(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.[7]
G11B 20/10

(11) 공개번호 10-2004-0055724
(43) 공개일자 2004년06월26일

(21) 출원번호 10-2003-7007604
(22) 출원일자 2003년06월05일
번역문 제출일자 2003년06월05일
(86) 국제출원번호 PCT/US2001/047014
(87) 국제공개번호 WO 2002/47081
(86) 국제출원출원일자 2001년12월06일
(87) 국제공개일자 2002년06월13일

(81) 지정국 국내특허 : 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아-헤르체고비나, 바베이도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 리히텐슈타인, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀랜드, 영국, 그루지야, 헝가리, 이스라엘, 아이슬란드, 일본, 케냐, 키르기즈, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 라이베리아, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 슬로베니아, 슬로바키아, 타지키스탄, 투르크메니스탄, 터어키, 트리니다드토바고, 우크라이나, 우간다, 우즈베키스탄, 베트남, 폴란드, 포르투칼, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 아랍에미리트, 안티구아바부다, 코스타리카, 도미니카연방, 알제리, 모로코, 탄자니아, 남아프리카, 벨리즈, 모잠비크, 에쿠아도르, 필리핀,

AP ARIPO특허: 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 시에라리온, 가나, 감비아, 짐바브웨,

EA 유라시아특허: 아르메니아, 아제르바이잔, 벨라루스, 키르기즈, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크메니스탄,

EP 유럽특허: 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴, 핀랜드, 사이프러스, 터어키,

OA OAPI특허: 부르키나파소, 베넹, 중앙아프리카, 콩고, 코트디브와르, 카메룬, 가봉, 기네, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기네비쏘, 적도기네,

(30) 우선권주장 60/251,731 2000년12월07일 미국(US)

(71) 출원인 쌘디스크 코포레이션
미합중국, 캘리포니아주 94089, 써니배일, 캐스피안코트 140

(72) 발명자 카와미,바만
미국,캘리포니아95138,산호세,킬러네이서클5899

사베트샤기,파쉬드
미국,캘리포니아95138,산호세,스노우돈플레이스5634

창,로버트,씨.
미국,캘리포니아95406,댄빌레,스탠톤코트10

(74) 대리인 박경재

심사청구 : 없음

(54) 비휘발성 메모리 카드, 콤팩트 디스크 또는 다른매체로부터 기록된 오디오, 비디오 또는 다른 컨텐츠를재생하기 위한 시스템, 방법 및, 장치

요약

휴대용 장치 또는 컴퓨터에 대한 보안 소프트웨어 시스템은 단순한 인터페이스를 장치 및 컴퓨터에 제공하고 보안 매체로부터 키들 및 컨텐츠를 검색 및 다이나믹하게 해독하지만 키들의 노출을 최소화하고 이런 복잡한 프로세스들을 관리하도록 이들 자체 시스템을 생성하기 위한 디바이스 또는 컴퓨터의 제조자들에 대한 필요성을 제거한다.

대표도

도 8

명세서

기술분야

본 출원은 2000. 12.7. 자로 출원된 미국 특허 가출원 제 60/251,731 호 '비휘발성 메모리 카드, 콤팩트 디스크 또는 다른 매체로부터의 기록된 오디오, 비디오 또는 다른 컨텐츠를 재생하기 위한 보안 소프트웨어 시스템' 및, 본원과 동일자로 동시에 출원되고 본원과 동일한 발명자가 발명한 대리인 참조 번호 M 9913-1 의 '비휘발성 메모리 카드, 콤팩트 디스크 또는 다른 매체로부터 기록된 오디오, 비디오 또는 다른 컨텐츠를 재생하기 위한 시스템, 방법 및, 장치'에 관련된다.

소스 코드는 다음의 파일을 포함하는 부속물로서 37 CFR 1.52 에 따른 콤팩트 디스크에서 제출되는데, 이들 각각은 본원에 참조로서 포함된다.

Sd_security＼Sd_oem＼Makefile, 11/05/01,2KB;

Sd_security＼Sd_oem＼Readme,11/05/2001,3KB;

Sd_security＼Sd_oem＼Sd_oem.c, 11/05/2001, 6KB;

Sd_security＼Sd_oem＼Sd_oem.h, 11/05/2001, 1KB;

Sd_security＼Sd_oem＼Sd_oem.inc, 11/05/2001, 1KB;

Sd_security＼Sd_oem＼Sdtypes.h, 11/05/2001, 3KB;

Sd_security＼Sd_oem＼vssver.scc 11/05/2001, 1KB;

Sd_security＼Security＼Tstsampl＼Dotest.c, 11/05/2001, 8KB;

Sd_security＼Security＼Tstsampl＼Makefile, 11/05/2001, 4KB;

Sd_security＼Security＼Tstsampl＼Readme 11/05/2001, 3KB;

Sd_security＼Security＼Tstsampl＼Regress.c, 11/05/2001, 26KB;

Sd_security＼Security＼Tstsampl＼Sdls.c, 11/05/2001, 10KB;

Sd_security＼Security＼Tstsampl＼Sdrm.c, 11/05/2001, 5KB;

Sd_security＼Security＼Tstsampl＼Securmmc.c, 11/05/2001, 6KB;

Sd_security＼Security＼Tstsampl＼Tstsampl.inc, 11/05/2001, 1KB;

Sd_security＼Security＼Tstsampl＼vssver.scc, 11/05/2001, 1KB;

Sd_security＼Security＼Err.h, 11/05/2001, 1KB;

Sd_security＼Security＼Fsentry.c, 11/05/2001, 7KB;

Sd_security＼Security＼keyInfo.h, 11/05/2001, 84KB;

Sd_security＼Security＼Makefile, 11/05/2001, 3KB;

Sd_security＼Security＼Readme, 11/05/2001, 4KB;

Sd_security＼Security＼Scdrv.c, 11/05/2001, 29KB;

Sd_security＼Security＼Scdrv.h, 11/05/2001, 5KB;

Sd_security＼Security＼Scfs.c, 11/05/2001, 13KB;

Sd_security＼Security＼Scfs.h, 11/05/2001, 4KB;

Sd_security＼Security＼Sdsec.h, 11/05/2001, 5KB;

Sd_security＼Security＼Sdsys.c, 11/05/2001, 2KB;

Sd_security＼Security＼Security.c, 11/05/2001, 64KB;

Sd_security＼Security＼Smanager.c, 11/05/2001, 7KB;

Sd_security＼Security＼Smanager.h, 11/05/2001, 2KB;

Sd_security＼Security＼Ssmapi.c, 11/05/2001, 3KB;

Sd_security＼Security＼vssver.scc, 11/05/2001, 1KB;

Sdaudlib＼HostFunc.c, 11/05/2001, 3KB;

Sdaudlib＼Inpoutp.c, 11/05/2001, 1KB;

Sdaudlib＼mssccprj.scc, 11/05/2001, 1KB;

Sdaudlib＼plmInfo.h, 11/05/2001, 16KB;

Sdaudlib＼Sd_plm.h, 11/05/2001, 5KB;

Sdaudlib＼Sd_tkm.h, 11/05/2001, 4KB;

Sdaudlib＼Sd_types.h, 11/05/2001, 2KB;

Sdaudlib＼Sdapi.h, 11/05/2001, 2KB;

Sdaudlib＼Sdaudapi.c, 11/05/2001, 91KB;

Sdaudlib＼Sdaudapi.h, 11/05/2001, 8KB;

Sdaudlib＼Sdaudlib.dsp, 11/05/2001, 4KB;

Sdaudlib＼Sdaudlib.dsw, 11/05/2001, 1KB;

Sdaudlib＼vssver.scc, 11/05/2001, 1KB.

본 발명은 전체적으로 그리고 상세하게는 메모리 카드, 콤팩트 디스크 또는 다른 매체로부터의 디지탈 오디오, 비디오 또는 다른 컨텐츠의 보안이 이루어진 재생에 관한 것이다.

배경기술

저작권이 있는 음악을 인터넷으로, 다른 통신 시스템에 의해서, 또는 소매 매장을 통해서 전자적으로 배포하는 가능성은 음악의 인가되지 않은 복제에 대한 관심에 의해서 제한된다. 이것은 또한 비디오 뿐만 아니라 다른 오디오 컨텐츠(content)에 대해서도 마찬가지다. 컨텐츠는 궁극적인 고객에게 암호화된 형태로 공급되는 것이 전형적이며, 고객은 암호화된 컨텐츠 파일을 개인용 컴퓨터 메모리, 휴대용 재생 장치의 메모리, 쓰기 가능한 콤팩트 디스크(CD)도는 비휘발성 메모리 카드와 같은 어떤 저장 매체상에 기록한다. 컨텐츠의 제공자는 컨텐츠의 인가되지 않은 복제의 가능성을 제거하고 싶어하지만, 발생하는 복제량을 최소화시키는 조치를 취하는데 만족해야만 한다. 이것은 기록 매체상에 컨텐츠의 보호를 제공하는 것을 포함한다. 비 휘발성 메모리 카드상에 저장된 컨텐츠의 보호는 콤팩트 디스크 또는 다른 기록 가능 매체에 저장될 수 있다.

컨텐츠 데이타 저장 매체로서 사용되기에 적절한 몇가지의 상업적으로 이용 가능한 비 휘발성 메모리 카드가 존재한다. 그 하나는 콤팩트 플래쉬(CF) 카드이고, 다른 것은 멀티미디어카드(MultiMediaCard;MMC) 이며, 또 다른 하나는 MMC 카드와 밀접하게 관련된 보안 디지탈(SD) 메모리 카드이다. 이 모든 세가지와 다른 것들은 본 출원의 양도인인 캘리포니아, 서니베일의 샌디스크 코포레이션으로부터 의 다양한 저장 역량으로서 이용 가능하다. MMC 에 대한 물리적이고 전기적인 상세는 캘리포니아, 큐페르티노의 멀티미디어카드 협회(MMCA)에 의해서 때때로 업데이트되고 출판되는 '멀티미디어카드 시스템의 상세'에 주어져 있다. 1999 년 6 월과 2000 년 1 월의 버젼 2.11 과 2.2 의 상세는 각각 본원에 참고로서 포함된다. MMC 의 제품은 샌디스크 코포레이션에 의해 간행된 2000년 4 월자의 '멀티미디어카드 제품 매뉴얼' 개정판 2 에 설명되어 있으며, 이러한 매뉴얼은 본원에 참고로서 포함된다. MMC 제품의 전기적인 작동은 토마스 엔. 툼스와 미키 홀츠만이 1998 년 11 월 4 일자로 출원하고 샌디스크사가 양도 받은 특허 출원 제 09/185,649 호 및, 09/186,064 호에 개시되어 있다. 이들 출원은 모두 본원에 참고로서 포함된다.

신규의 SD 카드는 MMC 카드와 유사하며, 평면상에서 같다. 그들 사이의 기본적인 차이는 카드와 호스트 사이의 보다 빠른 데이타 전송을 가능하게 하기 위하여 부가적인 데이타 접촉을 구비한다는 것이다. SD 카드의 다른 접촉부는, SD 카드를 받아들이도록 디자인된 소켓이 MMC 카드도 받아들이기 위하여, MMC 카드의 접촉부와 같다. SD 카드와의 전기적인 인터페이스는, 호스트의 작동에 대한 변화가 양쪽 유형의 카드를 수용하기 위하여 거의 필요하지 않도록, 위에서 참조된 상세의 버젼 2.11 에 설명된 MMC 제품과 대부분 역으로 호환되도록 만들어진다. SD 카드의 전기적인 인터페이스와 그것의 작동은 2000.8.17에 제출되어 계류중인 특허 출원 09/641,023 호에 설명되어 있으며, 이것은 본원에 참조로써 포함된다.

발명의 상세한 설명

암호화된 컨텐츠는 접근하기가 곤란하며, 암호화된 컨텐츠를 가진 메모리 카 드 또는 콤팩트 디스크는 암호화된 컨텐츠와 암호화되지 않은 컨텐츠에 접근하기 위한 특정의 명령과 루틴을 필요로 하는 특정의 구조를 각각 가진다. 본 발명의 소프트웨어는 그 어떤 최초의 장비 제조자(original equipment manufacturer; OEM)도 무수한 상이한 유형의 마이크로프로세서를 가진 무수한 상이한 유형의 장치에 설치할 수 있고 그리고 가동할 수 있는 간단한 솔루션(solution)이다. 이러한 장치들은 개인용 컴퓨터로부터 카 스테레오를 위한 휴대용 장치까지 걸쳐 있으며, 암호화될 수 있는 컨텐츠에 접근하고 싶어하는 그 어떤 장치라도 구비한다. 휴대용 장치는 휴대용 오디오 플레이어 또는 휴대 전화 또는 휴대용 오르가나이저(organizer) 또는 전체적으로 그 어떤 마이크로프로세서로 제어되는 휴대용 장치일 수 있다. 저장 매체는 플래쉬 메모리 또는 그 어떤 유형의 기록 가능한 디스크일 수 있다. 장치들은 소용량 또는 대용량의 메모리를 가진 단순하거나 또는 강력한 마이크로프로세서를 가질 수 있다. 소프트웨어는 암호화의 목적으로 단지 작은 버퍼를 이용하고 필요로 하며 제한된 처리용 전력과 메모리를 가진 환경에서조차 효과적으로 가동하도록 디자인된다. 이것은 그 어떤 유형의 일반적인 목적의 마이크로프로세서에 의해서 가동될 수 있거나, 또는 DSP 또는 ASIC 와 같은 특수한 목적의 마이크로프로세서에 의해서 가동될 수 있다. 더욱이, 암호화 및, 암호 해제(보안) 엔진과 같이, 소프트웨어의 컴퓨터적으로 요구되는 부분들은 DSP 에 의해서 수행될 수 있는 반면에, 소프트웨어의 다른 부분들은 마이크로프로세서 또는 ASIC 에 의해서 수행될 수 있다.

소프트웨어는 오디오, 비디오 및, 이메이지 인터페이스를 가져서 개별 유형 의 파일들 각각의 명령을 수신한다. 이러한 인터페이스들은 재생 목록과 다른 편리한 특징을 처리하는 것을 포함하는 재생 및, 기록을 조직화(organize) 할 수

있다. 따라서, 장치가 무엇이건 간에, 단지 명령을 인터페이스에 발행할 필요만이 있으며 소프트웨어는 보안 매체로부터의 데이타를 읽거나 쓰고, 오디오 비디오 또는 이미지 엔진내에 있는 그 어떤 공지의 오디오, 비디오 또는 이미지 파일 포맷으로부터의 데이타를 디코딩하고 압축 해제(decompressing)시키는 것을 처리할 것이다.

암호화와 암호 해제는 고립된 모듈내에서 발생하여 접근하기가 매우 어렵고 따라서 매체 또는 장치로부터 파일을 복사하기 위하여 암호화 키이에 접근하기를 희망하는 인가되지 않은 사람으로부터의 그 어떤 시도에 대해서도 고립된다. 컨텐츠는 작은 부분에서만 암호 해제되며, 다이내믹(dynamic) 키이 발생과 삭제의 방법은 암호 해제 키이의 노출을 최소화시킨다.

도면의 간단한 설명

도 1 은 보안 매체상에 정보를 읽고 쓰도록 사용된 장치에 대한 설명이다.

도 2 는 보안 매체에 접근하도록 사용된 장치의 개략적인 다이아그램이다.

도 3a 는 보안 매체의 층에 대한 이론적인 설명이다.

도 3b 는 보안 매체의 메모리 셀에 대한 물리적이고 논리적인 구조에 대한 설명이다.

도 4 는 매체 키이 블록(MKB) 이미지를 그것의 커다란 구성의 부분으로 나눈 것에 대한 설명이다.

도 5 는 인증 및, 암호 해제 과정의 일부에 대한 설명이다.

도 6 은 인증 및, 암호화 과정에 대한 설명이다.

도 7 은 도 6 에 도시된 인증 키이 교환 과정에 대한 개략적인 도면이다.

도 8은 본 발명의 소프트웨어의 모듈을 설명하는 블록 다이아그램이다.

도 9 는 본 발명에 따른 오디오 트랙의 재생에 대한 순서도 개관이다.

도 10 은 도 9 의 단계에서 도 4 에 도시된 MKB 이미지 처리에 대한 순서도이다.

실시예

도 1 을 참조하면, 컨텐츠의 보호가 음악과 같은 오디오 컨텐츠에 적용되는 예시적인 시스템이 설명되어 있다. 호스트 컴퓨터 장치(11)는 도시된 바와 같이 개인용 컴퓨터(PC)일 수 있으며, 음악 또는 다른 컨텐츠를 배포하도록 소매 상점에 위치한 매점에 있는 것일 수 있다. 이러한 예에서 SD 메모리 카드(13)가 음악을 저장하도록 사용된다. 카드(13)는 응용 장치안에 삽입될 수 있으며, 본 경우에서는 카드(13)상에 기록된 음악 또는 다른 오디오 컨텐츠를 개인용 이어폰을 통하여 재생하도록 배터리로 작동되는 휴대용 장치(PD)(15)이다. 컴퓨터 유니버설 시리얼 버스(USB) 접속(17)을 통하는 것과 같이 장치(17)를 호스트(11)에 접속시킴으로써 장치(15)내에 삽입되었을때 음악은 카드(13)상에 저장될 수 있다. 이와는 달리, 재생 장치(15)에 컨텐츠를 카드(13)로 기록하는 성능이 제공되어 있지 않거나, 또는 이와는 다르게 필요로 한다면, 카드 라이터(writer)/리이더(reader)(19)는 USB 연결(21)을 통해서 컴퓨터에 연결될 수 있으며, 카드(13)는 음악을 카드상에 기록하도록 그 안에 삽입된다. 카드(13)는 다음에 라이터/리이더(19)로부터 제거되어서 휴대용 장치(15)로 삽입되어 카드상에 기록된 오디오 컨텐츠를 재생한다. 호스트(11)는 그것이 4C 엔터티(entity) 및, SD 그룹의 보안 및, 인증 프로토콜에 따라서 카드(13)의 컨텐츠 데이타로부터 쓰고 읽는데 필요한 소프트웨어를 구비할때 인가된 순응 모듈(licensed compliant module; LCM)으로 지칭된다.

예로 든 휴대용 응용 장치(15)에 있는 전자 시스템은 도 2 에 도시되어 있다. 버스(23)를 통해서 함께 작동 가능하게 연결된 것은 컴퓨팅 유니트(MCU)(25)이고, 바람직스럽게는 일부 비 휘발성 플래쉬 메모리(25A), 시스템 메모리(27)를 가지며, 이것은 고속의 랜돔 액세스 메모리(RAM)인 것이 바람직스러우며, 메모리 카드(13)와 연결된 인터페이스 회로(29)를 가진다. USB 접속(17)은 또한 선택적으로는 MCU(25)에 제공된다. 필요하다면 오디오 또는 비디오 데이타와 같은 컨텐츠 데이타를 압축 해제시키고 그리고/또는 암호 해제시키도록 디지탈 신호 프로세서(DSP)(31)가 또한 포함되는데, 이것은 압축되고 그리고/또는 암호화된 형태로 저장된다. DSP(31)는 또한 프로세서의 일부로서 포함된 그 자체의 RAM 메모리(31)를 가진다. DSP(31)는 포함되거나 또는 포함되지 않을 수 있다. 더욱이, DSP 프로세서가 포함되지 않는다면, MCU(25)의 기능성을 수행할 수 있으며, 따라서 MCU(25)는 제거될 수 있다. 리이드 온리 메모리

(ROM)(32)는 본 발명의 소프트웨어의 일부 또는 전부를 저장할 수 있다. ROM(32)내의 소프트웨어 명령과 데이타는 ROM(32)으로부터 직접적으로 수행되거나 또는 읽혀질 수 있거나 또는 처음에 장치의 회로내에 포함된 RAM 메모리 안으로 들어갈 수 있다.

기록 가능한 매체상의 컨텐츠 보호에 대한 상세는 인텔 코포레이션, 인터네 셔널 비지니스 머신 코포레이션, 마쓰시다 일렉트릭 인더스트리 및, 도시바 코포레이션(4C 엔터티)에 의해 함께 설립되었다. 4C 엔터티의 다음 3 가지 간행물이 특히 여기에 관련되어 있는데, 이것들은 본원에 참조로서 포함된다: '기록 가능한 매체 상세에 대한 컨텐츠 보호, 개론 및, 공통적인 암호 요소', 개정판 0.94, 2000년 10 월, '기록 가능한 매체 상세에 대한 컨텐츠 보호, SD 메모리 카드 북' 개정판 0.95, 2001 년 5 월 및, 'C2 블록 시퍼 상세(block Cipher Specification), 개정판 0.9 2000 년 1 월 및, ' 기록가능한 매체 상세에 대한 컨텐츠 보호, DVD 북' 개정판 0.95, 2001년 5 월. SD 메모리 카드에 대한 이들 4C 엔터티 상세를 수행하기 위한 다른 자세한 상세들은 마스시타 일렉트릭 인더스트리(MEI), 샌디스크 코포레이션 및, 도시바 코포레이션(SD 그룹)에 의해서 설정되었다.

도 3a 를 참조하면, 메모리 카드(13)는 4 개의 별개층을 가지는 것으로 생각될 수 있다. 그러한 층들은 또한 보안 매체의 다른 유형으로 존재할 수도 있다.

가장 기초적인 레벨에 있어서, 데이타는 메모리 카드(13)의 물리적인 층(13d)에서 클러스터(cluster)에 배치된 메모리 셀에 저장된다. 저작권이 있는 데이타이거나 또는 암호화의 가치가 있는 것이라면 데이타는 암호화되거나 또는 보안이 이루어진다. 보안 컨텐츠를 암호화시키고 암호 해제시키도록 사용된 키이는 또한 암호화되어서 물리적인 층의 보안 부위안에 저장된다.

본 발명의 소프트웨어는 장치내에서 작동되어, 제조자(OEM)가 암호화된 데이타 및, 키이를 포함하는 메모리 셀에 접근하기 위한 매우 특정한 명령을 프로그램할 필요 없이, 장치가 암호화된 정보를 저장하고 검색할 수 있게 한다. 이것은 암 호화된 데이타를 장치로 전송하고, 장치내의 데이타를 암호 해제시키고, 장치로부터의 요청시에 오디오, 비디오 및, 이메이지 파일을 압축 해제하고 재생하는 방법을 포함한다. 즉, 장치는 '재생 트랙'과 같은 명령을 송신할 필요만이 있다. 소프트웨어는 명령을 받아들이고, 메모리 셀내에 저장된 암호화된 데이타를 검색하고, 암호화된 키이를 검색하고, 데이타를 조직하고 암호 해제시키며, 그것을 압축해제하고 포맷시켜서, 노래를 재생할 것이다.

논리층(13c)은 물리층(13d)의 클러스터와 메모리 셀에 대한 조직 구조를 구비한다. 2 개의 층(13c, 13d)은 카드(13)의 메모리를 구비하고 논리적으로 구성한다. 카드(13)가 보안 카드이므로, 안전 층(13b)은 아래의 층에 하우징된 보안 데이타에 대한 접근을 제어하고 제한한다.

응용층(13a)은 카드안에 저장된 컨텐츠에 접근하는 장치와 통신하는 메모리 카드(13)의 일부이다. 이것은 장치의 인터페이스 또는 접촉점(39)을 통하여 그것을 수행한다. 메모리 카드(13)는 바람직스럽게는 카드의 모든 층들(13a-d)의 제어와 함께 응용층(13)의 기능성 및, 카드의 작동을 처리하는 콘트롤러를 구비한다.

전술한 상세에 따르고 도 3a 의 층(13c, 13d)에 해당하는, SD 카드(13)와 같은 기록 매체의 물리 및, 논리 구조는 도 3b 에 도시되어 있다. 카드는 메모리 셀(33)의 어레이(array)배열 및, 메모리 콘트롤러(35)를 구비한다. 사용자 데이타, 명령 및, 상태 신호들은 콘트롤러(35)와 메모리 어레이(33) 사이에서 회로(37)에 걸쳐서 통신된다. 콘트롤러(35)는 소켓에 연결된 호스트 장치와 통신되는데, 여기에서 카드는 카드상의 일련의 전기 접촉부(39)를 통해서 삽입된다.

어레이(33)의 메모리 셀은 셀들의 중첩되지 않은 4 개의 부위로 분할되는데, 이들은 개별적으로 상이한 유형의 데이타를 저장하도록 지정된다. 본 경우에 있어서, 가장 큰 저장 용량 부위(41)는 사용자 데이타, 암호화된 오디오, 비디오 또는 다른 데이타를 저장하도록 지정된다. 사용자 데이타는 암호화되지 않은 데이타를 포함할 수 있거나 그렇지 않을 수도 있다. 메모리의 시스템 부위(43)는 카드 제조자의 64 비트 매체 식별부(IDmedia) 및, 4C 엔터티에 의해 제공되는 16 매체 키이 블록(MKB)를 저장하는데, 각각의 MKB 는 4k 바이트의 최대 크기를 가지며, 모든 것은 카드 제조자에 의해서 사전 기록된다. 16 MKB 들중 하나는 오디오 사용자 데이타와 함께 사용되도록 지정되며, 다른 것은 비디오 사용자 데이타용으로, 그리고 다른 것은 이미지 데이타용으로 지정된다. 시스템 부위(43)는 카드의 외측으로부터 읽혀지도록 접근될 수 있는 쓰기 보호된 부위이다. 은폐 부위(45)는 시스템 부위(43)내에 저장된 16 개의 개별 매체 키이 블록(MKB)에 해당하는 16 개의 사전 기록된 매체 유일의 키이(media unique key; Kmu)를 유지한다. 은폐 부위(45)는 쓰기 보호된 부위여서 메모리 카드 자체에 의해서만 접근될 수 있다. 보호 부위(47)는 성공적으로 명백한 상호 인가가 발생된 이후에만 접근될 수 있는 읽기/쓰기 부위이다. 임의 선택 타이틀 키이(randomly picked title key; Kt)와 복사 제어 정보(CCI)는 암호화된 형태로 보호 부위(47)내에 저장된다. 사용자 데이타 부위(41)내에 저장된 컨텐츠의 각 일부(파일)는 보호 부위(47)내에 암호화된 형태로 저장된 유일의 타이틀 키이로 암호화된다. 보호된 부위(47)내에 저장된 타이틀 키이와 CCI 는 매체 유일의 키이에 의해서 함께 연관되어 암호화되는데, 이것은 각 메모리 카드에 대해서 유일하며 그것의 은폐 부위(45)내에 저장된다. 사용자 데이타 부위(41)의 파일 시스템은 통상적으로 보통의 FAT 파일 시스템이다.

카드 메모리의 시스템 부위(43)내에 저장된 바와 같은 매체 키이 블록(MKB)는 인접한 레코드의 시퀀스를 포함하는데, 그러한 하나의 레코드는 도 4 에 도시되어 있다. 전체 MKB 이미지(49)는 64 Kbyte 이다. 이것은 512 바이트의 128 개 덩어리로 나뉘어지며, 그것이 처음에 기록하는 것의 전부 또는 일부를 포함하며 도면에서 MKB 덩어리(50)로 표시된 덩어리(1)는 그것의 구성 부분을 나타내도록 확대되어 있다. 덩어리(50)는 또한 다중의 레코드를 포함할 수도 있다. 제 1 피일드(51)는 레코드 유형을 포함하고, 제 2 피일드(53)는 레코드의 전체 길이를 포함하며, 나머지 피일드(55)는 키이 자체를 포함한다. 레코드 유형 및, 길이 피일드(51,53)내의 데이터는 암호화되지 않는다. MKB 의 각 레코드는 전체 길이내의 다중의 4 바이트이다. 도 5 의 블록(57)에 의해서 도시된 바와 같이, MKB 키이 레코드는 휴대 장치(PD), 인가된 순응 모듈(LCM) 또는 저장된 컨텐츠를 읽거나 또는 프로그래밍하도록 메모리 카드를 사용하는 다른 장치에 저장된 장치 키이에 의해서 암호 해제된다. 장치 키이(Kd1, Kd2, Kd3...)는 장치의 제조자에 의해서 도 2 의 휴대용 오디오 재생기의 MCU(25)내에 있는 비휘발성 플래쉬 메모리와 같은 응용 장치의 메모리 안으로 씌어진다. 장치 키이들은 4C 엔터티에 의해서 장치 제조자들에게 제공되며 신뢰성 있게 유지된다. 주어진 응용 장치내에 저장된 장치 키이들의 수는 장치의 유형에 달려있다.

도 5 의 처리를 수행하는 응용 장치(PD, LCM 또는 다른 장치)는 블록(57)의 암호화의 일부로서 매체 키이(Km)를 계산하는데, 이것은 도 9 및, 도 10 과 관련하여 더욱 상세하게 설명될 것이다. 삽입된 메모리 카드의 시스템 부위로부터 읽혀진 MKB 의 각 레코드(도 4)는 이러한 방식으로 항상 처리된다. MKB 의 처리가 완성된 이후에, 가장 최근에 계산된 Km 값은 블록(57)의 비밀 매체 키이 출력으로서 취해진다. 이러한 매체 키이(Km) 및, 매체 식별부(IDmedia)는 도 5 의 블록(59)에 의해서 표시된 바와 같이 C2 일방향 함수의 사용에 의해 조합되어, 매체 유일의 키이(Kmu)를 발생시킨다. 이러한 처리의 상세한 사항은 이전에 참고된 4C 엔터티의 간행물을 참고함으로써 얻을 수 있다.

도 6 은 도 3 의 메모리 공간 배당을 받은 메모리 카드(13)로 오디오 컨텐츠를 기록하거나 또는 그로부터 오디오 컨텐츠를 재생할 때 발생하는 모든 인증 및, 암호화 처리를 도시한다. 개인 컴퓨터 또는 다른 LCM (63)에서 발생하는 처리는 오디오 또는 다른 컨텐츠를 카드(13)상으로 레코딩하기 위해서 도시되어 있다. 마찬가지로, 휴대용 오디오 또는 다른 응용 장치(65)의 처리는 레코딩된 컨텐츠를 카드(13)로부터 읽는 것에 대해서 도시되어 있다. 양쪽에는 도 5를 참조하여 설명된 처리가 포함되어 있는데, 처리 블록(57,59)은 응용 장치(65)의 일부이며 그에 대응하는 처리 블록(57', 59')은 컨텐츠 레코딩 시스템(63)의 일부이다.

레코딩 컨텐츠의 일부로서, 임의로 할당된 키이(Kt)는 오디오의 하나의 파일(일부)를 암호화하도록 암호화 모듈(69)에 의해 사용되기 위한 라인(67)에서의 입력이거나 또는 라인(71)에서의 다른 컨텐츠 입력이다. 암호화된 파일은 다음에 메모리 카드(13)의 사용자 데이터 부위(41)내에 저장된다. 타이틀 키이가 레코딩된 컨텐츠를 암호화하는데 이용될 있기 위하여, 이전에 설명된 바와 같이, 타이틀 키이(Kt)의 암호화된 버전이 보호된 카드 메모리 부위(47)내에 저장된다. 타이틀 키이(Kt)의 암호화된 버전은 또한 시스템 메모리(27), MCU(25)의 RAM 메모리(25A) 또는 DSP(31)의 RAM 메모리(31A)들중 어느 것에 저장된다. 암호화된 타이틀 키이(Kt)를 장치의 메모리내에 저장하는 것은 보호된 카드 메모리 부위(47)에 접근할 필요성을 제거한다. 이것은 각각의 읽기를 위해서 보호된 부위(47)에 접근하는 것과 비교하여 상당한 시간 및, 처리 용량을 절감시키기 때문에 중요하다. 이것은 도 9를 참조하여 이후에 설명될 것이다. 타이틀 키이(Kt)와 복사 제어 정보(CCI)는 메모리 카드(61) 상의 LCM(63)과 모듈(81)내의 일련의 암호화 모듈(75,55 및, 79)에 의해서 암호화된다. 매체 유일의 키이(Kmu)는 모듈(77)에 의해서 사용된다. 인증 키이 교환(AKE) 모듈(83)은 모듈(59')에 의해서 계산되고 카드(61)의 은폐 부위(45)내에 저장된 매체 유일의 키이(Kmu)를 조합하여 각 모듈(79,81)에 의해 사용된 세션 키이(session key, Ks)를 발생시킨다. 응용 장치(65)가 기록된 암호화 컨텐츠를 암호 해제시키기 위하여, 동일한 참조 번호로 표시되지만 프라임(')이 부가되어 표시된 대응 모듈은 암호화 과정의 역을 수행하도록 사용된다.

도 7 은 메모리 카드의 보호된 부위(47)에 접근하기 위한 기술을 나타내는데, 카드와 일부 LCM 또는 응용 장치 사이에서 인증 및, 키이 교환(AKE) 시도 응답 프로토콜을 사용하는 것이다. 이러한 인증이 성공적일 때, 카드와 다른 모듈 또는 장치는 보안이 이루어진 공통의 세션 키이(Ks)를 공유한다. 전술한 과정과 프로토콜에 대한 상세한 설명은 이전에 알려진 4C 엔터티의 간행물을 참고하여 이루어질 수 있다.

SD 메모리 카드의 인증 부위에 대한 접근을 수행하는 것은 이전에 언급된 바와 같이 OEM 들이 4C 엔터티로부터 인가를 받아야만 하는 비밀 장치 키이를 사용할 것을 필요로 한다. 이러한 키이 값들을 보호하고 그들의 노출을 SDK SW 100 소프트웨어 층내에 제한하는 것은 소프트웨어 디자인에 있어서 중심적인 고려들중 하나이다. 이러한 키이들(그리고 세션 키이와 같은 다른 결과적인 값)을 단일의 내부 모듈에 고립시키면서 SD 메모리 카드 장치 드라이버와 같은 보안 매체가 상기 값들에 의존하는 작동을 수행하는 것은 강력하고 보안이 이루어진 방법론에서 달성된다. 일단 다시, SD 메모리 카드는 본 발명을 나타내도록 사용되며, 그러나 본 발명은 CD, 또는 카드나 원격 위치된 저장 장치내에 있을 수 있는 다른 보안이 이루어진 메모리에서 사용될 수 있다.

도 8은 상기에 언급된 과정으로써 암호화된 정보에 접근하기 위하여 휴대용 장치 또는 LCM 에서 작동되도록 디자인된 소프트웨어 시스템의 구현예를 도시한다. SanDisk 소프트웨어(SW100)는 완전한 턴-키이(turn-key) 소프트웨어 솔루션이어서 OEM 음악 재생기와 레코더들이 보안이 이루어진 디지탈(SD) 메모리 카드를 포함하는 보안 매체를 용이하게 지원할 수 있게 한다.

도8에 있어서, 휴대용 장치(15)내에 장착된 것으로 도시된 SW100 은 SD 카드(13)에 접근하기 위한 것이다. SW100 은 개인용 컴퓨터와 같은 그 어떤 인가를 받은 순응 모듈(licensed compliant module)내에 설치될 수도 있다. 오디오 인터페이스(105), 비디오 인터페이스(110) 및, 이메이징 인터페이스(115)가 장치로의 통 신의 지점들이다. 이러한 인터페이스들은 장치를 위한 통신의 단일 지점을 제공하며 일반적으로 장치로부터 단순한 명령을 수신함으로써, 장치가 보안 매체로부터 암호화된 데이터를 얻고, 데이터를 암호 해제시키고 그리고 처리하는 복잡성과 연관될 필요가 없게 한다. 모든 이러한 복잡한 과정은 SW100 에 의해서 취급된다. 인터페이스(105, 110, 115)들도, 예술가의 이미지 같은 이미지를 예술가의 노래 또는 재생 목록에 연관시키는 것과, 관리 재생 목록 같은 재생의 배치를 관리한다. 응용 프로그래밍 인터페이스(API)(130A)는 명령 디스패처(command dispacher)(CD)(130)내에 상주한다. CD(130)와 API(130a)는 인터페이스(105,110, 115)로부터 명령을 수신하여, 정보를 인터페이스로 중계하며, 모든 과정들을 조직화하는데, 상기 과정들은 SW100에서 발생한다- 장치(15)의 과정은 필요한 모든 암호화, 암호 해제 및, 압축 알고리듬과 함께, 보안 매체상에 저장된 컨텐츠의 레코딩과 재생에 관련된다. 이런 고른 인터페이스 및 제어에 대한 보다 많은 정보에 대해서는, 동시계류중이고, 이 출원과 동일자 출원되고 이출원과 동일한 발명자를 가지는, 미국 특허출원, 대리인 서류넘버, M-9913 US, 명칭'SYSTEM, METHOD, AND DEVICE FOR PLAYING BACK RECORDED AUDIO, VIDEO OR OTHER CONTENT FROM NON-VOLATILE MEMORY CARDS, COMPACT DISKS OR OTHER MEDIA'를 참고하시오.

SD 오디오 엔진(SDAE)(140), SD 비디오 엔진(SDVE)(150), 및, SD 이메이지 엔진(SDIE)(160)은, CD(130)로부터의 명령을 수신시에, 보안 매체상에 상주하는 오디오, 비디오, 이미지 컨텐츠를 각각 처리한다. 이것은 SDAE(140)가 AAC, WMA 및, MP3 와 같은 오디오를 위한 공지의 그 어떤 포맷이라도 처리할 수 있다는 점을 의 미한다. 마찬가지로, 이것은 SDVE(150)가 윈도우즈 미디어 파일과 같은 비디오 클립 또는 리얼 네트워크 파일 MPEG 과 같은 공지의 그 어떤 포맷이나 비디오 파일의 그 어떤 공지의 유형도 처리할 수 있다는 것을 의미한다. 마지막으로 SDIE(160)는 TIF, GIF, JPEG, 비트맵등과 같은 공지 유형의 그 어떤 이메이지 파일도 처리할 수 있다. 각 인터페이스는 보안이 이루어진 API(SAPI)와 보안이 이루어지지 않은 API(NSPI)를 가진다. 처리된 컨텐츠는 암호화될 수 있거나 또는 암호화되지 않을 수 있다. 암호화된 컨텐츠는 SAPI (140A, 150A, 160A)를 통해서 접근된다. 이러한 SAPI 는 SanDisk 보안 매니저(SSM)(180)과 통신한다. 보안 컨텐츠와 관련된 모든 명령은 SSM(180)을 통해서 전달된다. 이후에 보다 상세하게 설명되는 보안이 이루어진 디지털 보안 엔진(SDSE)(175)은 모든 암호화 및, 암호 해제 과정을 취급한다. 매체를 인증하고 컨텐츠를 암호 해제시키는데 사용된 키이들은 SDSE (175)내에 포함되어 그에 의해서만 전적으로 처리된다. 카드상에 상주하는 암호화되지 않은 컨텐츠는 NSAPI (140B, 150B, 160B)를 통해서 접근된다. 이러한 NSAPI 들은 매체상의 암호화되지 않은 컨텐츠에 접근하기 위하여 비 보안 파일 인터페이스(NSFI)(170)과 통신한다.

저장 매체내의 데이터를 읽고 쓰기 위하여, NSFI(170)와 SDSE(175)는 장치 구동부(190)와 통신한다. SD 카드의 예에 있는 장치 구동부(190)는 SD 카드(13)의 장치 인터페이스(39)의 접촉부로의, 그리고 그로부터의 신호를 관리하고 구동시킨다. 장치 구동부(190)는 다양한 장치 또는 매체의 장치 인터페이스에 대한 특정 유형에 주문 제작될 것이다. 메모리 카드 장치의 경우에, 구동부(190)는 장치(15)상 에 위치된 접촉부로의 그리고 그로부터의 신호를 관리하고 구동한다. 광 매체의 경우에, 장치 구동부(190)는 광 픽업 유니트(optical pick-up unit)를 포함하는 다양한 하드웨어 구성부로의 그리고 그로부터의 신호를 관리하고 구동한다. 장치 구동부(19)는 보안 장치 구동부 인터페이스(SDDI)(190a)와 비 보아 장치 구동부 인터페이스(NSDDI)(190B)를 포함한다. SDDI(190A)와 NSDDI(190B)는 장치 구동부(190)내에서 서로로부터 고립된다. SDDI(190A)는 SDSE(175)와 전용으로 통신하는 반면에, NSDDI(190B)는 NSFI(170)와 전용으로 통신한다.

장치 키이들과 SD 오디오 보안 계획에 중심적인 다른 값들은 하나의 제한된 보안 소프트웨어 모듈인, SD 보안 엔진(SDSE)(175)내에 하우징된다. 이러한 값들의 모든 조작은 이러한 모듈에 단독으로 제한된다. 값들은 SDSE(175) 위의 소프트웨어 층 안으로 또는 밖으로 결코 통과되지 않는다. 이러한 키이들을 포함하는 보안 서비스에 대한 모든 요청은 이러한 보안 모듈을 보호하는 SSM(180)에 의해서 제어되고 모니터된다. 보안 모듈의 아래에, SD 메모리 카드 장치 구동부(190)는 보안 접근을 수행한다. 이러한 구동부 서비스에 대한 요청은 비밀의(private) 구동부 보안 인터페이스인, 보안 장치 구동부 인터페이스(SDDI)(190a)를 통해서 이루어지는데, 이것은 보안 모듈에만 알려진다. SDSE(175)는 이러한 인터페이스(190a)를 사용하여 Get Media Key Block(MKB)와 같은 특수한 보안 명령을 수행한다. 비 보안 장치 구동부 인터페이스(NSDDI)(190B)도 장치 구동부(190)를 사용하여 카드(13)의 사용자 부위(41)내에 있는 그 어떤 암호화되지 않은 파일에 접근한다.

SW100 의 보안의 구성은 그것의 키이의 보안내에 거주한다. 비밀의 '소프트 키이(soft key)'는 일시적인 보안 부위에 장기간 저장되지 않는데, 이것이 키이를 손상시킬 가능성을 증가시키고 따라서 암호화된 컨텐츠의 손상 가능성을 증가시키기 때문이다. SW100 은 필요한 키이(또는 '소프트 키이')를 동적으로 발생시키고 그러한 특정 키이들에 대한 즉각적인 필요성이 없을 때 그들을 삭제하는 SDSE(175)내의 계획을 이용한다.

SW100 의 보안의 구성은 그것의 키이의 보안내에 거주한다. 비밀의 '소프트 키이(soft key)'는 일시적인 보안 부위에 장기간 저장되지 않는데, 이것이 키이를 손상시킬 가능성을 증가시키고 따라서 암호화된 컨텐츠의 손상 가능성을 증가시키기 때문이다. SW100 은 필요한 키이(또는 '소프트 키이')를 동적으로 발생시키고 그러한 특정 키이들에 대한 즉각적인 필요성이 없을 때 그들을 삭제하는 SDSE(175)내의 계획을 이용한다.

SW100 의 작용이 이제 보다 상세하게 설명될 것이다. SW100내에서, OEM 의 4C-인가를 받은 장치 키이가 사용될 것을 요구하는 장치로 명령이 내려진다. 이러한 키이들의 모든 처리는 SSM(180) 아래에 하우징된 SDSE(175) 모듈로만 제한된다.

SSM(180)이 보안 서비스에 대한 요청을 수신하였을때, 이것은 명령 요청 패킷(packet)을 SDSE(175)내의 처리_보안(process_security) 기능으로 통과시킴으로써 그것을 수행한다. 키이 값은 요청 패킷내에 결코 포함되지 않거나 또는 SDSE(175)의 위에 있는 소프트웨어 층에 노출되지 않는다.

SDSE(175)에 의해서 내부적으로 필요할때, 장치 키이들은 기능 호출을 통해서 OEM에 공급된 라이브러리(library) 내로 검색된다. SDSE 의 라이브러리(175)인 security.lib 은, 암호해제된 키이가 시스템의 보안 부위내에 상주하는 시간을 감소시키도록 설계된, 다음의 API 들을 포함한다.

1) SEC_AKE API;

2) SEC_ENC_TKEY API;

3) SEC_DEC_TKEY API;

4) SEC_GETCCI API;

5) SEC_UPDATECCI API.

SW 100 의 기능성과 구조는 본 출원의 내용중에 설명되어 있으며 보다 상세하게는 상기의 API들(1-5)의 기능성이 도 9 의 플로우차트내에 도시되어 있다. API들은 그들이 수행하는 해당 기능들 다음에 도시되어 있다. 모든 SW100 뿐만 아니라 이들 API 들의 수행에 대한 보다 상세한 내용은 본 출원의 부속물로 제출된 소스 코드에서 볼 수 있다. 그 소스코드는 이런 응용의 일부를 형성하여 이런 레퍼런스에 의해 명백히 통합된다.

장치 키이는 일단 획득되면 SD 메모리 카드로부터의 매체 키이 블록(MKB)와 조합되어 '매체 키이(media key)'를 형성한다. 이러한 값은 차후의 요청을 처리하는데 사용되도록 SDSE(175)내에 유지된다. 그러나, '유일의 매체 키이(Kmu)는 결코 SDSE(175)의 내측에 유지되지 않는다. 모든 보안 접근을 위한 기초를 형성하는 이러한 값은 실시간에 기초하여 (그리고 캐쉬(cache)되지 않으면서) 별도의 보안 예방 조치로서 항상 계산된다. SDSE(175)내의 키이 처리의 상세한 설명은 다음과 같다.

암호화 과정은 일반적인 용어에 있어서 보안 매체상에 위치된 컨텐츠의 인가되지 않은 복사를 중지시키도록 설계된 것이다. 본 발명에는 이것을 달성하기 위한 많은 특징들이 있다. 처음에, 예를 들면 노래와 같은 전체 파일은 즉시 암호 해제되지 않으며 그것이 손상되기 쉬운 메모리내로 저장된다. 휴대용 장치는 버퍼를 할당하며 SDSE(175)는 암호화된 컨텐츠의 덩어리를 한번에 읽고, 그것을 암호 해제시키며, 다음에 파일의 끝에 도달할때까지 계속해서 같은 버퍼에 쓴다.

도 6 및, 도 7 에 도시된 바와 같이, 매체 유일의 키이(Kmu)와 타이틀 키이(Kt)는 컨텐츠를 암호 해제하도록 마지막으로 사용된 키이들이다. 타이틀 키이를 보호하는 많은 방법이 있다. 키이를 장치(15)의 매우 보안이 이루어진 부위에 저장하면, 암호화된 버퍼가 읽혀지고 암호 해제될때마다 다른 것은 카드(13)의 보호된 부위(47)로부터 타이틀 키이를 읽게 된다.

도 9를 참조하면, 단계(205)에 있어서, 도 4 에 도시된 바와 같은, 64 킬로바이트인 MKB 이미지가 도 6 에 도시된 바와 같이 매체 키이(Km)를 처리하도록 읽혀져서, 매체 유일의 키이(Kmu)를 산출한다. 이러한 단계는 도 10 에 더 상세하게 설명되어 있으며 이것은 이후에 설명될 것이다. 장치 및, 매체의 상호 인증이 단계(205)에서 완료된 이후에, AKE 과정을 겪어서 세션 키이(Ks)를 산출하게 되는데, 상기 세션 키이는 단계(210)에서 (장치가 턴온(turn on)되거나 또는 활성 상태에 있는 한) 세션(session) 동안에 단지 사용될 수 있다. AKE 과정은 도 6을 다시 한번 참조함으로써 볼 수 있다. 단계 213 에 있어서, 매체 유일의 키이(Kmu)는 삭제된다. 단계(215)에 있어서, 세션 키이(Ks)는 메모리 카드(13)의 보호된 부위(47) 에 저장된 이중으로 암호화된 타이틀 키이(E(E(Kt)))를 암호 해제시키는데 사용된다. 결과는 한번 암호화된 타이틀 키이(E(Kt)) 이다. 단계 220에서, 이러한 암호화된 타이틀 키이(E(Kt))는 장치(15)의 메모리내에 저장된다. E(Kt) 는 시스템 메모리(27), MCU(25)의 RAM 메모리(25A), 또는 DSP(31)의 RAM 메모리(31A) 내에 저장될 수 있다. 타이틀 키이(Kt)는 각 타이틀에 대해서 특정적인 것으로서, 본 발명을 예시하도록 사용된 도 9 및, 오디오의 영역에서는 트랙으로서 지칭된다. 각 트랙은 다중의 파일들로 만들어질 수 있는데, 예를 들면, 긴 클래식 노래의 경우에 그러하다. 커다란 비디오 클립에 대해서는, 타이틀이 많은 파일을 포함할 수 있다. 따라서, 트랙의 암호화된 컨텐츠의 모든 차후의 읽기 및, 암호 해제에 대해서, 타이틀 키이는 그것이 지역 메모리내에 저장되기 때문에 메모리 카드로부터 검색될 필요가 없으며, 귀중한 시간 및, 컴퓨팅 자원이 절감될 수도 있고, 동시에, 타이틀 키이는 보

안 목적을 위해서 암호화되어 유지된다.

단계(225)에 있어서, 트랙의 일부가 재생된다. 이러한 부분은 트랙을 포함하는 파일의 그 어떤 것일 수 있다. 단계(225a)에 있어서, 매체 유일의 키이(Kmu)가 다시 한번 계산된다. 단계(225b)에 있어서, 지역 메모리에 저장된 암호화된 타이틀 키이가 암호 해제된다. 다음에, 단계(225c)에 있어서, 타이틀 키이는 카드 메모리 카드(13)의 사용자 부위(41)로부터의 컨텐츠를 포함하는 장치(15)의 버퍼로부터 컨텐츠를 암호 해제시키도록 사용된다. 버퍼가 암호 해제된 직후에, 타이틀 키이는 단계(225d)에서 삭제되며 매체 유일의 키이는 단계(225e)에서 삭제된다. 단계(225d)와 단계(225e)의 순서는 중요하지 않지만, 트랙의 일부를 읽는데 드는 시간동안에만 양쪽 키이들이 노출되는 것이 중요하다. 이러한 부분은 재생(암호 해제되고, 압축 해제되고, 그리고 디코딩된) 컨텐츠, 오디오 또는 다른 것의 1 초의 일부내지는 약 10 초에 이르기까지의 그 어느 곳일 수 있다. 바람직스럽게는 이것이 2 초이다. 그 부분을 읽는데 드는 시간은 처리 속도와 장치의 버퍼 크기를 포함하는 많은 인자에 달려있다. 이전에 설명된 바와 같이, SW 100 은 MCU(25) 또는 DSP(31)에 의해서 수행될 수 있고 장치(15)의 메모리(27, 25A, 31A 또는 32)들중 어느 것에 저장될 수 있으며, 따라서 처리 시간이 변화할 수 있다. 이것은 트랙의 모든 부분들이 단계(230)에서와 같이 읽혀질때까지 반복된다. 일단 모든 부분들이 읽혀졌다면, 만약 재생이 계속될 경우에 시스템은 단계(235)에서와 같이 다음의 트랙으로 이동할 수 있다. 이것은 예를 들어 사용자가 전체의 재생 목록을 재생하도록 선택하였다면 그렇게 될 수 있다는 것이다.

트랙의 모든 부분들이 읽혀져서 다음 트랙을 읽는 것이 시작될 때, 과정은 단계(215)에서 다시 시작될 것이며 카드(13)의 보호된 부위(47)로부터 다음의 이중으로 암호화된 타이틀 키이를 검색할 것이다. 이것은 다중의 트랙을 포함하고 있는 전체 재생 목록을 재생하도록 사용자가 장치를 작동 상태에 설정하였을 경우에 일반적인 경우이다. 세션이 폐쇄되면 (즉, 장치(15)가 턴온 되거나 턴오프(turn on or trun off)된다면), 새로운 세션 키이가 발생되어야 할 것이며 과정은 단계(210)에서 시작될 것이다. 메모리 카드가 제거되거나 또는 새롭게 삽입된다면, 장치와 매체는 다시 인증을 받아야 할 것이며 과정은 트랙을 읽기 위하여 단계(205)에서 다시 시작할 것이다.

도 10 은 위에서 설명된 도 9 의 단계(205)에서 매체 키이 블록을 처리하는 작동을 처리한다. 도 4에서 도시되었던 것과 같이, MKB 이미지(49)는 길이에 있어서 64 Kbyte 이다. MKB를 계산하도록 전체 이미지(49)를 즉시 읽는 것은 비효율적이며, 커다란 RAM 과 긴 처리 시간을 필요로 한다. 본 시스템은 RAM 요건을 감소시키고 처리 시간을 줄인다. MKB 이미지(49)는 덩어리(1) 내지 (128)로 분할된다. 각 덩어리는 512 바이트이며 MKB 의 레코드의 상이한 4 개 유형들중 하나를 포함할 수 있다: 0 ×81 로서 알려진 입증 매체 키이 레코드(the verify media key record, VMKR); 0×01 로서 알려진 계산 매체 키이 레코드(the calculate media key record, CMKR); 0×82 로서 알려진 조건부 계산 매체 키이 레코드(the conditionally calculate media key record, CCMKR); 또는 0×02 로서 알려진 종료 매체 키이 레코드(the end media key record, EMKR). 이러한 레코드들은 위에서 언급된 4C 엔터티의 기록 가능한 매체의 보호(CPRM) 명세에 설명되어 있다.

이러한 예에서, 덩어리의 길이와 버퍼의 길이는 같다. 그러나, 버퍼의 길이와 덩어리의 길이는 256 바이트 내지 4096 바이트이다. 각 레코드는 레코드 유형에 기초하여 특정의 작동을 수행하도록 검사되며 특정의 데이터는 매체 키이를 획득하도록 이후를 위해서 예비될 것이다. 레코드 길이는 레코드가 식별될때마다 버퍼 오프셋의 전체 길이에 부가된다. 덩어리의 수는 전체 길이를 덩어리 길이로 나눔으로써 계산된다. 덩어리의 수는 선택된 덩어리 데이터의 매체 키이 블록에 대한 인덱스(index)이다. 전체 길이의 나머지는 선택된 덩어리 데이터에 대하여 오프셋(offset)된다. 행과 열이 사용되어 암호화된 매체 키이와 조건부의 암호화된 매체 키이가 어느 곳에 있는지를 계산한다. 이러한 암호화된 키이들이 보관되어서 전자 코드북 모드 알고리듬에서의 암호 해제 C2 암호가 수행되어 매체 키이를 획득하도록 수행된다. 이러한 매체 키이는 다음에 올바른 최종의 매체 키이(Km)를 위해서 검증된다.

레코드의 수와 연관된 MKB 로부터 매체 키이(Km)를 획득하기 위한 MKB 덩어리마다 필요한 읽음(read)의 수인 T 는 아래와 같다.

레코드의 수 < T < (레코드의 수*2)

T: MKB 덩어리에 접근하는데 필요한 횟수

각 레코드는 상이한 길이와 데이터 값을 가진다. 각 레코드의 정보는 2 번 읽는 것 안에서 획득될 수 있다. 4 개의 레코드가 있기 때문에, 4 회의 읽음과 8 회의 읽음 사이에 MKB 덩어리를 처리하고 레코드를 획득할 필요가 있을 것이다.

따라서 읽음의 수인 T 는,

4 < T < 8 이다.

MKB 데이터의 512 바이트에 접근하는데 N ms 가 걸린다고 가정하자. 제 1 방법으로부터 매체 키이를 획득하도록

전체 64K 의 MKB 이미지에 접근하는데는 (128 * N)ms 가 걸릴 것이다. 제 2 의 방법으로부터, 최악의 경우인 시나리오에서는 매체 키이를 획득하는데 (8*N)ms 가 걸린다. 따라서 이러한 계획을 사용함으로써 상당한 시간이 절감된다. 평균적으로, 매체 키이(Km)를 획득하려면, 읽음의 수는 4 내지 6 의 범위에 있으며, 필요한 시간은 위에서 도시된 것보다 비례적으로 작다.

도 10에서 확장된, 도 9 의 단계(205)는, 최종의 매체 키이가 단계(205.75) 에서 생성될때까지 또는 매체가 단계(205.80)에서 거절될때까지 수행된다. 128 개의 덩어리 모두가 읽혀질 필요는 없으며, 덩어리마다 512 바이트 모두를 매체 키이를 계산하기 위하여 읽을 필요는 없다. MKB 데이터를 처리하는 것은 데이터의 덩어리를 한번에 요청하는 것과, 특정의 덩어리내에 있는 소기 위치를 지향하는 것과, 획득된 값을 계산하는 것을 필요로 한다. MKB 데이터 모두가 필요한 것은 아니다. 도 10 에 도시된 알고리듬은 MKB 데이터의 어떤 덩어리가 필요한지, 어떤 레코드가 처리되어야 하는지, 그리고 어디에 암호화된 데이터가 위치하는지를 계산하도록 수학적인 계산을 제공할 것이다.

단계 205.5 에 있어서, 버퍼 포인터(buffer pointer)는 데이터 버퍼로 설정되며 버퍼 오프셋은 클리어(clear)된다. 다음에 단계(205.10)에 있어서, 덩어리의 수는 그것이 최대 덩어리 수와 같은지 또는 그보다 큰지를 알기 위해서 점검된다. 만약 그렇다면, 에러가 단계(205.15)에서 반환될 것이다. 만약 그렇지 않다면, 덩어리의 수는 증가될 것이며 새로운 데이터가 단계(205.20)에서 버퍼로 로딩될 것이다. 다음에 버퍼 오프셋은 단계(205.25)에서 업데이트될 것이다. 따라서, 포인터는 올바른 위치 덩어리 수 더하기 오프셋)에 설정될 수 있다. 단계(205.30)에 있어서, 버퍼 포인터는 버퍼 오프셋으로 설정된다. 단계(205.40)에 있어서, 버퍼가 읽혀져서 포인터가 위치되는 오프셋에서 시작된다. 다음에 시스템은 그것이 어떠한 유형의 레코드를0 읽고 있는지를 결정할 것이다. 단계(205.40)에서 알 수 있는 바와 같이, 시스템은 우선 어떠한 유형의 레코드가 읽혀지고 있는지, 그리고 어떠한 레코드 길이가 그 레코드와 연관되어 있는지를 점검한다. 각 레코드의 레코드 길이는 버퍼 포인트가 차후의 레코드를 읽으면서 위치되어야 하는 곳을 결정하는데 사용될 것이다. 이것은 단계(205.49)에 의해서 반영되어, 버퍼의 오프셋을 업데이트시키고 버퍼 포인트를 새로운 오프셋에 설정시킨다.

레코드가 단계(205.42)에서 도시된 것과 같은 CMKR 이라면, 시스템은 암호화된 매체 키이(Km)가 단계(205.49)에서 위치된 곳인 올바른 MKB 위치에 대한 오프셋과 버퍼 덩어리의 수를 업데이트시킨다. 각 카드는 16 MKB를 가진다. 따라서, 시스템은 오프셋이 될 것인데 여기에서 암호화된 매체 키이는, 특정의 MKB 덩어리의 수로 가고, 버퍼(16 블록 × 512 바이트)를 할당하고, 암호화된 매체 키이를 읽도록 각 블록내의 오프셋으로 가게 된다. 다음에 시스템은 단계(205.50)에서 매체 키이를 암호 해제(계산)시키도록 장치(15)로부터 공급된 장치 키이(Kd)를 사용한다. 일단 매체 키이가 계산되면 다음의 단계는 매체 키이를 검증하게 된다.

레코드가 단계(205.44)에서 평가된 바와 같이 VMKR 이라면, 단계(205.50)에서의 제 1 시도, 또는 단계(205.65)에서의 제 2 시도에서, 이전에 계산되었던 매체 키이는 단계(205.55)에서의 기준 매체 키이(Km)와 비교될 것이다. 이것을 하기 위하여, 기준 매체 키이는 처음에 지역적으로 저장될 것이다. 키이가 같다면, 6 진법에 있어서 DEADBEEF 인, 통과(pass)가 반환될 것이며, 시스템은 조건부로 매체 키이를 계산할 필요가 없다. 다음의 레코드를 읽는 것을 어디에서 시작할지를 계산하기 위하여, VMKR 의 레코드 길이가 사용되어 버퍼 포인터를 다음의 레코드로 움직인다. 만일 키이가 같지 않다면, 언제 CCMKR 레코드가 단계(205.46)에서 읽히는지 다시 계산될 것이다. 이러한 레코드가 읽혀진다면, 버퍼 지점이 단계(205.49)에서 업데이트된 버퍼 오프셋에서 읽혀지도록 설정된 이후에 매체 키이가 다시 한번 단계(205.65)에서 계산될 것이며, 다음의 VMKR 이 읽혀질 때를 차후에 검증할 것이다. CCMKR 이 계산되는 최대의 시간 수는 시스템에 의해서 바람직스럽게는 한번 설정될 수 있다.

제 1 의 계산은 CMKR 이 밝혀졌을 때 발생한다. 이것이 VMKR 이 밝혀질 때 개시된 검증 과정 동안에 결정되는 바와 같이 성공적으로 계산된다면, 조건부로 매체 키이(Km)를 계산할 필요는 없다. 검증이 성공적이지 못하다면 CCMKR 이 밝혀졌을 때 매체 키이(Km)가 계산될 것이며 재검증될 것이다. 이것은 매체 키이를 계산할 수 있는 2 번의 기회가 있다는 것을 의미한다. 마지막으로, 단계(205.48)에서, 다음에 단계(205.75)에서 평가된 바와 같이 레코드가 EMKR 이라면, 버퍼 포인터가 단계(205.49)에서 이러한 유형의 레코드에 대하여 적절한 오프셋에 설정된 이후에, 시스템은 레코드의 끝에서 타당한 매체 키이(Km)가 존재하는 것을 검증할 것이며, 단계(205.75)에서 마지막 매체 키이(Km)가 발생될 것이다. 그러나, 만약 타당한 매체 키이가 단계(205.70)에서 반환되지 않는다면, 단계(205.85)에 의해서 도시된 바와 같이 처리는 도 9 의 단계(210)에서 계속될 것이다. 따라서 MKB 처리는 완료되다.

SDSE(175)내의 기능은 장치 구동부(19)에 대한 보안 장치 구동부 인테페이스(190A)을 사용함으로써 Get MKB 와 같은 보안 접근을 수행한다. 마찬가지의 장치 드라이버인 SDDI(190A)도 또한 그것이 직접적으로 호출할 수 있는 SDSE(175)내의 기능을 이용할 수 있다. 예를 들면, 인가 부위에 대한 읽기를 지시 하기 전에, SDDI(190A)는 처음에 SDSE(175)내의 sec_ake 기능을 호출하여야 한다. sec_ake 기능은 다시 SDDI(190a)로 호출된다. 이러한 SDSE(175) 내의 장치 키이의 고립을 용이하게 하는 '이중 호출 관계'는 SD-오디오 표준의 SW 100 의 수행에 대하여 유일하다.

SDSE(175)는 모든 키이-배향의 처리를 취급하고, 이러한 값들은 특정의 SD 명령들이 오디오 인터페이스(105), 비디오 인터페이스(110) 또는 이미지 인터페이스(115)에 의해 수신될 때 필요하기 때문에, 장치 구동부는 그것이 직접적으로 호출할 수 있는 SDSE(175)내의 기능을 이용하여야 한다. 기능을 수행할 때, SDSE 모듈(175)는 다시 장치 구

동부(190)의 비밀 보안 인터페이스인 SDDI(190A)으로 호출되어야 한다. 이러한 '이중 호출 관계'는 SDSE(175)와 장치 구동부(190) 사이에서 서로 짜여진 요청을 가능하게 하여, 키이 값이 보안 모듈내에 고립될 수 있게 한다.

SDSE(175) 소프트웨어 층은 보안 구동부 요청 패킷을 초기화하고 요청 패킷 포인터를 통과시키는 보안 구동부 인터페이스 가입 지점을 호출함으로써 비밀 인터페이스를 통하여 보안 장치 구동부 서비스를 실시한다.

참조로서 포함된 첨부의 소스 코드를 명확하게 하기 위하여, 다음의 표가 제공된다.

요청 패킷(sdapi.h 로 정의됨)은 다음과 같이 한정된 데이터 유형 SSMSERVE 로 구성되어 있다.

[표 2]

변수	변수 명칭
Typedef struct_mySecuredDrv	
{	
데이터 버퍼	UCHAR *buffer
데이터 블록의 수	UNIT16 noBlocks
적용 유일 번호	UNIT16 mkb_ID
시작 어드레스	UNIT16 lba
보안 플래그	INT16 securityFlag
드라이브 번호	INT16 driveNo
명령 인덱스	INT16 opCode
}	

명령 인덱스(INT16 opCode) 는 요청되는 서비스에 대한 명령을 유지한다. 지원된 명령은 다음을 포함한다:

[표 3]

명령	기능 코드 루틴
장치 식별	SDDRV_IDENT 0 # 한정
보안 식별	SDDRV_SECIDENT 1 # 한정
보안 읽기	SDDRV_SECRD 2 # 한정
보안 쓰기	SDDRV_SECWR 3 # 한정
보안 소거	SDDRV_SECERASE 4 # 한정
MKB 읽음	SDDRV_RDMKB 5 # 한정
MID 를 가져옴	SDDRV_GETMID 6 # 한정
챌린지(challange)를 설정	SDDRV_SETCHALGE 7 # 한정
챌린지를 가져옴	SDDRV_GETCHALGE 8 # 한정
응답을 설정	SDDRV_SETRESP 9 # 한정
응답을 가져옴	SDDRV_GETRESP 10 # 한정
보호된 부위의 크기 변화	SDDRV_CHANGESA 11 # 한정

보안 장치 구동부 서비스의 요청은 SDSE(175) 모듈로부터 발행된다. 예를 들면, 챌린지 발생(1) 기능은 챌린지(1)를 다음과 같이 보낸다.

[표 4]

명령	작동
보안 루틴을 호출하라	SDSECURITYDRV mySecDrv
드라이브 번호를 설정하라	mySecDrv.driveNo = (INT16)drv
메모리 어드레스를 매체내에 설정하라	mySecDrv.lba = 0
데이타 블록의 번호	mySecDrv.noBlocks = 1
챌린지를 설정하라	mySecDrv.opCode = SDDRV_SETCHALGE
챌린지(1)를 보내라	mySecDrv.buffer = Chlg 1
장치 구동부로 호출하라	scDDHandler(amp;mySecDrv)

모든 키이 조작은 SDSE(175)에 제한되어 있기 때문에, SSDI(190A)는 인증 키이 교환(AKE)을 수행하도록 또는 버스를 가로질러 전달되었던 데이타를 암호 해제시키도록 SDSE(175)에 의존하여야만 한다 (버스를 지나서 보내진 모든 데이타는 각 AKE 로부터 발생된 '세션 키이'를 사용하여 처음에 암호화된다는 점을 주목하라).

AKE 를 수행할때, SDSE(175)는 명령을 SD 메모리 카드(13)로 보내야만 하며, 따라서 이것은 다시 SDDI(190A)으로 호출되어야 한다. 이러한 호출 관계는 도 7 의 다이아그램에 개략되어 있으며 이것은 인증 부위의 읽기를 처리하는데 필요한 단계를 나타낸다.

보안 SDDI(190A)에 의해서 호출되었을때, SDSE(175)내의 sec_ake 함수는 비밀의 구동부 인터페이스를 통하여 보안 장치의 구동부로 4 개의 호출을 다시 수행한다. 이러한 4 개의 요청은 SDDRV_SETCHALGE, SDDRV_GETCHALGE, SDDRV_SETRESP 및, SDDRV_GETRESP 로 이루어진다. 이것은 보안 모듈이 도 7 에 도시된 바와 같은, 필요한 챌린지 설정/챌린지 가져오기, 응답 설정/응답 가져오기를 수행할 수 있게 한다. 결과적인 세션 키이는 보안 모듈내에 저장된다. 이것은 보안 장치 드라이버가 SDDI(190A)로부터 정보를 가져오도록 SDSE(175)의 bus_decrypt 기능으로 호출할때 데이타를 암호 해제시키도록 사용된다.

난수 발생

도6 및 7에 도시된 AKE 프로세스(83)는 난수의 발생을 필요로한다. SD 난수 발생(RNG)의 구현은 선행 난수(Vt)에 기초한 난수 시드(Seed) V(t+1)의 계산을 필요로 한다. 난수 시드 V(t+1)은 플래시 메모리에 저장될 필요가 있으며 차기 난수의 발생을 위해 시드로서 사용될 것이다. 이런 난수 및 시드는 C2_G, C2 암호 단방향 함수(C2 Cipher one-way function)로 사용된다.

난수 시드 Vt+1을 재기록하는 것이 가능하지 않은(즉, 시드 저장을 위해 지시된 EEPROM이 없음) 저비용 계산 플랫폼들에서, RNG 구현은 문제를 발생시킬 수 있다.

프로그램이 난수 발생기의 시드로서 예측가능한 수 또는 고정된 수를 사용하는 경우, 해커가 복제 방지를 깨거나 또는 키들을 훔칠 수 있다. 고정 시드를 가지고, 제1난수 및 도전(challenge)은 또한 동일한 SD 카드 주소가 접근될 경우 고정된다. 응답을 트래킹함으로써, 고유한 매체 키가 도날될 가능성이 있다. 여기에서, 우리는 PC 응용을 위한 솔루션으로서 시스템 타이머를 사용한다. 타이머 발생된 시드는 예측가능하지 않음을 보장하기 위하여, 난수 시드의 모든 바이트가 랜덤하도록 타이머를 셔플링한다(shuffle). 마이크로칩 PIC16xxx 같은, 시스템들 상에서, 실행되는 명령어들의 수인, 타이머1은 셔플링 이후에 난수 발생의 시드로서 사용될 수 있다.

주요 아이디어는 항상 증가하는 하나 이상의 타이머를 사용하고 이들이 고주파수로 업데이트되도록 최저속(slowest) 업데이팅(최상위) 바이트들의 주파수를 조 종하는 것이다. 최저속 업데이팅 (최상위) 바이트들의 값은 최고속 업데이팅 (최하위)바이트들의 함수로서 게산될 것이다. 난수 발생의 한가지 방법은 시스템 타이머를 사용하는 것이다. 그것은 4바이트의 프로세서 클럭 및 4바이트의 협정 세계시(Universal Coordinated Time)를 사용한다. 프로세서 클럭은

프로그램이 시작한 이래의 프로세서 시간의 수이다. 이 시간은 이런 적용에서 60인 CLOCKS_PER_SECONDS로 정련된다(refined). 그것은 마이크로소프트 C++ 실행시간 함수 'clock()'를 통해 얻어진다. 협정 세계시는 마이크로소프트 C++ 실행시간 함수 'time()'를 통해 얻어진다. 그것은 1899년 12월 31일 정오부터의 초의 수이다.

프로세서 클럭들은 0-3 바이트로 판독되고, 협정 세계시는 4-7 바이트들로 판독된다. 바이트 0 및 1이 빠르게 변하지만, 바이트 2 및 3은 덜 빠르게 변한다. 바이트 0 및 1이 변하는 페이스에 기초한 페이스로 변하도록 하여 바이트 2 및3이 바이트 2 및 3의 변화율을 증가시킬 수 있다. 이것은 다수의 방식으로 달성될 수 있다. 예를 들면, 배타적 OR함수를 가지고:

vt_1[2]=vt_1[2] ^ vt_1[0]

vt_1[3]=vt_1[3] ^ vt_1[1] ^ vt_1[0] 이다.

유사하게는, 바이트 6 및 7이 이하 실시예에 의해 보다 빠른 페이스로 변화되도록 될 수 있다:

Vt_1[6]=vt_1[0]+vt_1[1]+vt_1[2]+vt_1[3]+vt_1[4]+vt_1[5]+vt_1[6]

Vt_1[7]=vt_1[0] ^ vt_1[1] ^ vt_1[2] ^ vt_1[3] ^ vt_1[4] ^ vt_1[5] ^ vt_1[6] ^ vt_1[7]

변화의 페이스를 증가시키는 것은 또한 특정 바이트 위치들에서 타이머 값들 의 곱을 사용하여 또는 보다 빠른 이동 바이트들에 의존한 느린 이동 바이트들을 만드는 다른 방법들에 의해 달성될 수 있다. 보다 상세하게는, 이 명세서의 일부를 형성하는 첨부된 소스 코드를 참고하시오.

결과적으로, 고정 시드에 C2 암호를 적용하는 것이 새로운 난수 시드 V(t+1)를 발생시킬 것이다. C2_G는 C2 암호 단방향 함수이다. 그것은 제1 AKE에 대해 다른 초기 시드 'v1'를 발생시키기 위해 난수 키 'c1'를 고정 시드'v0'에 인가한다.

CurrentTime은 타이머 틱(tick)들을 판독하기 위해 DOS 소프트웨어 인터럽트 '1A' 함수 '0'을 사용하는 어셈플리 프로그램이다. 이하는 _CurrentTime PROC NEAR의 코드이다: push cx; mov ax, 0; int 1ah; mov dx,cx; pop cx; ret; _CurrentTime ENDP.

이득

본 발명의 시스템 및 방법은 다수의 방식들로 종래기술들 보다 이득적이다. 장치 키들 및 관련 세션 키들은 매우 격리되고 보호된 소프트웨어층에서 조종된다. 이들은 상부층들에서 결코 노출되지 않는다.

하부 디바이스 드라이버층도 키들에 직접 접근되지 않는다. 디바이스 키들은 매체 키를 발생시킬 때 OEM-제공된 라이브러리로부터 검색된다. 이런 키는 보안 엔진내에 유지되지만, 보안 기법의 핵심인 매체 고유 키(K_{mu})는 결코 저장되지 않는다. 보안 엔진에 대한 개인 인터페이스는 보안 엔진이 메모리 카드에 대한 하위레벨 접근을 얻도록 하면서 모든 보안관련 키들(예를들어, 디바이스 키, 매체 키, 세 션키)의 노출이 보안 엔지내에 한정되도록 한다. '이중 호출 관계(dual calling relationship)'는 보안 엔진 및 보안 디바이스 드라이버가 상호 서비스들의 섞이도록(interwoven) 한다.

본 발명의 특정 실시예들 및 이들의 장점들이 도시되고 기술되었지만, 다양한 변화들, 대체, 및 변형들이 첨부된 청구항으로 한정되는 본 발명의 사상 및 범위를 벗어나지 않고 본문에서 이루어질 수 있다. 예를 들면, SD 메모리 카드의 사용이 본 발명의 기능을 예시하도록 도시되었지만, 본 발명은 암호화 컨텐츠를 가지는 어떤 매체상에서도 사용될 수 있다. 임의 타입의 디바이스에 의해 또한 사용될 수도 있다.

(57) 청구의 범위

청구항 1.

디바이스로 제거가능한 매체상의, 컨텐츠를 가지는 프레임들을 포함하는 암호화 트랙에 접근하는 방법에 있어서,

상기 매체를 허가(authorizing)하는 단계; 및

(a)매체 고유키를 계산하는 단계;

(b)상기 매체 고유키로 상기 디바이스의 메모리에 저장된 타이틀 키를 해독하는 단계;

(c)프레임 그룹을 해독하는 단계;

(d)상기 해독된 타이틀 키를 삭제하는 단계;

(e)상기 매체 고유키를 삭제하는 단계; 및

(f)전체 트랙이 완료될 때까지 (a) 내지 (e)를 반복하는 단계;를 포함하는 프로세스에 의해 상기 트랙을 해독하는 단계;

를 포함하는 것을 특징으로 하는 방법.

청구항 2.
제1항에 있어서, 상기 매체를 허가하는 단계는,

매체키를 계산하는 단계;

상기 매체키로부터 매체 고유키를 계산하는 단계

상기 매체키를 삭제하는 단계;

상기 매체 고유키로부터 세션키를 계산하는 단계; 및

상기 매체고유키를 삭제하는 단계;

를 포함하는 것을 특징으로 하는 방법.

청구항 3.
제1항에 있어서,

계산된 세션키로 상기 매체에 저장된 이중 암호화된 타이틀 키를 해독하면서 단일 암호화된 타이틀 키를 생성하도록 상기 매체를 허가하는 단계; 및

상기 매체로부터의 상기 단일 암호화된 타이틀 키를 상기 디바이스의 메모리에 복사하는 단계;

를 더 포함하는 것을 특징으로 하는 방법.

청구항 4.
제2항에 있어서, 상기 매체 키를 계산하는 단계는,

(a)버퍼로부터 매체키 블럭의 제1레코드을 판독하는 단계;

(b)상기 제1레코드의 길이 및 유형에 기초하여 상기 버퍼 오프셋을 갱신하는 단계;

(c)상기 갱신된 버퍼 오프셋의 상기 매체 키 블럭의 다른 레코드를 판독하는 단계; 및

(d)상기 매체 키 블럭의 모든 필요한 레코드들 및 상기 매체 키가 계산될 때까지 (a)-(c)를 반복하는 단계;

를 포함하는 것을 특징으로 하는 방법.

청구항 5.
제1항에 있어서, 상기 프레임 그룹들은 디코딩되거나 또는 압축해제된 형태로 일이하 내지 약 5초의 컨텐츠를 포함하는 것을 특징으로 하는 방법.

청구항 6.

제1항에 있어서, 상기 트랙을 해독하는 단계는 하나 이상의 파일들을 해독하는 단계를 포함하되, 상기 파일들은 상기 프레임들을 포함하는 것을 특징으로 하는 방법.

청구항 7.
제1항에 있어서, 상기 트랙을 디코딩 및 압축해제하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 8.
디바이스로 제거가능한 매체상의, 컨텐츠를 가지는 프레임들을 포함하는 암호화 데이터 파일에 접근하는 방법에 있어서,

매체키를 계산하는 단계;

상기 매체키로부터 매체 고유키를 계산하는 단계;

상기 매체키를 삭제하는 단계;

상기 매체 고유키로부터 세션키를 계산하는 단계; 및

상기 매체고유키를 삭제하는 단계;

를 포함하는 처리에 의해 사용자 세션을 위한 매체를 허가하는 단계;

단일 암호화 타이틀 키를 발생시키기 위해 상기 세션 키로 상기 매체에 저장된 이중 암호화 타이틀 키를 해독하는 단계;

상기 매체로부터 상기 단일 암호화 타이틀 키를 상기 디바이스의 메모리에 복사하는 단계; 및

(a)상기 매체고유키를 계산하는 단계;

(b)상기 매체 고유키로 상기 디바이스의 메모리에 저장된 상기 타이틀 키를 해독하는 단계;

(c)프레임 그룹을 해독하는 단계;

(d)상기 해독된 타이틀 키를 삭제하는 단계;

(e)상기 매체 고유키를 삭제하는 단계; 및

(f)전체 파일이 완료될 때까지 (a) 내지 (e)를 반복하는 단계;

를 포함하는 프로세스에 의해 파일을 해독하는 단계;

를 포함하는 것을 특징으로 하는 방법.

청구항 9.
제8항에 있어서, 상기 매체키를 계산하는 단계는,

매체 키 블럭을 암호화 데이터의 바이트들을 포함하는 청크들로 분할하는 단계; 및

상기 블럭의 특정 청크내 오프셋에서의 판독을 위해 상기 버퍼를 설정함으로써 상기 매체 키 블럭내에서 키를 암호화하는 단계;

를 포함하는 것을 특징으로 하는 방법.

청구항 10.
제9항에 있어서,

상기 키를 해독하는 단계는,

(a)제1레코드로부터 매체 키를 계산하는 단계;

(b)상기 버퍼 오프셋을 갱신하는 단계;

(c)상기 갱신된 버퍼 오프셋의 제2레코드를 판독하는 단계; 및

(d)상기 계산된 매체 키를 기준 매체 키와 비교하여 제2레코드로 상기 매체키를 검증하는 단계;

를 포함하는 것을 특징으로 하는 방법.

청구항 11.
제10항에 있어서, 상기 버퍼 오프셋은 상기 매체 키 블럭의 제1 레코드의 타입 및 길이에 의해 결정되는 것을 특징으로 하는 방법.

청구항 12.
제8항에 있어서, 프레임 그룹은 일초 이하 내지 약 5초의 압축해제 및 디코딩된 오디오 컨텐츠를 포함하는 것을 특징으로 하는 방법.

청구항 13.
디바이스가 매체로부터 암호화 컨텐츠를 가지는 암호화 파일을 판독하도록 하고 암호화 컨텐츠를 가지는 암호화 파일을 매체에 기록하도록 하는 시스템에 있어서,

컴퓨팅 유닛, 및 시스템 메모리;

상기 디바이스로부터 명령어들을 수신하기 위한 인터페이스 수단;

(a)상기 매체로부터 상기 디바이스의 메모리에 암호화 타이틀키를 복사하고,

(b)상기 암호화 타이틀 키를 해독하며,

(c)상기 해독된 타이틀 키로 암호화 컨텐츠의 일부를 해독하고,

(d)상기 해독된 타이틀 키를 삭제하며,

(e)모든 상기 파일의 컨텐츠가 해독될 때까지, (a)-(d)를 반복하도록: 구성된 보안 다이나믹 해독 수단;을 포함하며,

상기 해독된 타이틀 키들은 시스템의 상기 보안 수단에 위치하고 상기 보안수단에 접근만 가능한 것을 특징으로 하는 시스템.

청구항 14.
제13항에 있어서, 상기 타이틀 키는 재생(played back)될 때 압축해제 및 디코딩된 상태로 5초 또는 그 이하의 컨텐츠를 해독하는 데 걸린 시간 동안 해독 상태인 것을 특징으로 하는 시스템.

청구항 15.
제13항에 있어서, 디지털 신호 프로세서를 더 포함하는 것을 특징으로 하는 시스템.

청구항 16.
제13항에 있어서, 인터페이스 수단 및 보안 다이나믹 암호화 수단은 상기 디바이스의 시스템 메모리에 저장되는 것을 특징으로 하는 시스템.

청구항 17.
제16항에 있어서, 상기 인터페이스 수단 및 보안 다이나믹 해독 수단은 컴퓨팅 유닛에 의해 실행되는 것을 특징으로 하는 시스템.

청구항 18.
제15항에 있어서, 상기 보안 다이나믹 해독 수단은 상기 디지털 신호 프로세서의 메모리에 저장되고, 상기 디지털 신호 프로세서에 의해 실행되는 것을 특징으로 하는 시스템.

청구항 19.
제18항에 있어서, 상기 인터페이스 수단은 상기 디지털 신호 프로세서에 의해 실행되는 것을 특징으로 하는 시스템.

청구항 20.
보안 매체상에 암호화 컨텐츠를 가지는 파일을 디바이스가 해독하도록 하는 시스템에 있어서,

상기 디바이스로부터 명령어들을 수신하기 위한 하나 이상의 사용자 인터페이스 모듈;

상기 하나 이상의 사용자 인터페이스 모듈로부터 명령어들을 수신하고 상기 보안 매체로부터 암호화 컨텐츠의 검색 및 저장을 관리하는 응용 프로그래밍 인터페이스;

상기 보안 매체로부터 디바이스의 메모리로 전송된 하나 이상의 암호화 키들 및 암호화 컨텐츠를 해독하기 위한 보안 엔진;을 포함하며,

상기 해독된 키들은 상기 암호화 컨텐츠를 해독하기 위해 사용되고,

상기 하나 이상의 키들은 암호화 데이터 세그먼트에 포함되며,

보안 엔진은 (a) 하나 이상의 상기 키들을 해독하고, (b)상기 하나 이상의 해독 키들을 사용하여 상기 암호화 컨텐츠의 일부를 해독하고, (c)상기 하나 이상의 해독된 키들을 삭제하며, (d)상기 컨텐츠의 모든 부분들이 해독될 때가지 (a)-(c)를 반복하는 것을 특징으로 하는 시스템.

청구항 21.
제20항에 있어서, 상기 컨텐츠는 AAC, MP3 또는 WMA 포맷으로 인코딩되는 것을 특징으로 하는 시스템.

청구항 22.
제21항에 있어서, 상기 하나 이상의 키들은, 일초 이하내지 약5초의 디코딩된 컨텐츠를 해독 및 처리하기 위한 시간 동안 해독 상태인 것을 특징으로 하는 시스템.

청구항 23.
제20항에 있어서, 상기 데이터 세그먼트는 단편부분(fractional portions)들로 버퍼링되고 해독되는 하나 이상의 암호화 키들을 포함하는 것을 특징으로 하는 시스템.

청구항 24.
제23항에 있어서, 상기 단편부분은 약 512바이트인 것을 특징으로 하는 시스템.

청구항 25.
제20항에 있어서, 상기 디바이스는 컴퓨팅 유닛, 시스템 메모리 및 하드웨어 인터페이스를 포함하는 것을 특징으로 하는 시스템.

청구항 26.
제20항에 있어서, 상기 디바이스는 디지털 신호 프로세서를 더 포함하는 것을 특징으로 하는 시스템.

청구항 27.
제25항에 있어서, 상기 디바이스의 시스템 메모리에 저장되는 것을 특징으로 하는 시스템.

청구항 28.
제27항에 있어서, 상기 시스템 메모리에 저장된 상기 소프트웨어 시스템은 상기 컴퓨팅 유닛에 의해 실행되는 것을 특징으로 하는 시스템.

청구항 29.
제26항에 있어서, 상기 디지털 신호 프로세서의 RAM에 저장되는 것을 특징으로 하는 시스템.

청구항 30.
제26항에 있어서, 상기 시스템의 일부는 상기 디바이스의 시스템 메모리에 저장되고 시스템의 일부는 상기 디지털 신호 프로세서의 RAM에 저장되는 것을 특징으로 하는 시스템.

청구항 31.
제30항에 있어서, 상기 디지털 신호 프로세서의 RAM에 저장된 상기 시스템의 일부는 상기 디지털 신호 프로세서에 의해 실행되는 것을 특징으로 하는 시스템.

청구항 32.
제30항에 있어서, 상기 디지털 신호 프로세서의 RAM에 저장된 시스템의 일부는 상기 보안 엔진을 포함하는 것을 특징으로 하는 시스템.

청구항 33.
제20항에 있어서, 오디오, 비디오 및 이미지들을 처리 및 전송하기 위한 하나 이상의 엔진들을 더 포함하며, 각각의 엔진은 보안 응용 프로그래밍 인터페이스, 매체의 상기 암호화 컨텐츠 및 키들에 접근하기 위한 보안 인터페이스(들), 및 상기 매체의 비암호화 컨텐츠에 접근하기 위한 비보안 인터페이스(들)을 포함하는 것을 특징으로 하는 시스템.

청구항 34.
제33항에 있어서, 보안 관리자 모듈을 더 포함하는 것을 특징으로 하는 시스템.

청구항 35.
제34항에 있어서, 상기 보안 인터페이스(들)은 상기 보안 관리자 모듈과 통신하고 모듈은 상기 보안 엔진과 통신하는 것을 특징으로 하는 시스템.

청구항 36.
제33항에 있어서, 디바이스 드라이버를 더 포함하고, 상기 보안 엔진은 상기 디바이스 드라이버를 통해 상기 컨텐츠 및 키들에 접근하는 것을 특징으로 하는 시스템.

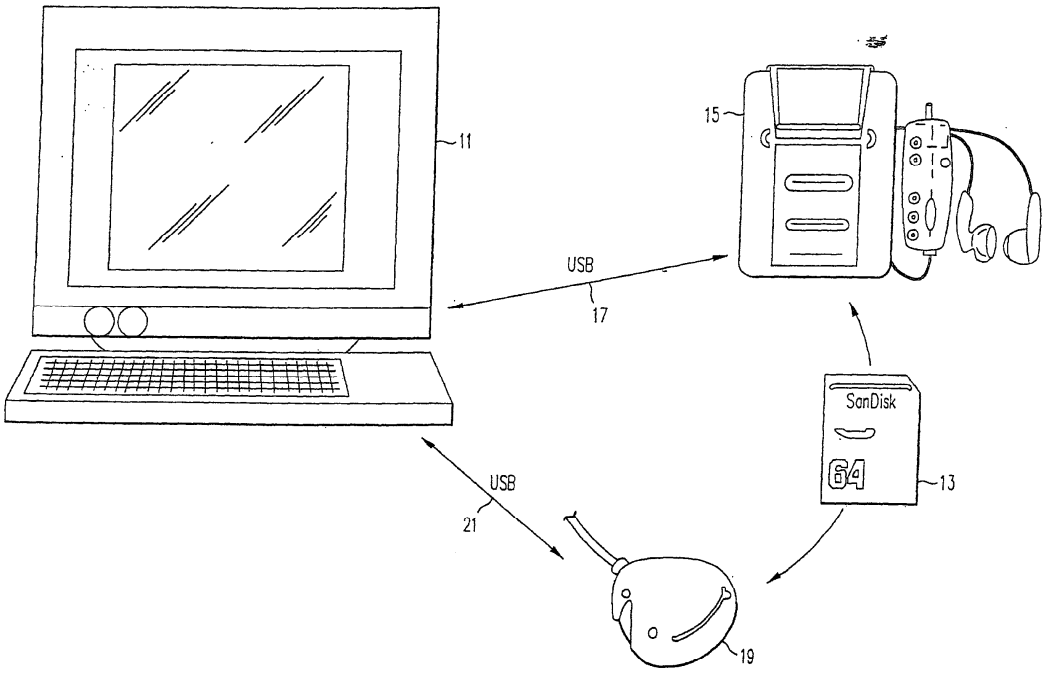
청구항 37.
제33항에 있어서, 오디오, 비디오 또는 이미지들을 처리 및 전송하기 위한 각각의 상기 하나 이상의 엔진들은 상기 매체의 비암호화 컨텐츠에 접근하기 위한 비보안 응용 프로그래밍을 더 포함하는 것을 특징으로 하는 시스템.

청구항 38.
제20항에 있어서, 상기 보안 엔진은 난수 발생기를 더 포함하며, 상기 발생기는 상기 난수를 발생시키기 위해 두 개 이상의 시스템 타이머를 사용하는 것을 특징으로 하는 시스템.

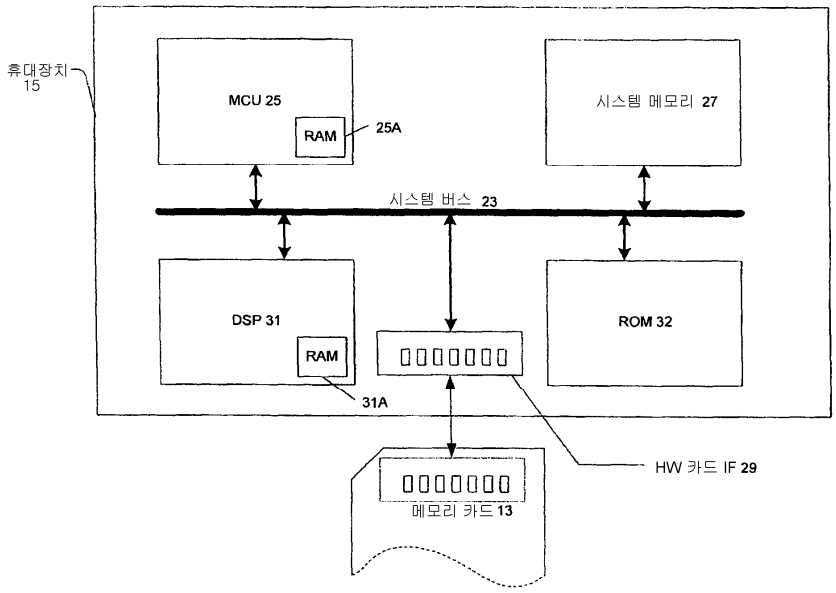
청구항 39.
제38항에 있어서, 상기 발생기는 상기 난수를 발생시키기 위해 사용된 타이머 틱들의 고유 주파수 갱신을 증가시키는 것을 특징으로 하는 시스템.

도면

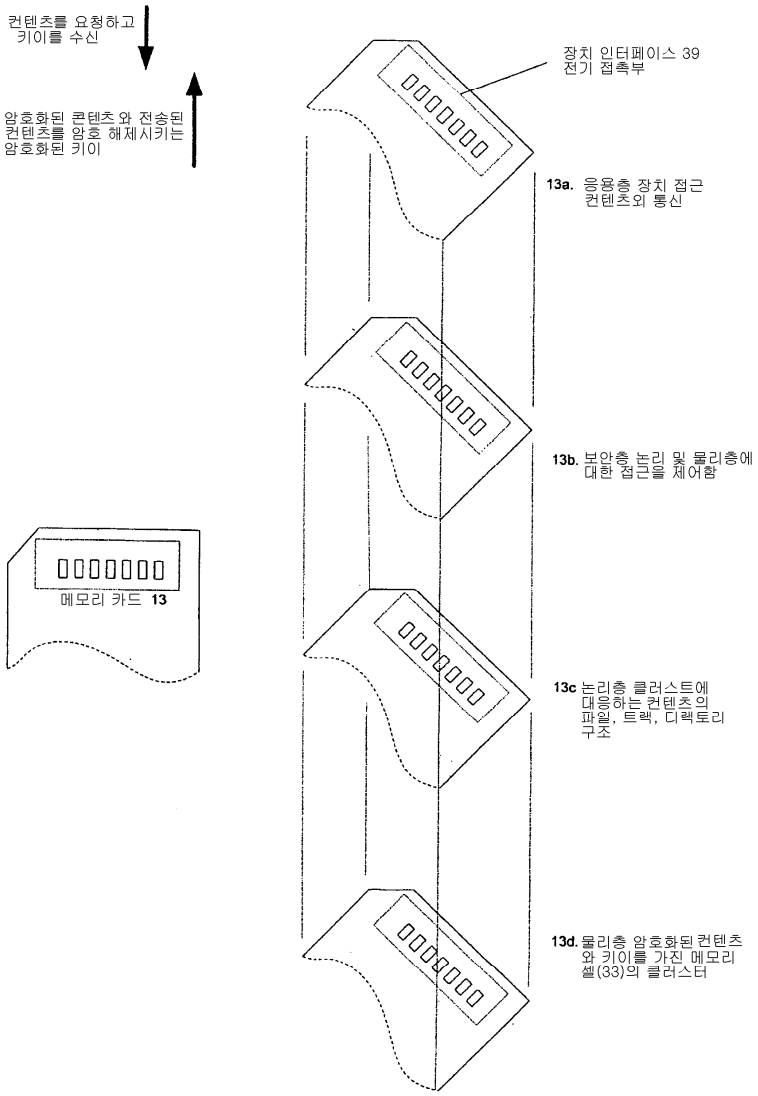
도면1


11
15
USB
17
SanDisk
64
13
USB
21
19

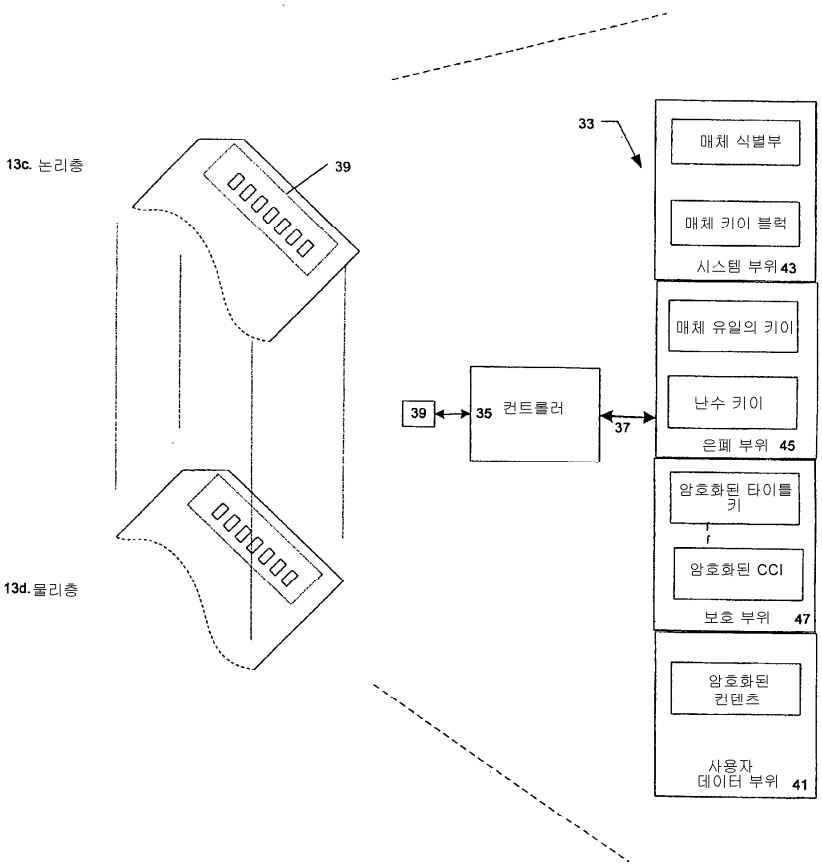
도면2


휴대장치
15
MCU 25
RAM
25A
시스템 메모리 27
시스템 버스 23
DSP 31
RAM
31A
ROM 32
HW 카드 IF 29
메모리 카드 13

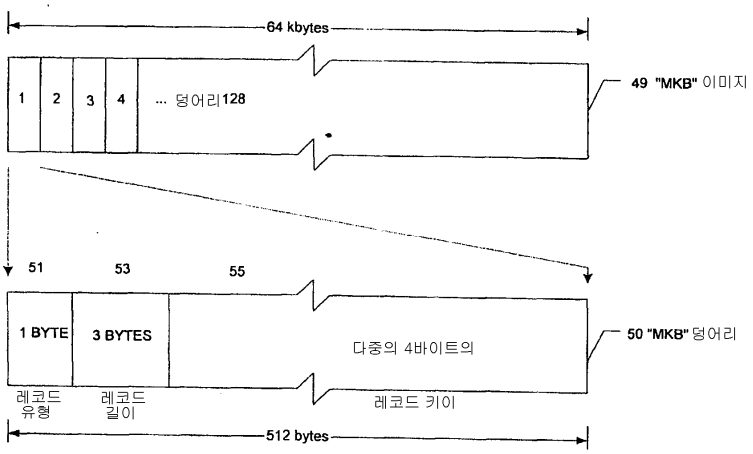
도면3a


컨텐츠를 요청하고
키이를 수신
암호화된 콘텐츠와 전송된
컨텐츠를 암호 해제시키는
암호화된 키이
메모리 카드 13
장치 인터페이스 39
전기 접촉부
13a. 응용층 장치 접근
컨텐츠외 통신
13b. 보안층 논리 및 물리층에
대한 접근을 제어함
13c 논리층 클러스트에
대응하는 컨텐츠의
파일, 트랙, 디렉토리
구조
13d. 물리층 암호화된 컨텐츠
와 키이를 가진 메모리
셀(33)의 클러스터

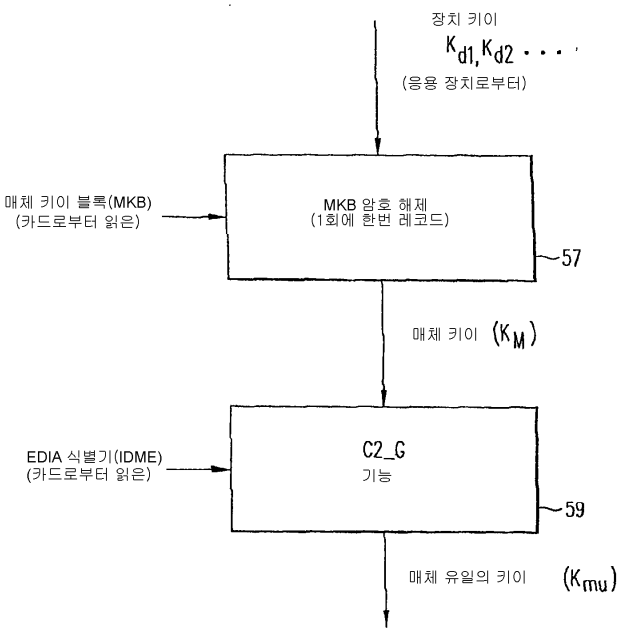
도면3b


13c. 논리층
39
13d. 물리층
33
매체 식별부
매체 키이 블럭
시스템 부위 43
매체 유일의 키이
난수 키이
은폐 부위 45
39
35 컨트롤러
37
암호화된 타이틀 키
암호화된 CCI
보호 부위 47
암호화된 컨텐츠
사용자 데이터 부위 41

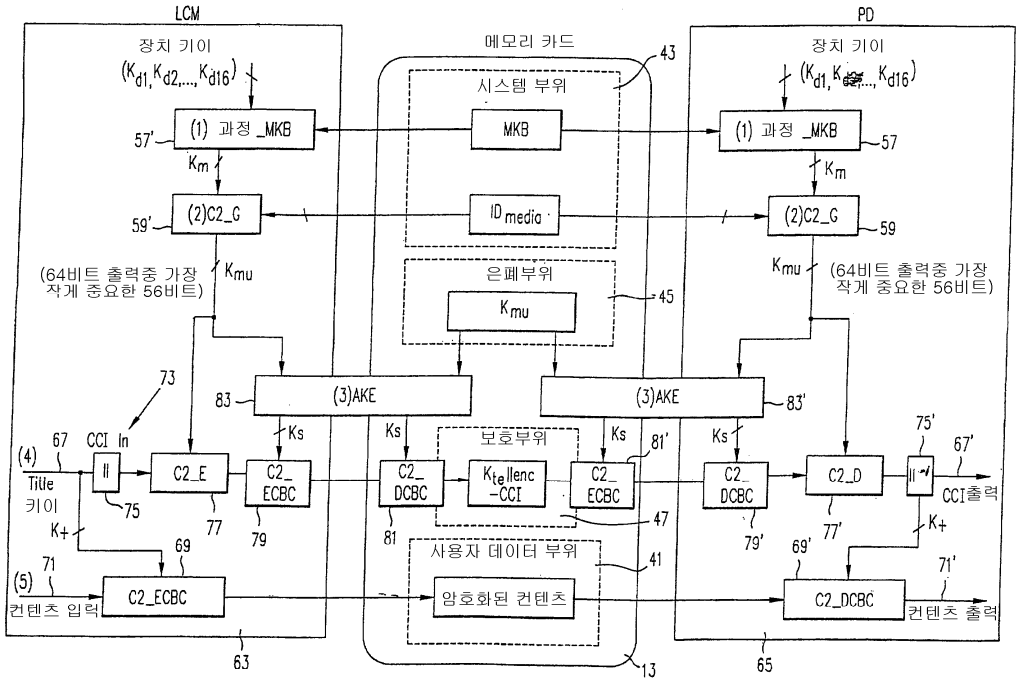
도면4


64 kbytes
1
2
3
4
... 덩어리128
49 "MKB" 이미지
51
53
55
1 BYTE
3 BYTES
다중의 4바이트의
50 "MKB" 덩어리
레코드 유형
레코드 길이
레코드 키이
512 bytes

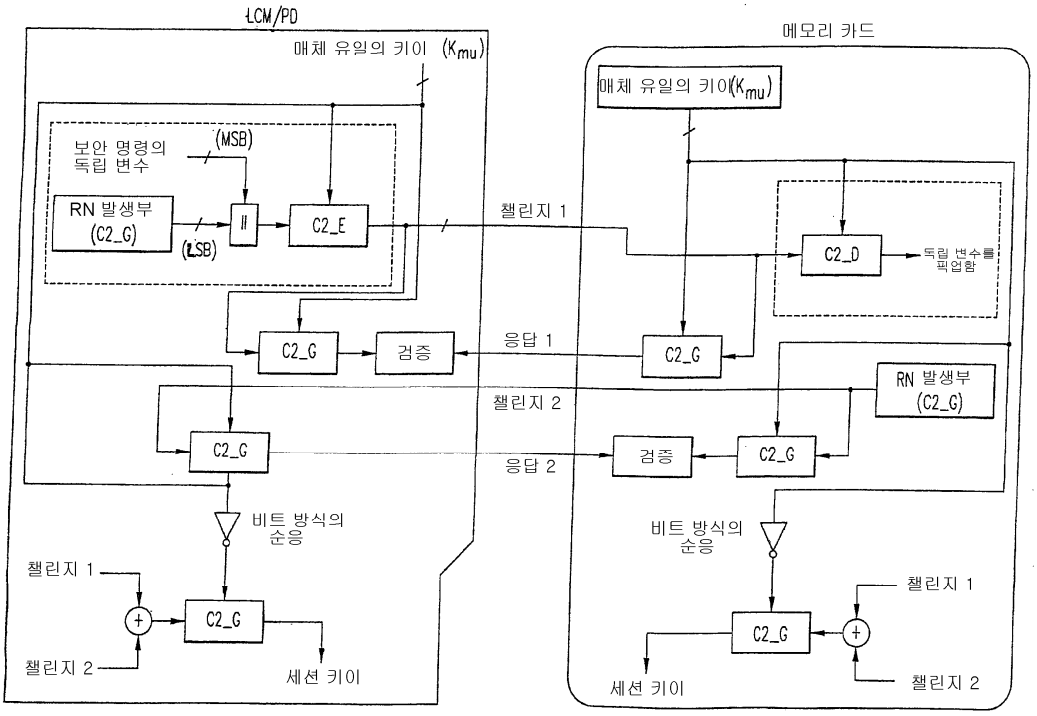
도면5


장치 키이
$K_{d1}, K_{d2} \cdots$
(응용 장치로부터)
매체 키이 블록(MKB)
(카드로부터 읽은)
MKB 암호 해제
(1회에 한번 레코드)
57
매체 키이 (K_M)
EDIA 식별기(IDME)
(카드로부터 읽은)
C2_G
기능
59
매체 유일의 키이 (K_{mu})

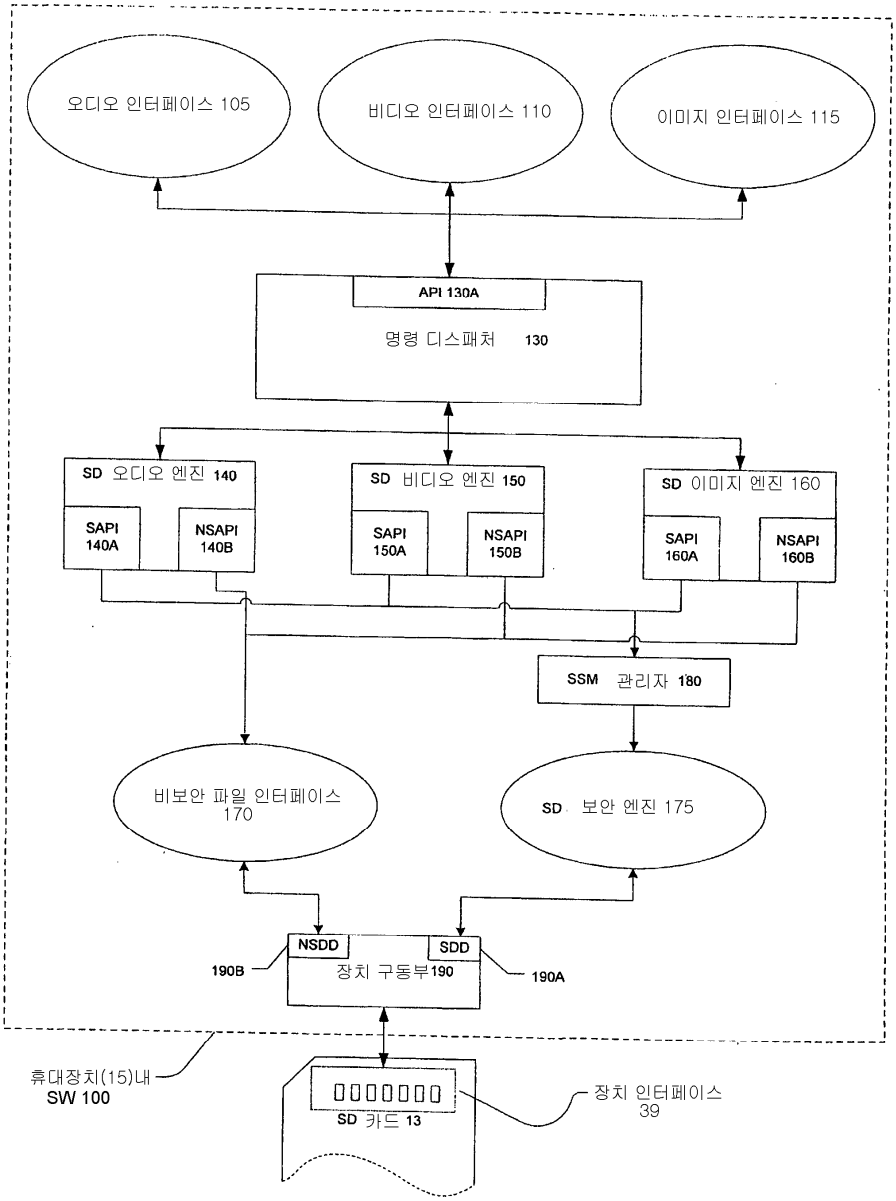
도면6


LCM
PD
메모리 카드
장치 키이
$(K_{d1}, K_{d2}, \ldots, K_{d16})$
시스템 부위
43
(1) 과정 _MKB
MKB
57'
57
K_m
(2)C2_G
ID_{media}
59'
59
(64비트 출력중 가장 작게 중요한 56비트)
K_{mu}
은폐부위
K_{mu}
45
73
(3)AKE
83
83'
Ks
보호부위
81'
(4)
Title 키이
67
CCI In
C2_E
C2_ ECBC
C2_ DCBC
$K_{tellenc}$ -CCI
C2_D
75'
67'
CCI출력
75
77
79
81
47
K_+
69
71
(5)
C2_ECBC
사용자 데이터 부위
41
암호화된 컨텐츠
C2_DCBC
컨텐츠 입력
컨텐츠 출력
63
13
65

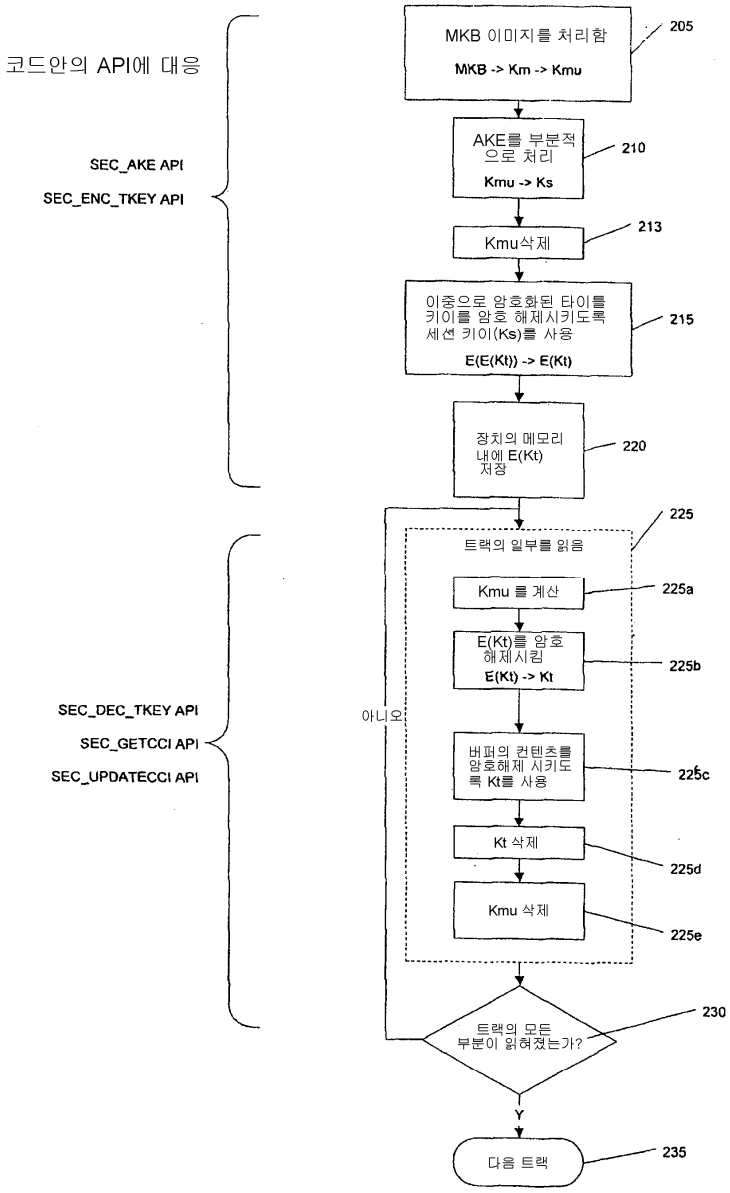
도면7


LCM/PD
매체 유일의 키이 (Kmu)
보안 명령의 독립 변수
(MSB)
RN 발생부 (C2_G)
(LSB)
II
C2_E
챌린지 1
C2_G
검증
응답 1
챌린지 2
C2_G
응답 2
비트 방식의 순응
챌린지 1
C2_G
챌린지 2
세션 키이
메모리 카드
매체 유일의 키이(Kmu)
C2_D
독립 변수를 픽업함
C2_G
RN 발생부 (C2_G)
검증
C2_G
비트 방식의 순응
챌린지 1
C2_G
세션 키이
챌린지 2

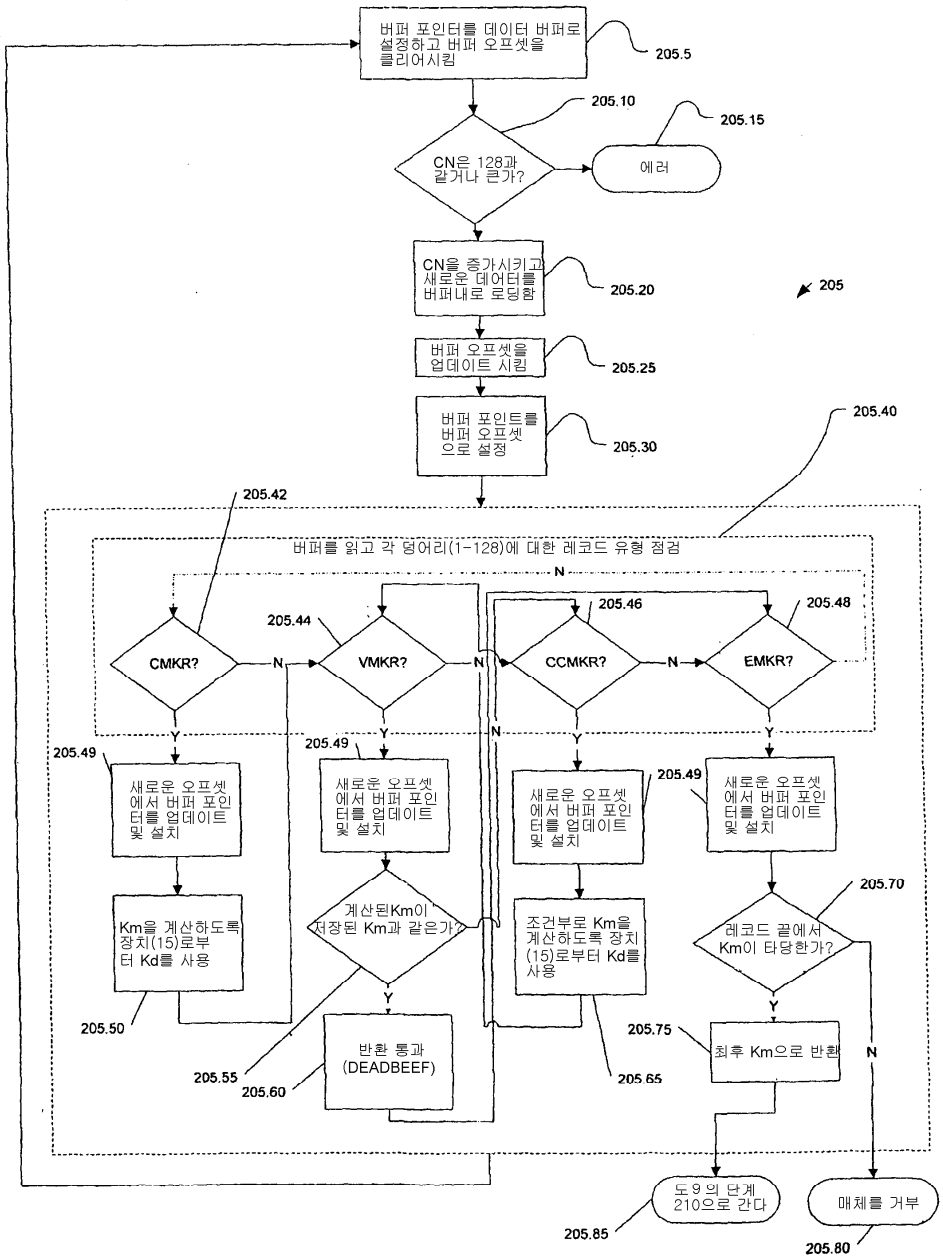
도면8


오디오 인터페이스 105
비디오 인터페이스 110
이미지 인터페이스 115
API 130A
명령 디스패처 130
SD 오디오 엔진 140
SAPI 140A
NSAPI 140B
SD 비디오 엔진 150
SAPI 150A
NSAPI 150B
SD 이미지 엔진 160
SAPI 160A
NSAPI 160B
SSM 관리자 180
비보안 파일 인터페이스 170
SD 보안 엔진 175
NSDD
SDD
190B
장치 구동부190
190A
휴대장치(15)내 SW 100
SD 카드 13
장치 인터페이스 39

도면9


코드안의 API에 대응
SEC_AKE API
SEC_ENC_TKEY API
SEC_DEC_TKEY API
SEC_GETCCI API
SEC_UPDATECCI API
MKB 이미지를 처리함
MKB -> Km -> Kmu
205
AKE를 부분적으로 처리
Kmu -> Ks
210
Kmu삭제
213
이중으로 암호화된 타이틀 키이를 암호 해제시키도록 세션 키이(Ks)를 사용
E(E(Kt)) -> E(Kt)
215
장치의 메모리 내에 E(Kt) 저장
220
225
트랙의 일부를 읽음
Kmu 를 계산
225a
E(Kt)를 암호 해제시킴
E(Kt) -> Kt
225b
아니오
버퍼의 컨텐츠를 암호해제 시키도록 Kt를 사용
225c
Kt 삭제
225d
Kmu 삭제
225e
트랙의 모든 부분이 읽혀졌는가?
230
Y
다음 트랙
235

도면10


버퍼 포인터를 데이터 버퍼로 설정하고 버퍼 오프셋을 클리어시킴
205.5
205.10
205.15
CN은 128과 같거나 큰가?
에러
CN을 증가시키고 새로운 데이터를 버퍼내로 로딩함
205.20
205
버퍼 오프셋을 업데이트 시킴
205.25
버퍼 포인트를 버퍼 오프셋으로 설정
205.30
205.40
205.42
버퍼를 읽고 각 덩어리(1-128)에 대한 레코드 유형 점검
CMKR?
VMKR?
CCMKR?
EMKR?
205.44
205.46
205.48
N
Y
205.49
새로운 오프셋에서 버퍼 포인터를 업데이트 및 설치
Km을 계산하도록 장치(15)로부터 Kd를 사용
205.50
계산된Km이 저장된 Km과 같은가?
205.55
205.60
반환 통과 (DEADBEEF)
조건부로 Km을 계산하도록 장치(15)로부터 Kd를 사용
205.65
레코드 끝에서 Km이 타당한가?
205.70
205.75
최후 Km으로 반환
도 9의 단계 210으로 간다
205.85
매체를 거부
205.80