(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년03월29일
(11) 등록번호 10-0950007
(24) 등록일자 2010년03월22일

(51) Int. Cl.
G11B 20/10 (2006.01) *G06F 11/30* (2006.01)
H04L 9/32 (2006.01) *G06F 12/14* (2006.01)
(21) 출원번호 10-2005-7025111
(22) 출원일자 2004년06월28일
심사청구일자 2008년04월24일
(85) 번역문제출일자 2005년12월27일
(65) 공개번호 10-2006-0039405
(43) 공개일자 2006년05월08일
(86) 국제출원번호 PCT/US2004/021048
(87) 국제공개번호 WO 2005/001666
국제공개일자 2005년01월06일
(30) 우선권주장
60/481,034 2003년06월27일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
US04792895 A1
US20020161996 A1
US4831541 A
전체 청구항 수 : 총 26 항

(73) 특허권자
디즈니엔터프라이지즈,인크.
미합중국캘리포니아주버어뱅크시사우스뷰나비스타스트리트500(우편번호91521)
(72) 발명자
왓슨 스콧
미국 캘리포니아주 91387 산타 클라리타 마이클 크레스트드라이브 15355
(74) 대리인
김태홍, 송승필

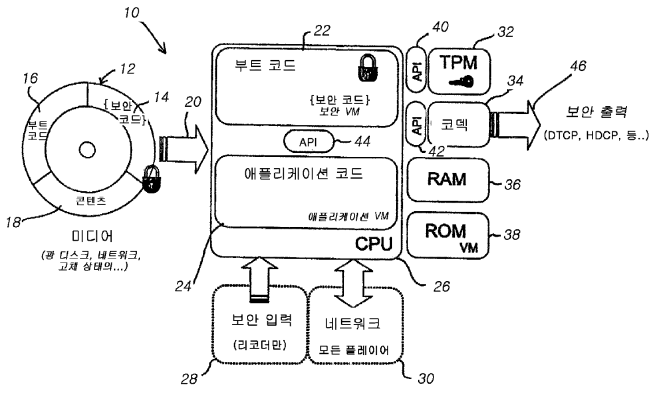
심사관 : 변성철

(54) 차세대 미디어 플레이어층에 대한 이중 가상 머신 및 신뢰플랫폼 모듈 아키텍쳐

(57) 요 약

네트워크로부터 다운로드 된 또는 미디어 플레이어로부터 장착된 미디어의 안전한 인증을 제공하기 위한 환경에 기초하는 소프트웨어 연산은 2개의 피어 모드(peer-mode) 작동 가상 머신을 포함한다. 로우-레벨 가상 머신은 디코딩 및 해독 기능을 제공하는 반면, 하이-레벨 가상 머신은 사용자 인터페이스, 입력 출력과 같은 애플리케이션 레벨 기능을 제공한다.

대 표 도



(30) 우선권주장
60/481,066 2003년07월07일 미국(US)
60/493,072 2003년08월05일 미국(US)

특허청구의 범위

청구항 1

매체(medium)에 미디어 보안 보호(security protection)를 제공하는 미디어 플레이어에 있어서,

상기 미디어 플레이어에 대해 보안 해독 및 디코딩 기능을 수행하도록 구성되는 보안 가상 머신(security virtual machine);

상기 미디어 플레이어에 대해 애플리케이션 레벨 기능을 수행하고 사용자 인터페이스를 제공하도록 구성되는 애플리케이션 가상 머신(application virtual machine); 및

상기 보안 가상 머신 및 상기 애플리케이션 가상 머신을 구동하도록 구성되는 처리 유닛을 포함하며,

상기 애플리케이션 가상 머신은 하이(high) 레벨 가상 머신이고, 상기 보안 가상 머신은 로우(low) 레벨 가상 머신인 것인, 미디어 플레이어.

청구항 2

제1항에 있어서,

상기 보안 가상 머신은 또한 제1 명령어 세트를 지원하도록 구성되고, 상기 애플리케이션 가상 머신은 또한 제2 명령어 세트를 지원하도록 구성되며, 상기 제1 명령어 세트는 상기 로우 레벨 가상 머신의 것이고, 상기 제2 명령어 세트는 상기 하이 레벨 가상 머신의 것인, 미디어 플레이어.

청구항 3

제1항에 있어서,

애플리케이션 프로그램 인터페이스(application program interface, API)를 더 포함하고, 상기 API는 상기 보안 가상 머신 및 상기 애플리케이션 가상 머신이 상기 API를 통해 서로 통신하는 것을 허용하는 것인, 미디어 플레이어.

청구항 4

제1항에 있어서, 상기 보안 가상 머신은 또한 상기 애플리케이션 가상 머신을 부트스트랩핑(bootstrapping) 하도록 구성되는 것인, 미디어 플레이어.

청구항 5

제1항에 있어서, 상기 매체는 보안 코드들을 포함하고, 상기 애플리케이션 가상 머신은 상기 보안 코드들을 상기 보안 가상 머신에 전달하는 것인, 미디어 플레이어.

청구항 6

제1항에 있어서, 상기 매체는 부트 코드(boot code) 및 콘텐트(content)를 포함하고, 상기 보안 가상 머신은 상기 부트 코드를 해독하도록 구성되는 것인, 미디어 플레이어.

청구항 7

제1항에 있어서, 상기 보안 가상 머신은 동기화 이벤트들과 관련된 정보를 상기 애플리케이션 가상 머신에 전달하도록 구성되는 것인, 미디어 플레이어.

청구항 8

제1항에 있어서, 상기 보안 가상 머신 및 상기 애플리케이션 가상 머신은 피어(peer) 관계를 갖는 것인, 미디어 플레이어.

청구항 9

제1항에 있어서, 상기 보안 가상 머신은 예외 처리 능력(exception handling capability)을 포함하지 않고, 상

기 애플리케이션 가상 머신은 예외 처리 능력을 포함하는 것인, 미디어 플레이어.

청구항 10

제1항에 있어서, 상기 보안 가상 머신은 복사 방지(copy protection) 알고리즘을 포함하는 것인, 미디어 플레이어.

청구항 11

제1항에 있어서, 상기 보안 가상 머신은 상기 애플리케이션 가상 머신보다 낮은 연산 복잡성을 갖는 것인, 미디어 플레이어.

청구항 12

매체(medium)에 미디어 보안 보호(security protection)를 제공하는 방법에 있어서,

처리 유닛 상에서 구동하는 보안 가상 머신을 이용하여, 미디어 플레이어에 대해 보안 해독 및 디코딩 기능을 수행하는 단계; 및

상기 처리 유닛 상에서 구동하는 애플리케이션 가상 머신을 이용하여, 상기 미디어 플레이어에 대해 애플리케이션 레벨 기능을 수행하는 단계를 포함하고,

상기 애플리케이션 가상 머신은 사용자 인터페이스를 제공하고, 상기 애플리케이션 가상 머신은 하이(high) 레벨 가상 머신이고, 상기 보안 가상 머신은 로우(low) 레벨 가상 머신인 것인, 미디어 보안 보호 제공 방법.

청구항 13

제12항에 있어서,

상기 보안 가상 머신을 이용하여 제1 명령어 세트를 지원하는 단계; 및

상기 애플리케이션 가상 머신을 이용하여 제2 명령어 세트를 지원하는 단계를 포함하고,

상기 제1 명령어 세트는 상기 로우 레벨 가상 머신의 것이고, 상기 제2 명령어 세트는 상기 하이 레벨 가상 머신의 것인, 미디어 보안 보호 제공 방법.

청구항 14

제12항에 있어서,

상기 보안 가상 머신 및 상기 애플리케이션 가상 머신이 애플리케이션 프로그램 인터페이스(API)를 통해 서로 통신하는 것을 허용하는 단계를 더 포함하는 미디어 보안 보호 제공 방법.

청구항 15

제12항에 있어서, 상기 애플리케이션 가상 머신은 상기 보안 가상 머신에 의해 부트스트랩핑되는 것인, 미디어 보안 보호 제공 방법.

청구항 16

제12항에 있어서,

상기 매체는 보안 코드들을 포함하고, 상기 방법은 상기 애플리케이션 가상 머신에 의해 상기 보안 코드들을 상기 매체로부터 상기 보안 가상 머신에 전달하는 단계를 더 포함하는 것인, 미디어 보안 보호 제공 방법.

청구항 17

제12항에 있어서, 상기 매체는 부트 코드 및 콘텐트를 포함하고, 상기 방법은 또한 상기 보안 가상 머신을 이용하여 상기 부트 코드를 해독하는 단계를 포함하는 미디어 보안 보호 제공 방법.

청구항 18

제12항에 있어서, 상기 보안 가상 머신에 의해 동기화 이벤트들과 관련된 정보를 상기 애플리케이션 가상 머신

에 전달하는 단계를 더 포함하는 미디어 보안 보호 제공 방법.

청구항 19

제12항에 있어서, 상기 보안 가상 머신 및 상기 애플리케이션 가상 머신은 피어(peer) 관계를 갖는 것인, 미디어 보안 보호 제공 방법.

청구항 20

제12항에 있어서,

상기 보안 가상 머신은 예외 처리 능력을 포함하지 않고, 상기 애플리케이션 가상 머신은 예외 처리 능력을 포함하는 것인, 미디어 보안 보호 제공 방법.

청구항 21

제12항에 있어서, 상기 보안 가상 머신은 복사 방지 알고리즘을 포함하는 것인, 미디어 보안 보호 제공 방법.

청구항 22

제12항에 있어서, 상기 보안 가상 머신은 상기 애플리케이션 가상 머신보다 낮은 연산 복잡성을 갖는 것인, 미디어 보안 보호 제공 방법.

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

제1항에 있어서, 상기 처리 유닛은 처리 유닛 명령어 세트를 갖고, 상기 보안 가상 머신은 또한 제1 명령어 세트를 지원하도록 구성되고, 상기 애플리케이션 가상 머신은 또한 제2 명령어 세트를 지원하도록 구성되며, 상기 제1 명령어 세트 및 상기 처리 유닛 명령어 세트 양자 모두 포인터들을 지원한다는 점에서 상기 제1 명령어 세트는 상기 처리 유닛 명령어 세트와 유사하고, 상기 제2 명령어 세트는 포인터들을 지원하지 않는 것인, 미디어 플레이어.

청구항 32

제12항에 있어서,

상기 보안 가상 머신을 이용하여 제1 명령어 세트를 지원하는 단계; 및

상기 애플리케이션 가상 머신을 이용하여 제2 명령어 세트를 지원하는 단계를 더 포함하고,

상기 처리 유닛은 처리 유닛 명령어 세트를 갖고, 상기 제1 명령어 세트 및 상기 처리 유닛 명령어 세트 양자 모두 포인터들을 지원한다는 점에서 상기 제1 명령어 세트는 상기 처리 유닛 명령어 세트와 유사하고, 상기 제2 명령어 세트는 포인터들을 지원하지 않는 것인, 미디어 보안 보호 제공 방법.

청구항 33

삭제

청구항 34

삭제

청구항 35

제1항에 있어서, 상기 보안 가상 머신은 또한 상기 보안 해독 및 디코딩 기능을 수행하기 위한 보안 명령어 세트를 사용하도록 구성되고, 상기 애플리케이션 가상 머신은 또한 상기 애플리케이션 레벨 기능을 수행하기 위한 애플리케이션 명령어 세트를 사용하도록 구성되며, 상기 보안 해독 및 디코딩 기능을 수행하기 위한 상기 보안 명령어 세트는 상기 애플리케이션 레벨 기능을 수행하기 위한 애플리케이션 명령어 세트와 상이한 것인, 미디어 플레이어.

청구항 36

제12항에 있어서, 상기 보안 가상 머신은 또한 상기 보안 해독 및 디코딩 기능을 수행하기 위한 보안 명령어 세트를 사용하도록 구성되고, 상기 애플리케이션 가상 머신은 또한 상기 애플리케이션 레벨 기능을 수행하기 위한 애플리케이션 명령어 세트를 사용하도록 구성되며, 상기 보안 해독 및 디코딩 기능을 수행하기 위한 상기 보안 명령어 세트는 상기 애플리케이션 레벨 기능을 수행하기 위한 애플리케이션 명령어 세트와 상이한 것인, 미디어 보안 보호 제공 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 제거 가능한 미디어 플레이어에 대한 복사 금지(Copy Protection)를 포함하는 보안의 새로운 시스템 및 방법 개발에 관한 것이다.

배 경 기 술

[0002] 가상 머신(VM)은 프로그램의 명령을 실제적으로 행하는 컴파일러 코드 및 마이크로프로세서(또는 "하드웨어 플랫폼(hardware platform)") 사이에서 인터페이스로서 실행되는 소프트웨어를 설명하기 위해 사용된 용어이다. 컴파일러는 특정 프로그래밍 언어로 쓰여진 명령문을 처리하고 그들을 컴퓨터의 프로세서가 사용하는 2진 기계어 또는 "코드(Code)"로 변환시키는 특정 프로그램이다.

[0003] 자바 프로그래밍 언어 및 런타임 환경의 개발자인 썬 마이크로시스템은 자바 가상 머신의 썬 마이크로시스템의 개발로 잘 알려져 있다. 자바 가상 머신은 컴퓨터의 프로세서(또는 "하드웨어 플랫폼")에 대한 컴파일된 자바 이진 코드(바이트 코드로 불림)를 해석하여 그것이 자바 프로그램 명령을 행할 수 있도록 한다.

[0004] 자바는 프로그래머에 의해 각각의 분리된 플랫폼에 대하여 재기록되거나 재컴파일되도록 하지 않고 임의의 플랫폼상에서 구동될 수 있는 애플리케이션 프로그램이 구축되도록 설계되었다. 일단 자바 가상 머신이 플랫폼에 대해 제공되고, 임의의 자바 프로그램이 그 플랫폼상에서 구동될 수 있다. 자바 가상 머신은 특정 명령 구간 및 플랫폼의 다른 특수성을 깨닫고 있으므로, 자바 가상 머신은 이것을 가능하게 한다.

[0005] 가상 머신은 이상적인 연산 머신이다. 실제 연산 머신과 같이, 가상 머신은 명령어 세트를 갖고, 구동시 여러가

지 메모리 영역을 다룬다. 가상 머신을 이용하여 프로그래밍 언어를 구현하는 것은 통상적인 것이다; 가장 잘 알려진 가상 머신은 UCSD 파스칼의 P-코드가 될 수 있다.

[0006] 다른 점에서 가상 머신은 컴퓨터를 구동시키는 작동 시스템 또는 임의의 프로그램 중 하나를 더 일반적으로 설명한다.

[0007] DVD 또는 CD 플레이어와 같은, 차세대 미디어 플레이어에는 복사 금지의 개선된 방법을 개발하기 위해 긴 펠트(felt)가 필요하다.

[0008] DVD에 사용된 하나의 공지된 콘텐츠 보안 시스템은 DVD 상에 데이터가 인코딩하는 콘텐츠 스크램블 시스템(CSS)이다. 그 후, DVD 플레이어는 DVD 플레이어가 40 비트 해독 키를 사용하여 디스크를 판독할 때, 데이터를 해독한다. 그러나, CSS의 치명적인 결함은 그것의 키와 알고리즘이 고정되도록 하는 것을 입증했다. 암호화 알고리즘은 역으로 처리되고, 기존의 DVD 디스크를 재생하는 모든 가능 복호 키는 이용가능하게 만들어진다. 일단 비밀이 탄로나면, 보안 알고리즘 또는 키를 갱신하기 위한 방법이 없기 때문에 시스템은 영원히 구동되지 않는다. 지금, 한번의 "클릭(click)"으로 DVD 콘텐츠로부터 모든 보안을 제거하는, 소비자가 이용가능한 많은 프로그램이 있다.

[0009] 콘텐츠 소유자는 특히, 콘텐츠 복제가 증가하기 때문에, 이것이 다시 일어나는 것을 원하지 않는다. 따라서, 다음 콘텐츠 보안 시스템은 이 방법에 있어서 공격받기 쉽지 않다.

[0010] 소프트웨어 벤더는 또한 주어진 컴퓨터의 본질이지만, 도용의 문제에 직면해 있고, 그들은 DVD에 대한 오락 산업에 사용되는 것과는 상이한 접근을 취한다. 역사적으로, 패키지 소프트웨어 프로그램(즉, 컴퓨터 게임) 제작자는 그들의 콘텐츠를 "절차형 보안(procedural security)"으로 보호한다. 즉, 보안 프로그램용 미리 정의된 고정적인 방법은 없고, 대신에 각 소프트웨어 생산자는 그들의 콘텐츠의 보안을 위해서 "보안 코드(security code)"를 적거나 획득한다. 이러한 절차형 보안 코드는 프로그램 기초에 의해 프로그램상의 복잡성과 기술을 변화시키고, 그러나 가장 중요하게, 그러한 프로그램은 상이한 보안 소프트웨어 구현을 포함하기 때문에, DVD 보안을 회피하도록 기록된 것과 같이, 일반적인 목적으로 "보안 제거(remove security)" 프로그램을 기록하는 것은 가능하지 않다.

[0011] 복사 금지의 공지된 다른 방법은 하드웨어 특정 명령을 기록하는 것이다. 그러한 방법의 문제점은 이것이 극히 제한적이라는 것이다. 이러한 방법에 있어서, 상이한 세트의 명령은 각각의 하드웨어 구성에 대하여 명령을 내려야만 한다. 이것은 다소 비실용적이다.

[0012] 따라서, 특정한 하드웨어가 아닌, 미디어 플레이어와 같은 하드웨어에 복사 금지를 제공하는 방법이 요구된다.

발명의 상세한 설명

[0013] 플랫폼 독립의 절차형 복사 금지의 시스템 및 방법은 미디어 플레이어에 제공된다. 본 명세서는 이중 가상 머신 아키텍처가 차세대 미디어 플계층에 제공됨으로써 솔루션을 제안한다. 본 명세서는 가상 머신 아키텍처의 특정 양태와 인터페이스하기 위한 신뢰 플랫폼 모듈(Trusted Platform Module:TPM)과 같은 하드웨어-기반 임베디드 보안 서브시스템의 이용을 더 제안한다.

[0014] 본 명세서에 따른 이중 가상 머신 아키텍처는 하이-레벨 가상 머신과 로우-레벨 가상 머신으로 구성된다. 로우-레벨 가상 머신은 로우-레벨 미디어 해독 및 디코딩 기능을 지원하도록 설계되어있고, 하이-레벨 가상 머신은 애플리케이션 계층 활동을 취급하도록 설계되어 있다. 따라서 아키텍처는 애플리케이션 소프트웨어로부터 보안 소프트웨어를 분할한다.

[0015] 일반적으로, 절차형 보안에 가장 적합한 가상 머신은 실제의 하드웨어 CPU의 명령 세트과 매우 유사하다. 즉, 가상 머신은 포인터를 지원하고, 실행 가능한 코드 및 데이터 사이에서 근본적으로 구별하지 않는다. 따라서, 이 제1 유형의 가상 머신은 "로우-레벨 VM" 또는 "보안 VM"으로 명명된다. 로우-레벨 가상 머신은 변형 억제 소프트웨어 기술을 지원하는 종래 CPU와 공통점을 갖도록 설계되었다.

[0016] 이와 같은 가상 머신의 단점은, 프로그래밍 에러 또는 예측하지 못한 런타임 조건은 치명적인 경향이 있다는 것이다. 보안 시스템에 있어서, 이것은 이점으로 고려될 수 있지만, 애플리케이션에 대해서는 (훨씬 더 복잡하고, 통상적으로 덜 집중적인(intensive) 테스트 커버리지를 가지는) 이것은 단점이다.

[0017] 애플리케이션에 대해서, 보다 많은 연산 디테일인 "비하인드-더-신즈(behind-the-scenes)"를 관리하는 "하이-레벨 VM"은 더욱 예측가능하고 확고한(robust) 방식으로 동작하는 보다 의존가능한 애플리케이션 프로그램이 개발

되도록 허용한다. "하이-레벨 가상 머신"의 대표적인 예는 자바(Java)이다. 예를 들어, 자바는 "포인터"의 개념 또는 명백한 메모리 관리(프로그래밍 에러의 공통 소스인)의 지원을 할 수 없지만, 프로그램 및 프로그래머가 예측할 수 있는 방법으로 예측할 수 없는 런타임 환경을 취급하는 것을 돕는 "예외 처리(exception handling)"를 지원한다.

[0018] 하이-레벨 또는 애플리케이션 레벨 가상 머신은 성능을 잘 살리도록(full featured) 설계되고, 리치 애플리케이션 인터페이스용으로 제공된다.

[0019] 따라서, 다른 적용과 결합해 작동하는 플랫폼 독립의 보안 기능을 제공하기 위해서, 로우-레벨 VM 및 하이-레벨의 VM의 양쪽 모두의 이점을 조합하는 것이 이상적이다. 또한, 신뢰 플랫폼 모듈은 실행 환경을 안전하게 질문하여 검증함으로써 신뢰있는 하드웨어 기반 루트를 제공한다.

[0020] 예시적인 구성예에서, 본 명세서는 DVD 및 CD 플레이어에서 갱신 가능한 보안 및 복사 금지의 목적으로 사용된다. 그러나, 이러한 아키텍처는 또한 하드 드라이브, 고체 메모리 또는 네트워크에 의하여 전달되는 것 상에 저장된 미디어의 재생을 지원한다.

[0021] 전술한 바와 같이, 로우-레벨 가상 머신은 로우-레벨 미디어 해독 및 디코딩 기능을 지원하도록 설계되었다. 차세대 미디어(NGM) 애플리케이션에 있어서, 로우-레벨 가상 머신은 하이-레벨 VM의 부트스트래핑에 대한 책임이 있을 수도 있다. 하이-레벨 VM은 진보된 사용자 인터페이스,기타(misc.) IO 및 네트워크 활동과 같은 애플리케이션 계층 활동을 처리한다.

[0022] 본 명세서의 이중 가상 머신 아키텍처는 신규한 것이다. 이중의 VM 아키텍처는 종래의 "스택 VM" 관계와 달리 "피어" 관계를 제공한다. 다른 것 위에서 구동하는 하나의 VM의 스택 관계의 일 예는, 윈도우 에뮬레이터(x86 에뮬레이터 또는 VM)를 구동시키는 PowerPC(Mac에서와 같은)일 것이며, 차례로 Java VM을 실행한다.

[0023] 게다가, 하드웨어 기반 신뢰 연산 모듈을 가지는 이러한 이중 VM 아키텍처의 결합은 신규한 것이다.

[0024] 본 명세서는 절차형 보안의 사용을 CD들과 DVD들과 같은 미디어에까지 연장한다. 게다가, 절차형 보안은 콘텐츠 소유자에게 선언적인 시스템보다 훨씬 더 유연한 권리를 관리해준다. 이 유연성은 CSS와 같은 종래의 정적인 보안 시스템에 의해 제공되는 단순한 복사 방지(Copy Protection:CP)에 반한 것으로서, 충분히 사용가능한 디지탈 권리 관리(DRM:Digital Rights Management) 시스템을 구현하기 위해서 사용될 수 있다.

[0025] 본 명세서의 상기 및 다른 목적, 특징 및 이점은 명세서의 특징 및 이점을 예시한 예시적인 실시예의 다음의 상세한 설명을 읽음으로써 명백해질 것이다.

실 시 예

[0029] 상세한 설명을 후술한다; 그러나, 개시된 실시예는 단지 본 발명의 일례이고, 다양한 형태로 구현될 수도 있다는 것이 이해되어야 한다. 따라서, 여기에 개시된 특정 구조 및 기능의 상세 설명은 제한적으로서가 아니라, 당업자가 가상적으로 임의의 적절하게 상세화된 구조의 본 개시를 다양하게 채용하도록 알려주기 위하여 단지 청구항의 기초로서 그리고 대표적인 기초로서 해석되어야 한다. 명세서는 첨부된 도면(도 1-2)에서 예시를 참조하여 상세하게 설명할 것이다.

[0030] 본 명세서의 시스템 및 방법은 미디어 플레이어 사용을 위한 이중 가상 머신아키텍처(dual virtual machine architecture)를 제공한다. 하나의 VM은 미디어 해독 및 디코딩과 같은 보안 기능을 지원하도록 설계되었다. 차세대 미디어 애플리케이션에 있어서, 로우-레벨 VM은 애플리케이션 레벨 VM의 부트스트랩핑에 대한 책임이 있을 수 있다. 하이-레벨 또는 애플리케이션 레벨 VM은 진보된 사용자 인터페이스, 기타(misc.) IO 및 네트워크 활동과 같은, 애플리케이션 계층 활동을 취급한다.

[0031] 도 1 및 도 2는 예시적인 실시예에 따른 컴퓨터 환경(10) 내에서 미디어 플레이어 아키텍처를 도시한다. 특히, 미디어 데이터 또는 콘텐츠(18), 미디어가 미디어 플레이어 상에서 재생되도록 하는 보안 코드(12), 및 부트 코드(16)를 포함하는 미디어 소스(예를 들어, DVD, 광 디스크, 고체 장치, 또는 네트워크)를 도시하였다.

[0032] 본 개시에 따른 미디어 재생 장치는 적어도 하나의 가상 머신(VM)을 구동 가능한 중앙 처리 장치(26)를 포함한다. 예시적인 실시예에서, 가상 머신은 CPU (26) 상에서 구동하는 로우-레벨 VM(예를 들어, 보안 VM)(22) 및 하이-레벨 VM(예를 들어, 애플리케이션 VM)(24)를 포함하는 이중 가상 머신 아키텍처이다. VM 내에서 구동되는 프로그램은 갱신 암호 알고리즘 뿐만 아니라 사용 규칙을 실행하고 시행할 수도 있다. 연산 환경(10)은 다양한 프로그램이 서로 통신하도록 하는 한 세트의 루틴 또는 프로토콜인 애플리케이션 프로그램 인터페이스(API)(40-

44)를 또한 포함할 수도 있다.

[0033] 일 양태에서, VM (22 또는 24) 중 임의의 하나는 다른 VM을 제어할 수도 있다. 다른 양태에서, 하이-레벨 및 로우-레벨 가상 머신은, 비 계층적 방식으로, 그들 간의 메시지를 전달하는 피어로서 기능한다. 이들 메시지들은, 하나의 가상 머신이 다른 가상 머신에서의 루틴을 호출하는 "외부 기능 호출(foreign-function calls)"로서, 또는 통신 채널을 따라 전달되는 일반적인 메시지로서 구현될 수도 있다.

[0034] 예를 들어, 애플리케이션 VM(또는 하이-레벨 VM)(24)는 미디어 콘텐츠(18)의 재생(및 따라서 투과성 디코딩)을 시작하기 위해서 보안-VM (또는 로우-레벨 VM)(22)을 호출할 것이다.

[0035] 마찬가지로, 보안 VM(22)에서의 코드는 애플리케이션(VM)(24)을 호출하여, 애플리케이션이 동기화 이벤트 또는 디코딩 문제(예를 들어 보안 또는 허용 문제)에 관해 알게 하도록 한다.

[0036] 예를 들어, 미디어가 자신의 본래 광 미디어의 복사가 된 상황에서, 보안 VM(22)은 애플리케이션 VM(24)에게 계속 재생하기 위해서 키를 필요로 한다는 것을 알려줄 것이다. 이에 응답하여, 애플리케이션 VM(24)는 애플리케이션 레벨 함수(25)를 통해, 사용자에게 그들이 사용자 인터페이스(27)를 통해 특정 기간 동안 이 영화를 "임대(rent)"할 수도 있다는 것을 통지하는 메시지를 표시할 것이다. 사용자가 이것을 하도록 선택하는 경우, 사용자는 스튜디오 서버와의 트랜잭션을 체결하여, 키를 포함하는 "불투명한 메시지(opaque message)" (VM에 의해서만 이해가능한)를 획득해야 한다. 다음 이 애플리케이션 VM(24)은 인증하기 위하여 키를 포함하는 메시지를 다시 보안 VM(24) 및 복사 방지 알고리즘(23)으로 전달한다.

[0037] 미디어 재생 장치는 프로세싱 모듈(예를 들어, 신뢰 프로세싱 모듈 또는 TPM)(32)을 더 포함한다. TPM 사양은 신뢰 연산 그룹 (TCG) (htttp://www. trustedcomputinggroup. org)에 의해 창립된 신뢰 연산 플랫폼 동맹(TCPA) 사양의 일부이다. TPM(32)은 해독 키를 포함하고 보안 암호 연산을 처리한다. 미디어 재생 장치는 가상 머신에서 구동되는 임의의 프로그램이 장치의 I/O 하드웨어 및 TPM에 질의하도록 허락하는 API(40,42)를 더 포함한다. 이것은 VM 내에서 사용되는 프로그램이 사용 규칙에 대한 우수한 선택을 하도록 한다. CPU(26)에 부착된 디코딩 모듈(34)은 인코딩된 오디오/비디오 스트림을 언패킹하기 위해 더 제공된다.

[0038] 일반적으로, 신뢰 플랫폼은 엔티티가 소프트웨어의 상태 또는 플랫폼에서의 연산 환경(10)을 결정하고, 데이터를 그 플랫폼에서의 특정 소프트웨어 환경으로 제공(seal)하는 것을 가능하게 한다. 그 엔티티는 연산 환경의 상태가 수용가능한지 여부를 추론하고, 그 플랫폼과의 트랜잭션을 수행한다. 트랜잭션이, 플랫폼에 저장되어야 하는 민감한 데이터를 포함하는 경우, 그 플랫폼에서 연산 환경의 상태가 엔티티를 수용할 수 없는 경우, 엔티티는 그 데이터가 신뢰할만한 포맷으로 유지되는 것을 보장할 수 있다.

[0039] 이것을 가능하게 하기 위해서, 신뢰 플랫폼은 엔티티가 신뢰 플랫폼에서 소프트웨어 환경을 추론하도록 하는 정보를 제공한다. 그 정보는 신뢰성 있게 측정되어 엔티티로 보고된다. 동시에, 신뢰 플랫폼은 복호 키를 암호화하여 키를 해독할 수 있기 전에 적소에 있을 소프트웨어 환경을 제시하는 수단을 제공한다.

[0040] "신뢰 측정 루트(trusted measurement root)"는 특정 플랫폼 특성을 측정하고, 측정 기억 장치 내의 측정 데이터를 기록하고, TPM(보전 메트릭스를 저장하여 보고하기 위한 신뢰 루트를 포함) 내의 최종 결과를 저장한다. 따라서 TPM은 모든 복호 키에 대한 안전한 저장 장소이다. TPM은 또한 가장 암호적인 연산 및 기능을 처리한다.

[0041] 게다가, 미디어 재생 장치는 안전하고, 보호된 입력 및 출력(28), 다른 플레이어와 네트워크하는 능력(30), 메모리 장치(예를 들어 , RAM (36) 및 ROM (38))를 더 포함한다.

[0042] 따라서, 예시적인 실시예에 따르면, 개별 가상 머신(VM)은 CPU를 포함하는 동일 연산 환경에서 구동된다. 본 아키텍처는 2개의 가상 매니저(즉, 하이-레벨 또는 애플리케이션 VM 및 로우-레벨 또는 보안 VM)로 분할되는데, 이 애플리케이션 및 보안 가상 매니저는 표준화된 API를 통해 통신한다. 애플리케이션 가상 매니저의 기능은, 보안 가상 매니저에서 실행되어지는 보안 코드에 네트워크 서비스를 제공하는 단계를 포함하고, 미디어 액세스 및 디코딩 기능은, 콘텐츠 보안이 애플리케이션 저작자에게 투명하도록 보안 VM에 의해 조정된다.

[0043] 연산 복잡성에 관해서는, 보안 VM은 시스템 리소스상의 낮은 임팩트를 갖고, 간단하고, 경량, 로우-레벨, 및 안전하며, 이 VM에 적당한 소프트웨어는 보안 벤더에 의해 제공될 수도 있다. 애플리케이션 VM은 비교적 더 큰 CPU 및 메모리 임팩트를 포함하고, 사용자 인터페이스 및 입력/출력 기능에 대한 책임이 있다.

[0044] 본 개시는 통상의 CD 및 DVD 플레이어와 같은 미디어 플레이어에서의 사용에 한정되는 것이 아니고, PC 또는 제거가능한 미디어 재생시키는 기능을 포함하는 좀더 일반화된 하드웨어 시스템에서의 구동으로 확장시킬 수

있다.

[0045] 본 개시의 바람직한 실시예의 상기 설명은 예시 및 설명의 목적으로 제시된 것이다. 본 개시의 다른 목적, 기능 및 이득은 후술한 부가물을 읽음으로써 명확해질 수 있다. 본 개시를 개시된 특정 형태로 제한하거나 철저하게 규명하고자 함이 아니다. 상기 교시의 관점에서 다양한 수정 및 변형이 가능하다.

도면의 간단한 설명

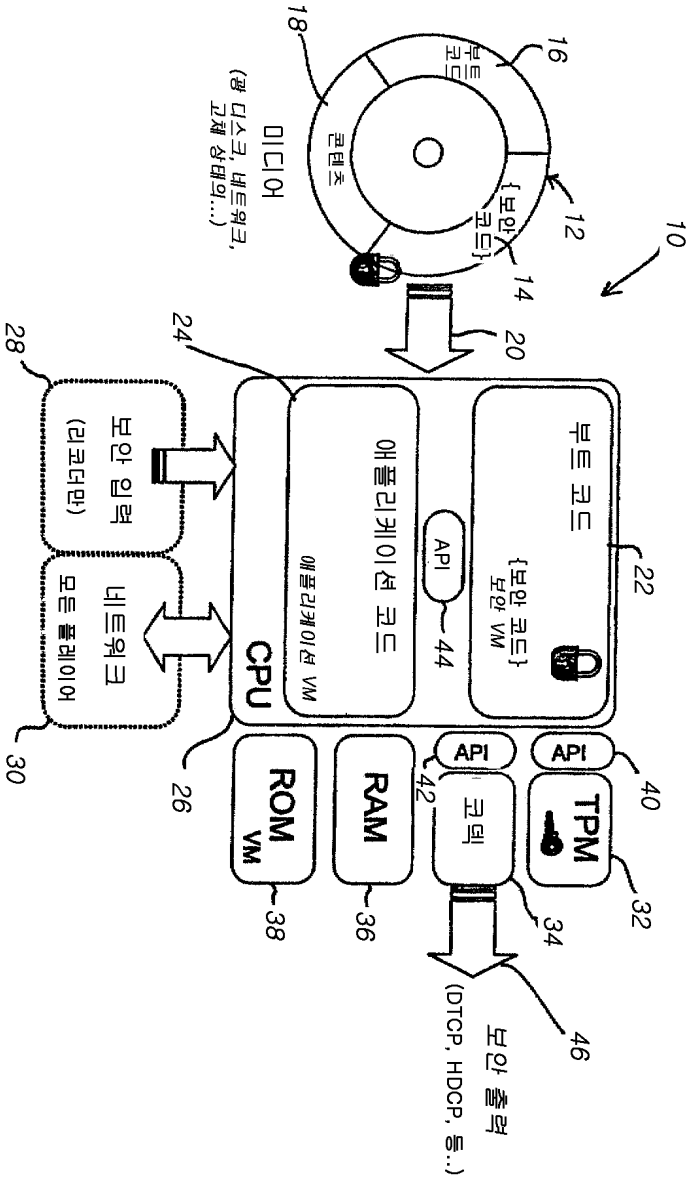
[0026] 도 1은 예시적인 실시예에 따른 연산 환경에서의 미디어 플레이어 아키텍처의 다이어그램이다.

[0027] 도 2는 예시적인 구성예에 따른 로우-레벨 가상 매니저 및 하이-레벨 가상 매니저의 상호 작용 및 기능을 묘사하는 블럭도이다.

[0028] 도시의 단순성 및 명료성을 위하여, 도면에 도시된 요소들은 필수적으로 일정한 비례로 도시되지 않았다는 것이 이해되어야 한다. 예를 들면, 소자 중 몇개의 치수는 명료성을 위해서 서로에 관하여 확대되었다. 게다가 적절하다고 고려되는 경우, 참조 번호는 대응하는 소자를 나타내기 위해서 도면 사이에 반복되었다.

도면

도면1



도면2