(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년09월21일
(11) 등록번호 10-1066727
(24) 등록일자 2011년09월15일

(51) Int. Cl.
G06F 21/00 (2006.01) *H04L 9/14* (2006.01)
G06F 13/14 (2006.01) *G06F 9/24* (2006.01)
(21) 출원번호 10-2009-7016608
(22) 출원일자(국제출원일자) 2007년12월20일
심사청구일자 2009년12월10일
(85) 번역문제출일자 2009년08월07일
(65) 공개번호 10-2009-0108706
(43) 공개일자 2009년10월16일
(86) 국제출원번호 PCT/US2007/026279
(87) 국제공개번호 WO 2008/085449
국제공개일자 2008년07월17일
(30) 우선권주장
11/620,689 2007년01월07일 미국(US)
(56) 선행기술조사문헌
EP01369764 A2*
US20030056107 A1*
US20050138409 A1
US6185678 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
애플 인크.
미합중국 95014 캘리포니아 쿠퍼티노 인피니트 루프 1
(72) 발명자
스미스, 마이클
미국 94086 캘리포니아주 써니베일 이스트 맥킨리 애비뉴 475
드 세사르, 조슈아
미국 95008 캘리포니아주 챔프벨 레가스 드라이브 678
(뒷면에 계속)
(74) 대리인
양영준, 서태준, 천세영, 백만기

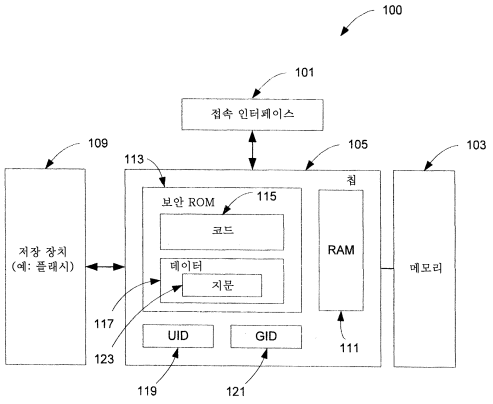
전체 청구항 수 : 총 18 항 심사관 : 경연정

(54) 컴퓨팅 장치의 보안 부팅

(57) 요 약

장치에 내장된 코드를 실행하여 장치의 메모리에 로딩되는 코드 이미지를 검증하는 방법 및 장치에 대하여 기재한다. 코드 이미지는 신뢰성 있는 코드 이미지로서 검증된 후에 실행될 수 있다. 내장 코드는 장치의 보안 ROM(read only memory) 칩에 저장될 수 있다. 한 실시예에서, 코드 이미지의 검증은 보안 ROM 칩 내에 저장된 키에 기초한다. 키는 각 장치에 고유할 수 있다. 키로의 접근은 해당 보안 ROM 칩에 의하여 제어될 수 있다. 장치는 검증된 코드 이미지를 실행한 후 동작 환경의 구축을 완료할 수 있다.

대 표 도 - 도1



(72) 발명자

드 아틀리, 달라스, 브레이크

미국 94114 캘리포니아주 샌프란시스코 넘버2 17번 스트리트 4508

라이트, 존, 앤드류

미국 94110 캘리포니아주 샌프란시스코 크레스센트 애비뉴 364

특허청구의 범위

청구항 1

컴퓨터를 이용해 구현되는 방법으로서,

장치에서 호스트로부터 제1 코드 이미지를 수신하는 것에 응답해서, 상기 장치의 시스템 코드를 실행해 상기 장치의 보안 메모리에 기반한 일련의 증명서(chain of certificates)에 따라 상기 제1 코드 이미지를 검증하는 단계;

상기 제1 코드 이미지가 성공적으로 검증되면 상기 제1 코드 이미지로부터 도출된 서명을 제1 헤더값으로 서명하는 단계;

상기 제1 코드 이미지를 상기 장치의 메인 메모리에 로딩하는 단계;

상기 시스템 코드를 실행해, 상기 제1 코드 이미지가 로딩된 이후, 상기 제1 코드 이미지의 상기 제1 헤더값에 따라 상기 제1 코드 이미지를 상기 보안 메모리에 저장된 키를 이용해 검증하는 단계; 및

상기 제1 코드 이미지가 로딩된 이후 상기 제1 코드 이미지를 성공적으로 검증한 것에 응답하여, 상기 장치의 메인 메모리로부터 상기 검증된 제1 코드 이미지를 실행하는 단계

를 포함하는 컴퓨터를 이용해 구현되는 방법.

청구항 2

제1항에 있어서,

상기 제1 코드 이미지로부터 해시값(hash value)을 도출하는 단계; 및

상기 키에 기초하여 상기 해시값과 상기 제1 코드 이미지로부터의 상기 제1 헤더값 간의 일치 여부를 판단하는 단계 - 상기 제1 헤더값이 상기 제1 코드 이미지에 디지털적으로 서명함 - 를 더 포함하는 컴퓨터를 이용해 구현되는 방법.

청구항 3

제2항에 있어서, 상기 검증된 제1 코드 이미지를 실행하는 단계는,

제2 헤더값을 갖는 제2 코드 이미지로부터 제2 해시값을 도출하는 단계 - 상기 제2 헤더값이 상기 제2 코드 이미지에 디지털적으로 서명함 -; 및

상기 키에 기초하여 상기 제2 해시값과 상기 제2 헤더값 간의 일치 여부를 판단하는 단계를 포함하는 컴퓨터를 이용해 구현되는 방법.

청구항 4

제3항에 있어서, 상기 판단하는 단계는,

상기 제2 해시값을 상기 키와 관련된 서명값으로 암호화하는 단계; 및

상기 서명값과 상기 제2 헤더값을 비교하는 단계를 포함하고,

상기 서명값은 상기 장치와 관련된 시드열(seed string)에 기초하는, 컴퓨터를 이용해 구현되는 방법.

청구항 5

제1항에 있어서, 상기 키는 상기 장치를 고유하게 식별하도록 구성되는, 컴퓨터를 이용해 구현되는 방법.

청구항 6

제1항에 있어서, 상기 장치는 휴대용 장치인, 컴퓨터를 이용해 구현되는 방법.

청구항 7

제1항에 있어서, 상기 제1 코드 이미지는 상기 장치의 동작 환경을 구축하도록 구성되는, 컴퓨터를 이용해 구현

되는 방법.

청구항 8

제1항에 있어서,

상기 제1 코드 이미지는 운영 체제(OS)의 커널(kernel)의 구성요소를 나타내고,

상기 방법은, 상기 OS의 상기 커널의 각 구성요소에 대하여 상기 커널의 모든 구성요소가 검증되고 실행될 때까지 각 구성요소의 코드 이미지를 검증하고 실행하는 것을 반복하는 단계를 더 포함하는 컴퓨터를 이용해 구현되는 방법.

청구항 9

기계에 의하여 실행될 때 기계로 하여금 방법을 수행하도록 하는 명령어들을 가지는 기계 판독 가능 매체로서, 상기 방법은:

호스트로부터 장치로 제1 코드 이미지를 수신하는 것에 응답해서, 상기 장치의 시스템 코드를 실행해 상기 장치의 보안 메모리에 기반한 일련의 증명서(chain of certificates)에 따라 상기 제1 코드 이미지를 검증하는 단계;

상기 제1 코드 이미지가 성공적으로 검증되면 상기 제1 코드 이미지로부터 도출된 서명을 상기 제1 코드 이미지의 제1 헤더값으로 서명하는 단계;

상기 제1 코드 이미지를 상기 장치의 메인 메모리에 로딩하는 단계;

상기 시스템 코드를 실행해, 상기 제1 코드 이미지가 로딩된 이후, 상기 제1 코드 이미지의 상기 제1 헤더값에 따라 상기 제1 코드 이미지를 상기 보안 메모리에 저장된 키를 이용해 검증하는 단계; 및

상기 제1 코드 이미지가 로딩된 이후 상기 제1 코드 이미지를 성공적으로 검증한 것에 응답하여, 상기 장치의 메인 메모리로부터 상기 검증된 제1 코드 이미지를 실행해 상기 장치를 위한 동작 환경을 구축하는 단계를 포함하는, 기계 판독 가능 매체.

청구항 10

제9항에 있어서, 상기 방법은,

상기 제1 코드 이미지로부터 해시값을 도출하는 단계; 및

상기 키에 기초하여 상기 해시값과 상기 제1 코드 이미지로부터의 제1 헤더값의 일치 여부를 판단하는 단계 - 상기 제1 헤더값이 제1 코드 이미지에 디지털적으로 서명함 - 를 더 포함하는 기계 판독 가능 매체.

청구항 11

삭제

청구항 12

삭제

청구항 13

제9항에 있어서, 상기 키는 상기 장치를 고유하게 식별하도록 구성되는, 기계 판독 가능 매체.

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

장치에서 호스트로부터 제1 코드 이미지를 수신하는 것에 응답해서, 상기 장치의 시스템 코드를 실행해 상기 장치의 보안 메모리에 기반한 일련의 증명서(chain of certificates)에 따라 상기 제1 코드 이미지를 검증하는 수단;

상기 제1 코드 이미지가 성공적으로 검증되면 상기 제1 코드 이미지로부터 도출된 서명을 상기 제1 코드 이미지의 제1 헤더로 서명하는 수단;

상기 제1 코드 이미지를 상기 장치의 메인 메모리에 로딩하는 수단;

상기 시스템 코드를 실행해, 상기 제1 코드 이미지가 로딩된 이후, 상기 제1 코드 이미지의 상기 제1 헤더에 따라 상기 제1 코드 이미지를 상기 보안 메모리에 저장된 키를 이용해 검증하는 수단; 및

상기 제1 코드 이미지가 로딩된 이후 상기 제1 코드 이미지를 성공적으로 검증한 것에 응답하여, 상기 장치의 메인 메모리로부터 상기 검증된 제1 코드 이미지를 실행해 상기 장치를 위한 동작 환경을 구축하는 수단을 포함하는 장치.

청구항 18

전자 장치에 있어서,

내장된 시스템 코드와 상기 전자 장치를 식별하는 키를 저장하는 보안 메모리;

제1 코드 이미지를 저장하는 대용량 저장 장치;

메인 메모리; 및

상기 보안 메모리, 상기 대용량 저장 장치 및 상기 메인 메모리에 연결된 프로세서를 포함하고,

상기 프로세서는,

장치에서 호스트로부터 제1 코드 이미지를 수신하는 것에 응답해서, 상기 시스템 코드를 실행해 상기 보안 메모리에 기반한 일련의 증명서(chain of certificates)에 따라 상기 제1 코드 이미지를 검증하는 단계;

상기 제1 코드 이미지가 성공적으로 검증되면 상기 제1 코드 이미지로부터 도출된 서명을 상기 제1 코드 이미지의 제1 헤더로 서명하는 단계;

상기 시스템 코드를 실행해, 상기 제1 코드 이미지가 상기 대용량 저장 장치로부터 로딩된 이후, 상기 제1 코드 이미지의 상기 제1 헤더에 따라 상기 제1 코드 이미지를 검증하는 단계; 및

상기 제1 코드 이미지가 로딩된 이후 상기 제1 코드 이미지를 성공적으로 검증하면, 상기 메인 메모리로부터 상기 검증된 제1 코드 이미지를 실행해 상기 전자 장치를 위한 동작 환경을 구축하는 단계를 실행하도록 구성되는, 전자 장치.

청구항 19

컴퓨터를 이용해 구현되는 방법으로서,

호스트로부터 휴대용 장치로 제1 실행 가능 이미지를 수신하는 것에 응답해서, 상기 휴대용 장치의 시스템 코드를 실행해 상기 휴대용 장치의 보안 메모리에 기반한 일련의 증명서(chain of certificates)에 따라 상기 제1 실행 가능 이미지를 검증하는 단계;

상기 제1 실행 가능 이미지가 성공적으로 검증되면 상기 제1 실행 가능 이미지로부터 도출된 서명을 상기 제1 실행 가능 이미지의 제1 헤더로 서명하는 단계;

상기 제1 실행 가능 이미지를 상기 휴대용 장치의 메인 메모리에 로딩하는 수단;

상기 시스템 코드를 실행해, 상기 제1 실행 가능 이미지가 로딩된 이후, 상기 제1 실행 가능 이미지의 상기 제1 헤더에 따라 상기 제1 실행 가능 이미지를 상기 메모리에 저장된 키 및 상기 휴대용 장치를 고유하게 식별하는 키를 이용해 검증하는 단계 - 상기 제1 실행 가능 이미지는 상기 휴대용 장치의 동작 환경을 제공하는 운영 체

제(OS)의 커널을 나타내고, 상기 제1 실행 가능 이미지는 상기 휴대용 장치의 대용량 저장 장치에 저장됨 -; 및

상기 제1 실행 가능 이미지가 로딩된 이후 상기 제1 실행 가능 이미지를 성공적으로 검증하면, 상기 휴대용 장치의 상기 메인 메모리로부터 상기 제1 실행 가능 이미지를 실행해 상기 휴대용 장치의 상기 운영 체제(OS)를 구축하기 위해 상기 OS의 상기 커널을 설정하는 단계를 포함하는 컴퓨터를 이용해 구현되는 방법.

청구항 20

제19항에 있어서, 상기 제1 실행 가능 이미지를 성공적으로 실행하면, 상기 커널이 상기 대용량 저장 장치로부터의 제2 실행 가능 이미지를 구성하고 로딩하여 상기 OS의 나머지를 구축하는 단계를 더 포함하고,

상기 제2 실행 가능 이미지는 사용자 애플리케이션, 사용자 데이터 및 라이브러리(library) 중 적어도 하나를 나타내는, 컴퓨터를 이용해 구현되는 방법.

청구항 21

제20항에 있어서, 내장된 코드들 및 상기 키는 제조자가 상기 휴대용 장치를 배포하기 전에 상기 보안 메모리에 내장되며, 상기 제1 실행 가능 이미지는 상기 키를 나타내는 데이터에 의하여 디지털적으로 서명되는, 컴퓨터를 이용해 구현되는 방법.

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

컴퓨터를 이용해 구현되는 방법으로서,

호스트로부터 장치로 제1 실행 가능 이미지를 수신하는 것에 응답해서, 상기 장치의 보안 메모리로부터의 시스템 코드를 실행해 상기 보안 메모리에 기반한 일련의 증명서에 따라 상기 제1 실행 가능 이미지를 검증하는 단계;

상기 제1 실행 가능 이미지가 성공적으로 검증되면 상기 제1 실행 가능 이미지로부터 도출된 서명을 상기 제1 실행 가능 이미지의 헤더로 서명하는 단계;

상기 서명을 가지고 서명된 상기 제1 실행 가능 이미지를 상기 장치와 연관된 대용량 저장 장치에 저장하는 단계;

상기 보안 메모리로부터 내장된 제2 실행 가능 이미지를 실행해 상기 대용량 저장 장치가 액세스되도록 상기 대용량 저장 장치를 초기화시키는 단계 - 상기 보안 메모리는 상기 장치를 고유하게 식별하는 고유 식별자(ID)를 저장함 -;

상기 대용량 저장 장치를 성공적으로 초기화시키면, 상기 보안 메모리에 내장된 상기 고유 식별자(ID)를 이용하여 상기 제1 실행 가능 이미지의 상기 헤더에 따라 상기 대용량 저장 장치에 저장된 제1 실행 가능 이미지를 찾아내고 검증하는 단계;

상기 제1 실행 가능 이미지를 성공적으로 검증하면, 상기 제1 실행 가능 이미지를 실행해 상기 장치에서 저레벨 하드웨어 초기화를 수행하는 단계; 및

상기 제1 실행 가능 이미지가 성공적으로 실행된 경우, 제3 실행 가능 이미지를 찾아내고 실행하는 단계 - 상기 제3 실행 가능 이미지는 상기 장치에 대한 운영 체제(OS)의 커널 이미지를 검증하고 로딩하기 위한 것이고, 상기 커널 이미지가 성공적으로 로딩된 경우 상기 장치에 대한 상기 OS의 나머지를 초기화하고 구성함 - 를 포함하는 컴퓨터를 이용해 구현되는 방법.

청구항 26

제25항에 있어서, 상기 제1, 제2 및 제3 실행 가능 이미지 중 임의의 실행 가능 이미지의 검증이 실패한 경우, 상기 방법은, 후속 실행 가능 이미지를 실행하지 않고 즉시 각각의 실행 가능 이미지를 복원하는 복원 처리를 행하는 단계를 더 포함하는 컴퓨터를 이용해 구현되는 방법.

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

명 세 서

기 술 분 야

[0001] 본 발명은 일반적으로 전자 보안에 관한 것이다. 특히, 본 발명은 컴퓨팅 장치를 안전하게 부팅하는 것에 관한 것이다.

배 경 기 술

[0002] 사람들의 일상 생활에 점점 더 많은 컴퓨팅 장치가 사용되면서, 보안성이 사용자와 콘텐츠 제공자에게 일반적인 관심사가 되어 왔다. 바이러스, 웜(worm), 트로이 목마, 주민번호 도용, 소프트웨어 및 미디어 콘텐츠 침해, 그리고 데이터 파괴 위협을 이용한 강탈이 만연하고 있다. 통상적으로, 이러한 공격은 시스템, 콘텐츠 제공자, 사용자 또는 애플리케이션에 있어서는 사적 영역일 장치 자원으로 접근을 노출시키는 악의적인 소프트웨어 코드를 설치하고 실행하는 것과 관련이 있다.

[0003] 예를 들어, 해커 프로그램은, 헐리우드 영화 또는 음악과 같은 오디오/비디오 콘텐츠를 재생하도록 개발된 소비자 컴퓨팅 장치에서 실행되고 있는 경우, 그 A/V 콘텐츠를 안전하게 하는데에 이용되는 암호화를 크래킹(cracking)하는 것을 잠재적으로 허용할 수 있다. 따라서, 이러한 장치에는 통상적으로 높은 수준의 보안성이 필요하다.

[0004] 운영 체제는 이러한 공격을 방어하기 위한 약간의 보안 특성을 제공할 수 있다. 하지만, 운영 체제의 보안 특성은 종종 매일 발생하는 새로운 공격을 따라가지 못한다. 더욱이, 컴퓨팅 장치를 부팅할 때, 보안 특성이 아직 초기화되지 않을 수 있고 바이패스(bypass) 및/또는 탬퍼링(tampering)에 취약하다.

[0005] 이러한 공격을 방어하는 다른 방법은 공장에서 컴퓨팅 장치가 출하된 후 어떠한 부가적인 소프트웨어를 설치 및/또는 실행하는 것을 완전히 봉쇄하는 것이다. 하지만, 이러한 엄격한 조치는 컴퓨팅 장치의 잠재적인 능력 및 융통성을 심각하게 제한한다. 컴퓨팅 장치의 업그레이드에 비용이 들면서 어려운 것은 물론, 장치 외부로부터의 소프트웨어 코드를 다운로드하고 실행할 필요가 점차 많아지는 애플리케이션을 활용할 수 없게 된다. 또한, 일반적으로 급속한 기술의 발전은 컴퓨팅 장치 내에 최초에 설치된 애플리케이션 또는 기능을 매우 단시간내에 구식으로 만들어 버린다.

[0006] 따라서, 현재의 보안 조치는 컴퓨팅 장치 내의 애플리케이션 및 콘텐츠를 보호하는 확실한 해법을 제시하지 못함은 물론 장치에 대한 소프트웨어 및 펌웨어를 업데이트하는 융통성을 제공하지 못한다.

발명의 상세한 설명

[0007] 장치에 내장된 코드를 실행하여 장치의 메모리에 로딩되는 코드 이미지를 검증하는 방법 및 장치에 대하여 본원

에 기재한다. 코드 이미지는 신뢰성 있는 코드 이미지인 것으로 검증된 후에 실행될 수 있다. 내장된 코드는 장치의 보안 ROM(read only memory)등의 보안 저장 영역에 저장될 수 있다. 한 실시예에서, 코드 이미지의 검증은 보안 저장 영역내에 저장된 키에 기초한다. 키는 각 장치에 고유할 수 있다. 키로의 접근은 연관된 보안 저장 영역에 의하여 제어될 수 있다. 장치는 검증된 코드 이미지를 실행한 후 동작 환경의 구축을 완료할 수 있다.

[0008] 대체 실시예에서, 코드 이미지는 장치의 동작 환경을 제공하는 운영 체제(OS)의 커널을 나타낼 수 있다. 코드 이미지는 장치의 대용량 저장 장치로부터 로딩될 수 있다. 코드 이미지는, 보안 ROM 내에 저장되어 있는, 장치를 고유하게 식별하는 키를 이용하여 성공적으로 검증된 경우, 장치의 메인 메모리에서 실행되어 장치의 동작 환경을 구축하도록 OS의 커널을 설정할 수 있다. 한 실시예에서, 장치는 휴대용 장치이다.

[0009] 대체 실시예에서, 장치의 보안 ROM으로부터 내장된 제1 실행 가능 이미지를 실행하여 장치와 연계된 대용량 저장 장치를 초기화시킴으로써 대용량 저장 장치로 접근을 가능하게 한다. 보안 ROM은 장치를 고유하게 식별하는 고유 식별자(ID)를 저장할 수 있다. 대용량 저장 장치를 성공적으로 초기화시킨 경우, 보안 ROM 내에 내장된 고유 식별자(ID)를 이용하여, 대용량 저장 장치에 저장된 제2 실행 가능 이미지를 찾아 내고 검증할 수 있다. 성공적으로 검증한 후, 제2 실행 가능 이미지를 실행하여 상기 장치에서 저레벨 하드웨어 초기화를 수행할 수 있다. 제2 실행 가능 이미지가, 제3 실행 가능 이미지를 찾아 내고 실행하도록, 성공적으로 실행되어, 장치에 대한 운영 체제(OS)의 커널 이미지를 검증하고 로딩할 수 있다. 장치에 대한 OS의 나머지를 초기화하고 구성하도록, 상기 커널 이미지가 성공적으로 실행될 수 있다.

[0010] 본 발명의 다른 특징은 첨부 도면과 이하의 상세한 설명으로부터 명백해질 것이다.

실 시 예

[0024] 컴퓨팅 장치의 보안 부팅 방법 및 장치에 대하여 여기에 기재한다. 이하의 설명에서, 특정한 여러 상세는 본 발명의 실시예의 설명을 통하여 제공되도록 기재된다. 하지만, 본 발명의 실시예는 이러한 특정 상세 없이도 실시될 수 있음은 당업자에게 자명하다. 다른 경우, 공지의 구성요소, 구조 및 기술은 본 기재의 이해를 명확히 하기 위해서 상세히 나타내지 않았다.

[0025] 명세서 한 실시예("one embodiment", "an embodiment")라는 언급은 그 실시예와 관련하여 기재되는 특별한 특징, 구조 또는 특성이 본 발명의 적어도 한 실시예에 포함될 수 있음을 의미한다. 명세서의 여러 곳에서 나타나는 "한 실시예에서"라는 것은 반드시 동일한 실시예를 지칭하지 않는다.

[0026] 첨부하는 도면에서 나타낸 처리는 하드웨어(예를 들어, 회로, 전용 로직 등), 소프트웨어(범용 컴퓨터 시스템 또는 전용 기계 상에서 실행되는 것과 같은), 또는 이들 둘의 조합으로 이루어지는 처리 로직에 의하여 수행된다. 처리가 일부 순차적인 동작의 관점에서 기술하지만, 기술된 동작의 일부는 상이한 순서로 수행될 수 있다. 또한, 일부 동작은 순차적이 아닌 병렬적으로 수행될 수 있다.

[0027] "호스트(host)" 및 "장치(device)"라는 용어는 호스트에 대한 폼 팩터(form factor) 대 장치에 대한 특정 폼 팩터를 지칭하기 보다는 일반적으로 데이터 처리 시스템을 지칭하고자 한다.

[0028] 한 실시예에서, 장치의 보안 부팅은 장치 내의 주요 자원이 동작 환경에서 확실히 보호되도록 설계될 수 있다. 한편, 장치의 보안 부팅은 장치 내에서 실행되는 소프트웨어를, 불필요한 관리, 재료(material) 및/또는 실행 비용이 필요 없이 상이한 정책 및 절차에서 업데이트하고 설치할 수 있는 융통성을 제공할 수 있다. 한 실시예에서, 장치를 부팅하는 보안은 장치 내에 함께 집적되는 ROM(Read Only Memory), 또는 보안 ROM이라 불리는 보안 저장 영역 내에 저장된 코드 및 데이터에 의하여 이루어진다. 보안 ROM의 내용은 장치 제조 단계에서 저장될 수 있다. 보안 ROM은 장치를 고유하게 식별하는 UID(Unique Identifier)와 관련될 수 있다. 장치에서 실행되는 소프트웨어 코드의 신뢰성은 UID에 기초하는 보안 ROM을 통해 서명되는 코드 이미지(code image)로부터 기원할 수 있다.

[0029] 한 실시예에 따르면, 장치의 보안 ROM은 신뢰 개체(trusted entity)의 기본적인 증명인 지문을 포함할 수 있다. 신뢰 개체를 통해 증명된 코드 이미지는 안심하고 지문에 기초한 보안 ROM을 통한 증명 처리에 따라 장치 내에서 실행되도록 할 수 있다. 한 실시예에서, 장치의 보안 부팅은 보안 ROM에 따라 신뢰 개체에 결합될 때 신뢰 소프트웨어 코드를 복원할 수 있다. 보안 ROM은 저장된 장치 UID에 기초한 지문을 통하여 증명된 코드 이미지로 신뢰를 확장할 수 있다. 한 실시예에서, 보안 ROM은 외부 연결로부터 다운로드되는 코드 이미지를 증명함으로써 애플리케이션 소프트웨어를 복원할 수 있다. 다른 실시예에서, 보안 ROM은 외부 연결을 통해 다운로드되

는 신뢰 소프트웨어 코드에 의하여 장치 내에 저장된 사용자 데이터를 강제로 클리닝(cleaning up)할 수 있다.

[0030] 도 1은 보안 부팅을 위한 시스템 구성요소의 한 실시예를 나타내는 블록도이다. 시스템(100)은 장치 내의 하나 이상의 칩에 내장될 수 있다. 한 실시예에서, 시스템(100)은 메모리 구성요소(103)와 연결되는 칩(105)을 포함할 수 있다. 칩(105)은 SRAM(Static Random Access Memory) 또는 EDRAM(Embedded Dynamic Random Access Memory)와 같은 RAM 구성요소(111)를 포함할 수 있다. 코드 이미지는 장치에 의하여 실행되기 전에 메모리 구성요소(103)에 로딩될 수 있다. 코드 이미지는 실행될 때 사용자 또는 시스템 애플리케이션을 지원하는 장치에 대한 사용자 애플리케이션, 시스템 애플리케이션 및/또는 동작 환경(예를 들어, 운영 체제)을 작동시킬 수 있다. 한 실시예에서, 메모리 구성요소(103)는 DDR(Double Data Rate) 메모리를 포함한다. 칩(105)은 코드(115) 및 관련 데이터(117)를 저장하는 ROM(113)을 포함할 수 있다. 코드(115)는 암호 작성 해시 기능(cryptographic hash function) SHA-1, SHA-22, SHA-256, SHA-384 및 SHA-512와 같은 SHA(Secure Hashing Algorithm) 해싱 기능의 구현을 포함할 수 있다. 또한, 코드(115)는 AES(Advanced Encryption Standard) 암호화와 같은 데이터 암호화 알고리즘의 구현을 포함할 수 있다. 한 실시예에서, 코드(115)는 장치에 대한 하드웨어를 초기화시켜서 USB(Universal Serial Bus)와 같은 접속 또는 통신 인터페이스를 지원할 수 있다. 코드(115)는 장치의 클록 레이트(clock rate)를 변경시키는 명령어를 포함할 수 있다. 본 출원 전체에서, SHA 및 AES는 설명을 위한 예로만 사용됨을 유의하여야 하며, 다른 해싱 및/또는 암호화 기술을 또한 이용할 수 있음을 인식할 것이다.

[0031] 한 실시예에서, 코드(115)는 메모리 구성요소(103) 또는 RAM(111)과 같은 장치 메모리에 코드 이미지를 로딩시킬 수 있다. 코드 이미지는 칩(105)과 연결되는 저장 장치 구성요소(109)로부터 로딩될 수 있다. 저장 장치 구성요소(109)는 플래시 메모리, 예를 들어, NAND 플래시, NOR 플래시 또는 다른 대용량 저장 장치(예: 하드 디스크) 구성요소일 수 있다. 다른 실시예에서, 코드 이미지는 장치 외부의 소스로부터 접속 인터페이스(101)를 통하여 로딩될 수 있다. 접속 인터페이스(101)는 USB 접속, 이더넷 접속 또는 무선 네트워크 접속(예: IEEE 802.1x) 등에 기초할 수 있다. 한 실시예에서, 코드(115)는 코드 이미지가 신뢰 코드만을 포함하는지 검증한 후에 장치 메모리에서 저장 장치 구성요소(109)로 코드 이미지를 저장시킬 수 있다.

[0032] 장치가 장치 메모리에 로딩된 코드 이미지를 실행시키기 전에, 코드(115)는 코드 이미지가 확실히 신뢰할만한지 확인하기 위해 로딩된 코드 이미지에 대한 검증 동작을 수행할 수 있다. 한 실시예에서, 코드(115)는 ROM 내의 데이터 섹션(117), UID(119) 및/또는 GID(Global Identifier)(121)과 같이 칩(105) 내에 포함된 데이터에 따라 로딩된 코드 이미지를 검증할 수 있다. UID(119)는 각 장치에 있어서 고유한 것일 수 있다. 한 실시예에서, 모든 장치는 단일 GID(121)와 연관되어 있다. 한 실시예에서, GID는 코드 이미지를 암호화하는데 사용하여 코드 검사(code inspection)를 방지할 수 있다. ROM(115)의 데이터 섹션(117)은 공개 키 인증서(public key certification)와 같은 신뢰 개체로부터의 서명에 기초한 지문을 저장할 수 있다. 한 실시예에서, 개별 장치는 동일 신뢰 개체에 기초한 지문(123)을 포함할 수 있다.

[0033] 도 2는 보안 부팅을 실행하는 시스템 구성요소의 한 실시예를 나타내는 블록도이다. 시스템(100)은 저장 장치 구성요소(109)로부터의 LLB(Low Level Boot) 코드 이미지(229)를 LLB(225)로서 RAM(111)에 로딩할 수 있다. LLB(225)는 시스템(100)의 장기 전력 관리에 관할 것일 수 있다. 한 실시예에서, LLB(225)는 시스템(100) 버전의 식별을 포함할 수 있다. 코드 이미지 LLB(225)는 코드(115)의 실행에 기초하여 로딩될 수 있다. 한 실시예에서, 코드 이미지 LLB(229)는 코드(115)의 실행을 통하여 코드 이미지 LLB(225)에 기초하여 RAM(111)으로부터 저장될 수 있다.

[0034] 한 실시예에 따라, 코드 이미지 iBoot(227)는 LLB(225)의 실행에 따른 코드 이미지 iBoot(231)에 기초하여 저장 장치(109)로부터 메모리 구성요소(111)로 로딩될 수 있다. 코드 이미지 iBoot(231)는 시스템(100)을 수용하는 장치에 동작 환경을 제공하는 운영 체제에 대하여 하드웨어 초기화를 시킬 수 있다. 장치는 성공적인 부팅 후에 동작 환경에 진입할 수 있다. 동작 환경은 장치 내에서 실행되는 여러 사용자 및/또는 시스템 애플리케이션을 지원할 수 있다. 한 실시예에서, 코드 이미지 iBoot(231)는 장치의 대용량 저장 장치 구성요소를 동작시키고, 사용자 인터페이스용 그래픽 구성요소를 초기화시키며/시키거나 장치에 대한 화면 구성요소 등을 활성화시킬 수 있다. 코드 이미지 iBoot(231)는 코드 이미지 LLB(225)의 실행을 통하여 코드 이미지 iBoot(227)에 기초하여 RAM(111)으로부터 저장될 수 있다.

[0035] 한 실시예에서, 코드 이미지 커널캐시(Kernelcache)(223)는 코드 이미지 커널캐시(233)에 기초하여 저장 장치(109)에서 메모리(103)로 로딩될 수 있다. 코드 이미지 커널캐시(223)는 장치에 대한 동작 환경을 지원하는 운

영 체제의 커널의 일부일 수 있다. 한 실시예에서, 코드 이미지 커널캐시(223)는 커널 및 운영 체제 구성요소(235)를 저장 장치(109)로부터 메모리(103)로 로딩시킬 수 있다. 운영 체제 구성요소는 사용자 애플리케이션, 라이브러리, 그래픽 사용자 인터페이스 구성요소 및/또는 사용자 데이터(235)를 포함할 수 있다. 사용자 데이터는 음악, 이미지, 비디오 또는 장치 사용자와 관련된 다른 디지털 콘텐츠를 포함할 수 있다. 예를 들어, 이러한 사용자 데이터는 사용이 제한되는 DRM(digital right management) 부합 데이터일 수 있다. 코드 이미지 커널캐시(223)는 커널 및 운영 체제 구성요소(235)를 메모리(103)에 로딩시킬 수 있다. 한 실시예에서, 코드 이미지 커널캐시(223)는 커널이 메모리(103)에서 실행되기 전에 그 커널이 확실히 신뢰성이 있는지 검증 과정을 거칠 수 있다. 다른 실시예에서, 코드 이미지 커널캐시(223)는 운영 체제가 메모리(103)에서 실행되기 전에 그 운영 체제의 신뢰성 확보를 위한 검증 과정을 거칠 수 있다. 코드 이미지 커널캐시(223)는 UID(119) 또는 지문(123)에 기초하여 운영 체제 구성요소(235)가 신뢰성이 있는지를 결정하도록 실행될 수 있다. 한 실시예에서, 코드 이미지 커널캐시(223)는 GID(121)에 따라 메모리(103)에서 운영 체제 구성요소(235)를 복호화시킬 수 있다. 한 실시예에서, 코드 이미지 커널캐시(223)는 운영 체제 구성요소(235)를 메모리(103)에서 저장 장치(109)로 저장하도록 실행될 수 있다. 코드 이미지 커널캐시(223)는 운영 체제 구성요소(235)를 저장 장치(109)에 저장하기 전에 암호화할 수 있다.

[0036] 한 실시예에서, UID(119)는 특권 모드(privileged mode)로 동작하는 일부 운영 체제 구성요소에 접근할 수 있다. 운영 체제의 커널은 애플리케이션이 특권 모드로 동작하는지에 따라 그 애플리케이션이 애플리케이션에 의한 UID(119)로의 접근을 거부 또는 승인할 수 있다. 한 실시예에서, 운영 체제의 커널은 애플리케이션의 해당 코드 이미지가 적정한 애플리케이션을 포함하는지에 기초하여 애플리케이션이 특권 모드로 동작할 수 있는지를 판단할 수 있다. DRM 시스템은 특권 모드로 동작하여 UID(119)에 기초하는 운영 체제 구성요소(235)의 사용자 데이터로의 접근을 제어할 수 있다. 애플리케이션은 DRM 시스템을 통하여 사용자 데이터에 접근할 수 있다. 일부 실시예에서, 운영 체제의 네트워크 유틸리티가 특권화될 수 있다. 네트워크 유틸리티는 기저 대역 칩과 같은 인터페이스 칩을 통하여 외부 자원과 장치를 연결시킬 수 있다. 다른 실시예에서, 바이러스 방지 소프트웨어는 특권 모드로 동작하는 운영 체제에 의하여 제공될 수 있다.

[0037] 따라서, 시스템 내에서 실행되는 모든 소프트웨어 구성요소는 소프트웨어 구성요소가 어떤 소정의 조건을 충족하지 않으면[예를 들어, 신용 업체(trust vendor)에서 제공하거나 장치의 제조나 소프트웨어 구성요소의 시험과 같은 어떤 환경 하에서], 실행 전에 검증 또는 인증이 행해져야 한다. 한 실시예에서, 시스템에서의 보안 저장 영역의 설정은 소정 조건과 연관될 수 있다. 그 결과, DRM 부합 데이터와 같은 임의의 데이터는 적절한 검증 또는 인증이 없이는 접근이나 손상이 되지 않는다.

[0038] 도 3은 보안 부팅을 수행하는 처리의 한 실시예를 나타내는 흐름도이다. 예를 들어, 처리(300)는 도 1의 시스템(100)에 의하여 수행될 수 있다. 한 실시예에 따르면, 장치의 부팅 처리시, 블록(301)에서 처리(300)의 처리 로직은 ROM 칩에서 명령어를 실행함으로써 장치 내에서 코드 이미지를 찾아낼 수 있다. 명령어는 도 1의 코드(115)에서와 같이 ROM 칩의 코드 섹션으로부터 읽어 올 수 있다. 코드 이미지는 장치의 메모리 구성요소 또는 저장 장치 구성요소에 저장될 수 있다. 메모리 구성요소는 RAM일 수 있다. 저장 장치 구성요소는 장치에 부착되는 플래시 메모리나 대용량 저장 장치일 수 있다. 한 실시예에서, 코드 이미지를 찾을 수 없는 경우, 블록(309)에서 부팅 처리는 인터럽트되고 장치는 DFU(Device Firmware Upgrade) 모드로 진입할 수 있다. 한 실시예에 따르면, 코드 이미지를 성공적으로 찾아낸 경우, 블록(303)에서 처리(300)의 처리 로직은 이 코드 이미지를 메모리에 로딩할 수 있다. 다른 실시예에서, 코드 이미지를 찾아낼 때 이 코드 이미지는 이미 로딩되어 있을 수 있다.

[0039] 한 실시예에 따르면, 블록(305)에서, 처리(300)의 처리 로직은 로딩된 코드 이미지가 도 1의 UID(119)와 같은 장치와 관련된 UID에 기초하여 신뢰성이 있는지를 검증할 수 있다. 처리(300)의 처리 로직은 코드 이미지로부터 헤더값(header value)을 추출할 수 있다. 코드 이미지 내의 헤더값의 위치는 미리 정해질 수 있다. 한 실시예에서, 헤더값은 코드 이미지 내의 속성값 페어(pair)에서의 기설정 속성에 기초하여 추출될 수 있다. 헤더값은 공지의 해싱 및 암호화 알고리즘을 통하여 장치의 UID에 따라 코드 이미지에 서명된 서명값을 포함할 수 있다. 한 실시예에서, 처리(300)의 처리 로직은 블록(305)에서 동일한 공지의 해싱 및 암호화 알고리즘을 통하여 UID에 따라 코드 이미지로부터 또 다른 서명값을 끌어낸다(derive). 처리(300)의 처리 로직은 끌어낸 서명값과 추출된 서명값을 비교하여 코드 이미지가 신뢰성이 있는지 검증한다. 한 실시예에서, 끌어낸 서명값과 추출된 서명값이 서로 일치하는 경우 검증이 성공할 수 있다. 그렇지 않으면 검증은 실패할 수 있다. 검증이 성공하지 못한 경우, 처리(300)의 처리 로직은 블록(309)에서 DFU 모드로 장치를 진입시킬 수 있다. 한 실시예에서, 처리(300)의 처리 로직은 블록(309)에서 장치가 DFU 모드로 진입하기 전에 메모리에서 코드 이미지를 제거

할 수 있다.

[0040] 검증이 성공하면, 처리(300)의 처리 로직은 블록(311)에서 코드 이미지를 실행할 수 있다. 한 실시예에서, 코드 이미지는 도 2에 도시한 225, 227 및 223과 같은 LLB, iBoot 또는 커널캐시일 수 있다. 처리(300)의 처리 로직은 블록(311)에서 장치에 대한 부팅 동작을 수행할 수 있다. 부팅 동작은 제품 식별, 시동 장치 전원 관리, 대용량 저장 장치 구성요소 작동, 사용자 인터페이스용 그래픽 구성요소 초기화, 화면 구성요소 활성화 및/또는 장치 하드웨어 초기화 등을 포함할 수 있다. 한 실시예에서, 부팅 동작은 도 2의 235에 도시한 것과 같은 커널 및 운영 체제 구성요소를 포함하여 메모리에 운영 체제를 로딩시키는 것을 포함할 수 있다. 처리(300)의 처리 로직은 메모리 내의 신뢰된 코드 이미지에 신뢰 표시기를 부착하여 검증이 성공했음을 나타낼 수 있다. 한 실시예에서, 메모리에 위치한 신뢰 표시기와 연관된 코드 이미지는 검증 없이 신뢰받은 코드로서 실행될 수 있다. 블록(313)에서, 처리(300)의 처리 로직은 장치가 완전히 부팅되었는지를 판단할 수 있다. 장치가 완전히 부팅된 경우, 블록(315)에서 장치는 동작 가능하게 되며 정상 동작 가능 모드로 진입한다. 한 실시예에서, 커널캐시(227)는 장치가 정상 동작한 후 사용자 모드에서 실행되는 사용자 애플리케이션을 시작시킬 수 있다. 사용자 모드에서 실행되는 애플리케이션은 도 2의 UID(119)와 GID(121)와 같은 장치 하드웨어 관련 정보에는 접근할 수 없다. 블록(313)에서의 부팅 동작이 실패하는 경우 장치는 DFU 모드로 진입할 수 있다.

[0041] 한 실시예에 따르면, 블록(317)에서, 처리(300)의 처리 로직이 장치 부팅 처리가 블록(313)에서 완료되지 않았다고 판단한 경우 부팅 처리가 계속될 수 있다. 처리(300)의 처리 로직이 현재 코드 이미지의 실행에 기초하여 블록(313)에서 다른 코드 이미지를 찾아낼 수 있다. 한 실시예에서, 코드 이미지 LLB를 실행시킴으로써 도 2에 도시한 것과 같은 코드 이미지 iBoot를 찾아낼 수 있다. 다른 실시예에서, 코드 이미지 iBoot를 실행시킴으로써 도 2에 도시한 것과 같은 코드 이미지 커널캐시를 찾아낼 수 있다. 일부 실시예에서, 코드 이미지 커널캐시를 실행시킴으로써 도 2에 도시한 것과 같은 커널 및 운영 체제 구성요소를 포함하는 코드 이미지를 찾아낼 수 있다. 처리(300)의 처리 로직은 블록(319)으로 귀환하여 블록(317)에서 다음 코드 이미지를 찾아낸 결과에 따라 부팅 처리를 계속한다.

[0042] 도 4는 UID 및 시드열(seed string)에 기초하여 코드 이미지로부터 서명을 생성하는 처리의 한 실시예를 나타내는 흐름도이다. 예를 들어, 처리(400)는 도 1에 도시한 것과 같은 시스템에 의하여 수행될 수 있다. 한 실시예에서, 처리(400)의 처리 로직은 도 2에 도시한 LLB(225), iBoot(227) 또는 커널캐시(223)와 같이, 코드 이미지(411)에 대하여 해싱 동작(409)을 수행한다. 해싱 동작은 암호 해시 기능 SHA-1, SHA-22, SHA-256, SHA-384 및 SHA-512와 같은 SHA(Secure Hashing Algorithm) 해싱 기능에 기초할 수 있다. 한 실시예에서, 해싱 동작(409)은 키열(413)을 생성할 수 있다. 키열(413)은 20 바이트 길이일 수 있다. 한 실시예에서, 처리(400)의 처리 로직은 블록(403)에서 암호화 동작을 수행하여 장치와 연관된 키열(413), UID(401) 및 시드열(407)에 기초하여 서명(405)을 생성한다. 한 실시예에서, 암호화 동작은 블록(403)에서 AES(Advanced Encryption Standard) 알고리즘에 기초할 수 있다. 처리(400)의 처리 로직은 키열(413) 20 바이트 중 4 바이트를 버리는 것과 같이, 블록(403)에서 키열(413)을 절삭(truncate)할 수 있다. 한 실시예에서, AES 알고리즘은 블록(403)에서 16 바이트에 기초할 수 있다. UID(401)는 도 1에 도시한 UID(119)와 같은 장치 내에 저장될 수 있다. 시드열(407)은 장치에 기초한 시드 생성 기능을 통해 생성될 수 있다. 한 실시예에서, 시드열(407)은 시드 생성 기능이 동일 장치에 대하여 적용될 때마다 동일할 수 있다.

[0043] 도 5는 도 1의 시스템에 따른 장치를 호스트가 보안적으로 부팅하기 위한 네트워크 연결의 한 실시예를 나타내는 블록도이다. 한 실시예에서, 장치는 호스트에 연결됨으로써 부팅을 위한 DFU 모드로 진입할 수 있다. 장치는 사용자로부터의 초기화에 기초하여 DFU 모드로 강제 진입할 수 있다. 한 실시예에서, 장치는 사용자가 장치의 버튼을 누르는 것과 같이 소정 동작을 행하는 것에 응답하여 DFU 모드로 진입할 수 있다. 사용자는 예를 들어, 사용자 데이터 클리닝, 하드웨어 드라이버 업그레이드, 사용자 애플리케이션 업그레이드 및/또는 새로운 애플리케이션 설치 등을 포함하는, 장치에 대한 시스템 관리 태스크를 수행하기 위하여 장치에 대하여 DFU 모드로 진입하도록 요청할 수 있다. 장치는 도 3의 블록(309)에 도시한 것처럼 부팅 과정 중 적어도 한 단계에서 부팅이 실패할 경우 자동으로 DFU 모드로 진입할 수 있다. 이와는 달리, 장치는 오류 데이터나 손상된 소프트웨어 구성요소가 검출되는 경우와 같이 정상 동작 중에 운영 체제가 비정상인 상황을 맞이할 때 DFU 모드로 진입할 수 있다.

[0044] 한 실시예에 따르면, 네트워크(500)는 호스트(503)와 연결되는 장치(501)를 포함할 수 있다. 장치(501)는 예를 들어, 연결된 호스트(503)로부터 운영 체제 구성요소를 복구하는 복구 데몬 애플리케이션(restoring daemon application)을 실행시키는 애플 컴퓨터사의 아이팟(iPod)과 같은 미디어 플레이어일 수 있다. 장치(501)는 접속 인터페이스 지원 TCP/IP 프로토콜을 통하여 호스트(503)와 연결될 수 있다. 접속 인터페이스는 USB, 무선

네트워크 또는 이더넷등에 기초할 수 있다. 한 실시예에서, 호스트(503)는 애플 컴퓨터사의 아이튠(iTune)과 같은 맥(Mac) 또는 윈도우 기반 컴퓨터 실행 애플리케이션 소프트웨어일 수 있다. 호스트(503)는 WAN(wide area network)(예: 인터넷) 또는 LAN(local area network)(예: 인트라넷 또는 피어 투 피어 네트워크)와 같은 네트워크(505)를 통하여 중앙 서버(507)에 연결될 수 있다. 한 실시예에서, 중앙 서버(507)는 공개적으로 접근 가능한 웹 서버에 기초할 수 있다. 이와는 달리, 서버(507)는 인트라넷 또는 로컬 서버일 수 있다.

[0045] 도 6은 호스트로부터 장치로 동작 환경을 보안적으로 복원하는 처리의 한 실시예를 나타내는 흐름도이다. 예를 들어, 처리(600)는 도 1 및/또는 도 5에 도시한 것과 같은 시스템에 의하여 수행될 수 있다. 한 실시예에서, 처리(600)의 처리 로직은 블록(601)에서 장치가 복원 모드(recovery mode)임을 나타내는 상태를 호스트 컴퓨터로 전송한다. 장치는 코드 이미지 검증 실패에 응답하여 복원 모드로 진입할 수 있다. 호스트 컴퓨터는 도 5에 도시한 것과 같이 처리(600)를 수행하는 장치에 연결될 수 있다. 한 실시예에서, 상태는 제품 ID 및/또는 업체 ID를 포함할 수 있다. 호스트 컴퓨터는 코드 이미지를 준비하여 수신되는 상태에 기초하여 연결된 장치를 복원시킬 수 있다. 한 실시예에서, 코드 이미지는 도 5에 도시한 네트워크(505)와 같은 네트워크를 통하여 연결되는 호스트 컴퓨터에 의하여 중앙 서버 컴퓨터로부터 검색될 수 있다. 한 실시예에 따르면, 블록(603)에서 처리(600)의 처리 로직은 호스트 컴퓨터로부터 도 1에 도시한 메모리(103)와 같이 장치의 메모리 구성요소로 코드 이미지를 수신할 수 있다. 처리(600)의 처리 로직은 블록(605)에서 호스트 컴퓨터로부터 명령을 수신하여 수신된 코드 이미지를 실행할 수 있다. 한 실시예에서, 처리(600)는 맥 기반 컴퓨터에서 실행되는 아이튠과 같이 호스트 컴퓨터에서 실행되는 복원 소프트웨어에 의하여 제어될 수 있다.

[0046] 한 실시예에 따르면, 블록(607)에서, 처리(600)의 처리 로직은 장치의 메모리에 수신된 코드 이미지에 첨부된 증명서를 추출할 수 있다. 코드 이미지는 도 2에 도시한 것과 같은 LLB, iBoot 및/또는 커널캐시일 수 있다. 코드 이미지는 RSA(Ralph Shamir Adelman) 공개 키 암호 작성법(cryptography)과 같은 공개 키 암호 작성법에 따라 암호화될 수 있다. 증명서는 X.509 표준에 기초한 키를 포함할 수 있다. 블록(609)에서, 처리(600)의 처리 로직은 도 1에 도시한 코드(115)와 같이 장치의 보안 ROM에 저장된 코드에 따라 증명서를 검증할 수 있다. 한 실시예에서, 처리(600)의 처리 로직은 일련의 증명서들을 증명하여 기본 증명서(root certificate)를 갖는 추출 증명서를 그 일련의 증명서들 내의 마지막 증명서로서 검증할 수 있다. 처리(600)의 처리 로직은 연결된 호스트 컴퓨터에서 증명서들을 검색할 수 있다. 한 실시예에서, 기본 증명서는 도 1에 도시한 지문(123)과 같이, 장치의 보안 ROM에 저장되어 있는 지문에 기초하여 검증될 수 있다. 기본 증명서는 애플 컴퓨터사가 발행할 수 있다. 검증이 실패하면, 처리(600)의 처리 로직은 장치를 다시 DFU 모드로 돌아가게 하여 블록(613)에서 복원되도록 한다.

[0047] 코드 이미지의 증명서가 성공적으로 검증되는 경우, 처리(600)의 처리 로직은 블록(615)에 복원 처리를 계속하여 검증된 증명서에 포함된 키에 기초하여 코드 이미지를 복호화한다. 블록(617)에서, 처리(600)의 처리 로직은 도 1에 도시한 UID(119)와 같이 장치의 보안 ROM에 저장된 UID에 기초하여 코드 이미지에서 해시 서명을 끌어낼 수 있다. 한 실시예에서, 해시 서명은 예를 들어 도 4에 도시한 것과 같은 처리에 따라 얻어질 수 있다. 블록(619)에서, 처리(600)의 처리 로직은 끌어낸 서명을 코드 이미지에 서명할 수 있다. 한 실시예에서, 끌어낸 서명을 코드 이미지의 헤더값으로 서명할 수 있다. 처리(600)의 처리 로직은 블록(621)에서, 예를 들어 도 1에 도시한 저장 장치(109)와 같이 장치의 저장 장치에 서명된 코드 이미지를 저장할 수 있다. 한 실시예에서, 서명된 코드 이미지를 저장하여 장치에서 검증이 실패된 다른 코드 이미지를 리페어(repair)할 수 있다. 한 실시예에서, 코드 이미지는 장치의 저장 장치에 저장되기 전에 실행될 수 있다. 다른 실시예에서, 코드 이미지가 성공적으로 실행된 뒤에 장치의 저장 장치에 저장될 수 있다.

[0048] 도 7은 호스트로부터 장치로 동작 환경의 보안 복원을 수행하는 처리의 한 실시예를 나타내는 상태도이다. 예를 들어, 상태(700)는 도 1 및/또는 도 5에 도시한 것과 같은 시스템의 어떤 동작 상태를 나타낼 수 있다. 한 실시예에서, 장치는 부팅 처리를 시작하는 초기 상태 Boot(701)에 진입할 수 있다. 장치의 보안 ROM에 저장된 명령은 상태 Boot(701)시 실행될 수 있다. 한 실시예에서, 상태 Boot(701)시, 도 2에 도시한 LLB(229)와 같이 저레벨 부팅 프로그램을 장치 내에서 찾을 수 있다. 저레벨 부팅 프로그램을 찾아내어, 장치의 메모리에 로딩시킬 수 있다. 한 실시예에서, 찾아낸 저레벨 부팅 프로그램은 도 3의 블록(305)에서 설명한 것과 같은 처리에 따라 신뢰성있는 코드 이미지인 것으로 검증될 수 있다. 저레벨 부팅 프로그램을 성공적으로 찾아내어 검증한 경우, 상태(700)는 천이 Success(711)에 따라 상태 Boot(701)에서 상태 LLB(703)로 진입할 수 있다. 그렇지 않으면, 한 실시예에 따라, 상태(700)는 장치가 DFU 모드로 진입하면서 천이 DFU(713)를 거쳐서 상태 Recovery1(717)로 진입할 수 있다.

[0049] 상태 Recovery1(717)에서, 장치는 호스트 컴퓨터에 연결되어 도 5에 도시한 것과 같은 복원 처리를 수행할 수

있다. 한 실시예에서, 장치는 상태 Recovery1(717)에 기초하여 상태를 공개할 수 있다. 호스트 컴퓨터는 장치로부터 수신한 상태에 대응되는 코드 이미지를 보낼 수 있다. 한 실시예에서, 코드 이미지는 도 2에 도시한 것과 같은 LLB(229)일 수 있다. 장치는 일련의 증명을 수행하여 도 1의 UID(119)와 지문(123)과 같이 장치의 보안 ROM에 저장된 UID 및 지문에 기초하여, 수신된 코드 이미지가 신뢰성이 있는지를 검증할 수 있다. 일련의 증명은 도 6에 도시한 블록(609)에서의 처리(600)와 유사한 처리에 기초하여 행해질 수 있다. 코드 이미지를 성공적으로 찾아내어 검증한 경우, 한 실시예서, 장치의 상태는 상태 Recovery1(717)에서 천이 Load(715)를 거쳐 상태 LLB(703)로 천이할 수 있다.

[0050] 한 실시예에서, 상태 LLB(701) 동안, 장치는 검증된 저레벨 부팅 프로그램(예를 들어, LLB 또는 전술한 저레벨 라이브러리)을 실행하여 도 2에 도시한 것과 같은 iBoot(231)와 같은 다른 부팅 프로그램을 장치 내에서 찾아낼 수 있다. 부팅 이미지를 찾아 내어, 상태 LLB(701)에서 장치의 메모리 구성요소에 로딩시킬 수 있다. 한 실시예에서, 부팅 이미지는 도 3의 블록(305)에서 전술한 것과 같은 처리에 따라 신뢰성 있는 코드 이미지인 것으로 검증될 수 있다. 부팅 이미지를 성공적으로 찾아 내어 검증한 경우, 상태(700)는 상태 LLB(703)에서 상태 iBoot(705)로 진입할 수 있다. 이와는 달리, 한 실시예에 따라, 상태(700)는 장치가 DFU 모드로 진입하면 상태 Recovery2(719)로 진입할 수 있다.

[0051] 상태 Receovery2(719)에서, 장치는 호스트 컴퓨터와 연결되어 도 5에 도시한 것과 같은 복원 처리를 수행할 수 있다. 한 실시예에서, 장치는 상태 Recovery2(719)에 기초하여 상태를 공개할 수 있다. 호스트 컴퓨터는 상태 Recovery2(719)에서 장치로부터 수신한 상태에 대응되는 코드 이미지를 보낼 수 있다. 한 실시예에서, 코드 이미지는 도 2에 도시한 것과 같은 iBoot(231)일 수 있다. 장치는 일련의 증명을 수행하여 도 1의 UID(119)와 지문(123)과 같이 장치의 보안 ROM에 저장된 UID 및 지문에 기초하여, 수신된 코드 이미지가 신뢰성이 있는지를 검증할 수 있다. 일련의 증명은 도 6에 도시한 블록(609)에서의 처리(600)와 유사한 처리에 기초하여 행해질 수 있다. 코드 이미지를 성공적으로 찾아 내어 검증한 경우, 한 실시예서, 장치의 상태는 상태 Recovery2(719)에서 상태 커널캐시(707)로 천이할 수 있다.

[0052] 한 실시예에 따르면, 상태 iBoot(705) 동안, 장치는 검증된 부팅 프로그램을 실행하여 도 2에 도시한 것과 같은 커널캐시(233)와 같은 커널 이미지를 장치 내에서 찾아낼 수 있다. 커널 이미지는 상태 iBoot(705) 동안 찾아져서, 장치의 메모리 구성요소에 로딩될 수 있다. 한 실시예에서, 커널 이미지는 도 3의 블록(305)에서 설명한 것과 같은 처리에 따라 신뢰성 있는 코드로서 검증될 수 있다. 커널 이미지를 성공적으로 찾아 내어 검증한 경우, 상태(700)는 상태 iBoot(705)에서 커널캐시(707)로 진입할 수 있다. 그렇지 않으면, 한 실시예에 따라 상태(700)는 장치가 DFU 모드로 진입하면 상태 Recovery3(721)로 진입할 수 있다.

[0053] 상태 Recovery3(721) 동안, 장치는 호스트 컴퓨터와 연결되어 도 5에 도시한 것과 같은 복원 처리를 수행할 수 있다. 한 실시예에서, 장치는 상태 Recovery3(721)에 기초하여 상태를 공개할 수 있다. 호스트 컴퓨터는 상태 Recovery3(721)에서 장치로부터 수신한 상태에 대응되는 코드 이미지를 보낼 수 있다. 한 실시예에서, 코드 이미지는 도 2의 커널캐시(233)와 같은 커널 이미지일 수 있다. 장치는 일련의 증명을 수행하여 도 1의 UID(119)와 지문(123)과 같이 장치의 보안 ROM에 저장된 UID 및 지문에 기초하여, 수신된 코드 이미지가 신뢰성이 있는지를 검증할 수 있다. 일련의 증명은 도 6에 도시한 블록(609)에서의 처리(600)와 유사한 처리에 기초하여 행해질 수 있다. 코드 이미지를 성공적으로 찾아 내어 검증한 경우, 한 실시예서, 장치의 상태는 상태 Recovery3(721)에서 상태 커널캐시(707)로 천이할 수 있다.

[0054] 한 실시예에서, 상태 커널캐시(707) 동안, 장치는 검증된 커널 이미지를 실행하여 도 2의 235와 같은 운영 체제 구성요소를 찾아낼 수 있다. 상태 커널캐시(707) 동안 검증된 커널 이미지의 실행에 따라, 찾아낸 운영 체제 구성요소를 장치의 메모리 구성요소에 로딩하여 신뢰성이 있는 것으로 검증할 수 있다. 한 실시예에서, 커널 이미지는 도 3의 블록(305)에서 설명한 것과 같은 처리에 따라 운영 체제 구성요소가 신뢰성이 있는지 판단할 수 있다. 커널 이미지에 기초하여 특권 모드를 신뢰된 운영 체제 구성요소에 할당하여 도 2의 UID(119) 또는 GID(123)와 같은, 장치의 하드웨어 레벨 인터페이스에 접근하게 할 수 있다. 사인된(signed) 서명이 없는 운영 체제 구성요소는 상태 커널캐시(707) 동안에는 사용자 모드 특권을 할당 받을 수 있다. 한 실시예에서, 운영 체제 구성요소는 장치의 하드웨어 레벨 인터페이스로의 접근이 허용되지 않을 수 있다. 운영 체제를 장치의 메모리에 성공적으로 로딩시킨 경우, 상태(700)는 상태 커널캐시(707)에서 정상 동작 환경에 해당되는 상태 OS(709)로 천이할 수 있다. 사용자 애플리케이션은 상태 OS(709) 동안 할당 받은 사용자 모드에서 실행을 시작할 수 있다. 한 실시예에서, 상태 커널캐시(707)에서 장치는 DFU 모드로 진입하여 연결된 호스트 컴퓨터로부터 기본 이미지를 수신하고 장치에 대한 운영 체제 구성요소를 복구 또는 업데이트한다.

[0055] 도 8은 호스트로부터 장치로 소프트웨어 구성요소를 보안적으로 복구하는 처리의 한 실시예를 나타내는 흐름도이다. 예를 들어, 처리(800)는 도 1 및/또는 도 5에 도시한 것과 같은 시스템에 의하여 수행될 수 있다. 한 실시예에서, 처리(800)의 처리 로직은 블록(801)에서 장치를 부팅 장치로서 구성할 수 있다. 부팅 장치는 DFU 모드에 있을 수 있다. 사용자는 장치의 정상 동작시 장치의 버튼을 눌러 부팅 장치를 DFU 모드로 구성시킬 수 있다. 처리(800)의 처리 로직은 장치 사용자에 의하여 의도적으로 활성화되어 손상된 애플리케이션 소프트웨어의 리페어, 펌웨어 구성요소 설치 또는 장치에 저장된 기존 사용자 데이터를 관리할 수 있다. 블록(803)에서, 한 실시예에 따르면, 처리(800)의 처리 로직은 호스트 컴퓨터와 네트워크 연결을 구축할 수 있다. 장치와 호스트 컴퓨터는 도 5에 도시한 것과 같은 네트워크 인터페이스를 통하여 연결될 수 있다. 애플 컴퓨터사의 아이튠과 같은 복구 소프트웨어는 장치와 통신하는 호스트 컴퓨터 상에서 실행될 수 있다. 처리(800)의 처리 로직은 블록(805)에서 상태를 호스트 컴퓨터에 공개하여 장치가 복구 모드에 있는지 네트워크 연결을 통하여 식별할 수 있다. 복구 모드의 장치는 또한 DFU 모드일 수 있다. 한 실시예에서, 상태는 장치 ID 및/또는 제품 ID와 같은 정보를 포함할 수 있다. 상태는 호스트 컴퓨터로부터 필요한 코드 이미지의 표시를 포함할 수 있다.

[0056] 한 실시예에 따르면, 블록(807)에서, 처리(800)의 처리 로직은 연결된 호스트 컴퓨터로부터 부팅 이미지를 수신할 수 있다. 부팅 이미지는 도 2에 도시한 LLB(229) 또는 iBoot(231)와 같은 부팅 로더(boot loader)를 포함할 수 있다. 한 실시예에서, 부팅 이미지는 도 2의 커널캐시(233)과 같은 커널 캐시를 포함할 수 있다. 부팅 이미지는 블록(805)에서 호스트 컴퓨터에 공개된 상태에 기초하여 수신될 수 있다. 한 실시예에서, 부팅 이미지는 도 1의 메모리(103)와 같은 장치의 메모리 구성요소에 로딩될 수 있다. 처리(800)의 처리 로직은 블록(809)에서 연결된 호스트 컴퓨터로부터 기본 이미지를 수신할 수 있다. 기본 이미지는 장치에 대한 운영 체제의 스트립트 다운 버전(stripped down version)의 RAM일 수 있다. 한 실시예에서, 기본 이미지는 복구 애플리케이션을 포함할 수 있다.

[0057] 한 실시예에 따르면, 블록(811)에서, 처리(800)의 처리 로직은 연결된 컴퓨터로부터 커맨드를 수신하여 수신된 부팅 이미지를 실행시킬 수 있다. 부팅 이미지는 부팅 로더일 수 있다. 처리(800)의 처리 로직은 이에 응답하여 블록(813)에서 부팅 이미지가 신뢰성이 있는지 검증할 수 있다. 한 실시예에서, 처리(800)의 처리 로직은 도 6에 도시한 것과 같은 처리를 수행하여 부팅 이미지가 도 1의 칩(105)과 같은 보안 ROM 칩에 기초하여 신뢰할 수 있는지를 판단할 수 있다. 한 실시예에서, 처리(800)의 처리 로직은 블록(815)에서 연결된 호스트 컴퓨터로부터 수신한 커널캐시에 대하여 신뢰된 부팅 이미지를 실행함으로써 신뢰성이 있는지 검증할 수 있다. 처리(800)의 처리 로직은 도 6에 도시한 것과 같은 처리를 수행하여 커널캐시가 도 1의 지문(123)과 같이 장치에 저장된 기본 증명 지문에 기초하여 신뢰할 수 있는지를 판단할 수 있다. 블록(817)에서, 처리(800)의 처리 로직은 기본 이미지로부터의 복구 데몬 애플리케이션에 대하여 신뢰된 커널캐시를 실행하여 신뢰성이 있는지를 검증할 수 있다. 한 실시예에서, 처리(800)의 처리 로직은 기본 이미지가 신뢰성있는 코드 이미지인지를 검증하여 복구 데몬 애플리케이션을 신뢰할 수 있는지를 판단할 수 있다. 처리(800)의 처리 로직은 도 6에 도시한 것과 같은 처리를 수행하여 기본 이미지에 포함된 복구 데몬 애플리케이션을 신뢰할 수 있는지 판단한다.

[0058] 한 실시예에 따르면, 블록(819)에서 처리(800)의 처리 로직은 복구 데몬 애플리케이션을 통하여 호스트 컴퓨터로부터의 커맨드 콜(command call)을 수신하여 소프트웨어 복구 동작을 수행할 수 있다. 한 실시예에서, 소프트웨어 복구 동작은 대용량 저장 장치의 파일 시스템의 분할 및 포맷팅(formatting), 장치 레벨 재저장 또는 신 펌웨어의 장치로의 로딩을 포함할 수 있다. 처리 로직은 기본 이미지에 포함되어 있는 OS를 기동시켜 복구 데몬을 장치에 론칭(launching)할 수 있다. 한 실시예에서, OS의 감소 부분 또는 최소 부분만이 기동된다. 이 데몬 애플리케이션은 XML(Extensible Markup Language) 프로토콜에 기초하여 연결된 호스트 컴퓨터에서 실행되는 복구 소프트웨어와 통신할 수 있다. 한 실시예에서, 복구 데몬은 호스트 컴퓨터에서 실행되는 복구 소프트웨어로 하여금 장치가 실행하는 임의의 커맨드를 발행하도록 할 수 있다. 커맨드는 RAM 디스크에 포함된 보조 툴의 실행 및/또는 라이브러리의 콜을 포함할 수 있다. 한 실시예에서, 커맨드는 대용량 저장 장치에 저장된 소프트웨어 전체 세트 및 장치의 프로그래머블 ROM을 대체하게 할 수 있다. 블록(821)에서, 처리(800)의 처리 로직은 연결된 호스트 컴퓨터로부터 커맨드를 수신하여 장치를 다시 시작하게 할 수 있다. 이에 응답하여, 처리(800)의 처리 로직은 장치를 리셋시킬 수 있다. 이어서, 장치는 장치의 대용량 저장 장치에 저장된 운영 체제로부터 재부팅될 수 있다.

[0059] 도 9는 호스트에서 장치로 애플리케이션을 보안적으로 업데이트하는 처리의 한 실시예를 나타내는 흐름도이다. 예를 들어, 처리(900)는 도 1 및/또는 도 5에 도시한 것과 같은 시스템에 의하여 수행될 수 있다. 처리(900)의 처리 로직은 블록(901)에서 호스트 컴퓨터와 네트워크 연결을 구축할 수 있다. 장치와 호스트 컴퓨터는 도 5에 도시한 것과 같은 네트워크 인터페이스를 통하여 연결될 수 있다. 애플 컴퓨터사의 아이튠과 같은 업데이트 소

프트웨어는 장치와 통신하도록 호스트 컴퓨터에서 실행될 수 있다. 처리(800)의 처리 로직은 블록(803)에서 상태를 호스트 컴퓨터에 공개하여 장치가 업데이트 모드에 있는지를 네트워크 연결을 통하여 식별할 수 있다. 업데이트 모드의 장치는 또한 DFU 모드일 수 있다. 한 실시예에서, 상태는 장치 ID 및/또는 제품 ID와 같은 정보를 포함할 수 있다. 상태는 장치 내에 현재 포함된 애플리케이션의 버전 ID의 표시를 포함할 수 있다.

[0060] 한 실시예에 따르면, 블록(905)에서, 처리(900)의 처리 로직은 연결된 호스트 컴퓨터로부터 코드 이미지를 수신할 수 있다. 코드 이미지는 블록(903)에서 호스트 컴퓨터가 수신하는 공개 상태로부터의 버전 ID에 기초하여 애플리케이션의 업데이트 버전에 관련된 소프트웨어 패키지를 포함할 수 있다. 한 실시예에서, 코드 이미지는 도 1에 도시한 메모리(103)와 같은 장치의 메모리 구성요소에 로딩될 수 있다. 한 실시예에 따르면, 블록(907)에서, 처리(900)의 처리 로직은 코드 이미지의 신뢰성에 대하여 검증할 수 있다. 처리(900)의 처리 로직은 도 6에 도시한 것과 같은 처리를 수행하여 코드 이미지가 도 1에 도시한 칩(105) 내의 지문(123)과 같이 보안 ROM 칩의 기본 증명서의 지문에 기초하여 신뢰할 수 있는지를 판단할 수 있다. 한 실시예에서, 처리(900)의 처리 로직은, 블록(909)에서 검증된 코드 이미지를 실행하여 포함된 소프트웨어 패키지에서 파일을 풀고 장치의 파일 시스템 내에 그 파일을 위치시킨다. 소프트웨어 패키지로부터의 파일은 새로운 파일 또는 장치에 대한 기존 파일의 업데이트 버전일 수 있다. 처리(900)의 처리 로직은 소프트웨어 패키지로부터의 파일에 대한 완전성 검사를 하여 그 파일을 장치의 파일 시스템에 위치시키기 전에 파일이 손상되거나 오류가 없도록 확실히 한다. 한 실시예에서, 파일의 완전성은 파일 컨텐츠 상의 해시에 따른 서명에 기초하여 검사될 수 있다. 블록(911)에서, 처리(900)의 처리 로직은 장치를 리셋하여 장치 내에 저장된 운영 체제로부터 재부팅시킨다.

[0061] 도 10은 미검증 코드 이미지를 실행하는 처리의 한 실시예를 나타내는 흐름도이다. 예를 들어, 처리(1000)는 도 1에 도시한 것과 같은 시스템에 의하여 수행될 수 있다. 블록(1001)에서, 처리(1000)의 처리 로직은 도 1의 UID(119)와 같은, 장치의 보안 ROM의 UID로의 접근을 못하게 한다. 한 실시예에서, 신뢰된 코드 이미지는 실행될 때 UID로의 접근을 차단하도록 구성될 수 있다. 다른 실시예에서, 하드웨어 스위치는 UID로의 접근을 차단하는 설정을 포함할 수 있다. UID의 접근 구성은 장치의 진단이나 시험 요건에 따라 특정될 수 있다. 신뢰된 코드 이미지는 도 1의 코드(115)와 같이 장치의 보안 ROM 내의 코드에 의하여 검증된 부팅 이미지일 수 있다. 한 실시예에서, 검증은 도 6에 도시한 것과 유사한 처리로 행해질 수 있다. 부팅 이미지는 도 2에 도시한 것과 같은 LLB(225) 또는 iBoot(227)일 수 있다. 블록(1003)에서, 처리(1000)의 처리 로직은 도 1의 RAM(111)과 같은 장치의 메모리 구성요소에 코드 이미지를 로딩할 수 있다. 한 실시예에서, 처리(1000)의 처리 로직은 현재 실행 중인 신뢰된 코드 이미지에 기초하여 코드 이미지를 로딩할 수 있다. 코드 이미지는 외부의 네트워크 연결이나 장치에 연결된 대용량 저장 장치로부터 로딩될 수 있다. 한 실시예에서, 코드 이미지는 장치에 대한 진단 소프트웨어를 포함할 수 있다.

[0062] 블록(1005)에서, 처리(1000)의 처리 로직은 프로그래밍 인터페이스를 활성화시켜 코드 이미지를 실행함으로써 장치 하드웨어에 접근할 수 있다. 장치 하드웨어는 장치 하드웨어 파라미터의 값을 판독하거나 설정함으로써 접근 가능하다. 처리 로직은 로딩된 코드 이미지로부터 해시값을 끌어내어 코드 이미지가 손상되지 않았는지(예를 들어, 오류가 없는지)를 판단할 수 있다. 이 판단은 끌어낸 해시값과 코드 이미지로부터의 헤더값의 비교에 기초할 수 있다. 한 실시예에서, 블록(1007)에서, 처리(1000)의 처리 로직은 UID가 비활성(inactive)인지를 판단할 수 있다. 장치 하드웨어에 접근하는 프로그래밍 인터페이스는 도 1의 코드(115)와 같이 보안 ROM 내의 코드를 실행하여 UID의 활성 여부를 판단할 수 있다. 블록(1009)에서, 처리(1000)의 처리 로직은 장치 하드웨어로의 접근 없이 계속하여 코드 이미지를 실행시킨다. 한 실시예에서, 장치 하드웨어로의 접근은 해당 UID의 활성 여부에 따라 장치의 보안 ROM 내의 코드에 의하여 제어될 수 있다. 다른 실시예에서, 사용자 데이터는 UID가 비활성일 때에는 접근 불가할 수 있다. 미검증 애플리케이션이 장치에 로딩되고 실행되는 경우일지라도, UID가 비활성이라면 어떠한 장치 하드웨어 또는 사용자 감응 데이터(user sensitive data)도 손상되지 않을 수 있다.

[0063] 도 11은 본 발명의 한 실시예와 함께 사용될 수 있는 데이터 처리 시스템의 한 예를 나타낸다. 예를 들어, 시스템(1100)은 도 5에 도시한 것과 같은 호스트를 포함하여 구현될 수 있다. 도 11은 컴퓨터 시스템의 다양한 구성요소를 나타내지만, 이와 같이 상세한 구성요소의 특정 구조나 상호 연결 방식이 본 발명과 밀접한 관계가 있는 것을 나타내고자 하는 것이 아님을 유의하여야 한다. 또한, 더 적거나 많은 구성요소를 구비한 네트워크 컴퓨터와 다른 데이터 처리 시스템도 본 발명과 사용될 수 있음을 인식할 것이다.

[0064] 도 11에 도시한 것처럼, 데이터 처리 시스템의 형태인 컴퓨터 시스템(1100)은 마이크로프로세서(들)(1105), ROM(1107), 휘발성 RAM(1109) 및 불휘발성 메모리(1111)에 연결되어 있는 버스(1103)를 포함한다. 마이크로프로세서(1105)는 메모리(1107, 1109, 1111)로부터의 명령을 검색하고 그 명령을 실행하여 전술한 동작을 행한다.

버스(1103)는 이러한 여러 구성요소를 서로 연결시키는 것은 물론, 이들 구성요소(1105, 1107, 1109, 1111)를 표시 제어기 및 표시 장치(1113)와, 마우스, 키보드, 모뎀, 네트워크 인터페이스, 프린터 및 기타 공지의 다른 장치일 수 있는 입출력(I/O) 장치에 서로 연결시킨다. 일반적으로, 입출력 장치(1115)는 입출력 제어기(1117)를 통하여 시스템에 연결되어 있다. 휘발성 RAM(1109)은 메모리에 데이터를 리프레시하거나 유지하기 위하여 전원을 지속적으로 필요로 하는 동적 RAM(DRAM)으로 일반적으로 구현된다.

[0065] 대용량 저장 장치(1111)는 통상적으로 자기 하드 드라이브, 자기 광 드라이브, 광 드라이브, DVD RAM, 플래시 메모리 또는 전원이 시스템에서 제거된 후에도 데이터(예: 대용량의 데이터)를 유지하는 다른 유형의 메모리 시스템이다. 일반적으로, 이것이 필요하지 않을 지라도 대용량 저장 장치(1111)는 랜덤 액세스 메모리일 수 있다. 도 11은 대용량 저장 장치(1111)가 데이터 처리 시스템에서 나머지 구성요소에 직접 연결되어 있는 로컬 장치임을 나타내지만, 본 발명은 시스템과 이격되어 있는 불휘발성 메모리, 예를 들어, 모뎀, 이더넷 인터페이스 또는 무선 네트워크와 네트워크 인터페이스를 통하여 데이터 처리 시스템에 연결되는 네트워트 저장 장치를 이용할 수 있다. 버스(1103)는 공지된 바와 같이 여러 브리지, 제어기 및/또는 어댑터(adapter)를 통하여 서로 연결되는 하나 이상의 버스를 포함할 수 있다.

[0066] 도 12는 본 발명의 한 실시예와 함께 사용될 수 있는 다른 데이터 처리 시스템의 한 예를 나타낸다. 예를 들어, 시스템(1200)은 도 1에 도시한 것과 같은 시스템의 일부로서 구현될 수 있다. 도 12에 도시한 데이터 처리 시스템(1200)은 하나 이상의 마이크로프로세서 또는 칩이 집적된 회로 상의 시스템일 수 있는 처리 시스템(1211)을 포함하고, 또한 처리 시스템에 의하여 실행되는 데이터 및 프로그램을 저장하는 메모리(1201)를 포함한다. 시스템(1200)은 또한 마이크로폰과 스피커를 포함하여 음악을 재생하거나 그 스피커와 마이크로폰을 통하여 전화 기능을 제공한다.

[0067] 표시 제어기 및 표시 장치(1207)는 사용자에 시각적인 사용자 인터페이스를 제공하고, 이러한 디지털 인터페이스는 OS X 운영 체제 소프트웨어가 실행될 때 MAC킨토시 컴퓨터 상에 나타나는 것과 유사한 그래픽 사용자 인터페이스를 포함할 수 있다. 또한, 시스템(1200)은 도 11의 시스템(1100)과 같이 다른 데이터 처리 시스템과 통신하는 하나 이상의 무선 트랜시버(1203)를 포함한다. 무선 트랜시버는 WiFi 트랜시버, 적외선 트랜시버, 블루투스 트랜시버 및/또는 무선 셀룰러 전화 트랜시버일 수 있다. 어떤 실시예에서는 도시하지 않은 부가적인 구성요소가 시스템(1200)의 일부가 될 수 있고, 어떤 실시예에서는 도 12에 도시한 것보다 적은 구성요소가 데이터 처리 시스템에 사용될 수도 있음을 인식할 것이다.

[0068] 데이터 처리 시스템(1200)은 또한 사용자가 시스템에 입력을 제공하게 하는 하나 이상의 입력 장치(1213)를 포함한다. 입력 장치는 키패드, 키보드, 터치 패널 또는 멀티 터치 패널일 수 있다. 데이터 처리 시스템(1200)은 또한 도크(dock)에 대한 커넥터일 수 있는 입출력 장치(1215)를 선택적으로 포함한다. 도시하지 않은 하나 이상의 버스가 공지된 바와 같이 여러 구성요소를 서로 연결시키는데 사용될 수 있음을 인식할 것이다. 도 12에 도시한 데이터 처리 시스템은 핸드헬드 컴퓨터(handheld computer), PDA(personal digital assistant), PDA 유사 기능을 가진 셀룰러 전화, 아이팟과 같은 미디어 플레이어, 또는 한 장치에 PDA 및 셀룰러 전화와 결합된 미디어 플레이어와 같이 이들 장치의 양상 또는 기능을 결합시키는 장치일 수 있다. 다른 실시예에서, 데이터 처리 시스템(1200)은 네트워크 컴퓨터, 다른 장치에 내장된 처리 시스템, 또는 도 12에 도시한 것보다 적거나 많을 수 있는 구성요소를 구비한 다른 유형의 데이터 처리 시스템일 수 있다.

[0069] 본 발명의 적어도 어떤 실시예는 휴대용 음악 및/또는 비디오 미디어 플레이어와 같은 디지털 미디어 플레이어의 일부일 수 있으며, 미디어를 표시하는 미디어 처리 시스템과 미디어를 저장하는 저장 장치를 포함할 수 있고 안테나 시스템과 미디어 처리 시스템에 연결되는 RF(radio frequency) 트랜시버(예를 들어, 셀룰러 전화용 RF 트랜시버)를 더 포함할 수 있다. 어떤 실시예에서, 원격 저장 장치에 저장되어 있는 미디어를 RF 트랜시버를 통하여 미디어 플레이어로 전송할 수 있다. 미디어는 예를 들어, 하나 이상의 음악 또는 다른 오디오, 정지 영상 또는 동영상일 수 있다.

[0070] 휴대용 미디어 플레이어는 캘리포니아, Cupertino 소재 애플 컴퓨터사의 iPod® 또는 iPod Nano® 미디어 플레이어의 클릭 휠 입력 장치, 터치 스크린 입력 장치, 푸시 버튼 장치, 이동 가능한 포인팅 입력 장치 또는 기타 입력 장치와 같은 미디어 선택 장치를 포함할 수 있다. 미디어 선택 장치는 저장 장치 및/또는 원격 저장 장치에 저장된 미디어를 선택하는데 사용될 수 있다. 휴대용 미디어 플레이어는 적어도 어떤 실시예에서 미디어 처리 시스템에 연결되어, 입력 장치를 통하여 선택되고 스피커나 이어폰을 통하여 또는 표시 장치 또는 표시 장치와 스피커나 이어폰 상에서 제공되는 미디어의 제목 또는 다른 표시자를 표시하는 표시 장치를 포함한다. 휴대용 미디어 플레이어의 예는 미국 공개 특허 출원 번호 제2003/0095096호와 2004/0224638호에 기재되어 있으며

이 둘은 참고로 여기에 편입된다.

[0071] 전술한 것의 일부분은 전용 로직 회로와 같은 로직 회로, 마이크로 제어기 또는 프로그램 명령어 코드를 실행하는 다른 형태의 처리 코어(processing core)로 구현될 수 있다. 따라서, 상기 전술한 기재가 교시하는 처리는 이러한 명령어를 실행하여 어떤 기능을 수행하게 하는 기계 실행 가능 명령어와 같은 프로그램 코드로 수행될 수 있다. 이러한 의미에서, "기계"는 중간 형태[또는 "앱스트랙트(abstract)"] 명령어를 프로세서 특정 명령어[예를 들어, "가상 기계"(예: Java Virtual Machine), 인터프리터(interpreter), 커먼 랭귀지 런타임(Common Language Runtime), 고레벨 랭귀지 가상 기계와 같은 앱스트랙트 실행 환경]으로 변환하는 기계, 및/또는 범용 프로세서 및/또는 전용 프로세서와 같이 명령어를 실행하도록 설계된 반도체 칩[예를 들어, 트랜지스터로 구현되는 "로직 회로"] 상에 배치되어 있는 전자 회로일 수 있다. 전술한 기재가 교시하는 처리는 또한 프로그램 코드의 실행 없이 그 처리(또는 그 일부)를 수행하도록 설계된 전자 회로에 의하여(이와는 달리, 기계 또는 기계와의 결합으로) 행해질 수 있다.

[0072] 또한, 본 발명은 여기서 기재한 동작을 수행하는 장치에 관한 것이다. 이 장치는 필요한 목적으로 특별히 구성될 수 있으며, 또는 컴퓨터에 저장된 컴퓨터 프로그램에 의하여 선택적으로 활성화되거나 재구성되는 범용 컴퓨터를 포함할 수 있다. 이러한 컴퓨터 프로그램은 예를 들어, 플로피 디스크, 광 디스크, CD-ROM, 광자기 디스크, ROM, RAM, EPROM, EEPROM, 자기 또는 광학 카드, 또는 전자 명령어를 저장하기에 적당한 어떠한 유형의 매체를 포함하지만 이에 한정되지 않는 컴퓨터 판독 가능 매체에 저장될 수 있고, 이들 각각은 컴퓨터 시스템 버스에 연결된다.

[0073] 기계 판독 가능 매체는 기계(예: 컴퓨터)로 판독되는 형태로 정보를 저장하거나 전송하는 임의의 메커니즘을 포함한다. 예를 들어, ROM; RAM; 자기 디스크 저장 매체; 광 저장 매체; 플래시 메모리 장치; 전기, 광학, 음향 또는 다른 형태의 전파 신호 등(예를 들어, 반송파, 적외선 신호, 디지털 신호 등)을 포함한다.

[0074] 제품(article of manufacture)은 프로그램 코드를 저장하는데 사용될 수 있다. 프로그램 코드를 저장하는 제품은 예를 들어, 하나 이상의 메모리[예를 들어, 하나 이상의 플래시 메모리, 랜덤 액세스 메모리(정적, 동적, 기타)], 광 디스크, CD-ROM, DVD ROM, EPROM, EEPROM, 자기 또는 광학 카드, 또는 전자 명령어를 저장하기에 적당한 어떠한 유형의 매체를 포함하지만 이에 한정되지 않는 것으로 실시될 수 있다. 또한, 프로그램 코드는 전파 매체에서 실시되는 데이터 신호를 통하여[예를 들어, 통신 링크(예: 네트워크 연결)를 통하여] 원격 컴퓨터(예: 서버)에서 요청 컴퓨터(예: 클라이언트)로 다운로드될 수 있다.

[0075] 전술한 상세한 설명은 컴퓨터 메모리 내의 데이터 비트 상의 동작의 알고리즘 및 상징적인 표현의 관점에서 제공되었다. 이러한 알고리즘적인 설명과 표현은 데이터 처리 기술의 당업자가 다른 당업자에게 그 동작의 내용을 가장 효과적으로 전달할 때 사용되는 도구이다. 알고리즘은 여기서 그리고 일반적으로 원하는 결과로 이끄는 동작의 모순이 없는 시퀀스로 여겨진다. 동작은 물리적인 양의 물리적인 조작을 요하는 것이다. 보통, 반드시 필요한 것은 아니지만 이러한 양은 저장, 전송, 결합, 비교 및 이와는 달리 조작 가능한 전기 또는 자기 신호의 형태를 취한다. 이러한 신호는 비트, 값, 요소, 심볼, 문자, 용어, 숫자 등으로 지칭하는 것이 주로 공통적인 사용의 이유로 가끔 편리하다는 것이 입증되었다.

[0076] 하지만, 이러한 그리고 유사한 모든 용어는 적절한 물리량과 관련되어 있고 이러한 양에 적용되는 단지 편리한 라벨임을 명심하여야 한다. 본 기재로부터 명백한 대로 달리 특별히 언급하지 않는 한, 명세서 전체를 통하여 "처리," "컴퓨팅," "계산," "판단," 또는 "표시" 등과 같은 용어를 이용하는 기재는, 컴퓨터 시스템의 레지스터 및 메모리 내의 물리(전자)량으로 표현되는 데이터를 조작하고 컴퓨터 시스템 메모리나 레지스터 또는 다른 정보 저장 장치, 전송 또는 표시 장치 내의 물리량으로 유사하게 표현되는 다른 데이터로 변환시키는 컴퓨터 시스템이나 유사 전자 컴퓨팅 장치의 조치 및 처리를 말한다.

[0077] 여기서 제공하는 처리 및 표시는 내재적으로 어떠한 특정 컴퓨터 또는 다른 장치에 관련된 것이 아니다. 다양한 범용 시스템을 여기서의 교시에 따른 프로그램과 함께 사용할 수 있거나, 좀 더 특별한 장치를 구성하여 전술한 동작을 수행하는 것이 편리하다는 것이 입증될 수 있다. 이러한 다양한 시스템에 대하여 필요로 하는 구조는 이하의 설명으로부터 명백할 것이다. 또한, 본 발명은 어느 특정한 프로그래밍 언어를 참고로 하여 기술되지 않는다. 다양한 프로그래밍 언어를 이용하여 여기서 기재한 바대로의 본 발명의 교시를 구현할 수 있음을 인식할 것이다.

[0078] 전술한 기재는 본 발명의 일부 예시적인 실시예를 설명한 것뿐이다. 당업자는 이러한 기재, 첨부 도면 및 청구범위로부터 본 발명의 사상과 범위에서 벗어남이 없이 다양한 변형예가 이루어질 수 있음을 쉽게 인식할

것이다.

도면의 간단한 설명

[0011] 본 발명은 예를 통하여 설명되며 유사한 도면 부호는 유사한 요소를 나타내는 첨부 도면의 형상으로 제한되지 않는다.

[0012] 도 1은 보안 부팅을 위한 시스템 구성 요소의 한 실시예를 나타내는 블록도이다.

[0013] 도 2는 보안 부팅을 실행하는 시스템 구성 요소의 한 실시예를 나타내는 블록도이다.

[0014] 도 3은 보안 부팅을 행하는 처리의 한 실시예를 나타내는 흐름도이다.

[0015] 도 4는 UID(Unique Identifier) 및 시드열(seed string)에 기초하여 코드 이미지로부터 서명(signature)을 생성하는 처리의 한 실시예를 나타내는 흐름도이다.

[0016] 도 5는 호스트가 보안적으로 장치를 부팅하는 네트워크 연결의 한 실시예를 나타내는 블록도이다.

[0017] 도 6은 호스트에서 장치로 동작 환경을 보안적으로 복원하는 처리의 한 실시예를 나타내는 흐름도이다.

[0018] 도 7은 호스트에서 장치로 동작 환경의 최소 보안 복원을 행하는 처리의 한 실시예를 나타내는 상태도이다.

[0019] 도 8은 호스트에서 장치로 소프트웨어 구성 요소를 보안적으로 복구하는 처리의 한 실시예를 나타내는 흐름도이다.

[0020] 도 9는 호스트에서 장치로 애플리케이션을 보안적으로 업데이트하는 처리의 한 실시예를 나타내는 흐름도이다.

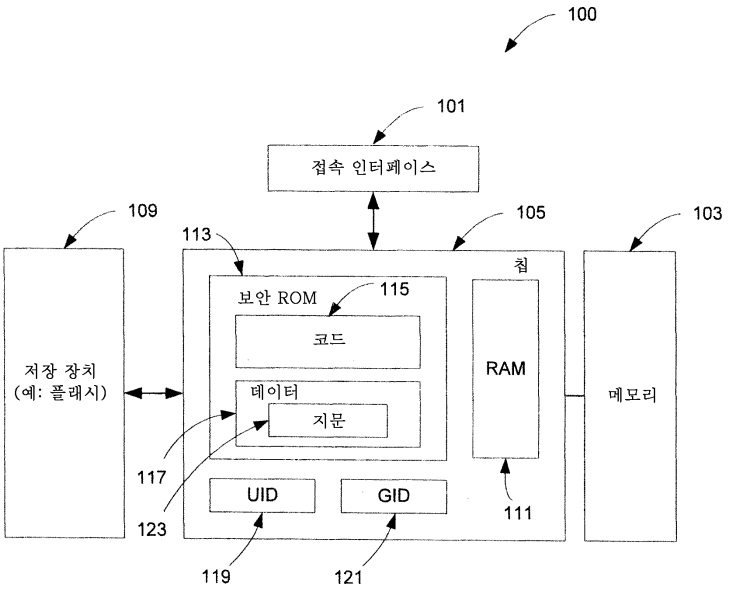
[0021] 도 10은 미인증 코드 이미지를 실행하는 처리의 한 실시예를 나타내는 흐름도이다.

[0022] 도 11은 여기서 기재하는 실시예와 함께 사용될 수 있는 통상의 컴퓨터 시스템의 한 예를 나타낸다.

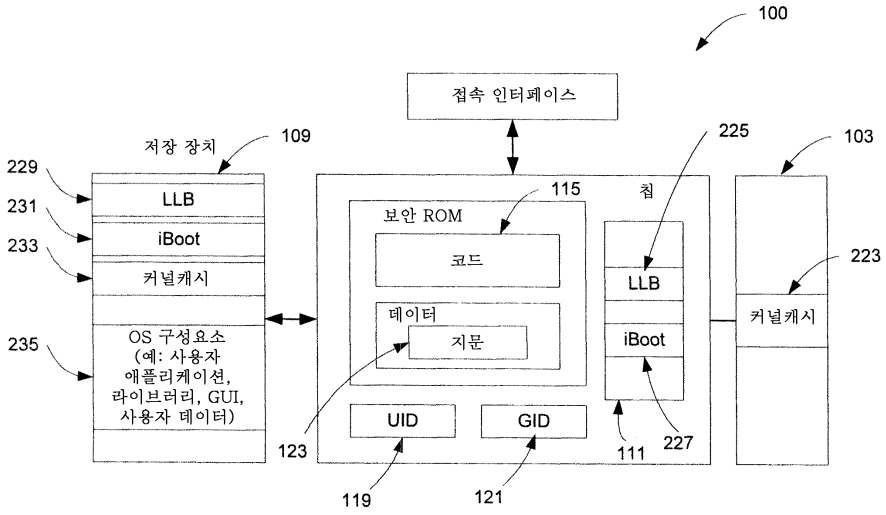
[0023] 도 12는 본 발명의 한 실시예와 함께 사용될 수 있는 데이터 처리 시스템의 한 예를 나타낸다.

도면

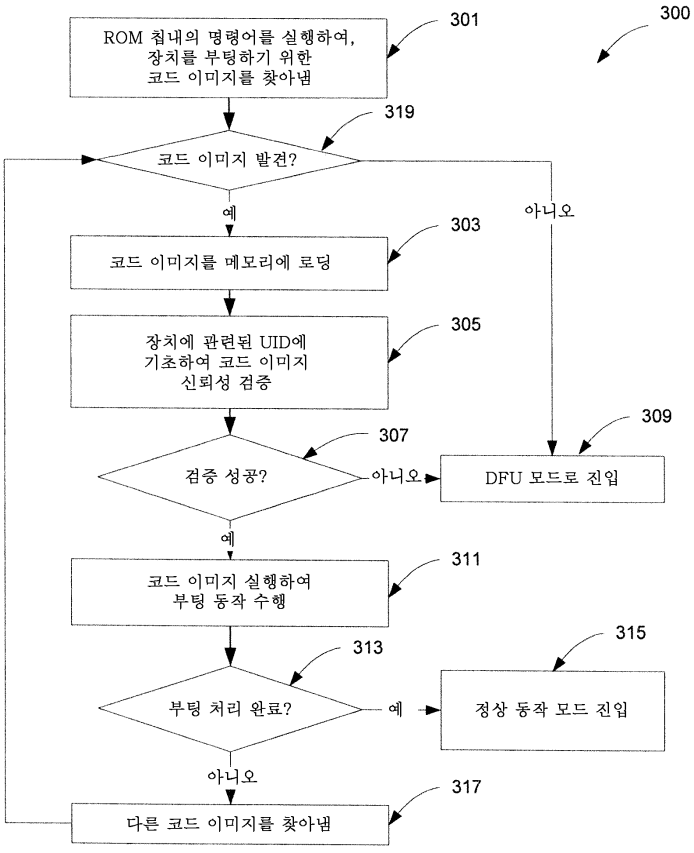
도면1



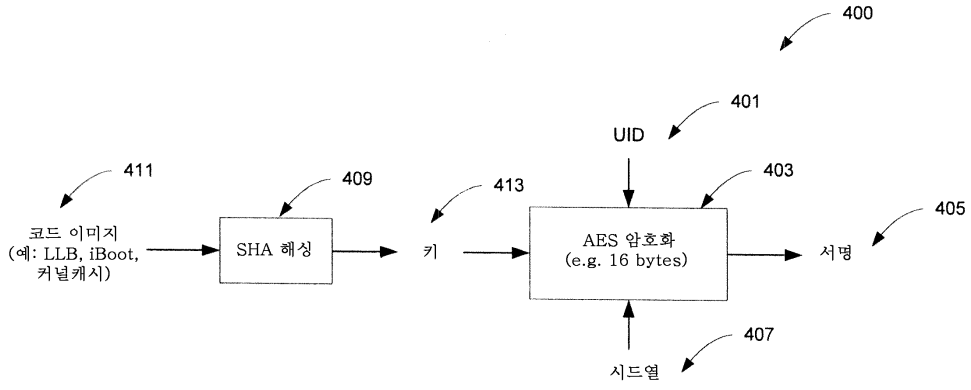
도면2


100
접속 인터페이스
저장 장치
109
229
231
233
235
LLB
iBoot
커널캐시
OS 구성요소
(예: 사용자
애플리케이션,
라이브러리, GUI,
사용자 데이터)
115
보안 ROM
코드
데이터
지문
UID
GID
123
119
121
칩
225
LLB
iBoot
111
227
103
223
커널캐시

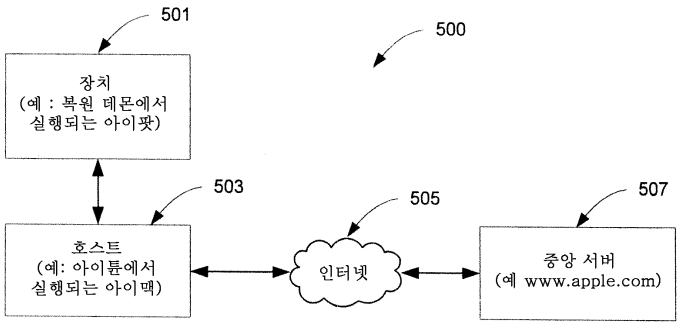
도면3


300
301
ROM 칩내의 명령어를 실행하여,
장치를 부팅하기 위한
코드 이미지를 찾아냄
319
코드 이미지 발견?
예
아니오
303
코드 이미지를 메모리에 로딩
305
장치에 관련된 UID에
기초하여 코드 이미지
신뢰성 검증
307
검증 성공?
아니오
309
DFU 모드로 진입
예
311
코드 이미지 실행하여
부팅 동작 수행
313
부팅 처리 완료?
예
315
정상 동작 모드 진입
아니오
317
다른 코드 이미지를 찾아냄

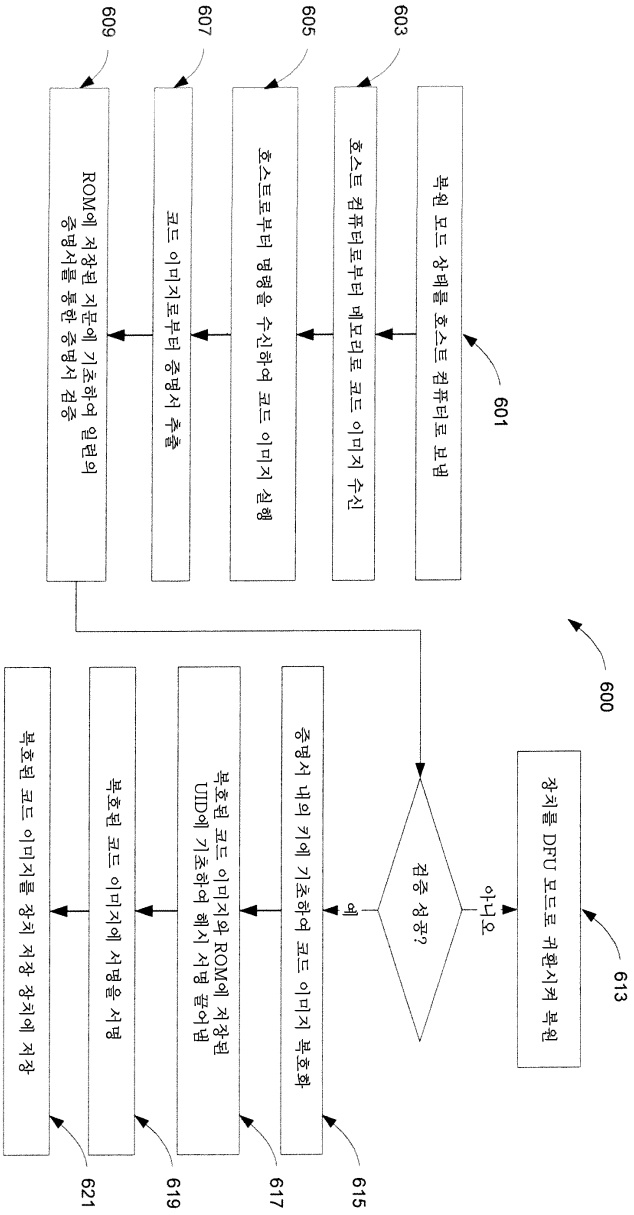
도면4


400
401
UID
411
409
413
403
405
코드 이미지
(예: LLB, iBoot,
커널캐시)
SHA 해싱
키
AES 암호화
(e.g. 16 bytes)
서명
407
시드열

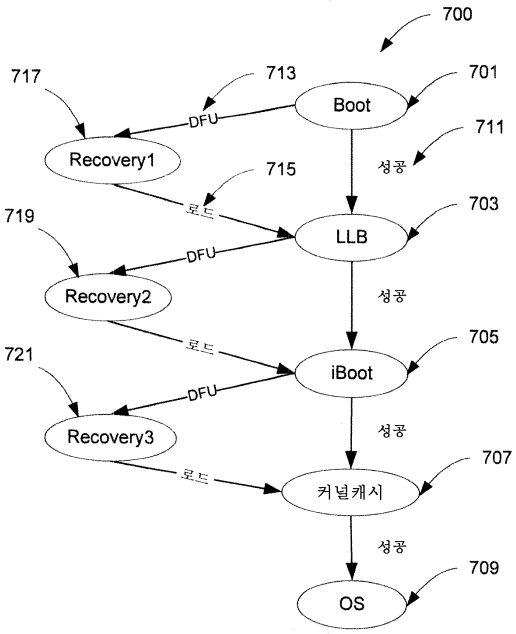
도면5


501
500
장치
(예 : 복원 데몬에서
실행되는 아이팟)
503
505
507
호스트
(예: 아이튠에서
실행되는 아이맥)
인터넷
중앙 서버
(예 www.apple.com)

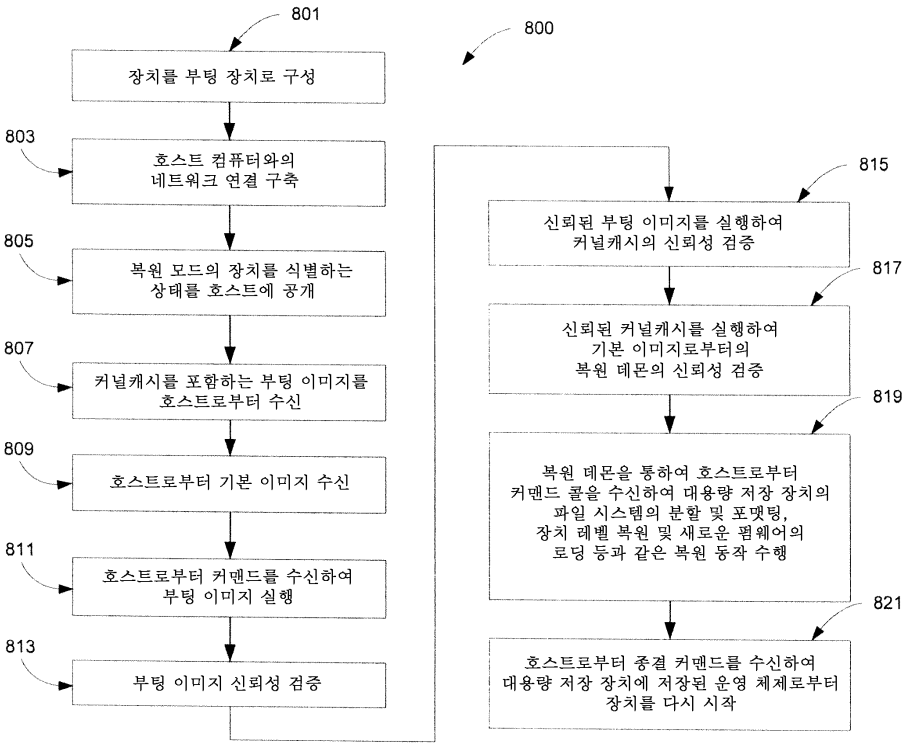
도면6


601
복원 모드 상태를 호스트 컴퓨터로 보냄
603
호스트 컴퓨터로부터 메모리로 코드 이미지 수신
605
호스트로부터 명령을 수신하여 코드 이미지 실행
607
코드 이미지로부터 증명서 추출
609
ROM에 저장된 지문에 기초하여 일련의 증명서를 통한 증명서 검증
600
검증 성공?
아니오
613
장치를 DFU 모드로 귀환시켜 복원
예
615
증명서 내의 키에 기초하여 코드 이미지 복호화
617
복호된 코드 이미지와 ROM에 저장된 UID에 기초하여 해시 서명 끝어냄
619
복호된 코드 이미지에 서명을 서명
621
복호된 코드 이미지를 장치 저장 장치에 저장

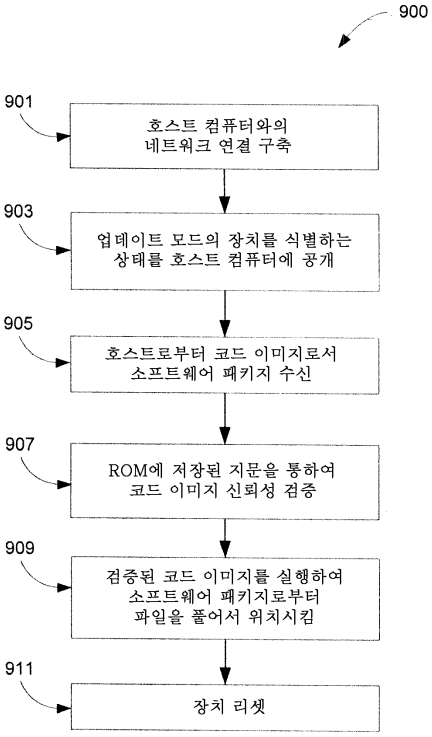
도면7


700
717
713
701
Boot
DFU
711
Recovery1
성공
715
로드
703
719
LLB
DFU
Recovery2
성공
로드
705
721
iBoot
DFU
성공
Recovery3
707
로드
커널캐시
성공
709
OS

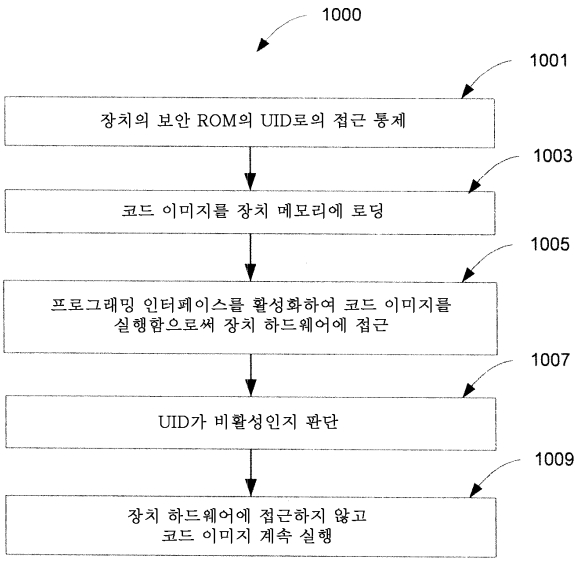
도면8


801
800
장치를 부팅 장치로 구성
803
호스트 컴퓨터와의
네트워크 연결 구축
805
복원 모드의 장치를 식별하는
상태를 호스트에 공개
807
커널캐시를 포함하는 부팅 이미지를
호스트로부터 수신
809
호스트로부터 기본 이미지 수신
811
호스트로부터 커맨드를 수신하여
부팅 이미지 실행
813
부팅 이미지 신뢰성 검증
815
신뢰된 부팅 이미지를 실행하여
커널캐시의 신뢰성 검증
817
신뢰된 커널캐시를 실행하여
기본 이미지로부터의
복원 데몬의 신뢰성 검증
819
복원 데몬을 통하여 호스트로부터
커맨드 콜을 수신하여 대용량 저장 장치의
파일 시스템의 분할 및 포맷팅,
장치 레벨 복원 및 새로운 펌웨어의
로딩 등과 같은 복원 동작 수행
821
호스트로부터 종결 커맨드를 수신하여
대용량 저장 장치에 저장된 운영 체제로부터
장치를 다시 시작

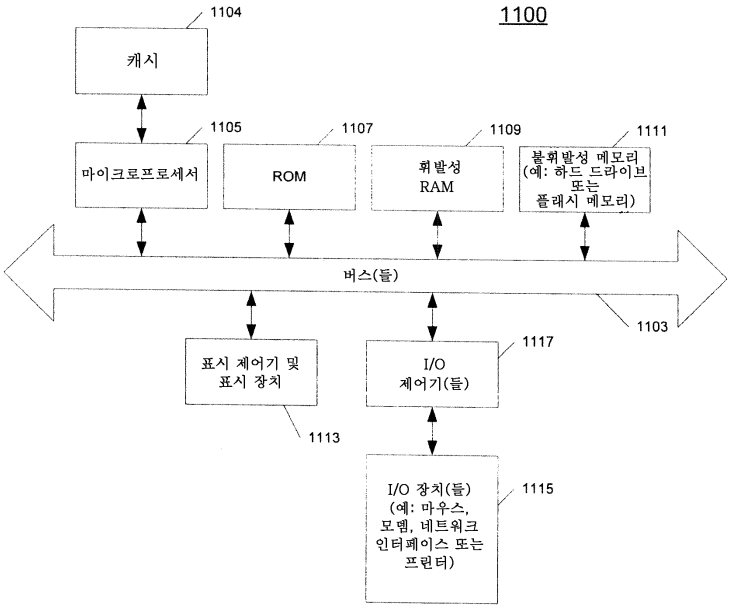
도면9


900
901
호스트 컴퓨터와의
네트워크 연결 구축
903
업데이트 모드의 장치를 식별하는
상태를 호스트 컴퓨터에 공개
905
호스트로부터 코드 이미지로서
소프트웨어 패키지 수신
907
ROM에 저장된 지문을 통하여
코드 이미지 신뢰성 검증
909
검증된 코드 이미지를 실행하여
소프트웨어 패키지로부터
파일을 풀어서 위치시킴
911
장치 리셋

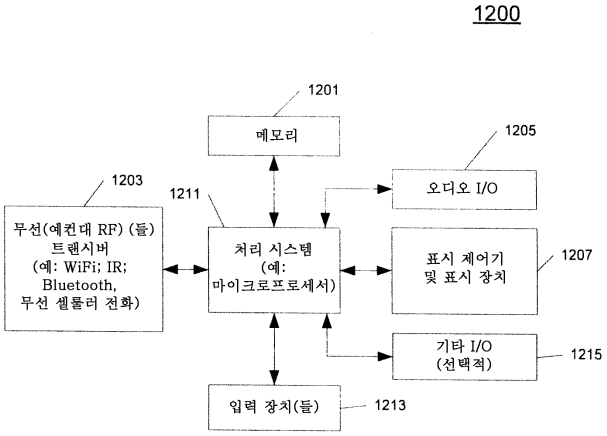
도면10


1000
1001
장치의 보안 ROM의 UID로의 접근 통제
1003
코드 이미지를 장치 메모리에 로딩
1005
프로그래밍 인터페이스를 활성화하여 코드 이미지를
실행함으로써 장치 하드웨어에 접근
1007
UID가 비활성인지 판단
1009
장치 하드웨어에 접근하지 않고
코드 이미지 계속 실행

도면11


1100
1104
캐시
1105
마이크로프로세서
1107
ROM
1109
휘발성 RAM
1111
불휘발성 메모리 (예: 하드 드라이브 또는 플래시 메모리)
버스(들)
1103
표시 제어기 및 표시 장치
1113
I/O 제어기(들)
1117
I/O 장치(들) (예: 마우스, 모뎀, 네트워크 인터페이스 또는 프린터)
1115

도면12


1200
1201
메모리
1205
오디오 I/O
1203
무선(예컨대 RF) (들) 트랜시버 (예: WiFi; IR; Bluetooth, 무선 셀룰러 전화)
1211
처리 시스템 (예: 마이크로프로세서)
표시 제어기 및 표시 장치
1207
기타 I/O (선택적)
1215
입력 장치(들)
1213