(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년08월13일
(11) 등록번호 10-0851631
(24) 등록일자 2008년08월05일

(51) Int. Cl.
G06F 21/00 (2006.01) *G06F 12/14* (2006.01)
G06F 1/00 (2006.01) *G06F 15/00* (2006.01)
(21) 출원번호 10-2006-7019183
(22) 출원일자 2006년09월18일
심사청구일자 2006년09월18일
번역문제출일자 2006년09월18일
(65) 공개번호 10-2006-0127206
(43) 공개일자 2006년12월11일
(86) 국제출원번호 PCT/IB2005/000562
국제출원일자 2005년03월03일
(87) 국제공개번호 WO 2005/091108
국제공개일자 2005년09월29일
(30) 우선권주장
10/804,855 2004년03월19일 미국(US)
(56) 선행기술조사문헌
US6385727B*
US2002/99946A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
노키아 코포레이션
핀랜드핀-02150 에스푸 카일알라덴티에 4
(72) 발명자
파테로 라우리
핀랜드 핀-00970 헬싱키 리칼란티에 4
(74) 대리인
리앤목특허법인

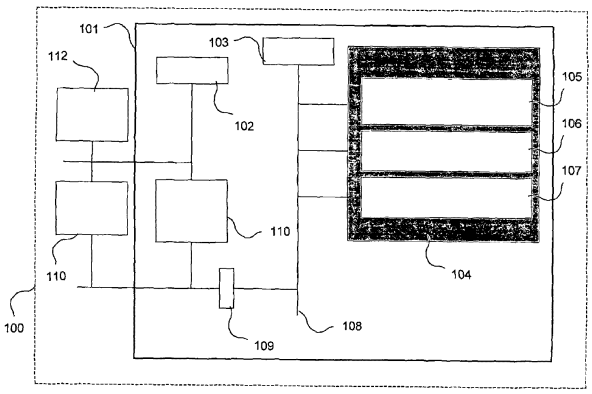
전체 청구항 수 : 총 32 항 심사관 : 노영철

(54) 보안 모드 제어 메모리

(57) 요 약

본 발명은 접근이 한정되는 보안 실행 환경(104)을 포함하는 전자 장치(101) 내에서 실행될 데이터의 보안을 향상시키기 위한 방법 및 시스템에 관련된다. 본 발명의 기본적 사상은, 장치의 부팅 시에 예를 들면 프로그램 코드의 형태를 가지는 데이터가 영구 메모리(112)로부터 임시 메모리(110)로 복제되는 것이다. 이러한 프로그램 코드의 무결성은 검증됨으로써, 프로그램 코드가 메모리들 사이에서 전송되는 동안에 변경되지 않았다는 점을 보증한다. 더 나아가, 보안 실행 환경에서 신규한 비밀 키가 생성된다. 이러한 신규 비밀 키는 장치 프로세서(103)에 의하여 사용됨으로써, 해당 프로그램 코드가 전송이 이루어지는 동안에 비밀 상태로 유지되었다는 점을 보장하기 위하여 임시 메모리 내에 저장될 프로그램 코드를 암호화한다. 그러면, 장치 프로세서는 암호화된 프로그램 코드를 임시 메모리에 기록한다.

대표도



특허청구의 범위

청구항 1

데이터 보안을 향상시키기 위한 방법으로서, 상기 데이터는 외부 소프트웨어 또는 외부 변경에 의한 접근이, 보안 제어 레지스터 수단 및 보안 실행 환경의 변경 방지 패키징에 의해 제한되는 상기 보안 실행 환경(secure execution environment, 104)을 포함하는 전자 장치(101)에서 실행되며, 상기 방법은,

신규 비밀 키를 상기 보안 실행 환경에서 반복적으로 생성하는 단계(S303)를 포함하되 상기 신규 비밀키를 생성하는 단계는 상기 장치가 부팅될 때 신규 비밀키를 생성하는 것을 포함하고;

저장소(110)에 기록될 데이터의 무결성을 상기 보안 실행 환경에서 검증하는 단계(S302);

상기 신규 비밀 키를 이용하여 상기 데이터를 상기 보안 실행 환경에서 암호화하는 단계(S304) 및

암호화된 상기 데이터를 저장소에 기록하는 단계(S305)를 포함하는 것을 특징으로 하는 방법.

청구항 2

삭제

청구항 3

제1항에 있어서,

신규 비밀 키는 런타임동안 반복적으로 생성되는 것을 특징으로 하는 방법.

청구항 4

제1항에 있어서,

상기 데이터는 프로그램 코드를 포함하는 것을 특징으로 하는 방법.

청구항 5

제1항에 있어서,

상기 저장소(110)는 임시 메모리를 포함하는 것을 특징으로 하는 방법.

청구항 6

제1항에 있어서,

부팅 시에, 상기 저장소(110)의 주소 위치를 주소 공간 내에 재정렬하는 단계로서, 상기 주소 공간 내의 상기 주소 위치의 순서를 변경시키는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 7

제4항에 있어서,

상기 저장소(110)에 기록될 상기 프로그램 코드를 상기 보안 실행 환경(104)에서 인증하여, 상기 프로그램 코드가 신뢰성 있는 프로그램 코드 제공자로부터 유래된 것임을 보장하는 단계(S403)를 더 포함하는 것을 특징으로 하는 방법.

청구항 8

제1항에 있어서, 상기 데이터 암호화 단계(S304)는,

상기 저장소(110) 내의 암호화된 상기 데이터가 기록될 위치의 주소를 상기 신규 비밀 키와 결합시키는 단계 및

상기 주소 및 상기 신규 비밀 키의 조합을 이용하여 상기 데이터를 암호화하는 단계로서, 암호화된 상기 데이터는 상기 주소와 관련되는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 9

제1항에 있어서, 신규 비밀 키를 생성하는 상기 생성 단계(S303)는,

복수 개의 신규 비밀 키들을 생성하는 단계로서, 각 신규 비밀 키는 데이터의 개별 하부 집합을 암호화하는데 사용되는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 10

제1항에 있어서,

상기 보안 실행 환경(104)에서, 상기 저장소(110) 내에 저장될 데이터에 대한 무결성 데이터를 연산하는 단계(S505) 및

연산된 상기 무결성 데이터를 저장하는 단계(S506)를 더 포함하는 것을 특징으로 하는 방법.

청구항 11

제10항에 있어서,

상기 무결성 데이터는 메시지 인증 코드를 포함하는 것을 특징으로 하는 방법.

청구항 12

제11항에 있어서,

상기 메시지 인증 코드는 생성된 신규 비밀 키를 이용하여 연산되는 것을 특징으로 하는 방법.

청구항 13

제12항에 있어서,

상이한 신규 비밀 키들을 이용하여, 상기 데이터의 상이한 부분들에 대하여 상이한 메시지 인증 코드들이 연산되는 것을 특징으로 하는 방법.

청구항 14

제13항에 있어서,

상기 보안 실행 환경(104)에서, 독출된 데이터에 관련된 상기 메시지 인증 코드의 정확성을 검증하는 단계(S602) 및

상기 메시지 인증 코드가 부정확할 경우 장치 동작을 중지시키는 단계(S603)를 더 포함하는 것을 특징으로 하는 방법.

청구항 15

제1항에 있어서,

상기 전자 장치(101) 내에 구현된 프로세서(103)를 보안 프로세서 실행 모드 및 정상 프로세서 실행 모드를 포함하는 적어도 두 개의 상이한 동작 모드 중 하나로 설정하는 단계 및

보안 회로부(105, 106, 107)의 적어도 하나의 저장 영역에 장치 보안에 관련된 보호된 데이터를 저장하는 단계를 더 포함하며,

보안 프로세서 동작 모드가 설정되었을 경우, 상기 프로세서에게는 상기 보호된 데이터가 위치된 상기 저장 영역으로의 접근 권한이 주어지고,

일반 프로세서 동작 모드가 설정되었을 경우, 상기 프로세서의 상기 저장 영역으로의 접근이 거부되는 것을 특징으로 하는 방법.

청구항 16

제15항에 있어서,

상기 프로세서 모드들의 설정 동작은 보호된 어플리케이션들에 의하여 수행되는 것을 특징으로 하는 방법.

청구항 17

데이터 보안을 향상시키기 위한 시스템으로서, 상기 데이터는 외부 소프트웨어 또는 외부 변경에 의한 접근이, 보안 제어 레지스터 수단 및 보안 실행 환경의 변경 방지 패키징에 의해 제한되는 보안 실행 환경(104)을 포함하는 전자 장치(101)에서 실행되는 시스템에 있어서,

신규 비밀 키를 상기 보안 실행 환경에서 반복적으로 생성하도록 상기 보안 실행 환경 내에 구현된 수단(103);

저장소(110)에 기록될 데이터의 무결성을 상기 보안 실행 환경에서 검증하도록 구현된 수단(103);

상기 신규 비밀 키를 이용하여 상기 데이터를 상기 보안 실행 환경에서 암호화하도록 구현된 수단(103) 및

암호화된 프로그램 코드를 저장소에 기록하도록 구현된 수단(103)을 포함하되, 상기 시스템은 상기 장치가 부팅되는 때에 신규 비밀키가 생성되도록 구현되는 것을 특징으로 하는 시스템.

청구항 18

삭제

청구항 19

제17항에 있어서, 상기 시스템은,

신규 비밀 키가 런타임동안 반복적으로 생성되도록 구현되는 것을 특징으로 하는 시스템.

청구항 20

제17항에 있어서,

상기 데이터는 프로그램 코드를 포함하는 것을 특징으로 하는 시스템.

청구항 21

제17항에 있어서,

상기 저장소(110)는 임시 메모리를 포함하는 것을 특징으로 하는 시스템.

청구항 22

제17항에 있어서,

부팅 시에, 상기 저장소(110)의 주소 위치를 주소 공간 내에 재정렬하도록 구현된 수단(103)을 더 포함하며,

상기 주소 공간 내의 상기 주소 위치의 순서는 변경되는 것을 특징으로 하는 시스템.

청구항 23

제20항에 있어서,

상기 저장소(110)에 기록될 상기 프로그램 코드를 상기 보안 실행 환경(104)에서 인증하여, 상기 프로그램 코드가 신뢰성 있는 프로그램 코드 제공자로부터 유래된 것임을 보장하도록 구현되는 수단(103)을 더 포함하는 것을 특징으로 하는 시스템.

청구항 24

제17항에 있어서, 상기 데이터를 암호화하도록 구현된 상기 수단(103)은,

상기 저장소(110) 내의 암호화된 상기 데이터가 기록될 위치의 주소를 상기 신규 비밀 키와 결합시키고,

상기 주소 및 상기 신규 비밀 키의 조합을 이용하여 상기 데이터를 암호화하도록 더욱 구현되며,

암호화된 상기 데이터는 상기 주소와 관련되는 것을 특징으로 하는 시스템.

청구항 25

제17항에 있어서,

상기 보안 실행 환경(104)에서, 상기 저장소(110) 내에 저장될 데이터에 대한 무결성 데이터를 연산하도록 구현된 수단(103) 및

연산된 상기 무결성 데이터를 저장하도록 구현된 수단(110, 112)을 더 포함하는 것을 특징으로 하는 시스템.

청구항 26

제25항에 있어서,

상기 무결성 데이터는 메시지 인증 코드를 포함하는 것을 특징으로 하는 시스템.

청구항 27

제26항에 있어서, 상기 연산 수단(103)은,

생성된 신규 비밀 키를 이용하여 상기 메시지 인증 코드를 연산하도록 구현되는 것을 특징으로 하는 시스템.

청구항 28

제27항에 있어서,

상기 보안 실행 환경(104)에서, 독출된 데이터에 관련된 상기 메시지 인증 코드의 정확성을 검증하고,

상기 메시지 인증 코드가 부정확할 경우 장치 동작을 중지시키도록 구현된 수단(103)을 더 포함하는 것을 특징으로 하는 시스템.

청구항 29

제17항에 있어서, 상기 시스템은,

적어도 두 개의 상이한 동작 모드들 중 하나로 설정될 수 있도록 구현된 프로세서(103) 및

장치 보안에 관련된 보호된 데이터가 위치되는 적어도 하나의 저장 영역을 포함하여 구현된 보안 회로부(105, 106, 107)를 더 포함하며,

보안 프로세서 동작 모드가 설정되었을 경우, 상기 프로세서에게는 상기 보호된 데이터가 위치된 상기 저장 영역으로의 접근 권한이 주어지고,

일반 프로세서 동작 모드가 설정되었을 경우, 상기 프로세서의 상기 저장 영역으로의 접근이 거부되는 것을 특징으로 하는 시스템.

청구항 30

제29항에 있어서,

상기 프로세서(103) 모드들의 설정 동작은 보호된 어플리케이션들에 의하여 수행되는 것을 특징으로 하는 시스템.

청구항 31

제17항 또는 제19항 내지 제30항 중 어느 한 항에 따른 시스템을 포함하는 이동 통신 단말기(100, 200).

청구항 32

제17항 또는 제19항 내지 제30항 중 어느 한 항에 따른 시스템을 포함하는 프로그램가능한 논리 장치(101, 201).

청구항 33

제32항에 있어서,

상기 프로그램가능한 논리 장치는 주문형 반도체의 형태로 구현되는 것을 특징으로 하는 프로그램가능한 논리 장치(101, 201).

청구항 34

삭제

청구항 35

컴퓨터에 의하여 실행될 수 있는 성분들이 장치(101) 내에 포함된 처리 유닛(103, 203)에서 실행될 때, 상기 장치(101)로 하여금 제1항 또는 제3항 내지 제16항 중 어느 한 항에 기술된 단계들을 수행하도록 야기하기 위한 컴퓨터에 의하여 실행될 수 있는 상기 성분들을 저장하는 컴퓨터에 의하여 독출될 수 있는 매체.

명 세 서

기 술 분 야

<1> 본 발명은 프로그램 코드 보안을 향상시키기 위한 방법으로서, 접근이 한정되는 보안 실행 환경을 포함하는 전자 장치 내에서 실행될 수 있는 프로그램 코드의 보안 향상 방법, 접근이 한정되는 보안 실행 환경을 포함하는 전자 장치 내에서 실행될 수 있는 프로그램 코드의 보안 향상 시스템, 해당 시스템을 포함하는 이동 통신 단말기, 해당 시스템을 포함하는 프로그램가능 논리 장치, 장치로 하여금 방법 청구항 중 어느 하나에 설명된 단계들을 수행하도록 야기할 수 있는 컴퓨터에 의하여 실행될 수 있는 구성 성분들을 포함하는 컴퓨터 프로그램, 및 장치로 하여금 방법 청구항들 중 어느 하나에 설명된 단계들을 실행하도록 야기하기 위한 컴퓨터에 의하여 실행될 수 있는 구성 성분들을 저장하는 컴퓨터에 의하여 독출될 수 있는 매체에 관련된다.

배 경 기 술

<2> 여러 가지 전자 장치, 예컨대 이동식 전기 통신 단말기, 휴대용 컴퓨터 및 PDA와 같은 전자 장치는, 보안 관련 구성요소, 이를테면 어플리케이션 프로그램, 암호키, 암호키 데이터 자료, 중간 암호 계산 결과, 패스워드, 외부에서 다운로드된 데이터를 위한 인증 수단 등으로의 접근을 요구한다. 전형적으로, 이들의 이러한 구성 요소 및 처리 동작이 전자 장치 내에서 비밀로 유지되는 것이 필요하다. 이상적으로, 장치의 보안 관련 구성 성분들이 공지된다면, 장치는 아마도 변경될 수 있으므로, 이러한 데이터는 가능한 한 적은 수의 사람들에게만 공지되어야 한다. 만일 그 보안 관련 구성요소가 알려져 있으면, 이 타입의 구성요소에 대한 접근 가능성은, 장치를 동작하려는 악의가 있는 의도가 있는 공격자의 행위를 용이하게 할 수 있다.

<3> 그러므로, 전자 장치 내의 프로세서가 보안 관련 구성 요소에 접근할 수 있는 보안 실행 환경(secure execution environment)이 소개된다. 이러한 보안 실행 환경에 대한 접근 및 보안 실행 환경 내에서의 및 외부로의 처리는 주의 깊게 제한되어야 한다. 이러한 보안 환경을 포함하고 있는 종래 기술 하드웨어는 흔히 변경-저항 패키징 내에서 보호된다. 보안 관련 구성 요소들의 공개 및 이들의 처리를 야기할 수 있는, 이러한 타입의 하드웨어에 대한 측정 및 테스트가 시도 또는 수행되는 것이 가능하여서는 안된다.

<4> 전술된 장치 아키텍처는 보안 공격에 대해 완벽하게 보안할 수 있다. 예를 들어, 보안 실행 환경외부에 위치하는 소프트웨어 공격자에 대한 향상된 보안 방법을 제공하는 것은 매우 바람직하다. 장치의 운영체제가 부팅될 때, 일반 소프트웨어가 시동되도록 확증하는 것은 일반적으로 용이한데, 그 이유는 이러한 목적 등을 위하여 특별히 개발된 보호된 어플리케이션 소프트웨어들이 이용되고, 이러한 보호된 어플리케이션들의 실행은 엄격히 제어되기 때문이다. 그러나, 후속 실행 동안, 공격자는 여러 가지 방법을 사용하여 "일반" 소프트웨어 어플리케이션의 변경을 시도할 수 있고, 장치 내에서 실행되는 모든 소에 대한 수정 가능성을 배제되어야 한다.

<5> 데이터 및 프로세서 코드의 보호는 매우 바람직한데, 그 이유는 악의의 공격자가 해당 장치를 훔치는 등의 방법에 의하여 장치에 접근할 수 있는 경우에 장치 내의 민감한 데이터에 접근 시도할 수 있기 때문이다. 이것은 디지털 권한 관리(DRM, Digital Rights Management) 시스템이 장치 내에 구현되었을 경우에도 그러하다. 이러한 DRM 시스템은 사용자가 콘텐츠에 대하여 가지는 접근의 타입을 결정하는 저작권에 의하여 보호되는 콘텐츠 및 관련된 디지털 권한을 저장한다. 그러므로, DRM 시스템은 허가되지 않는 유저에 의해 콘텐츠가 접근되거나, 이러한 콘텐츠가 오용하게 되고 또는 부당하게 배포되는 것을 방지하기 위하여 사용된다. 이러한 콘텐츠 및 권한이 경제적인 가치를 가지므로, 사용자는 DRM 제어 기능을 바이패스하는 것에 의해 콘텐츠에 액세스하려고 시도할 수 있다. 명백하게, 공격자가 장치를 조작하고자 시도할 수 있는 다양한 상이한 시나리오들이 예견될 수

있다.

<6> 전형적인 공격은 "개조된 칩 공격(modified-chip attack)"이라고 불리는 공격이다. 개조된 칩("mod-chip")공격에서, 공격자는 타겟 장치 아키텍처에 작은 칩을 부착한다. 그러면, 개조된 칩 공격은 장치 구조 내의 신호 및/또는 데이터를 수정하여 해당 장치를 조작한다. 개조된 칩 공격의 복잡성은 넓은 범위에 걸치는데, 그것은 단일 마이크로 프로세서 및 관련 소프트웨어를 포함하고 있는 디자인으로부터 수 천 개의 논리 게이트를 포함하고 있는 필드 프로그래머블 게이트 어레이(FPGA)를 통합하고 있는 매우 복합한 디자인까지이다. 무제한의 시간 및 복잡한 하드웨어/소프트웨어가 제공되면, 숙련된 공격자는 거의 모든 시스템에 침입할 수 있으며, 이러한 과감한 강제 공격으로부터 이러한 시스템을 보안하는 것은 실무상 매우 어렵다. 그러나, 거의 모든 경우에, 시스템의 허용 가능한 보안 수준은 채택된 보안 수단들이 공격자로부터 시스템의 침입을 하지 못하도록 방지하는 수준인 것이며, 그 이유는 그 이상으로 요구되는 수정된 칩 공격에 대한 복잡성은 수정된 칩 공격에 대한 방어 설계를 거의 불가능할 정도로 고가로 만들기 때문이다. 그러므로, 고수준의 복잡한 하드웨어/소프트웨어를 이용한 과감한 강제 공격으로부터 시스템을 보안하는 것은 실질적으로 불가능함에도 불구하고, 시스템은 덜 복잡하고 저가의 수정된 칩 공격을 이용하는 공격으로부터는 보호될 수 있다.

<7> 종래 기술에서, 시스템 및 장치를 보안하기 위하여 채택될 전형적인 방법들의 예를 들면 시스템/장치 버스로의 접근을 방해하는 것이었다. 프로그램 코드는 실행되는 동안은 물론 코드가 시스템 메모리 내에 저장되어 있는 동안에도 보호되어야 한다. 그러나, 장치 보안에 관련된 문제점들은 여전히 존재한다.

발명의 상세한 설명

<8> 본 발명의 목적은 전술된 문제점들을 완화시키고, 보안 실행 환경 외부에서 실행되는 소프트웨어의 공격자들로부터 시스템을 보안하기 위한 향상된 보안 시스템을 제공하는 것이다.

<9> 이러한 목적은 청구항 1항에서와 같이, 접근이 한정되는 보안 실행 환경을 포함하는 전자 장치 내에서 실행될 데이터에 대한 향상된 데이터 보안 방법에 의하거나, 청구항 17항에서와 같이, 접근이 한정되는 보안 실행 환경을 포함하는 전자 장치 내에서 실행될 데이터에 대한 향상된 데이터 보안 시스템에 의하거나, 청구항 31항에 따르는 시스템을 포함하는 이동 통신 단말기에 의하거나, 청구항 32항에 따르는 시스템을 포함하는 프로그램가능한 논리 장치에 의하거나, 청구항 34항에서와 같이, 장치로 하여금 방법 청구항들 중 어느 하나에 기술된 단계를 실행하도록 야기하는 컴퓨터에 의하여 실행될 수 있는 구성 성분을 포함하는 컴퓨터 프로그램에 의하거나, 청구항 35항에서와 같이, 장치로 하여금 방법 청구항들 중 어느 하나에 기술된 단계를 실행하도록 야기하는 컴퓨터에 의하여 실행될 수 있는 구성 성분을 기록하는 컴퓨터에 의하여 독출될 수 있는 매체에 의하여 달성될 수 있다.

<10> 본 발명의 기본 사상은, 장치가 동작 개시될 때 부팅되는 장치에서, 데이터가 영구 메모리(예를 들어 NAND 플래시 메모리)로부터 임시 메모리(예를 들어 RAM)로 복제되고, 이러한 임시 메모리로부터 데이터가 후속하여 실행되도록 한다는 것이다. 전형적으로 데이터는 프로그램 코드를 포함한다. 이러한 프로그램 코드의 무결성(integrity)은 검증됨으로써, 프로그램 코드가 NAND로부터 RAM으로 전송되는 동안에 변경되지 않았다는 것을 보장하여야 한다. 이러한 타입의 보안에 관련된 중요 동작은 보안 실행 환경 내에서 장치 프로세서에 의하여 수행된다. NAND 메모리는 언제나 외부 메모리(즉, 장치 외부에 위치된 메모리)인 것으로 간주될 수 있는 반면에, RAM은 일반적으로 장치 외부에 존재하는 것으로 간주되지만, 장치 내부(즉, 내부 메모리)에 존재하는 적은 양의 RAM도 존재할 수 있다는 점에 주의한다.

<11> 더 나아가, 장치의 부팅 또는 런타임 동안에, 신규한 비밀 키가 보안 실행 환경에서 생성된다. 이러한 신규한 보안 키는, 프로세서의 보안 동작 모드가 설정되면, (외부) RAM 내에 저장될 데이터를 암호화하여 비밀성을 제공하기 위하여(즉, 해당 데이터가 전송 동작 동안에 비밀로 유지되었다는 것을 보장하기 위하여) 장치 프로세서에 의하여 사용된다. 그러면, 장치 프로세서는 암호화된 데이터를 RAM에 기록한다. 데이터/프로그램 코드가 내부 RAM에 저장될 경우에 비하여 외부 RAM에 저장될 경우에 있어서 데이터/프로그램 코드의 암호화가 더욱 중요하다는 점에 주의하여야 하는데, 그 이유는 내부 RAM 자체는 장치 내부에 존재한다는 사실 때문에 상대적으로 안전하다고 간주될 수 있기 때문이다. 하지만, 보안 수준이 매우 높게 요청된다면 내부 RAM에 저장되어야 하는 데이터/프로그램 코드의 암호화 동작 역시 수행되어야 한다.

<12> 본 발명은 장점을 가지는데, 그 이유는 공격자가 RAM 내에 저장된 코드를 수정하고자 원한다면, 공격자는 암호화된 프로그램 코드가 복호화될 수 있도록 하는 비밀 키를 획득하여야 하기 때문이다. 프로그램 코드가 보안되지 않는다면, 공격자는 해당 코드를 수정하여 장치를 조작할 수 있다. 또한, 공격자는 암호화된 코드도 수정할

수 있지만, 이러한 수정은 다소 임의적인 효과만을 가지며, 장치가 예견된 방법으로 조작될 수 있도록 하는 효과는 가지지 못한다. 비밀 키의 보안 환경이 저장되는 장치의 비밀 실행 환경에 대한 접근이 엄격하게 통제되기 때문에, 공격자는 키를 획득하기 위하여 프로그램 코드를 암호분석(cryptoanalyze)하여야 한다. 신규한 부팅 시퀀스가 채택된다면, 상기한 프로시저는 반복되어야 한다. 이것은, 장치가 부팅될 때, 신규한 비밀 키가 생성되고 해당 프로그램 코드를 암호화하기 위하여 사용된다는 효과를 가져오는데, 그 결과 암호화 키가 반복적으로 변경된다는 사실에 기인하여 암호 분석 동작이 더욱 난해해 진다.

<13> 본 발명의 큰 또 다른 이점은, 실행 시간 동안 사용된 다소 강력하지 않은 알고리즘이 보상될 수 있다는 점이다. 강력한 암호화/무결성 메커니즘이 영구 저장소 내에 사용될 수 있지만, 이들은 런타임 동안에는 일반적으로 이용되지 않는다. 영구 메모리에 대한 일회성 벌크 접근(bulk access)을 수행하고, 따라서 데이터 또는 프로그램 코드의 방대한 양을 독출할 때, 데이터 무결성 검사에 관련된 성능 열화는 적을 뿐이다. 하지만, 런타임 동안에 이용되는 알고리즘이 약한 것인데, 그 이유는 이들은 실행을 지연시킴으로써 허용될 수 없는 레이턴시(latency)를 생성하기 때문이다. 그 결과 런타임 알고리즘을 보상하기 위하여, 빈번한 키 변경이 채택된다.

<14> 신규한 비밀 키의 생성 동작은 매번 신규로 부팅할 때마다 채택될 수 있지만, 무작위로 선택된 부팅 시퀀스마다 신규한 키를 생성하거나, 일정하게 선택된 부팅 시퀀스에서만 신규 비밀 키를 생성하는 것으로도 충분할 수 있으며, 여기서 "일정하게(regularly)" 라는 용어는 신규 키가 반드시 신규 부팅 시퀀스마다 생성되어야 한다는 것을 의미하는 것은 아니다. 이것은 한 편으로는 보안 목적 그리고 다른 한편으로는 신규한 키를 생성하는데 요구되는 처리량 사이의 트레이드-오프(trade-off)이다. 생성된 키 각각은 사전에 생성된 키에 비교할 때 신규하여야 하며, 이상적으로는 생성된 신규 키는 모든 다른 사전 생성된 키들에 비할 때 무작위여야 한다. 부팅시에 신규 비밀 키를 생성하는 것과 대안적으로(또는 부수적으로), 키들은 런타임 동안에도 생성될 수 있다. 이러한 경우에, 신규 키는 가능한 공격자가 비밀 키를 깨는데 요구되는 양의 데이터를 수집하기 이전에(또는 충분한 횟수의 연산을 수행하기 이전에) 생성되어야 한다. 런타임 키 변경 동작은 완전한 이전 키가 점진적으로 신규 키로 갱신될 때까지 이전 키의 일부가 변경되는 방식으로도 구현될 수 있다.

<15> 본 발명의 일 실시예에 따르면, 프로그램 코드가 보안 실행 환경에서 인증됨으로써, 프로그램 코드가 신뢰할 수 있는 코드 제공자로부터 유래했다는 것을 장치에게 확증할 수 있다. 인증 동작이 수행되지 않으면, 수정 칩 공격자는 이미 수정된 프로그램 코드를 장치에 공급할 수 있으며, 다시 말하면 공격자가 프로그램 코드의 사전-부트 수정 동작을 수행할 수 있다.

<16> 본 발명의 또 다른 실시예에 따르면, 암호화된 데이터가 기록되는 메모리 위치의 주소는 새로운 비밀 키와 결합된다. 예를 들어, 주소 공간 내의 주소 위치의 수치는 새로운 비밀 키에 연결되고(또는 다른 적합한 일부 방법에서는 결합되고). 그리고 연결된 결과가 상기 암호화 데이터가 저장소에 기록되기 이전에 상기 데이터를 암호화하는데 이용된다. 그러므로, 암호화 데이터는 해당 암호화 데이터가 저장된 주소에 관련된다.

<17> 이러한 실시예는, 예를 들어 공격자가 자신의 평문(plaintext)을 정의하는 소위 선택된 평문 공격을 마운팅하고, 이것을 암호화 하드웨어에 공급하며, 결과로 얻어지는 암호문(ciphertext)을 분석할 경우에 특히 유용하다. 이러한 공격은 고정된 프로그램 코드가 공격자가 선택한 (수정된) 프로그램 코드와 비교될 때 마운팅될 수 있지만, 실무상 구현하기는 어렵다. 그러나, 이러한 공격은 프로그램 코드 및 데이터 모두가 암호화될 때는 마운팅될 수 있다. 프로그램 코드가 고정되었지만, 데는 연속적으로 변화되며, 공격자는 결과적으로 동일한 키로 암호화된 (동일 주소 위치에 있는) 수 개의 상이한 데이터 집합들에 대한 접근 권한을 받을 수 있고, 따라서 암호 분석 동작을 수행할 수 있다. 전술된 연결 방법을 메모리에 기록될 데이터 및/또는 프로그램 코드를 암호화하기 위하여 사용함으로써, 암호화 키(즉, 연결(concatenation))는 각 주소 위치에 따라 상이할 수 있다. 이러한 사실은 암호 분석 동작을 매우 복잡하고 불가능하게 만든다.

<18> 본 발명의 또 다른 실시예에 따르면 복수 개의 신규 비밀 키가 장치 부팅 동작에서 생성되며, 여기서 각 비밀 키가 프로그램 코드의 개별 하부 집합을 암호화하는데 사용된다. 그러므로, 프로그램 코드의 상이한 부분을 암호화하기 위하여 상이한 비밀 키가 사용될 수 있다. 결과적으로, RAM과 같은 임시 저장 영역의 상이한 부분은 상이한 키에 의하여 암호화된 프로그램 코드를 포함할 수 있으며, 그 결과 프로그램 코드의 암호 분석 동작을 더욱 어렵게 한다.

<19> 본 발명의 또다른 실시예에 따르면, 주소 위치들은 부팅 동작에서 주소 공간 내에 배치(permute)된다. 이러한 배치, 또는 재정렬 동작은 공격자로 하여금 주소 공간 내에 특정 주소가 위치된 곳을 지득하기 어렵게 한다. 예를 들어, 주소 공간 내의 위치 번호 1024에 위치된 주소는, 재부팅시에 주소 위치 번호 2048로 매핑될 수 있

다. 후속 리부팅 동작에서, 이 주소는 예를 들어 위치 번호 512로 매핑될 수 있다. 이것은 시스템에 대한 공격을 어렵게 한다.

<20> 본 발명의 다른 실시예에 따르면, 무결성 데이터 형태의 리던던시(redundancy)가 채택된다. 체크섬, 메시지 인증 코드(MAC), 또는 다른 타입의 메시지 무결성 코드들과 같은 무결성 데이터의 사용 방법은 공지되었다. 무결성 데이터를 저장하는 것은, 이것이 추가적인 메모리를 요구하기 때문에 고가이다. 그러므로, 보안에 중요한 코드(보호된 어플리케이션들과 같은)에 대해서만 이들을 사용하도록 한정하는 것이 바람직하다.

<21> 무결성 코드의 저장이 고가이기 때문에, 무결성 코드를 형성하는 비트들의 수는 적은 것이 좋다. 그러나, 비트의 수가 적을수록, 개별 변경 동작을 검출할 가능성도 적어진다. 외부 메모리가 32-비트의 워드의 그룹으로 액세스된다고 가정한다. 만일 개별 32-비트 워드에 1-비트 짜리 체크섬이 부가된다면, 해당하는 1-비트 체크섬의 정확한 값을 추측할 확률은 50%이다. 만일 2 비트의 체크섬이 각 32-비트 워드에 추가된다면, 2비트 체크섬의 정확한 값을 추측해낼 확률은 25%이다. 본 발명의 바람직한 일 실시예에서, 암호화 체크섬(MAC과 같은)이 채택된다. MAC은 무결성이 제공되어야 하는 데이터 및 무결성 체크 키의 함수로써 연산된다. 이러한 무결성 체크 키는 전술된 바와 같은 신규 비밀 키를 포함하거나, 신규한 비밀 키들을 포함할 수 있다. 본 발명에서, 무결성 체크 키는 매 부팅시마다 변경되기 때문에, 상대적으로 적은 수의 비트들이 MAC에서 사용될 수 있다. 즉, 개별 부팅시마다 시도 변경(trial tampering) 동작이 수행되어야 한다)즉, 무결성 체크 키가 변경될 때마다 수행되어야 한다).

<22> 만일 복수 개의 새로운 비밀 키가 장치의 부팅시에 생성된다면, 상이한 메모리 위치에 대하 상이한 키를 이용하여 무결성 데이터가 연산됨으로써, 개별 메모리 위치에 존재하는 콘텐트가 다른 메모리 위치에서의 콘텐트로 스워핑(swapped)되지 않도록 한다.

<23> 데이터 또는 프로그램 코드가 RAM으로 써질 때, 관련된 MAC는 무결성 보호 저장 영역에 기록되는데, 이것은 장치 내에 위치되는 것이 바람직하지만, MAC은 외부 메모리에도 저장될 수 있다. 외부 또는 내부 메모리를 사용하는 것은, 성능 및 보안성에 대한 타협으로 결정된다. 장치의 내부 메모리로서 주문형 반도체(ASIC, Application Specific Integrated Circuit), FPGA, CPLD(Complex Programmable Logic Device) 또는 다른 일부 타입의 프로그램 가능한 하드웨어 형태로 구현될 수 있는 내부 메모리는 고속이고 보안성이 좋지만 상대적으로 고가이다. 반면에, 외부 메모리의 비용이 저렴하기 때문에 더 방대한 무결성 코드들이 저장됨으로써 보안성을 향상시킬 수 있다. 실무상 내부 메모리 및 외부 메모리의 조합이 보안 코드들의 저장을 위하여 채택될 것이다.

<24> 프로그램 코드가 실행되어야 할 때, 장치는 프로그램 코드를 RAM으로부터 독출한다. 장치는 프로그램 코드가 변경되지 않았는지 여부, 즉, MAC의 정당성을 확인한다. 만일 부정확한 MAC이 발견되면, 장치는 동작을 멈춤으로써 수정되었을 수 있는 프로그램 코드가 실행되지 않도록 한다.

<25> 발명의 또 다른 실시예에 따르면 아직, 장치 프로세서는 적어도 두 개의 상이한 동작 모드 중의 1개로 세팅될 수 있다. 해당 장치에서, 저장 회로는 장치 보안에 관한 보호된 자료가 위치하는 적어도 하나의 저장 영역과 함께 구현된다. 보안 프로세서 동작 모드가 설정될 경우, 프로세서에게는 저장 영역으로의 접근 권한이 주어지고, 일반 프로세서 동작 모드가 설정될 경우에는, 프로세서에게는 해당 저장 영역으로의 접근 권한이 거부된다. 프로세서 및 프로세서가 실행하는 어플리케이션에게 해당 저장 영역으로의 접근 권한이 부여되거나 부여되지 않는다는 사실이 실제 동작 모드를 정의하는 것이다. 더 나아가, 프로세서는 보안 프로세서 동작 모드가 설정 될 때. 가속기의 보안 제2 논리 인터페이스에 접근할 수 있다.

<26> 저장 회로의 저장 영역으로의 접근 동작은 프로세서의 보안 동작 모드를 정의한다. 보안 실행 모드에서 동작하는 동안 프로세서가 접근할 수 있는 저장 영역들은 보안 실행 환경(secure execution environment)라고 불린다. 이전에 언급된 바와 같이, 이러한 저장 영역은 외부적으로 다운로드된 데이터 기타를 위해 보안 관련 구성요소, 이를테면 예컨대 응용 프로그램, 암호 키, 암호 키 데이터 자료, 중간 암호 계산 결과, 패스워드, 인증 수단을 포함한다. 보안 실행 모드에서, 프로세서는 암호 키가 제공된 가속기의 보안 인터페이스에 액세스할 수 있다. 그러므로, 프로세서는 가속기 내에 키를 추가하거나, 가속기 내의 키를 변경할 수 있다. 이러한 특징은 중요하고 매우 유용한데, 그 이유는 일반(비보안) 처리 모드에서 장치에 부가되는 보안 제한 사항이 가혹하기 때문이다.

<27> 이상적으로, 보안 실행 환경의 내부에서 보안 중요 동작(security critical operation)을 수행하기 전형적으로 소형인 어플리케이션들인 소위 보호된 어플리케이션들만이, 비밀 암호 키를 관리하도록 허용될 수 있다. 보호

된 어플리케이션은 믿을 만한 제공자에 의해 발행될 수 있는 어플리케이션이며(그 경우에는, 제공자들은 반드시 인증되어야만 한다), 하지만 이들은 제 3자가 신뢰성이 있는지 여부에 관계없이 제 3자에 의하여 발생될 수도 있다. 후자의 경우, 아무런 인증 동작이 발생하지 않는다. 보호된 어플리케이션이 믿을 만한 제공자에 의하여 발생되어야 하는지 여부는 특정 콘텍스트로부터 결정되어야 한다. 일반적으로, 장치의 보안에 위험을 초래할 수 있는 권한을 가지거나, 이러한 권한이 어플리케이션들은 반드시 인증되어야 한다.

<28> 보호된 어플리케이션은 보안 환경 밖에서 실행되는 일반(normal) 어플리케이션의 일부라고 간주될 수 있다. 보호된 어플리케이션은 또한 장치에서 표준 기능성을 실행하기 위해 사용되는 어플리케이션을 포함할 수 있다. 예를 들면, 보호된 어플리케이션은 장치를 부팅하고, 운영체제를 장치에 탑재하기 위하여 사용된다. 심지어 장치 사용자에게도(그녀가 비인증 제3자인 것으로 간주될 수 없다 하여도) 비밀 암호키로의 접근 권한이 제공되지 않는 것이 바람직하다. 아마도, DRM 시스템이 장치 내에 구현되고, 디지털 콘텐츠 및 해당 DRM 시스템에 의해 제공되는 관련된 디지털 권한들이 경제적인 가치를 가지므로, 사용자는 DRM 제어 기능을 바이패스하는 것에 의해 콘텐츠에 액세스하려고 노력할 수 있다. 물론, 사용자에게 키로의 접근 권한이 부여되지 않아야 하는 다른 이유들도 존재한다. 예를 들면, 일반적인 보안 측면이 고려되어야 한다.

<29> 일반 장치 동작 모드에서, 장치 프로세서는 보안 환경 내에 위치한 보안 관련 데이터로의 접근 권한을 가지지 않는다. 보안 데이터는 암호 키 및 알고리즘, 회로를 부팅하기 위한 소프트웨어, 암호 키 자료로서 사용되는 임의의 수, 및 어플리케이션 프로그램 등을 포함한다. 이러한 보안 데이터로의 접근 및 이러한 보안 데이터의 처리는 제한된다. 전형적으로 이동 통신 단말기 내에 포함된 장치를 테스트 및/또는 디버깅할 때, 보안 관련 데이터로의 접근은 허용되지 않는다. 이러한 이유 때문에, 프로세서는 일반 상태, 또는 "비보안(unsecure)" 동작 상태에 위치되며, 이 모드에서 장치는 더 이상 보안 환경 내에서 보호된 데이터로의 접근 권한을 가지지 못한다. 결과적으로, 일반 모드에서는 프로세서 및 프로세서가 실행하는 상응하는 어플리케이션에는 가속기의 암호 키들로의 접근 권한이 부여되지 않는다.

<30> 본 발명의 다른 특징 및 다른 장점들은 첨부된 청구의 범위 및 후술되는 상세한 설명으로부터 명백해 질 것이다. 당업자들은 본 발명의 상이한 특징들이 결합되어 후술되는 실시예 이외의 다른 실시예를 생성할 수도 있다는 점을 이해할 것이다.

실 시 예

<38> 데이터 보안을 제공하기 위한 장치 아키텍쳐는 도 1에 보시된다. 이러한 시스템은 본 출원인의 국제 특허출원 공개 번호 제 WO2004/015553호에서 더 상세하게 개시되며, 해당 명세서는 본 명세서에 참조되어 통합된다. 데이터 보안을 제공하기 위한 회로는 주문형 반도체(ASIC, Application Specific Integrated Circuit, 101)의 형태로 구현된다. 아키텍처의 처리부는 CPU(103) 및 디지털 신호 처리기(DSP, 102)를 포함한다. ASIC(101)은 전자 장치(100), 이를테면 이동식 전기 통신 단말기, 휴대용 컴퓨터, PDA 등에 포함되고, 장치(100)의 "두뇌"에 해당하는 것으로 간주된다.

<39> 보안 환경(104)은 ASIC(101)이 부팅되는 ROM(105)을 포함한다. 이 ROM(105)은 부트 어플리케이션 소프트웨어 및 운영 체제를 포함한다. 보안 환경(104)에 존재하는 몇 가지 어플리케이션은 다른 어플리케이션 프로그램에 대한 우선 순위를 가진다. ASIC(101)이 포함될 수 있는 이동 통신 단말기에서, 부트 소프트웨어가 존재해야만 하는데, 이 소프트웨어는 단말기의 주요 기능성(main functionality)을 포함한다. 이러한 소프트웨어 없이, 단말기를 일반 동작 모드로 부팅하는 것은 가능하지 않다. 이러한 부트 소프트웨어를 제어함으로써 각 단말기의 최초 활성화를 제어할 수도 있다는 것은 유용하다.

<40> 보안 환경(104)은 데이터 및 어플리케이션, 즉 보호 데이터의 저장하기 위한 RAM(106)도 포함한다. RAM(106)은 소위 보호된 어플리케이션을 저장하는데, 이러한 어플리케이션은 보안 환경(104)의 내부에서 보안 중요 동작을 수행하기 위한 소형 어플리케이션이며, RAM(106)은 그 뿐만 아니라 객체, 이를테면 암호 키, 중간 암호 계산 결과 및 패스워드도 저장한다. 일반적으로, 보호된 어플리케이션을 채택하는 방법은 "일반" 어플리케이션들로 하여금 특정 보호된 어플리케이션으로부터 서비스를 요청하도록 하는 것이다. 새로운 보호된 어플리케이션은 언제든지 보안 환경(104)으로 다운로드될 수 있지만, 이러한 어플리케이션들이 ROM 내에 사주할 경우에는 그렇지 않다. 보안 환경(104) 소프트웨어는 보호된 어플리케이션들의 다운로드 및 실행을 제어한다. 보호된 어플리케이션은 보안 환경(104) 내의 모든 자원에 액세스할 수 있고, 이들은 보안 서비스의 제공을 위해 일반 어플리케이션과 통신할 수도 있다.

<41> 보안 환경(104)에서, 퓨즈 메모리(107)는 제조 과정에서 생성 및 ASIC(101) 내로 프로그램되는 고유 난수

(random number)를 포함하도록 구현된다. 이러한 난수는 ASIC(101)의 식별 표지로서 사용되고, 암호 동작을 위한 키를 유도하기 위하여 채택될 수도 있다. 더 나아가, 보안 제어 레지스터의 형태를 가지는 저장 회로 접근 제어 수단이 보안 환경(104)에 구현된다. 보안 제어 레지스터의 목적은 CPU(103)에게 보안 환경(104)으로의 접근 권한을 제공하거나, CPU(103)가 보안 환경(104)에 액세스하지 못하도록 하는 것이며, 이것은 레지스터의 모드 설정에 따라 수행된다. CPU(103)를 위한 동작 모드는 응용 프로그램 소프트웨어에 의하여 레지스터 내에 설정될 수 있으며, 그 결과 아키텍처는 외부 신호에 의존할 필요가 없게 된다. 보안의 관점으로 볼 때 이것은 바람직한데, 그 이유는 어플리케이션 프로그램 소프트웨어를 제어하는 것에 의해 프로세서 모드의 설정도 제어할 수 있기 때문이다. ASIC(101)에 연결된 외부 신호(미도시)를 이용하여 보안 제어 레지스터를 설정하는 것도 가능하다. 외부 신호를 사용하는 것에 의해, 모드 변경은 용이하고 고속으로 실행될 수 있으며, 이것은 시험 환경에서 바람직하다. 어플리케이션 소프트웨어 및 외부 신호와 같은 이러한 두 가지 모드 설정 수단의 조합도 구현 가능하다.

<42> 아키텍처는 버스(108) 상의 데이터 가시성(data visibility)의 제한을 위한 표준 브릿지 회로(109)를 더 포함한다. 아키텍처는 변경 방지 패키징(tamper resistant packaging) 내에서 밀폐되어야만 한다. 이러한 타입의 하드웨어에서, 보안 관련 구성 성분들의 공개 및 이러한 성분들의 처리를 야기할 수 있는 측정 및 시험을 시도(probe) 또는 수행하는 것은 가능하지 않다. DSP(102)는 직접 메모리 액세스(DMA) 유닛, RAM, 플래시 메모리와 같은 다른 주변 장치(110)에 대한 접근 권한을 가지며, 부가적 프로세서는 ASIC(101) 외부에서 제공될 수 있다. 본 발명의 특정 실시예들에서는 주변 장치들(110, 112)이 임시 메모리(예를 들어, RAM) 및 영구 메모리(예를 들어, NAND 플래시 메모리)를 각각 표시하는 것으로 도시된다.

<43> 데이터 보안을 제공하기 위한 장치 아키텍처의 또 다른 실시예는 도 2에 도시되며, 여기서 도 1에 도시된 부재와 상응하는 부재에 대해서는 동일한 부재 번호가 사용된다. 도 2에 도시된 아키텍처를 도 1에 예시된 아키텍처와 비교하면, 그 차이점은 전자 장치(200)가 착탈식 스마트카드(211)(예를 들면 SIM)를 포함하여 구현된다는 점이며, 이러한 카드도 보안 환경인 것으로 간주될 수 있다. 보안 목적을 달성하기 위하여, 이동 단말기(200) 및 스마트카드(211)는 신뢰성 있는 인증 기관(CA, certification authorities)에 의하여 발행된 디지털 인증서를 저장한다. 인증서는 이동 단말기(200) 및/또는 스마트카드(211)와 통신하는 활동자(actor)에게, 특정 인증서의 소지자가 상응하는 신뢰성 있는 인증 기관(CA)에 의하여 인증받았다는 점을 보장하기 위하여 사용된다. 인증 기관(CA)은 인증서에 서명하며, 그리고 인증서 소지자는 인증 기관(CA)에 의하여 서명된 인증서가 유효하다는 것을 검증하기 위하여는, 해당 인증 기관(CA)의 기밀 키에 상응하는 공개키를 소지하여야 한다. 상이한 장치들이 상이한 인증 기관(CA)들로부터의 인증서를 가질 수 있음에 주의한다. 이러한 경우에, 상이한 인증 기관(CA)은 상호 몇 가지 통신을 수행하여야 하는데, 예를 들어 그들의 공개 키를 교환하여야 한다. 인증서는 당업자에게 공지된 바 있으며, 주지된 표준 인증서에는 CCITT 권고 X.509에 포함된 인증서들이 있다.

<44> 도 3은 ASIC(101)의 부트 동작에 수반되는 프로시저들의 흐름도를 도시하고, 이에 대한 상세한 설명은 다음과 같이 제공된다. 시스템 부트(도 3에는 미도시)가 이루어지는 동안에 OS 커널을 ASIC에 로드하기 위하여 보호된 어플리케이션이 이용된다. 커널을 로드할 때, 보호된 어플리케이션은 무결성을 확인하고, 커널을 형성하는 프로그램 코드를 암호화한다. 계속하여, 커널 코드가 실행되는 동안, 커널 프로그램 코드가 보안 실행 환경(104)에서 복호되고, 해당 커널 코드와 관련된 무결성 코드들이 정확한지에 대한 점검이 수행된다. 전형적으로, 프로그램 코드에 수행되는 암호화 동작 및 상응하는 무결성 코드의 연산 동작에 관련하여, 보호된 어플리케이션이 부팅 시에 사용된다. 일반 실행 동작이 수행되는 동안에, 보안 실행 환경의 하드웨어는 복호화 및 무결성 검증 동작을 담당한다. 도면에서, 본 발명의 교시 사항을 예시하기 위하여 프로그램 코드 형태의 데이터가 사용된 점에 대하여 주의한다. 당업자는 모든 적합한 형태의 데이터가 사용될 수 있다는 점을 이해할 것이며, 따라서 "데이터"는 "프로그램 코드"를 의미하는 것으로 한정되어서는 안된다는 점을 이해할 것이다.

<45> OS가 ASIC(101)에 로드되고, 이에 따라 정확하게 동작하기 위하여 요구되는 기본 기능들이 ASIC에 제공되면, 프로그램 코드가 영구 메모리 112부터 독출되고("S301"이라고 표시되는 단계 301), 독출된 프로그램 코드는 후속하여 임시 메모리(110)로 기록되며, 이러한 임시 메모리로부터 프로그램 코드가 실행될 것이다. 이러한 프로그램 코드의 무결성은 검증됨으로써(S302) 해당 프로그램 코드가 NAND(112)로부터 RAM(110)으로 전송되는 동안에 변경되지 않았다는 것을 보장하여야 한다. 이러한 타입의 보안에 관련된 중요 동작은 보안 실행 환경(104) 내에서 CPU(103)에 의하여 수행된다. 장치가 부팅될 때, CPU는 보안 실행 환경(104)에서 신규한 비밀 키를 생성하고(S303), 생성된 비밀 키를 이용하여 검증된 프로그램 코드를 암호화(S304)함으로써 암호화 프로그램 코드를 RAM(110)으로 전송하는 동안에 프로그램 코드가 비밀인 상태로 유지되었다는 점을 보장한다. 높은 보안 수준을 요청하는 이러한 처리 동작에 대해서, 프로세서(103)의 보안 동작 모드가 설정된다.

<46> ASIC(101) 내에 높은 수준의 보안 수준을 제공하기 위하여, 부팅시에 복수 개의 신규 비밀 키 들이 생성되어야 한다는 점에 주의한다. 여기서 각 비밀 키가 프로그램 코드의 개별 하부 집합을 암호화하는데 사용될 수 있다. 그러므로, 프로그램 코드의 상이한 부분을 암호화하기 위하여 상이한 비밀 키가 사용될 수 있다. 후술되는 상세한 설명으로부터 명백하게 이해될 수 있는 바와 같이, 저장될 프로그램 코드에 무결성 코드들이 제공되어야 할 경우에, 복수 개의 비밀 키들을 생성하는 것은 특히 유용하다.

<47> 도 4는 본 발명의 또다른 실시예를 도시하는데, 도 4에 도시된 실시예에서 영구 저장소로부터 독출된 프로그램 코드는 더 인증되어(S404) 해당 프로그램 코드가 신뢰할 수 있는 프로그램 코드 제공자로부터 유래하였다는 것을 보장한다. 이것은 도 3과 관련하여 설명된 실시예와 비교할 때 추가적인 특징이다. 인증 동작이 수행된다는 점은, 프로그램 코드에 몇 가지 인증 수단이 제공되었다는 것을 의미하는데, 예를 들어서 신뢰성 있는 제공자 및 ASIC(101)에게 알려진 대칭 키에 의하여 암호화되거나, 디지털 서명에 의하여 서명되었다는 것을 의미한다.

<48> 장치의 보안 수준을 더 개선하기 위해, 무결성 코드가 ASIC(101)에서 사용될 수 있다. 이러한 목적을 위해 예컨대 체크섬 또는 메시지 확인 코드(MAC)를 사용하는 것이 당업계에 공지된 바 있다. 본 발명의 실시예는 프로그램 코드를 위한 무결성 코드의 연산을 다루며, 이에 대한 흐름도는 도 5에 제공된다. 도 5에 도시된 흐름도는 도 3 및 도 4에 도시된 흐름도들의 확장판으로 이해될 수 있다. 신규 비밀 키들이 생성되면(S504), RAM(110)에 저장될 프로그램 코드를 위해 MAC이 연산된다. MAC은 무결성이 제공되어야 하는 프로그램 코드 및 생성된 신규 비밀 키의 함수로써 연산된다. 일반적으로 사용되는 MAC 또는 암호화 체크섬은 데이터 암호화 표준(DES, Data Authentication Algorithm)에 기반한 데이터 인증 알고리즘이다.

<49> 실무상, 임시 메모리로 기록될 프로그램 코드의 각 시퀀스에 대해 하나의 MAC이 계산된다. 이것은, 만일 외부 메모리가 32-비트 워드의 그룹 형태로 접근된다면, 하나의 MAC이 프로그램 코드의 32-비트 워드 각각에 대해서 연산된다는 것을 의미한다. 따라서, 전술된 바와 같이 부팅시에 복수 개의 신규 비밀 키들이 생성되고, 개별 신규 키가 개별 MAC의 연산을 위하여 채택된다. 프로그램 코드가 RAM(110)으로 기록되면, 관련된 MAC은 장치 내에 위치된 무결성 보호 저장소(110, integrity protection storage)로 저장(S506)되는 것이 바람직하다.

<50> 도 6에서, 프로그램 코드가 실행될 때, CPU(103)(또는 DSP(102))가 프로그램 코드가 저장된 메모리(110)로부터 해당 프로그램 코드를 독출한다(S601). CPU는 MAC의 정당성을 확인하는데(S602), 다시 말하면 프로그램 코드가 수정되었는지가 점검된다. 만일 부정확한 MAC가 발견되면, CPU는 동작을 중단함으로써 수정되었을 수 있는 프로그램 코드가 ASIC(101)에 악영향을 끼치지 못하도록 방해한다.

<51> 앞서 논의된 바와 같이, 프로세서(103)는 두 가지 상이한 실행 모드에서 동작하는데, 그것은 일반 실행 모드 및 보안 실행 모드이며, 보안 모드가 일반 모드에 대해 우선권을 가진다. 이러한 두 가지 실행 모드들에 부가하여, 두 프로그램 모드들이 존재하는데, 이들은 "사용자 모드" 및 "관리자 모드"이며, 관리자 모드가 사용자 모드에 우선한다. 간략히 설명하면, 사용자 모드는 일반적으로 사용자 프로그램의 실행에 관련되고, 관리자 모드는 OS 커널 프로그램들의 실행에 관련된다. 전형적으로, 커널 프로그램은 인터럽트 핸들러, 어떤 프로그램들이 커널의 처리 시간을 공유하고, 어느 순서로 공유하는지를 결정하는 스케줄러, 및 개별 프로세스가 스케줄링되면 개별 프로세스에게 해당 프로세서에 대한 사용권을 제공하는 관리자(supervisor)를 포함한다. 이러한 커널 프로그램들은 그 명칭이 암시하는 바와 같이 장치 소프트웨어의 진정한 코어에 해당한다. 이러한 커널 프로그램은 매우 중요한 보안 사항이라고 간주될 수 있으며, 보안 실행 모드의 감시 프로그램 모드에서 작동된다. 부트 프로그램과 같은 보호된 어플리케이션도 매우 중요한 보안 사항이라고 간주될 수 있으나, 그들은 보안 실행 모드의 유저 모드에서 작동됨으로써, 이들은 커널 기능을 가질 수 없는데, 특히 이들이 관리자 모드에서 실행될 때 그러하다.

<52> 사용자 모드 및 관리자 모드는 본 발명의 기술적 사상의 중요한 본질이 아니기 때문에 더 이상 상세히 설명되지 않는다. 하지만, 암호화 동작은 처리 능력을 요구한다는 점 및 적절한 이유가 없이는 수행되어서는 안된다는 점이 이해되어야 한다. 더 나아가, 무결성 코드를 저장하려면 추가적인 메모리가 필요하기 때문에 무결성 코드들의 저장 동작은 고가이다. 그 결과로 장치 보안 및 장치 가격 사이의 트레이드-오프가 필요하다. 특정 타입의 프로그램 코드가 암호화 처리되어야 하고 다른 타입들은 그렇지 않은지 여부에 대한 선택을 하여야 한다면, 다음의 순서와 같은 우선권 순서가 존재한다.

<53> - 보안 실행 모드의 관리자 프로그램 모드 프로그램(가장 높은 우선 순위);

<54> - 보안 실행 모드의 사용자 모드 프로그램;

<55> - 일반 실행 모드의 관리자 프로그램 모드 프로그램;

<56> - 일반 실행 모드의 사용자 모드 프로그램(최하위 우선 순위).

<57> 당업자들은 본 발명에 포함되는 하드웨어는 전형적으로 도 3 내지 도 6에 관련하여 설명된 단계들을 수행하기 위한 적합한 소프트웨어를 실행한다는 점을 이해할 것이라는 점에 주의한다.

<58> 비록 본 발명이 특정한 예시적 실시예들을 참조하여 설명되었지만, 다른 다양한 변경, 수정 등이 가해질 수 있다는 점이 당업자에게는 명백할 것이다. 그러므로, 설명된 실시예들은 본 발명의 기술적 사상을 한정하는 것으로 이해되어서는 안되며, 본 발명의 기술적 사항은 청구의 범위에 의하여 정의된다.

산업상 이용 가능성

<59> 본 발명은 프로그램 코드 보안을 향상시키기 위한 방법에 사용될 수 있으며, 접근이 한정되는 보안 실행 환경을 포함하는 전자 장치 내에서 실행될 수 있는 프로그램 코드의 보안 향상 방법, 접근이 한정되는 보안 실행 환경을 포함하는 전자 장치 내에서 실행될 수 있는 프로그램 코드의 보안 향상 시스템에 적용가능하다.

도면의 간단한 설명

<31> 본 발명은 더욱 상세해 다음 도면들을 참조하여 더욱 상세히 설명될 것이다.

<32> 도 1은 본 발명이 바람직하게 적용될 수 있는 데이터 보안성을 제공하기 위한 장치 아키텍처의 개념적인 블록도이다.

<33> 도 2는 본 발명이 바람직하게 적용될 수 있는 데이터 보안성을 제공하기 위한 장치 아키텍처로서, 탈착식 스마트 카드를 이용하여 구현된 장치 아키텍처의 개념적인 블록도이다.

<34> 도 3은 본 발명의 일 실시예에 따르는 장치의 부팅시 수행되는 프로시저의 흐름도이다.

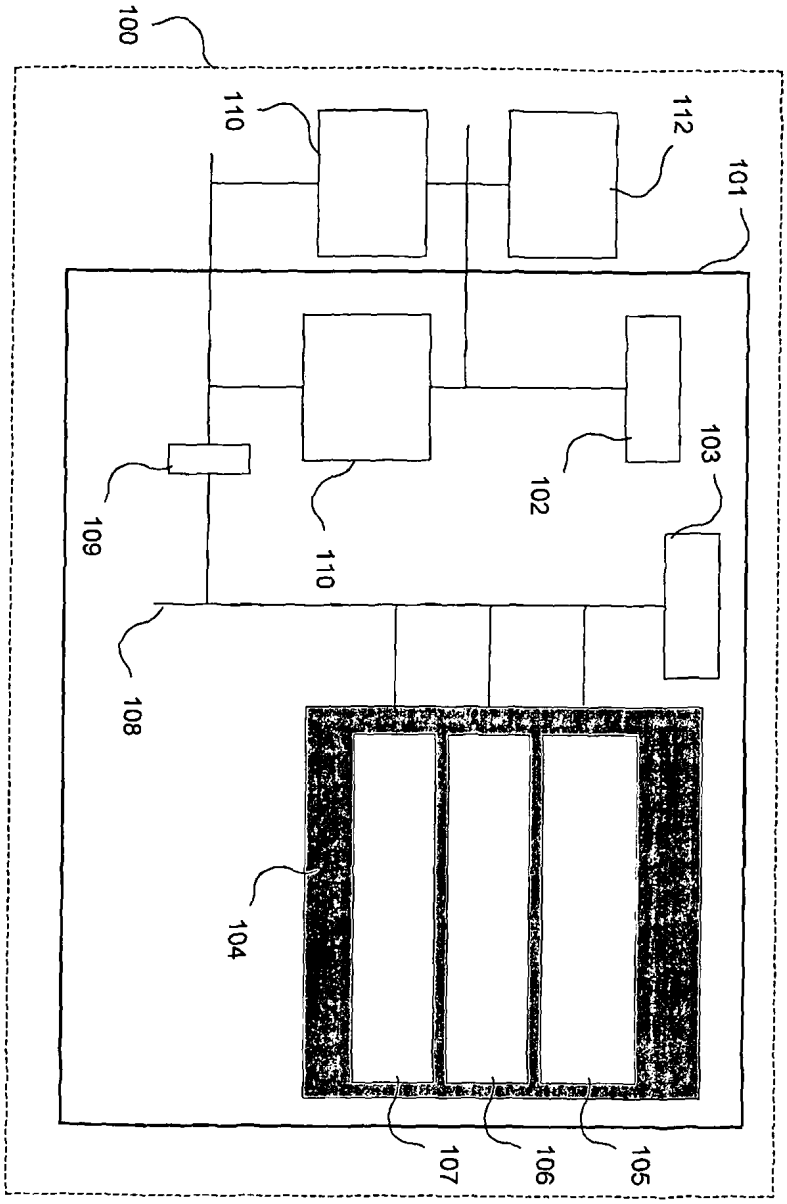
<35> 도 4는 본 발명의 다른 실시예에 따르는 장치의 부팅시 수행되는 프로시저의 흐름도이다.

<36> 도 5는 본 발명의 일 실시예에 따라, 프로그램 코드에 메시지 인증 코드를 제공하는 프로시저의 흐름도이다.

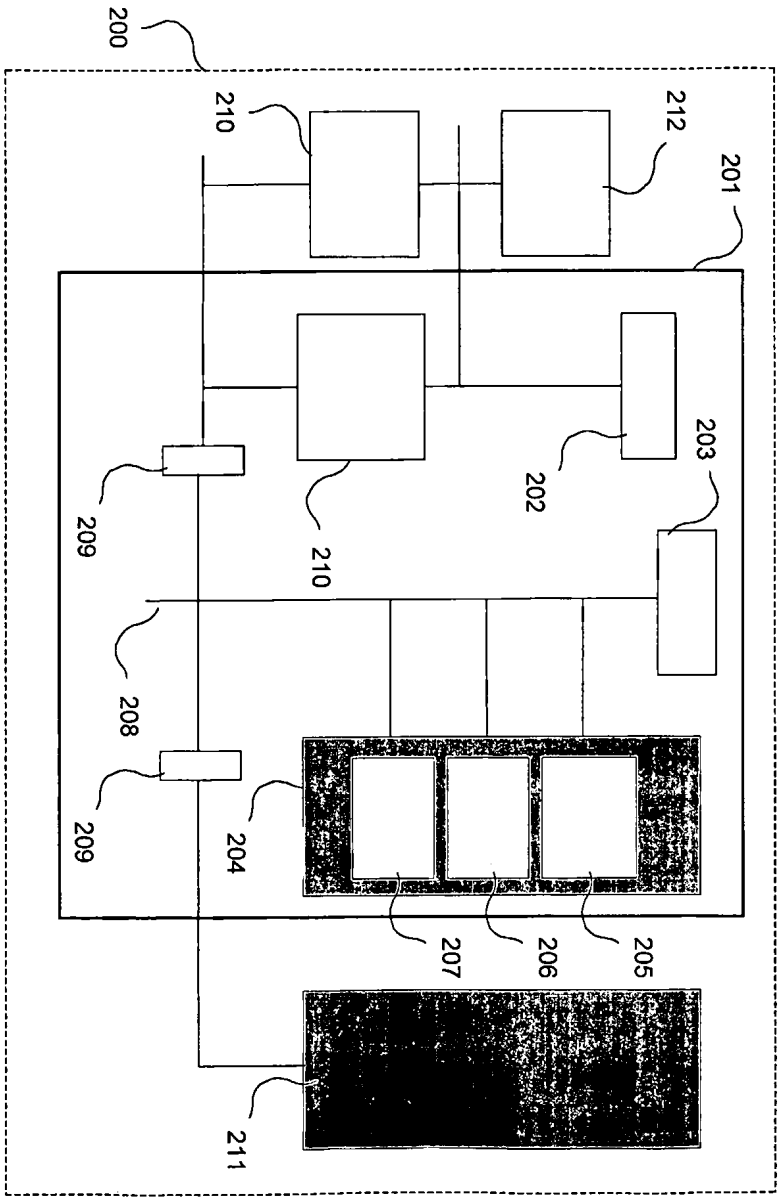
<37> 도 6은 본 발명의 다른 실시예에 따라, 메시지 인증 코드의 정확성을 검증하는 프로시저의 흐름도이다.

도면

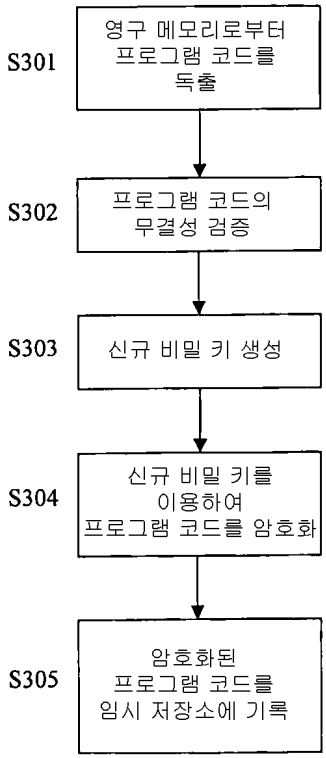
도면1



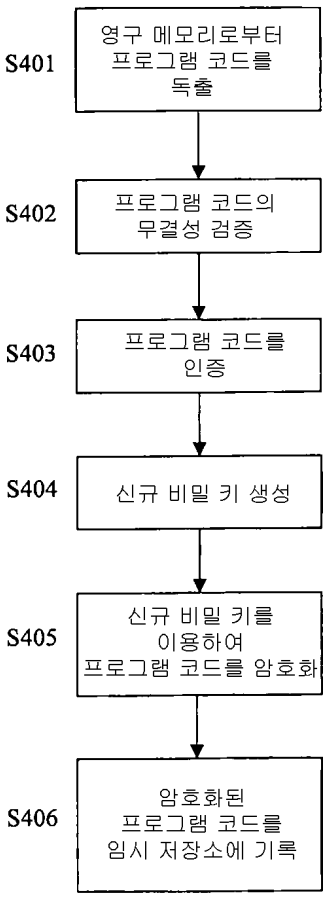
도면2



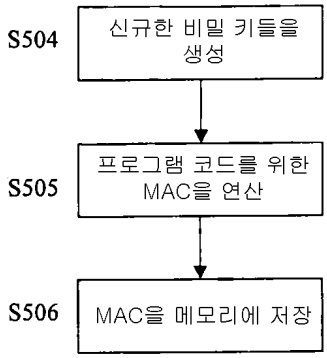
도면3


S301
영구 메모리로부터
프로그램 코드를
독출
S302
프로그램 코드의
무결성 검증
S303
신규 비밀 키 생성
S304
신규 비밀 키를
이용하여
프로그램 코드를 암호화
S305
암호화된
프로그램 코드를
임시 저장소에 기록

도면4


S401
영구 메모리로부터 프로그램 코드를 독출
S402
프로그램 코드의 무결성 검증
S403
프로그램 코드를 인증
S404
신규 비밀 키 생성
S405
신규 비밀 키를 이용하여 프로그램 코드를 암호화
S406
암호화된 프로그램 코드를 임시 저장소에 기록

도면5


S504
신규한 비밀 키들을 생성
S505
프로그램 코드를 위한 MAC을 연산
S506
MAC을 메모리에 저장

도면6