(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.
H04L 9/32 (2006.01)

(11) 공개번호 10-2007-0084259
(43) 공개일자 2007년08월24일

(21) 출원번호 10-2007-7011075
(22) 출원일자 2007년05월15일
심사청구일자 없음
번역문 제출일자 2007년05월15일
(86) 국제출원번호 PCT/US2005/040949
국제출원일자 2005년11월12일
(87) 국제공개번호 WO 2006/055424
국제공개일자 2006년05월26일

(30) 우선권주장
10/989,122 2004년11월15일 미국(US)
11/006,837 2004년12월08일 미국(US)
11/022,493 2004년12월22일 미국(US)
11/109,438 2005년04월19일 미국(US)

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

(72) 발명자
프랭크, 알렉산더
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
필립스, 토마스, 지.
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

(74) 대리인
양영준
백만기

전체 청구항 수 : 총 20 항

(54) 고립형 컴퓨팅 환경을 프로그래밍하기 위한 시스템 및 방법

(57) 요약

고립형 컴퓨팅 환경을 갖는 컴퓨터가 제공된다. 고립형 컴퓨팅 환경은 제조, 배포 및 판매시 사용될 수 있는 초기 프로그래밍을 가능하게 하기 위해 채택된다. 고립형 컴퓨팅 환경은 또한 인증된 소스 또는 인증된 코드가 최종-사용자 환경에서 사용될 수 있는 코드 및 구성 데이터로 고립형 컴퓨팅 환경을 업데이트할 수 있게 해준다. 최종 업데이팅을 촉진하기 위해, 컴퓨터는 인증된 코드가 인스톨되어 동작될 수 있을 때까지 기능이 제한된 모드에 놓이게 될 것이다. 고립형 컴퓨팅 환경을 제재하고 안전하게 업데이트하기 위한 방법 및 장치가 개시된다.

대표도

도 2

특허청구의 범위

청구항 1.

컴퓨터에 사용하기 위한 것이며 컴퓨터-실행가능 명령어들을 실행하는 것인 고립형 컴퓨팅 환경으로서,

코어 서비스;

업데이트 코드 및 구성 정보 중 적어도 하나를 포함하는 메시지를 수신하기 위한 인터페이스;

제1 키를 액세스하고, 상기 제1 키를 사용하여 상기 메시지의 디지털 서명을 검증하기 위한 암호화 서비스; 및

업데이트 서비스를 포함하고,

상기 업데이트 서비스는 상기 제1 키를 사용한 상기 메시지의 검증 후 상기 메시지에 응답하여 상기 코어 서비스를 업데이트하는 고립형 컴퓨팅 환경.

청구항 2.

제1항에 있어서,

구성 테이블을 더 포함하고,

상기 업데이트 서비스는 상기 구성 정보를 포함하는 상기 메시지에 응답하여, 상기 구성 테이블을 업데이트하는 고립형 컴퓨팅 환경.

청구항 3.

제1항에 있어서,

상기 컴퓨터의 기능을 제한하는 제재(sanction) 서비스를 더 포함하는 고립형 컴퓨팅 환경.

청구항 4.

제3항에 있어서,

상기 제재 서비스는 상기 제1 키에 의해 검증된 상기 메시지에 응답하여 상기 코어 서비스를 업데이트 하기 이전에 상기 컴퓨터의 기능을 제한하는 고립형 컴퓨팅 환경.

청구항 5.

제1항에 있어서,

호스티드(hosted) 애플리케이션을 더 포함하고,

상기 메시지는 코어 서비스 업데이트 데이터 및 호스티드 애플리케이션 업데이트 데이터 중 적어도 어느 하나를 포함하는 고립형 컴퓨팅 환경.

청구항 6.

제1항에 있어서,

상기 코어 서비스의 특징이 디지털적으로 검증되는 고립형 컴퓨팅 환경.

청구항 7.

제1항에 있어서,

애플리케이션 프로그램 인터페이스를 더 포함하고,

상기 애플리케이션 프로그램 인터페이스는, 상기 고립형 컴퓨팅 환경 상에 호스트되는 애플리케이션을 인스턴스화하는 것(instantiating), 상기 고립형 컴퓨팅 환경 상에 호스트되는 애플리케이션을 구성하는 것, 및 상기 고립형 컴퓨팅 환경을 업데이트하는 것 중 어느 하나를 용이하게 하는 표준 인터페이스를 상기 컴퓨터와 상기 고립형 컴퓨팅 환경 사이에 제공하는 고립형 컴퓨팅 환경.

청구항 8.

고립형 컴퓨팅 환경을 사용하여 동작하도록 구성된 컴퓨터로서,

프로세서;

상기 프로세서에 연결되고 프로세서 실행가능 명령어들을 저장하기 위한 메모리; 및

애플리케이션 프로그램 인터페이스를 통해 송신된 신호에 응답하는 고립형 컴퓨팅 환경을 포함하고,

상기 고립형 컴퓨팅 환경은:

암호화 회로;

처리 회로;

애플리케이션 프로그램 인터페이스, 제1 실행가능 코드 및 제1 구성을 저장하기 위한 보안 메모리; 및

상기 컴퓨터의 기능을 지연시키기 위한 제재(sanction) 회로를 포함하고,

상기 제재 회로는, 적어도 상기 애플리케이션 프로그램 인터페이스를 사용하여 상기 제1 실행가능 코드 및 상기 제1 구성 중 어느 하나가 대체될 때까지 상기 컴퓨터의 기능을 지연시키는 컴퓨터.

청구항 9.

제8항에 있어서,

쌍방향 데이터 전송을 위한 포트를 더 포함하고,

상기 포트를 통해 수신되는 데이터는 제2 실행가능 코드 및 제2 구성 중 적어도 어느 하나를 포함하는 컴퓨터.

청구항 10.

제8항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 상기 제1 실행가능 코드 및 상기 제1 구성 중 어느 하나를 업데이트하기 위한 제1 루틴을 포함하는 컴퓨터.

청구항 11.

제8항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 상기 고립형 컴퓨팅 환경으로의 보안 데이터 전송 접속을 위한 지원을 포함하는 컴퓨터.

청구항 12.

제8항에 있어서,

상기 컴퓨터와 상기 제재 회로 사이에 연결되는 트리거 메커니즘을 더 포함하고, 이에 의해 상기 제재 회로가 트리거될 때 상기 컴퓨터의 기능이 지연되는 컴퓨터.

청구항 13.

제12항에 있어서,

상기 트리거 메커니즘은 탬퍼 내성이 있는(tamper-resistant) 컴퓨터.

청구항 14.

제8항에 있어서,

상기 컴퓨터의 기능을 지연하는 것은, 상기 프로세서를 영구적 리셋하는 것과 상기 프로세서의 기능성을 감소시키는 것 중 어느 하나를 포함하는 컴퓨터.

청구항 15.

운영 체제를 갖는 컴퓨터에 고립형 컴퓨팅 환경을 프로그래밍하는 방법에 있어서,

고립형 컴퓨팅 환경을 포함하는 컴퓨터를 준비하는 단계;

상기 고립형 컴퓨팅 환경에 임시(provisional) 기능성을 프로그램하는 단계;

운영 체제로부터 실행가능 명령어들을 수신하는 단계;

상기 실행가능 명령어들을 인증하는 단계; 및

상기 고립형 컴퓨팅 환경을 상기 임시 기능성에 대해 업데이트된 기능성으로 리프로그램하는 단계를 포함하는 고립형 컴퓨팅 환경 프로그래밍 방법.

청구항 16.

제15항에 있어서,

상기 실행가능 명령어들에 대한 인증 실패시 상기 컴퓨터의 기능을 제한하는 단계를 더 포함하는 방법.

청구항 17.

제15항에 있어서,

상기 고립형 컴퓨팅 환경에 임시 기능성을 프로그래밍하는 상기 단계는 잠정적(interim) 암호화 키 및 적어도 다운로드 기능을 프로그래밍하는 단계를 포함하는 방법.

청구항 18.

제17항에 있어서,

상기 고립형 컴퓨팅 환경을 업데이트된 기능성으로 리프로그램하는 상기 단계는 상기 잠정적 암호화 키를 사용하여 상기 업데이트된 기능성을 인증하는 단계를 포함하는 방법.

청구항 19.

제17항에 있어서,

상기 고립형 컴퓨팅 환경을 업데이트된 기능성으로 리프로그램하는 상기 단계는 상기 잠정적 암호화 키를 사용하여 상기 업데이트된 기능성의 소스를 인증하는 단계를 포함하는 방법.

청구항 20.

제15항에 있어서,

상기 컴퓨터의 동작을 제한하는 단계를 더 포함하며,

상기 컴퓨터의 동작을 제한하는 단계는 프로세서의 기능성을 감소시키는 것 및 상기 컴퓨터를 리셋하는 것 중 적어도 어느 하나를 포함하는 방법.

명세서

기술분야

본 출원은 2004년 11월 15일자 미국 특허출원 10/989,122호의 CIP 출원인 2004년 12월 8일자 미국 특허출원 11/006,837호의 CIP 출원인 2004년 12월 22일자 미국 특허출원 11/022,493호의 CIP 출원이다.

배경기술

위 출원들에서 논의된 바와 같이, 신뢰형 컴퓨팅 베이스라고도 불리우는 고립형 컴퓨팅 환경을 사용하는 것은, 컴퓨터 특히 사용당 요금제(pay-per-use) 또는 현금 지불형(pay-as-you-go) 비즈니스 모델에 사용되는 컴퓨터의 운용을 관리함

에 있어 상당한 성능을 발휘하게 된다. 이러한 컴퓨터가 최종 사용자의 손에 있는 경우, 고립형 컴퓨팅 환경은 부재중인 서비스 프로바이더 또는 기타 이해 당사자의 이해 관계를 나타낼 수 있다. 고립형 컴퓨팅 환경은 부재중인 당사자를 대신하여 동작하기 때문에, 제조 및 전달 프로세스를 통해 해당 부재중인 당사자의 이해 관계를 완전하게 나타내어야 한다. 제조 중 고립형 컴퓨팅 환경을 프로그래밍하는 것은 고립형 컴퓨팅 환경의 활용을 불필요하게 특정 비즈니스 로직/정책, 애플리케이션 프로세서 벤더 또는 운영 체제 버전 등의 특정 운영 환경으로 제한할 수 있다. 이와 반대로, 최종 사용자에게로 전달한 이후 고립형 컴퓨팅 환경을 프로그래밍하는 것은 최종 사용자로 하여금 고립형 컴퓨팅 환경의 프로그래밍에 간섭할 수 있게 하므로 서비스 프로바이더에게는 단점이 된다.

<발명의 개요>

본 발명의 일 양상에 따르면, 고립형 컴퓨팅 환경의 구성 및 프로그래밍을 위한 방법과 장치는 인증을 위해 암호화 방식들을 사용한다. 일 실시예에서는, 고립형 컴퓨팅 환경이 공개 및 배포 이전에 보안성 제조 환경에서 프로그램될 것이다. 다른 실시예에서는, 고립형 컴퓨팅 환경이 일반적 시스템 환경 및 표준 키들로 초기에 프로그램된다. 따라서, 고립형 컴퓨팅 환경은 이러한 일반적 시스템 및 표준 키들을 사용하여 장래의 중간 생성물 또는 최종 프로그래밍과 키들을 인증할 것이다. 이러한 레벨의 프로그래밍은 보안성 제조 환경에서 또는 그 외부에서 발생할 수 있다. 또 다른 실시예에서, 고립형 컴퓨팅 환경의 프로그래밍은 보안성 제조 환경 완비 이후로 보류될 것이다. 본 실시예에서는, 대용량 저장 장치로부터 프로그래밍을 루틴 기반으로 다운로드하는 것을 운영 체제가 담당한다. 따라서, 다운로드된 데이터의 인증(authentication) 및 공인(authorization)은 고립형 컴퓨팅 환경이 담당할 것이다.

제조 단계의 후반부에 또는 필드에 전달(최종 사용자에게로의 전달을 포함함)된 이후에 인증 및 공인을 수행하기 위해서는, 고립형 컴퓨팅 환경이 셋업 또는 전송 키들과 초기 프로그램들로 프로그래밍되어 특정 기능성을 제공할 것이다. 그리고, 고립형 컴퓨팅 환경은 전달 사이클의 후반부에 믿을만한 소스에 의해 또는 인증된 데이터에 의해 업데이트될 것이다. 최종 프로그래밍을 보류하는 것에 의해, 고립형 컴퓨팅 환경의 활용 및 융통성이 매우 증대될 수 있다. 최종 프로그래밍을 강요하기 위해서는, 고립형 컴퓨팅 환경에 승인된 버전의 코드를 인스톨할 때까지 또는 시도된 다운로드가 인증에 실패하는 경우 컴퓨터의 활용을 제한하는 제재 조치가 컴퓨터에 부과되어도 좋다.

컴퓨터와 고립형 컴퓨팅 환경이 서로 다른 운영 환경 및 인스톨 환경에서 인터렉트할 수 있게 해주는 애플리케이션 프로그램 인터페이스에 의해 고립형 컴퓨팅 환경의 융통성이 더욱 증대될 수 있다.

실시예

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 실시예를 설명하는 것은 아니며, 이는 비록 불가능하지 않더라도 모든 가능한 실시예들을 설명한다는 것은 실용적이지 않기 때문이다. 현재의 기술 또는 본 출원의 출원일 이후에 개발될 기술들을 사용하여 다수의 대안적인 실시예들이 구현될 수 있지만, 이러한 것들은 여전히 본 발명의 특허청구범위의 범위에 포함되는 것이다.

"본 명세서에서 사용될 때, 용어 '____'는 ...을 의미하도록 정의된다"라는 문장 또는 이와 유사한 문장을 사용하여 본 명세서에서 명백히 정의되지 않는 한, 하나의 용어는 명백하게든 또는 암시적으로든, 그것의 일반적인 의미 또는 보통의 의미 이상으로, 그 용어의 의미를 제한하려는 의도는 아니며, 이러한 용어가 본 특허의 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 의미와 일치하는 방식으로 본 명세서에서 지칭되는 정도로, 단지 독자들을 혼동시키지 않기 위해 명확히 행해지며, 이러한 청구범위의 용어가 암시적으로든 또는 다르게든, 그 단일 의미로 제한되는 것은 아니다. 마지막으로, 임의의 구조의 언급 없이 단어 "수단"이라는 용어와 기능을 언급함으로써 청구항의 구성요소가 정의되지 않는다면, 청구항의 구성요소의 범위가 U.S.C. §112, 6번째 항에 기초하여 해석되어야 하는 것은 아니다.

대부분의 진보성이 있는 기능성 및 다수의 진보성이 있는 원리들은 소프트웨어 프로그램들 또는 명령어들 및 주문형 집적 회로 등의 집적 회로로 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 생성할 수 있을 것이다. 따라서, 간결화 및 본 발명에 따른 원리들 및 개념들을 모호하게 하는 위험성을 최소화하는 관점에서, 존재하더라도, 이러한 소프트웨어 및 IC들에 대한 더 이상의 논의는 바람직한 실시예의 원리들 및 개념들에 대하여 본질적인 것들에 제한될 것이다.

고립형 컴퓨팅 환경의 사용은, 인터넷 서비스 프로바이더, 임대 대리인, 은행 등의 서비스 프로바이더로 하여금 컴퓨터가 해당 서비스 프로바이더의 물리적 제어를 벗어나더라도 컴퓨터 상에 특정 정책들을 강요할 수 있게 해준다. 효과를 발휘하기 위해서는, 고립형 컴퓨팅 환경은 자신이 믿을만한 환경에서 프로그램된다는 것을 뜻하는 특정 레벨의 신뢰도를 요구하거나 또는 보안성 환경으로의 신뢰도 체인을 유지할 것이다.

이미 언급했지만, 요구되는 보안성 환경을 수립하고 유지하기가 보다 용이한 초창기 프로그래밍은 고립형 컴퓨팅 환경을 제한된 세트의 컴퓨팅 플랫폼 옵션들로 제한한다. 초기 셋업 및 사후 프로그래밍 또는 구성의 조합은 고립형 컴퓨팅 환경의 보안성을 보장하는 것을 도우면서 그 융통성에도 도움을 준다.

도 1은 컴퓨터(110)의 형태인 컴퓨팅 장치를 도시한다. 컴퓨터(110)의 컴포넌트는 처리 장치(120), 시스템 메모리(130) 및 시스템 메모리를 포함하는 각종 시스템 컴포넌트를 처리 장치(120)에 연결하는 시스템 버스(121)를 포함하지만 이에 제한되는 것은 아니다. 시스템 버스(121)는 메모리 버스 또는 메모리 컨트롤러, 주변기기 버스 및 각종 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스를 포함하는 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 예를 들어, 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.

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 이동불가식 매체를 포함한다. 예를 들어, 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 정보의 저장을 위해 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 이동불가식 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)에서 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터 등을 구현하고 모든 정보 전달 매체를 포함한다. "피변조 데이터 신호"라는 용어는, 신호내의 정보가 암호화되도록 그 신호의 하나 이상의 특성을 설정 또는 변경시킨 신호를 의미한다. 예를 들어, 통신 매체는 유선 네트워크 또는 다이렉트 유선 접속과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들의 모든 조합이 또한 컴퓨터 판독가능 매체의 범위에 포함될 수 있다.

시스템 메모리(130)는 판독 전용 메모리(ROM)(131) 및 랜덤 액세스 메모리(RAM)(132)와 같은 휘발성 및/또는 비휘발성 메모리의 형태인 컴퓨터 저장 매체를 포함한다. 시동 시 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 장치(120)에 즉시 액세스 가능하고 및/또는 현재 처리 장치(120)에 의해 동작되고 있는 데이터 및/또는 프로그램 모듈을 포함한다. 예를 들어, 도 1은 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136) 및 프로그램 데이터(137)를 도시하고 있지만 이에 제한되는 것은 아니다.

컴퓨터(110)는 또한 기타 이동식/이동불가식, 휘발성/비휘발성 컴퓨터 저장매체를 포함한다. 단지 예를 들어, 도 1은 이동불가식, 비휘발성 자기 매체로의 기록 또는 그로부터의 판독을 위한 하드 디스크 드라이브(141), 이동식, 비휘발성 자기 디스크(152)로의 기록 또는 그로부터의 판독을 위한 자기 디스크 드라이브(151), CD-ROM 또는 기타 광 매체 등의 이동식, 비휘발성 광 디스크(156)로의 기록 또는 그로부터의 판독을 위한 광 디스크 드라이브(155)를 포함한다. 예시적인 운영 환경에서 사용될 수 있는 기타 이동식/이동불가식, 휘발성/비휘발성 컴퓨터 기억 매체로는 자기 테이프 카세트, 플래시 메모리 카드, DVD, 디지털 비디오 테이프, 고체(solid state) RAM, 고체 ROM 등이 있지만 이에 제한되는 것은 아니다. 하드 디스크 드라이브(141)는 통상적으로 인터페이스(140)와 같은 이동불가식 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.

위에서 설명되고 도 1에 도시된 드라이브들 및 이들과 관련된 컴퓨터 저장 매체는, 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 및 컴퓨터(110)의 다른 데이터를 저장한다. 도 1에서, 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 이 컴포넌트들이 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136) 및 프로

그램 데이터(137)와 동일할 수도 있고 다를 수도 있다는 것이다. 이에 관해, 운영 체제(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146) 및 프로그램 데이터(147)에 다른 번호가 주어졌다는 것은 적어도 이들이 서로 다른 사본(copy)이라는 것을 도시한다. 사용자는 키보드(162) 및 통상 마우스, 트랙볼(trackball) 또는 터치 패드로 칭해지는 포인팅 장치(161) 등의 입력 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 장치(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.

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 다른 공통 네트워크 노드일 수 있고, 도 1에는 메모리 저장 장치(181)만이 도시되어 있지만, 통상적으로 컴퓨터(110)와 관련하여 상술된 구성요소의 대부분 또는 그 전부를 포함한다. 도 1에 도시된 논리적 접속으로는 LAN(171) 및 WAN(173)이 있지만, 다른 네트워크를 포함할 수도 있다. 이러한 네트워킹 환경은 사무실, 회사 전체에 걸친 컴퓨터 네트워크, 인트라넷 및 인터넷에서 일반적인 것이다.

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) 상에서의 통신을 설정하기 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. 예를 들어, 도 1은 애플리케이션 프로그램(185)이 메모리 장치(181)에 상주하는 것을 도시하고 있지만 이에 제한되는 것은 아니다.

통신 접속들(170, 172)은 장치가 다른 장치들과 통신할 수 있게 해준다. 통신 접속들(170, 172)은 통신 매체의 일 예이다. "피변조 데이터 신호"라는 용어는, 신호 내의 정보를 암호화하는 방식으로 그 신호의 하나 이상의 특성을 설정 또는 변경시킨 신호를 의미한다. 예를 들어, 통신 매체는 유선 네트워크 또는 다이렉트 유선 접속과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 컴퓨터 판독가능 매체는 저장 매체 및 통신 매체 양자 모두를 포함할 수 있다.

도 2를 참조하여 보다 구체적으로 논의되는 고립형 컴퓨팅 환경(125)은 프로그램들 및 데이터를 저장하고 이를 실행한다. 고립형 컴퓨팅 환경(125)은 컴퓨터(110)의 사용자와 서비스 프로바이더간 이해관계에 관한 동의 조항들을 컴퓨터(110)에 강요하도록 전개되고 구성된다.

고립형 컴퓨팅 환경(125)은 하나 이상의 방식으로 예시될 수 있다. 하나 이상의 분산형 소자들로 구현되는 경우, 고립형 컴퓨팅 환경(125)은 컴퓨터의 마더보드(도시되지 않음) 상에 전개될 수 있다. 마더보드는 소정 애플리케이션에 적합한 기반 기술을 탑재하는 임의의 회로 상호접속 및 컴포넌트일 수 있고, 이는 유리섬유 재료에서부터 성형 에폭시 수지, 마일라(mylar), 세라믹 등 다양하다. 고립형 컴퓨팅 환경(125)이 마더보드 상에 또는 마더보드 내에 배치되는 경우, 고립형 컴퓨팅 환경(125)은 에폭시로 피복되거나 상호접속 레이어들 또는 컴포넌트들 아래에 매립될 수 있다. 고립형 컴퓨팅 환경(125)을 피복하거나 매립하는 것은, 고립형 컴퓨팅 환경(125) 자체에 의해, 관련 파워 및 고립형 컴퓨팅 환경(125)으로의 접지 접속이나 고립형 컴퓨팅 환경(125)으로의 어드레스 접속들을 제거하거나 탬퍼링(tampering)하는 어려움을 증가시키게 될 수 있다. 이상적으로는, 고립형 컴퓨팅 환경(125)의 제거 또는 탈착(de-lidding)은 마더보드 및/또는 주변 컴포넌트들에 영구적인 손상을 초래하게 되고 컴퓨터(110)가 동작될 수 없게 한다.

고립형 컴퓨팅 환경(125)의 또 다른 예시가 도 1에 도시되어 있는데, 여기서 고립형 컴퓨팅 환경(125)은 처리 장치(120)에 통합된다. 이와 같이 처리 장치에 배치되는 것은 물리적 공격들에 대한 내성이 증가될 뿐만 아니라 처리 장치 레지스터들에 대한 액세스 및 데이터 시퀀스들의 모니터링이 보다 우수하게 되는 이점을 제공하게 될 것이다.

도 2를 참조하여, 이하 개략적이고 대표적인 고립형 컴퓨팅 환경이 논의되고 설명될 것이다. 이러한 고립형 컴퓨팅 환경은 위에 소개된 고립형 컴퓨팅 환경(125)이거나 또는 이와 유사한 것일 수 있다. 고립형 컴퓨팅 환경(125)은 메모리(202), 논리 회로(204) 및 클럭 또는 타이머(206)를 포함할 수 있고, 예를 들어, 타이머(206)는 실제 시간의 간격을 카운트하는 것에 의해 클럭을 구현하는데 사용될 수 있다. 메모리(202)는 휘발성 및 비휘발성 메모리 양자 모두를 포함할 수 있다.

고립형 컴퓨팅 환경(125)은 디지털 서명 검증 회로(208)를 더 포함할 수 있다. 예를 들어, 서버(도시되지 않음)의 검증 등 외부 엔티티에 대한 일방(one-way) 검증이 요구되는 경우, 난수 발생기(210)가 디지털 서명 검증 회로(208)의 일부일 수 있다. 디지털 서명 기술은 잘 알려져 있으므로, 해싱, 서명 검증, 대칭형 및 비대칭형 암호화 알고리즘 및 이들 각각의 키들 등은 본 명세서에 상세히 논의되지 않을 것이다.

고립형 컴퓨팅 환경(125)의 블럭들은 버스(212)에 의해 연결될 수 있다. 버스(212)는 외부 액세스를 위해 사용되는 시스템 또는 처리 장치 버스(214)와는 구별된다. 이와 같이, 별도의 버스들을 사용하는 것은 버스(212)에 의해 전달되는 데이터에 대한 액세스를 제한함으로써 보안성을 향상시켜 줄 것이다. 버스(212)는 밸런스드 데이터 라인들 등 보안 예방 조치들을 통합하여 메모리(202)에 저장된 암호화 키들(216)에 대한 파워 공격들이 보다 어렵게 하여 줄 것이다.

메모리(202)는, 암호화 키들(216)을 저장하는 것 이외에도, 적어도 하나의 검증 프로그램(218) 및 적어도 하나의 강제 프로그램(220)을 저장하는 비휘발성 메모리를 포함할 수 있다. 이들 프로그램은 이하 보다 상세히 논의될 것이다. 메모리(202)에는, 예를 들어, 해시 코드들 및/또는 공지된 BIOS 코드 또는 애플리케이션 프로그램들과 관련된 디지털 서명 정보 등의 기타 데이터(222)가 메모리(202)에 저장될 수 있다. 메모리(202)에 저장될 수 있는 다른 예의 데이터(222)로는 컴퓨터(110)의 현재 상태에 관련된 컴플라이언스(compliance) 데이터 또는 검증 프로그램(218) 또는 강제 프로그램(220)에 대한 다운로드된 업데이트의 검증을 위한 인증 정보 등이 있다. 메모리(202)의 비휘발성 메모리는 신뢰성있고 안전한 부트 프로세스를 운영 체제(144)와는 구별될 수 있게 해준다.

검증 프로그램(218) 및 강제 프로그램(220)이 고립형 컴퓨팅 환경(125)에 저장되는 것으로 도시되었지만, 예를 들어, 프로그램들의 디지털 서명 또는 해시를 메모리(202)의 데이터부(222) 등 고립형 컴퓨팅 환경에 저장해 놓고 그 프로그램들을 외부에 저장하여도 좋다. 대안적으로는, 디지털 서명들이 고립형 컴퓨팅 환경(125) 외부에 저장되는데, 이는 메타데이터로서 이들 프로그램에 첨부될 수 있다. 애플리케이션 프로그램을 모니터링하거나 측정할 때, 고립형 컴퓨팅 환경(125)은 프로그램의 실행 이전에 또는 프로그램의 실행 중에 애플리케이션 프로그램의 해시 또는 디지털 서명을 검증할 것이다. 프로그램들(218, 220) 및 메모리(202)에 저장된 데이터는 현금 지불형, 사용 시간당 요금제 비즈니스 모델과 관련되는 보안의 일부이기 때문에, 무권한 액세스 또는 탬퍼링으로부터 데이터가 보호되는 것이 중요할 것이다. 메모리(202)의 비권한 액세스는 논리 회로(204) 또는 디지털 서명 검증 회로(208)를 사용하거나 또는 이들 양자의 조합을 사용하여 제한될 수 있다. 메모리에 대한 액세스는 알려진 프로그램 코드, 즉 고립형 컴퓨팅 환경(125)이 신뢰하는 프로그램을 실행하는 프로세스로 한정될 것이다. 이러한 프로그램 코드는 검증 프로그램(218) 또는 강제 프로그램(220)일 수 있다. 그러나, 다른 프로그램들이 메모리(202)에 대한 액세스를 승인받을 수도 있다. 수리 또는 유지보수가 필요한 경우, 수리를 행하기 위해 적절한 자격을 갖는 네트워크 연결된 디바이스 상에서 지원되는 서비스 프로세스에게 메모리(202)에 대한 액세스가 승인될 수 있다.

고립형 컴퓨팅 환경(125)은 여러 기능들을 구비할 것이다. 고립형 컴퓨팅 환경의 일 기능은 자신을 비권한 업데이트 및 탬퍼링으로부터 보호하는 것이다. 고립형 컴퓨팅 환경(125)에 저장된 프로그램들 및 데이터는 제조시 주입되거나 또는 고립형 컴퓨팅 환경(125) 자신이 인증하는 서명으로 정확하게 서명되는 경우 다운로드 되거나 할 수 있다. 다른 기능은 컴퓨터(110)의 상태를 모니터링 및/또는 측정하여 컴퓨터(110)의 상태에 해킹 또는 기타 비권한 변경이 진행중인지 또는 발생하였는지를 판정하는 것이다. 모니터링 및 측정의 다른 양상은 리소스를 준비하는 것 및 이벤트 디스패처나 밸런스 매니저 등의 보안 기능들을 호스팅하는 것과 연관되는 기능들과 관련된 컴퓨터(110) 상태의 정당한 변화를 지원하는 것이다. 3번째 기능은 현재 BIOS 코드를 검증하고 BIOS 코드에 대한 업데이트 및 확장을 검증하는 것이다. 고립형 컴퓨팅 환경(125)의 또 다른 기능은 프로그램들 및 만료 일자들을 계량하기 위한 시간의 소스로서 신뢰성있는 클럭 또는 타이머를 제공하는 것이다. 클럭 또는 타이머는 또한 고립형 컴퓨팅 환경(125)이 컴퓨터(110)에 대해 액세스하는 것이 루틴하게 허용되는 것과 CPU 또는 버스 사이클들이 "부족하게(starved)" 되지 않는 것을 보장한다. 다른 기능은 컴퓨터(110)에서 비순응 상태가 판정될 때 제재를 가하는 것일 수 있다.

고립형 컴퓨팅 환경(125)의 또 다른 기능은 컴퓨터 사용을 관측하고 이러한 컴퓨터 사용이 프로바이더 또는 법적 소유자가 제공하는 비즈니스 용어들 및 정책들에 순응하는지 확인하는 독립적 슈퍼바이저 역할을 하는 것이다. 현금 지불형 및 가입 컴퓨터들이 이러한 기술의 예이다.

비권한 업데이트 및 탬퍼링으로부터 보호하기 위해, 메모리(202)가 보호되어야 한다. 이를 달성하기 위해, 메모리(202)는 예를 들어 컴퓨터(110)의 보안 동작 모드의 제어하에 디지털 서명에 의해 인증되는 업데이트 루틴 등 특정 프로그램에의해서만 액세스될 수 있게 된다. 메모리(202)는 운영 체제 또는 커넬 등의 다른 실행 환경에 의해 실행되는 프로그램에 의해서는 액세스될 수 없도록 될 것이다. 커넬은 통상적으로 컴퓨터(110)가 부팅중일 때 동작한다. 예를 들어, Intel(등록상표) 제조의 x86 프로세서는 여러 모드에서, 또는 링 구조의 실행(rings of execution)으로 동작될 수 있다. 링 0-2는 커넬이

차지하고, 링 3는 "사용자 모드(user mode)" 프로세스가 차지한다. 3번째 모드인 시스템 관리 모드(SMM; System Management Mode)는 BIOS가 차지한다. 보안성 메모리(202)에 대한 액세스를 갖는 프로그램은 커넬의 범위 밖에 있으므로 SMM에서 동작될 것이지만, BIOS를 보호해야 할 것이다. 대안적으로, 고립형 컴퓨팅 환경(125)은 전용 디바이스를 사용하여 CPU와는 독립적으로 구현될 수 있다.

고립형 컴퓨팅 환경(125)의 보안을 위해, 고립형 컴퓨팅 환경(125) 이외의 디바이스들은 단지 고립형 컴퓨팅 환경 메모리(202)에 전용인 물리적 메모리에 액세스할 수 없다. 이는 프로그램들(218, 220), 키들(216) 및 상태/동작 데이터(222)를 포함하는 고립형 컴퓨팅 환경 메모리(20)의 동작과 관련된 데이터를 고립형 컴퓨팅 환경만이 액세스하고 변경할 수 있도록 보장하는 유일한 방법이다. 디지털 서명 검증 회로(208)는 외부로부터, 즉 운영 체제(144)를 통해 들어오는 메모리(202)에 대한 모든 변경 요청들을 검증하는데 사용될 수 있다. 내부적으로 저장된 키들을 사용하여 디지털 서명을 확인하는 것에 의해, 신뢰성이 확보되지 않은 소스, 즉 운영 체제(144)에 의해 수신되는 데이터에 대한 신뢰가 성립될 수 있다.

도 3은 고립형 컴퓨팅 환경(125)의 논리적 개관을 도시한다. 암호화 서비스(302)는 내부 레이어들 또는 서비스들을 탬퍼링 또는 비권한 액세스로부터 보호하한다. 암호화 키들(304)은 도 2의 메모리(202) 등 보안성 메모리에 저장될 수 있다. 구성 테이블(306) 및 코어 서비스(308)는 고립형 컴퓨팅 환경(125)의 특징들 및 기능들을 프로그래밍하는데 사용될 수 있다. 애플리케이션 프로그램 인터페이스(API; Application Program Interface)(310)는 고립형 컴퓨팅 환경(125)과의 쌍방향 데이터 전송을 지원하는데 사용될 수 있다. 애플리케이션 프로그램 인터페이스(310)를 사용하는 것에 의해, 유효 통신의 커맨드 세트가 특정 포맷, 스키마 및/또는 보안 정책들로 제한될 수 있다. 이러한 형태에서는, 미리-결정된 세트의 특징들을 따르지 않는 데이터 전송들은 더 이상 처리되지 않고 거절될 것이다. 예를 들어, ICE의 일부 양상들을 업데이트하는 페이로드를 갖는 메시지는 신뢰된 권한자에 의해 디지털 서명될 것이 요구된다.

업데이트 서비스(312)는 수신 메시지의 인증에 응답하여 키들(304), 구성 테이블(306) 및 코어 서비스(308)를 변경하는데 사용될 수 있다. 도 2의 강제 프로그램(220)과 동일하거나 또는 이와 유사한 제재 서비스(314)는, 초기 제조 이후 배달되는 중에 또는 정상 동작동안 비순응 조건이 식별된 이후에 컴퓨터(110)의 기능성을 제한하도록 트리거될 수 있다. 제재 서비스는 활성화 메커니즘을 사용하여 컴퓨터의 기능을 지연시킬 수 있다. 활성화 메커니즘은 버스 드라이버 또는 클럭 회로에 내장되거나, 또는 마이크로 코드와 같이 마이크로프로세서(도시되지 않음)의 실리콘 컴포넌트일 수 있다. 대안적으로, 활성화 메커니즘은 컴퓨터의 주기적 리셋팅일 수 있다. 일 실시예에서, 리셋들간 주기는 랜덤하게 변화하지만, 일반적으로, 진단 및 유지보수 기능들이 수행될 수 있도록 충분히 길 것이다. 활성화 메커니즘 및 그 물리적 징후는 탬퍼-내성이 있도록 이루어질 수 있고, 이는 제재를 타파하기 위해 활성화 메커니즘을 디스에이블링하는 것이 사용될 수 있기 때문이다.

호스티드 애플리케이션(316) 또는 기타 기능적 프로그램이 고립형 컴퓨팅 환경(125)과 관련될 수 있다. 호스티드 애플리케이션(316)은 고립형 컴퓨팅 환경(125)에 초기에 인스톨되거나, 또는 예를 들어 제조 이후, 소매점에서 차후 인스턴스화되거나, 또는 고객에게 전달된 이후 인스턴스화될 수 있다. 호스티드 애플리케이션(316)은 사용당 요금제 컴퓨터의 계량과 관련될 수 있고, 측정된 기능들에 대한 감시기로서 동작하거나, 또는 보안성 측정이 바람직한 임의의 애플리케이션, 기능 또는 리소스일 수 있다. 단일 고립형 컴퓨팅 환경(125)에 의해 하나 이상의 호스티드 애플리케이션(316)이 지원될 수 있다[고립형 컴퓨팅 환경(125)이 'ICE'라 불리우는 경우, 호스티드 애플리케이션들은 'icicles'라 불리울 것이다.].

코어 서비스(308)와 호스티드 애플리케이션(316)간 관계는 애플리케이션에 의존하여 변경되고 시간에 따라 변경될 것이다. 예를 들어, 일 실시예에서는, 코어 서비스(308)가 고립형 컴퓨팅 환경(125)의 업데이트를 제어할 것이며, 이는 즉 업데이트에 대한 게이트키퍼 역할을 한다는 것이다. 다른 실시예에서는, 호스티드 애플리케이션(316)이 게이트키퍼 역할을 할 것이다. 또 다른 실시예에서는, 토큰 제출시 코어 서비스(308)가 호스티드 애플리케이션(316)에게 게이트키핑 책임을 양도 또는 위임할 것이다. 복잡하지만, 또 다른 실시예는 코어 서비스(308)와 호스티드 애플리케이션(316)간 게이트키핑 책임을 공유할 수 있다.

업데이트 서비스(312)는, 애플리케이션 프로그램 인터페이스(310)를 통해 수신되는 검증된 메시지에 응답하며, 코어 서비스(308), 구성 테이블(306), 키들(304) 및/또는 호스티드 애플리케이션(316)을 전체적으로 업데이트하거나 리프로그램하는데 사용될 수 있다. 코어 서비스(308)를 업데이트할 때는, 하나 이상의 동작 루틴들이 업데이트될 것이다. 동작 루틴들은 컴퓨터(110) 상에서 이루어지는 측정들 및 최종-사용자 동작에 부과되는 제재들에 영향을 줄 수 있다. 구성 테이블(306)에 대한 업데이트는 지불 스케쥴, 측정 기준, 디폴트 셋팅 등을 포함할 수 있다. 키들에 이루어지는 업데이트들은 예전 키들을 대체하거나 또는 새로운 키들을 인스톨하여 후속 메시지들의 확인을 가능하게 할 수 있다. 따라서, 검증된 메시지에 수신되는 데이터는 코어 서비스들(308), 구성 테이블(306) 셋팅들, 호스티드 애플리케이션(316) 또는 키들(304)이 사용하는 루틴들 중 임의의 것 또는 모두를 포함할 수 있다. 특히 손상된 것일 수 있는 이전 버전의 업데이트를 재로딩하는 것을 방지하기 위해서는, 타임스탬프들의 일련 번호들이 검증된 메시지의 일부로서 사용될 수 있다.

도 4는 컴퓨터(110) 등의 컴퓨터에서 대표적인 라이프사이클 단계들을 도시한다. 또한, 고립형 컴퓨팅 환경(125) 등 컴퓨터의 보안 영역들을 보호하는데 사용될 수 있는 대표적인 임시 암호화 키들이 도시된다. 칩 테스팅과 같이 빠른 제조 프로세스 초기에, 초기 키(402)가 컴퓨터(110) 또는 그 컴포넌트에 주입될 수 있다. 이러한 주입은 일반적으로 메모리 회로(202)에 직접 기입함으로써 발생될 수 있는데, 이는 메모리 회로(202)가 이러한 직접 배선 프로세스로부터 메모리(202)를 보호하기에 충분한 보호 회로 및 물리적 경화제로 둘러쌓이기 이전에 행해진다.

제조(404) 후, 초기 키(402)는 셋업 키(406)로 대체될 것이다. 셋업 키(406)는 제조자, 또는 그 서브프로세스와 프로그래밍 페이즈(408) 사이에 공유될 수 있다. 프로그래밍 중, 운영 체제 및 애플리케이션이 인스톨될 것이다. 컴퓨터(110)의 최종 도착지가 알려지면 로컬리제이션(localization)이 발생할 것이다. 몇몇 신뢰된 환경에서는, 고립형 컴퓨팅 환경(125)이 제조 중 공개될 것이고, 이는 보다 표준형인 고립형 컴퓨팅 환경이 인스톨될 수 있게 해준다. 특정 운영체제 및 기능적 환경들에 속박되는 것은 암호화 잠금에 의해 부과되는 제한 없이 연기될 것이다. 일 실시예에서는, 운영 제제의 인스톨 및 셋업이 고립형 컴퓨팅 환경(125)의 셋업 및 프로그래밍과 관련될 것이다. 다른 실시예에서는, 모든 고립형 컴퓨팅 환경들에게 해당 그룹에 있는 모든 머신들이 단일 키로 프로그램되도록 해주는 표준 키 또는 그룹 키가 주어질 것이다. 그리고, 고립형 컴퓨팅 환경(125)에 프로그램되는 고유 식별자가 사용되어 각 개별 고립형 컴퓨팅 환경(125)에 대한 암호화 키들 및 환경을 구별하고 개인화할 것이다. 프로그래밍(408) 중, 임시 암호화 키(410) 또한 인스톨될 것이다. 임시 키(410)는 프로그래밍 동작과 셋업/구성 동작(412) 사이에 공유될 것이다. 셋업/구성(412)은 소매 배달 센터 또는 서비스 프로바이더 스테이징(staging) 영역일 수 있다. 셋업 및 구성 중, 개별 사용자 계좌가 프로그램되고, 이메일 서비스가 수립되고, 인터넷 접속이 프로그램되며, 사용 당 요금제 컴퓨터인 경우, 사용 계획 및 초기 동작 대금이 인스톨될 것이다.

셋업/구성 프로세스 중, 동작 키(414) 및 일부 경우에는 유지보수 키(418)가 인스톨될 것이다. 동작 키(414) 또는 키들은 최종 사용자와 공유되지 않지만, 동작 키(들)는 컴퓨터(110)에 상주하고, 보다 구체적으로는 고립형 컴퓨팅 환경(125)에 저장될 것이다. 여러 세트의 대칭형 키들이 대칭형 키 암호화 환경에 인스톨되거나, 또는 비대칭형 암호화가 사용되는 경우 루트 인증서 및 공개 키가 인스톨될 것이다. 이 시점에 컴퓨터는 최종 사용자에 의한 유익한 동작(416)에 대해 준비될 것이다. 동작(416) 중, 유지보수가 필요할 것이다. 특별한 액세스가 요구되는 경우, 동작 키(414)는 통과되고 유지보수 키(418)가 사용되어 컴퓨터(110)를 특별 유지보수 모드(420)에 놓게 된다. 유지보수 프로세스(420)가 완료될 때, 컴퓨터는 동작 모드(416)로 되돌아올 것이다.

동작 키(414)는 정상 동작에 사용되어, 예를 들어, 현금 지불형 계좌에 값을 가산한다. 다른 특별 목적 키들이 임의의 시점에 보조적 용도로 인스톨될 수 있는데, 이는 디지털 서명된 업데이트를 포함하지만 이에 제한되는 것은 아니다. 제조 및 배달 프로세스의 각 단계에서, 코어 서비스(308)의 코드, 구성 테이블(306) 및 키들(304)은 해당 단계에서 수행되는 동작들을 반영하거나, 또는 다음 단계를 위해 스케쥴링된 액티비티를 허용하도록 업데이트된다. 이러한 순차 프로세싱에서의 키 관리 프로세스는 해당 분야에 공지되어 있다.

컴퓨터(110) 또는 고립형 컴퓨팅 환경(125) 등의 컴퓨팅 환경의 라이프사이클의 각 단계에서는, 인스톨된 키들 및 인증을 위해 부여된 토큰들을 사용하여 신뢰가 수립될 수 있다. 디지털 서명된 업데이트 등 수용가능한 토큰들을 부여함으로서, 송신측에 의해 신뢰가 수립된다. 제조 프로세스 초기에만 컴퓨터(110) 또는 고립형 컴퓨팅 환경(125)이 초기 키(402)의 주입 등 신뢰된 액션에 대한 지원 환경에 의지할 것이다. 이러한 초기 신뢰 환경을 넘어서면, 컴퓨터(110) 또는 고립형 컴퓨팅 환경(125)은 부여된 선험적 임의 데이터를 신뢰할 것으로 기대되지 않는다. 또한, 고립형 컴퓨팅 환경은 요청을 인증하고 그 권한을 검증한다. 예를 들어, 다양한 고립형 컴퓨팅 환경 컴포넌트들의 업데이트 및 인스톨에 변화하는 정책들을 적용할 수 있을 것이다. 예를 들어, 코어 서비스 및 구성 서비스를 업데이트하는 것은 매우 엄격하게 제한된 그룹에게 허가되는 한편, 호스티드 애플리케이션을 업데이트하는 것은 그 저작자에게 허가된다.

고립형 컴퓨팅 환경(125)의 초기 프로그래밍 및 초기 키(402)의 주입 이후 임의의 시간에, 예를 들어, 제조(404) 단계에, 고립형 컴퓨팅 환경(125)은 컴퓨터(110) 상에 제재를 부과할 것이다. 이러한 제재는, 기능성을, 컴퓨터(110)의 프로그래밍 및 셋업에 있어서 차후 단계들을 수행하는데 적합하지만 최종 사용자에 의한 이로운 사용을 방지하기에는 충분한 최소 세트의 유용한 루틴들 또는 서비스들로 제한한다. 이러한 제재로는 영구적 리셋, 명령어 세트 감소, 동작 속도 저감, 스크린 영역 또는 컬러의 최소화 등이 포함되지만, 배달 단계 중 이러한 제재가 변경될 수 있다. 이들 초기 제재는 고립형 컴퓨팅 환경(125)이 최종 사용자 동작에 적합한 코어 서비스(308), 구성 테이블(306) 셋팅 및 키들(304)로 업데이트될 때 취소될 것이다. 컴퓨터(110), 또는 보다 구체적으로는, 고립형 컴퓨팅 환경(125)의 구성에 대한 검증의 일부로서, 코어 서비스(308)의 특징이 디지털적으로 검증될 수 있다. 예를 들어, 이러한 검증은 루트 인증서 또는 공지된 해시 등의 디지털 서명이나 해시 및 기존 수립 신뢰 토큰을 사용할 것이다.

예를 들어, 적절한 소프트웨어 모듈들을 연속 인스톨하는 것에 의해, 인스턴스화 프로그래밍 및 셋업이 일련의 단계로서 발생할 때, "최종(last)" 모듈의 인스톨에 의해 제재가 취소될 수 있다. 즉, 최종 모듈은 하부 모듈들이 인스톨되고 보존되는 것을 판정하고 나서 제재를 취소할 수 있다.

대안적으로는, 제조 또는 고객 주문처리 프로세스에서의 후반부까지 제재가 부과되지 않을 수 있다. 일 실시예에서, 제재 서비스(314)는, 고립형 컴퓨팅 환경(125)이 업데이트되지 않으면 제재를 활성화하기 이전에, 리셋 횟수, 캘린더 날짜 또는 구축 경과일 등의 트리거 이벤트를 모니터한다.

도 5에서는, 컴퓨터의 고립형 컴퓨팅 환경에서의 프로그래밍을 연기하는 방법을 도시하는 흐름도가 논의되고 설명될 것이다. 컴퓨터(110) 등의 컴퓨터가 제공된다(502). 컴퓨터가 물리적 고립형 컴퓨팅 환경을 포함하거나 또는 고립형 컴퓨팅 환경이 운영 체제에 의해 구현되거나 할 수 있다. 일반적으로, 하드웨어 고립형 컴퓨팅 환경이 탬퍼링 및 공격으로부터 보다 안전하다.

최종 사용자에게로의 배달 이전에, 바람직하게는 제조 프로세스의 초기에, 고립형 컴퓨팅 환경은 전송 기능성을 갖도록 프로그램된다(504). 전송 기능성은 현재 및/또는 후속 단계들을 고객 배달을 지향하도록 할 수 있는 능력으로 제한되지만, 최종, 최종-사용자, 기능성을 포함하지 않을 수 있다. 전송 기능성을 프로그래밍하는 것은 키들(406, 410) 등의 전송 암호화 키 및 코어 서비스(308)에서의 적어도 하나의 다운로드 기능을 프로그래밍하는 것을 포함할 수 있다. 다운로드 기능은 중간 기능성 또는 최종-사용자 기능성의 후속 다운로드를 가능하게 할 수 있다.

제조 및 배달 사이클의 강제 완료를 돕기 위해, 컴퓨터의 동작이 제한되거나(506) 또는 감소될 것이다. 이미 논의되었고 우선권 서류들에서 논의되었던 바와 같은 다수의 옵션들이 컴퓨터 동작을 제한하는데 사용될 수 있고, 이는 처리 장치(120)의 기능성을 감소시키는 것 또는 컴퓨터(110)를 주기적으로 리셋하는 것을 포함할 수 있다.

인입 메시지가 수신되고 인증될 것이다(508). 이러한 데이터는 업데이트된 기능성에 대응하거나 또는 키들(304), 코어 서비스들(308) 또는 구성 테이블(306)을 업데이트하기 위한 구성 데이터 등의 기타 데이터 셋팅에 대응할 수 있다. 고립형 컴퓨팅 환경은 이러한 인증된 데이터에 의해 리프로그램될 것이다(510). 이러한 데이터가 디지털 서명 등의 그 자신의 보안 토큰을 포함하거나, 또는 고립형 컴퓨팅 환경이 데이터의 인증을 검출하기 위해 전송키를 사용하여 호스트와의 인증 세션을 수립할 수 있다. 동작-레벨 코드가 인스톨되어 검증되는 경우, 컴퓨터(110)의 전송 구성과 관련된 제재가 제거되어(512), 사용자에 의한 이로운 사용을 가능하게 한다.

위에 설명된 바와 같은 프로세스를 따르는 것에 의하면, 고립형 컴퓨팅 환경(125) 또는 기타 보안 환경은, 고립형 컴퓨팅 환경에 오류를 일으킬 가능성이 적은 제조 프로세스 초기에 보장될 수 있다. 기타 중간 제조 및 배달 동작들은 각 단계가 적절한 키를 부여하는 것에 의해 변화를 줄 수 있는 자신의 권리를 입증할 수 있는 정도로 업데이트 및 변화를 행할 권한이 주어진다. 구성 테이블(306) 및 코어 서비스(308)의 최종 업데이트는 고립형 컴퓨팅 환경이 보다 넓은 범위의 컴퓨터 구성, 운영 체재 및 로컬리제이션을 통해 사용될 수 있도록 해준다. 이와 동시에, 고립형 컴퓨팅 환경은 제조 및 배달을 지원하기에 충분한 기능성을 유지하지만, 중간 암호화 키의 사용을 통해 자신의 완전성/신뢰가치를 보존한다.

본 발명의 각종 상이한 실시예들의 상세한 설명이 상술된 텍스트에 설명되어 있지만, 본 발명의 범위는 본 명세서의 후반부에 설명되어 있는 청구범위의 단어에 의해 정의된다는 것을 이해해야 한다. 상세한 설명은 단지 예시적인 것으로 해석되어야 하며, 모든 가능한 실시예를 설명하는 것은 비현실적이기 때문에 상세한 설명은 본 발명의 모든 가능한 실시예를 설명하지 않는다. 현재의 기술 또는 본 특허의 출원일 이후에 개발되는 기술을 이용하여 각종 대안의 실시예들이 구현될 수 있으나, 역시 본 발명을 정의하는 청구범위의 범위내에 있을 것이다.

따라서, 본 발명의 취지 및 범위를 벗어나지 않으면서, 본 발명에 설명되고 도시된 기술 및 구조에 많은 변형 및 수정이 있을 수 있다. 따라서, 본 명세서에 설명된 방법 및 장치는 단지 예시적인 것이며, 본 발명의 범위를 제한하지 않는다는 것을 이해해야 한다.

도면의 간단한 설명

도 1은 컴퓨터의 개략 블럭도.

도 2는 고립형 컴퓨팅 환경의 개략 블럭도.

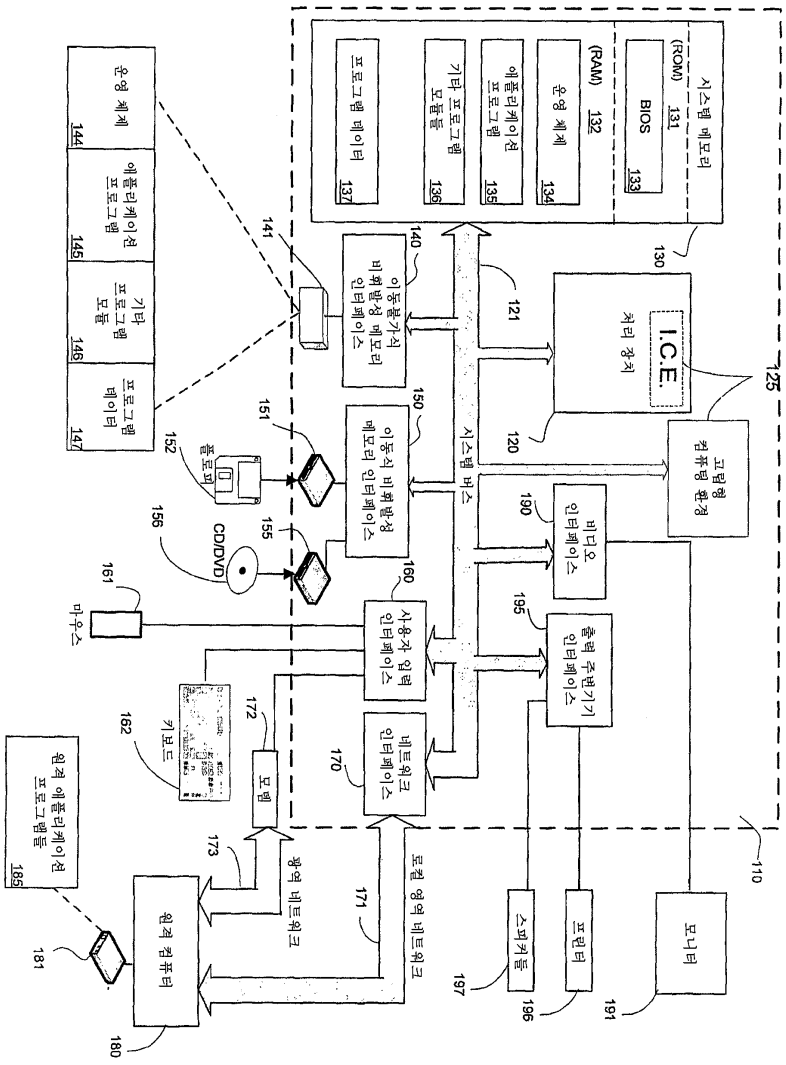
도 3은 고립형 컴퓨팅 환경 서비스들간 기능적 관계를 도시하는 개략 블럭도.

도 4는 고립형 컴퓨팅 환경의 연속적인 프로그래밍을 위한 시스템을 나타내는 블럭도.

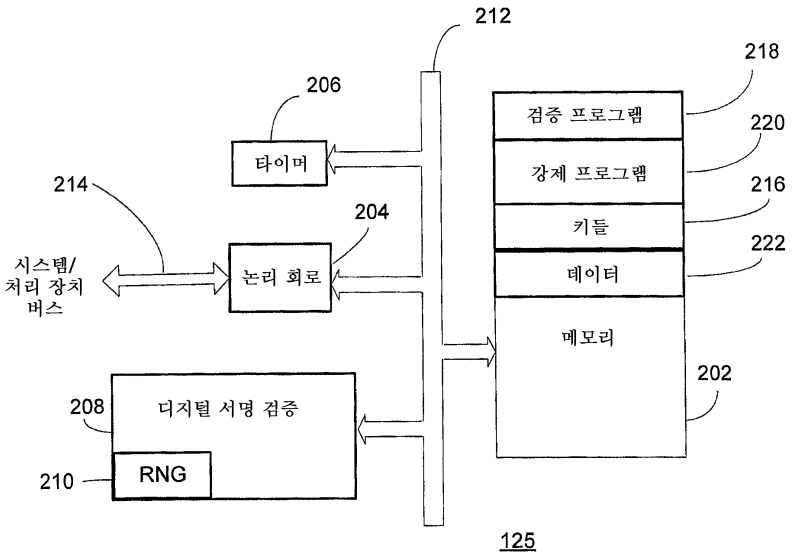
도 5는 고립형 컴퓨팅 환경의 프로그래밍 및 셋업을 보류하는 방법을 도시하는 흐름도.

도면

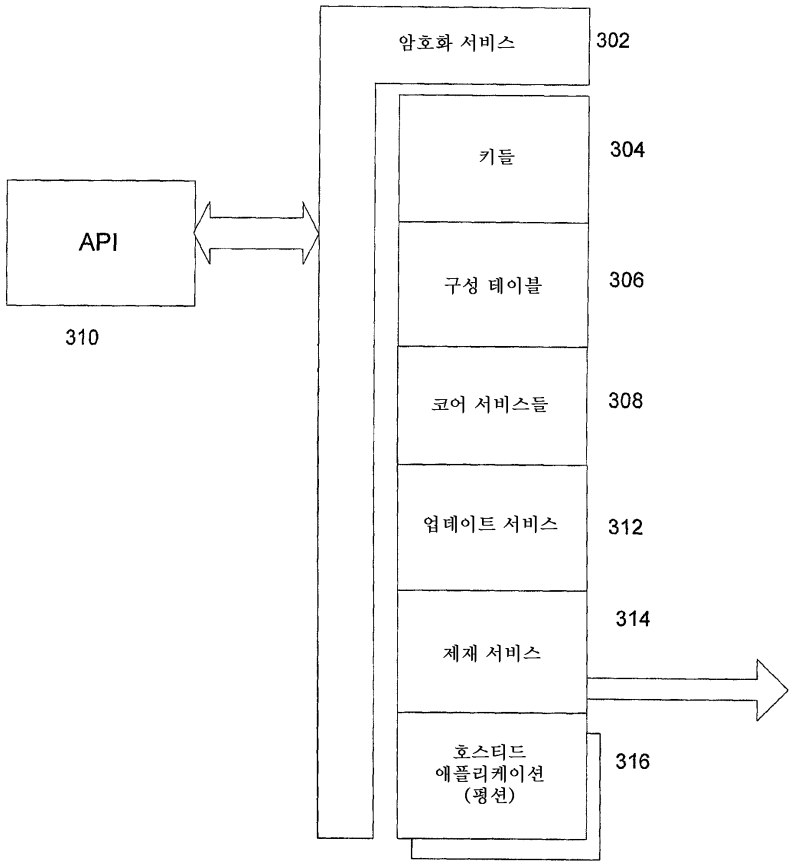
도면1



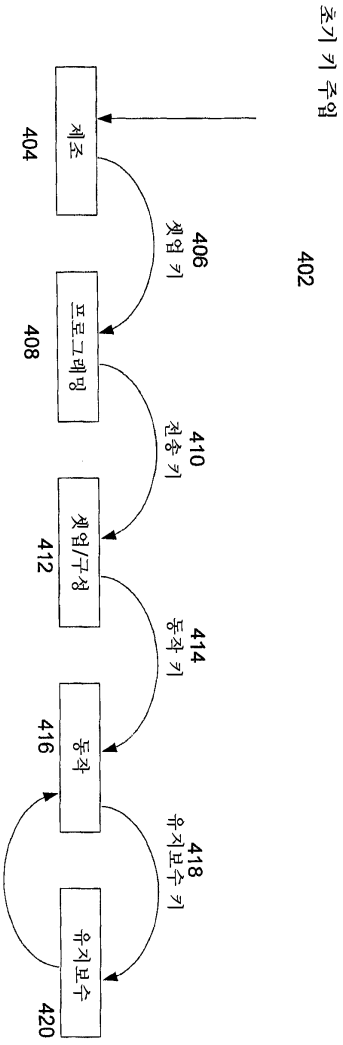
도면2


212
218
206
검증 프로그램
220
타이머
강제 프로그램
214
204
216
키들
시스템/
처리 장치
버스
논리 회로
222
데이터
메모리
202
208
디지털 서명 검증
RNG
210
125

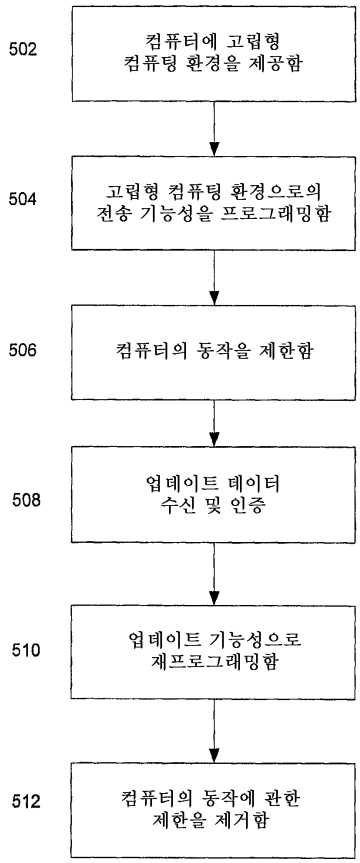
도면3


암호화 서비스
302
키들
304
API
구성 테이블
306
310
코어 서비스들
308
업데이트 서비스
312
314
제재 서비스
호스티드
애플리케이션
(평션)
316

도면4


초기 키 주입
402
제조
404
셋업 키
406
프로그래밍
408
권송 키
410
셋업/구성
412
동작 키
414
동작
416
유지보수 키
418
유지보수
420

도면5


502
컴퓨터에 고립형 컴퓨팅 환경을 제공함
504
고립형 컴퓨팅 환경으로의 전송 기능성을 프로그래밍함
506
컴퓨터의 동작을 제한함
508
업데이트 데이터 수신 및 인증
510
업데이트 기능성으로 재프로그래밍함
512
컴퓨터의 동작에 관한 제한을 제거함