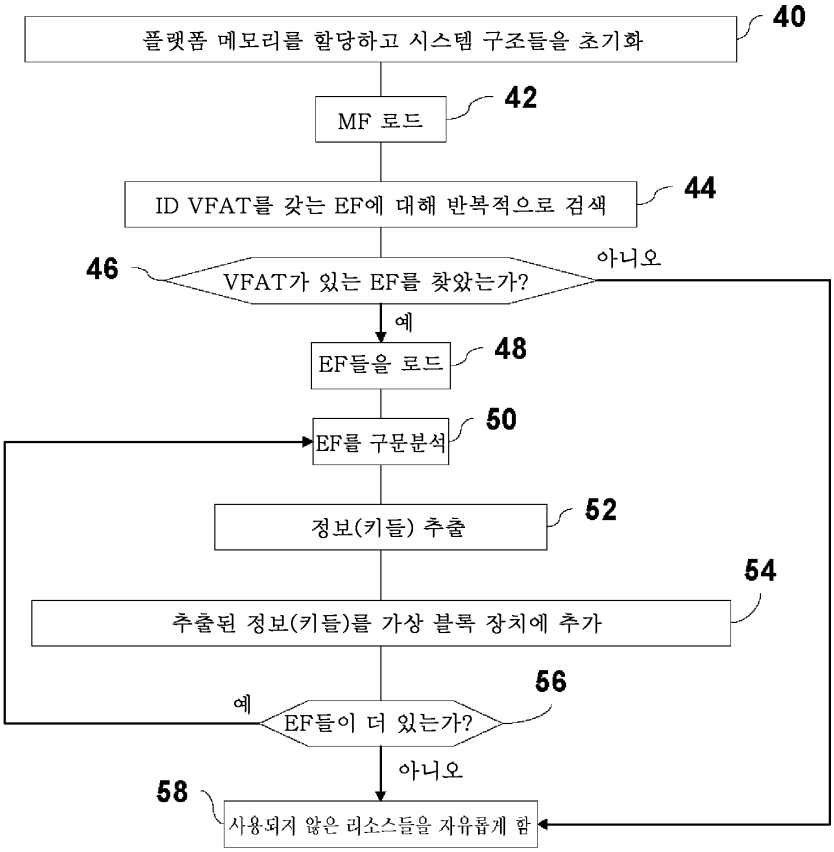
도면4


28
30
부팅 관리자(EFI)
24
BitLocker
동적으로
바인딩
INT 1Ah TCG
인터페이스
INT 9h
키보드
인터페이스
INT 13h
블록 장치 인터페이스
14
블록 장치
에뮬레이터
26
HID 드라이버
디스크
드라이버
블록 장치
드라이버
16
TPM MP
드라이버
USB 스택
EFI
18
생체 인식
장치
TPM
HID 장치들
USB 디스크

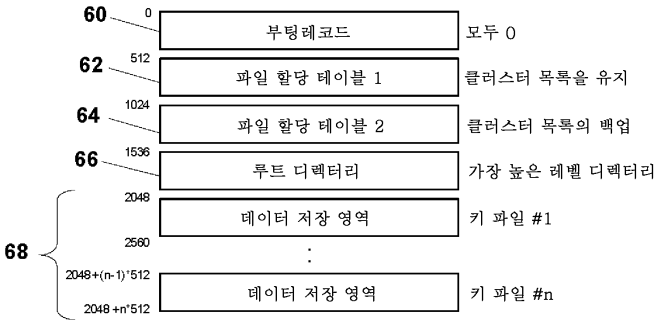
도면5


40
32
MF
34
DF
EF
ID: VFAT
EF
ID: AAAAA
36
EF
ID: VFAT
EF
ID: XXXXX
EF
ID:YYYYY
LFN: E5E1A420-0134-4677-88B1-855E9DC200F7.BEK
파일 속성 : 시스템, 숨김, 판독 전용
타임스탬프 생성: 2005. 6. 2 12:34:26
ISO 7816-4 ID = XXXXX
:
LFN: AE324B14-5264-E32A-9A23-225AB6823A32.BEK
파일 속성 : 시스템, 숨김, 판독 전용
타임스탬프 생성: 2005. 6. 2 12:34:26
ISO 7816-4 ID = YYYYY
38

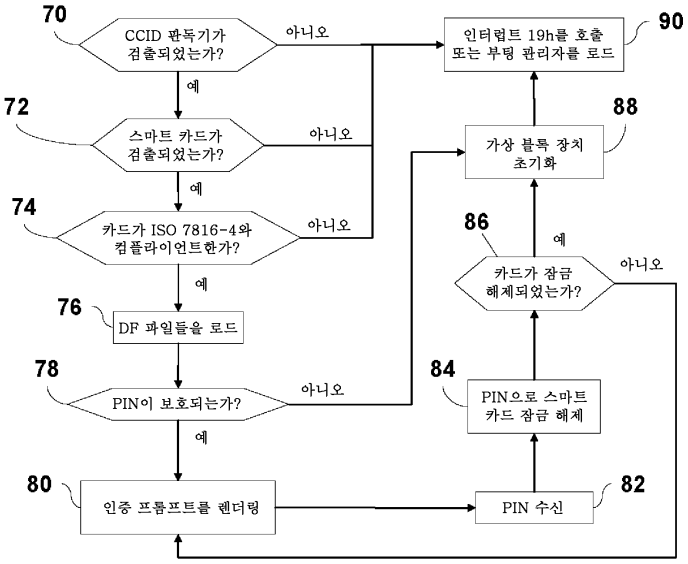
도면6


플랫폼 메모리를 할당하고 시스템 구조들을 초기화
40
MF 로드
42
ID VFAT를 갖는 EF에 대해 반복적으로 검색
44
VFAT가 있는 EF를 찾았는가?
46
아니오
예
EF들을 로드
48
EF를 구문분석
50
정보(키들) 추출
52
추출된 정보(키들)를 가상 블록 장치에 추가
54
예
EF들이 더 있는가?
56
아니오
사용되지 않은 리소스들을 자유롭게 함
58

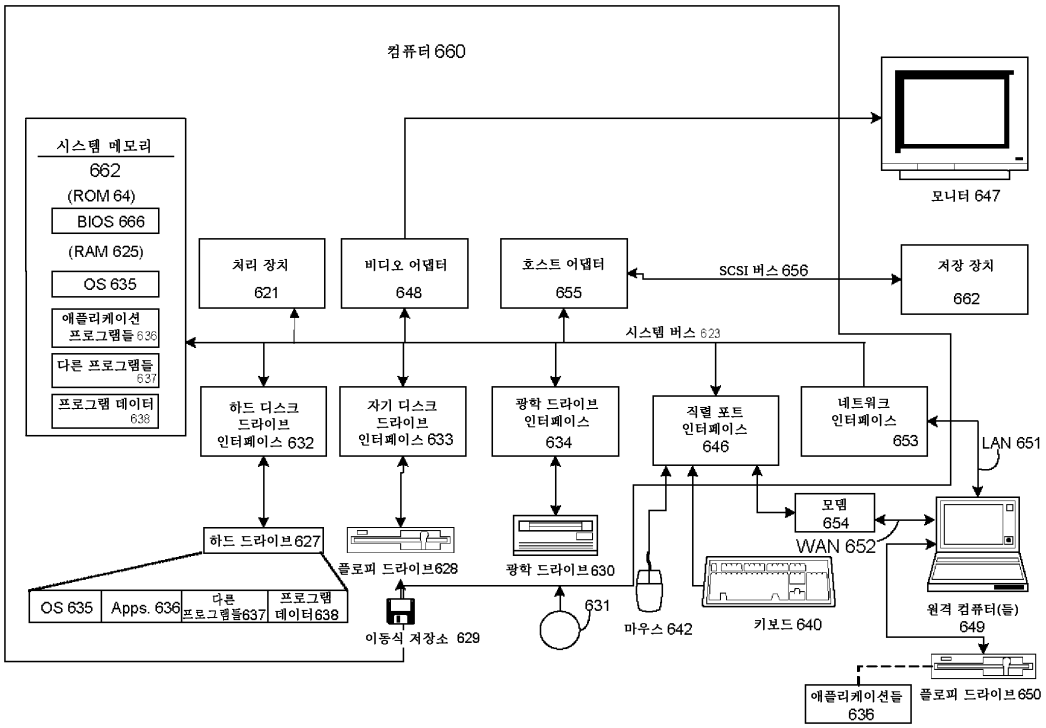
도면7


60
0
부팅레코드
모두 0
62
512
파일 할당 테이블 1
클러스터 목록을 유지
64
1024
파일 할당 테이블 2
클러스터 목록의 백업
66
1536
루트 디렉터리
가장 높은 레벨 디렉터리
68
2048
데이터 저장 영역
키 파일 #1
2560
2048+(n-1)*512
데이터 저장 영역
키 파일 #n
2048+n*512

도면8


70
CCID 판독기가 검출되었는가?
아니오
예
72
스마트 카드가 검출되었는가?
아니오
예
74
카드가 ISO 7816-4와 컴플라이언트한가?
아니오
예
76
DF 파일들을 로드
78
PIN이 보호되는가?
아니오
예
80
인증 프롬프트를 렌더링
82
PIN 수신
84
PIN으로 스마트 카드 잠금 해제
86
카드가 잠금 해제되었는가?
예
아니오
88
가상 블록 장치 초기화
90
인터럽트 19h를 호출 또는 부팅 관리자를 로드

도면9


컴퓨터 660
시스템 메모리
662
(ROM 64)
BIOS 666
(RAM 625)
OS 635
애플리케이션 프로그램들 636
다른 프로그램들 637
프로그램 데이터 638
처리 장치 621
비디오 어댑터 648
호스트 어댑터 655
SCSI 버스 656
저장 장치 662
모니터 647
시스템 버스 623
하드 디스크 드라이브 인터페이스 632
자기 디스크 드라이브 인터페이스 633
광학 드라이브 인터페이스 634
직렬 포트 인터페이스 646
네트워크 인터페이스 653
LAN 651
모뎀 654
WAN 652
하드 드라이브 627
플로피 드라이브628
광학 드라이브630
OS 635
Apps. 636
다른 프로그램들637
프로그램 데이터638
이동식 저장소 629
631
마우스 642
키보드 640
원격 컴퓨터(들) 649
애플리케이션들 636
플로피 드라이브650