(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년03월25일
(11) 등록번호 10-1247044
(24) 등록일자 2013년03월18일

(51) 국제특허분류(Int. Cl.)
G06F 11/30 (2006.01) *G06F 12/14* (2006.01)
H04L 9/32 (2006.01)
(21) 출원번호 10-2007-7024156
(22) 출원일자(국제) 2006년03월22일
심사청구일자 2011년03월02일
(85) 번역문제출일자 2007년10월19일
(65) 공개번호 10-2007-0122502
(43) 공개일자 2007년12월31일
(86) 국제출원번호 PCT/US2006/010327
(87) 국제공개번호 WO 2006/115639
국제공개일자 2006년11월02일
(30) 우선권주장
11/202,840 2005년08월12일 미국(US)
60/673,979 2005년04월22일 미국(US)
(56) 선행기술조사문헌
US20041022589 A1
US5943248 A
전체 청구항 수 : 총 20 항

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
마쉬, 데이비드 제이.
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
르네리스, 케네스
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
(뒷면에 계속)
(74) 대리인
제일특허법인

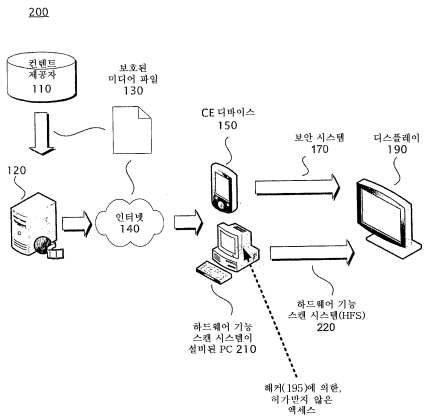
심사관 : 이정은

(54) 발명의 명칭 **디바이스 인증을 위한 하드웨어 기능 스캔**

(57) 요 약

하드웨어 기능 스캔을 수행함으로써 그래픽 칩 또는 기타 하드웨어 칩 또는 하드웨어 디바이스의 인증을 검증하는 시스템 및 방법이 제공된다.

대 표 도 - 도2



(72) 발명자

블라이스, 데이비드 알.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

데비퀘, 커트 에이.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

특허청구의 범위

청구항 1

컴퓨터 실행가능 명령어들을 기록한 컴퓨터 판독가능 기록 매체로서,

상기 컴퓨터 실행가능 명령어들은, 실행될 때, 하나 이상의 프로세서로 하여금,

하드웨어 디바이스에게 동작 세트(a set of operations)를 수행할 것을 요청하고 - 상기 하드웨어 디바이스는 그래픽 디바이스이고, 상기 동작 세트 중 적어도 하나의 동작은 상기 그래픽 디바이스를 모방하는 에뮬레이션에 의해서는 구현될 수 없는 그래픽 처리(graphics calculation)이고, 상기 그래픽 디바이스는 상기 그래픽 처리의 결과에 기초하여 상기 에뮬레이션(emulation)과 구별가능함 -,

상기 그래픽 처리의 결과에 기초하여 상기 동작 세트의 수행 결과들이 인증 디바이스의 동작 결과(behavior)와 일치(consistent)하는지를 검증하고,

상기 결과들이 일치함을 검증하는 것에 응답하여, 보안 액세스가 설정되도록 허가하는 컴퓨터 판독가능 기록 매체.

청구항 2

제1항에 있어서,

상기 인증 디바이스의 동작 결과는 미리 계산되는 컴퓨터 판독가능 기록 매체.

청구항 3

제1항에 있어서,

상기 인증 디바이스의 동작 결과는 검증 시에 계산되는 컴퓨터 판독가능 기록 매체.

청구항 4

제1항에 있어서,

상기 동작 세트는 상기 하드웨어 디바이스의 복수의 별개의 부분에 의해 수행되는 컴퓨터 판독가능 기록 매체.

청구항 5

제1항에 있어서,

상기 동작 세트는 기능들의 집합(a collection of functions)으로부터 무작위로 선택되는 컴퓨터 판독가능 기록 매체.

청구항 6

제1항에 있어서,

상기 동작 세트에 대한 입력은 무작위로 선택되는 컴퓨터 판독가능 기록 매체.

청구항 7

제1항에 있어서,

상기 검증은 적어도 상기 하드웨어 디바이스의 일부분이 인증되었음을 가리키는 컴퓨터 판독가능 기록 매체.

청구항 8

제1항에 있어서,

상기 하드웨어 디바이스의 일부분에 의해 수행되는 상기 동작 세트는 상기 디바이스 제조에 전용되는(proprietary) 컴퓨터 판독가능 기록 매체.

청구항 9

제1항에 있어서,

상기 동작 세트는 무효화(revoke), 및 교체된 또는 확장된 동작 세트로 갱신될 수 있는 컴퓨터 판독가능 기록 매체.

청구항 10

디바이스를 인증하는 방법으로서,

하드웨어 디바이스에게 동작을 수행할 것을 요청하는 단계 - 상기 하드웨어 디바이스는 그래픽 디바이스이고, 상기 동작은 상기 그래픽 디바이스를 모방하는 에뮬레이션에 의해서는 구현될 수 없는 그래픽 처리를 포함하고, 상기 그래픽 디바이스는 상기 그래픽 처리의 결과에 기초하여 상기 에뮬레이션과 구별가능함 -;

상기 그래픽 처리의 결과에 기초하여 상기 동작의 수행 결과가 인증 디바이스의 동작 결과와 일치하는지를 검증하는 단계 - 상기 검증하는 단계는 적어도 하나의 값을 선택하고, 상기 값을 파라미터로서 상기 동작에 넘기고, 또한 상기 값을 파라미터로서 상기 동작의 소프트웨어 에뮬레이션에 넘김으로써 수행됨 -; 및

상기 결과들이 일치함을 검증하는 것에 응답하여, 보안 액세스가 설정되도록 허가하는 단계

를 포함하는, 디바이스를 인증하는 방법.

청구항 11

제10항에 있어서,

상기 동작은 복수의 집적 회로 칩 상에서 수행되는, 디바이스를 인증하는 방법.

청구항 12

제10항에 있어서,

상기 동작은 하나의 집적 회로 칩 상에서 수행되는, 디바이스를 인증하는 방법.

청구항 13

제10항에 있어서,

상기 적어도 하나의 값은 무작위로 선택되는, 디바이스를 인증하는 방법.

청구항 14

제10항에 있어서,

상기 동작은 동작 세트로부터 무작위로 선택되는, 디바이스를 인증하는 방법.

청구항 15

제10항에 있어서,

공지된 결과는 테이블 내에 저장되는, 디바이스를 인증하는 방법.

청구항 16

제10항에 있어서,

상기 동작은 디바이스 제조에 전용되는, 디바이스를 인증하는 방법.

청구항 17

디바이스를 인증하기 위한 시스템으로서,

보호 비디오 스트림(protected video stream)을 디스플레이하기 위한 디스플레이어;

상기 디스플레이어에의 액세스를 제공하기 위한 드라이버 - 상기 드라이버는, 상기 디스플레이어를 인증하기 위한 인증자(authenticator)를 더 포함하고, 상기 인증자는,

상기 디스플레이어에게 동작 세트를 수행할 것을 요청하고, 상기 디스플레이어는 그래픽 디바이스를 포함하고, 상기 동작 세트 중 적어도 하나의 동작은 상기 그래픽 디바이스를 모방하는 에뮬레이션에 의해서는 구현될 수 없는 그래픽 처리이고, 상기 그래픽 디바이스는 상기 그래픽 처리의 결과에 기초하여 상기 에뮬레이션과 구별가능하고,

상기 그래픽 처리의 결과에 기초하여 상기 동작 세트의 결과들이 인증 디바이스의 동작 결과와 일치하는지를 검증하고,

상기 결과들이 일치함을 검증하는 것에 응답하여, 보안 액세스를 설정하는 것을 허가하도록 구성됨 -; 및

상기 드라이버를 실행하기 위한 환경이 신뢰됨을 검증하는 운영 체제

를 포함하는 디바이스를 인증하기 위한 시스템.

청구항 18

제17항에 있어서,

상기 인증자는 에뮬레이터(emulator)를 포함하는, 디바이스를 인증하기 위한 시스템.

청구항 19

제17항에 있어서,

상기 인증자는 룩업 테이블(lookup table)을 포함하는, 디바이스를 인증하기 위한 시스템.

청구항 20

제17항에 있어서,

상기 드라이버는 인증서로 서명되는, 디바이스를 인증하기 위한 시스템.

명 세 서

기 술 분 야

[0001] 본 출원은 2005년 4월 22일에 출원된 미국 가특허 출원 60/673,979의 권리를 주장한 것으로, 그 내용은 본원에 참조로 포함된다.

배 경 기 술

[0002] 본 개시물은 일반적으로 컴퓨터 보안에 관한 것이며 더욱 자세히는 검증 방법에 관한 것이다. 이러한 시스템은 여러 인터페이스에 의해 연결될 수 있는 임의의 개수의 컴포넌트를 포함할 수 있다. 이러한 시스템에서 보호 컨텐트의 소유자는 통상적으로 컨텐트를 전송하기 전에 충분한 보안이 이루어지는 지를 확인한다. 신뢰의 고리(chain of trust)는 이러한 시스템 내에 보안을 구축하기 위해 사용될 수 있다. 이들 시스템의 사용에 있어, 중요한 컨텐트의 전송이 증가하고, 허가받지 않은 사용자가 보호 컨텐트로의 액세스를 얻는 데 있어 더욱 정교해지고 있다는 사실 때문에 보안의 증진은 더 큰 관심사가 되고 있다.

[0003] 고급 컨텐트 또는 정보의 제공자는 PC와 같은 종래의 개방형 컴퓨팅 시스템이 안전함을 보장받기 원할 수 있다. PC 및 많은 프로세서 기반 시스템이 일반적으로, 하드웨어 컴포넌트가 쉽게 이동될 수 있고 대체될 수 있는 개방형 시스템을 나타낸다. 이러한 개방형 시스템은 컨텐트로의 허가받지 않은 액세스에 대한 다중 액세스 포인트를 나타낼 수 있다.

실 시 예

[0012] 첨부된 도면과 연관되어 다음에 제공되는 상세한 서술은 본 예의 서술로 쓰이기 위함일 뿐이며 본 예가 구성되거나 활용될 수 있는 유일한 형태를 나타내기 위함은 아니다. 본 서술은 그 예의 기능과 그 예를 구성하고 작동시키기 위한 일련의 단계를 제시한다. 그러나, 동일하거나 또는 동등한 기능 및 순서가 서로다른 예를 통해 달성될 수 있다.

[0013] 비록 본 예가 PC 기반의 시스템에 구현되고 있는 것으로 본원에서 기술되고 묘사되지만, 기술된 시스템은 제한

이 아니라 예시로서 제공된다. 당업자는 본 예가 다양한 여러 타입의 컴퓨팅 시스템에서의 응용에 적합함을 인식할 것이다.

[0014] 도 1은 하드웨어 기능 스캔("HFS") 시스템을 갖지 않는(180) 종래의 보호된 미디어 파일(130)을 재생하는데 이용될 수 있는, 종래의 PC(160) 또는 CE 디바이스(150)를 도시하는 블록도이다. 하드웨어 기능 스캔 시스템이 없는 이러한 PC(160)는 보호된 미디어 파일(130)이 해커나 기타 허가받지 않은 자에 의해 쉽게 인터셉트되도록 허용한다. 컨텐트 제공자(110)는 일반적으로 미디어 서버(120)에 연결된다. 컨텐트 제공자(110)는 일반적으로 미디어 서버(120)에 보호된 미디어 파일(130)을 배치한다. 보호된 미디어 파일(130)이 미디어 서버에서, 서비스 제공자에 의해 제공된 컨텐트로부터 생성되거나, 서비스 제공자가 보호된 미디어 파일(130)을 미디어 서버(120)에 제공할 수 있다. 보호된 미디어 파일(130)은 일반적으로 오디오 및 비주얼 정보 등을 포함한다. 미디어 서버(120)는 일반적으로 인터넷(140)에 연결되며, 인터넷(140)은 PC(160) 또는 CE 디바이스(150)에 일반적으로 연결된다. PC(160) 또는 CE 디바이스(150)는 프로세서가 설비된 디바이스의 단지 두 예일 뿐이다. 여러 가지 장치가 PC(160) 또는 CE 디바이스(150)를 대신하여 동등하게 사용될 수 있음이 명확하다. 다음의 서술에서 PC라는 용어가 CE 디바이스, 프로세서 보드 디바이스 등을 포함할 수도 있음이 이해될 것이다. CE 디바이스(150)는 이러한 디바이스의 고정된 구성 때문에 일반적으로 쉽게 훼손되지 않는다. 반대로, PC(160)은 쉽게 액세스 될 수 있는 개방형 시스템이다.

[0015] PC(160)는 일반적으로 종래 보안 시스템(170)의 일부이며, 보안 시스템(170)은 일반적으로 컨텐트 제공자(110)에게 해커에 의한 허가받지 않은 액세스가 일어나지 않을 것이라고 안심시킬 수 있는 보호 방법 및 PC 컴포넌트를 포함한다.

[0016] 종래의 보안 시스템(170)은 일반적으로 이미지 정보를 보일 수 있게 렌더링하는 디스플레이(190) 및 CPU를 포함할 수 있다. 종래의 PC 시스템에서는, PC(160)가 외부 디스플레이 또는 모니터(190)에 연결된다. 그래픽 중심의 시스템은 표시되는 객체를 렌더링하는 것을 돕기 위해 종래의 그래픽 프로세서를 이용할 수 있다. CPU 내의 프로세서와 그래픽 디바이스의 프로세서 간의 접속은 그 시점에서 해커(195)에 의한, 허가받지 않은 액세스를 허용할 수 있다. 이러한 "보안 시스템"(170)은 디스플레이(190)에서 보호된 미디어 파일(130)을 재생하는 것을 허용할 수 있다. 일반적으로 그래픽 디바이스(175)의 프로세서에 제공되는 컨텐트는 암호화되지 않는다.

[0017] 도 2는 하드웨어 기능 스캔 시스템(220)을 구비하는 PC(210)를 도시하는 블록도이다. 컨텐트 제공자(110)는 일반적으로 미디어 서버(120)에 연결되어 있다. 컨텐트 제공자(110)는 일반적으로 미디어 서버(120)에 보호된 미디어 파일(130)을 배치하며, 보호된 미디어 파일(130)은 일반적으로 오디오 및 비주얼 정보 등을 포함한다. 미디어 서버(120)는 보통 인터넷(140)에 연결되며, 인터넷(140)은 일반적으로 PC(210)에 연결된다.

[0018] 보안 시스템(270) 내의 PC(210)는 일반적으로 이미지 정보를 보일 수 있게 렌더링하는 디스플레이(190)에 연결될 수 있다. PC(210) 및 그것의 보안 시스템(270)은 하드웨어 기능 스캔 시스템(220)을 포함한다. 하드웨어 기능 스캔("HFS") 시스템은, 해커 또는 기타 허가받지 않은 자(195)가 보호된 미디어 파일(130)의 보호되지 않은 버전을 취약한 포인트(340)에서 액세스하지 않음을 보장하기 위해 컨텐트 제공자(110)에 의해 요청된 보안 허가를 더 확인할 수 있다. 하드웨어 기능 스캔은 일반적으로 PC(160)에서 보안 허가를 확인하기 위해 실행되며, 보안 허가는 일반적으로 포인트(340)에서, 보호된 미디어 파일(130)의 해커(195)에 의한, 허가받지 않은 액세스를 방지하기 위한 적당한 하드웨어 구성(의 일부분)을 가리킨다.

[0019] 도 3은 하드웨어 기능 스캔 시스템을 구비한 PC(210)의 CPU(320) 및 그래픽 디바이스(350) 사이에서 수행되는 하드웨어 기능 스캔을 도시하는 블록도이다. 하드웨어 기능 스캔을 갖는 PC(210) 성능은 일반적으로 버스(340)에 연결된 CPU(320)를 수용할 수 있는 컴퓨터 프로세서 보드(310)를 포함한다. 버스(340)는 또한 그래픽 디바이스(350)에 연결될 수 있다. 그래픽 디바이스(350)는 고유한 방법으로 형상을 렌더링할 수 있는 복잡한 IC를 나타낼 수 있다. 일반적으로, 전형적인 그래픽 디바이스의 복잡도 및 그것이 보유하는 임의의 고유한 렌더링 서명은 해커가 존재하지 않고 그래픽 디바이스(350)가 존재함을 검증하기 위해 쓰일 수 있다.

[0020] 해커(195)에 의한, 허가받지 않은 액세스는 진짜 그래픽 디바이스를 모방하고자 시도하는 디바이스 에뮬레이션의 사용을 통해 시도될 수 있으며, 이는 해커로 하여금 보호되지 않은 미디어(360)에 액세스하고 이를 카피하도록 허용할 것이다. 이러한 구성에서 CPU(320)는 "진짜" 그래픽 디바이스가 존재하고 있다는 것 이외에는 어떤 정보도 가질 수 없을 것이다. CPU(320)는 에뮬레이터가 보호되지 않은 컨텐트를 인터셉트하고 있다고 가리키는 어떤 지시도 수신하지 못할 것이다. 진짜 그래픽 디바이스를 모방하는 그래픽 디바이스 에뮬레이션은 진짜 그래픽 디바이스의 복잡성을 흉내 내지 못할 수 있으므로, 그 복잡성을 테스트하는 진짜 그래픽 디바이스 하드웨어 기능 스캔 시스템(220)의 고유한 렌더링 서명을 만들지 못할 수 있으며 고유한 렌더링 서명은 해커를 감지할

수 있다. 그러므로 진짜 그래픽 디바이스를 모방하는 디바이스는 하드웨어 기능 스캔을 포함하는 시스템(220)에 의해 검증받지 못할 수 있다.

[0021] 컨텐트 제공자(110)는 일반적으로, 보호된 미디어 파일(130)을 디지털로 암호화함으로써 보호된 미디어 파일(130)을 허가없이 카피하거나 보는 것을 방지한다. 이러한 시스템은 일반적으로 신뢰의 고리 구조에 의존한다. 보호된 미디어 파일(130)은 CE 디바이스(150) 또는 PC(210)에 전달하기 위해 조건에 맞는 임의의 현행 암호화 방법을 사용하여 암호화될 수 있다. 예를 들어, PC(210)가 보호된 미디어 파일(130)을 보도록 컨텐트 제공자(110)에 의해 허가를 받은 경우에, (보안 메커니즘을 통해) PC(210)에 보호된 미디어 파일(130)의 해독을 허용하는 암호화 키가 주어질 것이다.

[0022] 참조로써 그 전체가 본원에 포함되어 있는, 1999년 4월 12일에 출원된 출원번호가 09/290,363인 미국 특허출원과 2002년 6월 28일에 각각 출원된 출원번호가 10/185,527, 10/185,278 및 10/185,511인 미국 특허출원에 디지털 권한 관리(Digital Rights Management) 암호화 시스템의 예가 개시되어 있다. 허가받은 PC(210)가 CPU(320)를 이용하여 보호된 미디어 파일(130)을 해독하고 보호되지 않은 미디어(360)를 작성할 수 있다. 보호되지 않은 미디어(360)는 통상적으로 재-암호화되거나 암호화되지 않은 형태로 버스(340)를 통해, 그래픽 디바이스(350)에 전달되는데, 그래픽 디바이스(350)는 보호되지 않은 미디어(360)를, 디스플레이(190)에 의해 디스플레이될 수 있는 비디오 신호(370)로 변환할 수 있다.

[0023] 상술된 바와 같이, 보호되지 않은 미디어(360)는 해커(195)에 의한, 허가받지 않은 액세스가 행해지기 쉬운데, 이는 버스(340) 상에서 보호되지 않은 미디어(360)를 인터셉트하는 해커나 임의의 허가받지 않은 사용자의 형태를 취할 수 있다. 일단 보호된 미디어 파일(130)이 CPU(320)에 의해 해독되면, 이 파일은 보호되지 않은 미디어(360)로 되는데, 이는 그래픽 디바이스(350)를, 보호되지 않은 미디어(360)를 캡처 및 카피할 수 있는 또 다른 디바이스로 교체할 수 있는 해커에 의해 허가받지 않은 카피를 허락하기 쉬운 상태로 된다. 컨텐트의 전달을 보호하도록 주의를 기울이던 컨텐트 제공자(110)는 또한 해커(195)로부터 컨텐트를 보호하기 위한 조치를 취하기를 바랄 수 있다.

[0024] PC는 일반적으로 PC를 훼손하기 다소 쉽게 만드는 개방형 아키텍처를 갖는다. CE 디바이스(150)가, 해커가 그래픽 디바이스(350)를 보호되지 않은 미디어(360)의 카피가 가능한 디바이스로 바꾸기에 어려울 수 있는 폐쇄형 박스 시스템일 수 있는 반면에, PC(210)는 해커 또는 기타 임의의 허가받지 않은 자가 그래픽 디바이스(350)를, 그래픽 디바이스(350)를 모방하고 보호되지 않은 미디어 파일(130)을 카피할 수 있는 디바이스로 바꾸기에 쉬울 수 있는 개방형 박스 시스템이다. 그러므로, 컨텐트 제공자(110)가 보호된 미디어 파일(130)이 PC(210)에 다운로드 또는 스트림되는 것을 허용하기 전에, 컨텐트 제공자(110)는 PC(210)가 HFS(220)에 의해 제공되는 보안 동의를 갖고, 해커 또는 임의의 허가받지 않은 사용자에 의해 설치된, 그래픽 디바이스(350)를 모방하는 어떤 다른 캡처 디바이스가 아닌 그래픽 디바이스(350)에 연결되기를 요구할 수 있다.

[0025] 그래픽 디바이스(350)는 그래픽 디바이스(350)의 인증을 검증하기 위해 CPU(320)에 의해 퀴리될 수 있는 디지털로 서명된 인증서를 포함할 수 있다. 그러나, 그래픽 디바이스(350)를 생성하는 데에 사용된 제조 프로세스의 특성 때문에 각각의 그래픽 디바이스(350) 내에 고유한 인증서 또는 기타 고유한 식별자를 인코드하는 것은 비용면에서 효율적이지 않을 수 있다. 그래픽 디바이스(350)의 인증을 증명하기 위한 더 간단하거나 더 비용 효율적인 해결방안이 사용될 수 있거나, 디바이스 인증서 솔루션을 증대시키도록 사용될 수 있다. CPU(320)는 하드웨어 기능 스캔 시스템(220)을 채용할 수 있다.

[0026] 그래픽 디바이스(350)는 일반적으로 복잡한 배열로 서로 연결된 하나 이상의 IC를 통한 다수의 논리 게이트로 이루어진 복잡한 디바이스이다. 그래픽 디바이스(350)는 또한 형상 및 기타 그래픽적인 요소를 고유한 방식으로 렌더링할 수 있다. 그래픽 디바이스(350)가 형상 및 기타 그래픽적인 요소를 렌더링할 수 있는 고유한 방식은, 그래픽 디바이스(350)를 모방하는 어떤 다른 디바이스가 아닌, 진짜 그래픽 디바이스(350)에 연결되었는지를 검증하기 위하여 CPU(320)에 의해 활용될 수 있다. CPU(230)는 그래픽 디바이스(350)에 형상 또는 기타 그래픽적인 요소를 제공하여 렌더링하고 그 렌더링의 결과를 예상된 결과와 비교하는 것과 같은, 그래픽 디바이스(350)의 고유한 복잡한 하드웨어 구성을 테스트하도록 하는 퀴리를 수행함으로써, 하드웨어 기능 스캔(220)을 실행할 수 있다. 일반적으로 그래픽 디바이스(350)의 복잡성 때문에 해커 또는 기타 허가받지 않은 자가 에뮬레이션으로 올바른 응답을 하드웨어 기능 스캔(220)에 복사하거나 만드는 것은 어렵다.

[0027] 그래픽 디바이스(350)를 고유하게 식별하기 위해, 특정한 그래픽 디바이스(350) 만이 그래픽 디바이스(350)를 검증하는 회답 또는 응답을 제공할 수 있는 방식으로 그래픽 디바이스(350)의 퀴리 또는 요청이 구성될 수 있다. 이는, 그래픽 디바이스(350) 및 일반적인 그래픽 디바이스가 수많은 게이트의 복잡한 배열로 구성되고

그것에 대해 일반적으로 복잡한 상태 모델을 구현하였기 때문에 일반적으로는 가능하다. 그러므로, 그래픽 디바이스의 두 개의 서로 다르게 제조된 모델로 만들어진 동일한 질문 또는 요청이 서로 다른 회답을 하거나 서로 다른 결과를 도출할 수 있다. 일반적으로 회답 또는 도출된 결과의 해석이 그래픽 디바이스(350)를 식별할 것이다.

[0028] 예를 들어, CPU(320)가 그래픽 디바이스(350)에 3차원 형상을 보내고 그래픽 디바이스(350)가 음영과 같은, 3차원 공간에서의 변형을 수행하도록 요청할 수 있다. 그래픽 디바이스(350)는 그 후 변형되거나 렌더링된 3차원 형상의 결과를 CPU(320)에 보낼 수 있다. CPU(320)는 변형된 복잡한 3차원 형상의 수학적인 표현이 CPU(320)에 의해 예상된 결과와 일치하는지를 판별하기 위해, 도출된 결과를 검사할 수 있다. 비교는 룩업(lookup) 테이블 또는 하드웨어의 소프트웨어 에뮬레이션 등을 참고함으로써 이루어질 수 있다.

[0029] 다른 예에서는, CPU(320)가 복잡한 수학적 표현을 저장했을 수 있다. 통상적인 표현은 제조된 모델에 대한 고유하고 공지된 회답을 일반적으로 계산하는 그래픽 디바이스(350)의 영역에 작용할 것이다. 또한, 통상적인 표현은 랜덤 데이터 및/또는 스스로 무작위로 선택될 수 있는 표현을, 표현에 대하여 랜덤 파라미터와 일치하는 형태로 또한 포함할 수 있다. 예를 들면, 계산된 결과는 활용될 수 있는 공지된 라운딩 에러 또는 고유한 개수의 자릿수를 가질 수 있다. 또한, 다른 예에서는, 집적 회로 내의 추가적인 바운더리 스캔 회로 소자가 공장에서 그래픽 디바이스(350)의 기능을 검증하도록 추가되었을 수 있도록 그래픽 디바이스(350)가 제조될 수 있다. 바운더리 스캔 회로 소자는 그래픽 디바이스(350)의 각 모델에 대해 고유할 수 있으며 CPU(320)는 그래픽 디바이스(350)를 확인하기 위해 바운더리 스캔 회로 소자를 쿼리하고 그 결과를 분석할 수 있다.

[0030] 도 4는 하드웨어 기능 스캔을 수행하는 예시적인 프로세스를 도시하는 흐름도이다. 시퀀스(400)는 일반적으로 (도 3의)CPU(320)상에서 실행되지만, 임의의 프로세서상에서 실행될 수도 있다.

[0031] 블록(410)에서 CPU는 일반적으로 그래픽 디바이스의 인증을 검증하기 위한 쿼리를 그래픽 디바이스에 보낸다. 상술된 바와 같이, 합의된 랜덤 값이 사용되며 오직 진짜 그래픽 디바이스만이 그래픽 디바이스를 검증하는 회답 또는 응답을 제공할 수 있는 방법으로 쿼리가 구성될 수 있다.

[0032] 또한, 블록(415)에서, 그래픽 디바이스는 그 후 일반적으로 쿼리를 처리하여 결과를 생성하고 그 결과를 평가하기 위해 CPU에 넘길 것이다.

[0033] 다음으로 블록(420)에서, CPU는 일반적으로 쿼리의 결과를 그래픽 디바이스로부터 수신한다. CPU가 그래픽 디바이스가 진짜인지를 판별하기 위해 쿼리의 결과를 수신할 필요가 없을 수도 있음에 유의해야 한다. 그래픽 디바이스가 쿼리의 결과를 CPU에 보내지 않고, 그래픽 디바이스가 올바른 회답을 갖고 있다고 증명하기 위해, 영지식증명(zero-knowledge-proof)이 그래픽 디바이스와 함께 사용될 수 있다. 예를 들면, 그래픽 디바이스 및 CPU가, 그래픽 디바이스로의 후속하는 메세지에 대한 키로서 쿼리의 결과를 사용할 수 있으며, 그래픽 디바이스가 쿼리에 올바른 응답을 생성하는 경우에 한하여 그래픽 디바이스가 계속 작동할 수 있는데, 이는 기능을 계속할 것을 허용하는 후속 메세지를 수신할 수 없을지도 모르기 때문이다.

[0034] 블록(430)에서, CPU는 그 후 일반적으로 그래픽 디바이스로부터 수신된 쿼리의 결과를 예상된 결과와 비교한다. 그러면 CPU는 비교의 결과를 분석하고 비교가 통과했는지 실패했는지의 여부를 판별한다. 비교가 실패한 경우에는 블록(440)에서 검증이 일반적으로 종료될 것이다.

[0035] 블록(440)에서 프로세스를 종결하는 것은, 허가받지 않은 그래픽 디바이스 또는 해커가 존재함을 가리킬 수 있는, 그래픽 디바이스로부터 반환된 결과가 예상된 결과와 다르다고 판별한 CPU의 결과일 수 있다. 확인되지 않았기 때문에 그래픽 디바이스에 보안 동의가 발행될 수 없으므로, 실행의 흐름은 일반적으로 이 시점에서 끝난다.

[0036] 블록(450)에서 프로세스를 계속하는 것은, 그래픽 디바이스로부터 반환된 결과가 예상된 결과에 비교되었을 때에 기준에 허용 가능하다고 판별한 CPU의 결과일 수 있다. 보안 시스템은 그래픽 디바이스가 하드웨어 기능 스캔을 통과했으며 에뮬레이션 디바이스를 갖는 해커가 아니라 인증된 그래픽 디바이스라고 결론지을 수 있다. CPU는 그러면 그래픽 디바이스 대신에 보안 허가를 발행할 수 있는데, 보안 허가는 그래픽 디바이스가 검증되었음을 가리킨다.

[0037] 도 5는 하드웨어 기능 스캔 시스템이 구현될 수 있는 예시적인 컴퓨팅 환경을 도시하는 블록도이다.

[0038] 하드웨어 기능 스캔 시스템이 설비된 PC(210; 도 2)는 일반적으로 운영 체제(505)를 실행하여 애플리케이션(510)을 실행시킨다. 애플리케이션(510)은 일반적으로 상호운용성 게이트웨이(520)에 연결된다. 상호운용성

게이트웨이(520)는 일반적으로 하드웨어 드라이버(530)에 연결되며, 또한, 상호운용성 게이트웨이(520)는 하드웨어 드라이버(530)에 결합된 보안을 가질 수 있다. 하드웨어 드라이버(530)는 일반적으로 하드웨어 추상화 계층(535)에 연결되며, 하드웨어 추상화 계층(535)은 하드웨어 디바이스(540)에 연결될 수 있다.

[0039] 운영 체제(505)는 사용자 모드(580) 및 커널 모드(590)를 구현할 수 있다. 애플리케이션(510)은 일반적으로 사용자 모드(580)에서 실행되며, 상호운용성 게이트웨이(520)도 또한 일반적으로 사용자 모드(580)에서 실행된다. 하드웨어 드라이버(530)는 일반적으로 커널 모드(590)에서 실행된다. 운영 체제(505)는 일반적으로 보안상의 이유로 사용자 모드(580) 및 커널 모드(590)를 구현한다. 해커에 의한 액세스에 대해 더 취약할 수 있는 PC(210)의 요소로의 액세스를 커널 모드(590)가 가질 수 있기 때문에 운영 체제(505)는, 운영 체제(505)가 커널 모드(590)에 제공할 수 있는 것보다 더 적은 보안 허가를 갖는 사용자 모드(580)에 제공할 수 있다. 운영 체제(505)는 디지털로 서명되지 않고 신뢰되지 않는 컴포넌트가 커널 모드(590)에서 실행되도록 허용하지 않을 수 있다. 운영 체제(505)는 일반적으로 더 적은 액세스를 사용자 모드(580)에 제공하며, 이에 따라 해커에 대해 더욱 취약할 수 있는, PC(210)의 요소로의 더 낮은 레벨의 액세스를 제공한다. 운영 체제(505)는 또한 일반적으로 사용자 모드(580) 및 커널 모드(590)를 동시에 실행할 수도 있고, 둘 이상의 사용자 모드(580)의 인스턴스를 동시에 또한 실행할 수도 있다.

[0040] 또한, 운영 체제(505)는 일반적으로 서로 다른 레벨의 보안 실행 환경을 포함함으로써 추가적인 보안 계층을 구현할 수 있다.

[0041] 운영 체제(505)는 보호된 실행 환경(570) 및 보호되지 않은 실행 환경(580)을 포함할 수 있는데, 보호되지 않은 실행 환경(580)은 보호된 실행 환경(570)보다 더 적은 보안 허가를 포함한다. 운영 체제(505)가 상호운용성 게이트웨이(520) 또는 하드웨어 드라이버(530)로 하여금 보호된 실행 환경(570)에서 로드되거나 실행되도록 허용하기 전에 운영 체제(505)는 일반적으로 보안 요건 세트를 부과한다. 예를 들면, 보안 요건은 어떤 형태의 디지털 서명 또는 다른 디지털 신뢰의 증명일 수 있다. 이 방식으로, 운영 체제(505)는 상호운용성 게이트웨이(520) 또는 하드웨어 드라이버(530)를 신뢰하고 상호운용성 게이트웨이(520) 또는 하드웨어 드라이버(530)에게 운영 체제(505)가 제어하는 PC(210)의 자원으로의 더 많은 액세스를 승인할 수 있다. 또한, 애플리케이션(510)으로 하여금 로드되거나 실행되도록 허용하기 전에 운영 체제(505)는 일반적으로 다 작은 보안 요건의 세트를 일반적으로 구현할 수 있지만, 운영 체제(505)는 애플리케이션(510)에게 운영 체제(505)가 제어하는 PC(210)의 자원으로의 더 적은 액세스를 승인할 수 있다.

[0042] 하드웨어 드라이버(530)가 커널 모드(590) 및 보호된 실행 환경(570) 양쪽 모두에서 실행할 수 있으므로, 이 보안의 레벨은 컨텐트 제공자에 있어서 하드웨어 디바이스(540)를 인증하기에 만족스러울 것이다. 또한, 커널 모드(590)는, 하드웨어 드라이버(530)가 합법적인 소스로부터 수신되었음을 증명하기 위해 하드웨어 드라이버(530)가 커널 모드(590)에서 로드 및 실행될 수 있기 전에 디지털로 서명되고 신뢰되도록 요구할 수 있다. 오직 신뢰되는 드라이버만이 로드될 수 있음이 또한 중요함을 유의해야 한다. 예를 들면, 커널 모드 내의 다른 모든 드라이버가 또한 신뢰되지 않는다면 단지 하드웨어 드라이버(530)가 신뢰된다는 이유만으로 커널 모드 문제가 해결되는 것은 아니다. 이 컨셉은 문서 내 어딘가에서 포착될 필요가 있다.

[0043] 운영 체제(505)는 디지털 권한 관리("DRM")를 구현할 수 있다. 컨텐트 제공자는 DRM을 신뢰하고 컨텐트 제공자가 순서대로 컨텐트에 대해 DRM에 주어진 정책을 DRM이 구현할 것을 요구할 수 있다. 그러면 DRM은 컨텐트가 디지털로 서명된 컴포넌트(드라이버 및 사용자 모드 컴포넌트)와 함께 사용되었음을 검증하고, 요청되는 경우에는, 그래픽 드라이버가 하드웨어 기능 스캔(220)을 받았음을 검증한다. 컨텐트 제공자(110)는 하드웨어 드라이버(530)가 컨텐트 제공자 대신에 하드웨어(540)를 인증했음에 만족할 수 있으며, 그러므로 이 취약한 포인트에서 해커가 일반적으로 컨텐트 제공자의 컨텐트를 카피하기 위해, 진짜 하드웨어 디바이스(540)를 하드웨어 디바이스(540)의 에뮬레이션과 바꾸어 놓지 않았음에 컨텐트 제공자는 만족할 수 있다.

[0044] 진짜 하드웨어 디바이스(540)(예를 들면, 그래픽 하드웨어;545)가 허가받지 않은 카피를 방지하는 보안 요소를 구현함으로써 카피로부터의 보호를 제공할 수 있는 반면에, 해커 또는 기타 허가받지 않은 제3자는 하드웨어 디바이스(540)의 에뮬레이션을 생성하고 그것을 PC(210)에 삽입할 수 있다. 이러한 위조 하드웨어 디바이스는 진짜 하드웨어 디바이스(540)인 것처럼 보일 수 있지만, 해커 또는 기타 허가받지 않은 제3자는 보안 기능이 가능하지 않을 때에 보안 기능이 가능하다고 보고하도록 에뮬레이트된 하드웨어 디바이스를 구축해 놓았을 수 있다. 그렇게 함으로서, 하드웨어 드라이버(530), 예를 들면, 그래픽 드라이버(535)가 정보의 취약한 버전을 컨텐트 제공자(110)로부터 위조 하드웨어 디바이스에 제공할 수 있으며, 위조 하드웨어 디바이스는 정보를 자유롭게 카피할 수 있다.

[0045] 따라서, 운영 체제(505)는 서명되고 신뢰되는 드라이버(530)를 사용하고, 하드웨어 드라이버(530)에게 하드웨어 추상화 계층(535)을 사용하는 하드웨어 기능 스캔(220)을 수행하도록 요청함으로써 하드웨어 디바이스(540)가 진짜임을 검증할 수 있다. 하드웨어 기능 스캔(220)은 하드웨어 디바이스(540)가 진짜 하드웨어 디바이스인지 아닌지의 여부와 해커에 의해 마련된 에뮬레이션이 아님을 판별할 수 있다. 또한, 커널 모드(590)의 무결성을 확실히 하기 위하여 운영 체제(505)는 모든 컴포넌트가 커널 모드에 대하여 로드되며 서명 및 신뢰됨을 검증할 수 있다.

[0046] 하드웨어 기능 스캔(220)는 일반적으로 하드웨어 드라이버(530)에 의해 하드웨어(540)에 보내진 쿼리이다. 쿼리는 하드웨어 디바이스(540)의 고유한 복잡 하드웨어 구성을 테스트하기 위해 작성될 수 있다. 하드웨어 디바이스(540)는 복잡한 디바이스일 수 있으며 하드웨어 디바이스(540)의 에뮬레이션이 올바른 응답을 카피하거나 만들기가 어려울 수 있다. 즉, 하드웨어 기능 스캔(220)을 수행할 때, 쿼리에 대한 회답이 일반적으로 하드웨어 디바이스(540)를 고유하게 식별하는 방식으로, 하드웨어 드라이버(530)에 의해 쿼리가 구성될 수 있다.

[0047] 또한, 하드웨어 드라이버(530)는 하드웨어 디바이스(540)에 보내는 쿼리의 테이블(550)을 저장할 수 있다. 이 쿼리는 랜덤 입력 데이터를 받아들일 수 있으며, 하드웨어 드라이버(530)가 그 다음에 쿼리에 대한 입력을 랜덤하게 선택할 수 있다. 하드웨어 드라이버(530)는 그 후 하드웨어(540)에 의해 반환된 회답을 예상한 회답과 비교한다. 이러한 비교가 하드웨어(540)로부터 회답을 요청함으로써 직접 행해질 수 있거나 혹은 이와 달리 하드웨어가 알맞은 회답을 생성한 경우에서만 성공할 추후의 동작의 회답을 사용함으로써 간접적으로 행해질 수 있다. 하드웨어 드라이버(530)가 회답이 서로 일치한다고 판별하는 경우에, 하드웨어 드라이버(530)는 하드웨어 디바이스(540)가 검증 및 인증된 것으로 또한 결정할 수 있다.

[0048] 다른 예를 들면, 하드웨어 드라이버(530)가 하드웨어(540)의 임의의 부분의 에뮬레이터(560)를 구현할 수 있다. 에뮬레이터(560)가 하드웨어(540)의 에뮬레이션일 수 있어서 하드웨어 드라이버(530)가 에뮬레이터(560)를 사용하여 값을 선택하고 동작을 수행할 수 있고, 그 후 동일한 값 및 요청을 하드웨어(540)에 넘길 수 있어서 그 결과 하드웨어(540)가 동일한 값을 갖는 동일한 동작을 수행할 수 있다. 하드웨어 드라이버(530)는 그 후 하드웨어(540)가 검증 및 인증되었는지를 판별하기 위해 에뮬레이터(560)와 하드웨어(540)에 의해 수행됨에 따라 동작의 결과를 검증한다.

[0049] 일단 하드웨어 드라이버(530)가 하드웨어 기능 스캔(220)을 수행하고 진짜 하드웨어 디바이스(540)가 적소에 있는 것으로 판별하면, 하드웨어 드라이버(530)는 상술한 바와 같이 하드웨어 디바이스(540)를 인증 및 검증하는 기능을 수행하고 컨텐트 제공자와 합의한 신뢰를 만족했을 수 있다.

[0050] 다른 예를 들면, 하드웨어 기능 스캔(220)을 이용한 하드웨어 디바이스(540)를 검증하는 것에 응답하여 하드웨어 드라이버(530)에 의해 제공될 수 있는 보안 증명을 다루기 위해, PC(210)가 상호운용성 게이트웨이(520)(예를 들면, 출력 보호 매니저;525)를 사용할 수 있다. 또한, 하드웨어 드라이버(530)에 의해 제공되지만 상호운용성 게이트웨이에 의해서는 제공되지 않는 기능에 대한 임의의 타입의 액세스를 해커가 갖는 것으로부터 방지하고자, 상호운용성 게이트웨이(520)는 하드웨어 드라이버(530)에 의해 제공된 기능의 축소된 서브셋을 제공할 수 있다.

[0051] 도 6은 하드웨어 기능 스캔의 결과를 미디어 파이프라인(523)에 전달할 수 있는 출력 보호 관리 소프트웨어 모듈에 연결된 하드웨어 기능 스캔 프로세스를 도시하는 블록도이다.

[0052] 하드웨어 기능 스캔 시스템은 하드웨어 기능 스캔의 결과를 미디어 파이프라인(523)과 같은, 컨텐트 배포를 제어하는 시스템에 전달하기 위해 형성될 수 있다. 그 시스템은 출력 보호 관리 모듈(525), 그래픽 드라이버(535) 및 그래픽 하드웨어(545)를 포함할 수 있다.

[0053] 출력 보호 관리 모듈(525)은 PC의 보호된 환경 내에서 실행하며 PC의 사용자 모드에서 또한 실행할 수 있는 소프트웨어에 구현된 모듈일 수 있다. 출력 보호 관리 모듈(525)은, 보안 인증 또는 그래픽 드라이버(545)가 신뢰된다고 가리키는 기타의 보안 증명 형태를 수신하며 하드웨어 기능 스캔 쿼리(410) 및 하드웨어 기능 스캔 응답(420)을 포함할 수 있는 하드웨어 기능 스캔(220)을 수행하는 보안 권한자 역할을 할 수 있다. 또한, 출력 보호 관리 모듈(525)은 보안 인증 또는 커널 모드(590)가 신뢰된다고 가리키는 기타의 보안 증명의 형태를 수신할 수 있다. 출력 보호 관리 모듈(525)은 보안 인증 또는 기타의 보안 증명의 형태의 실재물을 컨텐트 제공자에게 전달한다.

[0054] 그래픽 드라이버(535)는 일반적으로 통상의 PC에 구현되어, 상술한 대로 그래픽 하드웨어(545)로의 일관적이며 단일한 액세스 포인트를 제공한다. 그래픽 하드웨어(525)는, PC가 명령하는 대로 형상 또는 기타 그래픽적인

정보를 랜더링하기 위해 사용된 임의의 하드웨어 디바이스일 수도 있다. 그래픽 하드웨어(545)는 하나의 집적 회로 칩을 포함하거나 집적 회로 칩의 임의의 조합으로 만들어질 수 있다.

[0055] 컨텐트 제공자가 그래픽 드라이버(535)를 신뢰해서 고급 또는 고가의 컨텐트를 PC에서 재생하기 전에 컨텐트 제공자는 그래픽 하드웨어(545)가 검증되었으며 인증되었다는 증명을 요청할 수 있다. 컨텐트 제공자는 그래픽 드라이버(535)와 직접적으로 통신하지 못할 수 있으며, 그래픽 드라이버(535)가 하드웨어 기능 스캔 쿼리(410)를 수행했는지와 그래픽 하드웨어(545)를 검증한 하드웨어 기능 스캔 회답(420)을 수신했는지를 판별하지 못할 수 있다.

[0056] 그러나, 상호운용성 게이트웨이(520), 출력 보호 매니저(525), 및 하드웨어 드라이버(530)가 신뢰됨을 검증함으로써 보호된 환경(570)이 신뢰 됨을 검증할 수 있는 운영 체제(505)가 신뢰되고, 그래픽 드라이버(535)가 신뢰되기 때문에, 그러한 신뢰를 요청할 수 있는 컨텐트가 상호운용성 게이트웨이(520)로부터 하드웨어 드라이버(530)에 주어지기 전에, 컨텐트 제공자는, 하드웨어 드라이버(530)에게 하드웨어(540)가 신뢰 되는지를 검증하기 위해 하드웨어 기능 스캔(220)을 활용하는 것을 시행하게 할 수 있다.

[0057] 예를 들어, 그래픽 드라이버(535)는 하드웨어 기능 스캔 쿼리(410)를 생성할 수 있고 하드웨어 기능 스캔 쿼리(410)를 그래픽 하드웨어(545)에 보낼 수 있다. 그래픽 하드웨어(545)가 합법적이며 컨텐트를 카피하려는 목적을 위해 해커에 의해 마련된 위조물 또는 다른 디바이스가 아님을 하드웨어 기능 스캔 회답(420)이 확인하는 방법으로 하드웨어 기능 스캔 쿼리(410)가 일반적으로 구성된다. 또한, 하드웨어 기능 스캔 쿼리(410)는 임의의 형태를 취할 수 있지만, 일반적으로는 해커가 에뮬레이트 하기에 어려운 그래픽 하드웨어(545)의 영역 내에서 수행될 기능의 형태를 취한다. 또한, 하드웨어 기능 스캔 회답(420)은 그래픽 하드웨어(545)가 알맞은 회답을 독립적으로 만들었는지를 판별하기 위해 임의의 형태를 취할 수 있으나, 일반적으로 예상된 회답에 비교될 수 있는 데이터의 형태를 취한다.

[0058] 다음에, 그래픽 드라이버(535)는 하드웨어 기능 스캔 회답을 결정한다. 예를 들면, 그래픽 드라이버(535)가 그래픽 드라이버(535) 내부에 저장된 룩업 테이블로부터 회답을 이끌어냈을 수 있거나, 그래픽 드라이버(535)가 무작위로 또는 세트된 스케줄 값으로부터 값을 고르고, 그래픽 하드웨어(545)의 내부 에뮬레이션에 값을 넘기거나 그것의 어떤 조합을 사용했을 수 있다. 또한, 다른 방법의 예를 들면, 그래픽 하드웨어(545)와 그래픽 드라이버(535) 양쪽 모두가 쿼리(410)에 대한 입력으로 사용할 랜덤 값에 대해 그래픽 하드웨어(545) 및 그래픽 드라이버(535)가 합의했을 수 있다. 그래픽 드라이버(535) 내에 저장된 그래픽 하드웨어(545)의 내부 에뮬레이션은 그러면 내부 에뮬레이션에 대한 입력으로 선택된 값을 사용하여 회답을 계산할 수 있다.

[0059] 다음에, 그래픽 하드웨어(545)는 하드웨어 기능 스캔 회답(420)을 계산하고 그 값을 일반적으로 그래픽 드라이버(535)에 반환한다.

[0060] 그러면 그래픽 드라이버(535)는 그래픽 하드웨어(545)가 만든 하드웨어 기능 스캔 회답(420)이 그래픽 드라이버(535)가 계산한 회답과 일치하는지를 판별한다. 그래픽 드라이버(535)가 회답이 예상된 회답과 동일하다고 판별하면, 그래픽 드라이버(535)는 그래픽 하드웨어(545)의 인증을 확증할 수 있다. 예를 들면, 그래픽 드라이버(535)는 그 후에 출력 보호 관리 모듈(525)에 보안 통지를 보낼 수 있다.

[0061] 다음에, 출력 보호 관리 모듈(525) 또는 그래픽 드라이버(535)는 보안 상태를 필요한 만큼 오랫동안 저장할 수 있다. 출력 보호 관리 모듈은 또한 그래픽 하드웨어(545)의 인증을 재검증하기 위해 그래픽 드라이버(535)로 하여금 새로운 하드웨어 기능 스캔 쿼리(410)를 보내도록 하는 요청을 그래픽 드라이버(535)에 발행한다.

[0062] 마지막으로, 일단 출력 보호 관리 모듈(525)이 그래픽 드라이버(535)로 하여금 그래픽 드라이버(535)와 그래픽 하드웨어(545) 간의 채널(660)을 검증하도록 요청할 수 있다. 일단 그래픽 드라이버(535)가 그래픽 드라이버(535)와 그래픽 하드웨어(545) 간의 채널(660)을 검증하면, 출력 보호 관리 모듈(525)이 컨텐트 제공자와 통신하고 그래픽 드라이버(535)와 그래픽 하드웨어(545) 간의 연결이 안전하며 고급 또는 보호된 컨텐트를 재생할 수 있음을 가리킬 수 있다.

[0063] 당업자는 프로그램 명령어를 저장하기 위해 활용된 저장 디바이스가 네트워크에 걸쳐 분산될 수 있음을 인식할 것이다. 예를 들어 원격 컴퓨터는 소프트웨어로 기술된 프로세스의 예를 저장할 수 있다. 로컬 컴퓨터 또는 단말기 컴퓨터는 원격 컴퓨터를 액세스할 수 있고 프로그램을 실행하기 위해 소프트웨어의 일부 또는 전부를 다운로드할 수 있다. 다른 방법으로는 로컬 컴퓨터가 필요한 소프트웨어의 일부를 다운로드하거나, 일부 소프트웨어 명령어는 로컬 단말기에서 그리고 일부는 원격 컴퓨터(또는 컴퓨터 네트워크)에서 실행함으로써 분산적으로 프로세스할 수 있다. 당업자는 또한 당해 기술 분야에 공지된 통상의 기술을 활용함으로써 그 소프트웨어

명령어의 전부 또는 일부가 DSP, 프로그램가능한 논리 어레이 등과 같은 전용 회로에 의해 수행될 수 있음을 인식할 것이다.

도면의 간단한 설명

[0004] 본 설명은 첨부한 도면을 고려하여 파악되는 다음의 자세한 서술로 더 잘 이해될 것이다.

[0005] 도 1은 하드웨어 기능 스캔("HFS") 시스템이 없는 종래의 PC 및 보안 시스템을 갖는 CE 디바이스를 도시하는 블록도.

[0006] 도 2는 하드웨어 기능 스캔 시스템을 갖는 종래의 PC 및 보안 시스템을 갖는 CE 디바이스를 도시하는 블록도.

[0007] 도 3은 하드웨어 기능 스캔 시스템에 의해 보호되고 있는 프로세서의 CPU 및 그래픽 디바이스를 도시하는 블록도.

[0008] 도 4는 하드웨어 기능 스캔을 실행하기 위한 예시적인 프로세스를 도시하는 흐름도.

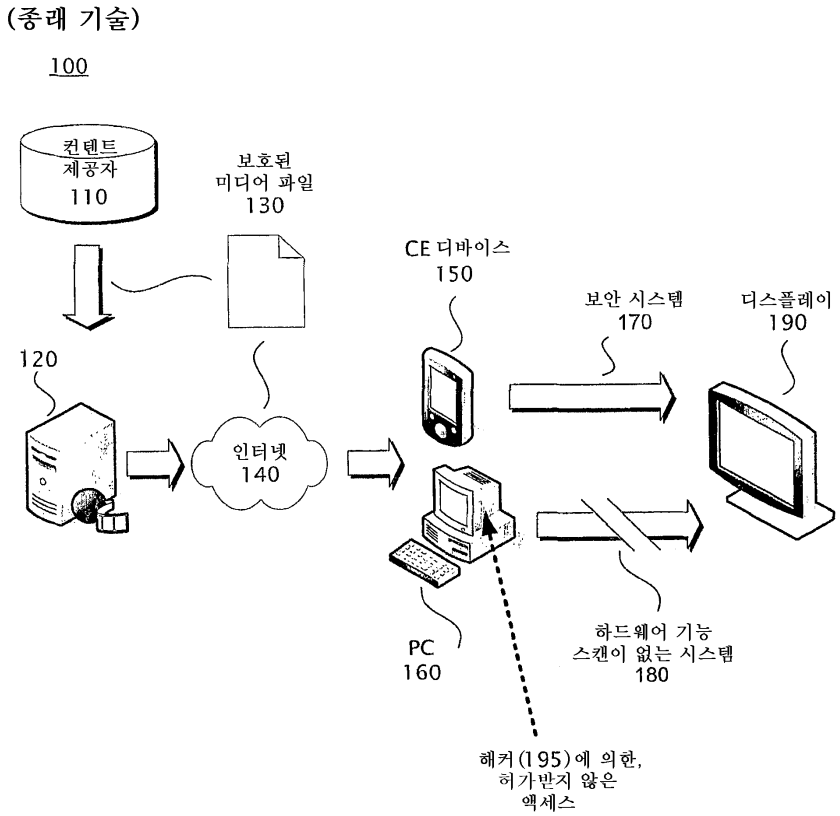
[0009] 도 5는 하드웨어 기능 스캔 시스템이 구현될 수 있는 예시적인 컴퓨팅 환경을 도시하는 블록도.

[0010] 도 6은 하드웨어 기능 스캔 시스템의 예시적인 구현예 및 하드웨어 기능 스캔 시스템의 예시적인 구현예의 요소들 간의 정보의 교환을 도시하는 블록도.

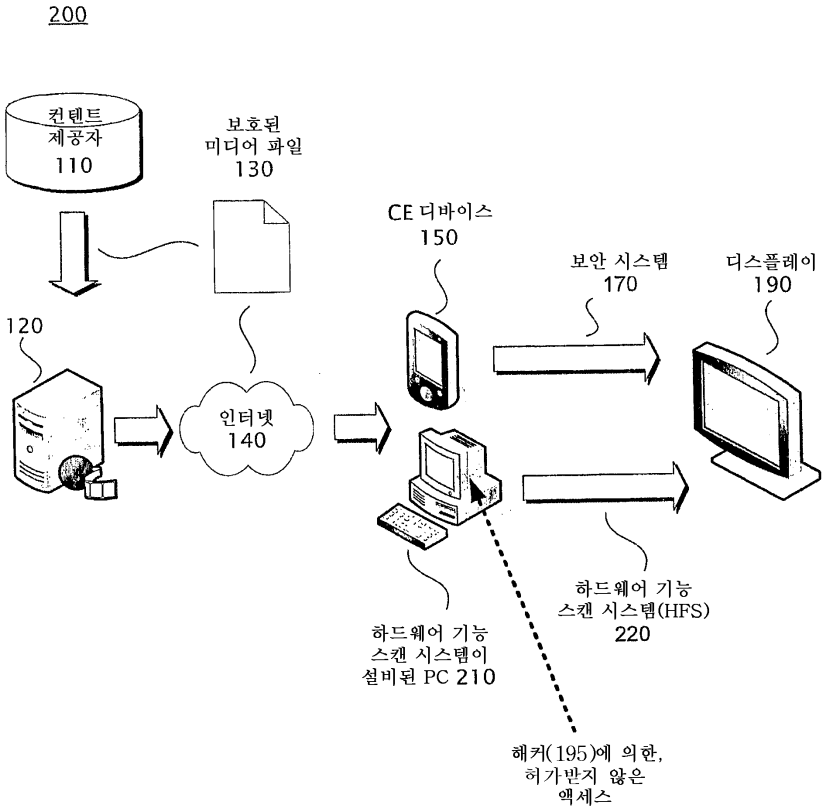
[0011] 첨부된 도면에서 동일한 참조 번호가 동등한 부분을 나타내기 위해 사용된다.

도면

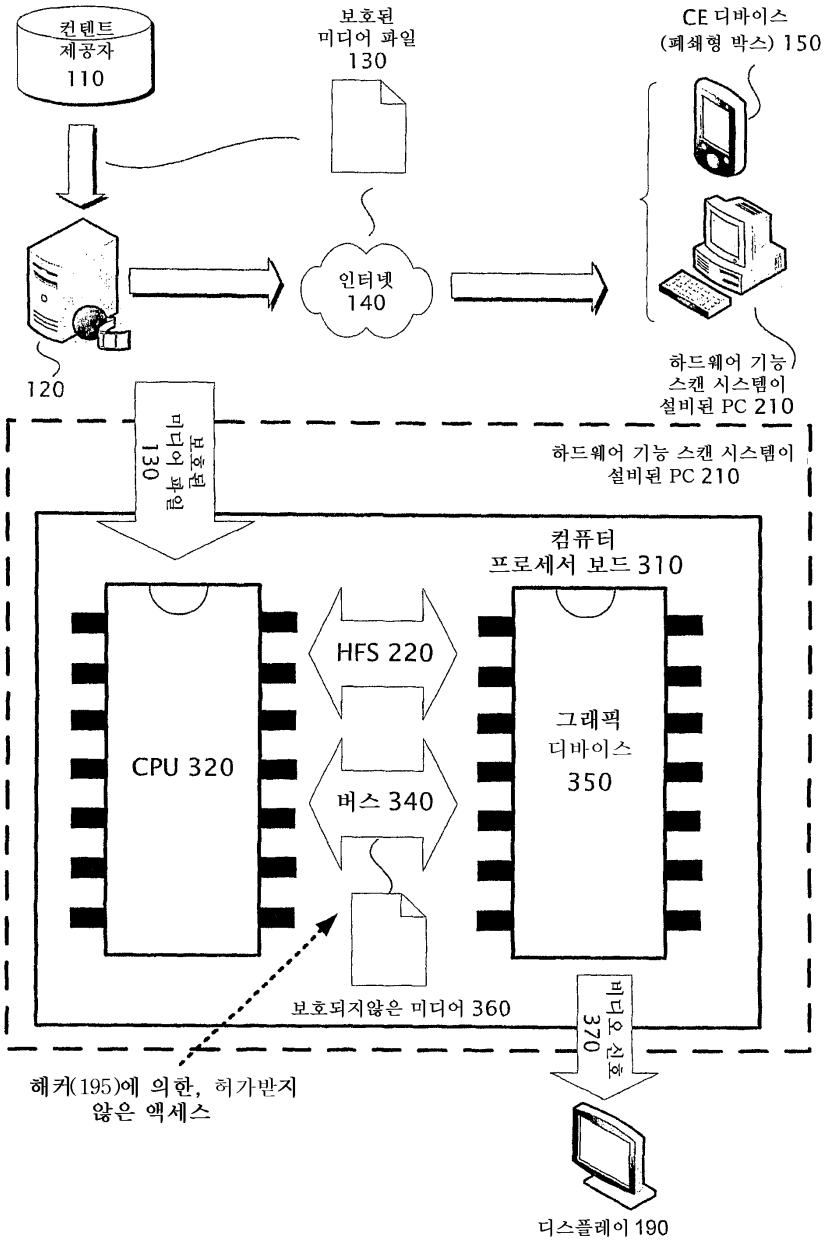
도면1



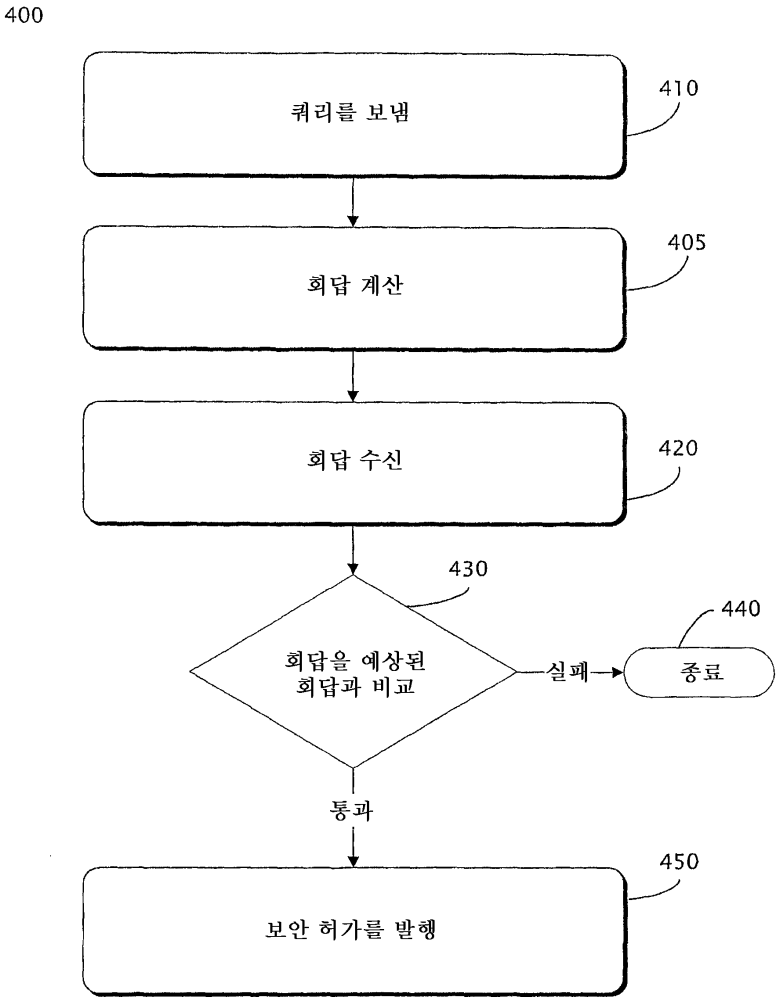
도면2


200
컨텐트
제공자
110
보호된
미디어 파일
130
CE 디바이스
150
보안 시스템
170
디스플레이
190
120
인터넷
140
하드웨어 기능
스캔 시스템(HFS)
220
하드웨어 기능
스캔 시스템이
설비된 PC 210
해커(195)에 의한,
허가받지 않은
액세스

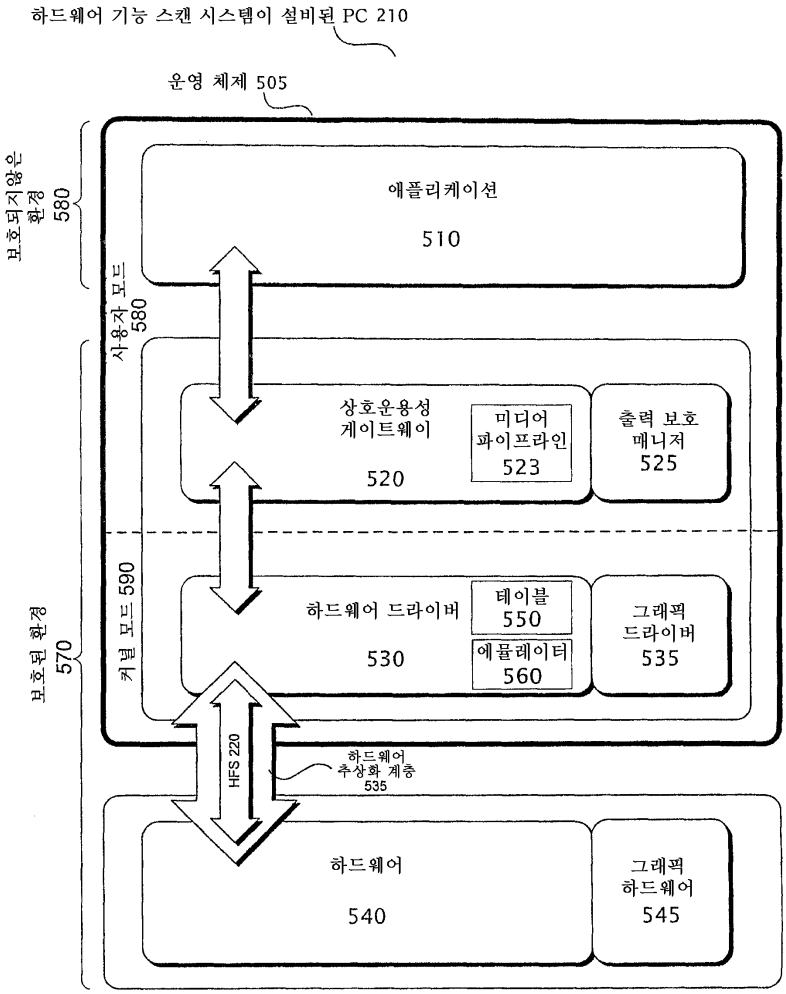
도면3


컨텐트
제공자
110
보호된
미디어 파일
130
CE 디바이스
(폐쇄형 박스) 150
인터넷
140
120
하드웨어 기능
스캔 시스템이
설비된 PC 210
보호된
미디어 파일
130
하드웨어 기능 스캔 시스템이
설비된 PC 210
컴퓨터
프로세서 보드 310
HFS 220
CPU 320
버스 340
그래픽
디바이스
350
보호되지않은 미디어 360
비디오 신호
370
해커(195)에 의한, 허가받지
않은 액세스
디스플레이 190

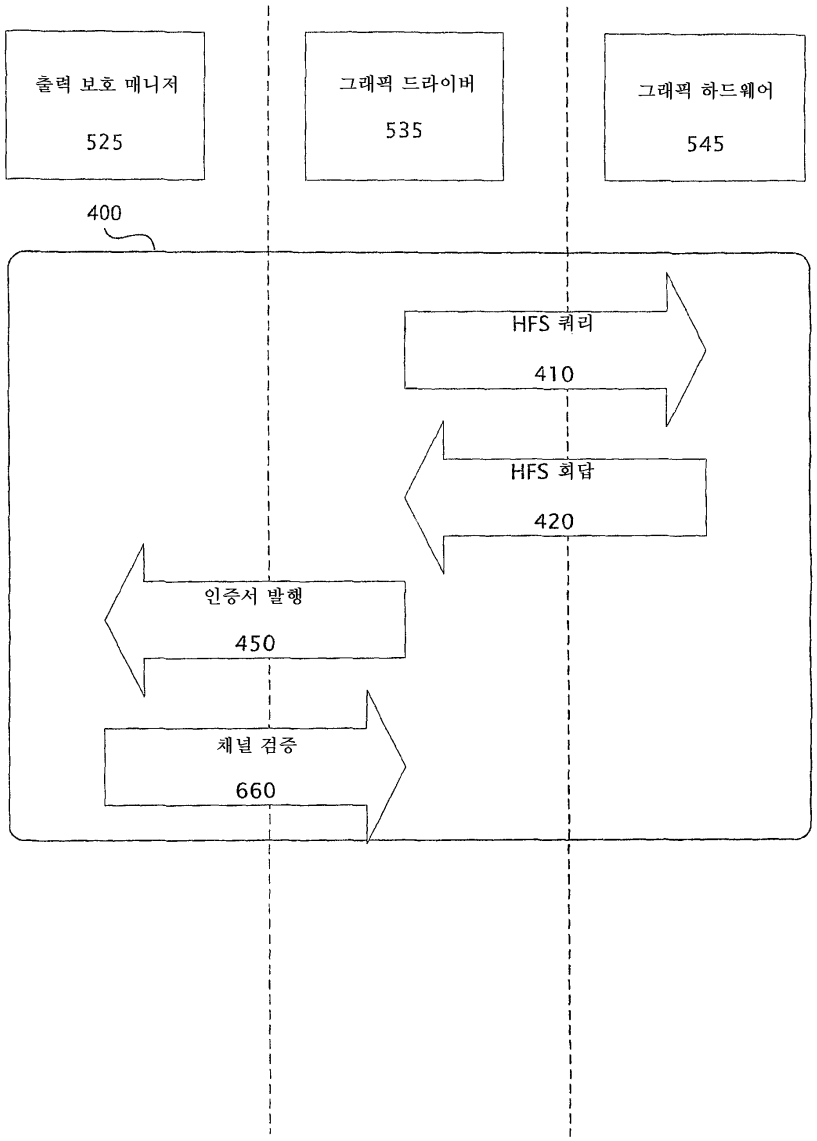
도면4


400
410
쿼리를 보냄
405
회답 계산
420
회답 수신
430
회답을 예상된 회답과 비교
440
실패
종료
통과
450
보안 허가를 발행

도면5


하드웨어 기능 스캔 시스템이 설비된 PC 210
운영 체제 505
보호되지않은 환경 580
애플리케이션 510
사용자 모드 580
상호운용성 게이트웨이 520
미디어 파이프라인 523
출력 보호 매니저 525
커널 모드 590
하드웨어 드라이버 530
테이블 550
에뮬레이터 560
그래픽 드라이버 535
보호된 환경 570
HFS 220
하드웨어 추상화 계층 535
하드웨어 540
그래픽 하드웨어 545

도면6


출력 보호 매니저
525
그래픽 드라이버
535
그래픽 하드웨어
545
400
HFS 쿼리
410
HFS 회답
420
인증서 발행
450
채널 검증
660