(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년03월24일
(11) 등록번호 10-2092299
(24) 등록일자 2020년03월17일

(51) 국제특허분류(Int. Cl.)
G06F 21/70 (2013.01)
(21) 출원번호 10-2013-0015816
(22) 출원일자 2013년02월14일
심사청구일자 2018년02월12일
(65) 공개번호 10-2013-0093565
(43) 공개일자 2013년08월22일
(30) 우선권주장
13/396,582 2012년02월14일 미국(US)
(56) 선행기술조사문헌
US20110131423 A1*
(뒷면에 계속)

(73) 특허권자
야누스 테크놀로지스, 인코퍼레이티드
미국 캘리포니아 94019 하프 문 베이 메인 스트리트 795
(72) 발명자
첸-후아 왕
대만 타이페이 시티 스린 디스트릭트 신안 로드 119
소핀 라스킨
미국 캘리포니아 94022 로스 알토스 넬라 래인 757
레오니드 로젠보임
미국 캘리포니아 95032 로스 가토스 윌로우 드라이브 15280
(74) 대리인
특허법인(유)화우

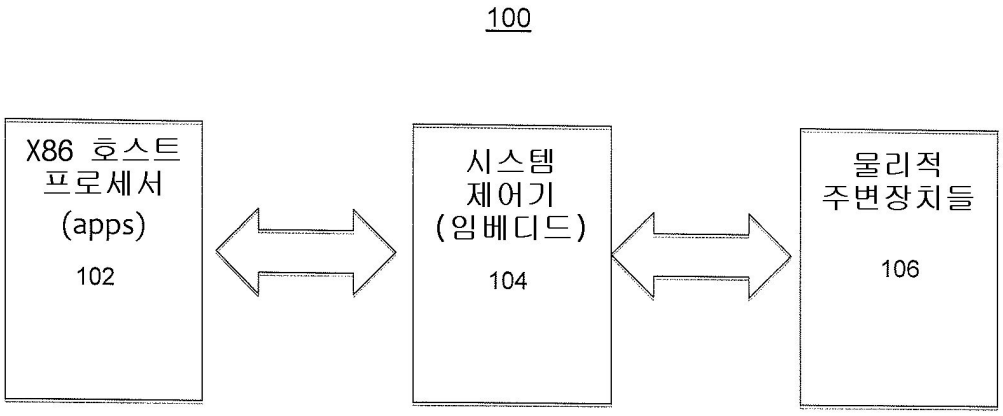
전체 청구항 수 : 총 14 항 심사관 : 정성훈

(54) 발명의 명칭 **보안-강화 컴퓨터 시스템 및 방법**

(57) 요 약

일반적으로, 본 발명은 보강된 데이터 보안을 위해 설계된 컴퓨터를 제공한다. 실시예들에서, 그 구조는 각각 자체 처리 유닛과 메모리를 갖는 2 개의 하위-시스템과, 상기 2 개의 하위-시스템과 외부 세계를 상호연결하는 정의된 세트의 인터페이스들을 포함한다. 하나의 하위-시스템은 컴퓨터 어플리케이션들을 실행하기에 친숙한 환경을 제공하도록 설계된다. 다른 하나의 하위-시스템은 입력 디바이스 및 출력 디바이스를 통해 제 1 하위-시스템과 사용자 간에 보안 브릿지를 제공하도록 설계된다.

대 표 도



(56) 선행기술조사문헌
US20110199308 A1*
US20080263658 A1*
US07987497 B1*
WO2012003486 A1*
*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

삭제

청구항 2

삭제

청구항 3

컴퓨터 시스템에 있어서,

어플리케이션들을 실행하도록 구성되는 제 1 프로세서를 포함하는 제 1 하위시스템;

보안 펌웨어를 실행하도록 구성되는 별개의 제 2 프로세서를 포함하는 제 2 하위시스템; 및

상기 제 2 하위시스템에 연결되는 주변장치들을 포함하며,

상기 어플리케이션들에 의한 상기 주변장치들에 대한 액세스는 상기 제 2 프로세서에서 실행되는 상기 보안 펌웨어에 의해 제어되고, 상기 보안 펌웨어는 상기 제 1 하위 시스템의 대응되는 주변장치 연결들을 에뮬레이트하고,

상기 주변장치들은 비디오 디스플레이를 포함하며, 상기 컴퓨터 시스템은:

상기 제 1 하위 시스템에 연결되는 제 1 입력부, 상기 제 2 하위시스템에 연결되는 제 2 입력부, 및 상기 비디오 디스플레이에 연결되는 출력부를 갖는 비디오 멀티플렉서를 더 포함하며,

상기 출력부를 구동하기 위한 상기 제 1 입력부 및 상기 제 2 입력부로부터의 콘텐츠의 선택은 상기 제 2 프로세서에서 실행되는 상기 보안 펌웨어에 의하여 제어되고,

상기 주변장치들은 키보드를 더 포함하고, 상기 보안 펌웨어는 스타트업 시퀀스를 포함하고,

상기 키보드 및 상기 비디오 디스플레이를 위한 출력은 상기 제 2 프로세서에 의해 독점적으로(exclusively) 제어되며,

상기 제 1 프로세서에 의한 상기 키보드 및 상기 비디오 디스플레이에 대한 액세스는 방지되는 컴퓨터 시스템.

청구항 4

제 3 항에 있어서,

상기 주변장치들은 상기 제 1 하위시스템을 위한 운영 시스템 및 어플리케이션 소프트웨어를 포함하는 디스크 드라이브를 포함하고, 상기 컴퓨터 시스템은:

상기 보안 펌웨어에 의하여 유지되는 에뮬레이트된 디스크 드라이브를 더 포함하며,

상기 제 1 프로세서에 의한 상기 디스크 드라이브의 어플리케이션 소프트웨어 및 운영 시스템에 대한 액세스는 상기 에뮬레이트된 디스크 드라이브를 통해 제어되는 컴퓨터 시스템.

청구항 5

제 4 항에 있어서,

상기 보안 펌웨어는 상기 에뮬레이트된 디스크 드라이브의 1 이상의 특정-시간 이미지들(time-specific images)을 유지하는 컴퓨터 시스템.

청구항 6

제 4 항에 있어서,

상기 디스크 드라이브는 SSD(solid state disk drive)를 포함하는 컴퓨터 시스템.

청구항 7

제 4 항에 있어서,

상기 보안 펌웨어는 상기 디스크 드라이브의 모든 데이터를 암호화하고,

상기 암호화를 위한 키들은 상기 보안 펌웨어에 의해 독점적으로 유지되는 컴퓨터 시스템.

청구항 8

제 3 항에 있어서,

상기 주변장치들은 네트워크 연결부를 포함하며,

상기 보안 펌웨어는 상기 제 1 하위시스템에 의한 외부 네트워크에 대한 액서스를 제어하는 컴퓨터 시스템.

청구항 9

제 8 항에 있어서,

상기 제 1 하위시스템과 상기 외부 네트워크 간의 모든 통신들을 위한, 상기 보안 펌웨어 의해 유지되는 VPN 터널을 더 포함하는 컴퓨터 시스템.

청구항 10

삭제

청구항 11

삭제

청구항 12

컴퓨터 시스템을 보안설정하는 방법에 있어서,

어플리케이션들을 실행하기 위한 제 1 프로세서를 포함하는 상기 컴퓨터 시스템의 제 1 하위시스템을 구성하는 단계;

보안 펌웨어를 실행하기 위한 제 2 개별 프로세서를 포함하는 상기 컴퓨터 시스템의 제 2 하위시스템을 구성하는 단계;

주변장치들을 상기 제 2 하위시스템에 연결하는 단계; 및

상기 제 1 하위시스템의 대응되는 주변장치 연결들을 에뮬레이트하는 상기 제 2 프로세서에서 실행되는 상기 보안 펌웨어를 이용하여 상기 어플리케이션들에 의한 상기 주변장치에 대한 액세스를 제어하는 단계를 포함하고,

상기 주변장치들은 비디오 디스플레이를 포함하며, 상기 방법은:

상기 컴퓨터 시스템에 비디오 멀티플렉서를 제공하는 단계;

상기 비디오 멀티플렉서의 제 1 입력부를 상기 제 1 하위시스템에 연결하는 단계;

상기 비디오 멀티플렉서의 제 2 입력부를 상기 제 2 하위시스템에 연결하는 단계;

상기 비디오 멀티플렉서의 출력부를 상기 비디오 디스플레이에 연결하는 단계; 및

상기 제 2 프로세서에서 실행되는 상기 보안 펌웨어를 이용하여 상기 출력부를 구동하기 위한 상기 제 1 입력부 및 상기 제 2 입력부로부터의 콘텐츠의 선택을 제어하는 단계를 더 포함하고,

상기 주변장치들은 키보드를 더 포함하며, 상기 방법은:

상기 보안 펌웨어에 의해 실행되는 스타트업 시퀀스를 더 포함하고,

상기 키보드 및 상기 비디오 디스플레이를 위한 출력은 상기 제 2 프로세서에 의해 독점적으로 제어되며,

상기 제 1 프로세서에 의한 상기 키보드 및 상기 비디오 디스플레이에 대한 액세스는 방지되는 방법.

청구항 13

제 12 항에 있어서,

상기 주변장치들은 상기 제 1 하위시스템을 위한 어플리케이션 소프트웨어 및 운영 시스템을 포함하는 디스크 드라이브를 포함하고, 상기 방법은:

상기 보안 펌웨어를 이용하여 에뮬레이트된 디스크 드라이브를 유지시키는 단계를 더 포함하며,

상기 제 1 프로세서에 의한 상기 디스크 드라이브의 어플리케이션 소프트웨어 및 상기 운영 시스템에 대한 액세스는 상기 에뮬레이트된 디스크 드라이브를 통해 제어되는 방법.

청구항 14

제 13 항에 있어서,

상기 보안 펌웨어를 이용하여 상기 에뮬레이트된 디스크 드라이브의 1 이상의 특정-시간 이미지들을 유지시키는 단계를 더 포함하는 방법.

청구항 15

제 13 항에 있어서,

상기 디스크 드라이브는 SSD를 포함하는 방법.

청구항 16

제 13 항에 있어서,

상기 보안 펌웨어를 이용하여 상기 디스크 드라이브의 모든 데이터를 암호화하는 단계; 및

상기 보안 펌웨어에 의해 상기 암호화를 위한 키들을 독점적으로 유지시키는 단계를 더 포함하는 방법.

청구항 17

제 12 항에 있어서,

상기 주변장치들은 네트워크 연결부를 포함하며, 상기 방법은:

상기 보안 펌웨어를 이용하여 상기 제 1 하위시스템에 의한 외부 네트워크에 대한 모든 액세스들을 제어하는 단계를 더 포함하는 방법.

청구항 18

제 17 항에 있어서,

상기 보안 펌웨어를 이용하여, 상기 제 1 하위시스템과 상기 외부 네트워크 간의 모든 통신들을 위한 VPN 터널을 유지시키는 단계를 더 포함하는 방법.

청구항 19

삭제

청구항 20

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 일반적으로 컴퓨터에 관한 것이며, 구체적으로 정보 보안의 중요성이 증대된 기업 및 정보 기관들에서 이용되는 컴퓨터에 관한 것이다.

배 경 기 술

[0002] 전형적으로, 개인 및 기업 데이터 보안 기능들은 적정가격(affordability)으로 설계되는 양산형(consumer-grade) 개인용 컴퓨터들과 기본적으로 동일한 하드웨어 구조 위에 애드온 소프트웨어 모듈(add-on software modules) 형태로 구현된다. 때로는 사용자 진위를 보다 엄격하게 인증하기 위한 목적으로 특정-보안(security-specific) 애드온 하드웨어 모듈(예를 들어, 스마트카드, 바이오메트릭스)도 구현될 수 있다. 전형적으로, 보안 기능의 벌크는, 때로 운영 시스템에 통합되지만 메모리 내에 속해 있으며 여타 소프트웨어 어플리케이션처럼 실행되는 애드온 소프트웨어 구성요소들로서 구현된다.

[0003] 이 전형적 접근법에 의한 심각한 문제점은, 보안 기능이 소프트웨어로 구현될 경우, 다수의 상이한 방식에서 보안에 위협을 받을 수 있다는 점이다. 컴퓨터를 운영하는 정상적인 과정에서, 사용자는 때로 몇몇 소프트웨어 구성요소들을 부가하거나 또는 수정한다 - 이는 소프트웨어 구성요소들을 부가 및 교체하여, 일반적 연산 구조에 폭넓은 임무(tasks) 및 할당(assignment)에 있어서의 융통성 및 이용가능성을 부여한다. 이는 공격자(attacker)에게도 소프트웨어 모듈들을 수정하거나 부가하여 컴퓨터 시스템의 보안을 위협할 기회의 창(window)를 열어주는 것과 같은 역할을 한다.

[0004] 새로운 소프트웨어 구성요소가 도입될 때, 공격을 감행하도록 의도된 기능을 포함하거나, 또는 공격을 원활히 하기 위해 외부에서 부당하게 이용될 수 있는 프로그래밍 에러들을 포함할 우려가 존재한다. 또한, 보안 소프트웨어는 어플리케이션 소프트웨어와 유사하게 보급 및 설치되기 때문에, 그 또한 같은 위험에 취약할 수 있다.

[0005] 전형적 범용 컴퓨터에서, 전체 RAM(random-access memory)은 물리적으로 프로세서에 의해 액세스되거나 시스템이 복수의 프로세서들을 포함할 경우 모든 프로세서들에 의해 액세스될 수 있는 단일 대형 유니폼 뱅크(single large uniform bank)로 구성된다. 메모리 액세스의 균일성은, RAM의 사용에 있어 최고의 융통성 - 이는 컴퓨터의 결정적 리소스들 중 하나임 - 을 제공하며, 운영 시스템 및 어플리케이션 소프트웨어에 의한 RAM의 최적의 활용성을 이끌어낸다. 균일한 RAM 구조는 비용 효율성에 있어 효과적인 한편, 동시에 실행되는 프로그램들이 서로의 메모리 영역들, 또는 운영 시스템이나 그 구성요소들에 의해 점유되는 메모리에 액세스할 수 있다는 것을 또한 의미한다. 이와 같이, 균일한 RAM 구조의 이러한 특징은 컴퓨터의 보안을 위협하는 가장 많이 이용되는 수단(the most used vehicle)이 되어 왔다.

[0006] 또한, 현대의 컴퓨터 시스템들은 "가상 메모리"라 칭해지는 메커니즘을 채용하고 있으며, 여기서 메모리 관리 유닛(memory management unit)("MMU")이라 칭해지는 프로세서에 임베디드되는 하드웨어 구성요소가 메모리 어드레스 전환의 기능을 수행한다. 가상 메모리의 부가는 RAM이 섹션들로 분할되도록 하며, 각각의 섹션은 특정 소프트웨어 구성요소 또는 그들의 그룹에 할당된다. 또한, 가상 메모리는 상이한 소프트웨어 구성요소 또는 운영 시스템에 속하는 메모리에 대한 의도하지 않은 액세스를 방지한다. 가상 메모리 메커니즘은 과도한 소프트웨어의 거동이 전체적으로 시스템의 안정성에 충격을 주는 것을 방지하는 데 상당히 효과적임이 증명되었으나, 이것이 악의적인 사보타주(malicious sabotage)를 방지하기 위해 의도된 것은 아니며, 모든 운영 시스템에서는 진단 목적으로 계획된 MMU에 의하여 제공되는 보호책들을 회피하기 위한 문서 메커니즘(documented mechanism)이 존재한다. 이들 메커니즘은 흔히 부당하게 이용되어 컴퓨터 및 그 내부에 포함되는 데이터의 보안을 위협한다.

[0007] 상승된 보안 레벨을 달성하기 위한 통상적인 한가지 접근법에서는, 보안 메커니즘의 몇몇 부분이 개별적 전용 하드웨어 모듈에서 구현되며, 이는 추가적인 부정 조작 방지(tamper-resistant) 특징부로 설계되어 잠재적인 불청객(intruder)에게 난이도를 부가시킨다. 아마도 하드웨어-보강 컴퓨터 보안 특징부들 중 공개되지 않은 최초의 예들 중 하나는 IBM HSM(Hardware Security Module)이었으며, 이는 사무용 금고(office safe)와 유사하게 설계된 강건한 외장(rugged enclosure) 내에 구성된 자체 메모리 및 저장 시스템을 갖는 소형 독립 컴퓨터(stand-alone computer)이었다. 은행 카드들의 개인 식별 번호들은 클리어-텍스트 폼(clear-text form)으로 은행 직원들조차도 이들 코드들에 접근하지 못하도록 HSM 내에 저장되어 있었다. ATM(automated teller machine)이 카드 소지자(card holder)의 식별을 증명할 필요가 있을 때, 암호화된 시도-응답 시퀀스(cryptographic challenge-response sequence)가 개시되어 PIN이 통신 링크에 걸쳐 그대로 전달(transmitted verbatim over the communication links)되지 않고 HSM이 입증(verification) 과정을 안전하게 수행한다.

[0008] (GSM 기반의) 세계 표준 휴대폰 시스템의 스마트-카드 접근식 사용자-인증 메커니즘(smart-card approach user-authentication mechanism)은 하드웨어 보안 모듈이 손톱 크기로 소형화되고, 각각의 사용자에게 이러한 디바이

스가 제공된다는 점을 제외하고 유사한 메커니즘을 갖는다. SIM 카드 구조는 임베디드된 메모리 칩을 손상시키지 않고서는 분해하기(disassemble) 어렵게 한다.

[0009] 다른 종래의 접근법으로는 현재 제조되는 몇몇의 개인용 컴퓨터들 내에 구성되는 트러스터 플랫폼 모듈(Truster Platform Module)이 있다. TPM은 제한된 액세스를 가진 작은 칩으로 몇몇 보안-관련 식별 정보와 몇몇 암호 키(encryption key)들을 포함한다는 점에서 SIM 카드와 다소 유사하다. TPM의 핵심은 공격자가 이 식별 정보를 수정해서 컴퓨터 및 그 사용자를 악의적으로 식별하여 시스템 내의 어딘가에 존재하는 보안 메커니즘을 피하려는 행위를 방지하는 것이다. 하지만, 그것의 불리한 면은 TPM 내에 포함된 키와 숫자들이 보호책의 단지 하나의 부분에 불과하며, 보호책의 나머지 부분들은 운영 시스템 및 어플리케이션 소프트웨어 구성요소들에서 전형적으로 구현된다는 것이다. 따라서, TPM은 추가적인 겹겹의 보호책을 제공하여, 미인증 사용자에 의해 몇몇 키 보안-관련 정보 표시가 수정되는 것을 불가능하게 한다. 하지만, TPM은 시스템 소프트웨어의 다른 부분들과 그것의 통신에 있어 상당한 취약성이 남아 있어, 성공적인 공격을 위해 악의적으로 사용될 수 있다.

발명의 내용

해결하려는 과제

[0010] 따라서, 컴퓨터 시스템 보안에 있어서의 개선된 접근법들에 대한 필요성이 있다.

과제의 해결 수단

[0011] 본 발명은 보강된 데이터 보안을 위해 설계된 컴퓨터 구조에 관한 것이다. 실시예들에서, 상기 구조는 2 개의 하위-시스템들 - 각각 자체 프로세싱 유닛 및 메모리들을 가짐 - , 및 2 개의 하위-시스템들과 외부 세계를 상호연결하는 분명하게 정의된 세트의 인터페이스들을 포함한다.

[0012] 특정 실시형태들에 따르면, 상기 2 개의 하위-시스템들 중 하나는 오늘날의 개인용 컴퓨터의 대부분을 차지하는 x86과 같은 대중적 프로세서 구조 주위에 구성되며, 어플리케이션-프로세서 하위-시스템으로서 지정된다. 이 프로세서 구조는 광범위한 어플리케이션 소프트웨어 및 그를 위해 이용가능한 운영 시스템들에 대해 선택되며, 사용자가 그들이 선택한 어플리케이션 소프트웨어를 설치하는 데 있어 융통성을 극대화하는 것을 목적으로 한다. x86 구조 주위에 설계되는 종래의 개인용 컴퓨터와는 달리, 이 어플리케이션-프로세서의 모든 주변장치 연결부들은 실제의 외부 또는 내부 주변장치들 대신 다른 하위시스템으로 라우팅된다. 따라서, x86으로 실행될 수 있는 소프트웨어는 사실상 제한되지 않으나, 이 소프트웨어나 그것의 데이터에 대한 외부의 액세스는 이들 어플리케이션들 및 그들의 데이터를 안전하게 유지시키는 데 필요한 보호책들을 부여하는 전용 시스템-프로세서 하위-시스템에 의해 엄격하게 제어된다.

[0013] 특정한 추가 실시형태들에 따르면, 시스템-프로세서로서 지정된 다른 하위-시스템은 기본적으로 임베디드 시스템이다. 이는, 프로세서와 함께 제공되는 임베디드 소프트웨어 시스템을 실행시키고, 어떠한 상황 하에서도 컴퓨터의 최종-사용자에 의해 수정될 수 없으며, 대신에 펌웨어로서 참조되어야 한다. 임베디드 시스템이게 되면, 최종-사용자나 여느 제 3의 개발자 모두 그 소프트웨어 구성요소들을 기록하거나 수정할 수 없기 때문에 하위-프로세서 모듈의 프로세서 구조의 세부사항들(specifics)은 중요하지 않다. 시스템-프로세서는 기본적으로 외부 세계와 자체 하드웨어 하위시스템에서 운영되는 본질적으로 불안정한 어플리케이션 소프트웨어 환경(inherently insecure application software environment) 사이의 "브릿지(bridge)"로서의 역할을 한다. 실시예들에서, 시스템-프로세서는 주변장치 연결부의 각각의 타입을 위한 2 개의 포트들을 가지며, 그 중 하나는 실제 주변장치에 연결되고, 다른 하나는 어플리케이션-프로세서 하위-시스템에 연결된다. 펌웨어는 시스템-프로세서 하드웨어와 함께 어플리케이션-프로세서 하위시스템의 이득을 위한 각 타입의 주변장치 디바이스를 에뮬레이트(emulate)하는 한편, 지원 타입의 주변장치들 각각에 대해 적절하며 어플리케이션 소프트웨어와 그 데이터에 대한 최고의 보호 수준을 항시 유지하는 데 필요한 규칙들과 메커니즘들의 세트를 부여한다. 내부 및 외부 주변장치들 모두는 시스템-프로세서에 연결되고 주변장치 에뮬레이션 펌웨어 기능(peripheral emulation firmware functionality)에 의해 이용된다.

[0014] 이들 및 다른 실시형태에 따르면, 본 발명의 실시예들에 따른 컴퓨터 시스템은, 어플리케이션들을 실행하도록 구성되는 제 1 프로세서를 포함하는 제 1 하위시스템; 보안 펌웨어를 실행하도록 구성되는 제 2 개별 프로세서를 포함하는 제 2 하위시스템; 및 상기 제 2 하위시스템에 연결되는 주변장치들을 포함하며, 상기 어플리케이션들에 의한 상기 주변장치들에 대한 액세스는 상기 제 1 하위시스템의 대응되는 주변장치 연결들을 에뮬레이트하는 상기 제 2 프로세서에서 실행되는 상기 보안 펌웨어에 의해 제어된다.

[0015] 또한 이들 및 다른 실시형태들에 따르면, 본 발명의 실시예들에 따른 컴퓨터 시스템을 보안설정하는 방법은, 어플리케이션들을 실행하기 위한 제 1 프로세서를 포함하는 상기 컴퓨터 시스템의 제 1 하위시스템을 구성하는 단계; 보안 펌웨어를 실행하기 위한 제 2 개별 프로세서를 포함하는 상기 컴퓨터 시스템의 제 2 하위시스템을 구성하는 단계; 주변장치들을 상기 제 2 하위시스템에 연결하는 단계; 및 상기 제 1 하위시스템의 대응되는 주변장치 연결들을 에뮬레이트하는 상기 제 2 프로세서에서 실행되는 상기 보안 펌웨어를 이용하여 상기 어플리케이션들에 의한 상기 주변장치에 대한 액세스를 제어하는 단계를 포함한다.

[0016] 이들 및 다른 실시예들의 추가 실시형태들에서, 본 발명의 실시예들에 따른 시스템은 독립 컴퓨터 시스템을 포함하며, 상기 독립 컴퓨터 시스템은, 어플리케이션들을 실행하도록 구성되는 제 1 프로세서를 포함하는 제 1 하위시스템; 보안 펌웨어를 실행하도록 구성되는 제 2 개별 프로세서를 포함하는 제 2 하위시스템; 및 상기 독립 컴퓨터 시스템을 제어하는 기구(organization)에 의하여 호스팅되는 보안 인트라넷을 포함하며, 상기 어플리케이션들에 의한 상기 보안 인트라넷에 대한 액세스는 상기 제 1 하위시스템의 대응되는 물리적 네트워크 연결을 에뮬레이트하는 상기 제 2 프로세서에서 실행되는 상기 보안 펌웨어에 의하여 제어된다.

도면의 간단한 설명

[0017] 당업자라면 본 발명의 이들 및 다른 실시형태들과 특징들이 첨부 도면과 연계되는 본 발명의 특정 실시예의 후속 설명부를 참조하면 명확히 이해될 것이다.

도 1은 본 발명의 실시예들에 따른 보안 컴퓨터 구조를 예시한 톱-레벨 도(top-level diagram),

도 2는 본 발명의 원리들에 따른 컴퓨터 시스템의 기능적 블록도,

도 3은 본 발명의 실시예들에 따른 컴퓨터 시스템의 디스플레이 및 스타트업 기능을 보안설정하는 예시적 실시형태들을 예시한 도,

도 4는 본 발명의 실시예들에 따른 컴퓨터 시스템의 네트워크 연결 기능을 보안설정하는 예시적 실시형태들을 예시한 도,

도 5는 본 발명의 실시예들에 따른 컴퓨터 시스템의 주변장치들에 대한 어플리케이션 액세스를 보안설정하는 예시적 실시형태들을 예시한 도,

도 6은 본 발명의 실시예들에 따른 컴퓨터 시스템을 보안설정하는 예시적 프로세스를 예시한 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0018] 이제부터, 당업자들이 본 발명을 실행할 수 있도록 본 발명을 나타내는 예시들로서 제공되는 도면들을 참조하여 본 발명에 대해 상세히 설명될 것이다. 유의할 것으로, 이후의 도면 및 예시들은 본 발명의 범위를 단일 실시예로 제한하려는 것이 아니며, 설명되거나 예시된 요소들 중 일부 또는 모두를 상호교환하는 방식의 다른 실시예들도 가능하다. 또한, 본 발명의 특정 요소들이 알려진 구성요소들을 이용하여 부분적으로 또는 전적으로 구현될 수 있을 경우, 이러한 알려진 구성요소들 중 본 발명의 이해를 돕는데 필요한 부분들에 대해서만 설명될 것이며, 이러한 알려진 구성요소들 중 다른 부분들은 본 발명의 해석을 방해하지 않기 위해 생략될 것이다. 소프트웨어로 구현되는 것으로 설명되는 실시예들은 그것으로만 제한되지 않으며, 본 명세서에서 구체적으로 설명되지 않는 한 당업자들에게 명백한 바와 같이 하드웨어나, 소프트웨어와 하드웨어의 조합으로 구현되는 실시예들을 포함하여 이루어질 수 있다. 본 명세서에서, 단일 구성요소를 나타내는 실시예는 제한적인 것으로 간주되어서는 안되며, 오히려 본 발명은 본 명세서에서 명확하게 언급되지 않는 한 복수의 동일한 구성요소를 포함하는 다른 실시예들까지 포괄하도록 되어 있다. 또한, 출원인들은 이와 같이 명확하게 기술되지 않는 한 본 명세서의 어떤 용어나 청구범위들도 비통상적이거나 특수한 의미를 설명하려 의도하지 않았다. 나아가, 본 발명은 예시의 방식으로 본 명세서에서 언급되는 알려진 구성요소들에 대한 현재와 미래의 알려진 등가적요소들까지 포괄한다.

[0019] 일반적으로, 본 발명은 보강된 데이터 보안을 위해 설계된 컴퓨터 구조를 제공한다. 실시예들에서, 상기 구조는 2 개의 하위-시스템들 - 각각 자체 프로세싱 유닛 및 메모리들을 가짐 - , 및 상기 2 개의 하위-시스템들과 외부 세계를 상호연결하는 정의된 세트의 인터페이스들을 포함한다. 하나의 하위-시스템은 컴퓨터 어플리케이션들을 실행하는 데 친숙한 환경을 제공하도록 설계된다. 다른 하위-시스템은 입력 및 출력 디바이스들을 통해 제 1 하위-시스템과 사용자들 간의 보안 브릿지(secure bridge)를 제공하도록 설계된다.

[0020] 도 1은 본 발명의 실시형태들에 따른 예시적 시스템 구조물(100)을 예시한 블록도이다.

[0021] 도 1에 도시된 바와 같이, 2 개의 하위-시스템들 중 하나는 어플리케이션-프로세서 하위시스템(102)으로서 지정된 오늘날의 개인용 컴퓨터의 대부분을 점하는 x86과 같은 대중적 마이크로프로세서 구조물(popular microprocessor architecture) 주위에 구성되는 것이 바람직하다. 이 구조는 광범위한 어플리케이션 소프트웨어 및 그를 위해 이용가능한 운영 시스템들에 대해 선택되며, 사용자가 선택한 어플리케이션 소프트웨어를 설치하는 데 있어서의 사용자의 융통성을 극대화하는 것을 목적으로 한다. 또한 x86 구조 주위에 설계되는 종래의 개인용 컴퓨터와는 달리, 이 어플리케이션-프로세서는 그것의 모든 주변장치 연결부들이 실제의 외부 또는 내부 주변장치들 대신 다른 하위시스템으로 라우팅된다. 따라서, x86에서 실행될 수 있는 소프트웨어는 사실상 제한되지 않는 한편, 이 소프트웨어 또는 그것의 데이터에 대한 외부의 액세스가 이들 어플리케이션들 및 그들의 데이터를 안전하게 유지시키는 데 필요한 보호책들을 부여하는 전용 하위-프로세서 하위-시스템에 의해 엄격하게 제어된다.

[0022] 이러한 다른 하위-시스템, 즉 시스템 프로세서(104)는 임베디드 시스템(embedded system)인 것이 바람직하다. 이와 같이, 이는 프로세서와 함께 제공되는 지정된 소프트웨어 시스템을 실행하고, 어떠한 상황 하에서도 컴퓨터의 최종-사용자에 의해 수정될 수 없으며, 대신에 펌웨어로서 참조되어야 한다. 임베디드 시스템이게 되면, 최종-사용자나 여느 제 3의 개발자 모두 그 소프트웨어 구성요소들을 기록하거나 수정하도록 허용되지 않기 때문에 시스템-프로세서 모듈의 프로세서 구조의 세부사항들은 중요하지 않다. 시스템-프로세서(104)는 기본적으로 외부 세계와 자체 하드웨어 하위시스템(102)에서 운영되는 본질적으로 불안정한 어플리케이션 소프트웨어 환경 사이의 "브릿지"로서의 역할을 한다.

[0023] 통상적으로, 주변장치들(106)은 컴퓨터 사용자와 시스템(100)의 기능들 간의 인터페이스를 제공하는 여하한의 타입의 디바이스를 포함한다. 이러한 디바이스에는 디스플레이, 스피커, 프린터 등과 같은 출력 디바이스, 및 키보드, 마우스, 터치패드, 터치스크린 등과 같은 입력 디바이스가 포함될 수 있다. 주변장치들(106)의 수 및 타입은 어플리케이션 프로세서(102 및 104)를 하우징하는 디바이스의 특정 폼 팩터(form factor)에 따라 정해질 수 있다. 예를 들어, 상기 폼 팩터가 종래의 데스크탑 컴퓨터의 것인 본 발명의 실시예에서는, 주변장치들(106)이 외부에 부착되는 디스플레이, 키보드 및 마우스를 포함할 수 있다. 폼 팩터가 종래의 노트북 컴퓨터의 것인 경우, 주변장치들(106)에는 통합 디스플레이, 키보드 및 터치패드가 포함될 수 있다. 폼 팩터가 컴퓨터 또는 스마트 폰의 것인 경우, 주변장치들(106)에는 통합 디스플레이/터치스크린이 포함될 수 있다. 시스템(100)을 위한 상이한 타입의 폼 팩터들 간의 주변장치들(106)은 반드시 상호 배타적인 것도 아니며 그들이 시간에 걸쳐 일정한 것도 아니라는 데 유의하여야 한다. 예를 들어, 많은 종래의 터치패드 컴퓨터 시스템들은 선택적 개별 키보드들 및 마우스들(예를 들어, USB 또는 블루투스를 통해 연결됨)을 이용하여 작동될 수 있다. 마찬가지로, 많은 종래의 데스크탑 컴퓨터 시스템들은 선택적 터치스크린들 또는 음성-명령 디바이스들을 이용하여 작동될 수 있다.

[0024] 몇몇 실시예들에서, 시스템(100)은 통상적 컴퓨터 시스템으로 나타나도록 설계되며, 시스템 프로세서(102)의 추가적인 보안 특징부들이 내부에 임베디드되어 평범한 관찰자(casual observer)에겐 쉽게 띄지 않는다. 예를 들어, 시스템(100)은 통상적인 랩탑 컴퓨터로서 나타날 수 있으며, 통상적인 접이식 디스플레이, 빌트-인 키보드, 스피커 및 위치 결정 디바이스(pointing device)를 갖는다. 다른 가능한 실시예들에서, 시스템 프로세서(102) 및 어플리케이션 프로세서(104)는 개별적으로 하우징되거나, 함께 하우징되거나, 또는 주변장치들(106) 중 특정 주변장치와 개별적으로 하우징된다. 하지만, 추가적인 보안 실시형태들에 대해, 프로세서 하위-시스템(102 및 104)은 동일한 외장 내에, 심지어는 같은 회로 보드 상에, 그리고 아마도 심지어는 같은 ASIC, SOC 또는 FPGA 상의 2 개의 개별 프로세서 코어로서 가능한 한 많이 통합되는 것이 바람직하다. 예를 들어, 본 발명자들은 이들 하위시스템들 간의 어떠한 형태의 노출된 상호연결들(exposed interconnections)도 공격자에 의해 잠재적으로 악의적으로 이용될 수 있다는 것을 인식하였다. 따라서, 이러한 상호연결부들은 (예를 들어 같은 집적 회로 및/또는 회로 보드 내에) 가능한 한 액세스가 어렵게 만들어지는 것이 바람직하다. 시스템 프로세서(104)에 대한 주변장치(106)의 연결부들과 관련하여, 이들은 시스템(100)의 특정 폼 팩터에 따라 통합적이거나 개별적일 수 있다.

[0025] 어떤 주어진 시스템(100)의 모든 주변장치들이 시스템 프로세서(104)에 의하여 제어되는 액세스를 가질 필요는 없다는 데 유의하여야 한다. 하지만, 통상적으로 시스템의 적어도 가장 유용하거나 중요한 주변장치들, 예컨대 키보드 및 마우스와 같은 입력 디바이스들과, 디스플레이와 같은 가장 유용한 출력 디바이스들은 제어된다. 이와 관련하여, 본 발명자들은, 이러한 주변장치들은 통상적으로 상기 입력/출력 디바이스들 - 이들을 통해 어플리케이션 프로세서(102)의 특정한 구현이 작업자나 다른 컴퓨터와 [예를 들어, 네트워크나 다른 통신 링크를 통

해] 접속(interface)하여 그것의 작업들 및/또는 데이터에 액세스하거나 그를 제어할 수 있음 - 포함하는 것을 인식하였다. 이와 같이, 어플리케이션 프로세서(102)의 데이터 및 명령들(operations)을 외부 세계에 실질적으로 노출시키는 모든 주변장치들(106)은 시스템 프로세서(104)를 통해 라우팅되는 것이 바람직하다. 따라서, "주변장치"라는 용어는 실제 주변장치와 프로세서를 주변장치에 연결하는 연결부(예를 들어, 포트) 둘 모두를 포괄하는 것으로 해석되어야 한다. 따라서, 시스템 프로세서(104)는 이들 통신시에 최대 보안의 물리적 지점(most secure physical point)에서 주변장치들(106)과 프로세서(102) 간의 통신을 차단하는 것이 바람직하다. 부언하면, 어플리케이션 프로세서(102)는 그것에 직접 또는 그것의 작업 환경에 직접적으로 연결되거나, 또는 외부에 노출되거나 외부에서 액세스가능한 연결부들을 통해 연결되는 어떠한 중요한 주변장치도 갖지 않으며, 오히려 이들 연결부들은 시스템 프로세서(104)를 통해 라우팅되거나 그에 의해 제어된다.

[0026] 또한, 타입에 따라서, 주변장치들(106)은 통상적으로 어플리케이션 프로세서(102) 및 시스템 프로세서(104)를 하우징하는 디바이스에 대해 내부나 외부에 있을 수 있다는 데 유의하여야 한다. 보다 상세히 후술되겠지만 제한적인 것으로 해석해서는 안되는 시스템(100)의 바람직한 일 실시예로는 데스크탑 또는 노트북 컴퓨터의 폼 팩터의 디바이스의 실시예가 있다. 이러한 실시예에서, 주변장치들(106)은 부착 디스플레이(attached display), 키보드 및 위치 결정 디바이스(예를 들어, 터치패드 및/또는 스틱 마우스), 및 내장 스피커 및 무선 모뎀(예를 들어, 802.11 a/b/g/n)을 포함한다. 이러한 실시예에서의 주변장치들(106)은 종래의 잭(jack) 또는 인터페이스들, 예컨대 USB, RJ-45, Firewire, eSATA, VGA, HDMI, DVI, DisplayPort 및 MiniDisplayPort를 포함하는 시스템(100)의 대응되는 잭을 통해 부착되는 입력 또는 출력 외부 디바이스를 더 포함할 수 있다. 당업자라면, 본 예시들을 숙지한 후에 보다 적은 수나 추가적인 타입의 인터페이스들 및/또는 주변장치들로 본 발명을 구현할 방법에 대해 인지할 수 있을 것이다.

[0027] 통상적으로, 시스템-프로세서(104)는 주변장치 연결부의 각각의 타입에 대해 2 개의 연결부들을 가지며, 그들 중 하나는 실제 주변장치(106)에 대한 연결부이고 다른 하나는 어플리케이션-프로세서 하위-시스템(102)에 대한 연결부이다. 보다 상세히 후술되는 바와 같이, 시스템-프로세서(104) 하드웨어와 함께 제공되는 펌웨어는 어플리케이션-프로세서 하위시스템(102)의 이득을 위해 시스템 프로세서(104)에 실제로 연결되는 각 타입의 주변장치(106)를 에뮬레이트한다. 또한, 펌웨어는 지원 타입의 주변장치들 각각에 대해 적절하며 어플리케이션 소프트웨어 및 그것의 데이터에 대한 최고 수준의 보호를 항시 유지시키는 데 필요한 규칙들 및 메커니즘들의 세트를 부여하는 것이 바람직하다. 실시예들에서, 모든 내부 및 외부 주변장치들(106)은 시스템-프로세서(104)에 연결되며 주변장치 에뮬레이션 펌웨어 기능에 의해 이용된다.

[0028] 도 1에 상세히 도시되진 않았으나, 어플리케이션 프로세서(102) 및 시스템 제어기(104)는 메모리, 메모리 및 I/O 어드레싱 스페이스(addressing space), 운영 시스템 소프트웨어, 어플리케이션 소프트웨어, 그래픽 프로세서, 사운드 프로세서 및 프로세서 버스들을 더 포함할 수 있다. 예를 들어, 시스템(100)의 폼 팩터가 데스크탑 또는 노트북 컴퓨터인 경우, 시스템(100)은 PCI 버스, 운영 시스템, 예컨대 Windows 7, 및 연관된 BIOS 소프트웨어 및 어플리케이션 소프트웨어, 예컨대 Windows Office를 저장하는 RAM 및 ROM 메모리와 같은 종래의 개인용 컴퓨터 구성요소들을 포함할 수 있다. 시스템(100)은 이러한 종래의 개인용 컴퓨터 구성요소들, 예컨대 XGA 그래픽 프로세서[예를 들어, Intel x86, AMD 통합 그래픽(AMD integrated graphics) 또는 외부 프로세서, 예컨대 nVidia에 의해 제공되는 것들], 5.1 오디오 프로세서, USB 입력부 및 출력부(USB inputs and outputs), 이더넷 인터페이스, 직렬/병렬 인터페이스 등을 더 포함할 수 있다. 시스템 프로세서(104)에 의한 이러한 구성요소들의 제어 및 어플리케이션 프로세서(102)와의 그들의 상호운용이 본 발명의 일 실시형태이고, 이들 세부사항들은 아래와 같이 제공될 수 있다. 하지만, 어플리케이션 프로세서(102)의 추가적인 구현상의 세부사항들은 본 발명의 명료성을 위해 생략될 것이다. 또한, 당업자라면 이들 예시들을 숙지한 후에 패드 컴퓨터 및 스파트 폰과 같은 다른 타입의 폰 팩터에 대한 프로세서(102)의 여러 대안적인 실시예들을 이해할 수 있을 것이다.

[0029] 지금부터, 시스템-프로세서 하위-시스템(104)의 실시예들은 그것이 에뮬레이트하고 지원하는 주변장치 타입의 예시들과 시스템-레벨 운영 논리 및 보안(system-level operational logic and security)의 지원을 목적으로 하는 다양한 보조 기능들의 관점에서 설명될 것이다.

[0030] 도 2와 관련하여 예시된 것과 같은 실시예들에서는, 시스템(100), 시스템 프로세서(204)는 어플리케이션 프로세서(202)와 키보드(206), 비디오 먹스(video mux; 208) 및 펌웨어(214)에 커플링된다. 어플리케이션 프로세서(202)는 상술된 바와 같이 어플리케이션 프로세서(102)에 대응될 수 있다. 시스템 프로세서(204)는 종래의 프로세서나, 상표 등록된(proprietary) 프로세서나, 또는 미래의 프로세서, 예컨대 x86 프로세서, 커스텀 ASIC 또는 SOC, ARM 프로세서 등에 의하여 구현될 수 있다. 펌웨어(214)는 시스템 프로세서(204)에 전용되고, 본 명세서에서 기술되고 후술되는 시스템 프로세서(204) 및 그것의 기능을 제어하는 데 필요한 어플리케이션 소프트웨

어 및 모든 운영 시스템을 포함하는 ROM(예를 들어, 플래쉬)에서 구현되는 것이 바람직하다. 당업자라면 펌웨어(214)를 포함하는 소프트웨어의 언어 및 구조는 사용되는 프로세서(204) 및/또는 운영 시스템을 구현하는 데 사용되는 프로세서의 타입에 따라 정해질 수 있다는 것을 이해할 것이다. 또한, 당업자라면 상술된 설명들을 숙지한 후에 프로세서(204)의 기능을 구현하는 펌웨어 및 소프트웨어를 종래의 운영 시스템 및 어플리케이션들과 함께 구현할 방법 또한 이해할 것이다. 시스템 프로세서(204)는 나타내지 않는 추가 기능들 및/또는 구성요소들, 예컨대 프로세서 버스, RAM/어플리케이션 메모리, 그래픽 프로세서 기능, 입력/출력 포트 등을 포함할 수 있다.

[0031] 나타낸 바와 같이, 비디오 멀티플렉서(208)는 적어도 2 개의 입력부(216, 218) 및 1 개의 출력부(220)를 포함한다. 비디오 멀티플렉서의 출력부(220)는 컴퓨터 디스플레이(210)에 연결된다. 비디오 멀티플렉서(208)의 입력부들(218) 중 하나는, 운영 및 보안 관련 정보 및 최종-사용자와의 상호운용을 통신하는 데 사용되며, 시스템-프로세서(204) 펌웨어 내에 임베디드되는 여하한의 어플리케이션의 목적을 위해 사용되는, 시스템-프로세서 하위-시스템(204)에 대해 내부의 비디오-그래픽 모듈에 연결된다. 비디오 멀티플렉서(208)의 다른 입력부(216)는 어플리케이션 프로세서 하위-시스템(202)의 비디오-그래픽 모듈의 출력부에 연결되어, 어플리케이션에 의해 생성되는 그래픽이 멀티플렉서(208)를 향하고, 상기 멀티플렉서를 통해 시스템 프로세서(204) 및 제어 신호(222)의 제어 하에 조건부로 디스플레이 모니터(210)로 향하게 된다. 실시예들에서, 멀티플렉서(208)는 분해능의 비율을 정하고(scale) 디스플레이 모니터(210)의 프레임 속도 및 실제 해상도 및 바람직한 디스플레이 모드에 적절하도록 비디오 입력부들의 프레임 속도를 조정한다. 시스템 프로세서(204)에 의해 결정되는 시스템의 보안 및 운영 모드에 따라, 어플리케이션 그래픽 출력부(216)는 디스플레이로부터 전체적으로 차단되거나, 모니터(210) 상의 작은 창으로서 표시되거나, 또는 디스플레이 모니터(210)의 전체 크기가 되도록 될 수 있다. 또한, 시스템-프로세서(204)의 그래픽들(218) 자체는 운영의 운영 및 보안 모드에 따른 다양한 방식으로 모니터(210)로 조건부로 라우팅될 수 있다.

[0032] 실시예들에서, 초기 시스템의 스타트-업 및 인증(authentication) 동안, 비디오 먹스(208)는 신호(222)를 통한 시스템 프로세서(204)의 제어 하에, 전체 디스플레이(210)가 시스템-프로세서(204) 그래픽에 대해 전용되게 한다. 또한, 시스템 프로세서(204)는 키보드(206)(및 터치패드 등과 같은 다른 입력 디바이스들), 및 사용자를 적절히 인증하는 데 필요한 상호운용을 제어하며, 사용자에게 디스플레이(210)를 통해 이 프로세스의 진행 및 결과들을 알린다. 실시예들에서, 운영 시스템 및 어플리케이션 소프트웨어가 사전에 어플리케이션-프로세서(202)에서 활성화되었다 하더라도 어플리케이션-프로세서(202)의 비디오 출력들은 인증이 성공될 때까지는 보이는 것이 허용되지 않는다(not allowed to be viewed). 인증이 성공적으로 완료되고 시스템 프로세서(204)가 정상적인 운영 모드를 선언하면, 이는 신호(222)를 통해 비디오 먹스(208)로 하여금 입력부(216)로부터의 어플리케이션 그래픽들이 전체 스크린을 점하게 한다.

[0033] 실시예들에서, 시스템-레벨 정보가 전달될 필요가 있는 경우나 시스템-프로세서(204)와의 상호운용을 필요로 하는 특정 키-조합이 눌러진 경우를 제외하고, 시스템-프로세서(204) 그래픽은 성공적인 인증 후의 대부분의 시간 동안 가시적이지 않다. 이러한 시간에, 비디오 먹스(208)는 시스템-프로세서(204) 그래픽이 어플리케이션 그래픽 위의 오버레이(overlay)로서 표시되도록 할 수 있다. 특수한 조건들 하에서, 예를 들어 사용자가 인증되었으나 어플리케이션-프로세서(202)가 시동되거나(activated) 있거나 재시동되고 있는 경우, 또는 시스템-프로세서(204) 펌웨어 내에 임베디드되는 어플리케이션 실행되고 있는 경우, 비디오 먹스(208)는 어플리케이션 그래픽 비디오(216)가 스크린(210) 상에 작은 창으로서 표시되도록 할 수 있으며, 따라서 사용자가 시스템-프로세서(204)와 상호운용하는 동안 그것의 진행을 모니터링할 수 있다.

[0034] 멀티플렉서(208)는 시스템 프로세서(204)로부터의 신호(222)에 의해 제어된다. 이는 크로마-키(chroma-key), 오버레이, 윈도잉(windowing) 등과 같은 다중 소스들로부터 비디오와 그래픽을 혼합하는 종래의 기술, 등록된 기술, 또는 미래의 기술을 이용할 수 있다. 이와 같이, 멀티플렉서(208)의 구현상의 세부사항들은 사용되는 특정 멀티플렉싱 기술에 따라 정해지며, 따라서 그것의 더욱 세부적인 사항들은 본 발명의 간명성을 위해 여기서는 생략된다. 실시예들에서, 어플리케이션 프로세서(202)가 표준 XGA 그래픽 제어기(standard XGA graphics controller)를 포함하는 경우, 표준 XGA 인터페이스가 인터페이스(216)를 구현하는 데 사용된다. 멀티플렉서(208)는 본 명세서에서 인용 참조되는, 계류중인 출원 13/241,073에 기술된 추가적인 비디오 보안 기능을 이용할 수 있다는 것을 또한 유의하여야 한다.

[0035] 인증은 종래의 기술, 등록된 기술 또는 미래의 기술을 포함하여 이루어질 수 있으며, 당업자라면 많은 가능한 대안들을 이해할 것이다. 한 가지 비-제한적인 예시로서, 시스템 프로세서(204)는 사용자로 하여금 사용자명, 패스워드, 보안 키, 바이오메트릭스(예를 들어, 지문)와 같은 증명들(credentials)을 기입/제공할 것을 유도한

다(prompt). 이들 증명들은 국지적으로(locally) 저장된 증명들과 비교되거나, 또는 시스템 프로세서(204)가 상기 증명들을 비교용 원격 인증 서버로 전송할 수 있다. 또한, 국지적으로-저장된 증명들은 시간-제한적(time-limited)이며, 필요에 따라 외부 소스로부터 갱신되거나 폐기될 수 있다.

[0036] 먹스(208)를 통한 디스플레이(210)로의 어플리케이션 프로세서 그래픽(216)의 제공과 유사하게, 시스템 프로세서(204)는 인증에 성공할 때까지 키보드(206) 및 다른 주변장치들에 대한 프로세서(202)의 액세스를 방지한다. 보다 상세히 후술되겠지만, 성공적인 인증 후에, 프로세서(204)는 펌웨어(214)에 의해 제공되는 에뮬레이션 기능에 의해 제어되는 바와 같이 버스(224)를 통하여 키보드(206) 및 다른 주변장치들에 대해 프로세서(202)에 의해 에뮬레이트된 액세스를 허용한다.

[0037] 지금부터, 본 발명의 실시형태들에 따른 컴퓨터 시스템(100)의 데이터 네트워킹 기능을 보안설정하기 위한 예시적 접근법들에 대해 설명될 것이다. 실시예들에서, 시스템(100)은 특정 직원이 사용하기 위해 기업(corporation)이 구입한 독립 컴퓨터이다. 도 3에 예시된 바와 같이, 이들 및 다른 실시예들에서 기업은 공공 액세스 데이터 네트워크(322)(예를 들어, 인터넷)에 상호연결되는 사설 네트워크(320)(예를 들어, 인트라넷)를 추가로 보유/유지시키는 것이 바람직하다. 사설 네트워크(320)는 바람직하게 보안설정되고, 방화벽, 침입 탐지(intrusion detection) 및 다른 포렌식(forensic) 및 구조-레벨 보호 메커니즘들과 같은 종래 디바이스들의 조합에 의해 모든 관련 위협들로부터 충분히 보호된다. 따라서, 본 발명의 이러한 실시예들 중 일 실시형태는 이들 기존의 보호 수단들의 모든 장점과 그들의 중앙집중 구축 및 관리(centralized procurement and management)로부터의 이득을 취한다.

[0038] 도 3에 예시된 것과 같은 실시예들에서는, 시스템-프로세서 하위-시스템(204)에 적어도 2 개의 물리적 네트워킹 인터페이스들이 존재한다. 이들 네트워크 인터페이스들 중 하나[통상적으로 기가비트 이더넷 포트(Gigabit Ethernet port; 304)]는 "백 투 백(back to back)" 구성으로 어플리케이션-프로세서 하위-시스템(202)의 유사 포트(302)에 연결되며, 어플리케이션-프로세서(202)에 대해 이용가능한 유일한 물리적 네트워킹 연결부이다. 그러므로, 프로세서(202)의 어플리케이션에서 비롯되는 모든 트래픽이 시스템-프로세서(204)에 의하여 가로채이게 되고, 어플리케이션으로 향하는 여하한의 패킷은 먼저 시스템-프로세서(204)를 통과하여야 한다.

[0039] 시스템-프로세서(204)의 1 이상의 물리적 네트워킹 인터페이스들(306) 중 다른 하나는 통상적으로 다른 기가비트 이더넷이다. 다른 타입들에는 Wi-Fi와 같은 무선 네트워크 인터페이스 모듈들이 포함될 수 있다. 이들 인터페이스들 중 하나 또는 모두는 자동적으로 검출될 수 있는 가용 물리적 네트워크에 연결될 수 있다. 그 다음, [펌웨어(214)에서 구현되는 것이 바람직한] 시스템 프로세서의 네트워크 관리 기능부(330)는 검출된 네트워크가 식별되고 인증될 수 있는지를 판정한다. 예를 들어, 네트워크 관리 기능부(330)는 이 컴퓨터를 구입한 지정된 기업의 인터넷(320)인지의 여부를 확인할 수 있다. 예를 들어, 네트워크 관리 기능부(330)는 기업 내에서 사용되는 어드레스들의 범위를 저장하고 검출된 네트워크의 어드레스를 이 범위와 비교할 수 있다. 또한, 네트워크 관리 기능부(330)는 알려진 서버들의 리스트 중 하나에 대한 연결을 시도하고, 서버 암호 인증(server cryptographic certificate)을 회수하며 국지적으로 저장된 인증 데이터베이스에 대해 상기 인증을 확인할 수 있다. 인증이 통과되고 직접 연결된 네트워크가 안전한 것으로 판단되면, 그후 시스템-프로세서(204)는 제 1 네트워킹 인터페이스(302/304)와 활성(active) 외부 인터페이스(306) 사이로 모든 패킷들을 전달(forward)한다.

[0040] 인증 프로세스가 성공적이지 않은 경우, 또는 프로세스가 전체적으로 바이패스(bypassed)되도록 높아진 보호 수준을 원하는 몇몇 경우에, 가용 네트워크는 불안정적인 것으로 판단되고, [펌웨어(214)에서 구현되는 것이 바람직한] 시스템-프로세서(204)의 VPN 클라이언트(332)와 기업 인트라넷(320)의 지정된 VPN 게이트웨이(310)의 VPN 서버(334) 사이에 가상의 사설 네트워크("VPN") 연결부(VPN Tunnel로 알려짐; 308)가 구성된다. VPN 터널(308)이 구성되면, 어플리케이션-프로세서(202)에 연결되는 제 1 인터페이스(304)로 오가는 모든 트래픽이 VPN 터널(308)을 독점적으로 통과하여, 어플리케이션 소프트웨어 및 그것의 운영 시스템이, 컴퓨터가 그것을 사용하는 직원에 의해 돌아다니는 동안 여타 공공 또는 사설 네트워크(308)에 연결되는 경우에도 마치 컴퓨터가 기업 보안 네트워크(320)(예를 들어, 인트라넷)에 국지적으로 연결된 것처럼 작동한다. 바람직한 실시예는 VPN 클라이언트(332)를 통해 VPN 터널(308) 상으로 어플리케이션-프로세서(202)로 오가는 미가공(raw) 이더넷 트래픽을 캡슐화하기(encapsulate) 위하여 이더넷 오버 IP 프로토콜 312/314(Ethernet over IP protocol 312/314)[Ethernet over IP protocol 312/314)(여기서, EoIP 312는 펌웨어(214)에서 구현되는 것이 바람직함]을 활용한다. 서버(334) 끝에서, VPN 게이트웨이(310)는 패킷들을 해독(decrypting) 및 확인한 후에 그들을 기업 인트라넷(320)으로 전송한다. 당업자라면 VPN 터널(308)이 전송된 데이터 패킷들의 암호화 및 암호 사인들(cryptographic signatures)을 이용한 이들 패킷들의 신뢰성 확인을 제공한다는 것을 이해할 수 있을 것이다.

따라서, 비밀 정보를 포함하는 데이터 패킷들이 기업 데이터 서버와 컴퓨터 내의 어플리케이션-프로세서(202)에서 실행되는 어플리케이션 소프트웨어 간에 교환되는 경우, 이러한 데이터는 공공 액세스 네트워크(322) 링크에서의 전송되는 동안 도청(eavesdropping)이나 도중의 데이터 변형(en-route data modification)으로부터 보호된다.

[0041] 실시예들에서, 시스템 프로세서(204)의 펌웨어(214) 내에 임베디드되는 어플리케이션들은 동일한 VPN 터널(308)을 액세스하며, 아울러 국지적 가용 네트워크(322)를 직접적으로 액세스한다. 따라서, 임베디드 어플리케이션이 여하한의 민감한 정보를 전송하도록 되어 있는 경우, 이는 VPN 터널(308)을 통해 그것의 트래픽을 실행해야만 한다. 그럼에도 불구하고, 화상-회의 에이전트(video-conferencing agent)와 같은 특정 예시의 임베디드 어플리케이션들에서는, 제어기(204)가 사용자로 하여금 VPN 터널(308)을 통해 기업 네트워크로 진행하고, 그로부터 잠재적으로 다른 VPN 터널을 통해 이어나가는 보안 연결을 선택하도록 하고, 연결이 안전하다는 지표를 표시할 수 있다. 추가적으로 또는 대안적으로, (예를 들어, 성능상의 이유로) 보안 연결이 바람직하지 않은 경우, 제어기(204)는 사용자로 하여금 국지적 가용 네트워크(322)에 직접적으로 액세스함으로써 화상-회의 연결을 조성할 수 있게 하며, 사용자에게 연결이 안전하지 않음을 알려 사용자가 민감한 정보에 대해 논의하는 것을 피할 수 있도록 해야 한다.

[0042] 시스템-프로세서(204)의 펌웨어(214)에 임베디드되는 다른 어플리케이션들은 가상 디스크 이미지(보다 상세히 후술됨)를 백업하고 동기화시키는 어플리케이션들을 포함하여, 상술된 동일한 VPN 터널(308)을 통해 기업 저장 서버에 통신할 수 있다.

[0043] 현재의 컴퓨터들에 중요한 주변장치로는 디스크 드라이브가 있으며, 따라서 지금부터는 본 발명에 따른 이 주변장치에 대한 액세스를 제어하는 예시적 방법에 대해 도 4와 연결하여 설명될 것이다. 도 4에 도시된 바와 같이, 자기 매체-기반 회전 디스크들을 플래쉬-메모리 기반 SSD(Solid State drive)들(410)로 대체하려는 경향이 있다. 알려진 바와 같이, 상기 드라이브는 통상적으로 종래의 컴퓨터가 그것의 소프트웨어와 그것의 데이터의 중요한 부분들 모두를 유지시키는 곳이다. 컴퓨터가 파워-업되는(powered-up) 경우, ROM 또는 플래쉬 메모리로부터 실행되는 소형 로우-레벨 펌웨어, 통상적으로 "BIOS"라 칭해지는 "Basic Input/Output System"가 메모리 및 디스크 드라이브를 초기화하고 디스크 드라이브로부터 메인 메모리로 운영 시스템 소프트웨어를 로딩하도록 진행하는데, 이 프로세스는 "부트-스트랩핑(boot strapping)" 또는 "부팅"이라 칭해진다. 운영 시스템이 실행을 개시하면, 그것은 어플리케이션 소프트웨어와 소프트웨어 라이브러리, 디바이스 드라이버 및 구성 파일들을 읽어들이기 위해 디스크 드라이브를 연속적으로 액세스한다.

[0044] 운영 시스템이 어떠한 보안-관련 메커니즘들을 구현하는 경우, 이들 보호 요소들에서 사용되는 키들 및 패스워드들 또한 같은 디스크에 저장된다. 독립 환경에서 작동될 필요가 있는 소프트웨어 어플리케이션은 네트워킹을 이용할 수 없을 때, 그것의 모든 데이터와, 실행가능한 코드 및 구성 데이터를 디스크 드라이브에 저장할 필요가 있다. 이들 및 다른 이유들로, 디스크 드라이브는 다양한 잠재적 위협들(상기 위협들 중 가장 주된 위협은 디스크 드라이브 자체 또는 전체 컴퓨터의 절도, 및 그에 이어서 디스크에 포함된 데이터의 추출 가능성이다)로부터 보호될 필요가 있다. 현재의 컴퓨터들은 점점 더 디스크 드라이브에 저장되는 데이터를 암호화시키는 것이 일상화되고 있는데, 이때 디스크의 전체 콘텐츠가 단일 키로 암호화된다. 이는 이러한 단일 키가 보안 위협을 받을 경우 데이터를 취약하게 하는 동시에, 새로운 잠재적인 문제, 즉 컴퓨터 사용자가 암호 키를 잊어버리거나 잃어버릴 경우 이 컴퓨터에 할당된 기업이나 직원들 중 어느 누구도 디스크의 어떠한 데이터도 더 이상 회수할 수 없다는 문제를 발생시킨다.

[0045] 본 명세서에 개시된 컴퓨터 구조의 일 실시형태는 디스크 드라이브가 구현되는 방식에 있다. 도 4와 연결하여 예시된 것과 같은 실시예에서는, 어플리케이션 소프트웨어 및 운영 시스템을 실행하는 어플리케이션-프로세서(202)는 어떠한 보안 조치들도 구현할 필요가 없으며, 컴퓨터의 실제 디스크 드라이브(410)에 직접 액세스할 필요가 없다. 그 대신, 어플리케이션-프로세서(202)의 대용량-기억 주변장치 연결부(mass-storage peripheral connection), 통상적으로 Serial-ATA(또는 "SATA") Host 제어기(404)가 시스템-프로세서(204)에서 호환가능한 인터페이스, 즉 Serial-ATA Target 인터페이스(406)에 연결된다. 이 인터페이스(406)는 어플리케이션-프로세서(202)에 의해 생성되는 표준 ATA 명령들에 응답하고, 시스템-레벨 펌웨어(214)와 함께 어플리케이션 프로세서(202)에 에뮬레이트된 디스크 드라이브(414)를 제공한다. 펌웨어(214)에 의해 구현되는 디스크 드라이브를 에뮬레이트하는 프로세스는 가상 환경에서 전개되는 기술들과 유사하다 - 에뮬레이트된 디스크 드라이브(414)는 실제적으로 실제 디스크 드라이브(410)에서 특정 포맷으로 저장되는 파일들의 집합이다.따라서, 시스템-프로세서(204)는 그를 실제 디스크 드라이브(예를 들어, 자기 또는 다른 매체의 HDD), 또는 보다 바람직하게는 Solid State Disk Drive("SSD")(410)에 연결하는 제 2 대용량-기억장치 인터페이스(408)를 갖는다. 이러한 SSD(예를

들어, 플래쉬 메모리 또는 다른 비-휘발성 메모리 기술들, 예컨대 강유전성 RAM 및 상-변화 RAM에 의해 구현됨)는 자기 드라이브를 뛰어 넘는 개선된 성능을 제공하며, 기본적으로 후술되는 에뮬레이션 프로세스 및 암호화가 부과할 수 있는 어떠한 성능 저하도 가질 수 있다.

[0046] 나타낸 예시적 구조에서, 시스템-프로세서(204)의 펌웨어(214)는 파일들과 메인 인덱스 파일의 집합으로서 1 이상의 에뮬레이트된 디스크 드라이브들(414)의 맵을 유지하며, 이는 실제 디스크 드라이브(410)에서의 특정 포맷으로 파일 시스템에 걸쳐 저장된다. 에뮬레이트된 디스크 드라이브(414)가 다수의 파일들에 걸쳐 있어야(span) 하는 몇 가지 이유들이 존재한다. 먼저, 전체적으로 활용되는 디스크 드라이브는 드물며, 따라서 사용되지 않는 저장 공간에 실제 디스크 드라이브의 저장 공간을 할당할 필요가 없다. 따라서, 에뮬레이트된 드라이브(414)의 제공된 공간을 파일들의 집합으로 분할하는 것은 그 어드레스 공간의 스파스 핸들링(sparse handing) 및 사용되고 있지 않은 영역들에 대한 실제 저장의 생략을 가능하게 한다. 둘째로, 특정 시간에 체크-포인트로 알려진 시간의 특정 지점에서 그것의 콘텐츠에 대응되는 에뮬레이트된 드라이브의 일정한 이미지(consistent image)(416)를 유지할 필요성이 존재하며, 그후 모든 순차적으로 변경된 데이터는 새로운 별도의 파일들로 실제 드라이브에 기록되어, 에뮬레이트된 디스크(414)가 연속적으로 사용되는 동안에도 그것의 컨텐츠가 체크-포인트의 시간에 이용가능하게 유지된다. 체크-포인팅이 왜 필요한지에 대한 다양한 이유들이 존재하며, 그 이유 중 하나는 중앙 기업 데이터 저장 볼트(central corporate data storage vault)에서 디스크의 콘텐츠를 백업하거나 동기화하고 컴퓨터가 손실되거나 손상될 경우 데이터의 손실을 방지하기 위한 능력에 있다. 체크-포인팅 및 백업 기능은 시스템-프로세서(204)의 펌웨어(214)에 의하여 구현되며, 프로세서(202)의 운영 시스템 또는 어플리케이션 소프트웨어와는 독립적이다. 또한, 컴퓨터가 활발하게 이용되지 않고 주기적인 백업 프로세스만을 수행할 필요가 있음에 따라, 시스템-프로세서(204)는 어플리케이션-프로세서(202)보다 훨씬 더 적은 전력을 소모하기 때문에, 컴퓨터가 기업 백업 서버와 완전하게 통신할 수 있고 에뮬레이트된 디스크(414)의 최신 영역을 서버에 독립적으로 안전하게 전송할 수 있을 때는 어플리케이션-프로세서(202)에 전력을 제공할 필요가 없다.

[0047] 에뮬레이트된 디스크(414)의 데이터는 실제 디스크 드라이브에 저장될 때 암호화되는 것이 바람직하다. 혼란을 피하기 위해, 어플리케이션-프로세서(202)와 시스템-프로세서(204) 간에 교환되는 데이터 블록들은 암호화되지 않고, 클리어-텍스트(clear-text)로 전송된다. 그 다음, 이 데이터는 실제 SSD(410)에 기록되기 전에 (예를 들어, AES 256을 이용하는) 프로세스(412)에 의해 암호화된다. 에뮬레이트된 디스크 데이터의 암호화에 이용되는 암호 키는 어플리케이션-프로세서(202)의 메모리 공간 내에는 전혀 존재하지 않는 것이 바람직하며, 따라서 컴퓨터 상에 악성 소프트웨어를 설치하는 수단에 의한 이 암호 키를 목표로 하는 어떠한 공격도 무력화된다. 시스템-프로세서(204)의 내부에서도, 에뮬레이트된-디스크(414) 암호 키는 결코 메인 메모리 내에 저장되어서는 안되며, 대신 암호 키들을 위한 별도의 메모리 공간이 정상적인 작동 동안 이들 키들을 저장하는 데 이용되어야 한다. 또한, 특수하게 보호되는 키-저장 메모리는, 이들 키들이 결코 실제 SSD(410)에 저장될 필요가 없고, 대신 시스템-프로세서(206)를 구성하는 칩들 중 하나 내에 보존되도록 비-휘발성 메모리 기술을 이용하여 만들어지는 것이 바람직하다.

[0048] 에뮬레이트된 디스크 보호 레벨의 추가적인 향상을 위해 에뮬레이트된 디스크(414)는 복수의 파일들 내에 저장되기 때문에, 이들 파일들 각각에 특정 암호 키가 지정되어 실제 드라이브가 도난될 경우 전체 디스크의 보안을 위협하는 데 필요한 시간의 양이 배가되도록 하는 것이 바람직하다. 정상 작동 동안, 에뮬레이트된 디스크를 나타내는 파일들 중 여하한의 파일에 의해 사용되는 모든 암호 키들은 시스템-프로세서(204) 지정 보안-모듈 메모리 내에 존재하는 것이 바람직하다.

[0049] 백업 프로세스 동안, 펌웨어(214)는 에뮬레이트된 디스크 데이터(414)를 해독하고, 그를 압축시키며 상술된 VPN 터널 암호 프로토콜 및 키들을 이용하는 전송을 위해 그를 재-암호화한다. 이러한 방식으로, 네트워크에 걸쳐 에뮬레이트된 디스크 암호 키들을 전송하거나 기업 서버에 암호 키들을 저장할 필요가 없어, 이들 서버들 중 하나가 보안 위협을 받게 되는 경우에도 모든 기업 컴퓨터의 데이터가 보안 위협을 받게될 위험을 최소화시킨다. 하지만, 컴퓨터가 손실 또는 손상될 경우, 그것의 에뮬레이트된 디스크(414) 내에 저장되는 데이터는 기업 서버들 중 하나에서 안전하게 유지되며, 새로운 컴퓨터가 동일한 사용자를 위해 신속하게 마련되고 에뮬레이트된 디스크 이미지(416)를 새로운 컴퓨터에 복사함으로써 완전 가동(full operation)으로 복원될 수 있다.

[0050] 어플리케이션-프로세서(202) 및 그것의 소프트웨어의 성능에서 디스크 에뮬레이션의 충격을 최소화시키기 위해서는, 시스템-프로세서(204)는 몇가지 대용량-기억 명령들을 동시에 수행할 수 있어야 한다. 이는 Serial-ATA 명령 세트의 NCQ 기능(Native Command Queuing feature)을 이용하여 구현될 수 있다. 따라서, 한 번에 모두 행해지는 몇 가지의 저장-관련 활동들[저장 명령이 SATA-Target 하드웨어(406)에 의해 메모리 내에 수신되고, 메모리 내의 다수의 데이터 블록이 블록(412)에 의해 암호화되거나 해독되며, 또 다른 저장 명령이 실제 드라이

브에 연결되는 SATA-Host 인터페이스(408)에서 실행되며, 이들 모두는 동시에 수행되어 시스템-프로세서(204)에 의해 적어도 3 가지의 저장 작업들이 동시에 처리될 수 있음]이 존재할 수 있다. 이는, 어플리케이션-프로세서(202) 및 그것이 실행중인 운영 시스템이 첫 번째 명령에 대한 응답을 수신하기 전에 추가적인 저장 명령들을 생성할 수 있는 한, 디스크 드라이브(414)를 에뮬레이트하는 동안 이들 명령들을 다룰 때 증가되는 복잡도에 의해 불가피한 지연을 효과적으로 가려줄 것이다.

[0051] 상술된 그룹들에 속하지 않는 주변장치들, 및 선택적으로 그에 속하는 몇몇은 통상적으로 USB(Universal Serial Bus)를 이용하여 접속된다. 본 발명의 실시형태들에 따른 이들 주변장치들에 대한 안전한 액세스를 제공하는 예시적 방법에 대해 도 5와 연결지어 후술될 것이다. 이러한 주변장치들에는 키보드, 마우스, 프린터, 무선 모뎀, 카드 리더기, 외장 디스크 드라이브 및 다양한 어플리케이션-전용의 I/O 디바이스들(application-specific I/O devices)이 포함될 수 있다. 알려진 바와 같이, USB를 통해 연결될 수 있는 주변장치들은, 각각 컴퓨터 시스템의 보안에 있어 고유한 영향(unique implication)을 갖는 다수의 카테고리에 속한다. 몇몇 USB 주변장치들은 상당히 안전한 것으로 간주되나, 다른 것들은 최근 수년 동안 디지털 정보의 불순한 절도(sophisticated theft) 및 전자적 사보타주(electronic sabotage)를 꾀하는 것으로 매우 잘 알려져 왔다. 따라서, 어떤 주변장치가 허용되는지, 어떤 주변장치가 금지되는지, 그리고 어떤 주변장치가 제한되거나 제어되는 방식으로 사용될 수 있는지를 선택하는 일련의 정책들(a set of policies; 512)을 부과하는 네트워크 보호 방화벽과 다소 비슷하게, USB 주변장치 연결부가 전체적으로 보호의 추가 단계를 거치도록 하는 것이 바람직하다. 예를 들어, 몇몇 회사들은 외부의 USB 대용량-저장 디바이스(예컨대 플래쉬 드라이브)의 이용을 금할 수 있다. 다른 회사들은 그들의 사용을 허용하도록 선택할 수 있으나, 그들이 동일 회사에서 제공된 것이거나, 플래쉬 드라이브로부터 읽어들이거나 거기에 기록된 모든 파일이 임베디드될 수 있는 악성코드나 감사 목적의 검사를 위해 기업 보안팀으로 전송되어야 하는 경우에 한해서만 허용한다.

[0052] 상술된 고려사항들의 견지에서, 도 5에 예시된 본 발명의 실시예들에서의 시스템-프로세서(204)는 2 개의 USB 포트를 포함하는 것이 바람직하며, 상기 USB 포트 중 하나는 확장 타겟(augmented target; 504)으로서 작용하고, 다른 하나는 정상 USB 호스트(502)로서 작용한다. 호스트 포트(502)는 실제 USB 주변장치들(506)에 연결하는 데 이용되며, 상기 주변장치들 중 일부 - 예를 들어, 화상-회의 카메라, 오디오 스피커, 마이크로폰, 키보드 및 터치-패드 - 는 컴퓨터 내부에 있고, 수 개의 표준 USB 포트는 외부 주변장치를 연결하는 데 이용된다.

[0053] USB 타겟 포트(504)는 어플리케이션-프로세서 하위-시스템(202)에 존재하는 표준 USB 포트(508)와 직접적으로 "백-투-백"으로 연결되며, 복수의 USB 주변장치들을 에뮬레이트하기 위해 특수 지정된 펌웨어(214)에 의하여 보완되는 적절한 하드웨어 리소스들을 포함한다. 정규 USB 타겟 제어기들은 단일 타겟 디바이스만을 구현하기에 충분한 리소스들을 포함한다(이 경우에는 충분하지 않을 수 있음). 본 명세서에 개시되는 컴퓨터 구조의 목적을 위해, USB 타겟 포트(504)는 보다 많은 하드웨어 리소스들 및 논리를 구현하는 것이 바람직하며, 각각 자체 디바이스 어드레스를 갖는 다수(예를 들어, 적어도 8 개)의 독립적 USB 주변장치들을 에뮬레이트할 수 있다(514). 이들은 실제로 USB 허브를 통해 어플리케이션-프로세서(202)에 연결되는 수 개의 독립적 USB 디바이스들인 것처럼 어플리케이션-프로세서(202) 운영 시스템 USB 소프트웨어 스택에 의하여 열거된다. 에뮬레이트된 USB 디바이스들(514)의 기능을 제공하도록 의존되는 실제 디바이스들 중 몇몇은 어플리케이션-프로세서(202)에서 USB 인터페이스(508)의 최대 속도보다 느린 속도를 가지기 때문에, USB 타겟 포트(504) 또한 USB 허브들에 필요한 논리를 구현할 경우 전체 시스템 성능의 이득을 얻을 수 있다. 다시 말해, 확장 USB 타겟 포트(504)에 필요한 리소스들은 USB 허브(510) 및 수 개의 독립적인 USB 확장 디바이스들(506)의 리소스들에 대응된다.

[0054] 시스템 프로세서(204)의 펌웨어(214)는 하드웨어 기능을 보완하고, 에뮬레이트된 USB 주변장치들(514) 각각을 USB 호스트 포트(502)에 연결되는 실제 USB 디바이스들(506) 중 하나에 맵핑하지만, 그는 일련의 보안 정책들(512)과 부합하는 방식으로 이행된다. 이들 정책들(512)은 펌웨어(214)에 의해 국지적으로 저장되며, 시간이 지남에 따라 사용자의 개입 없이 기업 서버로부터 자동적으로 회수될 수 있다. 이들 정책들(512)은 기본적으로 무해한 것으로 간주되는 디바이스들의 특정 등급의 투명 브릿징(transparent bridging) 및 맵핑을 허용할 수 있다. 그렇지만 여전히, 심지어 가장 무해한 USB 디바이스(예를 들어, 외부 마우스)조차도 그들의 USB 데이터 구조의 유효성(validity)이 체크되어, 악성적으로-제조된(maliciously-crafted) USB 패킷에 의해 악의적으로 사용될 수 있는 어플리케이션 운영 시스템 USB 스택에 존재할 수 있는 어떠한 취약성도 엔진(512)에 의해 보호되어야 한다.

[0055] 다른 USB 디바이스들은 그들의 등급 및 하위-등급을 토대로 엔진(512)에 의해 차단될 수 있다. 빌트-인 보안 쉴드(built-in security shield)를 가지며 기업에 의해 제공되는 매우 특수한 플래쉬 디바이스들이 허용되는 반

면, 예를 들어 USB 플래쉬 디바이스들이 일반적으로 차단될 수 있도록, 몇몇 USB 디바이스들은 제조자 및 모델 코드를 토대로 또는 훨씬 더 특별하게는 그들의 일련 번호를 토대로 하여 허용될 수 있다.

[0056] 몇몇 경우에, 허용되는 디바이스들은 특정 작용에 의해 에뮬레이트된다(514). 예를 들어, 컴퓨터 키보드가 USB 디바이스이고, 이는 일반적으로 허용되어야 하지만, 예외적으로 특정 키 조합들은 시스템-프로세서(204)에 의하여 가로채어져서 어플리케이션-프로세서(202)에는 전달되지 않아야하고, 특정 시스템-레벨 기능들이 적용되도록 요청하는 데 이용되어야 한다. 다른 예시는 허용된 USB 플래쉬 디바이스가 감사 추적(audit trail)을 유지하기 위한 정책(512)에 의하여 필요할 경우이다. 이러한 경우에, USB 플래쉬 드라이브로 오가는 데이터를 읽어들이고 기록하기 위한 명령들은 시스템-프로세서의 2 개의 USB 포트(502, 504)를 거쳐 전송되지만, 이들 명령들은 수반되는 데이터와 함께 로컬 드라이브의 특수 파일 내에 기록되며, 순차적으로 저장 및 이후의 감사를 위해 VPN 터널을 통해 기업 서버로 전송될 것이다.

[0057] USB 허브(510)의 논리로 확장된 USB 타겟 포트(504), 즉 분리형-트랜잭션(split-transaction)의 지원부에 의하면, 브릿징이 투명하게 이행될 수 있다. 분리형-트랜잭션 지원부는 시스템-프로세서(204)가, (허용될 경우) 동일한 요청을 실제 USB 디바이스(506)에 전달함으로써 어플리케이션-프로세서(202) USB 스택으로부터의 명령을 처리한 다음, 그것이 준비가 되었을 때 응답을 돌려 보낼 수 있게 한다. 분리형-트랜잭션 지원부를 이용하지 않을 경우, 펌웨어가 미리 예측된 USB 명령에 대한 응답을 준비하고 그를 적절한 USB 타겟 디바이스 엔드-포인트 버퍼(USB Target device end-point buffer) 내에 저장할 필요가 있으며, 이는 실제 USB 디바이스와 에뮬레이트된 USB 디바이스 간의 브릿징에 있어 항상 충분한 투명성을 허용하는 것은 아니다.

[0058] 상술된 것 외에, 때때로 순수 작업 목적에 필요하고 어플리케이션 소프트웨어 및 운영 시스템에 영향을 미치지 않는 것들을 포함하는 2 개의 하위-시스템들(202 및 204) 간에 다수의 연결들이 존재할 필요성이 존재한다. 이러한 하나의 연결은 어플리케이션-프로세서 하위-시스템(202)의 전원을 제어하고, 일반 컴퓨터 전원을 에뮬레이트하기 위한 시스템-프로세서 하위 시스템(204)의 능력일 수 있다. 다른 연결은 저-속 시리얼 포트를 통해 이행되는 것이 바람직한 어플리케이션-프로세서(202)에서의 "BIOS" 로우-레벨 운영 소프트웨어를 제어하고 모니터링 할 수 있다. 이는 운영 감독 기능이 시스템-프로세서(204)의 펌웨어(214)에 임베디드되도록 하며, 기업 서버 팜에 의한 원격 구성 및 기업 서버 팜으로의 운영 시스템의 부트스트랩 프로세스의 보고를 허용한다.

[0059] 지금부터, 본 발명의 실시예들에 따른 컴퓨터 시스템(100)에 대한 안전한 액세스를 제공하는 예시적 프로세스가도 6과 연결지어 설명될 것이다.

[0060] 도 6에 나타낸 바와 같이, 프로세싱은 초기 시스템 스타트-업(S602) 동안, 예를 들어 시스템 파워 온/리셋 버튼이 눌러질 때 개시된다. 초기에는, 단계 S604에 나타낸 바와 같이, 시스템 프로세서(204)에서 작동하는 펌웨어(214)는 시스템(100)의 전체 제어를 담당하며 모든 시스템 주변장치들에 대한 어플리케이션 프로세서(202)의 액세스를 차단한다. 예를 들어, 시스템 프로세서(204)는 키보드(206) 및 유사 입력 디바이스, 예컨대 마우에 대한 액세스를 차단한다. 부언하면, 이러한 주변장치들을 통해 시스템(100)에 부착되는 경우에도, 그들로부터의 신호들은 시스템 프로세서(204)에만 제공되어, 이들 신호들은 어플리케이션 프로세서(202)까지 이어지지는 않는다. 이와 유사하게, 시스템 프로세서(204)는 비디오 먹스(208)가 디스플레이(210) 상에 나타날 어플리케이션 프로세서(202)로부터의 비디오 출력들을 차단하게 한다. 한편, 시스템 프로세서(204)는 비디오 먹스(208)가 시스템 프로세서(204)에 의해 출력되는 스타트업 스크린을 표시하게 할 수 있다.

[0061] 단계 S606에 나타낸 바와 같이, 시스템 프로세서(204)는 어플리케이션 프로세서(202)가 부팅될 수 있게 한다. 다른 실시예에서, 이 단계는 사용자가 인증될 때까지 개시되지 않는다. 어느 경우이든, 어플리케이션 프로세서(202)의 스타트업 동안 시스템 프로세서(204)는 [예를 들어, 프로세서(202)가 Window 7과 같은 운영 시스템을 로딩할 수 있도록 하기 위한] 디스크 드라이브(410)에 대한 어플리케이션 프로세서(202)의 액세스를 제어하고, 입력부 및 출력부들이 실제로는 시스템 프로세서(204)에 의해 차단되더라도 어플리케이션 프로세서(202)를 위한 BIOS/운영 시스템에 대한 에뮬레이트된 키보드 및 디스플레이 연결을 제공할 수 있다.

[0062] 다음 단계 S608에서, 시스템 프로세서(204)는 키보드(206)[및 터치패드 등과 같은 다른 입력 디바이스들], 및 사용자를 적절히 인증하는 데 필요한 상호운용들을 독점적으로 제어하고, 디스플레이(210)를 통해 사용자에게 이 프로세스의 진행 및 결과들을 알린다. 상술된 바와 같이, 인증은 어떠한 종래의 기술, 등록된 기술, 미래의 기술도 포함할 수 있으며, 당업자라면 여러 가능한 대안들을 이해할 수 있을 것이다. 하나의 비-제한적 예시에서, 시스템 프로세서(204)는 사용자로 하여금 사용자명, 패스워드, 보안 키, 바이오메트릭스(예를 들어, 지문)와 같은 증명들을 기입/제공할 것을 유도할 수 있다. 이들 증명들은 국지적으로 저장된 증명들과 비교되거나, 또는 상기 증명들을 비교용 원격 인증 서버로 전달할 수 있다.

[0063] 단계 S610에서 인증이 성공적이지 않은 것으로 판정될 경우, 단계 S612에서 오류 스크린이 표시되고, 부착된 주변장치들 상에서의 사용자로부터의 모든 추가적인 입력들은 무시된다.

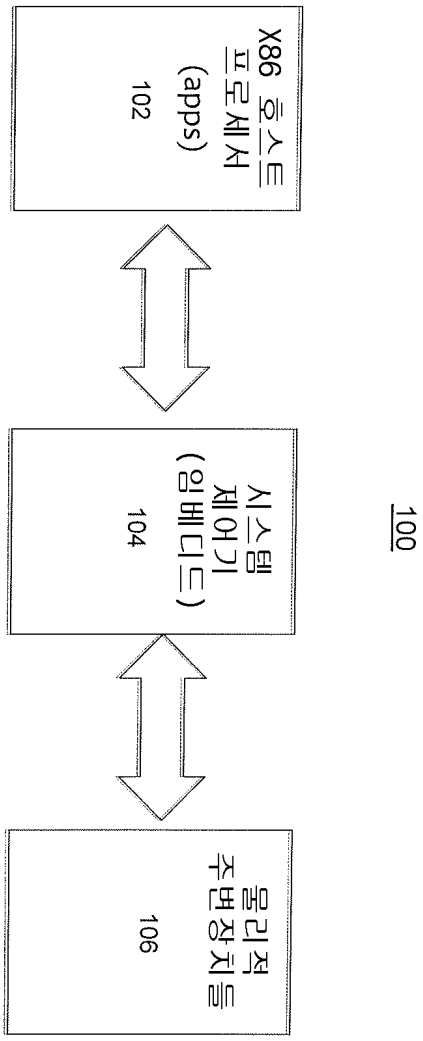
[0064] 그와는 달리, 단계 S610에서 인증이 성공적인 것으로 판정되는 경우, 시스템 프로세서(204)는 S614에서 정상 작동 모드를 알리고, 비디오 먹스(208)가 어플리케이션 프로세서(202) 그래픽이 전체 스크린을 점할 수 있게 한다. 이와 유사하게, 성공적인 인증 후에, 프로세서(204)는 펌웨어(214)에 프로그래밍된 에뮬레이션 기능에 의해 제공되는 바와 같이 버스(224)를 통하여 키보드(206) 및 다른 주변장치들에 대해 프로세서(202)에 의해 에뮬레이트된 액세스를 허용한다. "인증된" 상태는 영구적일 필요는 없다는 데 유의하여야 한다. 예를 들어, 시스템의 비활성 또는 부분적인 셧다운이 발생된 경우, 시스템은 잠금 상태가 되고 미인증 상태로 되돌릴 수 있다. 이 경우에, 어플리케이션 프로세서(202)가 여전히 실행중이라고 할지라도 시스템을 작동 상태로 유지시키는 데 필요한 특정 저장장치 및 네트워크를 제외하고 프로세서(204)에 의해 주변장치로의 액세스가 차단된다. 이 점에서, 시스템 프로세서(204)는 로그온 스크린을 표시하고 사용자와 재-인증을 위해 상호운용할 수 있다.

[0065] 또 다른 예시로서, 시스템이 "인증된" 상태에 있는 동안에도, 시스템 프로세서(204)는 디스플레이(210) 상에 "오버레이" 모드의 메시지나 그래픽을 표시함으로써 작업자의 주의를 주기적으로 환기시키고, 키보드 상의 사전-정의된 키의 조합들[이는 시스템 프로세서(204) 펌웨어의 제어 하에 메뉴가 팝업 상태로 생성되도록 할 수 있음]을 입력함으로써 작업자와 시스템 프로세서(204) 간의 상호운용을 가능하게 한다. 이러한 상호운용은 네트워크 셋팅을 조정하거나, 유지보수 기능을 수행하거나, 또는 시스템 프로세서(204) 펌웨어 내에 구성된 다른 기능, 예컨대 보안 음성 또는 비디오 통신을 적용하는 데 이용될 수 있다.

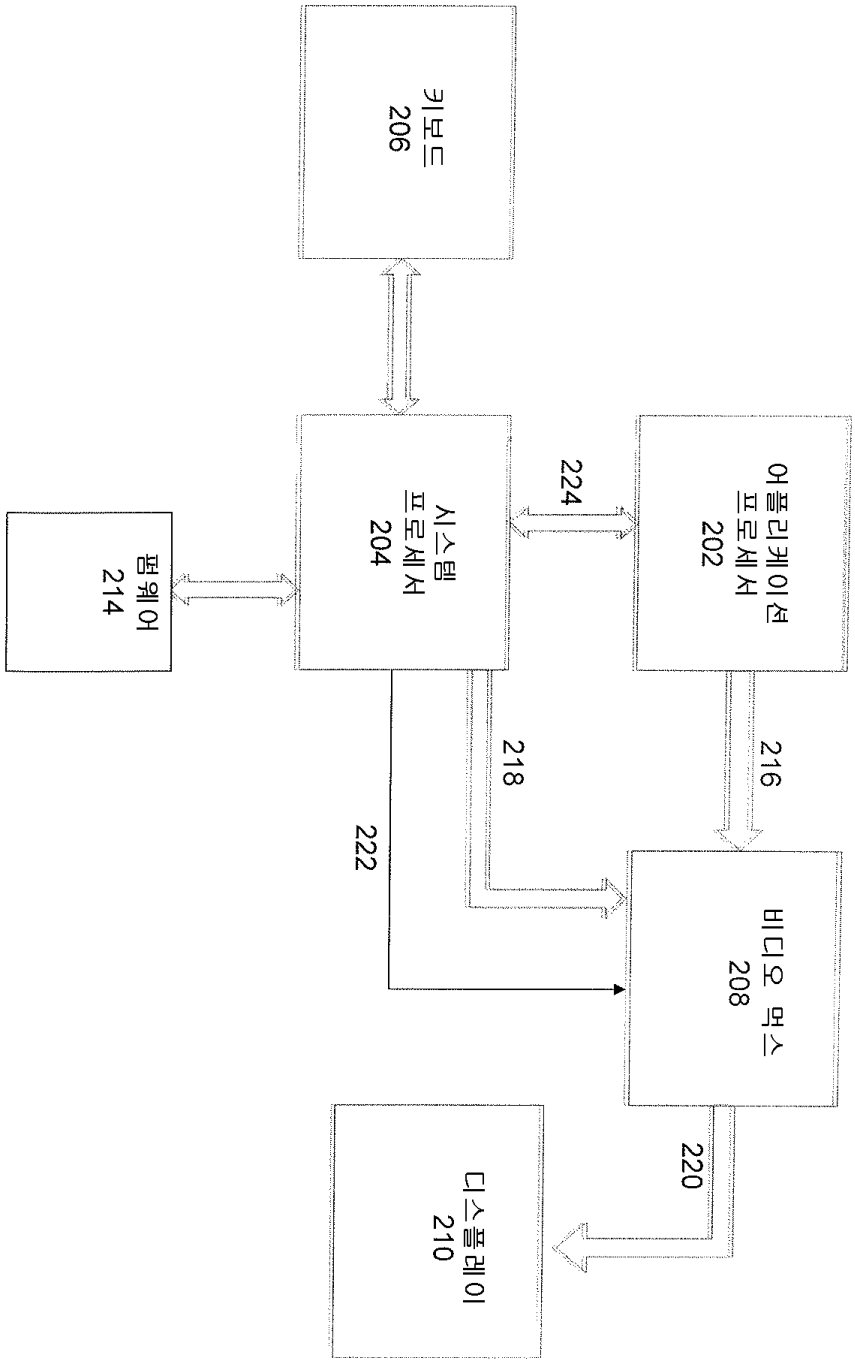
[0066] 본 발명이 바람직한 실시예들을 기준으로 구체적으로 설명되었으나, 당업자라면 본 발명의 기술적 사상 및 범위를 벗어나지 않는 형태 및 세부사항에 있어서의 변경 및 수정들이 가해질 수도 있다는 것을 명확히 이해하여야 한다. 후속 청구범위들은 이러한 변경 및 수정들을 포괄하도록 이루어져 있다.

도면

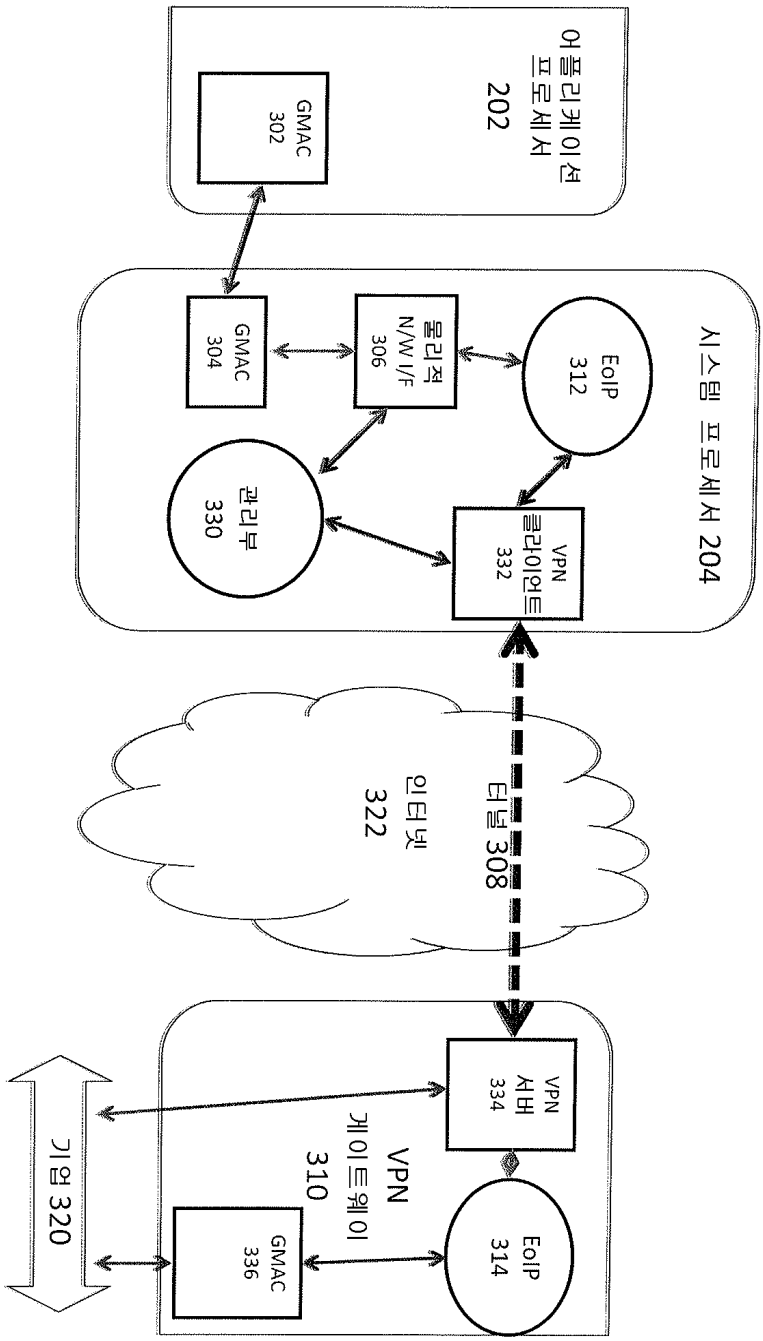
도면1



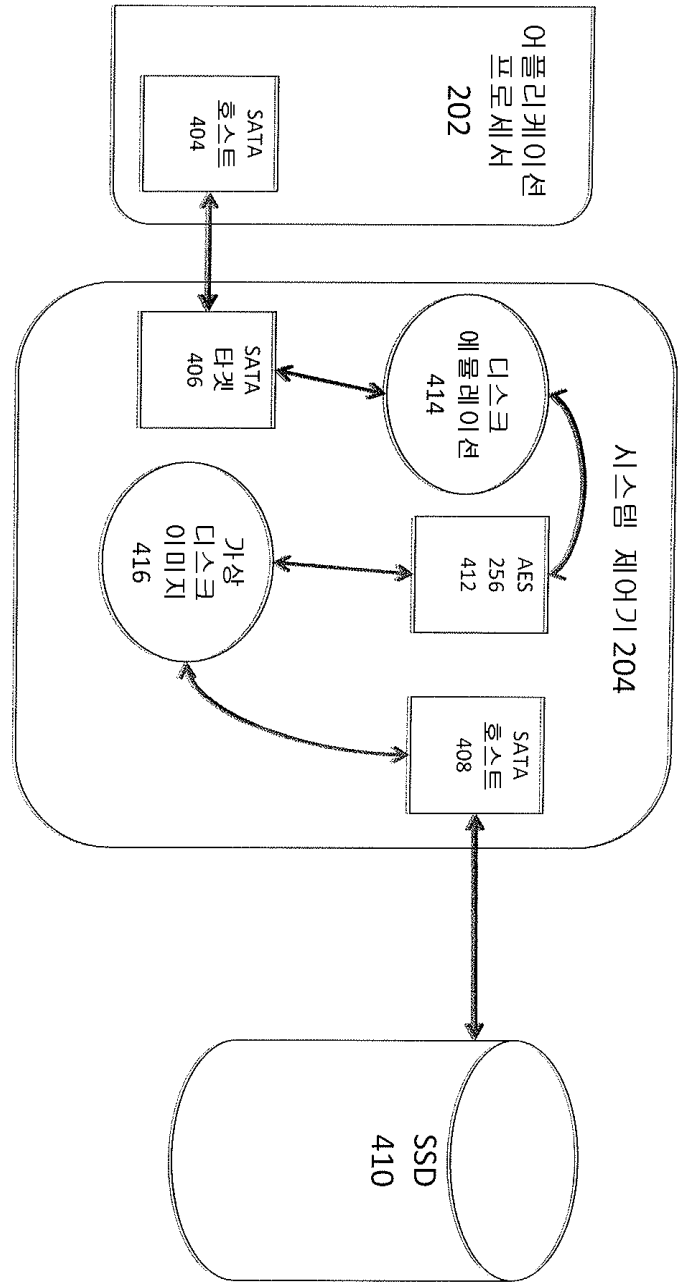
도면2


키보드
206
어플리케이션
프로세서
202
224
시스템
프로세서
204
펌웨어
214
216
218
222
비디오 믹스
208
220
디스플레이
210

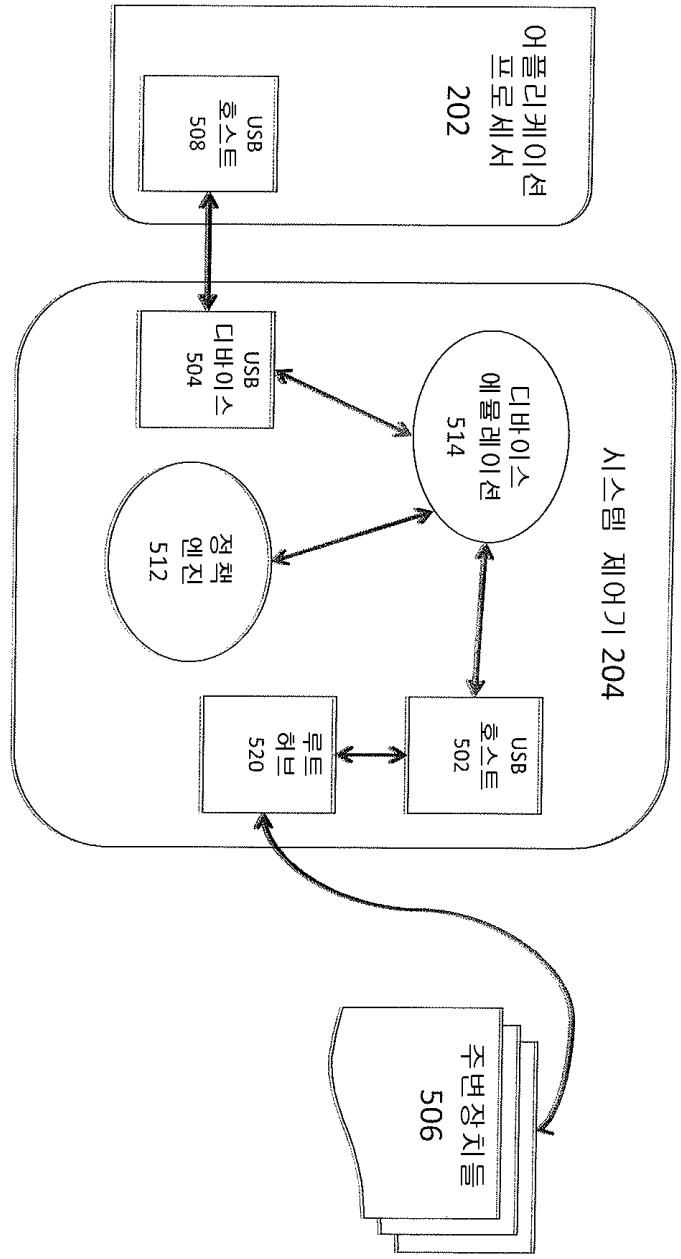
도면3


어플리케이션 프로세서 202
GMAC 302
시스템 프로세서 204
GMAC 304
물리적 N/W I/F 306
EoIP 312
관리부 330
VPN 클라이언트 332
터널 308
인터넷 322
VPN 서버 334
VPN 게이트웨이 310
GMAC 336
EoIP 314
기업 320

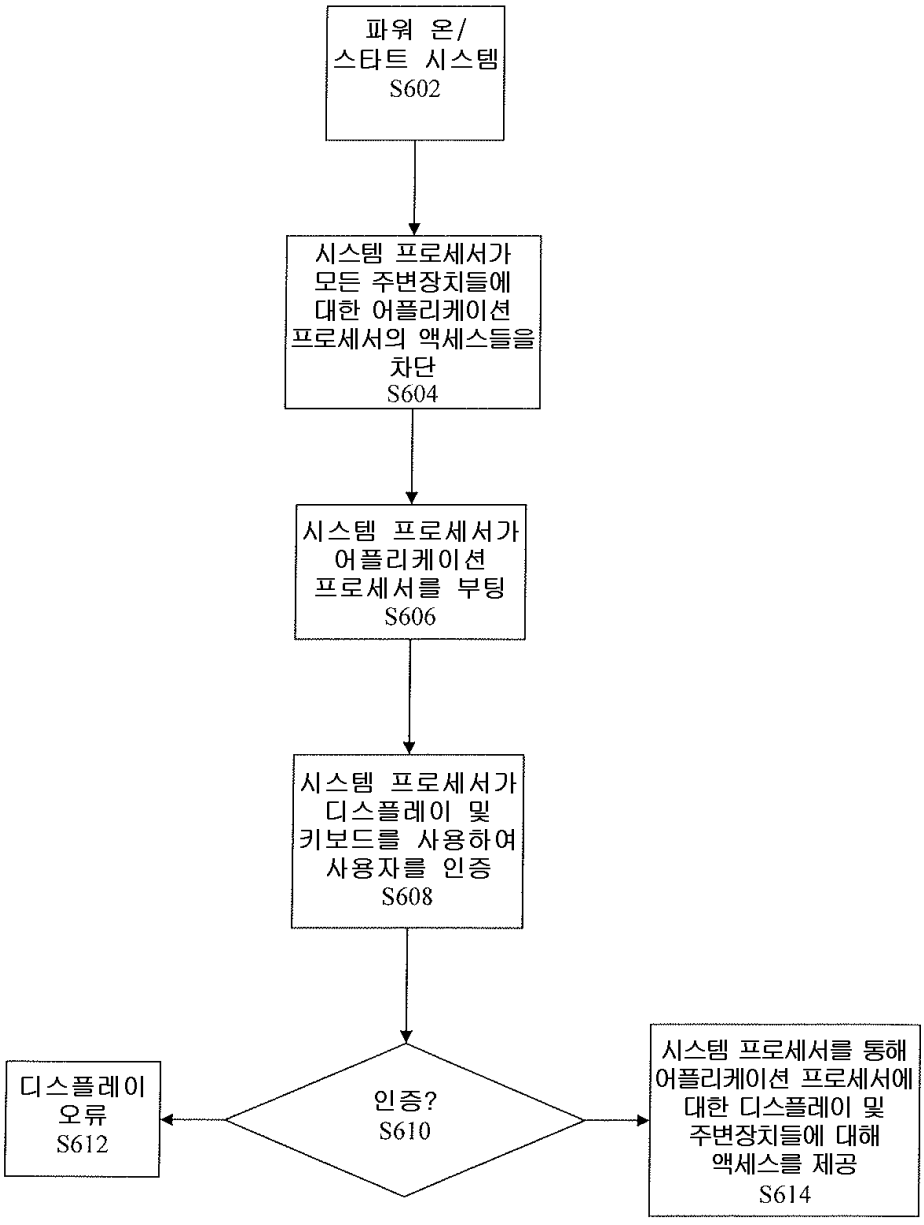
도면4


어플리케이션 프로세서 202
SATA 호스트 404
시스템 제어기 204
SATA 타겟 406
디스크 에뮬레이션 414
AES 256 412
가상 디스크 이미지 416
SATA 호스트 408
SSD 410

도면5


어플리케이션 프로세서 202
USB 호스트 508
시스템 제어기 204
USB 디바이스 504
디바이스 에뮬레이션 514
정책 엔진 512
루트 허브 520
USB 호스트 502
주변장치들 506

도면6


파워 온/
스타트 시스템
S602
시스템 프로세서가
모든 주변장치들에
대한 어플리케이션
프로세서의 액세스들을
차단
S604
시스템 프로세서가
어플리케이션
프로세서를 부팅
S606
시스템 프로세서가
디스플레이 및
키보드를 사용하여
사용자를 인증
S608
인증?
S610
디스플레이
오류
S612
시스템 프로세서를 통해
어플리케이션 프로세서에
대한 디스플레이 및
주변장치들에 대해
액세스를 제공
S614