(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0042889
(43) 공개일자 2008년05월15일

(51) Int. Cl.
G06F 21/00 (2006.01) *G06F 9/44* (2006.01)
H04L 9/00 (2006.01)
(21) 출원번호 10-2008-7006042
(22) 출원일자 2008년03월12일
심사청구일자 없음
번역문제출일자 2008년03월12일
(86) 국제출원번호 PCT/US2006/034632
국제출원일자 2006년09월02일
(87) 국제공개번호 WO 2007/032975
국제공개일자 2007년03월22일
(30) 우선권주장
11/224,418 2005년09월12일 미국(US)

(71) 출원인
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
쑤, 장웨이
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
필립스, 토마스
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
(뒷면에 계속)
(74) 대리인
양영준, 백만기

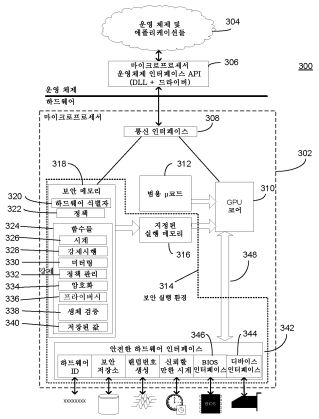
전체 청구항 수 : 총 20 항

(54) 전자 장치에서 사용하기 위한 처리 유닛, 컴퓨터 및 이를동작시키는 방법

(57) 요 약

전자 장치 내에서 사용하기 위한 처리 유닛이 표준 명령어 처리 및 통신 인터페이스를 포함하고 또한 운영 체제에서 발견되는 것에 추가하여 또는 그 위치에서 함수적 기능을 포함한다. 처리 유닛 내의 보안 메모리는 하드웨어 식별자, 정책 데이터, 및 보안 시계, 정책 관리, 및 정책 강제시행과 같은 서브시스템 함수를 포함할 수 있다. 보안 메모리 내부의 함수 내의 데이터는 외부의 처리 유닛으로부터 액세스불가능하다.

대표도 - 도3



(72) 발명자

프랑크, 알렉산더

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

스티브, 커트 에이.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

아도트, 이삭 피.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

홀, 마틴 에이치.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

두푸스, 제임스 에스.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

특허청구의 범위

청구항 1

전자 장치에서 사용하기 위한 처리 유닛으로서,

명령어 처리 유닛;

통신 인터페이스;

식별 표지(indicia);

정책 관리 서킷(circuit);

강제시행 서킷;

단조적으로(monotonically) 증가하는 시간 기반을 제공하는 시계 서킷; 및

상기 전자 장치의 동작을 사용 정책에 따라 규제하는, 상기 사용 정책에 대응하는 데이터를 저장하는 위조방지 메모리

를 포함하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 2

제1항에 있어서, 상기 사용 정책은 상기 전자 장치 내의 리소스 사용에 대응하는 시스템 설정을 지정하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 3

제1항에 있어서, 상기 사용 정책은 시간에 의한 미터링 및 사용에 의한 미터링 중 적어도 하나에 대응하는 동작 값을 포함하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 4

제1항에 있어서, 프라이버시 함수-상기 프라이버시 함수는 사용자 데이터에 대응하는 정보를 보호하기 위한 것임-를 수행하기 위한, 상기 위조방지 메모리 내에 저장된 소프트웨어 코드를 더 포함하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 5

제1항에 있어서, 상기 통신 인터페이스는 정책 업데이트를 전달하기 위한 데이터를 애플리케이션 프로그램 인터페이스에 제공하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 6

제1항에 있어서, 상기 정책 관리 서킷은 상기 전자 장치의 사용을 언제 미터링할 것인지를 결정하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 7

제1항에 있어서, 상기 강제시행 서킷은, 동작이 상기 정책을 따르지 않는다고 상기 정책 관리 서킷이 판정하는 경우에 상기 전자 장치의 상기 동작을 제한하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 8

제1항에 있어서, 생체 인증 함수를 수행하기 위한, 상기 위조방지 메모리에 저장된 소프트웨어 코드를 더 포함하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 9

제1항에 있어서, 암호화 함수를 수행하기 위한, 상기 위조방지 메모리에 저장된 소프트웨어 코드를 더 포함함으로써, 정책 업데이트가, 설치 전에 암호적으로 검증되는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 10

제9항에 있어서, 상기 암호화 함수는 상기 전자 장치의 다른 컴포넌트와 신뢰되는 관계를 구축하도록 동작가능한 전자 장치에서 사용하기 위한 처리 유닛.

청구항 11

제1항에 있어서, 상기 정책은 하드웨어 컨피규레이션을 정의하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 12

제1항에 있어서, 상기 정책은, 외부 시스템 메모리를 상기 위조방지 메모리에 배정함으로써 상기 외부 시스템 메모리를 일반적 사용으로부터 제외하는 메모리 컨피규레이션을 정의하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 13

제1항에 있어서, 저장된 값 함수를 수행하기 위한, 상기 위조방지 메모리에 저장된 소프트웨어 코드를 더 포함하는 전자 장치에서 사용하기 위한 처리 유닛.

청구항 14

메모리 컨피규레이션, 처리 용량, 미터링 요건, 및 주변장치에 대한 인증 중 적어도 하나에 대응하는 정책에 따른 사용에 적응된 컴퓨터로서,

휘발성 메모리;

비휘발성 메모리;

입력 인터페이스;

통신 인터페이스; 및

상기 휘발성 메모리, 상기 비휘발성 메모리, 상기 입력 인터페이스, 및 상기 통신 인터페이스에 결합된 처리 유닛

을 포함하고, 상기 처리 유닛은,

명령어 처리 유닛;

데이터 버스 인터페이스;

정책 관리 함수;

강제시행 함수;

위조방지 시계; 및

상기 정책을 저장하는 보안 메모리

를 포함하고, 상기 컴퓨터는 상기 보안 메모리에 저장된 상기 정책에 따라 동작하는 컴퓨터.

청구항 15

제14항에 있어서, 상기 정책에 대응하는 데이터는 상기 입력 인터페이스 및 상기 통신 인터페이스 중 하나를 통해 수신되는 컴퓨터.

청구항 16

제14항에 있어서, 상기 처리 유닛은 암호화 함수를 더 포함하는 컴퓨터.

청구항 17

위조방지 메모리를 구비한 처리 유닛을 갖는 컴퓨터를 동작시키는 방법으로서,

상기 컴퓨터를 부팅하기 위한 컴퓨터 명령어를 실행하는 단계;

상기 위조방지 메모리로부터 정책을 판독하기 위한 컴퓨터 명령어를 실행하는 단계-상기 정책은 메모리 컨피규레이션, 처리 용량, 미터링 요건, 및 주변장치에 대한 인증 중 적어도 하나에 대응함-; 및

상기 정책에 따라 상기 컴퓨터를 동작시키기 위한 컴퓨터 명령어를 실행하는 단계

를 포함하는 컴퓨터를 동작시키는 방법.

청구항 18

제17항에 있어서,

상기 컴퓨터를 제한된 사용 모드에 두는 단계;

시간 표시(indication)를 포함하는 복원 코드를 수신하는 단계; 및

내부 시계 함수와 상기 시간 표시를 비교하는 단계

를 더 포함하는 컴퓨터를 동작시키는 방법.

청구항 19

제17항에 있어서,

상기 컴퓨터의 미터링된 사용을 상기 정책이 언제 요구할지를 결정하는 단계; 및

상기 정책에 따라 상기 사용을 미터링하는 단계

를 더 포함하는 컴퓨터를 동작시키는 방법.

청구항 20

제17항에 있어서, 상기 정책에 따라 상기 컴퓨터를 동작시키기 위한 컴퓨터 명령어를 실행하는 단계는, 시스템 메모리를 상기 위조방지 메모리에 재배정하여 상기 컴퓨터에 의한 일반적 사용을 불가능하게 만들기 위한 컴퓨터 명령어를 실행하는 단계를 더 포함하는 컴퓨터를 동작시키는 방법.

명 세 서

배 경 기 술

<1> 소프트웨어 동작 플랫폼 또는 운영 체제를 호스팅하는 하드웨어 처리 플랫폼을 구비한 아키텍처를 사용하여 동작하는 컴퓨터가 사용되고 있다. 운영 체제는 처리 플랫폼에 (적어도 광범위한 파라미터의 범위 내에서) 독립적으로 설계되고, 거꾸로 말하면 처리 플랫폼은 (일반적으로 동일 광범위한 파라미터의 범위 내에서) 운영 체제와 독립적으로 설계된다. 예를 들면, 리눅스(Linux) 또는 마이크로소프트 윈도우즈(Microsoft Windows)는 인텔 x86 프로세서의 대부분의 버전에서 실행될 수 있다. 가상 머신 모니터(virtual machine monitor: VMM) 또는 하이퍼바이저(hypervisor)를 사용함으로써, 양쪽의 운영 체제를 동시에 실행시키는 것이 가능하다. 유사하게는, UNIX와 같은 일부 운영 체제는, 두 가지 이상의 프로세서(예를 들면, IBM PowerPC와 Sun Sparc 프로세서)상에서 실행될 수 있다.

<2> 처리 플랫폼과 운영 체제 간의 독립성은 소위 해커들에 의해 유발될 수 있는 보안 위험을 초래하는데, 이는 프로세서와 운영 체제 간, 즉, 컴퓨터의 하드웨어와 소프트웨어 간에 신뢰를 구축하기 어렵기 때문인 것이 일부 이유이다. 현재의 마이크로프로세서는, 주어진 명령어를 맹목적으로 실행하는 "페치 및 실행" 사이클에 진입하며, 실행되는 명령어의 내용이나 분기(ramification)를 고려하지도 않고 전자 장치의 사용에 관련된 정책 결정에 참여하지도 않는다.

발명의 상세한 설명

<3> 시스템 기능이 내장된 처리 유닛은, 예를 들면, 컴퓨터, 셀룰러폰, PDA, 미디어 플레이어와 같은 전자 장치의 사용 당 지불(pay-per-use), 현금 지불(pay-as-you-go), 또는 기타 미터링식 동작을 시행하는데 사용하기 위한 보안 및/또는 운영 정책을 시행하기 위한 보안 기반을 제공한다. 처리 유닛은, 대부분의 또는 모든 현대식 마

이크로프로세서에서 발견되는 특징 및 함수적 지원을 포함하며 또한 하드웨어 식별자, 위조방지 시계, 및 보안 저장소를 제공하는 추가적인 함수들을 지원할 수 있다. 암호화 유닛과 같은 기타 함수적 기능들도 존재할 수 있다. 결과적으로 처리 유닛은, 사용 정책에 부합하여 동작될 수 있는 컴퓨터에 대한 기반을 구축하기 위한 임의의 외부 컴포넌트, 특히 운영 체제 소프트웨어, 신뢰되는 컴퓨팅 모듈(TCM), 또는 보안 부팅 BIOS에 의존하지 않는다.

<4> 부팅시에, 처리 유닛은 어떤 정책이 유효한지를 결정하고 예를 들면, 이용가능한 메모리, 주변기기의 개수나 타입, 또는 네트워크 통신에 대한 제한을 설정하는 등의, 정책에 따른 시스템 컨피규레이션을 설정한다. 시계는 소정의 기간 동안의 사용과 같은, 사용을 미터링하는 데 사용하고 시스템 시계를 이용하여 위조하는 것을 감지하기 위한 참고로서 신뢰할 만한 시간을 제공한다.

실 시 예

<9> 이하에서는 다수의 상이한 실시예에 대한 상세한 설명을 개시하지만, 이러한 설명의 법률적 범위는 본 명세서의 끝 부분에 개시되는 특허청구범위의 용어들에 의해 제한된다는 점이 이해되어야 한다. 상세한 설명은 단지 예시적인 것으로 고려되어야 하고, 모든 가능한 실시예들을 설명하는 것은 아니며, 이는 비록 불가능하지 않더라도 모든 가능한 실시예들을 설명한다는 것은 실용적이지 않기 때문이다. 현재의 기술 또는 본 특허의 출원일 이후에 개발된 기술들을 사용하여 다수의 대안적인 실시예들이 구현될 수 있지만, 이러한 것들도 역시 본 발명의 특허청구범위의 범위에 포함되는 것이다.

<10> "본 명세서에서 사용될 때, 용어 '____'는 ...을 의미하도록 여기에서 정의된다"라는 문장 또는 이와 유사한 문장을 사용하여 본 명세서에서 명시적으로 정의되지 않는 한, 이러한 용어는 명시적으로든 또는 암시적으로든, 그것의 일반적인 의미 또는 보통의 의미 이상으로, 그 용어의 의미를 제한하려는 의도는 아니며, 이러한 용어가 본 명세서의 임의의 섹션에 있는 임의의 문장(청구항의 언어가 아니라)에 기초하여 범위가 제한되도록 해석되어서는 안 된다는 것 또한 이해할 것이다. 본 명세서의 끝 부분에 있는 청구범위에 언급된 임의의 용어가 단일 의미로 일관된 방식으로 본 명세서에서 지칭되는 점에서, 이는 단지 독자들을 혼동시키지 않도록 단순 명료하게 기술한 것으로, 이러한 청구범위의 용어가 암시적으로든 또는 다르게든, 그 단일 의미로 제한되는 것은 아니다. 마지막으로, 임의의 구조의 언급 없이 "수단"이라는 단어와 기능을 언급함으로써 청구항의 구성요소가 정의되지 않는다면, 임의의 청구항의 구성요소의 범위가 35 U.S.C. § 112, 6번째 항의 적용에 기초하여 해석되어야 하는 것은 아니다.

<11> 대부분의 진보성이 있는 기능성 및 다수의 진보성이 있는 원리들은 소프트웨어 프로그램들 또는 명령어들 및 주문형 집적 회로와 같은 집적 회로(ICs)와 함께, 또는 이들에 의해 가장 잘 구현된다. 당업자라면, 예를 들어, 사용가능한 시간, 현재 기술 및 경제적인 고려들에 의해 동기부여가 되는 상당한 노력 및 많은 설계 선택 사항들에도 불구하고, 본 명세서에 개시되는 개념들 및 원리들에 도움을 받을 때, 최소한의 실험으로 이러한 소프트웨어 명령어들과 프로그램들 및 IC들을 용이하게 생성할 수 있을 것이다. 따라서, 간결화 및 본 발명에 따른 원리들 및 개념들을 모호하게 하는 임의의 위험성을 최소화하는 관점에서, 이러한 소프트웨어 및 IC들에 대한 더 이상의 논의는 혹시 존재하더라도 바람직한 실시예의 원리들 및 개념들에 대하여 본질적인 것들로 제한될 것이다.

<12> 도 1은 사용 당 지불 컴퓨터 시스템을 구현하는데 사용될 수 있는 네트워크(10)를 도시한다. 네트워크(10)는 인터넷, 가상 사설망(VPN), 또는 하나 이상의 컴퓨터, 통신 장치, 데이터베이스 등이 서로 통신가능하게 접속되도록 허용하는 임의의 기타 네트워크일 수 있다. 네트워크(10)는 이더넷(16) 및 라우터(18), 및 지상통신선(20)을 통해 퍼스널 컴퓨터(12) 및 컴퓨터 단말기(14)에 접속될 수 있다. 한편, 네트워크(10)는 무선 통신국(26) 및 무선 링크(28)를 통해 랩탑 컴퓨터(22) 및 PDA(24)에 무선으로 접속될 수 있다. 유사하게는, 서버(30)가 통신 링크(32)를 사용하여 네트워크(10)에 접속될 수 있고 메인프레임(34)은 다른 통신 링크(36)를 사용하여 네트워크(10)에 접속될 수 있다.

<13> 도 2는 네트워크(10)에 접속될 수 있고 동적 소프트웨어 공급 시스템과의 하나 이상의 컴포넌트를 구현하는 데 사용될 수 있는 컴퓨터(110)의 형태인 컴퓨팅 장치를 도시한다. 컴퓨터(110)의 컴포넌트들은 처리 장치(120), 시스템 메모리(130), 및 시스템 메모리를 비롯한 각종 시스템 컴포넌트들을 처리 장치(120)에 연결시키는 시스템 버스(121)를 포함하지만 이에 제한되는 것은 아니다. 시스템 버스(121)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. 예로서, 이러한 아키텍처는 ISA(Industry Standard Architecture) 버스, MCA(Micro Channel Architecture) 버스, EISA(Enhanced ISA) 버스, VESA(Video Electronics Standard

Association) 로컬 버스, 그리고 메자닌 버스(Mezzanine bus)로도 알려진 PCI(Peripheral Component Interconnect) 버스 등을 포함하지만 이에 제한되는 것은 아니다.

<14> 처리 장치(120)는 인텔, 또는 당 기술분야에 공지된 기타의 것들에서 사용가능한 마이크로프로세서와 같은 마이크로프로세서일 수 있다. 처리 장치는 하나의 클립일 수 있거나 다중 프로세서 유닛일 수 있어서 연관된 주변 장치 클립들(도시 생략) 또는 기능 블록들(도시 생략)을 포함할 수 있다. 이러한 연관된 클립들은 전처리 프로세서, 파이프라인 클립, 단순 버퍼 및 드라이버들을 포함할 수 있거나, 또는 몇몇 현재 기술 컴퓨터 아키텍처에 공지된 "노스브리지(Northbridge)"와 "사우스브리지(Southbridge)" 클립과 같은 보다 복잡한 클립들/클립 세트를 포함할 수 있다. 처리 유닛(120)은 또한 전체적 처리 유닛의 부분으로서 마이크로프로세서와 또는 연관될 클립과 동일한 실리콘 상에, 보안 실행 환경(125)을 포함한다. 보안 실행 환경(125) 및 처리 장치(120), 또는 동등한 장치들과 그것의 상호작용은, 도 3 및 도 4에 관련하여 아래에서 더욱 자세히 논의된다.

<15> 컴퓨터(110)는 통상적으로 각종 컴퓨터 판독가능 매체를 포함한다. 컴퓨터(110)에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 이동식 및 비이동식 매체를 둘 다 포함한다. 예로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함하지만 이에 제한되는 것은 아니다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터(110)에 의해 액세스되고 원하는 정보를 저장할 수 있는 임의의 기타 매체를 포함하지만 이에 제한되는 것은 아니다. 통신 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터 등을 구현하고 모든 정보 전달 매체를 포함한다. "피변조 데이터 신호"라는 용어는, 신호 내에 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들의 모든 조합이 또한 컴퓨터 판독가능 매체의 영역 안에 포함되는 것으로 한다.

<16> 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. 시동 중과 같은 때에, 컴퓨터(110) 내의 구성요소들 사이의 정보 전송을 돕는 기본 루틴을 포함하는 기본 입/출력 시스템(BIOS)(133)은 통상적으로 ROM(131)에 저장되어 있다. RAM(132)은 통상적으로 처리 유닛(120)이 즉시 액세스 할 수 있고 및/또는 현재 동작시키고 있는 데이터 및/또는 프로그램 모듈을 포함한다. 예로서, 도 1은 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136) 및 프로그램 데이터(137)를 도시하고 있지만 이에 제한되는 것은 아니다.

<17> 컴퓨터(110)는 또한 기타 이동식/비이동식, 휘발성/비휘발성 컴퓨터 저장매체를 포함한다. 단지 예로서, 도 1은 비이동식·비휘발성 자기 매체에 기록을 하거나 그로부터 판독을 하는 하드 디스크 드라이브(141), 이동식·비휘발성 자기 디스크(152)에 기록을 하거나 그로부터 판독을 하는 자기 디스크 드라이브(151), CD-ROM 또는 기타 광 매체 등의 이동식·비휘발성 광 디스크(156)에 기록을 하거나 그로부터 판독을 하는 광 디스크 드라이브(155)를 포함한다. 예시적인 운영 환경에서 사용될 수 있는 기타 이동식/비이동식, 휘발성/비휘발성 컴퓨터 기억 매체로는 자기 테이프 카세트, 플래시 메모리 카드, DVD, 디지털 비디오 테이프, 고상(solid state) RAM, 고상 ROM 등이 있지만 이에 제한되는 것은 아니다. 하드 디스크 드라이브(141)는 통상적으로 인터페이스(140)와 같은 비이동식 메모리 인터페이스를 통해 시스템 버스(121)에 접속되고, 자기 디스크 드라이브(151) 및 광 디스크 드라이브(155)는 통상적으로 인터페이스(150)와 같은 이동식 메모리 인터페이스에 의해 시스템 버스(121)에 접속된다.

<18> 위에서 설명되고 도 2에 도시된 드라이브들 및 이들과 관련된 컴퓨터 저장 매체는, 컴퓨터(110)를 위해, 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 및 기타 데이터를 저장한다. 도 2에서, 예를 들어, 하드 디스크 드라이브(141)는 운영 체제(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146), 및 프로그램 데이터(147)를 저장하는 것으로 도시되어 있다. 여기서 주의할 점은 이들 컴포넌트가 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136), 및 프로그램 데이터(137)와 동일하거나 그와 다를 수 있다는 것이다. 이에 관해, 운영 체제(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146) 및 프로그램 데이터(147)에 다른 번호가 부여되어 있다는 것은 적어도 이들이 다른 사본(copy)이라는 것을 나타내기 위한 것이다. 사용자는 키보드(162), 마이크(163) 및 마우스, 트랙볼(trackball) 또는 터치 패드와 같은 포인팅 장치(161) 등의 입

력 장치를 통해 명령 및 정보를 컴퓨터(110)에 입력할 수 있다. 다른 입력 장치(도시 생략)로는 마이크, 조이스틱, 게임 패드, 위성 안테나, 스캐너 등을 포함할 수 있다. 이들 및 기타 입력 장치는 종종 시스템 버스에 결합된 사용자 입력 인터페이스(160)를 통해 처리 유닛(120)에 접속되지만, 병렬 포트, 게임 포트 또는 USB(universal serial bus) 등의 다른 인터페이스 및 버스 구조에 의해 접속될 수도 있다. 모니터(191) 또는 다른 유형의 디스플레이 장치도 비디오 인터페이스(190) 등의 인터페이스를 통해 시스템 버스(121)에 접속될 수 있다. 모니터 외에, 컴퓨터는 스피커(197) 및 프린터(196) 등의 기타 주변 출력 장치를 포함할 수 있고, 이들은 출력 주변장치 인터페이스(195)를 통해 접속될 수 있다.

<19> 컴퓨터(110)는 원격 컴퓨터(180)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(180)는 또 하나의 퍼스널 컴퓨터, 핸드-헬드 장치, 서버, 라우터, 네트워크 PC, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있고, 통상적으로 컴퓨터(110)와 관련하여 상술된 구성 요소들의 대부분 또는 그 전부를 포함한다. 도 1에 도시된 논리적 접속으로는 LAN(171) 및 WAN(173)이 있지만, 기타 네트워크를 포함할 수도 있다. 이러한 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷, 및 인터넷에서 일반적인 것이다.

<20> LAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 네트워크 인터페이스 또는 어댑터(170)를 통해 LAN(171)에 접속된다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 통상적으로 인터넷과 같은 WAN(173)을 통해 통신을 구축하기 위한 모뎀(172) 또는 기타 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(172)은 사용자 입력 인터페이스(160) 또는 기타 적절한 메커니즘을 통해 시스템 버스(121)에 접속된다. 네트워크화된 환경에서, 컴퓨터(110) 또는 그의 일부와 관련하여 기술된 프로그램 모듈은 원격 메모리 저장 장치에 저장될 수 있다. 예로서, 도 2는 원격 애플리케이션 프로그램(185)이 원격 컴퓨터(180)에 있는 것으로 도시하고 있지만 이에 제한되는 것은 아니다. 도시된 네트워크 접속은 예시적인 것이며 이 컴퓨터들 사이에 통신 링크를 구축하는 기타 수단이 사용될 수 있다는 것을 이해할 것이다.

<21> 도 3은 컴퓨터(300)의 단순화된 블록도를 도시한다. 컴퓨터는 처리 유닛(302)을 포함하는데, 이는 처리 유닛(120)과 유사하거나 동일할 수 있다. 본 블록도는 또한 인터페이스 애플리케이션 프로그램 인터페이스(API)(306)에 의해 처리 유닛(302)에 결합된 운영 체제 및 애플리케이션(304)을 갖는 컴퓨터(300)를 도시한다. API(306)는 처리 유닛(302) 내의 통신 인터페이스(308)와 통신할 수 있다. 통신 인터페이스(308)는 인터럽트 핸들러, 또는 메시지 처리 핸들러, 파싱 유닛, 등의 형태를 취한다. 종래의 마이크로프로세서에서처럼, 처리 유닛(302)은 범용 마이크로코드(312) 세트를 사용하여 통신 인터페이스(308)를 통해 수신되는 범용 명령어를 처리하는 일반적 처리 유닛(GPU) 코어(310)를 포함할 수 있다. GPU 코어(310)의 동작 및 범용 마이크로코드(312)와의 관계는 당해 산업에서 문서화되어 이해되고 있으며, Intel Pentium™ 시리즈, Advanced Risc Machines Limited로부터의 ARM™ 프로세서, 및 IBM의 PowerPC™ 프로세서와 같은 프로세서로 예시된다.

<22> 보안 실행 환경(314)은 GPU 코어 및 마이크로코드(310, 312)에 의해 제공되는 일반적 처리 용량을 보완할 수 있다. 보안 실행 환경(314)은 지정된 실행 메모리(316)를 포함할 수 있다. 지정된 실행 메모리(316)는 처리 유닛(302) 내에서 높아진 권한 레벨을 갖는 명령어의 실행을 위한 매우 안전한 위치를 제공할 수 있다. 이 높아진 권한 레벨의 동작은 처리 유닛(302)으로 하여금 처리 유닛(302)의 외부로부터는 직접 액세스할 수 없는 코드를 실행하도록 허용한다. 예를 들면, 특정 인터럽트 벡터가 처리 유닛(302)을 보안 동작으로 설정할 수 있거나, 또는 보안 리소스를 요구하는 컨텐트를 위한 명령어가 구해질(evaluate) 수 있다. 이 높아진 권한 모드에서 동작할 때에, 처리 유닛(302)은 완전한 서브시스템으로서 동작하여, 보안 처리 환경을 구축하기 위해, 임의의 외부 자산, 예를 들면 BIOS 리소스, 프로그램 메모리, 또는 TCM을 요구하지 않는다.

<23> 보안 메모리(318)는 위조방지 방식으로, 컴퓨터(302)의 보안 동작에 연관된 코드 및 데이터를 저장할 수 있다. 통신 인터페이스(308)는 프로세서(302)에 삽입되는 어떤 명령어가 보안 메모리(318)로 향하게 될 것인지를, 후속하여 지정된 실행 메모리(316) 내에서의 실행을 위해, 결정할 수 있다. 보안 메모리(318) 내의 데이터는 미터링, 리포팅, 업데이트 요건 등과 같은 정책 관련 동작 지시(directives)를 지정할 수 있는 식별 표지(indicia) 즉, 하드웨어 식별자(320) 및 정책 데이터(322)를 포함할 수 있다. 보안 메모리(318)는 또한 각종 함수들(324)을 수행하는 데에 요구되는 코드 또는 데이터를 포함할 수 있다. 함수들(324)은, 몇 가지만 열거해 보면, 시계(326) 즉, 시계 함수를 수행하는 타이머, 강제시행 함수(328), 미터링(330), 정책 관리(332), 암호화(334), 프라이버시(336), 생체 검증(338), 및 저장된 값(340)을 포함할 수 있다.

<24> 시계(326)는 시간 추정을 위한 믿을만한 기반을 제공할 수 있고 운영 체제(134)에 의해 유지되는 시스템 시계에 대한 확인으로서 사용될 수 있어서 시스템 시계를 변경하여 컴퓨터(300)를 부정하게 사용하려는 시도를 막도록

돕는다. 시계(326)는 또한 예를 들면, 업그레이드 가능성을 확인하기 위해 호스트 서버와의 통신을 요구하고자, 정책 관리(332)에 따라 사용될 수도 있다. 강제시행 함수(328)는 지정된 실행 메모리(316)에 로드되어 컴퓨터(300)가 정책(322)의 하나 이상의 구성요소에 따르지 않는다고 판정되는 경우에 수행될 수 있다. 이러한 액션은 처리 유닛(302)에게 보안 실행 환경(314)에 의해 사용되는 일반적으로 이용가능한 시스템 메모리를 할당하도록 명령함으로써 시스템 메모리(132)를 제한하는 것을 포함할 수 있다. 보안 실행 환경(314)에 시스템 메모리(134)를 재할당함으로써, 시스템 메모리(134)는 근본적으로 사용자 목적을 위하여 이용할 수는 없게 제작된다.

<25> 다른 함수(324)는 미터링(330)일 수 있다. 미터링(330)은 예를 들면, 계류중인 미국특허 출원 제11/006,837호에서 논의된 것과 같은 각종 기술 및 측정법을 포함할 수 있다. 미터링할지의 여부 및 측정을 위한 어떤 특정한 항목들이 정책(322)의 함수가 될 것인지는 정책 관리 함수(332)에 의해 수행된다. 암호화 함수(334)는 디지털 서명 검증, 디지털 서명, 랜덤 번호 생성, 및 복호/해독에 사용될 수 있다. 이러한 함수 중 임의의 것 또는 모두는 보안 메모리(318)로의 업데이트를 검증하거나 처리 유닛(302) 외부 엔티티에 대한 신뢰를 컴퓨터(300)의 내부이건 외부이건 간에 구축하는데 사용될 수 있다.

<26> 보안 실행 환경(314)은 몇 개의 특수 목적 함수로 하여금 개발되고 사용되도록 허용할 수 있다. 프라이버시 관리자(336)는 사용자 또는 관심이 있는 사람들의 개인적인 정보를 관리하는 데 사용될 수 있다. 예를 들면, 프라이버시 관리자(336)는 온라인 구매를 이용하기 위한 주소 및 신용카드 데이터를 보유하기 위한 "지갑(wallet)" 함수를 수행하는데 사용될 수 있다. 생체 검증 함수(338)는 개인의 신원을 검증하기 위한 외부의 생체 센서를 구비하여 사용될 수 있다. 이러한 신원 검증은 예를 들면, 개인정보 관리자(336) 내에 개인적인 정보를 업데이트하기 위해 또는 디지털 서명을 적용할 경우에 사용될 수 있다. 상술한 바와 같이, 암호화 함수(334)는 외부의 생체 센서(도시 생략)에 대한 신뢰 및 보안 채널을 구축하는 데 사용될 수 있다.

<27> 저장된 값 함수(340)는 또한 사용 당 지불 컴퓨터에서 시간에 대해 지불하는 식의 사용을 위해, 또는 외부의 구매, 예를 들면, 온라인 주식 상거래를 하는 동안에 수행될 수 있다.

<28> 지정된 실행 메모리(316) 내에서의 실행을 위해 보안 메모리(318)로부터의 데이터 및 함수의 사용은 안전한 하드웨어 인터페이스(342)의 제시를 허용한다. 안전한 하드웨어 인터페이스(342)는 주변의 장치(344) 또는 BIOS(346)로의 제한된 및/또는 모니터링된 액세스를 허용한다. 또한 함수(324)는 운영 체제(134)를 비롯한, 외부의 프로그램들로 하여금 안전한 하드웨어 인터페이스(342) 내의 GPU(310)와의 논리적 접속(348)을 통해 하드웨어 ID 및 랜덤 번호 생성과 같은 보안 설비를 액세스하도록 허용하는 데 사용될 수 있다. 또한, 코드로 구현되고 보안 메모리(318)에 저장된 상술한 각각의 함수는 로직으로 구현되고 물리적 회로로서 인스턴스화될 수 있다. 함수적 작용을 하드웨어와 소프트웨어 사이에 맵핑시키는 동작은 당 업계에 공지되어 있으므로 본원에서 더 자세히 논의되지는 않는다.

<29> 동작에서는, 데이터 또는 하나 이상의 함수가 보안 메모리(318)로부터 지정된 실행 메모리(316)로 로딩되게 하는 지정된 인터럽트가 통신 인터페이스(308)에 의해 처리될 수 있다. GPU(310)는 함수를 수행하기 위해 지정된 실행 메모리(316)로부터 실행할 수 있다. 일 실시예에서는, 이용가능한 함수들(324)은 운영 체제(134)에서 이용가능한 표준 함수를 보완하거나 이를 대신할 수 있다. 이러한 방식으로 컨피규어링될 때에, 대응하는 운영 체제(134)는 처리 유닛(302)과 짝을 이룰 때에만 동작할 것이다. 이 컨셉을 다른 레벨로 옮겨보면, 처리 유닛(302)의 다른 실시예가 프로그램되어 지정된 실행 메모리(316)로부터 실행되지 않은 채로 외부의 운영 체제 함수를 끄집어낼 수 있다. 예를 들면, 외부의 운영 체제(134)에 의해 메모리를 배정하려는 시도가 거부되거나 내부에 저장된 함수에게 재지시될 수 있다. 이러한 방식으로 컨피규어링되는 경우에는, 처리 유닛(302)을 위해 특별히 컨피규어링된 운영 체제만이 올바르게 동작할 것이다. 또 다른 실시예에서는, 인증된 소프트웨어 및 하드웨어가 존재함을 확인하기 위해 정책 데이터(322) 및 정책 관리 함수(332)가 운영 체제(134), 애플리케이션 프로그램(135), 및 하드웨어 파라미터를 테스트할 수 있다.

<30> 다른 실시예에서는, 컴퓨터(300)는 보통의 BIOS 시작 절차를 사용하여 부팅한다. 운영 체제(134)가 활성화되고 있는 시점에서는, 처리 유닛(302)이 정책 데이터(322)에 따라 컴퓨터(300)를 컨피규어링하도록 실행하기 위해 정책 관리 함수(332)를 지정된 실행 메모리(316)에 로드할 수 있다. 컨피규레이션 프로세스는 메모리, 처리 용량, 주변장치 이용가능성 및 사용의 배정뿐만 아니라 미터링 요건도 포함할 수 있다. 미터링이 시행되어야하는 경우에는, 어떤 측정이 취해져야 할 지와 같은, 미터링에 관련하는 정책들이 예를 들면, CPU 사용에 의해 또는 소정의 시간동안 활성화될 수 있다. 또한, 기간별로 또는 액티비티(activity) 당 사용 요금이 부과되는 경우에는, 저장된 값 밸런스가 저장된 값 함수(340)를 사용하여 유지될 수 있다. 컴퓨터(300)가 정책(322)에 따라 구

컨피규어링된 경우에는, 통상의 부팅 프로세스는 운영 체제(134) 및 기타 애플리케이션 프로그램(135)을 활성화 및 인스턴스화함으로써 계속될 수 있다. 다른 실시예에서 이 정책은 부팅 프로세스 또는 통상의 동작 사이클 내의 상이한 시점에 적용될 수 있다.

<31> 정책에 부합하지 않음이 발견되는 경우, 강제시행 함수(328)가 활성화될 수 있다. 강제시행 정책 및 동작의 논의는 계류중인 미국특허 출원 제11/152,214호에서 찾을 수 있다. 강제시행 함수(328)는 컴퓨터를 정책(322)에 따라 복원하려는 모든 시도가 실패한 경우에 컴퓨터(300)를 대안적인 동작 모드에 둘 수 있다. 예를 들면, 일 실시예에서는, 시스템 메모리(130)로서의 사용으로부터 메모리를 재배정하고 그것을 보안 메모리(318)로서 지정함으로써 제재가 가해질 수 있다. 보안 메모리(318)가 운영 체제(134)를 포함하는 외부의 프로그램에 의해 어드레싱이 가능하지 않기 때문에, 컴퓨터의 동작은 그러한 메모리 배정에 의해 제한될 (심지어는 엄격하게) 수 있다.

<32> 정책 및 강제시행 함수가 처리 유닛(302) 내에 유지되기 때문에, 시스템상의 일부 통상적인 어택이 어렵거나 불가능하다. 예를 들면, 정책은 외부 메모리의 정책 메모리 섹션을 대신함으로써 "스푸핑(spoofed)"되지 않을 수 있다. 유사하게는, 정책 및 강제시행 함수는 실행 사이클 및 그것들의 개개의 어드레스 범위를 차단함으로써 "스타빙(starved)"되지 않을 수 있다.

<33> 컴퓨터(300)를 통상의 동작으로 복귀시키기 위해, 라이센싱 권한 또는 서비스 제공자(도시 생략)로부터 복원 코드가 요구되어 컴퓨터(300)에 삽입될 필요가 있을 수 있다. 복원 코드는 하드웨어 ID(320), 저장된 값 보충, 및 시계(326)를 검증하는 데 사용되는 "no-earlier-than" 시간을 포함할 수 있다. 복원 코드는 통상적으로 처리 유닛(302)에 의한 확인을 위해 암호화되고 서명될 수 있다.

<34> 보안 메모리(318) 내의 데이터에 대한 추가적인 업데이트는, 예를 들면, 업데이트가 디지털 서명에 의해 검증되는 등의, 특정한 기준들이 부합될 경우에만 허용될 수 있다.

<35> 도 4는 도 3에 도시된 처리 유닛(302)의 대안적인 실시예를 도시하는 컴퓨터(400)의 블록도이다. 컴퓨터(400)는 처리 유닛(402), 운영 체제(404) 및 마이크로프로세서 운영 체제 인터페이스 애플리케이션 프로그램 인터페이스(API)(406)를 갖는다. 처리 유닛(402)은 인터럽트 특성 또는 어드레스 범위와 같은 기준에 기초하여 데이터 트래픽을 적합한 마이크로프로세서 함수로 지시함으로써 통신 인터페이스(308)와 유사한 방식으로 동작할 수 있는 통신 인터페이스(408)를 포함한다. 처리 유닛(402)은 통상의 일반적 처리 유닛(GPU)(410) 및 대응하는 범용 마이크로코드(412)를 가질 수 있다. 보안 실행 환경(414)은 별도의 보안 코어 프로세서(416)의 추가와 함께 보안 실행 환경(314) 내에서 발견되는, 동일하거나 유사한 함수를 포함할 수 있다. 보안 코어 프로세서(416)는 GPU 코어(410)로부터의 독립성의 추가적인 레벨 및 처리 유닛(402)의 보안상의 대응하는 증가를 허용할 수 있다.

<36> 보안 메모리(418)는 도 3에 관련하여 위에서 논의된 대로 동작하는 범용 함수(424)(예를 들면, 시계(426), 강제시행(428), 미터링(430), 정책 관리(432), 및 암호화(434))외에도 하드웨어 ID(420) 및 정책 데이터(422)를 포함할 수 있다. 또한, 개인정보 관리(436), 생체 검증(438), 및 저장된 값(440)과 같은 특수 목적 함수가 존재할 수 있다. 범용 및 특수 목적 함수들(424)은 당업자에 의해 쉽게 추측되는 기타 함수로서, 제한이 아닌 예시로서 주어진다.

<37> 신뢰할 만한 시계 및 랜덤 번호 생성자와 같은 함수의 제시뿐만 아니라, 장치 인터페이스(444) 및 BIOS 인터페이스(446)와 같은, 안전한 하드웨어 인터페이스(442)에 대한 장치의 제시가 가상 접속(448)을 통해 이루어질 수 있다. 안전한 코어 프로세서(416) 내에서의 GPU 코어(410) 간의 통신은 통신 버스(450)를 통해 이루어질 수 있다. 일 실시예에서, 통신 버스(450)는 보안 코어 프로세서(416)로부터 GPU(410)까지 신뢰되는 관계를 확장하도록 보안 채널을 통해 데이터를 전송할 수 있다.

<38> 상술된 것은 컴퓨터 사용의 섬세한 미터링을 위한 하드웨어 및 소프트웨어를 포함하는 몇 가지의 특정 실시예이다. 컴퓨터(110)의 하나 이상의 컴포넌트의 활성화 레벨을 모니터링 및 평가하고 적합한 비즈니스 규범을 적용함으로써 이익이 되는 사용을 결정하고 측정하는 보다 공정하고 정확한 방법이 개시된다. 이는 넓은 범위의 가정, 사무실 및 기업 사용 당 지불 또는 미터링식 사용 애플리케이션에 이익을 준다. 그러나, 당업자는 활성화 모니터링, 다중 레이트 스케줄뿐만 아니라, 적합한 사용 스케쥴을 결정하는 것에 관련된 보다 더욱 또는 덜 복잡한 규범을 위해 하드웨어 또는 소프트웨어의 서로 다른 조합의 사용을 포함하지만 이에 제한되지는 않는, 이러한 실시예들에 각종 변경 및 변화가 만들어질 수 있음을 이해할 것이다. 따라서, 본 명세서 및 도면은 제한적인 의의보다는 예시적으로 간주되며, 이러한 모든 변경들은 본 특허의 요지 내에 포함되는 것이다.

도면의 간단한 설명

<5> 도 1은 컴퓨터 네트워크의 간략하고 대표적인 블록도;

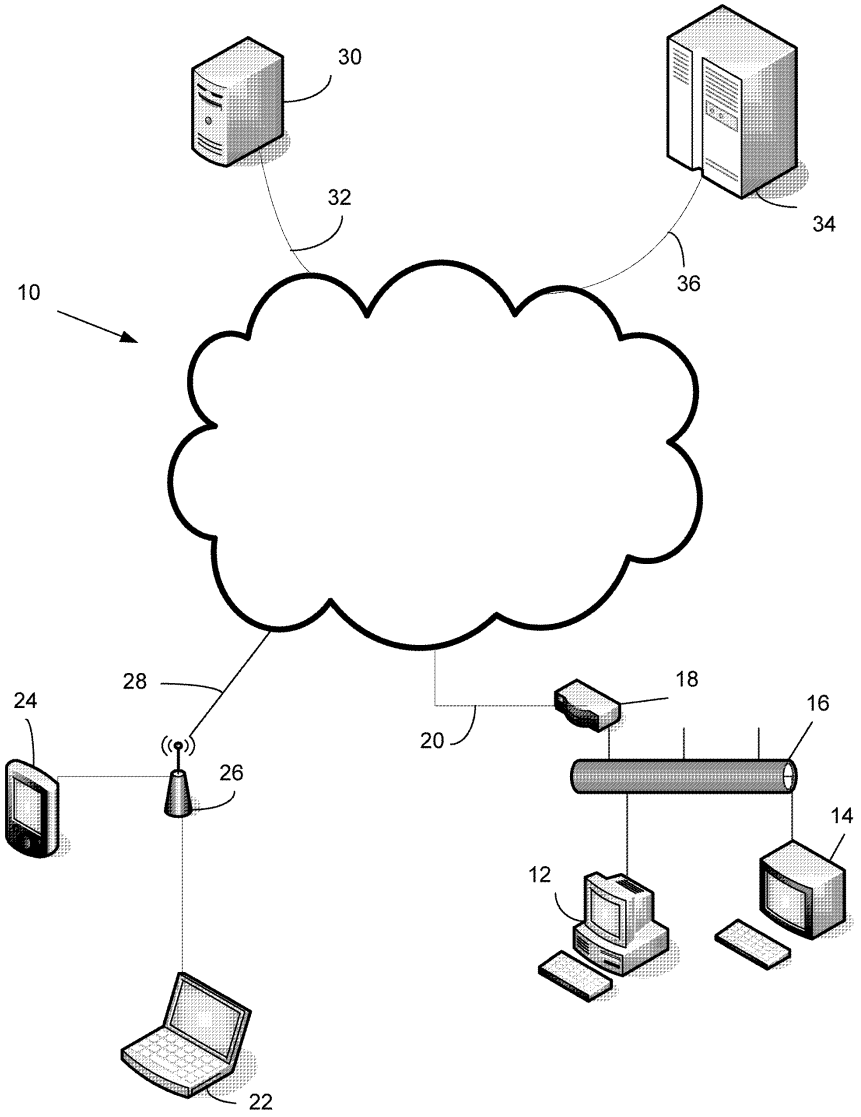
<6> 도 2는 도 1의 네트워트에 연결될 수 있는 컴퓨터의 블록도;

<7> 도 3은 처리 유닛의 세부를 도시하는 컴퓨터의 블록도;

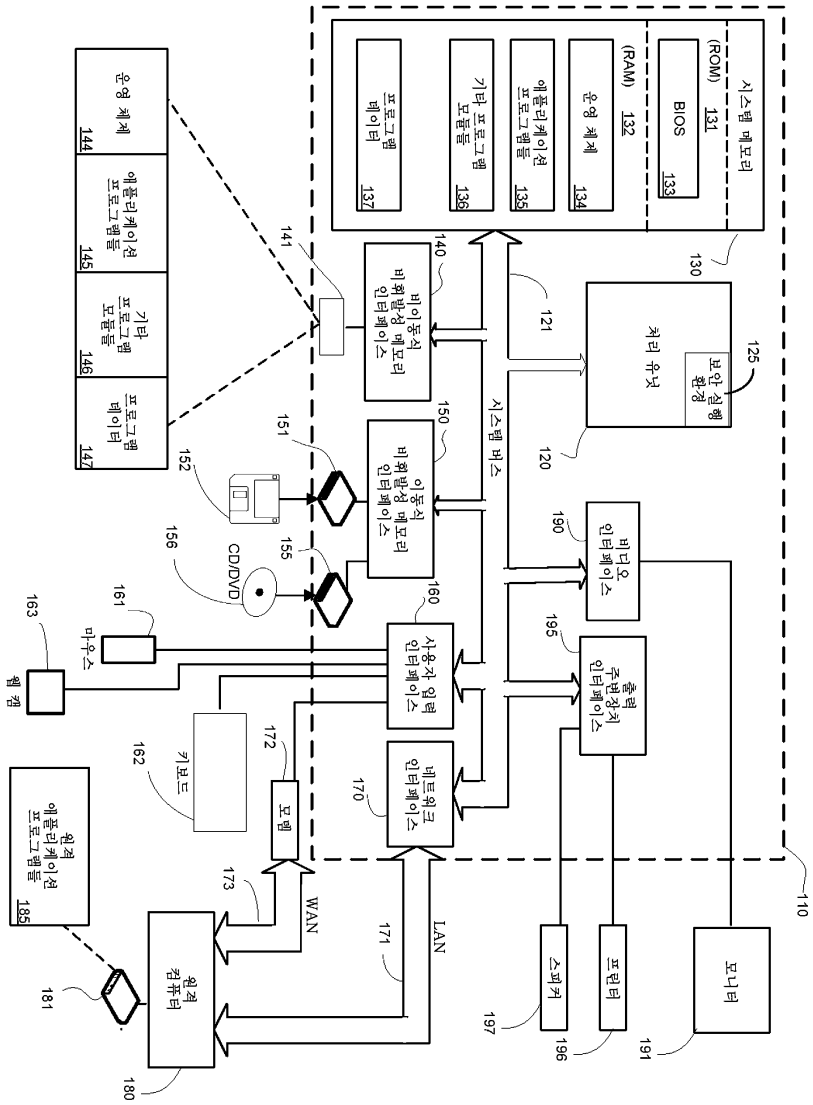
<8> 도 4는 도 3의 처리 유닛의 대안적인 실시예의 세부를 도시하는 컴퓨터의 블록도.

도면

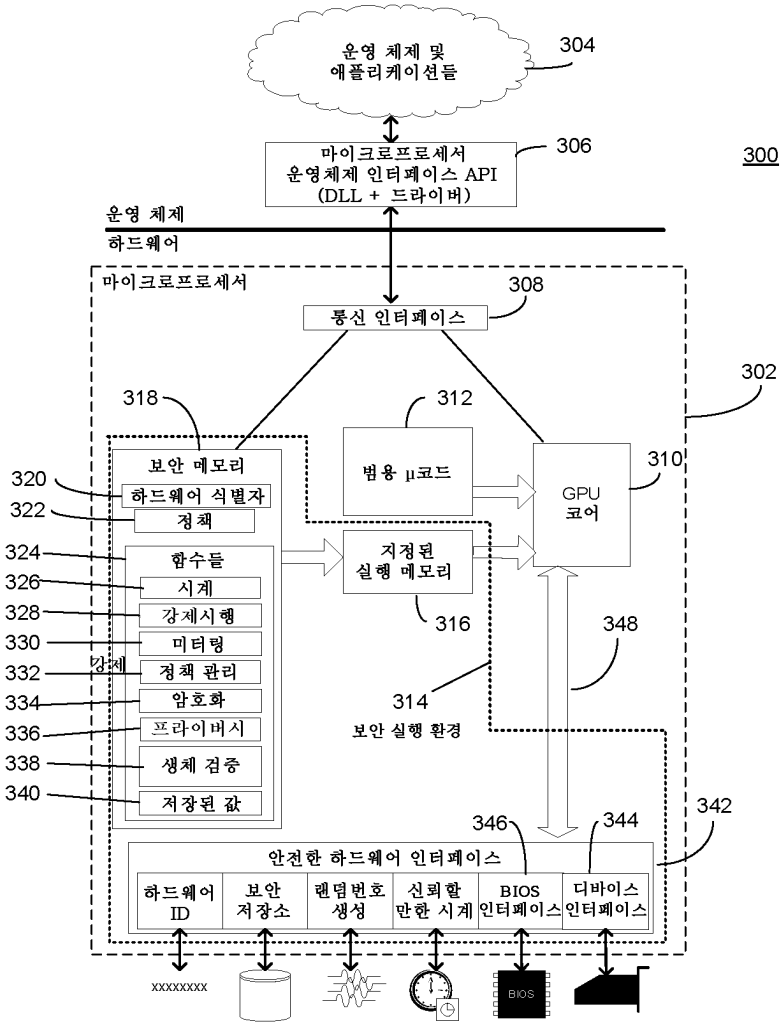
도면1



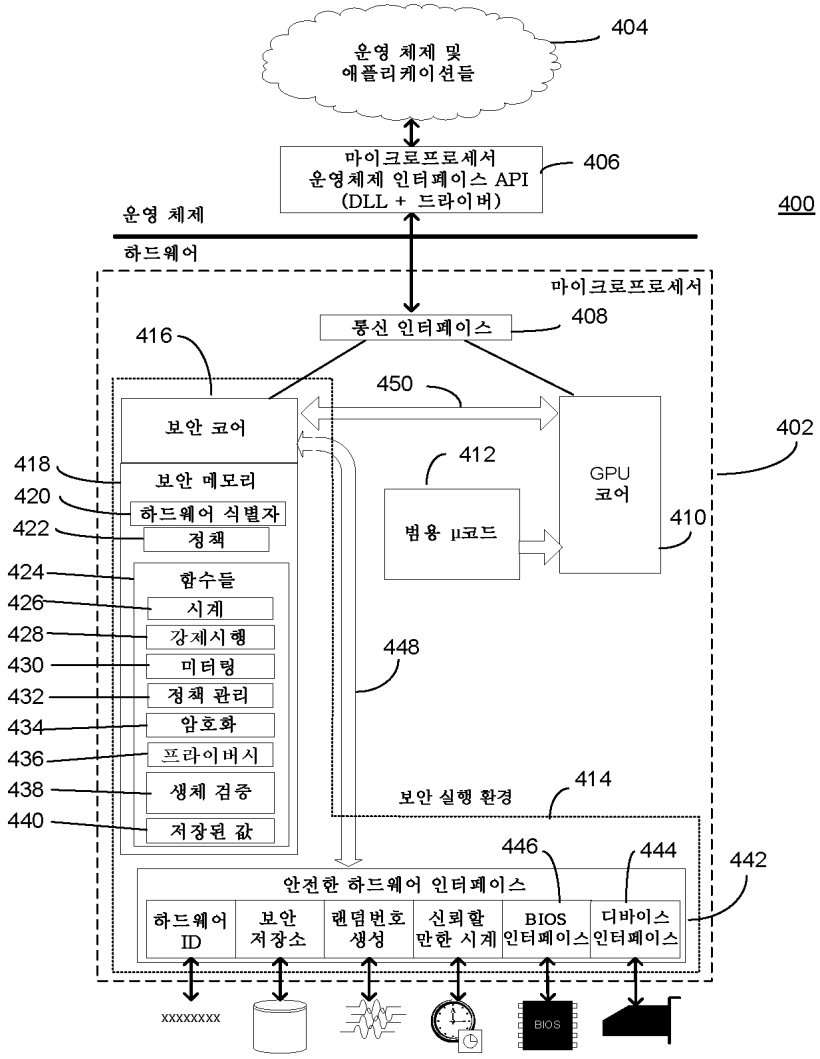
도면2


시스템 메모리
(ROM) 131
BIOS 133
(RAM) 132
운영 체제 134
애플리케이션 프로그램들 135
기타 프로그램 모듈들 136
프로그램 데이터 137
130
121
처리 유닛
보안 실행 환경
125
120
시스템 버스
비이동식 비휘발성 메모리 인터페이스
140
141
이동식 비휘발성 메모리 인터페이스
150
151
152
CD/DVD
155
156
160
사용자 입력 인터페이스
161
마우스
163
웹 캠
162
키보드
172
모뎀
170
네트워크 인터페이스
173
WAN
171
LAN
비디오 인터페이스
190
출력 주변장치 인터페이스
195
스피커
197
196
프린터
191
모니터
110
운영 체제 144
애플리케이션 프로그램들 145
기타 프로그램 모듈들 146
프로그램 데이터 147
원격 애플리케이션 프로그램들 185
181
원격 컴퓨터
180

도면3


운영 체제 및
애플리케이션들
304
마이크로프로세서
운영체제 인터페이스 API
(DLL + 드라이버)
306
300
운영 체제
하드웨어
마이크로프로세서
308
통신 인터페이스
302
318
312
보안 메모리
범용 μ코드
310
320
하드웨어 식별자
GPU
코어
322
정책
324
함수들
지정된
실행 메모리
326
시계
328
강제시행
316
348
330
미터링
강제
332
정책 관리
314
334
암호화
보안 실행 환경
336
프라이버시
338
생체 검증
340
저장된 값
342
346
344
안전한 하드웨어 인터페이스
하드웨어
ID
보안
저장소
랜덤번호
생성
신뢰할
만한 시계
BIOS
인터페이스
디바이스
인터페이스
xxxxxxxx
BIOS

도면4


운영 체제 및
애플리케이션들
404
마이크로프로세서
운영체제 인터페이스 API
(DLL + 드라이버)
406
운영 체제
400
하드웨어
마이크로프로세서
통신 인터페이스
408
416
450
보안 코어
402
412
418
보안 메모리
GPU
코어
420
하드웨어 식별자
범용 μ코드
410
422
정책
424
함수들
426
시계
428
강제시행
448
430
미터링
432
정책 관리
434
암호화
436
프라이버시
438
생체 검증
414
보안 실행 환경
440
저장된 값
446
444
442
안전한 하드웨어 인터페이스
하드웨어
ID
보안
저장소
랜덤번호
생성
신뢰할
만한 시계
BIOS
인터페이스
디바이스
인터페이스
xxxxxxxx
BIOS