(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년07월03일
(11) 등록번호 10-0843814
(24) 등록일자 2008년06월27일

(51) Int. Cl.
G06F 19/00 (2006.01)
(21) 출원번호 10-2001-0044272
(22) 출원일자 2001년07월23일
심사청구일자 2006년07월21일
(65) 공개번호 10-2002-0008797
(43) 공개일자 2002년01월31일
(30) 우선권주장
JP-P-2000-00222123 2000년07월24일 일본(JP)
JP-P-2000-00247463 2000년08월17일 일본(JP)
(56) 선행기술조사문헌
EP0994475 A1*
KR1019980081632 A*
KR1019980025007 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
소니 가부시끼 가이샤
일본국 도쿄도 미나토쿠 코난 1-7-1
(72) 발명자
오까우에다꾸미
일본도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내
(74) 대리인
구영창, 장수길

전체 청구항 수 : 총 12 항 심사관 : 이해평

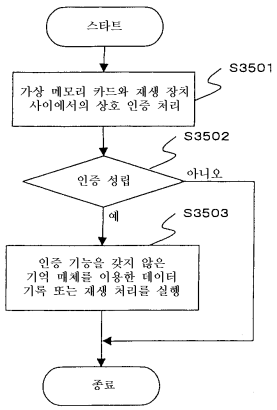
(54) 데이터 처리 장치, 데이터 처리 방법, 라이센스 시스템, 및 기록 매체

(57) 요 약

인증 처리를 확실하게 실행시켜, 정당한 라이센스를 갖는 장치에서만 콘텐츠 이용을 가능하게 하는 시스템을 실현한다.

인증 키를 유효화 키 블록(EKB)에 의해 데이터 처리 장치에 제공한다. 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에도 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리의 성립을 기억 장치로부터의 데이터 재생 처리 또는 기억 장치에 대한 데이터 기록 처리의 실행 조건으로 한다. 부정한 데이터 처리 장치에서는 복호 불가능한 유효화 키 블록(EKB)에 의해 인증 키를 제공하는 구성으로 함으로써, 정당한 데이터 처리 장치만이 가상 메모리 디바이스와의 인증이 성립하며, 콘텐츠 이용이 가능하게 된다.

대표도 - 도36



특허청구의 범위

청구항 1

기억 장치로부터의 데이터 재생 또는 기억 장치에 대한 데이터 기록을 행하는 데이터 처리 장치로서,

상기 데이터 처리 장치는,

데이터 처리 장치와 기억 장치 사이에서의 상호 인증의 성립을 조건으로 하여, 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 구성을 포함하고,

상기 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에, 상기 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리를 실행하고, 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리의 성립을 조건으로 하여 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 구성을 포함하는 것을 특징으로 하는 데이터 처리 장치.

청구항 2

제1항에 있어서,

상기 데이터 처리 장치는,

데이터 재생 또는 데이터 기록을 행하는 기억 장치가 상호 인증 가능한지의 여부를 판정하여, 상호 인증 처리가 가능한 경우에는 해당 기억 장치와의 사이에서 상호 인증 처리를 실행하는 구성을 포함하는 것을 특징으로 하는 데이터 처리 장치.

청구항 3

제1항에 있어서,

상기 데이터 처리 장치는,

복수의 데이터 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하는 패스 상의 갱신 키, 및 하위 키에 의한 상위 키의 암호화 처리 데이터를 포함하는 유효화 키 블록(EKB)에 의해 암호화된 인증 키인 EKB 배신 인증 키를 포함하고,

상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리는, 상기 EKB 배신 인증 키와, 상기 가상 메모리 디바이스에 사전에 저장된 인증 키를 적용하여 실행하는 구성인 것을 특징으로 하는 데이터 처리 장치.

청구항 4

제3항에 있어서,

상기 EKB 배신 인증 키를 포함하는 유효화 키 블록(EKB)은, 상기 키 트리의 리프를 구성하는 데이터 처리 장치 중, 정당한 라이센스를 갖는 데이터 처리 장치에서만 복호 가능하고, 정당한 라이센스를 갖지 않은 부정한 데이터 처리 장치에서는 복호 불가능한 유효화 키 블록(EKB)으로서 구성되고, 부정 데이터 처리 장치에 있어서의 상기 가상 메모리 디바이스와의 인증 성립을 방지하여, 부정 데이터 처리 장치의 배제(리보크)를 행하는 구성인 것을 특징으로 하는 데이터 처리 장치.

청구항 5

제3항에 있어서,

상기 유효화 키 블록(EKB)에 의해 암호화되어 제공되는 EKB 배신 인증 키는 세대 (버전) 관리가 이루어지고, 세대마다의 갱신 처리가 실행되는 구성인 것을 특징으로 하는 데이터 처리 장치.

청구항 6

제1항에 있어서,

상기 데이터 처리 장치는,

복수의 정보 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리 구성 중 자기 리프에 대응하여 설정된 리프 키를, 해당 데이터 처리 장치 고유의 스토리지 키(Kstd)에 의해 암호화하여 데이터 처리 장치 내의 기억 수단에 저장한 구성을 포함하는 것을 특징으로 하는 데이터 처리 장치.

청구항 7

제1항에 있어서,

상기 데이터 처리 장치는,

복수의 데이터 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리 구성 중 자기 리프에 대응하여 설정된 리프 키에 기초하여, 상기 키 트리의 자기 리프로부터 상위에 이르는 패스 상의 복수단의 상이한 노드 키를 개별적으로 암호화한 암호화 키의 집합으로서의 디바이스 키 블록(DKB)을 데이터 처리 장치 내의 기억 수단에 저장한 구성을 포함하는 것을 특징으로 하는 데이터 처리 장치.

청구항 8

기억 장치로부터의 데이터 재생 또는 기억 장치에 대한 데이터 기록을 행하는 데이터 처리 방법으로서,

상기 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에, 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리를 실행하는 단계, 및

상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리의 성립을 조건으로 하여 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 단계

를 포함하는 것을 특징으로 하는 데이터 처리 방법.

청구항 9

제8항에 있어서,

상기 데이터 처리 방법에 있어서,

데이터 재생 또는 데이터 기록을 행하는 기억 장치가 상호 인증 가능한지의 여부를 판정하는 단계를 더 포함하고,

상호 인증 처리가 가능한 경우에는 해당 기억 장치와의 사이에서 상호 인증 처리를 실행하는 것을 특징으로 하는 데이터 처리 방법.

청구항 10

제8항에 있어서,

상기 데이터 처리 방법에 있어서,

상기 데이터 처리 장치는, 복수의 데이터 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하는 패스 상의 갱신 키, 및 하위 키에 의한 상위 키의 암호화 처리 데이터를 포함하는 유효화 키 블록(EKB)에 의해 암호화된 인증 키인 EKB 배신 인증 키를 포함하고,

상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리는, 상기 EKB 배신 인증 키와, 상기 가상 메모리 디바이스에 사전에 저장된 인증 키를 적용하여 실행하는 것을 특징으로 하는 데이터 처리 방법.

청구항 11

데이터 처리 장치에 대한 라이센스를 부여하는 라이센스 시스템으로서,

복수의 데이터 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하는 패스 상의 갱신 키, 및 하위 키에 의한 상위 키의 암호화 처리 데이터

를 포함하는 유효화 키 블록(EKB)에 의해 암호화된 인증 키인 EKB 배신 인증 키를 제공하고,

데이터 처리 장치는, 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에 있어서도 상기 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리의 성립을, 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리의 실행 조건으로 한 구성을 포함하고,

상기 EKB 배신 인증 키를 제공하는 유효화 키 블록(EKB)은, 상기 키 트리의 리프를 구성하는 데이터 처리 장치 중, 정당한 라이센스를 갖는 데이터 처리 장치에서만 복호 가능하고, 정당한 라이센스를 갖지 않은 부정한 데이터 처리 장치에서는 복호 불가능한 유효화 키 블록(EKB)으로서 제공함으로써, 부정한 데이터 처리 장치에서의 상기 가상 메모리 디바이스와의 인증 성립을 방지하여, 부정한 데이터 처리 장치에서의 콘텐츠 이용을 배제 가능하게 한 구성을 포함하는 것을 특징으로 하는 라이센스 시스템.

청구항 12

기억 장치로부터의 데이터 재생 또는 기억 장치에 대한 데이터 기록을 행하는 데이터 처리를 컴퓨터 시스템상에서 실행시키는 컴퓨터 프로그램을 기록한 컴퓨터에 의해 판독 가능한 기록 매체로서,

상기 컴퓨터 프로그램은,

상기 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에, 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리를 실행하는 단계, 및

상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리의 성립을 조건으로 하여 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 단계

를 포함하는 것을 특징으로 하는 컴퓨터에 의해 판독 가능한 기록 매체.

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<91> 본 발명은, 데이터 처리 장치, 데이터 처리 방법, 및 라이센스 시스템, 및 프로그램 제공 매체에 관한 것이다. 특히, 트리 구조의 계층적 키 배신 방식을 이용함으로써 인증 키 배신을 정당 라이센스를 갖는 디바이스에 대해서만 복호 가능한 형태로 배신함으로써, 계층적 키 배신 트리의 관리 하의 디바이스에 있어서의 콘텐츠 이용의 라이센스 관리를 가능하게 함과 함께, 상호 인증 처리 기능을 갖지 않은 기억 장치를 이용한 데이터 기록 또는 데이터 재생에 있어서도 라이센스에 기초한 콘텐츠 이용 제한을 가능하게 한 데이터 처리 장치, 데이터 처리 방법, 및 라이센스 시스템, 및 프로그램 제공 매체에 관한 것이다.

<92> 최근, 음악 데이터, 게임 프로그램, 화상 데이터 등, 여러가지 소프트웨어 데이터(이하, 이들을 콘텐츠(Content)라고 함)를 인터넷 등의 네트워크, 또는 메모리 카드, DVD, CD 등의 유통 가능한 기억 매체를 통해 유통시키는 콘텐츠 유통이 활발해져 왔다. 이들의 유통 콘텐츠는 사용자가 소유하는 PC (Personal Computer), 재생 전용기, 또는 게임 기기에서의 콘텐츠 데이터의 수신, 또는 메모리 카드, CD, DVD 등의 기억 매체의 장착에 의해 콘텐츠 재생 처리가 실행되거나, 또는 외부로부터의 입력 콘텐츠를 재생기, PC 등에 내장한 기록 디바이스, 예를 들면 메모리 카드, 하드디스크 등에 저장하고, 재차 저장 매체로부터 재생하는 등의 방법에 의해 이용된다.

<93> 재생 장치, 게임 기기, PC 등의 정보 기기에는 유통 콘텐츠를 네트워크로부터 수신하기 위해, 또는 DVD, CD 등을 액세스하기 위한 인터페이스를 구비하고, 또한 콘텐츠의 재생에 필요한 제어 수단, 프로그램, 데이터의 메모리 영역으로서 사용되는 RAM, ROM 등을 포함한다.

<94> 음악 데이터, 화상 데이터, 또는 프로그램 등의 여러 콘텐츠는 재생 기기로서 이용되는 재생 장치, 게임 기기, PC 등의 정보 기기 본체로부터의 사용자 지시, 또는 접속된 입력 수단을 통한 사용자의 지시에 따라, 예를 들면 내장 또는 착탈 가능한 기억 매체로부터 호출되고, 정보 기기 본체, 또는 접속된 디스플레이, 스피커 등을 통해 재생된다.

<95> 게임 프로그램, 음악 데이터, 화상 데이터 등, 많은 소프트웨어 콘텐츠는 일반적으로 그 작성자, 판매자에게 반포권 등이 보유되고 있다. 따라서, 이들 콘텐츠의 배포시에는 일정한 이용 제한 즉, 정규 사용자에 대해서만 소프트웨어의 사용을 허락하고, 허가가 없는 복제 등이 행해지지 않도록 하는 즉, 시큐러티를 고려한 구성을 취하는 것이 일반적이다.

<96> 사용자에 대한 이용 제한을 실현하는 하나의 수법이, 배포 콘텐츠의 암호화 처리이다. 즉, 예를 들면 인터넷 등을 통해 암호화된 음성 데이터, 화상 데이터, 게임 프로그램 등의 각종 콘텐츠를 배포함과 함께, 정규 사용자라고 확인된 자에 대해서만, 배포된 암호화 콘텐츠를 복호하는 수단 즉, 복호 키를 부여하는 구성이다.

<97> 암호화 데이터는, 소정의 수속에 의한 복호화 처리에 의해 이용 가능한 복호 데이터(평문(平文))로 복귀할 수 있다. 이러한 정보의 암호화 처리에 암호화 키를 이용하여, 복호화 처리에 복호화 키를 이용하는 데이터 암호화, 복호화 방법은 종래부터 잘 알려져 있다.

<98> 암호화 키와 복호화 키를 이용하는 데이터 암호화 복호화 방법의 형태에는 여러가지 종류가 있지만, 그 하나의 예로서 소위 공통 키 암호화 방식이라고 하는 방식이 있다. 공통 키 암호화 방식은, 데이터의 암호화 처리에 이용하는 암호화 키와 데이터의 복호화에 이용하는 복호화 키를 공통의 것으로 하여, 정규 사용자에게 이들 암호화 처리, 복호화에 이용하는 공통 키를 부여하고, 키를 갖지 않은 부정 사용자에 의한 데이터 액세스를 배제하는 것이다. 이 방식의 대표적인 방식으로 DES(Data encryption standard: 데이터 암호 표준)가 있다.

<99> 상술된 암호화 처리, 복호화에 이용되는 암호화 키, 복호화 키는 예를 들면 어느 한 패스워드 등에 기초하여 해시 함수 등의 일방향성 함수를 적용하여 얻을 수 있다. 일방향성 함수란, 그 출력으로부터 반대로 입력을 구하는 것은 매우 곤란한 함수이다. 예를 들면 사용자가 정한 패스워드를 입력으로 하여 일방향성 함수를 적용하고, 그 출력에 기초하여 암호화 키, 복호화 키를 생성하는 것이다. 이와 같이 함으로써 얻어진 암호화 키, 복호화 키로부터, 반대로 그 오리지널 데이터인 패스워드를 구하는 것은 실질적으로 불가능하게 된다.

<100> 또한, 암호화할 때에 사용하는 암호화 키에 의한 처리와, 복호할 때에 사용하는 복호화 키의 처리를 다른 알고리즘으로 한 방식이 소위 공개 키 암호화 방식이라고 하는 것이다. 공개 키 암호화 방식은, 불특정 사용자가 사용 가능한 공개 키를 사용하는 방법으로서, 특정 개인에 대한 암호화 문서를, 그 특정 개인이 발행한 공개 키를 이용하여 암호화 처리를 행한다. 공개 키에 의해 암호화된 문서는, 그 암호화 처리에 사용된 공개 키에 대응하는 비밀 키에 의해서만 복호 처리가 가능하다. 비밀 키는, 공개 키를 발행한 개인만이 소유하므로, 그 공개 키에 의해 암호화된 문서는 비밀 키를 갖는 개인만을 복호할 수 있다. 공개 키 암호화 방식의 대표적인 것에는 RSA(Rivest-Shamir-Adleman) 암호가 있다. 이러한 암호화 방식을 이용함으로써, 암호화 콘텐츠를 정규 사용자에 대해서만 복호 가능하게 하는 시스템이 가능하게 된다.

발명이 이루고자 하는 기술적 과제

<101> 상기된 바와 같은 콘텐츠 배신 시스템에서는 콘텐츠를 암호화하여 사용자에게 네트워크, 또는 DVD, CD 등의 기록 매체에 저장하여 제공하며, 암호화 콘텐츠를 복호하는 콘텐츠 키를 정당한 사용자에게만 제공하는 구성이 많이 채용되고 있다. 콘텐츠 키 자체의 부정한 복사 등을 막기 위한 콘텐츠 키를 암호화하여 정당한 사용자에게 제공하고, 정당한 사용자만이 갖는 복호 키를 이용하여 암호화 콘텐츠 키를 복호하여 콘텐츠 키를 사용 가능하게 하는 구성이 제안되고 있다.

<102> 정당한 사용자인지의 여부의 판단은, 일반적으로는 예를 들면 콘텐츠의 송신자인 콘텐츠 프로바이더와 사용자 디바이스 사이에서 콘텐츠, 또는 콘텐츠 키의 배신 전에 인증 처리를 실행함으로써 행한다. 일반적인 인증 처리에 있어서는, 상대의 확인을 행함과 함께 그 통신에서만 유효한 세션 키를 생성하여, 인증이 성립한 경우에 생성한 세션 키를 이용하여 데이터, 예를 들면 콘텐츠 또는 콘텐츠 키를 암호화하여 통신을 행한다. 인증 방식에는 공통 키 암호 방식을 이용한 상호 인증과, 공개 키 방식을 사용한 인증 방식이 있지만, 공통 키를 사용한 인증에서는 시스템 와이드에서 공통인 키가 필요하게 되며, 갱신 처리 시에 불편하다. 또한, 공개 키 방식에서는 계산 부하가 크고, 또한 필요한 메모리량도 커져, 각 디바이스에 이러한 처리 수단을 설치하는 것은 바람직한 구성이라고는 할 수 없다.

<103> 본 발명에서는, 상술된 바와 같은 데이터의 송신자, 수신자 사이의 상호 인증 처리에 의지하지 않고, 트리 구조의 계층적 키 배신 방식을 이용하여, 정당한 라이센스를 갖는 디바이스에 대해서만 복호 가능한 형태로 인증 키를 배신함으로써, 계층적 키 배신 트리의 관리 하의 디바이스에 있어서의 콘텐츠 이용의 라이센스 관리를 가능하게 함과 함께, 상호 인증 처리 기능을 갖지 않은 기억 장치를 이용한 데이터 기록 또는 데이터 재생에서도, 라이센스에 기초한 콘텐츠 이용 제한을 가능하게 한 데이터 처리 장치, 데이터 처리 방법, 및 라이센스 시스템,

및 프로그램 제공 매체를 제공하는 것을 목적으로 한다.

발명의 구성 및 작용

<104> 본 발명의 제1 측면은, 기억 장치로부터의 데이터 재생 또는 기억 장치에 대한 데이터 기록을 행하는 데이터 처리 장치에 있어서, 상기 데이터 처리 장치는, 데이터 처리 장치와 기억 장치 사이에서의 상호 인증의 성립을 조건으로 하여, 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 구성을 포함하고, 상기 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에, 상기 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리를 실행하고, 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리의 성립을 조건으로 하여 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 구성을 포함하는 것을 특징으로 하는 데이터 처리 장치에 있다.

<105> 또한, 본 발명의 데이터 처리 장치의 일 실시예에 있어서, 상기 데이터 처리 장치는, 데이터 재생 또는 데이터 기록을 행하는 기억 장치가 상호 인증 가능한지의 여부를 판정하여, 상기 상호 인증 처리가 가능한 경우에는 상기 기억 장치와의 사이에서 상기 상호 인증 처리를 실행하는 구성을 포함하는 것을 특징으로 한다.

<106> 또한, 본 발명의 데이터 처리 장치의 일 실시예에서, 상기 데이터 처리 장치는, 복수의 데이터 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 포함하는 패스 상의 갱신 키, 및 하위 키에 의한 상위 키의 암호화 처리 데이터를 포함하는 유효화 키 블록(EKB)에 의해 암호화된 인증 키인 EKB 배신 인증 키를 갖고, 상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리는 상기 EKB 배신 인증 키와, 상기가상 메모리 디바이스에 사전에 저장된 인증 키를 적용하여 실행하는 구성인 것을 특징으로 한다.

<107> 또한, 본 발명의 데이터 처리 장치의 일 실시예에 있어서, 상기 EKB 배신 인증 키를 포함하는 유효화 키 블록(EKB)은, 상기 키 트리의 리프를 포함하는 데이터 처리 장치 중, 정당한 라이센스를 갖는 데이터 처리 장치에서만 복호 가능하고, 정당한 라이센스를 갖지 않은 부정한 데이터 처리 장치에서는 복호 불가능한 유효화 키 블록(EKB)으로서 구성하며, 부정한 데이터 처리 장치에서의 상기 가상 메모리 디바이스와의 인증 성립을 방지하여, 부정한 데이터 처리 장치의 배제(리보크)를 행하는 구성인 것을 특징으로 한다.

<108> 또한, 본 발명의 데이터 처리 장치의 일 실시예에 있어서, 상기 유효화 키 블럭(EKB)에 의해 암호화되고 제공되는 EKB 배신 인증 키는, 세대 (버전) 관리가 이루어지며, 세대마다의 갱신 처리가 실행되는 구성인 것을 특징으로 한다.

<109> 또한, 본 발명의 데이터 처리 장치의 일 실시예에 있어서, 상기 데이터 처리 장치는 복수의 정보 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리 구성 중 자기 리프에 대응하여 설정된 리프 키를, 그 데이터 처리 장치 고유의 스토리지 키(Kstd)로 암호화하여 데이터 처리 장치 내의 기억 수단에 저장한 구성을 포함하는 것을 특징으로 한다.

<110> 또한, 본 발명의 데이터 처리 장치의 일 실시예에 있어서, 상기 데이터 처리 장치는 복수의 데이터 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리 구성 중 자기 리프에 대응하여 설정된 리프 키에 기초하여 상기 키 트리의 자기 리프로부터 상위에 이르는 패스 상의 복수단이 다른 노드 키를 개별로 암호화한 암호화 키의 집합으로서의 디바이스 키 블록(DKB)을 데이터 처리 장치 내의 기억 수단에 저장한 구성을 포함하는 것을 특징으로 한다.

<111> 또한, 본 발명의 제2 측면은, 기억 장치로부터의 데이터 재생 또는 기억 장치에 대한 데이터 기록을 행하는 데이터 처리 방법에 있어서, 상기 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에, 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리를 실행하는 단계, 및 상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리의 성립을 조건으로 하여 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 단계를 포함하는 것을 특징으로 하는 데이터 처리 방법에 있다.

<112> 또한, 본 발명의 데이터 처리 방법의 일 실시예에 있어서, 데이터 재생 또는 데이터 기록을 행하는 기억 장치가 상호 인증 가능한지의 여부를 판정하는 단계를 포함하고, 상호 인증 처리가 가능한 경우에는 상기 기억 장치와의 사이에서 상호 인증 처리를 실행하는 것을 특징으로 한다.

<113> 또한, 본 발명의 데이터 처리 방법의 일 실시예에 있어서, 상기 데이터 처리 장치는, 복수의 데이터 처리 장치

를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 포함하는 패스 상의 갱신 키, 및 하위 키에 의한 상위 키의 암호화 처리 데이터를 포함하는 유효화 키 블록(EKB)에 의해 암호화된 인증 키인 EKB 배신 인증 키를 구비하고, 상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리는 상기 EKB 배신 인증 키와, 상기 가상 메모리 디바이스에 사전에 저장된 인증 키를 적용하여 실행하는 것을 특징으로 한다.

<114> 또한, 본 발명의 제3 측면은, 데이터 처리 장치에 대한 라이센스를 부여하는 라이센스 시스템에 있어서, 복수의 데이터 처리 장치를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 포함하는 패스 상의 갱신 키, 및 하위 키에 의한 상위 키의 암호화 처리 데이터를 포함하는 유효화 키 블록(EKB)에 의해 암호화된 인증 키인 EKB 배신 인증 키를 제공하고, 데이터 처리 장치는 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에서도 상기 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리의 성립을, 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리의 실행 조건으로 한 구성을 포함하고, 상기 EKB 배신 인증 키를 제공하는 유효화 키 블록(EKB)은, 상기 키 트리의 리프를 포함하는 데이터 처리 장치 중 정당한 라이센스를 갖는 데이터 처리 장치에서만 복호 가능하여, 정당한 라이센스를 갖지 않은 부정한 데이터 처리 장치에서는 복호 불가능한 유효화 키 블록(EKB)으로서 제공함으로써, 부정한 데이터 처리 장치에서의 상기 가상 메모리 디바이스와의 인증 성립을 방지하여, 부정한 데이터 처리 장치에서의 콘텐츠 이용을 배제 가능하게 한 구성을 포함하는 것을 특징으로 하는 라이센스 시스템에 있다.

<115> 또한, 본 발명의 제4 측면은, 기억 장치로부터의 데이터 재생 또는 기억 장치에 대한 데이터 기록을 행하는 데이터 처리를 컴퓨터 시스템 상에서 실행시키는 컴퓨터 프로그램을 제공하는 프로그램 제공 매체에 있어서, 상기 컴퓨터 프로그램은, 상기 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에, 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리를 실행하는 단계, 및 상기 데이터 처리 장치와 상기 가상 메모리 디바이스 사이에서 실행되는 상호 인증 처리의 성립을 조건으로 하여 상기 기억 장치로부터의 데이터 재생 처리 또는 상기 기억 장치에 대한 데이터 기록 처리를 실행하는 단계를 포함하는 것을 특징으로 하는 프로그램 제공 매체에 있다.

<116> 또, 본 발명의 제4 측면에 따른 프로그램 제공 매체는, 예를 들면 여러가지 프로그램 코드를 실행 가능한 범용 컴퓨터 시스템에 대해 컴퓨터 프로그램을 컴퓨터 판독 가능한 형식으로 제공하는 매체이다. 매체는, CD나 FD, MO 등의 기록 매체, 또는 네트워크 등의 전송 매체 등 그 형태는 특별히 한정되지 않는다.

<117> 이러한 프로그램 제공 매체는, 컴퓨터 시스템 상에서 소정의 컴퓨터 프로그램의 기능을 실현하기 위한, 컴퓨터 프로그램과 제공 매체와의 구조상 또는 기능상의 협동적 관계를 정의한 것이다. 바꾸어 말하면, 그 제공 매체를 통해 컴퓨터 프로그램을 컴퓨터 시스템에 인스톨함으로써, 컴퓨터 시스템 상에서는 협동적 작용이 발휘되고, 본 발명의 다른 측면과 동일한 작용 효과를 얻을 수 있는 것이다.

<118> 본 발명의 또 다른 목적, 특징이나 이점은 후술하는 본 발명의 실시예나 첨부하는 도면에 기초한 상세한 설명에 의해 분명해질 것이다.

<119> <발명의 실시예>

<120> [시스템 개요]

<121> 도 1에는 본 발명의 데이터 처리 시스템의 적용 가능한 콘텐츠 배신 시스템예가 도시되어 있다. 콘텐츠 배신 수단(10)은 데이터 처리 수단(20)에 대하여, 콘텐츠 또는 콘텐츠 키, 기타, 인증 처리 키 등의 데이터를 암호화하여 송신한다. 데이터 처리 수단(20)에서는, 수신한 암호화 콘텐츠, 또는 암호화 콘텐츠 키 등을 복호하여 콘텐츠 또는 콘텐츠 키 등을 취득하여, 화상 데이터, 음성 데이터의 재생, 또는 각종 프로그램을 실행한다. 콘텐츠의 배신 수단(10)과 데이터 처리 수단(20) 사이의 데이터 교환은 인터넷 등의 네트워크를 통해 또는 DVD, CD, 그 밖의 유통 가능한 기억 매체를 통해 실행된다.

<122> 데이터 처리 수단(20)은, 예를 들면 플래시 메모리 등의 기억 수단을 구비한 메모리 카드 등의 데이터 기억 수단(30)에 데이터를 저장하여 보존한다. 데이터 기억 수단(30)에는, 암호 처리 기능을 구비한 기억 수단으로서의 예를 들면 메모리 카드(구체예로서는 메모리 스틱(Memory Stick: 상표))가 포함된다. 데이터 처리 수단(20)으로부터 데이터 기억 수단(30)에 대한 데이터 저장 처리, 및 데이터 기억 수단(30)으로부터 데이터 처리 수단에 대한 데이터 이동시에는 상호 인증 처리, 및 데이터의 암호 처리가 실행되어 부정한 데이터 복사의 방지를 꾀할 수 있다.

<123> 또, 데이터 처리 수단(20)에 포함되는 각 기기 사이에서의 콘텐츠 데이터의 이동도 가능하고, 이 때에도 기기 사이의 상호 인증 처리, 데이터의 암호 처리가 실행된다.

<124> 콘텐츠 배신 수단(10)으로서는 인터넷(11), 위성 방송(12), 전화 회선(13), DVD, CD 등의 미디어(14) 등이 있고, 한편 데이터 처리 수단(20)의 디바이스로서는 퍼스널 컴퓨터(PC: 21), 포터블 디바이스(PD: 22), 휴대 전화, PDA(Personal Digital Atssistants) 등의 휴대 기기(23), DVD, CD 플레이어 등의 기록 재생기, 게임 단말(24), 메모리 카드(ex. 메모리 스틱(상표))를 이용한 재생 장치(25) 등이 있다. 이들 데이터 처리 수단(20)의 각 디바이스는 콘텐츠 배신 수단(10)으로부터 제공되는 콘텐츠를 네트워크 등의 통신 수단 또는 다른 데이터 처리 수단, 또는 데이터 기억 수단(30)으로부터 취득 가능하다.

<125> 도 2에는 대표적인 콘텐츠 데이터의 이동 처리예가 도시되어 있다. 도 2에 도시된 시스템은 퍼스널 컴퓨터(PC: 100), 재생 장치(200) 및 기억 장치(300) 사이에서의 데이터(콘텐츠)의 이동 처리예를 나타낸다. PC(100)는 프로그램 및 데이터 기억용의 하드디스크(HD)를 구비하고, 또한 외부 기억 매체로서의 CD, DVD 등을 장착 가능한 구성을 구비한다.

<126> 퍼스널 컴퓨터(PC: 100)는 인터넷, 공중 회선 등의 각종 네트워크에 접속 가능하고, 예를 들면 EMD(Electronic Music Distribution: 전자 음악 배신) 등의 서비스를 제공하는 도시하지 않은 서비스 프로바이더의 호스트 컴퓨터로부터 네트워크를 하여 오디오 데이터, 화상 데이터, 프로그램 등의 각종 데이터를 수신하고, 수신한 데이터를 필요에 따라 복호하여, 재생 장치(200)에 출력한다. 또한, 퍼스널 컴퓨터(PC: 100)는 콘텐츠 데이터를 수신하는 데 있어서, 필요에 따라 서비스 프로바이더의 호스트 컴퓨터 사이에서 인증 처리 및 과금 처리 등을 행한다. 또한, 퍼스널 컴퓨터(PC: 100)는 예를 들면 CD, DVD로부터 입력한 데이터를 재생 장치(200)로 출력한다.

<127> 기억 장치(300)는 재생 장치(200)에 대해 착탈 가능한 장치, 예를 들면 메모리 스틱(Memory Stick: 상표)이고, 플래시 메모리 등의 재기입 가능한 반도체 메모리를 내장하고 있다.

<128> 도 2에 도시된 바와 같이, PC(100), 재생 장치(200), 기억 장치(300) 사이에서의 데이터 이동, 예를 들면 음악 데이터, 화상 데이터 등의 데이터 재생, 데이터 기록, 데이터 복사 등의 처리시에는 데이터 이동 기기 사이에서 상호 인증 처리가 실행되며, 부정한 기기를 이용한 데이터 이동은 방지된다. 이들의 처리에 대해서는 후술한다. 또한, 콘텐츠 데이터의 네트워크 또는 각종 기억 매체를 통하는 배신, 또한 PC와 재생 장치 상호간, 또는 재생 장치와 메모리 카드 등의 기억 장치 사이에서의 콘텐츠 이동시에는 콘텐츠를 암호화함으로써 데이터의 시큐러티가 보전된다.

<129> [키 배신 구성으로서의 트리(tree) 구조에 대하여]

<130> 상술된 바와 같은 콘텐츠에 대한 암호 처리에 적용하는 암호 키, 예를 들면 콘텐츠의 암호 처리에 적용하는 콘텐츠 키, 또는 콘텐츠 키를 암호화하기 위한 콘텐츠 키 암호화 키 등의 여러 암호 처리 키를, 안전하게 정당한 라이센스를 갖는 디바이스에 배신하는 구성으로서, 계층 키 트리 구성에 대해 도 3 이하를 이용하여 설명한다.

<131> 도 3의 최하단에 도시된 번호 0~15는 콘텐츠 데이터의 재생, 실행을 행하는 데이터 처리 수단(20)을 포함하는 개개의 디바이스, 예를 들면 콘텐츠(음악 데이터) 재생 장치이다. 즉, 도 3에 도시된 계층 트리 구조의 각 리프(leaf)가 각각의 디바이스에 상당한다.

<132> 각 디바이스 0~15는 제조 시 또는 출하 시, 또는 그 후에 있어서 도 3에 도시된 계층 트리 구조에 있어서의 자신의 리프로부터 루트에 이르기까지의 노드에 할당된 키(노드 키) 및 각 리프의 리프 키를 포함하는 키 세트를 메모리에 저장한다. 도 3의 최하단에 도시된 K0000~K1111이 각 디바이스 0~15에 각각 할당된 리프 키이고, 최상단의 KR(루트 키)로부터 최하단으로부터, 2번째 마디(노드)에 기재된 키: KR~K111을 노드 키로 한다.

<133> 도 3에 도시된 트리 구성에 있어서, 예를 들면 디바이스 0은 리프 키 K0000와, 노드 키: K000, K00, K0, KR을 소유한다. 디바이스 5는 K0101, K010, K01, K0, KR을 소유한다. 디바이스 15는 K1111, K111, K11, K1, KR을 소유한다. 또, 도 3의 트리에는 디바이스가 0~15의 16개만 기재되며, 트리 구조도 4단 구성의 균형을 잡은 좌우 대칭 구성으로서 도시되어 있지만, 더 많은 디바이스가 트리 내에 구성되며, 또한 트리의 각 부에서 다른 단수 구성을 구비한 것이 가능하다.

<134> 또한, 도 3의 트리 구조에 포함되는 각 디바이스에는 여러 기록 매체, 예를 들면 디바이스 매립형 또는 디바이스에 착탈 가능하게 구성된 플래시 메모리 등을 사용한 메모리 카드, DVD, CD, MD 등 여러 타입의 기억 장치를

이용 가능한 디바이스가 포함되어 있다. 또한, 여러 어플리케이션 서비스가 공존 가능하다. 이러한 다른 디바이스, 다른 어플리케이션의 공존 구성 위에 도 3에 도시된 콘텐츠 또는 키 배포 구성인 계층 트리 구조가 적용된다.

<135> 이들의 여러가지 디바이스, 어플리케이션이 공존하는 시스템에 있어서, 예를 들면 도 3의 점선으로 된 부분 즉, 디바이스 0, 1, 2, 3을 동일한 기록 매체를 이용하는 하나의 그룹으로서 설정한다. 예를 들면, 이 점선으로 된 그룹 내에 포함되는 디바이스에 대해서는 통합하여 공통의 콘텐츠를 암호화하여 프로바이더로부터 송부하거나, 각 디바이스 공통으로 사용하는 콘텐츠 키를 송부하거나, 또는 각 디바이스로부터 프로바이더 또는 결제 기관 등에 콘텐츠 요금의 지불 데이터를 역시 암호화하여 출력한다고 한 처리가 실행된다. 콘텐츠 프로바이더, 또는 결제 처리 기관 등, 각 디바이스와의 데이터 송수신을 행하는 기관은 도 3의 점선으로 된 부분 즉, 디바이스 0, 1, 2, 3을 하나의 그룹으로서 일괄하여 데이터를 송부하는 처리를 실행한다. 이러한 그룹은, 도 3의 트리 중에 복수개 존재한다. 콘텐츠 프로바이더, 또는 결제 처리 기관 등, 각 디바이스와의 데이터 송수신을 행하는 기관은 메시지 데이터 배신 수단으로서 기능한다.

<136> 또, 노드 키, 리프 키는 어떤 하나의 키 관리 센터에 의해 통괄하여 관리해도 좋고, 각 그룹에 대한 여러가지 데이터 송수신을 행하는 프로바이더, 결제 기관 등의 메시지 데이터 배신 수단에 의해 그룹마다 관리하는 구성으로 해도 좋다. 이들의 노드 키, 리프 키는 예를 들면 키의 누설 등의 경우에 갱신 처리가 실행되며, 이 갱신 처리는 키 관리 센터, 프로바이더, 결제 기관 등이 실행된다.

<137> 이 트리 구조에 있어서, 도 3으로부터 분명히 알 수 있듯이 하나의 그룹에 포함되는 3개의 디바이스 0, 1, 2, 3은 노드 키로서 공통의 키 K00, K0, KR을 보유한다. 이 노드 키 공유 구성을 이용함으로써, 예를 들면 공통의 콘텐츠 키를 디바이스 0, 1, 2, 3에만 제공하는 것이 가능하게 된다. 예를 들면, 공통으로 보유하는 노드 키 K00 자체를 콘텐츠 키로서 설정하면, 새로운 키 송부를 실행하지 않고 디바이스 0, 1, 2, 3에만 공통의 콘텐츠 키의 설정이 가능하다. 또한, 새로운 콘텐츠 키 Kcon을 노드 키 K00으로 암호화한 값 Enc(K00, Kcon)을 네트워크를 통해 또는 기록 매체에 저장하여 디바이스 0, 1, 2, 3에 배포하면, 디바이스 0, 1, 2, 3만이 각각의 디바이스에서 보유하는 공유 노드 키 K00을 이용하여 암호 Enc(K00, Kcon)를 풀어 콘텐츠 키: Kcon을 얻는 것이 가능하게 된다. 또, Enc(Ka, Kb)는 Kb를 Ka에 의해 암호화한 데이터인 것을 나타낸다.

<138> 또한, 어느 한 시점 t에서 디바이스 3이 소유하는 키: K0011, K001, K00, K0, KR이 공격자(해커)에 의해 해석되어 노출된 것이 발각된 경우, 그 이후 시스템(디바이스 0, 1, 2, 3의 그룹)에서 송수신되는 데이터를 지키기 위해 디바이스 3을 시스템으로부터 분리할 필요가 있다. 그로 인해, 노드 키: K001, K00, K0, KR을 각각 새로운 키 K(t)001, K(t)00, K(t)0, K(t)R에 갱신하고, 디바이스 0, 1, 2에 그 갱신 키를 전할 필요가 있다. 여기서, K(t)aaa는, 키 Kaaa의 세대(Generation): t의 갱신 키인 것을 나타낸다.

<139> 갱신 키의 배포 처리에 대하여 설명한다. 키의 갱신은, 예를 들면 도 4의 (a)에 도시된 유효화 키 블록(EKB: Enabling Key Block)이라고 하는 블록 데이터에 의해 구성되는 테이블을 예를 들면 네트워크, 또는 기록 매체에 저장하여 디바이스 0, 1, 2로 공급함으로써 실행된다. 또, 유효화 키 블록(EKB)은 도 3에 도시된 바와 같은 트리 구조를 포함하는 각 리프에 대응하는 디바이스에 새롭게 갱신된 키를 배포하기 위한 암호화 키에 의해 구성된다. 유효화 키 블록(EKB)은 키 갱신 블록(KRB: Key Renewal Block)이라고 하는 경우도 있다.

<140> 도 4의 (a)에 도시된 유효화 키 블록(EKB)에는 노드 키 갱신이 필요한 디바이스만이 갱신 가능한 데이터 구성을 구비한 블록 데이터로서 구성된다. 도 4의 예는 도 3에 도시된 트리 구조 내의 디바이스 0, 1, 2에 있어서 세대 t의 갱신 노드 키를 배포하는 것을 목적으로 하여 형성된 블록 데이터이다. 도 3으로부터 분명히 알 수 있듯이 디바이스 0, 디바이스 1은 갱신 노드 키로서 K(t)00, K(t)0, K(t)R이 필요하고, 디바이스 2는 갱신 노드 키로서 K(t)001, K(t)00, K(t)0, K(t)R이 필요하다.

<141> 도 4의 (a)의 EKB에 도시된 바와 같이 EKB에는 복수의 암호화 키가 포함된다. 최하단의 암호화 키는 Enc(K0010, K(t)001)이다. 이것은 디바이스 2가 구비한 리프 키 K0010에 의해 암호화된 갱신 노드 키 K(t)001이고, 디바이스 2는 자신이 구비한 리프 키에 의해 이 암호화 키를 복호하고, K(t)001을 얻을 수 있다. 또한, 복호에 의해 얻은 K(t)001을 이용하여, 도 4의 (a)의 하부로부터 2단째의 암호화 키 Enc(K(t)001, K(t)00)를 복호 가능하게 하고, 갱신 노드 키 K(t)00을 얻을 수 있다. 이하 순차적으로, 도 4의 (a) 위에서 2단째의 암호화 키 Enc(K(t)00, K(t)0)를 복호하고, 갱신 노드 키 K(t)0, 도 4의 (a) 위에서 1단째의 암호화 키 Enc(K(t)0, K(t)R)를 복호하여 K(t)R을 얻는다. 한편, 디바이스 K0000, K0001은 노드 키 K000은 갱신하는 대상에 포함되지 않고, 갱신 노드 키로서 필요한 것은 K(t)00, K(t)0, K(t)R이다. 디바이스 K0000, K0001은 도 4의 (a)의 위에서 3단째의 암호화 키 Enc(K000, K(t)00)를 복호하여 K(t)00을 취득하고, 이하 도 4의 (a)의 위에서 2단째의 암

호화 키 Enc(K(t)00, K(t)0)를 복호하고, 갱신 노드 키 K(t)0, 도 4의 (a)의 위에서 1단째의 암호화 키 Enc(K(t)0, K(t)R)를 복호하고 K(t)R을 얻는다. 이와 같이 하여, 디바이스 0, 1, 2는 갱신한 키 K(t)001, K(t)00, K(t)0, K(t)R을 얻을 수 있다. 또, 도 4의 (a)의 인덱스는 복호 키로서 사용하는 노드 키, 리프 키의 절대 번지를 나타낸다.

<142> 도 3에 도시된 트리 구조의 상위단의 노드 키: K(t)0, K(t)R의 갱신이 불필요하고, 노드 키 K00만의 갱신 처리가 필요한 경우에는 도 4의 (b)의 유효화 키 블록(EKB)을 이용하는 것으로, 갱신 노드 키 K(t)00을 디바이스 0, 1, 2에 배포할 수 있다.

<143> 도 4의 (b)에 도시된 EKB는 예를 들면 특정한 그룹에서 공유하는 새로운 콘텐츠 키를 배포하는 경우에 이용 가능하다. 구체예로서, 도 3에 점선으로 도시된 그룹 내의 디바이스 0, 1, 2, 3이 있는 기록 매체를 이용하고, 새로운 공통의 콘텐츠 키 K(t)con이 필요하다. 이 때, 디바이스 0, 1, 2, 3의 공통의 노드 키 K00을 갱신한 K(t)00을 이용하여 새로운 공통의 갱신 콘텐츠 키: K(t)con을 암호화한 데이터 Enc(K(t), K(t)con)를 도 4의 (b)에 도시된 EKB와 함께 배포한다. 이 배포에 의해 디바이스 4 등 그 밖의 그룹의 기기에서는 복호되지 않은 데이터로서의 배포가 가능하게 된다.

<144> 즉, 디바이스 0, 1, 2는 EKB를 처리하여 얻은 K(t)00을 이용하여 상기 암호문을 복호하면, t 시점에서의 콘텐츠 키 K(t)con을 얻는 것이 가능하게 된다.

<145> [EKB를 사용한 콘텐츠 키의 배포]

<146> 도 5에는, t 시점에서의 콘텐츠 키 K(t)con을 얻는 처리예로서, K(t)00을 이용하여 새로운 공통의 콘텐츠 키 K(t)con을 암호화한 데이터 Enc(K(t)00, K(t)con)와, 도 4의 (b)에 도시된 EKB를 기록 매체를 통해 수령한 디바이스 0의 처리가 도시되어 있다. 즉, EKB 에 의한 암호화 메시지 데이터를 콘텐츠 키 K(t)con으로 한 예이다.

<147> 도 5에 도시된 바와 같이, 디바이스 0은 기록 매체에 저장되어 있는 세대: t 시점의 EKB와 자신이 사전에 저장하고 있는 노드 키 K000을 이용하여 상술한 바와 같은 EKB 처리에 의해 노드 키 K(t)00을 생성한다. 또한, 복호한 갱신 노드 키 K(t)00을 이용하여 갱신 콘텐츠 키 K(t)con을 복호하여, 나중에 그것을 사용하기 위해 자신만이 구비한 리프 키 K0000으로 암호화하여 저장한다.

<148> [EKB의 포맷]

<149> 도 6에는 유효화 키 블록(EKB)의 포맷예가 도시되어 있다. 버전(601)은 유효화 키 블록(EKB)의 버전을 나타내는 식별자이다. 또, 버전은 최신 EKB를 식별하는 기능과 콘텐츠와의 대응 관계를 나타내는 기능을 갖는다. 깊이는 유효화 키 블록(EKB)의 배포처의 디바이스에 대한 계층 트리의 계층 수를 나타낸다. 데이터 포인터(603)는 유효화 키 블록(EKB) 내의 데이터부의 위치를 나타내는 포인터이고, 태그 포인터(604)는 태그부의 위치, 서명 포인터(605)는 서명의 위치를 나타내는 포인터이다.

<150> 데이터부(606)는 예를 들면 갱신하는 노드 키를 암호화한 데이터를 저장한다. 예를 들면, 도 5에 도시된 바와 같은 갱신된 노드 키에 관한 각 암호화 키 등을 저장한다.

<151> 태그부(607)는 데이터부에 저장된 암호화된 노드 키, 리프 키의 위치 관계를 나타내는 태그이다. 이 태그의 부여 룰을 도 7을 이용하여 설명한다. 도 7에서는 데이터로서 먼저 도 4의 (a)에서 설명한 유효화 키 블록(EKB)을 송부하는 예가 도시되어 있다. 이 때의 데이터는 도 7의 표(b)에 도시된 바와 같다. 이 때의 암호화 키에 포함되는 톱 노드의 어드레스를 톱 노드 어드레스라고 한다. 이 경우에는, 루트 키의 갱신 키 K(t)R이 포함되므로, 톱 노드 어드레스는 KR이 된다. 이 때, 예를 들면 최상단의 데이터 Enc(K(t)0, K(t)R)는 도 7의 (a)에 도시된 계층 트리에 나타내는 위치에 있다. 여기서, 다음 데이터는 Enc(K(t)00, K(t)0)이고, 트리 상에서는 앞 데이터의 좌측밑의 위치에 있다. 데이터가 있는 경우에는 태그가 0, 없는 경우에는 1이 설정된다. 태그는 {좌측(L) 태그, 우측(R) 태그}로서 설정된다. 최상단의 데이터 Enc(K(t)0, K(t)R)의 좌측으로는 데이터가 있으므로, L 태그 = 0, 우측으로는 데이터가 없으므로, R 태그 = 1이 된다. 이하, 모든 데이터에 태그가 설정되며, 도 7의 (c)에 도시된 데이터 열, 및 태그 열이 구성된다.

<152> 태그는 데이터 Enc(Kxxx, Kyyy)가 트리 구조의 어디에 위치하는지를 나타내기 위해 설정되는 것이다. 데이터부에 저장되는 키 데이터 Enc(Kxxx, Kyyy) …는 단순하게 암호화된 키의 나열 데이터에 지나지 않으므로, 상술된 태그에 의해 데이터로서 저장된 암호화 키의 트리 상의 위치를 판별 가능하게 한 것이다. 상술한 태그를 이용하지 않고, 앞의 도 4에서 설명한 구성과 같이 암호화 데이터에 대응시킨 노드 인덱스를 이용하여, 예를 들면

<153> 0: Enc(K(t)0, K(t)root)

<154> 00: Enc(K(t)00, K(t)0)

<155> 000: Enc(K((t)000, K(T)00)

<156> …

<157> 와 같은 데이터 구성으로 하는 것도 가능하지만, 이러한 인덱스를 이용한 구성으로 하면 중복된 데이터가 되어 데이터량이 증대하며, 네트워크를 통하는 배신 등에서는 바람직하지 못하다. 이것에 대해, 상술된 태그를 키 위치를 나타내는 색인 데이터로서 이용함으로써, 적은 데이터량으로 키 위치의 판별이 가능하게 된다.

<158> 도 6으로 되돌아가, EKB 포맷에 대해 더욱 설명한다. 서명(Signature)은 유효화 키 블록(EKB)을 발행한 예를 들면 키 관리 센터, 콘텐츠 로바이더, 결제 기관 등이 실행하는 전자 서명이다. EKB를 수령한 디바이스는 서명 검증에 의해 정당한 유효화 키 블록(EKB) 발행자가 발행한 유효화 키 블록(EKB)인 것을 확인한다.

<159> [EKB를 사용한 콘텐츠 키 및 콘텐츠의 배신]

<160> 상술된 예에서는 콘텐츠 키만을 EKB와 함께 송부하는 예에 대해 설명했지만, 콘텐츠 키로 암호화한 콘텐츠와, 콘텐츠 키 암호 키로 암호화한 콘텐츠 키와, EKB에 의해 암호화한 콘텐츠 키 암호 키를 더불어 송부하는 구성에 대해 이하에 설명한다.

<161> 도 8에는 이 데이터 구성이 도시되어 있다. 도 8의 (a)에 도시된 구성에 있어서, Enc(Kcon, content: 801)는 콘텐츠(Content)를 콘텐츠 키(Kcon)로 암호화한 데이터이고, Enc(KEK, Kcon: 802)는 콘텐츠 키(Kcon)를 콘텐츠 키 암호 키(KEK: Key Encryption Key)로 암호화한 데이터이고, Enc(EKB, KEK: 803)는 콘텐츠 키 암호 키 KEK를 유효화 키 블록(EKB)에 의해 암호화한 데이터인 것을 나타낸다.

<162> 여기서, 콘텐츠 키 암호 키 KEK는 도 3에서 도시된 노드 키(K000, K00…), 또는 루트 키(KR) 자체라도 좋고, 또한 노드 키(K000, K00…), 또는 루트 키(KR)에 의해 암호화된 키라도 좋다.

<163> 도 8의 (b)에는 복수의 콘텐츠가 미디어에 기록되고, 각각이 동일한 Enc(EKB, KEK: 805)를 이용하는 경우의 구성예가 도시되어 있다. 이러한 구성에서는 각 데이터에 동일한 Enc(EKB, KEK)를 부가하지 않고, Enc(EKB, KEK)에 링크하는 링크처를 나타내는 데이터를 각 데이터에 부가하는 구성으로 할 수 있다.

<164> 도 9에는 콘텐츠 키 암호 키 KEK가, 도 3에 도시된 노드 키 K00을 갱신한 갱신 노드 키 K(t)00으로서 구성한 경우의 예가 도시되어 있다. 이 경우, 도 3의 점선 프레임으로 된 그룹에 있어서 디바이스 3이 예를 들면 키의 누설에 의해 리보크(배제)되고 있다고 해도, 다른 그룹의 멤버 즉, 디바이스 0, 1, 2에 대해 도 9에 도시된 (a) 유효화 키 블록(EKB)과, (b) 콘텐츠 키(Kcon)를 콘텐츠 키 암호 키(KEK = K(t)00)로 암호화한 데이터와, (c) 콘텐츠(content)를 콘텐츠 키(Kcon)로 암호화한 데이터를 배신함으로써, 디바이스 0, 1, 2는 콘텐츠를 얻을 수 있다.

<165> 도 9의 우측에는 디바이스 0에서의 복호 순서를 나타내고 있다. 디바이스 0은 우선 수령한 유효화 키 블록으로부터 자신이 보유하는 리프 키 K000을 이용한 복호 처리에 의해 콘텐츠 키 암호 키(KEK = K(t)00)를 취득한다. 이어서, K(t)00에 의한 복호에 의해 콘텐츠 키 Kcon을 취득하고, 또한 콘텐츠 키 Kcon에 의해 콘텐츠의 복호를 행한다. 이들의 처리에 의해, 디바이스 0은 콘텐츠를 이용 가능하게 된다. 디바이스 1, 2에서도 각각 다른 처리 수순으로 EKB를 처리함으로써, 콘텐츠 키 암호 키(KEK = K(t)00)를 취득하는 것이 가능하게 되고, 마찬가지로 콘텐츠를 이용하는 것이 가능하게 된다.

<166> 도 3에 도시된 다른 그룹의 디바이스 4, 5, 6…은 이 동일한 데이터(EKB)를 수신했다고 해도 자신이 보유하는 리프 키, 노드 키를 이용하여 콘텐츠 키 암호 키(KEK = K(t)00)를 취득할 수 없다. 마찬가지로 리보크된 디바이스 3에 있어서도 자신이 보유하는 리프 키, 노드 키에서는 콘텐츠 키 암호 키(KEK = K(t)00)를 취득할 수 없어, 정당한 권리를 갖는 디바이스만이 콘텐츠를 복호하여 이용하는 것이 가능하게 된다.

<167> 이와 같이, EKB를 이용한 콘텐츠 키의 배송을 이용하면, 데이터량을 적게 하면서, 또한 안전하게 정당 권리자만이 복호 가능하게 한 암호화 콘텐츠를 배신하는 것이 가능하게 된다.

<168> 또, 유효화 키 블록(EKB), 콘텐츠 키, 암호화 콘텐츠 등은 네트워크를 통해 안전하게 배신하는 것이 가능한 구성이지만, 유효화 키 블록(EKB), 콘텐츠 키, 암호화 콘텐츠를 DVD, CD 등의 기록 매체에 저장하여 사용자에게 제공하는 것도 가능하다. 이 경우, 기록 매체에 저장된 암호화 콘텐츠의 복호에는 동일한 기록 매체에 저장된

유효화 키 블록(EKB)의 복호에 의해 얻어지는 콘텐츠 키를 사용하도록 구성하면, 사전에 정당 권리자만이 보유하는 리프 키, 노드 키에 의해서만 이용 가능한 암호화 콘텐츠의 배포 처리 즉, 이용 가능한 사용자 디바이스를 한정된 콘텐츠 배포가 간이한 구성으로 실현 가능하게 된다.

<169> 도 10에는 기록 매체에 암호화 콘텐츠와 함께 유효화 키 블록(EKB)을 저장한 구성예가 도시되어 있다. 도 10에 도시된 예에서는 기록 매체에 콘텐츠 C1~C4가 저장되며, 또한 각 저장 콘텐츠에 대응하는 유효화 키 블록(EKB)을 대응시킨 데이터가 저장되고, 또한 버전 M의 유효화 키 블록(EKB_M)이 저장되어 있다. 예를 들면 EKB_1은 콘텐츠 C1을 암호화한 콘텐츠 키 Kcon1을 생성하는데 사용되며, 예를 들면 EKB_2는 콘텐츠 C2를 암호화한 콘텐츠 키 Kcon2를 생성하는데 사용된다. 이 예에서는, 버전 M의 유효화 키 블록(EKB_M)이 기록 매체에 저장되어 있고, 콘텐츠 C3, C4는 유효화 키 블록(EKB-M)에 대응되므로, 유효화 키 블록(EKB_M)의 복호에 의해 콘텐츠 C3, C4의 콘텐츠 키를 취득할 수 있다. EKB_1, EKB_2는 디스크에 저장되지 않으므로, 새로운 제공 수단, 예를 들면 네트워크 배신, 또는 기록 매체에 의한 배신에 의해 각각의 콘텐츠 키를 복호하기 위해 필요한 EKB_1, EKB_2를 취득하는 것이 필요하다.

<170> [계층 트리 구조의 카테고리 분류]

<171> 암호 키를 루트 키, 노드 키, 리프 키 등, 도 3의 계층 트리 구조로서 구성하고, 콘텐츠 키, 인증 키, ICV 생성 키, 또는 프로그램 코드, 데이터 등을 유효화 키 블록(EKB)과 함께 암호화하여 배신하는 구성에 대해 설명했지만, 노드 키 등을 정의하고 있는 계층 트리 구조를 각 디바이스의 카테고리마다 분류하여 효율적인 키 갱신 처리, 암호화 키 배신, 데이터 배신을 실행하는 구성에 대하여 이하 설명한다.

<172> 도 11에는 계층 트리 구조의 카테고리의 분류의 일례가 도시되어 있다. 도 11에서, 계층 트리 구조의 최상단에는 루트 키 Kroot1101이 설정되며, 이하의 중간단에는 노드 키(1102)가 설정되며, 최하단에는 리프 키(1103)가 설정된다. 각 디바이스는 개개의 리프 키와, 리프 키로부터 루트 키에 이르는 일련의 노드 키, 루트 키를 보유한다.

<173> 여기서, 일례로서 최상단으로부터 제M 단째의 어느 한 노드를 카테고리 노드(1104)로서 설정한다. 즉, 제M 단째의 노드의 각각을 특정 카테고리의 디바이스 설정 노드로 한다. 제M 단의 하나의 노드를 정점으로서 이하, M+1단 이하의 노드, 리프는 그 카테고리에 포함되는 디바이스에 관한 노드 및 리프로 한다.

<174> 예를 들면, 도 11의 제M 단째의 하나의 노드(1105)에는 카테고리 [메모리 스틱(상표)]이 설정되며, 이 노드 이하에 연속해 있는 노드, 리프는 메모리 스틱을 사용한 여러가지 디바이스를 포함하는 카테고리 전용의 노드 또는 리프로서 설정된다. 즉, 노드(1105) 이하를 메모리 스틱의 카테고리로 정의되는 디바이스의 관련 노드, 및 리프의 집합으로서 정의한다.

<175> 또한, M단으로부터 수단분 하위의 단을 서브카테고리 노드(1106)로서 설정할 수 있다. 예를 들면, 도면에 도시된 바와 같이 카테고리 [메모리 스틱] 노드(1105)의 2단 아래의 노드에 메모리 스틱을 사용한 디바이스의 카테고리에 포함되는 서브카테고리 노드로 하여, [재생 전용기]의 노드를 설정한다. 또한, 서브카테고리 노드인 재생 전용기의 노드(1106) 이하에 재생 전용기의 카테고리에 포함되는 음악 재생 기능이 부가된 전화의 노드(1107)가 설정되며, 또한 그 하위에 음악 재생 기능이 부가된 전화의 카테고리에 포함되는 [PHS] 노드(1108)와 [휴대 전화] 노드(1109)를 설정할 수 있다.

<176> 또한, 카테고리, 서브카테고리는 디바이스의 종류뿐만 아니라, 예를 들면 어느 한 메이커, 콘텐츠 프로바이더, 결제 기관 등이 독자적으로 관리하는 노드 즉, 처리 단위, 관할 단위, 또는 제공 서비스 단위 등, 임의의 단위(이들을 총칭하여 이하, 엔티티라고 함)로 설정하는 것이 가능하다. 예를 들면 하나의 카테고리 노드를 게임 기기 메이커가 판매하는 게임 기기 XYZ 전용의 정점 노드로서 설정하면, 메이커가 판매하는 게임 기기 XYZ에 그 정점 노드 이하의 하단의 노드 키, 리프 키를 저장하여 판매하는 것이 가능하게 되고, 그 후 암호화 콘텐츠의 배신, 또는 각종 키의 배신, 갱신 처리를 그 정점 노드 키 이하의 노드 키, 리프 키에 의해 구성되는 유효화 키 블록(EKB)을 생성하여 배신하고, 정점 노드 이하의 디바이스에 대해서만 이용 가능한 데이터가 배신 가능하게 된다.

<177> 이와 같이, 하나의 노드를 정점으로 하여, 이하의 노드를 그 정점 노드로 정의된 카테고리, 또는 서브카테고리의 관련 노드로서 설정하는 구성으로 함으로써, 카테고리단, 또는 서브카테고리단의 하나의 정점 노드를 관리하는 메이커, 콘텐츠 프로바이더 등이 그 노드를 정점으로 하는 유효화 키 블록(EKB)을 독자적으로 생성하여, 정점 노드 이하에 속하는 디바이스에 배신하는 구성이 가능하게 되고, 정점 노드에 속하지 않은 다른 카테고리의 노드에 속하는 디바이스에는 전혀 영향을 미치지 않고 키 갱신을 실행할 수 있다.

<178> [간략 EKB에 의한 키 배신 구성]

<179> 먼저 설명한 예를 들면 도 3의 트리 구성에 있어서, 키, 예를 들면 콘텐츠 키를 소정 디바이스(리프)앞으로 송부하는 경우, 키 배포처 디바이스가 소유하고 있는 리프 키, 노드 키를 이용하여 복호 가능한 유효화 키 블록(EKB)을 생성하여 제공한다. 예를 들면 도 12의 (a)에 도시된 트리 구성에 있어서, 리프를 포함하는 디바이스 a, g, j에 대해 키, 예를 들면 콘텐츠 키를 송신하는 경우, a, g, j의 각 노드에 있어서 복호 가능한 유효화 키 블록(EKB)을 생성하여 배신한다.

<180> 예를 들면 갱신 루트 키 K(t)root에서 콘텐츠 키 K(t)con을 암호화 처리하고, EKB와 함께 배신하는 경우를 생각한다. 이 경우, 디바이스 a, g, j는 각각이 도 12의 (b)에 도시된 리프 및 노드 키를 이용하여 EKB의 처리를 실행하여 K(t)root를 취득하고, 취득한 갱신 루트 키 K(t)root에 의해 콘텐츠 키 K(t)con의 복호 처리를 실행하여 콘텐츠 키를 얻는다.

<181> 이 경우에 제공되는 유효화 키 블록(EKB)의 구성은 도 13에 도시된 바와 같다. 도 13에 도시된 유효화 키 블록(EKB)은 앞의 도 6에서 설명한 유효화 키 블록(EKB)의 포맷에 따라 구성된 것으로, 데이터(암호화 키)와 대응하는 태그를 구비한다. 태그는 먼저 도 7을 이용하여 설명한 바와 같이 좌측(L), 우측(R), 각각의 방향으로 데이터가 있으면 0, 없으면 1을 나타내고 있다.

<182> 유효화 키 블록(EKB)을 수령한 디바이스는 유효화 키 블록(EKB)의 암호화 키와 태그에 기초하여 순차 암호화 키의 복호 처리를 실행하여 상위 노드의 갱신 키를 취득해 간다. 도 13에 도시된 바와 같이 유효화 키 블록(EKB)은 루트로부터 리프까지의 단 수(깊이)가 많을수록 그 데이터량은 증가한다. 단 수(깊이)는 디바이스(리프)수에 따라 증대하는 것으로, 키의 전송처가 되는 디바이스 수가 많은 경우에는 EKB의 데이터량이 더욱 증대하게 된다.

<183> 이러한 유효화 키 블록(EKB)의 데이터량의 삭감을 가능하게 한 구성에 대해 설명한다. 도 14는 유효화 키 블록(EKB)을 키 배신 디바이스에 따라 간략화하여 구성한 예를 도시한다.

<184> 도 13과 마찬가지로, 리프를 포함하는 디바이스 a, g, j에 대해 키, 예를 들면 콘텐츠 키를 송신하는 경우를 상정한다. 도 14의 (a)에 도시된 바와 같이 키 배신 디바이스에 의해서만 구성되는 트리를 구축한다. 이 경우, 도 12의 (b)에 도시된 구성에 기초하여 새로운 트리 구성으로서 도 14의 (b)의 트리 구성이 구축된다. Kroot로부터 Kj까지는 전혀 분기가 없어 하나의 브랜치만이 존재하면 되고, Kroot로부터 Ka 및 Kg에 이르기 위해서는 K0에 분기점을 구성할 뿐으로, 2 분기 구성의 도 14의 (a)의 트리가 구축된다.

<185> 도 14의 (a)에 도시된 바와 같이 노드로서 K0만을 구비한 간략화한 트리가 생성된다. 갱신 키 배신을 위한 유효화 키 블록(EKB)은 이들의 간략 트리에 기초하여 생성한다. 도 14의 (a)에 도시된 트리는 유효화 키 블록(EKB)을 복호 가능한 말단 노드 또는 리프를 최하단으로 한 2 분기형 트리를 포함하는 패스를 선택하여 불필요한 노드를 생략함으로써 재구축되는 재구축 계층 트리이다. 갱신 키 배신을 위한 유효화 키 블록(EKB)은 이 재구축 계층 트리의 노드 또는 리프에 대응하는 키에만 기초하여 구성된다.

<186> 앞의 도 13에서 설명한 유효화 키 블록(EKB)은 각 리프 a, g, j로부터 Kroot에 이르기까지의 모든 키를 암호화한 데이터를 저장했었지만, 간략화 EKB는 간략화한 트리를 포함하는 노드에 대한 것만의 암호화 데이터를 저장한다. 도 14의 (b)에 도시된 바와 같이 태그는 3 비트 구성을 구비한다. 제1 및 제2 비트는 도 13의 예와, 동일한 의미를 갖고, 좌측(L), 우측(R), 각각의 방향으로 데이터가 있으면 0, 없으면 1을 나타낸다. 제3번째의 비트는 EKB 내에 암호화 키가 저장되어 있는지의 여부를 나타내기 위한 비트로서, 데이터가 저장되어 있는 경우에는 1, 데이터가 없는 경우에는 0으로서 설정된다.

<187> 데이터 통신망, 또는 기억 매체에 저장되어 디바이스(리프)에 제공되는 유효화 키 블록(EKB)은 도 14의 (b)에 도시된 바와 같이, 도 13에 도시된 구성에 비교하면, 데이터량이 대폭 삭감된 것이 된다. 도 14에 도시된 유효화 키 블록(EKB)을 수령한 각 디바이스는 태그의 제3 비트에 1이 저장된 부분의 데이터만을 순차 복호함으로써, 소정의 암호화 키의 복호를 실현할 수 있다. 예를 들면 디바이스 a는 암호화 데이터 Enc(Ka, K(t)0)를 리프 키 Ka로 복호하여, 노드 키 K(t)0을 취득하여, 노드 키 K(t)0에 의해 암호화 데이터 Enc(K(t)0, K(t)root)를 복호하여 K(t)root를 취득한다. 디바이스 j는 암호화 데이터 Enc(Kj, K(t)root)를 리프 키 Kj에서 복호하여, K(t)root를 취득한다.

<188> 이와 같이, 전송처의 디바이스에 의해서만 구성되는 간략화한 새로운 트리 구성을 구축하여, 구축된 트리를 포함하는 리프 및 노드의 키만을 이용하여 유효화 키 블록(EKB)을 생성함으로써, 적은 데이터량의 유효화 키 블록

(EKB)을 생성하는 것이 가능하게 되고, 유효화 키 블록(EKB)의 데이터 배신이 효율적으로 실행 가능하게 된다.

<189> 또, 간략화한 계층 트리 구성은 후단에서 설명하는 엔티티 단위의 EKB 관리 구성에 있어서 특히 유효하게 활용 가능하다. 엔티티는 키 배신 구성으로서의 트리 구성을 포함하는 노드 또는 리프로부터 선택한 복수의 노드 또는 리프의 집합체 블록이다. 엔티티는 디바이스의 종류에 따라 설정되는 집합이거나, 또는 디바이스 제공 메이커, 콘텐츠 프로바이더, 결제 기관 등의 관리 단위 등, 어떤 공통점을 갖은 처리 단위, 관할 단위, 또는 제공 서비스 단위 등, 여러 형태의 집합으로서 설정된다. 하나의 엔티티에는, 어떤 공통의 카테고리로 분류되는 디바이스가 모여 있고, 예를 들면 복수의 엔티티의 정점 노드(서브루트)에 의해 상술된 바와 같은 간략화한 트리를 재구축하여 EKB를 생성함으로써, 선택된 엔티티에 속하는 디바이스에 있어서 복호 가능한 간략화된 유효화 키 블록(EKB)의 생성, 배신이 가능하게 된다. 엔티티 단위의 관리 구성에 대해서는 후단에서 상세히 설명한다.

<190> 또, 이러한 유효화 키 블록(EKB)은 광 디스크, DVD 등의 정보 기록 매체에 저장한 구성으로 하는 것이 가능하다. 예를 들면, 상술된 암호화 키 데이터에 의해 구성되는 데이터부와, 암호화 키 데이터의 계층 트리 구조에서의 위치 식별 데이터로서의 태그부를 포함하는 유효화 키 블록(EKB)에 또한, 갱신 노드 키에 의해 암호화한 콘텐츠 등의 메시지 데이터를 저장한 정보 기록 매체를 각 디바이스에 제공하는 구성이 가능하다. 디바이스는 유효화 키 블록(EKB)에 포함되는 암호화 키 데이터를 태그부의 식별 데이터에 따라 순차 추출하여 복호하고, 콘텐츠의 복호에 필요한 키를 취득하여 콘텐츠의 이용을 행하는 것이 가능하게 된다. 물론, 유효화 키 블록(EKB)을 인터넷 등의 네트워크를 통해 배신하는 구성으로 해도 된다.

<191> [암호 처리 기능을 구비한 기억 장치와 데이터 처리 장치 사이의 데이터 이동]

<192> 이어서, 상술된 계층 트리 구성을 적용한 유효화 키 블록(EKB)에 의해 배신되는 암호 처리 키를 적용한 처리 구성에 대해 암호 처리 기능을 구비한 기억 장치, 예를 들면 메모리 스틱(상표) 등의 메모리 카드와, 데이터 재생 장치 사이에서의 데이터 이동 처리를 중심으로 하여 설명한다.

<193> 도 15는 서로 콘텐츠 데이터의 이동을 실행 가능한 재생 장치와 암호 처리 기능을 구비한 메모리 카드 등의 기억 장치의 상세 구성을 도시하는 블록도이다.

<194> 도 15에 도시된 바와 같이 기억 장치(300)는 예를 들면 주제어 모듈(31), 통신 인터페이스(32), 제어 모듈(33), 플래시 메모리(34) 및 플래시 메모리 관리 모듈(35)을 구비한다. 이하, 각 모듈에 대해 설명한다.

<195> [제어 모듈(33)]

<196> 도 15에 도시된 바와 같이 제어 모듈(33)은 예를 들면 난수 발생 유닛(50), 기억 유닛(51), 키 생성/연산 유닛(52), 상호 인증 유닛(53), 암호화/복호 유닛(54) 및 제어 유닛(55)을 구비한다. 제어 모듈(33)은 싱글 칩의 암호 처리 전용의 집적 회로로서, 다층 구조를 구비하고, 내부의 메모리 셀은 알루미늄층 등의 더미층에 삽입되어 있다. 또한, 제어 모듈(33)은 동작 전압 또는 동작 주파수의 폭이 좁고, 외부로부터 부정하게 데이터를 판독할 수 없도록 내탬퍼성을 갖고 있다. 난수 발생 유닛(50)은 난수 발생 지시를 받으면, 64 비트(8 바이트)의 난수를 발생시킨다.

<197> 기억 유닛(51)은, 예를 들면 EEPROM(Electrically Erasable Programmable Read Only Memory) 등의 불휘발성 메모리로서, 인증 처리에 필요한 키 데이터 등의 여러 데이터를 기억하고 있다. 도 16은, 기억 유닛(51)에 기억되어 있는 데이터를 설명하기 위한 도면이다. 도 16에 도시된 바와 같이 기억 유닛(51)은 인증 키 데이터 IK0～IK31, 장치 식별 데이터 IDm 및 기억용 키 데이터 Kstm을 기억하고 있다.

<198> 인증 키 데이터 IK0～IK31은 기억 장치(300)가 재생 장치(200) 사이에서 상호 인증을 행할 때에 이용되는 키 데이터로서, 후술된 바와 같이 상호 인증을 행할 때마다 인증 키 데이터 IK0～IK31 중 하나의 인증 키 데이터가 랜덤하게 선택된다. 또, 인증 키 데이터 IK0～IK31 및 기억용 키 데이터 Kstm은 기억 장치(300)의 외부로부터 판독되지 않게 되어 있다. 장치 식별 데이터 IDm은 기억 장치(300)에 대해 고유하게 붙여진 식별 데이터로서, 후술된 바와 같이 기억 장치(300)가 재생 장치(200) 사이에서 상호 인증을 행할 때에 판독되어 재생 장치(200)로 출력된다. 기억용 키 데이터 Kstm은 후술된 바와 같이 콘텐츠의 암호화에 이용되는 콘텐츠 키 데이터 CK를 암호화하여 플래시 메모리(34)에 기억할 때에 이용된다.

<199> 키 생성/연산 유닛(52)은, 예를 들면 ISO/IEC9797의 MAC(Message Authentication Code) 연산 등의 여러가지의 연산을 행하여 키 데이터를 생성한다. 이 때, MAC 연산에는, 예를 들면 "Block cipher Algorithm"로서

FIPSPUB46-2로 규정되는 DES(Data Encryption Standard)가 이용된다. MAC 연산은, 임의의 길이의 데이터를 고정의 길이로 압축하는 일방향성 해시 함수 연산이고, 함수치가 비밀 키에 의존하여 정해진다.

<200> 상호 인증 유닛(53)은, 재생 장치(200)로부터 오디오 데이터를 입력하여 플래시 메모리(34)에 기입하는 동작을 행하는 데 앞서서, 재생 장치(200) 사이에서 상호 인증 처리를 행한다. 또한, 상호 인증 유닛(53)은 플래시 메모리(34)로부터 오디오 데이터를 판독하여 재생 장치(200)로 출력하는 동작을 행하는 데 앞서서, 재생 장치(200) 사이에서 상호 인증 처리를 행한다. 또한, 상호 인증 유닛(53)은 상호 인증 처리에 있어서, 상술한 MAC 연산을 행한다. 상기 상호 인증 처리에서는 기억 유닛(51)에 기억되어 있는 데이터가 이용된다.

<201> 암호화/복호 유닛(54)은 DES, IDEA, MISTY 등의 블록 암호 알고리즘에서의 암호화를 행한다. 사용하는 모드는 FIPS PUB81 "DES MODES OF OPERATION"으로 규정되어 있는 ECB(Electronic Code Book) 모드 및 CBC(Cipher Block Chaining) 모드이다. 또한, 암호화/복호 유닛(54)은 DES, IDEA, MISTY 등의 블록 복호 알고리즘에서의 복호를 행한다. 사용하는 모드는 상기 ECB 모드 및 CBC 모드이다. 상기 ECB 모드 및 CBC 모드의 블록 암호화/복호에서는, 지정된 키 데이터를 이용하여 지정된 데이터를 암호화/복호한다. 제어 유닛(55)은 난수 발생 유닛(50), 기억 유닛(51), 키 생성/연산 유닛(52), 상호 인증 유닛(53) 및 암호화/복호 유닛(54)의 처리를 통괄하여 제어한다.

<202> [플래시 메모리(34)]

<203> 플래시 메모리(34)는 예를 들면 32M 바이트의 기억 용량을 갖는다. 플래시 메모리(34)에는 상호 인증 유닛(53)에 의한 재생 장치(200)와 기억 장치(300) 사이의 상호 인증 처리에 의해 쌍방이 정당한 장치라고 인정됐을 때에 재생 장치(200)로부터 입력한 오디오 데이터 또는 화상 데이터 등, 각종 데이터가 기입된다. 또한, 플래시 메모리(34)로부터는 상호 인증 유닛(53)에 의한 재생 장치(200)와 기억 장치(300) 사이의 상호 인증 처리에 의해 정당한 상대라고 인정됐을 때에 오디오 데이터, 화상 데이터 등이 판독되어 재생 장치(200)로 출력된다.

<204> 이하, 플래시 메모리(34)에 기억되는 데이터 및 그 포맷에 대해 설명한다. 도 17은 플래시 메모리(34)에 기억되는 데이터를 설명하기 위한 도면이다. 도 17에 도시된 바와 같이 플래시 메모리(34)에는 예를 들면 재생 관리 파일, 복수의 트랙 데이터(재생 데이터) 파일이 기억되어 있다. 여기서, 재생 관리 파일은 트랙 데이터 파일의 재생을 관리하는 관리 데이터를 구비하고, 트랙 데이터 파일은 각각 대응하는 트랙 데이터(오디오 데이터)를 갖고 있다. 또, 본 실시예에서는, 트랙 데이터는, 예를 들면 1곡분의 오디오 데이터를 의미한다. 이하, 플래시 메모리(34)에 기억되는 데이터를 오디오 데이터로 한 경우의 예에 대해 설명한다.

<205> 도 18에는, 재생 관리 파일의 구성이 도시되어 있고, 도 19에는 하나(1곡)의 ATRAC3 데이터 파일의 구성이 도시되어 있다. 재생 관리 파일은 16KB 고정 길이의 파일이다. ATRAC3 데이터 파일은 곡 단위로, 선두의 속성 헤더와, 그것에 계속되는 실제 암호화된 음악 데이터를 포함한다. 속성 헤더도 16KB 고정 길이가 되며, 재생 관리 파일과 유사한 구성을 구비한다.

<206> 재생 관리 파일은 헤더, 1 바이트 코드의 메모리 카드의 이름 NM1-S, 2 바이트 코드의 메모리 카드의 이름 NM2-S, 곡순의 재생 테이블 TRKTBL, 메모리 카드 전체의 부가 정보 INF-S를 포함한다. 데이터 파일 선두의 속성 헤더는 헤더, 1 바이트 코드의 곡명 NM1, 2 바이트 코드의 곡명 NM2, 트랙의 키 정보 등의 트랙 정보 TRKINF, 파트 정보 PRTINF와, 트랙의 부가 정보 INF를 포함한다. 헤더에는 총 파트 수, 이름의 속성, 부가 정보 사이즈의 정보 등이 포함된다.

<207> 속성 헤더에 대해 ATRAC3의 음악 데이터가 계속된다. 음악 데이터는 16KB의 블록마다 구획되고, 각 블록의 선두에 헤더가 부가되어 있다. 헤더에는 암호를 복호하기 위한 초기치가 포함된다. 또, 암호화의 처리를 받는 것은 ATRAC3 데이터 파일 중 음악 데이터 등의 콘텐츠 데이터뿐이며, 그 외의 재생 관리 파일, 헤더 등의 데이터는 암호화되지 않는다.

<208> 도 20에는 재생 관리 파일 PBLIST의 상세한 데이터 구성이 도시되어 있다. 재생 관리 파일 PBLIST는 1 클러스터(1 블록 = 16KB)의 사이즈이다. 도 20의 (a)에 도시된 헤더는 32 바이트로 이루어진다. 도 20의 (b)에 도시된 헤더 이외의 부분은 메모리 카드 전체에 대한 이름 NM1-S(256 바이트), 이름 NM2-S(512 바이트), 암호화된 콘텐츠 키(CONTENTSKEY), MAC, S-YMDhms와, 재생 순서를 관리하는 테이블 TRKTBL(800 바이트), 메모리 카드 전체에 대한 부가 정보 INF-S(14720 바이트) 및 마지막으로 헤더 내의 정보의 일부가 재차 기록되어 있다. 이들이 다른 종류의 데이터군 각각의 선두는 재생 관리 파일 내에서 소정의 위치가 되도록 규정되어 있다.

<209> 재생 관리 파일은 도 20의 (a)에 도시된 (0x0000) 및 (0x0010)로 표시되는 선두로부터 32 바이트가 헤더이다.

또, 파일 중에서 선두로부터 16 바이트 단위로 구획된 단위를 슬롯이라고 칭한다. 파일의 제1 및 제2 슬롯에 배치되는 헤더에는 하기의 의미, 기능, 값을 구비한 데이터가 선두로부터 순서대로 배분된다. 또, Reserved라고 표기되는 데이터는 미정의의 데이터를 나타내고 있다. 통상 널(0x00)이 쓰여지지만, 무엇이 씌어져 있어도 Reserved의 데이터가 무시된다. 장래의 버전에서는 변경이 있을 수 있다. 또한, 이 부분에의 기입은 금지한다. Option이라고 쓰여진 부분도 사용하지 않은 경우에는 전부 Reserved와 동일하게 취급된다.

<210> BLKID-TL0(4 바이트)

<211> 의미 : BLOCKID FILE ID

<212> 기능 : 재생 관리 파일의 선두인 것을 식별하기 위한 값

<213> 값 : 고정치 = "TL = 0"(예를 들면, 0x544C2D30)

<214> MCode(2 바이트)

<215> 의미 : MAKER CODE

<216> 기능 : 기록한 기기의 메이커, 모델을 식별하는 코드

<217> 값 : 상위 10 비트(메이커 코드), 하위 6 비트(기종 코드)

<218> REVISION(4 바이트)

<219> 의미 : PBLIST의 재기록 횟수

<220> 기능 : 재생 관리 파일을 재기록할 때마다 인크리먼트

<221> 값 : 0부터 시작, +1씩 증가함

<222> SN1C+L(2 바이트)

<223> 의미 : NM1-S 영역에 쓰여지는 메모리 카드의 이름(1 바이트)의 속성을 나타냄

<224> 기능 : 사용하는 문자 코드와 언어 코드를 각 1 바이트로 나타냄

<225> 값 : 문자 코드(C)는 상위 1 바이트로 하기와 같이 문자를 구별함

<226> 00 : 문자 코드는 설정하지 않음. 단순한 2진수로서 취급하는 것

<227> 01 : ASCII(American Standard Code for Information Interchange)

<228> 02 : ASCII+KANA 03 : modifided8859-1

<229> 81 : MS-JIS 82 : KS C 5601-1989 83 : GB(Great Britain)2312-80

<230> 90 : S-JIS(Japanese Industrial Standards)(for Voice)

<231> 언어 코드(L)는 하위 1 바이트로 하기와 같이 EBU Tech 3258 규정에 준하여 언어를 구별함

<232> 00 : 설정하지 않음 08 : German 09 : English 0A : Spanish

<233> 0F : French 15 : Italian 1D : Dutch

<234> 65 : Korean 69 : Japanese 75 : Chinese

<235> 데이터가 없는 경우 올 제로로 하는 것

<236> SN2C+L(2 바이트)

<237> 의미 : NM2-S 영역에 쓰여진 메모리 카드의 이름(2 바이트)의 속성을 나타냄

<238> 기능 : 사용하는 문자 코드와 언어 코드를 각 1 바이트로 나타냄

<239> 값 : 상술된 SN1C+L과 동일

<240> SINFSIZE(2 바이트)

<241> 의미 : INF-S 영역에 쓰여진 메모리 카드 전체에 관한 부가 정보의 모두를 합계한 사이즈를 나타냄

<242> 기능 : 데이터 사이즈를 16 바이트 단위의 크기로 기술, 없는 경우에는 반드시 올 제로로 하는 것

<243> 값 : 사이즈는 0x0001로부터 0x39C(924)

<244> T-TRK(2 바이트)

<245> 의미 : TOTAL, TRACK NUMBER

<246> 기능 : 총 트랙 수

<247> 값 : 1로부터 0x0190(최대 400 트랙), 데이터가 없는 경우에는 올 제로로 하는 것

<248> VerNo(2 바이트)

<249> 의미 : 포맷의 버전 번호

<250> 기능 : 상위가 메이저 버전 번호, 하위가 마이너 버전 번호. 저작권 대응형인지의 여부 즉, 상술된 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의한 배신 키의 사용 대상인지의 여부를 나타내는 데이터로서도 사용됨

<251> 값 : 예 0x0100(Ver1.0)

<252> 0x0203(Ver2.3)

<253> 상술된 헤더에 계속되는 영역에 쓰여진 데이터(도 20의 (b))에 대해 이하에 설명한다.

<254> NM1-S

<255> 의미 : 메모리 카드 전체에 관한 1 바이트의 이름

<256> 기능 : 1 바이트의 문자 코드로 나타낸 가변 길이의 이름 데이터(최대 256), 이름 데이터의 종료는 반드시 종단 코드(0x00)를 기입하는 것

<257> 사이즈는 이 종단 코드로부터 계산하는 것, 데이터가 없는 경우에는 적어도 선두(0x0020)로부터 널(0x00)을 1 바이트 이상 기록하는 것

<258> 값 : 각종 문자 코드

<259> NM2-S

<260> 의미 : 메모리 카드 전체에 관한 2 바이트의 이름

<261> 기능 : 2 바이트의 문자 코드로 나타낸 가변 길이의 이름 데이터(최대 512), 이름 데이터의 종료는 반드시 종단 코드(0x00)를 기입하는 것

<262> 사이즈는 이 종단 코드로부터 계산하는 것, 데이터가 없는 경우에는 적어도 선두(0x0120)로부터 널(0x00)을 2 바이트 이상 기록하는 것

<263> 값 : 각종 문자 코드

<264> EKB_version(4 바이트)

<265> 의미 : 상술된 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의해 제공되는 콘텐츠 키의 세대 번호, 및/또는 유효화 키 블록(EKB)의 파일명을 나타냄

<266> 기능 : 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의해 제공되는 콘텐츠 키를 구하기 위한 유효화 키 블록(EKB)을 나타냄

<267> 값 : 0으로부터 0xFF까지

<268> E(Kstm, Kcon)(8 바이트)

<269> 의미 : 콘텐츠마다의 암호 처리용의 키인 콘텐츠 키를 메모리 카드의 스토리지 키(Kstm)로 암호화한 데이터

<270> 기능 : 콘텐츠의 암호 처리에 사용됨

<271> 값 : 0으로부터 0xFFFFFFFFFFFFFFFF까지

<272> E(KEKn, Kcon)(8 바이트)

<273> 의미 : 콘텐츠마다의 암호 처리용의 키인 콘텐츠 키를 상술된 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의해 제공되는 키 암호화 키 KEKn에 의해 암호화한 데이터

<274> 기능 : 콘텐츠의 암호 처리에 사용됨

<275> 값 : 0으로부터 0xFFFFFFFFFFFFFFFF까지

<276> C_MAC[0](8 바이트)

<277> 의미 : 저작권 정보 개찬 체크치

<278> 기능 : 재생 관리 파일 내의 데이터, 최종 콘텐츠 기록 등의 콘텐츠 처리 일시를 나타내는 S-YMDhms 다른 데이터에 기초하여 생성되는 개찬 체크용의 값. 일시 데이터 S-YMDhms가 개찬되어 있는 경우에는 C_MAC[0]의 체크시에 개찬이 있었다고 판정되며, 콘텐츠의 재생이 실행되지 않음

<279> 값 : 0으로부터 0xFFFFFFFFFFFFFFFF까지

<280> MGR

<281> 의미 : 콘텐츠 키의 종류

<282> 기능 : 0x00에서 콘텐츠 키 Kcon과, E(KEKn, Kcon)의 양방이 있으며, 0x01에서, E(KEKn, Kcon)만 있음

<283> 값 : 0으로부터 0x01까지

<284> S-YMDhms(4 바이트)(Option)

<285> 의미 : 신뢰할 수 있는 시계를 구비한 기기로 기록한 년 ·월 ·일 ·시간 ·분 ·초

<286> 기능 : 콘텐츠의 최종 기록 일시 등, 콘텐츠 최종 처리 일시를 식별하기 위한 값. 콘텐츠의 처리 시에 갱신됨

<287> 값 : 25-31 비트 년 0-99(1980-2079)

<288> 21-24 비트 월 0-12

<289> 16-20 비트 일 0-31

<290> 11-15 비트 시간 0-23

<291> 05-10 비트 분 0-59

<292> 00-04 비트 초 0-29(2초 단위)

<293> 또, S-YMDhms는 콘텐츠 기록 시 등의 콘텐츠 처리 시에 갱신되며, 갱신된 데이터에 기초하여 상술된 C-MAC[0]도 갱신되어 저장된다.

<294> TRK-nnn

<295> 의미 : 재생하는 ATRAC3 데이터 파일의 SQN(시퀀스) 번호

<296> 기능 : TRKINF 중 FNo를 기술함

<297> 값 : 1로부터 400(0x190)

<298> 트랙이 존재하지 않을 때에는 올 제로로 하는 것

<299> INF-S

<300> 의미 : 메모리 카드 전체에 관한 부가 정보 데이터(예를 들면, 사진, 가사, 해설 등의 정보)

<301> 기능 : 헤더를 따른 가변 길이의 부가 정보 데이터

<302> 복수의 다른 부가 정보가 배열되어지는 경우가 있다. 각각에 ID와 데이터 사이즈가 붙어진다. 개개의 헤더를 포함하는 부가 정보 데이터는 최소 16 바이트 이상으로 4 바이트의 정수배 단위로 구성된다. 그 상세한 내용에 대해서는 후술한다.

<303> 값 : 부가 정보 데이터 구성을 참조

<304> 재생 관리 파일의 최후의 슬롯으로서, 헤더 내의 것과 동일한 BLKID-TLO과, MCode와, REVISION이 쓰여진다.

<305> 민간용 오디오 기기로서, 메모리 카드가 기록중에 빠지거나, 전원이 끊어지는 경우가 있어, 부활했을 때에 이들의 이상 발생을 검출하는 것이 필요하게 된다. 상술된 바와 같이, REVISION을 블록의 선두와 말미에 기입하고, 이 값을 재기입할 때마다 +1 인크리먼트하도록 하고 있다. 만약, 블록 도중에 이상 종료가 발생하면, 선두와 말미의 REVISION의 값이 일치하지 않고, 이상 종료를 검출할 수 있다. REVISION이 2개 존재하므로, 높은 확률로 이상 종료를 검출할 수 있다. 이상 종료의 검출시에는 에러 메시지의 표시 등의 경고가 발생한다.

<306> 또한, 1 블록(16KB)의 선두 부분에 고정치 BLKID-TLO을 삽입하고 있으므로, FAT가 깨진 경우의 수복의 목표에 고정치를 사용할 수 있다. 즉, 각 블록의 선두의 고정치를 보면, 파일의 종류를 판별하는 것이 가능하다. 또한, 이 고정치 BLKID-TLO은, 블록의 헤더 및 블록의 종단 부분에 2중으로 기술하므로, 그 신뢰성의 체크를 행할 수 있다. 또, 재생 관리 파일 PBLIST의 동일한 것을 2중으로 기록해도 좋다.

<307> ATRAC3 데이터 파일은 트랙 정보 관리 파일과 비교하여, 상당히 큰 데이터량이고, ATRAC3 데이터 파일에 관해서는 블록 번호 BLOCK SERIAL이 붙어진다. 단, ATRAC3 데이터 파일은 통상 복수의 파일이 메모리 카드 상에 존재하므로, CONNUMO으로 콘텐츠의 구별을 한 후에 BLOCK SERIAL을 붙이지 않으면, 중복이 발생하여 FAT가 깨진 경우의 파일의 복구가 곤란하게 된다. 환언하면 단일의 ATRAC3 데이터 파일은 복수의 BLOCK으로 구성됨과 함께, 이산하여 배치될 가능성이 있기 때문에, 동일 ATRAC3 데이터 파일을 포함하는 BLOCK을 판별하기 위해 CONNUMO을 이용함과 함께, 동일 ATRAC3 데이터 파일 내의 승강순을 블록 번호 BLOCK SERIAL로 결정한다.

<308> 마찬가지로, FAT의 파괴까지 이르지는 않지만, 논리를 잘못하여 파일로 하여 문제점이 있는 경우에 기입한 메이커의 기종을 특정할 수 있도록 메이커 코드(MCode)가 블록의 선두와 말미에 기록되어 있다.

<309> 도 20의 (c)에는 부가 정보 데이터의 구성이 도시되어 있다. 부가 정보의 선두에 하기의 헤더가 쓰여진다. 헤더 이후에 가변 길이의 데이터가 쓰여진다.

<310> INF

<311> 의미 : FIELD ID

<312> 기능 : 부가 정보 데이터의 선두를 나타내는 고정치

<313> 값 : 0x69

<314> ID

<315> 의미 : 부가 정보 키 코드

<316> 기능 : 부가 정보의 분류를 나타냄

<317> 값 : 0으로부터 0xFF

<318> SIZE

<319> 의미 : 개별 부가 정보의 크기

<320> 기능 : 데이터 사이즈는 자유롭지만, 반드시 4 바이트의 정수배가 아니면 안된다. 또한, 최소 16 바이트 이상의 것. 데이터의 끝에서 나머지가 나오는 경우에는 널(0x00)로 매립해 둘 것

<321> 값 : 16으로부터 14784(0x39C0)

<322> MCode

<323> 의미 : MAKER CODE

<324> 기능 : 기록한 기기의 메이커, 모델을 식별하는 코드

<325> 값 : 상위 10 비트(메이커 코드), 하위 6 비트(기종 코드)

<326> C+L

<327> 의미 : 선두에서 12 바이트째로부터의 데이터 영역에 쓰여진 문자의 속성을 나타냄

<328> 기능 : 사용하는 문자 코드와 언어 코드를 각 1 바이트로 나타냄

<329> 값 : 상술된 SNC+L과 동일함

<330> DATA

<331> 의미 : 개별의 부가 정보 데이터

<332> 기능 : 가변 길이 데이터로 나타냄. 실제 데이터의 선두는 항상 12 바이트째에서 시작. 길이(사이즈)는 최소 4 바이트 이상. 항상 4 바이트의 정수배가 아니면 안됨. 마지막 데이터로부터 나머지가 있는 경우에는 널(0x00)로 매립하는 것

<333> 값 : 내용에 따라 개별로 정의됨

<334> 도 21에는 ATRAC3 데이터 파일 A3Dnnnn의 데이터 배열예가 도시되어 있다. 도 21에는 데이터 파일의 속성 헤더(1 블록)와, 음악 데이터 파일(1 블록)이 도시되어 있다. 도 21에는, 이 2블록(16 ×2 = 32K 바이트)의 각 슬롯의 선두 바이트(0x0000~0x7FF0)가 도시되어 있다. 도 22에 분리하여 도시된 바와 같이, 속성 헤더의 선두에서 32 바이트가 헤더이고, 256 바이트가 곡명 영역 NM1(256 바이트)이고, 512 바이트가 곡명 영역 NM2(512 바이트)이다. 속성 헤더의 헤더에는 하기의 데이터가 쓰여진다.

<335> BLKID-HD0(4 바이트)

<336> 의미 : BLOCKID FILE ID

<337> 기능 : ATRAC3 데이터 파일의 선두인 것을 식별하기 위한 값

<338> 값 : 고정치 = "HD = 0"(예를 들면, 0x48442D30)

<339> MCode(2 바이트)

<340> 의미 : MAKER CODE

<341> 기능 : 기록한 기기의 메이커, 모델을 식별하는 코드

<342> 값 : 상위 10 비트(메이커 코드), 하위 6 비트(기종 코드)

<343> BLOCK SERIAL(4 바이트)

<344> 의미 : 트랙마다 붙여진 연속 번호

<345> 기능 : 블록의 선두는 0에서 시작. 다음 블록은 +1씩 인크리먼트 편집되어도 값을 변화시키지 않음

<346> 값 : 0에서 시작 0xFFFFFFFF까지

<347> N1C+L(2 바이트)

<348> 의미 : 트랙(곡명) 데이터(NM1)의 속성을 나타냄

<349> 기능 : NM1에 사용되는 문자 코드와 언어 코드를 각 1 바이트로 나타냄

<350> 값 : SN1C+L과 동일

<351> N2C+L(2 바이트)

<352> 의미 : 트랙(곡명) 데이터(NM2)의 속성을 나타냄

<353> 기능 : NM2에 사용되는 문자 코드와 언어 코드를 각 1 바이트로 나타냄

<354> 값 : SN1C+L과 동일

<355> INFSIZE(2 바이트)

<356> 의미 : 트랙에 관한 모든 부가 정보를 합계한 사이즈를 나타냄

<357> 기능 : 데이터 사이즈를 16 바이트 단위의 크기로 기술. 없는 경우에는 반드시 올 제로로 하는 것

<358> 값 : 사이즈는 0x0000로부터 0x3C6(966)

<359> T-PRT(2 바이트)

<360> 의미 : 토탈 파트 수

<361> 기능 : 트랙을 포함하는 파트 수를 나타냄. 통상은 1

<362> 값 : 1로부터 0x285(645dec)

<363> T-SU(4 바이트)

<364> 의미 : 토탈 SU(사운드 유닛) 수, SU는 파트의 최소 단위이고, 또한 ATRAC3으로 오디오 데이터를 압축할 때의 최소의 데이터 단위임. 44.1㎑의 샘플링 주파수로 얻어진 1024 샘플분(1024 ×16 비트 ×2 채널)의 오디오 데이터를 약 1/10로 압축한 수백 바이트의 데이터가 SU임. 1SU는 시간으로 환산하여 약 23m초가 됨. 통상은 수천에 미치는 SU에 의해 하나의 파트가 구성됨. 1 클러스터가 42개의 SU로 구성되는 경우, 1 클러스터로 약 1초의 소리를 나타낼 수 있음. 하나의 트랙을 포함하는 파트의 수는 부가 정보 사이즈에 영향을 받음. 파트 수는 1 블록 중에서 헤더나 곡명, 부가 정보 데이터 등을 제외한 수로 결정되기 때문에, 부가 정보가 전혀 없는 상태가 최대 수(645개)의 파트를 사용할 수 있는 조건이 됨.

<365> 기능 : 1 트랙 중 실제의 총 SU 수를 나타냄. 곡의 연주 시간에 상당함

<366> 값 : 0x01로부터 0x001FFFFF

<367> INX(2 바이트)(Option)

<368> 의미 : INDEX의 상대 장소

<369> 기능 : 곡의 클라이막스 부분(특징적인 부분)의 선두를 나타내는 포인터. 곡의 선두에서의 위치를 SU의 개수를 1/4 한 수로 지정함. 이것은, 통상의 SU의 4배 길이의 시간(약 93ms)에 상당함

<370> 값 : 0으로부터 0xFFFF(최대, 약 6084초)

<371> XT(2 바이트)(Option)

<372> 의미 : INDEX의 재생 시간

<373> 기능 : INX-nnn에서 지정된 선두로부터 재생해야 할 시간의 SU의 개수를 1/4한 수로 지정함. 이것은, 통상의 SU의 4배 길이의 시간(약 93ms)에 상당함

<374> 값 : 0x0000 : 무설정 0x01로부터 0xFFFE(최대 6084초)

<375> 0xFFFF : 곡의 끝까지

<376> 이어서 곡명 영역 NM1 및 NM2에 대해 설명한다.

<377> NM1

<378> 의미 : 곡명을 나타내는 문자열

<379> 기능 : 1 바이트의 문자 코드로 나타낸 가변 길이의 곡명(최대 256)

<380> 이름 데이터의 종료는 반드시 종단 코드(0x00)를 기입하는 것

<381> 사이즈는 이 종단 코드로부터 계산하는 것. 데이터가 없는 경우에는 적어도 선두(0x0020)로부터 널(0x00)을 1 바이트 이상 기록하는 것

<382> 값 : 각종 문자 코드

<383> NM2

<384> 의미 : 곡명을 나타내는 문자열

<385> 기능 : 2 바이트의 문자 코드로 나타낸 가변 길이의 이름 데이터(최대 512), 이름 데이터의 종료는 반드시 종단 코드(0x00)를 기입하는 것

<386> 사이즈는 이 종단 코드로부터 계산하는 것. 데이터가 없는 경우에는 적어도 선두(0x0120)로부터 널(0x00)를 2 바이트 이상 기록하는 것

<387> 값 : 각종 문자 코드

<388> 속성 헤더의 고정 위치(0x320)로부터 시작된, 80 바이트의 데이터를 트랙 정보 영역 TRKINF라고 하고, 주로 시큐러티 관계, 복사 제어 관계의 정보를 일괄하여 관리한다. 도 23에는 TRKINF의 부분이 도시되어 있다. TRKINF 내의 데이터에 대하여, 배치 순서에 따라 이하에 설명한다.

<389> EKI(1 바이트)

<390> 의미 : 상술된 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의해 제공되는 암호화 콘텐츠 키: E(KEKn, Kcon)를 구비하는지의 여부를 나타냄

<391> 기능 : bit7 = 1로 키가 있고, bit7 = 0으로 없음. bit7 = 0인 경우에는 EKB_version, E(KEKn, Kcon)은 참조하지 않음

<392> 값 : 0으로부터 0xFF까지

<393> EKB_version(4 바이트)

<394> 의미 : 상술된 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의해 제공되는 콘텐츠 키의 세대 번호, 및/또는 유효화 키 블록(EKB)의 파일명을 나타냄

<395> 기능 : 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의해 제공되는 콘텐츠 키를 구하기 위한 유효화 키 블록(EKB)을 나타냄

<396> 값 : 0으로부터 0xFF까지

<397> E(Kstm, Kcon)(8 바이트)

<398> 의미 : 콘텐츠마다의 암호 처리용의 키인 콘텐츠 키를 메모리 카드의 스토리지 키(Kstm)로 암호화한 데이터

<399> 기능 : 콘텐츠의 암호 처리에 사용됨

<400> 값 : 0으로부터 0xFFFFFFFFFFFFFFFF까지

<401> E(KEKn, Kcon)(8 바이트)

<402> 의미 : 콘텐츠마다의 암호 처리용의 키인 콘텐츠 키를 상술된 계층 트리 구성에 의한 유효화 키 블록(EKB)에 의해 제공되는 키 암호화 키 KEKn에 의해 암호화한 데이터

<403> 기능 : 콘텐츠의 암호 처리에 사용됨

<404> 값 : 0으로부터 0xFFFFFFFFFFFFFFFF까지

<405> C_MAC[n](8 바이트)

<406> 의미 : 저작권 정보 개찬 체크치

<407> 기능 : 콘텐츠 누적 번호를 포함하는 복수의 TRKINF의 내용과 은닉 시퀀스 번호로부터 작성되는 값. 은닉 시퀀스 번호란 메모리 카드의 은닉 영역에 기록되어 있는 시퀀스 번호를 말함. 저작권 대응이 아닌 레코더는 은닉 영역을 판독할 수 없음. 또한, 저작권 대응의 전용 레코더, 또는 메모리 카드를 읽는 것을 가능하게 하는 어플리케이션을 탑재한 퍼스널 컴퓨터는 은닉 영역을 액세스할 수 있음

<408> A(1 바이트)

<409> 의미 : 파트의 속성

<410> 기능 : 파트 내의 압축 모드 등의 정보를 나타냄

<411> 값 : 도 24를 참조하여 이하에 설명함

<412> 단, N = 0, 1의 모노럴은 bit7이 1로 서브 신호를 0, 메인 신호(L+R)만의 특별한 Joint 모드를 모노럴로서 규정함. bit2, bit1의 정보는 통상의 재생기는 무시해도 상관없음

<413> A의 비트 0은, 엠퍼시스의 온/오프의 정보를 형성하고, 비트 1은 재생 SKIP이나 통상 재생의 정보를 형성하고, 비트 2는 데이터 구분, 예를 들면 오디오 데이터나 FAX 등의 다른 데이터의 정보를 형성한다. 비트 3은 미정의 상태이다. 비트 4, 5, 6을 조합함에 따라 도시된 바와 같이 ATRAC3의 모드 정보가 규정된다. 즉, N은 이 3 비트로 표시되는 모드의 값이고, 모노(N = 0, 1), LP(N = 2), SP(N = 4), EX(N = 5), HQ(N = 7)의 5 종류의 모드

에 대해 기록 시간(64MB의 메모리 카드인 경우), 데이터 전송 레이트, 1 블록 내의 SU 수가 나타내여져 있다. 1SU의 바이트 수는 (모노: 136 바이트, LP: 192 바이트, SP: 304 바이트, EX: 384 바이트, HQ: 512 바이트)이다. 또한, 비트 7에 의해 ATRAC3의 모드(0: Dual 1: Joint)가 나타내여진다.

<414> 일례로서, 64MB의 메모리 카드를 사용하고, SP 모드인 경우에 대해 설명한다. 64MB의 메모리 카드에는 3968 블록이 있다. SP 모드에서는, 1SU가 304 바이트이므로, 1 블록에 53SU가 존재한다. 1SU는 (1024 / 44100)초에 상당한다. 따라서, 1 블록은 (1024 / 44100) ×53 ×(3968 - 16) = 4863초 = 81분

<415> 전송 레이트는,

<416> (44100 / 1024) ×304 ×8 = 104737bps

<417> 가 된다.

<418> LT(1 바이트)

<419> 의미 : 재생 제한 플래그(비트 7 및 비트 6)와 시큐러티 버전(비트 5 - 비트 0)

<420> 기능 : 이 트랙에 대해 제한 사항이 있는 것을 나타냄

<421> 값 : 비트 7 : 0 = 제한 없음 1 = 제한 있음

<422> 비트 6 : 0 = 기한 내 1 = 기한 마감

<423> 비트 5 - 비트 0 : 시큐러티 버전0(0 이외이면 재생 금지로 함)

<424> FNo(2 바이트)

<425> 의미 : 파일 번호

<426> 기능 : 최초로 기록되었을 때의 트랙 번호, 또한 이 값은 메모리 카드 내의 은닉 영역에 기록된 MAC 계산용 값의 위치를 특정함

<427> 값 : 1로부터 0x190(400)

<428> MG(D) SERIAL-nnn(16 바이트(upper: 8, Lower: 8))

<429> 의미 : 기록 기기의 시큐러티 블록(시큐러티 IC20)의 일련 번호

<430> 기능 : 기록 기기마다 전부 다른 고유의 값

<431> 값 : 0으로부터 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

<432> CONNUM(4 바이트)

<433> 의미 : 콘텐츠 누적 번호

<434> 기능 : 곡마다 누적되어 가는 고유의 값으로 기록 기기의 시큐러티 블록에 의해 관리됨. 2의 32승, 42억곡분 준비되어 있고, 기록한 곡의 식별에 사용함

<435> 값 : 0으로부터 0xFFFFFFFF

<436> YMDhms-S(4 바이트)(Option)

<437> 의미 : 재생 제한이 있는 트랙의 재생 개시 일시

<438> 기능 : EMD에서 지정하는 재생 개시를 허가하는 일시

<439> 값 : 상술된 일시의 표기와 동일함

<440> YMDhms-E(4 바이트)(Option)

<441> 의미 : 재생 제한이 있는 트랙의 재생 종료 일시

<442> 기능 : EMD에서 지정하는 재생 허가를 종료하는 일시

<443> 값 : 상술된 일시의 표기와 동일함

<444> XCC(1 바이트)

<445> 의미 : 이하에 설명하는 CC의 확장부

<446> 기능 : 복사 제어

<447> CT(1 바이트)(Option)

<448> 의미 : 재생 횟수

<449> 기능 : 재생 허가된 횟수 내에서 실제로 재생할 수 있는 횟수. 재생할 때마다 디크리먼트함

<450> 값 : 0x00~0xFF 미사용 시에는 0x00임

<451> LT의 bit7이 1이고 CT의 값이 00인 경우에는 재생을 금지하는 것

<452> CC(1 바이트)

<453> 의미 : COPY CONTROL

<454> 기능 : 복사 제어

<455> 값 : 도 25에 도시된 바와 같이 비트 6 및 7에 의해 복사 제어 정보를 나타내며, 비트 4 및 5에 의해 고속 디지털 복사에 관한 복사 제어 정보를 나타내며, 비트 2 및 3에 의해 시큐러티 블록 인증 레벨을 나타냄. 비트 0 및 1은 미정의

<456> CC의 예 : (bit7, 6) 11 : 무제한의 복사를 허가, 01 : 복사 금지, 00 : 1회의 복사를 허가(bit3, 2) 00 : 아날로그 내지 디지털인으로부터의 녹음, MG 인증 레벨은 0으로 함

<457> CD로부터의 디지털 녹음에서는 (bit7, 6)은 00,(bit3, 2)는 00이 됨

<458> CN(1 바이트)(Option)

<459> 의미 : 고속 디지털 복사 HSCMS(High speed Serial Copy Management System)에 있어서의 복사 허가 횟수

<460> 기능 : 복사 1회나 무제한 복사의 구별을 확장하고, 횟수로 지정함. 복사 제1 세대인 경우에만 유효하고, 복사마다 감산함

<461> 값 : 00 : 복사 금지, 01로부터 0xFE : 횟수, 0xFF : 횟수 무제한.

<462> 상술된 트랙 정보 영역 TRKINF에 계속하여 0x0370으로부터 시작되는 24 바이트의 데이터를 파트 관리용의 파트 정보 영역 PRTINF라고 하고, 하나의 트랙을 복수의 파트로 구성하는 경우에 시간 축의 순서대로 PRTINF를 열거한다. 도 26에 PRTINF의 부분이 도시되어 있다. PRTINF 내의 데이터에 대해 배치 순서에 따라 이하에 설명한다.

<463> PRTSIZE(4 바이트)

<464> 의미 : 파트 사이즈

<465> 기능 : 파트의 크기를 나타냄. 클래스터: 2 바이트(최상위), 개시 SU: 1 바이트(상위), 종료 SU: 1 바이트(최하위)

<466> 값 : 클러스터: 1로부터 0x1F40(8000), 개시 SU: 0으로부터 0xA0(160), 종료 SU: 0으로부터 0xA0(160)(단, SU의 세는 방법은 0, 1, 2와 0로부터 개시함)

<467> PRTKEY(8 바이트)

<468> 의미 : 파트를 암호화하기 위한 값

<469> 기능 : 초기치 = 0, 편집 시에는 편집의 규칙에 따르는 것

<470> 값 : 0로부터 0xFFFFFFFFFFFFFFFF

<471> CONNUM0(4 바이트)

<472> 의미 : 최초로 만들어진 콘텐츠 누적 번호 키

<473> 기능 : 콘텐츠를 고유하게 하기 위한 ID의 역할

<474> 값 : 콘텐츠 누적 번호 초기치 키와 동일한 값이 됨

<475> 도 21로 되돌아간다. ATRAC3 데이터 파일의 속성 헤더 중에는 도 21에 도시된 바와 같이 부가 정보 INF가 포함된다. INF는 트랙에 관한 부가 정보 데이터이고, 헤더를 따른 가변 길이의 부가 정보 데이터. 복수의 다른 부가 정보가 배열되어지는 경우가 있다. 각각에 ID와 데이터 사이즈가 부가되어 있다. 개개의 헤더를 포함하는 부가 정보 데이터는 최소 16 바이트 이상이고 4 바이트의 정수배의 단위이다.

<476> 상술된 속성 헤더에 대해 ATRAC3 데이터 파일의 각 블록의 데이터가 계속된다. 도 27에 도시된 바와 같이 블록마다 헤더가 부가된다. 각 블록의 데이터에 대해 이하에 설명한다.

<477> BLKID-A3D(4 바이트)

<478> 의미 : BLOCKID FILE ID

<479> 기능 : ATRAC3 데이터의 선두인 것을 식별하기 위한 값

<480> 값 : 고정치 = "A3D"(예를 들면 0x41334420)

<481> MCode(2 바이트)

<482> 의미 : MAKER CODE

<483> 기능 : 기록한 기기의 메이커, 모델을 식별하는 코드

<484> 값 : 상위 10 비트(메이커 코드), 하위 6 비트(기종 코드)

<485> CONNUMO(4 바이트)

<486> 의미 : 최초로 만들어진 콘텐츠 누적 번호

<487> 기능 : 콘텐츠를 고유하게 하기 위한 ID의 역할, 편집되어도 값은 변화시키지 않음

<488> 값 : 콘텐츠 누적 번호 초기치 키와 동일한 값이 됨

<489> BLOCK SERIAL(4 바이트)

<490> 의미 : 트랙마다 붙여진 연속 번호

<491> 기능 : 블록의 선두는 0으로부터 시작하여 다음 블록은 +1씩 인크리먼트 편집되어도 값을 변화시키지 않음

<492> 값 : 0부터 시작, 0xFFFFFFFF까지

<493> BLOCK-SEED(8 바이트)

<494> 의미 : 1 블록을 암호화하기 위한 하나의 키

<495> 기능 : 블록의 선두는 기록 기기의 시큐러티 블록으로 난수를 생성, 계속되는 블록은 +1 인크리먼트된 값, 이 값이 없어지면, 1 블록에 상당하는 약 1초동안, 소리가 나지 않기 때문에, 헤더와 블록 말미에 동일한 것이 2중으로 쓰여짐. 편집되어도 값을 변화시키지 않음

<496> 값 : 초기는 8 바이트의 난수

<497> INITIALIZATION VECTOR(8 바이트)

<498> 의미 : 블록마다 ATRAC3 데이터를 암호화, 복호화할 때에 필요한 초기치

<499> 기능 : 블록의 선두는 0부터 시작, 다음 블록은 최후의 SU의 최후 암호화된 8 바이트의 값. 디바이드된 블록 도중으로부터 시작된 경우에는 개시 SU 직전의 최후 8 바이트를 이용함. 편집되어도 값을 변화시키지 않음

<500> 값 : 0부터 0xFFFFFFFFFFFFFFFF

<501> SU-nnn

<502> 의미 : 사운드 유닛의 데이터

<503> 기능 : 1024 샘플로부터 압축된 데이터, 압축 모드에 의해 출력되는 바이트 수가 다름. 편집되어도 값을 변화

시키지 않음(일례로서, SP 모드 시에는 N = 384 바이트)

<504> 값 : ATRAC3의 데이터값.

<505> 도 21에서는 N = 384이므로, 1 블록에 42SU가 쓰여진다. 또한, 1 블록의 선두의 두개의 슬롯(4 바이트)이 헤더가 되고, 최후 1 슬롯(2 바이트)에 BLKID-A3D, MCode, CONNUMO, BLOCK SERIAL이 2중으로 쓰여진다. 따라서, 1 블록의 나머지 영역 M 바이트는 (16, 384 - 384 ×42 - 16 ×3 = 208 바이트)가 된다. 이 중에 상술된 바와 같이 8 바이트의 BLOCK SEED가 2중으로 기록된다.

<506> 여기서, 플래시 메모리(34)에 기억되어 있는 데이터는 후술된 바와 같이 예를 들면 ATRAC3 방식으로 압축되어 있다. 압축의 단위가 사운드 유닛 SU 이다. 따라서, 기억 장치(300)로부터 재생 장치(200)에 데이터를 판독하는 경우에는 판독의 최소 단위는 상기 사운드 유닛 SU가 된다. 오디오 데이터의 압축 방식은, ATRAC3 등의 ATRAC 방식 이외의 CODEC 방식이라도 좋다.

<507> 블록 시드 데이터 BS는 각 블록마다 예를 들면 난수를 발생시켜 생성된 데이터이다.

<508> 〔플래시 메모리 관리 모듈(35)〕

<509> 플래시 메모리 관리 모듈(35)은 플래시 메모리(34)에의 데이터의 기입, 플래시 메모리(34)로부터의 데이터의 판독 등의 제어를 행한다.

<510> 도 15에 도시된 재생 장치(200)의 구성에 대해 설명한다. 재생 장치(200)는, 예를 들면 주제어 모듈(41), 통신 인터페이스(42), 제어 모듈(43), 편집 모듈(44), 압축/신장 모듈(45), 스피커(46), D/A 변환기(47) 및 A/D 변환기(48)를 구비한다.

<511> 〔주제어 모듈(41)〕

<512> 주제어 모듈(41)은 재생 장치(200)의 처리를 통괄적으로 제어한다.

<513> 〔제어 모듈(43)〕

<514> 도 15에 도시된 바와 같이 제어 모듈(43)은 예를 들면 난수 발생 유닛(60), 기억 유닛(61), 키 생성/키 연산 유닛(62), 상호 인증 유닛(63), 암호화/복호 유닛(64) 및 제어 유닛(65)을 구비한다. 제어 모듈(43)은 제어 모듈(33)과 같이 싱글 칩의 암호 처리 전용의 집적 회로로서, 다층 구조를 구비하고, 내부의 메모리 셀은 알루미늄층 등의 더미층에 삽입되어 있다. 또한, 제어 모듈(43)은 동작 전압 또는 동작 주파수의 폭이 좁아, 외부로부터 부정하게 데이터를 판독할 수 없도록 내탬퍼성을 갖고 있다. 난수 발생 유닛(60)은 난수 발생 지시를 받으면, 64 비트(8 바이트)의 난수를 발생시킨다. 기억 유닛(61)은 인증 처리에 필요한 여러 데이터를 기억하고 있다.

<515> 키 생성/키 연산 유닛(62)은, 예를 들면 ISO/IEC9797의 MAC 연산 방식을 이용한 연산 등의 여러 연산을 행하여 키 데이터를 생성한다. 이 때, "Block cipher Algorithm"로서 FIPS PUB 46-2에 규정되는 DES가 이용된다.

<516> 상호 인증 유닛(63)은 예를 들면 컴퓨터로부터 입력한 오디오 데이터를 기억 장치(300)로 출력하는 동작을 행하는데 앞서, 기억 장치(300) 사이에서 상호 인증 처리를 행한다. 또한, 상호 인증 유닛(63)은 기억 장치(300)로부터 오디오 데이터를 입력하는 동작을 행하는데 앞서 기억 장치(300) 사이에서 상호 인증 처리를 행한다. 또한, 상호 인증 유닛(63)은 상호 인증 처리에 있어서 상술된 MAC 연산을 행한다. 상기 상호 인증 처리에서는 기억 유닛(61)에 기억되어 있는 데이터가 이용된다. 또, 상호 인증 유닛(63)은 필요에 따라 예를 들면 퍼스널 컴퓨터(PC: 100) 또는 네트워크 상의 컴퓨터사이에서 오디오 데이터의 입출력을 행하는 동작에 앞서 퍼스널 컴퓨터(PC: 100) 또는 네트워크 상의 컴퓨터 사이에서 상호 인증 처리를 행한다.

<517> 암호화/복호 유닛(64)은 상술된 바와 같이 FIPS PUB(81)에 규정된 ECB 모드 및 CBC 모드를 선택적으로 이용하여 블록 암호화를 행한다.

<518> 암호화/복호 유닛(64)은 FIPS(81)의 모드 중, ECB 모드 및 CBC 모드의 복호를 선택적으로 행한다. 여기서, 암호화/복호 유닛(64)은 CBC 모드에 있어서, 예를 들면 56 비트의 키 데이터 k를 이용하여 암호문을 64 비트로 이루어지는 암호화 블록을 단위로 하여 복호하여 평문을 생성한다.

<519> 제어 유닛(65)은 난수 발생 유닛(60), 기억 유닛(61), 키 생성/키 연산 유닛(62), 상호 인증 유닛(63) 및 암호화/복호 유닛(64)의 처리를 통괄적으로 제어한다.

<520> 〔편집 모듈(44)〕

<521> 편집 모듈(44)은, 예를 들면 도 16에 도시된 바와 같이 기억 장치(300)의 플래시 메모리(34) 내에 기억된 트랙 데이터 파일을 사용자로부터의 조작 지시에 기초하여 편집하여 새로운 트랙 데이터 파일을 생성한다.

<522> 〔압축/신장 모듈(45)〕

<523> 압축/신장 모듈(45)은 예를 들면 기억 장치(300)로부터 입력한 암호화된 오디오 데이터를 복호한 후에 재생할 때에 ATRAC3 방식으로 압축되는 오디오 데이터를 신장하고, 상기 신장한 오디오 데이터를 D/A 변환기(47)를 출력한다. 또한, 예를 들면, CD, DVD 또는 PC1로부터 입력한 오디오 데이터를 기억 장치(300)에 기억할 때에 상기 오디오 데이터를 ATRAC3 방식으로 압축한다.

<524> 〔D/A 변환기(47)〕

<525> D/A 변환기(47)는 압축/신장 모듈(45)로부터 입력한 디지털 형식의 오디오 데이터를 아날로그 형식의 오디오 데이터로 변환하여 스피커(46)로 출력한다.

<526> 〔스피커(46)〕

<527> 스피커(46)는 D/A 변환기(47)로부터 입력한 오디오 데이터에 따른 음향을 출력한다.

<528> 〔A/D 변환기48〕

<529> A/D 변환기(48)는 예를 들면 CD 플레이어(7)로부터 입력한 아날로그 형식의 오디오 데이터를 디지털 형식으로 변환하여 압축/신장 모듈(45)로 출력한다.

<530> 〔메모리(49)〕

<531> 메모리(49)는 예를 들면 E2PROM(ex. 플래시 메모리)이고, 상술된 키 유효화 블록(EKB), 또는 EKB에 기초하여 생성되는 디바이스 키 블록(DKB) 등의 키 데이터, 디바이스 식별자로서의 디바이스 ID 등이 저장된다.

<532> [콘텐츠 데이터의 기억 장치에 대한 저장 처리 및 재생 처리]

<533> 도 15에 도시된 재생 장치(200)와, 기억 장치(300) 사이에서는 콘텐츠 데이터의 이동 즉, 재생 장치(200)로부터 기억 장치(300)의 플래시 메모리(34)에 대한 데이터 기록 처리가 실행되며, 또한 기억 장치(300)의 플래시 메모리(34)로부터 재생 장치(200)에 대한 데이터 재생 처리가 실행된다.

<534> 이 데이터 기록 및 재생 처리에 대해 이하와 같이 설명한다. 우선, 재생 장치(200)로부터 기억 장치(300)의 플래시 메모리(34)에 대한 데이터 기록 처리를 도 28의 플로우를 이용하여 설명한다.

<535> 재생 장치 및 기억 장치는 데이터 이동에 앞서 우선 단계 S2701, S2702에 도시된 상호 인증 처리를 실행한다. 도 29에는 공통 키 암호 방식을 이용한 상호 인증 방법(ISO/IEC 9798-2)이 도시되어 있다. 도 29에서는 공통 키 암호 방식으로서 DES를 이용하고 있지만, 공통 키 암호 방식이면 다른 방식도 가능하다. 도 29에서, 우선 B가 64 비트의 난수 Rb를 생성하고, Rb 및 자기의 ID인 ID(b)를 A로 송신한다. 이것을 수신한 A는 새롭게 64 비트의 난수 Ra를 생성하고, Ra, Rb, ID(b)의 순으로, DES의 CBC 모드로 키 Kab를 이용하여 데이터를 암호화하고, B로 반송한다. 또, 키 Kab는 A 및 B에 공통의 비밀 키로서 각각의 기록 소자 내에 저장하는 키이다. DES의 CBC 모드를 이용한 키 Kab에 의한 암호화 처리는, 예를 들면 DES를 이용한 처리에서는 초기치와 Ra를 배타적 논리합하고, DES 암호화부에 있어서 키 Kab를 이용하여 암호화하고, 암호문 E1을 생성하고, 계속해서 암호문 E1과 Rb를 배타적 논리합하고, DES 암호화부에서 키 Kab를 이용하여 암호화하고, 암호문 E2를 생성하고, 또한 암호문 E2와 ID(b)를 배타적 논리합하고, DES 암호화부에서 키 Kab를 이용하여 암호화하여 생성한 암호문 E3에 따라 송신 데이터(Token-AB)를 생성한다.

<536> 이것을 수신한 B는 수신 데이터를 역시 공통의 비밀 키로서 각각의 기록 소자 내에 저장하는 키 Kab(인증 키)로 복호화한다. 수신 데이터의 복호화 방법은, 우선 암호문 E1을 인증 키 Kab에서 복호화하고, 난수 Ra를 얻는다. 이어서, 암호문 E2를 인증 키 Kab에서 복호화하고, 그 결과와 E1을 배타적 논리합하고, Rb를 얻는다. 마지막으로, 암호문 E3을 인증 키 Kab에서 복호화하고, 그 결과와 E2를 배타적 논리합하여 ID(b)를 얻는다. 이렇게 함으로써 얻어진 Ra, Rb, ID(b) 중 Rb 및 ID(b)가 B가 송신한 것과 일치할지 검증한다. 이 검증에 통과한 경우, B는 A를 정당한 것으로서 인증한다.

<537> 이어서 B는, 인증 후에 사용하는 세션 키(Kses)를 생성한다(생성 방법은 난수를 이용함). 그리고, Rb, Ra,

Kses의 순으로, DES의 CBC 모드로 인증 키 Kab를 이용하여 암호화하고, A로 반송한다.

<538> 이것을 수신한 A는 수신 데이터를 인증 키 Kab에서 복호화한다. 수신 데이터의 복호화 방법은 B의 복호화 처리와 마찬가지이므로, 여기서는 상세한 내용을 생략한다. 이렇게 해서 얻어진 Rb, Ra, Kses 중 Rb 및 Ra가, A가 송신한 것과 일치한지를 검증한다. 이 검증에 통과한 경우, A는 B를 정당한 것으로서 인증한다. 상호 상대를 인증한 후에는 세션 키 Kses는 인증 후의 비밀 통신을 위한 공통 키로서 이용된다.

<539> 또, 수신 데이터의 검증시에 부정, 불일치가 발견된 경우에는 상호 인증이 실패한 것으로서 처리를 종료(S2703에서 No)한다.

<540> 상호 인증이 성립(S2703에서 Yes)한 경우에는 단계 S2704에서, 재생 장치가 콘텐츠 키 Kcon의 생성 처리를 실행한다. 이 처리는 도 15의 난수 생성 유닛(60)으로 생성한 난수를 이용하여 키 생성/키 연산 유닛(62)에 있어서 실행된다.

<541> 이어서, 단계 S2705에서(1) 콘텐츠 키 Kcon을 유효화 키 블록(EKB)으로부터 취득되는 암호화 키 KEK를 이용하여 암호화 처리하여, E(KEK, Kcon)를 생성함과 함께,(2) 콘텐츠 키 Kcon을 인증 처리에 있어서 생성한 세션 키(Kses)로 암호화 처리를 실행하여, E(Kses, Kcon)를 생성하여 기억 장치(메모리 카드)로 송신한다.

<542> 단계 S2706에서는 기억 장치가 재생 장치로부터 수신한 E(Kses, Kcon)를 세션 키로 복호하여 콘텐츠 키 Kcon을 취득하고, 또한 Kcon을 기억 장치에 사전에 저장되어 있는 스토리지 키 Kstm에 의해 암호화하여 E(Kstm, Kcon)를 생성하고, 이것을 재생 장치로 송신한다.

<543> 이어서, 재생 장치는 단계 S2707에 있어서, 단계 S2705에서 생성한 E(KEK, Kcon), 및 단계 S2706에서 기억 장치로부터 수신한 E(Kstm, Kcon)를 이용하여 데이터 파일(도 21 참조)을 포함하는 트랙 정보 영역 TRKINF 데이터를 생성하고, 데이터 파일의 포맷 처리 후, 이것을 기억 장치(메모리 카드)로 송신한다.

<544> 단계 S2708에서, 기억 장치(메모리 카드)는 재생 장치로부터 수신한 데이터 파일을 플래시 메모리에 저장한다.

<545> 이러한 처리에 의해 데이터 파일의 트랙 정보 영역 TRKINF 데이터에는 먼저 설명한 도 21, 도 23에 도시된 바와 같이 콘텐츠 키 Kcon을 유효화 키 블록(EKB)으로부터 취득되는 암호화 키 KEK를 이용하여 암호화 처리한 E(KEK, Kcon)와, 콘텐츠 키 Kcon을 기억 장치에 사전에 저장되어 있는 스토리지 키 Kstm에 의해 암호화한 E(Kstm, Kcon) 두개의 암호화 콘텐츠 키가 저장된다.

<546> 또, 음악 데이터, 화상 데이터 등의 암호화 처리는 콘텐츠 키 Kcon을 그대로 콘텐츠의 암호화 키로서 적용하여 실행하거나, 또는 콘텐츠를 포함하는 파트, 또는 블록 등을 단위로 하여 콘텐츠 키와 다른 키 생성 데이터에 기초하여 각 파트 단위, 또는 블록 단위의 암호화 키를 개별로 생성하여 각 파트 단위, 또는 블록 단위의 암호화 처리를 행하는 구성으로 하는 것이 가능하다.

<547> 이러한 데이터 파일을 이용한 재생 처리에서는 재생 장치는 E(KEK, Kcon)와, E(Kstm, Kcon) 중 하나를 선택적으로 적용하여 콘텐츠 키 Kcon을 취득 가능하게 한다.

<548> 이어서, 재생 장치(200)가 기억 장치(300)의 플래시 메모리(34)에 저장된 데이터의 판독 처리 즉, 재생 처리를 실행하는 경우의 처리를 도 30의 플로우를 이용하여 설명한다.

<549> 재생 장치 및 기억 장치는, 데이터 이동에 앞서, 우선 단계 S2901, S2902에 도시된 상호 인증 처리를 실행한다. 이 처리는, 먼저 설명한 도 29의 처리와 마찬가지이다. 상호 인증이 실패한 경우(S2903로 No)는 처리를 종료한다.

<550> 상호 인증이 성립(S2903에서 Yes)한 경우에는 단계 S2904에서 기억 장치가 재생 장치에 대해 데이터 파일을 송신한다. 데이터 파일을 수신한 재생 장치는 데이터 파일 내의 트랙 정보 영역 TRKINF 데이터를 검사하고, 콘텐츠 키(Kcon)의 저장 상황을 판별한다. 이 판별 처리는 키 유효화 블록(EKB)에 의해 취득되는 암호화 키 KEK에 의해 암호화된 콘텐츠 키 즉, E(KEK. Kcon)가 저장되어 있는지의 여부를 판별하는 처리이다. E(KEK. Kcon)의 유무는, 앞의 도 21, 도 23에서 설명한 데이터 파일 내의 트랙 정보 영역 TRKINF 데이터의 [EKI]의 데이터에 의해 판별 가능하다.

<551> E(KEK. Kcon)가 저장되어 있는 경우(단계 S2906에서 Yes)는 단계 S2907로 진행하고, 키 유효화 블록(EKB)의 처리에 의해 암호화 키 KEK를 취득하여, 취득한 암호화 키 KEK에 의해 E(KEK. Kcon)를 복호하여 콘텐츠 키 Kcon을 취득한다.

<552> E(KEK. Kcon)가 저장되지 않은 경우(단계 S2906에서 No)는 단계 S2908에서, 기억 장치의 제어 모듈(33)에 있어서 기억 장치에 사전에 저장되어 있는 스토리지 키 Kstm에 의해 암호화한 E(Kstm, Kcon)를 스토리지 키 Kstm에 의해 복호하고, 또한 상호 인증 처리에 있어서 재생 장치 및 기억 장치에서 공유한 세션 키 Kses에서 암호화한 데이터 E(Kses, Kcon)를 생성하여, 재생 장치로 송신한다.

<553> 재생 장치는 단계 S2909에서 기억 장치로부터 수신한 E(Kses, Kcon)를 세션 키 Kses에서 복호하여 콘텐츠 키 Kcon을 취득한다.

<554> 단계 S2910에서는, 단계 S2907, 또는 단계 S2909 중 하나에서 취득한 콘텐츠 키 Kcon에 의해 암호화 콘텐츠의 복호를 행한다.

<555> 이와 같이, 암호화 콘텐츠의 재생 처리에 있어서, 재생 장치는 E(KEK, Kcon)를 유효화 키 블록(EKB)으로부터 취득되는 암호화 키 KEK를 이용하여 복호하거나, 또는 기억 장치에 사전에 저장되어 있는 스토리지 키 Kstm에 의해 암호화한 E(Kstm, Kcon)에 기초하는 처리를 실행하거나, 하나의 처리를 실행함으로써 콘텐츠 키 Kcon을 취득할 수 있다.

<556> 또, 음악 데이터, 화상 데이터 등의 복호 처리는 콘텐츠 키 Kcon을 그대로 콘텐츠의 복호 키로서 적용하여 실행하거나, 또는 콘텐츠를 포함하는 파트, 또는 블록 등을 단위로 하여 콘텐츠 키와 다른 키 생성 데이터에 기초하여 각 파트 단위, 또는 블록 단위의 복호 키를 개별로 생성하여 각 파트 단위, 또는 블록 단위의 복호 처리를 행하는 구성으로 하는 것이 가능하다.

<557> [KEK를 저장한 EKB의 포맷]

<558> 먼저 도 6을 이용하여 유효화 키 블록(EKB)의 개략적인 포맷에 대해 설명했지만, 또한 키 암호화 키(KEK)를 유효화 키 블록(EKB)에 저장하여 보유하는 경우의 구체적인 데이터 구성예에 대해 설명한다.

<559> 도 31에는 키 암호화 키(KEK)를 유효화 키 블록(EKB)에 저장한 데이터인 EKB인 배신 키 허가 정보 파일의 구성예가 도시되어 있다. 디바이스(재생 장치)는 이 파일로부터 필요에 따라 키 암호화 키(KEK)를 추출하여, KEK에 의해 E(KEK, Kcon)를 복호하여 콘텐츠 키: Kcon을 취득하여 콘텐츠의 복호를 실행한다. 각 데이터에 대해 설명한다.

<560> BLKID-EKB(4 바이트)

<561> 의미 : BLOCKID FILE ID

<562> 기능 : 배신 키 정보 파일의 선두인 것을 식별하기 위한 값

<563> 값 : 고정치 = "EKB"(예를 들면, 0x454B4220)

<564> MCode(2 바이트)

<565> 의미 : MAKER CODE

<566> 기능 : 기록한 기기의 메이커, 모델을 식별하는 코드

<567> 값 : 상위 10 비트(메이커 코드), 하위 6 비트(기종 코드)

<568> LKF

<569> 의미 : LINK FILE INFORMATION

<570> 기능 : 이 EKB에 의해 취득되는 KEK가 적용 가능한 콘텐츠 데이터인 링크파일을 식별함

<571> 값 : 0~0xFF

<572> bit7 : 재생 관리 파일(PBLIST)에 사용 : 1, 미사용 : 0

<573> bit6 : 개찬 체크치(ICV)에 사용 : 1, 미사용 : 0

<574> bit5~0 : 리저브

<575> LINK count

<576> 의미 : LINK COUNT

<577> 기능 : 링크하고 있는 파일(예를 들면 ATRACK3 파일) 수

<578> 값 : 0~0xFFFFFFFF

<579> Version

<580> 의미 : VERSION

<581> 기능 : 배신 키 허가 정보 파일의 버전을 나타냄

<582> 값 : 0~0xFFFFFFFF

<583> EA

<584> 의미 : Encryption Algorithm

<585> 기능 : 배신 키 허가 정보 파일의 트레이스 처리 알고리즘을 나타냄

<586> 값 : 0~0xFF

<587> 00h : 3DES : 트리플 DES 모드에 의한 처리

<588> 01h : DES : 싱글 DES 모드에 의한 처리

<589> 또, 트리플 DES 모드에 의한 처리는 2 종류 이상의 암호 처리 키를 이용하는 암호 처리이고, 싱글 DES 모드는 하나의 키에 의한 처리이다.

<590> KEK1

<591> 의미 : Key Encrypting Key

<592> 기능 : 키 유효화 블록(EKB) 내의 루트 키(최상위) 키로 암호화된 콘텐츠 키 암호 키

<593> 값 : 0~0xFFFFFFFFFFFFFFFF

<594> KEK2

<595> 의미 : Key Encrypting Key

<596> 기능 : 키 유효화 블록(EKB) 내의 루트 키(최상위) 키로 암호화된 콘텐츠 키 암호 키

<597> 값 : 0~0xFFFFFFFFFFFFFFFF

<598> E(Version)

<599> 의미 : Encrypted Version

<600> 기능 : 키 유효화 블록(EKB) 내의 루트 키(최상위) 키로 암호화된 버전 번호. 복호 시의 하위 4 바이트는 리저브

<601> 값 : 0~0xFFFFFFFFFFFFFFFF

<602> Size of tag part

<603> 의미 : Size of tag part

<604> 기능 : 배신 키 허가 정보 파일을 포함하는 데이터의 태그 부분의 사이즈(Byte)

<605> 값 : 0~0xFFFFFFFF

<606> Size of Key part

<607> 의미 : Size of key part

<608> 기능 : 배신 키 허가 정보 파일을 포함하는 데이터의 키 부분의 사이즈(Byte)

<609> 값 : 0~0xFFFFFFFF

<610> Size of Sign part

<611> 의미 : Size of sign part

<612> 기능 : 배신 키 허가 정보 파일을 포함하는 데이터의 사인 부분의 사이즈(Byte)

<613> 값 : 0~0xFFFFFFFF

<614> Tag part

<615> 의미 : Tag part

<616> 기능 : 배신 키 허가 정보 파일을 포함하는 데이터의 태그 부분의 데이터

<617> 값 : 모든 값

<618> 8 바이트에 미치지 않은 경우에는 0으로 매립하여 8 바이트로 함

<619> Key part

<620> 의미 : Key part

<621> 기능 : 배신 키 허가 정보 파일을 포함하는 데이터의 키 부분의 데이터

<622> 값 : 모든 값

<623> Signature part

<624> 의미 : Signature part

<625> 기능 : 배신 키 허가 정보 파일을 포함하는 데이터의 서명(Signature) 부분의 데이터

<626> 값 : 모든 값

<627> 상술된 설명 및 도 31에 의해 도시된 바와 같이, 디바이스에 대해 제공되는 배신 키 허가 정보 파일에는 그 배신 키 허가 정보 파일로부터 취득되는 KEK가 적용 가능한 콘텐츠 데이터인 링크파일을 식별하기 위한 식별 데이터[LKF]가 저장되고, 또한 링크하고 있는 파일(예를 들면 ATRACK3 파일) 수로서의 데이터[Linc Count]가 저장된다. 재생 장치는 [LKF], [Link Count]을 참조함으로써, 그 배신 키 허가 정보 파일로부터 취득되는 KEK를 적용해야 할 데이터가 존재하는지의 여부 및 그 수를 알 수 있다.

<628> [링크 정보를 이용한 데이터 복호, 재생 처리]

<629> 상술된 배신 키 허가 정보 파일에 포함되는 링크파일을 식별하기 위한 식별 데이터[LKF], 링크하고 있는 파일(예를 들면 ATRACK3 파일) 수로서의 데이터[Linc Count]를 이용하여 효율적으로 데이터의 복호, 재생을 실행하는 처리 형태에 대해 이하와 같이 설명한다.

<630> 도 32에는 기억 장치의 데이터 저장 영역, 예를 들면 도 15에 도시된 기억 장치(300)의 플래시 메모리(34)에 저장된 데이터 파일 구성예가 도시되어 있다. 여기서는 음악 데이터(HIFI)의 디렉토리 구성만을 예로서 도시하고 있지만, 또한 화상 파일 등의 디렉토리가 존재해도 좋다.

<631> 도 32에 도시된 음악 데이터의 디렉토리에는 재생 관리 파일(PBLIST), 암호화 콘텐츠로서 복수의 ATRACK3 데이터 파일(A3D)이 포함된다. 또한, 기억 장치에는 복수의 유효화 키 블록 파일(EKBn)이 저장된다. ATRACK3 데이터 파일(A3D)의 복호 처리에 적용하는 콘텐츠 키를 취득하기 위한 유효화 키 블록 파일(EKBn)은 ATRACK3 데이터 파일(A3D)에 포함되는 포인터에 의해 판별된다. 도 32에 도시된 바와 같이 하나의 유효화 키 블록 파일(EKB1: 3101)은 복수(3)의 ATRACK3 데이터 파일(A3D)의 복호 처리에 적용된다.

<632> 이 경우, 유효화 키 블록 파일(EKB1: 3101)에 대응하는 배신 키 허가 정보 파일의 [Linc Count]에는 3개의 콘텐츠에 적용되는 것을 나타내는 데이터가 저장된다.

<633> 도 32와 같은 복수의 콘텐츠 파일, 복수의 유효화 키 블록 파일을 저장한 기억 장치인 메모리 카드로부터 콘텐츠를 복호하여, 재생하는 경우의 처리 플로우가 도 33에 도시되어 있다.

<634> 도 33의 처리는, 예를 들면 기억 장치로서의 메모리 카드를 재생 장치로 세트했을 때, 또는 메모리 카드를 장착한 재생 장치의 전원을 ON 했을 때에 재생 장치가 실행하는 처리이다.

<635> 우선, 단계 S3201에서 재생 장치는 각각의 EKB 파일의 트랙 정보를 판독하고, [Linc Count]를 체크한다. 또한, [Linc Count]의 카운트 수가 많은 것부터 순서대로 사전에 정해진 개수[n]의 EKB 파일을 선택한다. 개수[n]는

재생 장치의 소정 메모리 영역 즉, 키 암호화 키: KEK를 저장 보유하는 영역에 저장 가능한 개수에 상당하는 개수로서 설정된다.

<636> 이어서, 단계 S3202에서 선택한 EKB의 처리에 의해 복수[n]의 키 암호화 키: KEK를 취득하고, 이들을 재생 장치의 키 저장 영역으로서 설정된 RAM의 소정 영역에 저장한다.

<637> 이어서, 재생 장치는 단계 S3203에서, 복호, 재생하는 콘텐츠를 선택한다. 또한 단계 S3204에서 그 선택 콘텐츠의 복호에 적용하는 KEK가 RAM에 저장되어 있는지의 여부를 판정하고, Yes인 경우에는 단계 S3205로 진행하며, 그 대응 KEK에 기초하여 E(KEK, Kcon)를 복호하여 콘텐츠 키를 취득하여, 단계 S3209에서 재생 즉, 취득한 콘텐츠 키에 의한 데이터의 복호, 재생 처리를 실행한다.

<638> 단계 S3204에서, 선택 콘텐츠의 복호에 적용하는 KEK가 RAM에 저장되지 않은 경우에는 단계 S3206에서 스토리지 키로 암호화된 콘텐츠 키 즉, E(Kstm, Kcon)의 유무를 판정하고, 어떤 경우에는 단계 S3207에서 E(Kstm, Kcon)의 복호 처리에 의해 콘텐츠 키를 취득하여 단계 S3209에서 재생 즉, 취득한 콘텐츠 키에 의한 데이터의 복호, 재생 처리를 실행한다.

<639> 또한, 단계 S3206에서 E(Kstm, Kcon)이 없다고 판정되면, 그 복호 대상 콘텐츠에 적용해야 할 EKB를 기억 장치로부터 취득하여, 취득한 EKB의 복호 처리에 의해 KEK를 취득하고, 취득한 KEK에 의한 E(KEK, Kcon)의 복호 처리를 실행하여 콘텐츠 키를 취득하여, 단계 S3209에서 재생 즉, 취득한 콘텐츠 키에 의한 데이터의 복호, 재생 처리를 실행한다.

<640> 이와 같이, 재생 장치는 사전에 기억 장치에 저장한 복수의 키 유효화 블록(EKB)의 [Linc Count]를 체크하고, [Linc Count]의 카운트 수가 많은 EKB의 복호를 실행하여, 키 암호화 키: KEK를 저장해 두는 구성으로 함으로써, 콘텐츠 재생 처리시에 높은 확률로 RAM에 저장한 KEK를 적용 가능하게 되고, 효율적인 콘텐츠 재생을 실행할 수 있다.

<641> [키 유효화 블록(EKB)에 의한 인증 키 배신]

<642> 상술된 유효화 키 블록(EKB)을 사용한 키의 배신에 있어서, 인증 처리를 실행할 때에 사용하는 인증 키 IKn을 배신함으로써, 안전한 비밀 키로서 공유하는 인증 키를 제공하고, 공통 키 방식에 따른 인증 처리를 실행하는 구성에 대하여 설명한다.

<643> 공통 키 암호 방식을 이용한 상호 인증 방법(ISO/IEC 9798-2)은 먼저 도 29를 이용하여 설명한 처리이고, 데이터 송수신이 실행되기 전의 처리로서 쌍방의 정당성을 확인하기 위한 처리로서 실행된다. 인증 처리에서는 데이터의 송수신을 행하는, 예를 들면 재생 장치와 기억 장치는 인증 키 Kab를 공유한다. 이 공통 키 Kab를 상술된 유효화 키 블록(EKB)을 사용하여 재생 장치에 배신한다.

<644> 도 34 및 도 35에는, 복수의 디바이스에 공통의 인증 키 IKn을 유효화 키 블록(EKB)에 의해 배신하는 구성예가 도시되어 있다. 도 34는 디바이스 0, 1, 2, 3에 대하여 복호 가능한 인증 키 IKn을 배신하는 예를, 도 35는 디바이스 0, 1, 2, 3 중 디바이스(3)를 리보크(배제)하여 디바이스 0, 1, 2에 대해서만 복호 가능한 인증 키를 배신하는 예를 도시한다.

<645> 도 34의 예에서는, 갱신 노드 키 K(t)00에 의해 인증 키 IKn을 암호화한 데이터(b)와 함께 디바이스 0, 1, 2, 3에서 각각 구비한 노드 키, 리프 키를 이용하여 갱신된 노드 키 K(t)00을 복호 가능한 유효화 키 블록(EKB)을 생성하여 배신한다. 각각의 디바이스는 도 34의 우측에 도시된 바와 같이 우선, EKB를 처리(복호)함으로써, 갱신된 노드 키 K(t)00을 취득하고, 이어서 취득한 노드 키 K(t)00을 이용하여 암호화된 인증 키: Enc(K(t)00, IKn)를 복호하여 인증 키 IKn을 얻는 것이 가능하게 된다.

<646> 그 밖의 디바이스 4, 5, 6, 7, …은 동일한 유효화 키 블록(EKB)을 수신해도 자신이 보유하는 노드 키, 리프 키에서는 EKB를 처리하여 갱신된 노드 키 K(t)00을 취득할 수 없으므로, 안전하여 정당한 디바이스에 대해서만 인증 키를 송부할 수 있다.

<647> 한편, 도 35의 예는 디바이스 3이 예를 들면 키의 누설에 의해 리보크(배제)되어 있다고 해도, 다른 그룹의 멤버 즉, 디바이스 0, 1, 2에 대해서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 배신한 예이다. 도 35에 도시된 (a) 유효화 키 블록(EKB)과, (b) 인증 키(IKn)를 노드 키(K(t)00)로 암호화한 데이터를 배신한다.

<648> 도 35의 우측에는 복호 수순을 나타내고 있다. 디바이스 0, 1, 2는 우선 수령한 유효화 키 블록으로부터 자신이 보유하는 리프 키 또는 노드 키를 이용한 복호 처리에 의해 갱신 노드 키(K(t)00)를 취득한다. 이어서,

K(t)00에 의한 복호에 의해 인증 키 IKn을 취득한다.

<649> 다른 그룹의 디바이스, 예를 들면 디바이스 4, 5, 6, …은 이 동일한 데이터(EKB)를 수신했다고 해도 자신이 보유하는 리프 키, 노드 키를 이용하여 갱신 노드 키(K(t)00)를 취득할 수 없다. 마찬가지로 리보크된 디바이스 3에서도 자신이 보유하는 리프 키, 노드 키에서는 갱신 노드 키(K(t)00)를 취득할 수 없고, 정당한 권리를 구비한 디바이스만이 인증 키를 복호하여 이용하는 것이 가능하게 된다.

<650> 이와 같이, EKB를 이용한 인증 키의 배송을 이용하면, 데이터량을 적게 하고, 또한 안전하게 정당 권리자만이 복호 가능하게 한 인증 키를 배신하는 것이 가능하게 된다. 또, 유효화 키 블록(EKB)에 의해 암호화되고 제공되는 EKB 배신 인증 키는 세대(버전) 관리가 이루어져, 세대마다의 갱신 처리가 실행되고, 임의의 타이밍에서의 디바이스의 리보크(배제)가 가능하다.

<651> 상술된 EKB에 의한 인증 키의 제공 처리에 의해 리보크된 디바이스(재생 장치)에서는 기억 장치(예를 들면 메모리 카드)와의 인증 처리가 성립하지 않고, 데이터의 부정한 복호가 불가능하게 된다.

<652> 또한, EKB를 이용한 인증 키의 배송을 이용하면, 메모리 카드 이외의 기억 매체, 예를 들면 재생 장치에 내장한 하드 디스크 등의 기억 매체에 대한 데이터 저장, 재생 처리에 대한 제어도 가능하게 된다.

<653> 앞의 도 28~도 30을 이용하여 설명한 바와 같이 기억 장치를 이용한 콘텐츠의 기록, 재생 처리에서는 상호 인증 처리가 실행되고, 상호 인증 처리의 성립을 조건으로 하여, 데이터의 기록 및 재생이 가능하게 된다. 이 인증 처리 프로그램은 메모리 카드와 같은 상호 인증 처리가 가능한 기억 장치 사이에서의 처리에서는 유효하게 작용하지만, 예를 들면 재생 장치가 하드 디스크, CD-R 등 암호 처리 기능을 갖지 않는 즉, 상호 인증을 실행 불가능한 기억 매체에 대해 데이터 저장, 데이터 재생시에는 의미를 이루지 않게 된다. 그러나, 본 발명의 시스템에서는 이러한 인증 불가능한 기기를 이용한 데이터 저장, 또는 데이터 재생 처리에서도 인증 처리 프로그램을 실행시키는 구성으로 한다. 하드 디스크, CD-R 등은 상호 인증이 불가능하므로, 가상의 메모리 카드(메모리 스틱)를 재생 장치에 구성하며, 가상 메모리 카드와 재생 장치 사이에서 인증 처리를 실행시켜, 인증 성립을 조건으로 하여 인증 기능을 갖지 않은 기억 매체에 대한 데이터 저장 처리, 또는 기억 매체로부터의 데이터 재생을 가능하게 한다.

<654> 이들의 가상 메모리 카드를 사용한 데이터 기록, 재생 처리 플로우는 도 36에 도시되어 있다. 우선, 재생 장치는 재생 장치 내의 가상 메모리 카드 사이에서 상호 인증 처리를 실행한다. 단계 S3502에서, 인증 성립했는지의 여부를 판정하고, 성립한 것을 조건으로 하여 단계 S3503으로 진행하며, 인증 기능을 갖지 않은 기억 매체, 예를 들면 하드디스크, CD-R, DVD 등을 이용한 데이터 기록, 재생 처리를 실행한다.

<655> 단계 S3502에서, 인증이 성립하지 않았다고 판정된 경우에는 단계 S3503의 인증 기능을 갖지 않은 기억 매체, 예를 들면 하드 디스크, CD-R, DVD 등을 이용한 데이터 기록, 재생 처리가 실행되지 않는다.

<656> 여기서, 가상 메모리 카드에는 사전에 앞의 도 16에서 설명한 인증 키 데이터를 저장한 구성으로 하고, 재생 장치가 사용하는 인증 키를 상술한 바와 같이 키 유효화 블록으로 제공하는 구성으로 한다.

<657> 이와 같이, 재생 장치의 인증 키를 키 유효화 블록(EKB)으로 제공함으로써, 정당한 라이센스를 구비한 디바이스(재생 장치)에 대해서만, 가상 메모리 카드와의 상호 인증 가능한 인증 키를 배신하는 것이 가능하게 된다. 따라서, 부정한 기기 즉, 리보크된 재생 장치에는 유효한 인증 키가 배신하지 않은 처리가 가능하게 된다. 유효한 인증 키가 제공되지 않은 재생 장치는 상호 인증이 불성립이 되며, 인증 기능을 구비한 메모리 카드뿐만 아니라, 인증 기능을 갖지 않은 기억 매체, 예를 들면 하드디스크, CD-R, DVD 등을 이용한 데이터 기록, 재생 처리가 실행되지 않고, 부정한 기기에 의한 데이터 기록, 재생을 배제하는 것이 가능하게 된다.

<658> 즉, 인증 키를 제공하는 유효화 키 블록(EKB)을 키 트리의 리프를 포함하는 데이터 처리 장치 중 정당한 라이센스를 구비한 데이터 처리 장치에서만 복호 가능하며, 정당 라이센스를 갖지 않은 부정한 데이터 처리 장치에서는 복호 불가능한 유효화 키 블록(EKB)으로서 제공함으로써, 부정한 데이터 처리 장치에서의 가상 메모리 디바이스와의 인증 성립을 방지하여, 부정 데이터 처리 장치에서의 콘텐츠 이용을 배제 가능하게 한 구성을 구비한 라이센스 시스템이 실현된다.

<659> [체크치(ICV: Integrity Check Value) 저장 구성]

<660> 이어서, 콘텐츠의 개찬을 방지하기 위해 콘텐츠의 인테그리티·체크치(ICV)를 생성하여 콘텐츠에 대응시켜 ICV의 계산에 의해 콘텐츠 개찬의 유무를 판정하는 처리 구성에 대해 설명한다.

<661> 콘텐츠의 인티그리티 체크치(ICV)는 예를 들면 콘텐츠에 대한 해시 함수를 이용하여 계산되며, ICV = hash(Kicv, C1, C2, …)에 의해 계산된다. Kicv는 ICV 생성 키이다. C1, C2는 콘텐츠의 정보이고, 콘텐츠의 중요 정보의 메시지 인증 부호(MAC: Message authentication Code)가 사용된다. 상술된 바와 같이 [MAC]는 도 21에서 설명한 ATRAC3 데이터 파일에도 포함된다. 이들을 사용하여 인티그리티 체크치(ICV)의 계산이 이루어진다.

<662> DES 암호 처리 구성을 이용한 MAC치 생성예가 도 37에 도시되어 있다. 도 37의 구성에 도시된 바와 같이 대상이 되는 메시지를 8 바이트 단위로 분할하며,(이하, 분할된 메시지를 M1, M2, …, MN으로 함), 우선 초기치(Initial Value(이하, IV로 함))와 M1을 배타적 논리합한다(그 결과를 I1로 함). 이어서, I1을 DES 암호화부에 넣고, 키(이하, K1로 함)를 이용하여 암호화한다(출력을 E1로 함). 계속하여, E1 및 M2를 배타적 논리합하고, 그 출력 I2를 DES 암호화부에 넣고, 키 K1을 이용하여 암호화한다(출력 E2). 이하, 이것을 반복하고, 모든 메시지에 대해 암호화 처리를 실시한다. 마지막으로 나온 EN이 메시지 인증 부호(MAC(Message Authentication Code))로 된다. 또, 메시지로서는 검증 대상이 되는 콘텐츠 및 헤더 정보 등의 콘텐츠 관련 데이터를 포함하는 부분 데이터가 사용 가능하다.

<663> 이러한 콘텐츠의 MAC치와 ICV 생성 키 Kicv에 해시 함수를 적용하여 이용하여 콘텐츠의 인티그리티 체크치(ICV)가 생성된다. 개찬이 없는 것이 보증된 예를 들면 콘텐츠 생성 시에 생성한 ICV와, 새롭게 콘텐츠에 기초하여 생성한 ICV를 비교하여 동일한 ICV를 얻을 수 있으면 콘텐츠에 개찬이 없는 것이 보증되며, ICV가 다르면 개찬이 있었다고 판정된다.

<664> 상술된 바와 같은 인티그리티 체크치(ICV)는 콘텐츠 개개에 대해 생성되는 복수의 콘텐츠 MAC치에 의해 하나의 인티그리티 체크치(ICV)를 생성하는 것이 가능하다. 복수의 MAC에 의한 ICV의 계산은, 예를 들면 ICV = MAC(Kicv, C_MAC [0]‖C_MAC[1]‖C_MAC[2]‖…)에 의해 생성한다.

<665> 콘텐츠 생성 시에 생성한 ICV를 저장해 두고, 체크 처리 시에 생성 ICV와 저장 ICV의 비교 처리를 행한다. 양 ICV가 일치하면 개찬이 없다고 판정하고, ICV가 불일치한 경우에는 개찬이 있다고 판정되며, 데이터 재생 등의 처리 제한이 이루어진다.

<666> 메모리 카드 등의 기억 장치에는 음악 콘텐츠뿐만 아니라, 화상 데이터, 게임 프로그램 데이터 등, 카테고리가 다르지만 저장된다. 이들 각 카테고리의 콘텐츠도 개찬의 방지를 도모하기 때문에 각 카테고리마다 인티그리티 체크치(ICV)를 생성하여 저장하는 것이 콘텐츠 개찬 체크를 위해서는 유효한 수단이 된다.

<667> 그러나, 메모리에 저장하는 콘텐츠 수가 증대하면, 검증용의 체크치를 정규 콘텐츠 데이터에 기초하여 생성하며, 저장하여 관리하는 것이 곤란하게 된다. 특히, 최근 플래시 메모리를 사용한 메모리 카드 등의 용량이 큰 매체에서는 음악 데이터, 화상 데이터, 프로그램 데이터 등 여러 카테고리의 콘텐츠 데이터가 메모리에 저장된다. 이러한 환경에서는 체크치의 생성 처리, 저장 처리, 개찬 체크 처리의 관리는 곤란하게 된다. 저장 데이터 전체에 대한 체크치를 생성하면, 체크 대상이 된 데이터 전체에 대한 체크치 생성 처리를 실행하는 것이 필요하다. 예를 들면 DES-CBC 모드에서 생성되는 메시지 인증 부호(MAC)에 의해 체크치 ICV를 구하는 수법을 행하는 경우, 데이터 전체에 대한 DES-CBC의 처리를 실행하는 것이 필요하다. 이 계산량은 데이터 길이가 길어짐에 따라 증대되며, 처리 효율인 점에서 문제가 있다.

<668> 기억 장치로서 사용 가능한 메모리 카드에는 많은 카테고리가 다른 콘텐츠가 저장된다. 이들의 카테고리가 다른 콘텐츠의 개찬 체크 관리를 카테고리마다 독립한 인티그리티 체크치(ICV)를 생성하여 실행하는 구성으로 함으로써, ICV의 체크시, 또는 ICV의 변경시, 예를 들면 데이터 변경 시의 새로운 인티그리티 체크치(ICV)의 생성 처리가 하나의 카테고리 내의 데이터를 대상으로서 실행 가능하게 되며, 다른 카테고리에 영향을 미치는 일이 없다. 이렇게 카테고리마다의 복수 인티그리티 체크치(ICV)를 저장하는 구성에 대해 설명한다.

<669> 도 38에는 기억 장치에 저장되는 데이터 구성과, 각각의 인테그리티 체크치(ICV)의 저장 구성예가 도시되어 있다. 메모리 카드 등의 기억부(플래시 메모리)에는 도 38에 도시된 바와 같이 음악 데이터의 디렉토리에 재생 관리 파일(PBLIST), 암호화 콘텐츠로서 복수의 ATRACK3 데이터 파일(A3D)이 포함되고, 또한 메모리에는 복수의 카테고리에 속하는 콘텐츠 데이터(#1~#n)가 저장된다. 복수의 카테고리는, 예를 들면 음악 데이터, 화상 데이터, 게임 프로그램 등이다. 또한, 동일한 화상 데이터라도 각각의 데이터 제공원에 따라 다른 디렉토리로서 독립 카테고리로서 관리해도 좋다.

<670> 또한, 상술된 유효화 키 블록(EKB)의 관리 단위(엔티티)를 1카테고리로서 설정해도 좋다. 즉, 어떤 유효화 키 블록(EKB)에 의해 취득되는 키 암호 키: KEK에 의해 복호되는 콘텐츠 키 Kcon을 적용 가능한 콘텐츠 집합을 하

나의 카테고리로서 설정해도 좋다.

<671> 재생 관리 파일(PBLIST), 암호화 콘텐츠로서 복수의 ATRACK3 데이터 파일(A3D) 각각에는 개찬 체크를 위한 메시지 인증 부호(MAC(Message Authentication Code))가 포함되고, 이들의 MAC치에 기초하여 인티그리티 체크치(ICV(con))가 생성된다. 복수의 콘텐츠의 MAC치는 플래시 메모리의 시퀀스 페이지에 MAC 리스트로서 저장, 관리되며, 이들의 MAC 리스트에 기초하여 ICV 생성 키 Kicv를 적용하여 얻을 수 있는 인티그리티 체크치(ICV(con))가 저장 보존된다.

<672> 콘텐츠 MAC치를 저장하는 시퀀스 페이지 포맷이 도 39에 도시되어 있다. 시퀀스 페이지 영역은 일반 콘텐츠 데이터의 기입 금지 영역으로서 설정된 영역이다. 도 39의 시퀀스 페이지 구성에 대해 설명한다.

<673> E(kSTR, kCON)는 메모리 카드의 스토리지 키로 암호화한 콘텐츠 키이다. ID(upper),(lower)는 메모리 카드의 식별자(ID)의 저장 영역이다. C_MAC[0]는 재생 관리 파일(PBLIST)의 구성 데이터에 기초하여 생성된 MAC치이다. C_MAC[1]는 콘텐츠, 예를 들면 ATRACK3 데이터 파일 #1의 데이터에 기초하여 생성된 MAC치, 이하 콘텐츠마다 MAC치가 저장된다. 이들의 MAC치에 기초하여 인티그리티 체크치(ICV(con))가 생성되며, 생성된 ICV(con)가 시리얼 프로토콜을 통해 메모리에 기입된다. 또, 다른 키 시스템에 대응하기 위해, 각각의 키 시스템으로부터 생성되는 ICV를 각각 다른 영역에 저장하는 구성으로 하는 것이 바람직하다.

<674> 또한, 카테고리마다 개찬 체크를 위해 생성되는 각 카테고리마다의 인티그리티 체크치(ICV)는 메모리 카드의 기억부(플래시 메모리)의 풀페이지에 기록된다. 풀페이지도 또한, 일반 데이터 기입의 금지된 영역으로서 설정되어 있다.

<675> 각 카테고리마다의 인티그리티 체크치(ICV)를 저장하는 풀 페이지 포맷이 도 40에 도시되어 있다. #0_revision은, 카테고리 #0의 갱신 데이터가 설정되며, 갱신된 경우에는 인크리먼트된다. #0_version은 카테고리 #0의 버전, #0_E(KEK, Kicv)는 카테고리 #0의 키 암호화 키(KEK)로 암호화한 ICV 생성 키(Kicv)이고, ICV0은 카테고리 #0의 인티그리티 체크치(ICV)값이다. 이하, 동일한 데이터가 각 카테고리마다 EKB#15까지 저장 가능하게 되어 있다.

<676> ICV의 체크는, 파워 온시, 또는 메모리 카드 등의 기억 장치가 재생 장치에 세트된 것을 조건으로 하여 개시된다. 도 41에는 ICV 체크를 포함하는 처리 플로우가 도시되어 있다.

<677> 우선, 재생 장치가 파워 온, 또는 새로운 메모리 카드 등이 장착된 것을 검지하면, 단계 S4001에서 재생 장치와 기억 장치 사이의 상호 인증이 가능한지의 여부가 판정되며, 가능한 경우에는 단계 S4002에서 기억 장치와 재생 장치 사이에서의 상호 인증 처리(도 29 참조)가 실행된다. 또한, 단계 S4001에서 재생 장치와 기억 장치 사이의 상호 인증이 가능하지 않다고 판정된 경우에는, 단계 S4003에서 상술된 가상 메모리 카드와 재생 장치사이의 상호 인증 처리가 실행된다.

<678> 단계 S4004에서 상호 인증이 성립했는지의 여부가 판정되며, 불성립하는 경우에는 이하의 처리는 실행되지 않고 종료한다. 상호 인증이 성립하는 경우에는 단계 S4005에서 ICV의 계산이 실행된다. ICV는 상술된 바와 같이 각 파일의 MAC치에 기초하여 산출된다.

<679> 이어서 단계 S4006에서, 계산에 의해 산출된 생성 ICV와, 사전에 저장하고 있는 저장 ICV와의 비교가 실행된다. 양 ICV가 일치한 경우에는 데이터 개찬이 없다고 판정되고, 단계 S4007에서 데이터 재생 등의 여러가지 처리가 실행된다. 한편, ICV가 불일치한 경우에는 데이터 개찬이 있다고 판정되고, 데이터의 재생 등을 행하지 않고 처리를 종료한다. 이러한 처리를 실행함으로써 데이터 개찬의 방지, 개찬된 데이터의 재생이 배제된다.

<680> 이와 같이, 카테고리가 다른 콘텐츠에 대해 카테고리마다 독립한 인티그리티 체크치(ICV)를 생성하여 관리하는 구성으로 함으로써, ICV의 체크시, 또는 ICV의 변경시, 예를 들면 데이터 변경시의 새로운 인티그리티 체크치(ICV)의 생성 처리가 하나의 카테고리 내의 데이터를 대상으로 하여 실행 가능하게 되며, 다른 카테고리에 영향을 미치게 하는 일이 없다.

<681> [확장 MAC 구성]

<682> 상술된 재생 관리 파일 또는 ATRACK3 데이터 파일의 데이터 내용의 란에서 설명한 데이터 개찬 체크용의 MAC(Message Authentication Code)의 생성, 및 각 파일에 대한 저장 처리의 변형예로서, 확장 MAC의 생성, 저장 처리에 대해 이하에 설명한다.

<683> 도 42에는 확장 MAC의 생성, 저장 처리예가 도시되어 있다. 도 42에는 앞의 도 21~도 23에서 도시된 ATRACK3

데이터 파일의 일부가 도시되어 있다. 데이터 개찬 체크용의 MAC(Message Authentication Code)는, 예를 들면 ATRACK3 데이터 파일 중 몇개의 데이터 항목의 데이터에 기초하여, 앞의 도 37에서 설명한 처리에 의해 생성되는 값이고, 사전에 파일에 저장된 MAC와, 체크 시의 생성 MAC의 비교에 의해 데이터 개찬의 유무를 판정한다.

<684> 예를 들면, 도 42에 도시된 ATRACK3 데이터 파일에 저장되는 MAC는 그 MAC에 의한 개찬 체크 대상 데이터가 「INF-seq#」로부터의 복수의 데이터 항목에 설정되며, 사전에 이들 MAC 대상 데이터 항목에 기초하여 생성된 MAC가 파일에 저장되게 된다. 즉, MAC(INF-seq#‖A‖LT‖…)이다. ( ) 내의 데이터가 MAC의 대상 즉, 개찬 유무의 판정 대상이 되는 데이터이다.

<685> 그러나, ATRACK3 데이터 파일 중에는 여러가지 정보 데이터가 저장되며, 개찬 체크 대상 데이터가 증가하는 경우가 있다. 이러한 증가한 체크 대상 데이터도 포함시켜 새로운 MAC를 생성하며, 이것을 확장 MAC로서 파일 내에 저장함과 함께, 종래의 개찬 체크 대상 데이터만을 대상으로 하여 생성되는 오리지널 MAC에 대해서는 기본적으로 개찬 체크 대상 영역을 불변으로 하여 설정한 구성에 대해 설명한다.

<686> 도 42에는 먼저 설명한 INF-seq# 이하의 데이터를 개찬 체크 대상 데이터로서 설정하여, 생성되는 오리지널 MAC701이 ATRACK3 데이터 파일에 저장되어 있다.

<687> 또한, ATRACK3 데이터 파일 내의 INF 스페이스에 기록되는 몇개의 정보 중에 개찬 체크의 대상으로 해야 할 데이터가 존재하는 경우, 오리지널 MAC(701)의 MAC 생성 대상 데이터를 포함하는 데이터, 여기서는 [INF-seq#]을 포함하여 그 밖의 INF 스페이스 내의 개찬 체크 대상 데이터에 기초하여 새로운 MAC를 생성하고, 이것을 확장 MAC로서 데이터 파일중에 저장한다.

<688> 도 42에서 확장 MAC[MAC(INF)](702)는 MAC(INF-Seq#‖path‖MAC(profile)‖Others…)에 의해 생성되며, 이와 같이 확장 MAC는 오리지널 MAC의 MAC 생성 대상 데이터의 일부를 포함하며, 그 외의 개찬 체크 대상과 함께 데이터에 기초하여 생성된다.

<689> 또한, 확장 MAC의 재기록 시 즉, 확장 MAC의 대상 데이터 즉, INF 영역의 [path] 이하의 데이터의 재기록에 의해 새로운 확장 MAC를 그 재기입 데이터에 기초하여 재생성하여 재저장하는 처리를 실행할 때에는 확장 MAC에 포함되며, 또한 오리지널 MAC의 대상 데이터이기도 한 [INF-seq#]의 재기록을 행하여 새로운 확장 MAC의 생성, 저장 처리를 실행한다.

<690> 이 경우, 오리지널 MAC에 대해서도 그 대상 데이터인 [INF-seq#]의 재기록이 실행되므로, 새롭게 오리지널 MAC의 계산을 실행한다. 즉, 확장 MAC의 갱신시에는 오리지널 MAC의 재생성, 재저장 처리를 더불어 실행한다.

<691> [INF-seq#]의 재기록은 예를 들면 새로운 난수의 발생에 의한 재기입 처리, 또는 INF-seq# 데이터의 인크리먼트 처리 등에 의해 실행 가능하다.

<692> 이와 같이, 개찬 체크 대상 데이터의 증가에 대응하여 생성되는 확장 MAC의 MAC 생성 대상 데이터에 오리지널 MAC의 MAC 대상 데이터의 일부를 포함하여 쌍방의 MAC가 공통되는 MAC 대상 데이터를 존재시키며, 확장 MAC의 갱신시에는 오리지널 MAC의 재생성도 더불어 실행하는 구성으로 했으므로, 오리지널 MAC의 MAC 대상 데이터 영역을 넓히지 않고, 새로운 개찬 체크용 데이터인 예를 들면 INF 내의 데이터의 재기록 처리를 항상 오리지널 MAC에 반영시키는 것이 가능하게 된다.

<693> [기억 장치 및 재생 장치 사이에서의 EKB 처리]

<694> 이어서, 상술된 트리 구조의 키 배신 시스템을 적용한 유효화 키 블록(EKB)을 이용하여 암호화 콘텐츠의 복호처리에 적용하는 콘텐츠 키를 취득하는 구체적 처리 구성에 대하여 설명한다.

<695> 도 43에는 ATRACK3 데이터 등의 암호화 콘텐츠를 저장한, 예를 들면 메모리 스틱 등의 기억 장치(100)와, 콘텐츠 재생을 실행하는 재생 장치 A200, 재생 장치 B300가 도시되어 있다.

<696> 기억 장치(100)에는 암호화 콘텐츠로서, 도 21 등을 이용하여 설명한 ATRACK3 데이터 파일이 저장되며, 재생 장치에서 콘텐츠를 재생하기 위해서는 콘텐츠의 복호에 필요한 콘텐츠 키 Kcon을 취득하는 것이 필요하다.

<697> 우선, 재생 장치가 기억 장치로부터 콘텐츠 키를 직접 취득하는 처리 형태에 대해 도 43에 도시된 기억 장치(800)와 재생 장치 A810에서 설명한다. 우선, 기억 장치(800)와, 재생 장치 A810은 인증 처리 기능을 실행하는 상호의 제어 모듈(801, 811) 사이에서 상호 인증 처리를 실행한다. 상호 인증은, 예를 들면 먼저 설명한 도 8의 공통 키 암호 방식, 또는 공개 키 암호 방식에 의한 상호 인증 처리로서 실행한다. 이 경우, 기억 장치

(800)와, 재생 장치 A810은 각각의 제어 처리 모듈(801, 811)이 인증 처리 실행 알고리즘을 구비하고, 또한 인증 처리에 필요한 키를 저장하는 것이 필요하다.

<698> 기억 장치(800)는 재생 장치 A810과의 상호 인증의 성립 후 기억 장치(800) 내의 제어 모듈(801)에 있어서, 플래시 메모리(802)에 저장한 ATRACK3 데이터 파일로부터 기억 장치의 스토리지 키 Kstm에서 암호화된 콘텐츠 키: E(Kstm, Kcon) 또는 먼저 설명한 EKB 파일의 처리에 의해 취득 가능한 키 암호 키(KEK)로 암호화된 콘텐츠 키: E(KEK, Kcon) 중 하나를 추출하고, 복호 처리를 실행하여 콘텐츠 키 Kcon을 취득한다.

<699> 기억 장치(800)는 재생 장치 A810와의 상호 인증 시에 생성한 세션 키 Kses를 이용하여 콘텐츠 키 Kcon의 재암호화를 실행하고, 생성한 암호화 데이터: E(Kses, Kcon)를 재생 장치 A810으로 송부한다. 재생 장치 A810은 제어 모듈(811)에서 수령한 암호화 콘텐츠 키 E(Kses, Kcon)를 세션 키 Kses에서 복호하여 콘텐츠 키를 취득한다.

<700> 이상, 설명한 수법이 기억 장치측에서 콘텐츠 키를 복호하여 추출하여, 이것을 재차 세션 키로 암호화하여 재생 장치로 송부하는 수법이다.

<701> 이어서, 기억 장치측에서는 복호 처리를 실행하지 않고, 재생 장치측에서 콘텐츠 키를 취득하는 처리를 실행 형태에 대해 설명한다.

<702> 이 처리 형태를 도 43의 기억 장치(800)와 재생 장치 B830 사이의 처리로서 설명한다. 기억 장치(800)는 ATRACK3 데이터 파일 내의 유효화 키 블록(EKB) 버전(또는 제너레이션)으로부터 콘텐츠 키의 취득에도 필요한 대응 유효화 키 블록(EKB)을 특정하고, 특정된 EKB를 재생 장치 B830으로 송부한다.

<703> 재생 장치 B830은 기억 장치로부터 EKB를 수령하고, 사전에 재생 장치 내의 메모리, 예를 들면 E2PROM(ex. 플래시 메모리) 내에 저장한 디바이스 키 블록(DKB)을 이용하여 수령 EKB의 처리를 실행하고, 키 암호 키(KEK)를 취득한다.

<704> 여기서, 디바이스 키 블록(DKB)에 대해 설명한다. 도 44를 이용하여 디바이스 키 블록(DKB)의 구성을 설명한다. 상술된 바와 같이 콘텐츠 재생 장치 등의 각 디바이스는 도 44의 (a)에 도시된 트리 구조의 키 배신 구성의 말단 즉, 리프로부터 상위의 루트에 연속해 있는 각 노드의 키를 구비한다. 예를 들면 도 44의 (a)에 도시된 말단 노드의 세트 5(SET5)에 대응하는 디바이스는 리프 키로서의 K101, 노드 키로서 K10, K1로부터, 루트 키 Kroot에 이르는 키 세트, 또는 서브 카테고리 노드 키에 이르는 키 세트, 또는 카테고리 노드에 이르는 키 세트를 보유한다.

<705> 이들의 각 키는 디바이스에 있어서 암호화되어 디바이스 내의 메모리, 예를 들면 E2PROM에 저장된다. 이러한 각 디바이스에 보존되는 리프로부터 특정 노드(ex. 서브 카테고리 노드) 또는 루트까지의 키에 대응하는 키 세트의 암호화 키 세트가 디바이스 키 블록(DKB)이다.

<706> 디바이스 키 블록(DKB)의 데이터 구성예가 도 44의 (b)에 도시되어 있다. 도 44의 (b)에 도시된 바와 같이 DKB는 노드 키, 및 루트 키를 리프 키로 암호화한 데이터와, 리프 키를 디바이스(ex. 재생 장치)의 스토리지 키: Kstd에서 암호화한 데이터를 구비한 암호화 키 블록으로서 구성된다. 디바이스(ex. 재생 장치)는 이 디바이스 키 블록(DKB) 내의 Enc(Kstd, Kleaf)를 자신의 스토리지 키: Kstd를 이용하여 복호하고, 리프 키 Kleaf를 취득하고, 또 취득한 리프키 Kleaf를 이용하여 고위의 암호화 노드 키, 암호화 루트 키를 직접 복호하는 것이 가능하게 되고, EKB의 하위 키로부터 순차 복호하여 상위 키를 취득해 가는 처리의 생략이 가능하게 된다. 또 디바이스 키 블록(DKB)에는 리프의 식별자인 리프 ID를 포함한다.

<707> 디바이스 고유의 스토리지 키는 각 세트(디바이스)마다 다른 키이고, 사전에 디바이스 내의 시큐어 메모리(ex. SAM) 내에 저장하거나, 또는 리프 ID에 기초하여 구하는 것이 가능한 구성으로 해도 좋다. 즉, 디바이스의 제어 모듈(암호 처리부)에 있어서, 리프 ID에 기초하여 생성하는 구성으로 해도 좋다. 구체적으로는, 소정의 세트 단위로 공통으로 저장된 마스터 키 Kmas에 기초하여 리프 ID 에 대한 해시를 적용하고, Kstd = hash(Kmas, 리프 ID)로서 구하는 구성으로 해도 좋다.

<708> 도 43으로 되돌아가 콘텐츠 키의 취득 처리의 설명을 계속한다. 기억 장치(800)로부터 유효화 키 블록(EKB)을 수령한 재생 장치 B830은, 제어 모듈(831)에 있어서 메모리(832)에 저장한 디바이스 키 블록(DKB)의 복호에 의해 얻어지는 노드 키, 루트 키 등을 적용하여 EKB에 의해 암호화된 키 암호화 키(KEK)를 취득한다. EKB의 처리 수법은 먼저 도 5 또는 도 9를 이용하여 설명한 바와 같은 수법이다.

<709> 재생 장치 B830은, 유효화 키 블록(EKB)의 처리에 의해 취득한 키 암호화 키(KEK)를 이용하고, 또한 기억 장치

(800)로부터 수령한 암호화 콘텐츠 키: E(KEK, Kcon)의 복호 처리를 실행하여 콘텐츠 키를 취득한다.

<710> 또한, 도 43의 재생 장치 B830의 메모리(E2PROM: 832)에 저장된 이니셜 EKB는 디바이스(재생 장치 B830)에 당초부터 저장되는 간략화한 EKB 파일로서, 예를 들면 상술된 도 11을 이용한 설명 중에 기재한 카테고리 노드에 있어서, 하나의 카테고리 노드(예를 들면, 카테고리 = 메모리 스틱)의 하위에 접속되는 리프에 대응하는 디바이스에 공통으로 저장되는 암호화 키 블록이다.

<711> 예를 들면 카테고리 노드가 구비하는 키가 K01이면 K01로 암호화된 루트 키: Enc(K01, Kroot)가 이니셜 EKB로서 저장된다. 디바이스는 이니셜 EKB의 처리에 의해 루트 키를 취득하는 것이 가능하게 되고, 예를 들면 루트 키에 의해 암호화된 키 암호화 키(KEK)를 저장한 EKB를 수령한 경우에는 이니셜 EKB로부터 얻은 루트 키를 이용하여 키 암호화 키(KEK)를 취득하는 것이 가능하게 된다.

<712> 또한, 이니셜 EKB는 하나의 카테고리 노드에 속하는 디바이스에 공통으로 제공하는 구성으로 하는 형태에 한하지 않고, 복수의 카테고리 노드에 공통으로 구성해도 좋다. 예를 들면 메모리 스틱의 카테고리 노드의 노드 키 K01, 콘텐츠 재생 기능을 구비한 PC의 카테고리 노드의 노드 키를 K10, 네트워크 대응의 형태 재생 장치의 카테고리 노드의 노드 키를 K11로 했을 때, 이들의 각 디바이스에 사전에 Enc(K01, Kroot), Enc(K10, Kroot), Enc(K11, Kroot)의 3 종류의 암호화 루트 키를 저장한 이니셜 EKB를 설정하여 출하함으로써, 각각의 다른 디바이스에 있어서 공통으로 이용 가능한 암호화 콘텐츠의 배신을 행하는 것이 가능하게 된다.

<713> 도 45에는, 재생 장치의 메모리(ex. E2PROM)에 디바이스 키 블록(DKB)과, 이니셜 EKB로서 자기 녹음, 자기 재생용의 유효화 키 블록(EKB)을 저장한 구성예가 도시되어 있다. 또한, 도 46에는 이들의 키 블록을 이용한 콘텐츠 키의 취득 처리예가 도시되어 있다.

<714> 도 45의 구성에 대해 설명한다. 디바이스(ex. 기록 재생기)는 도 45의 (a)의 리프에 대응하는 디바이스로서, 트리 구성의 제8단째에 구성되는 카테고리 노드 Kn8의 카테고리에 속하는 디바이스이다. 디바이스에는 (b)에 도시된 Enc(Kstd, Kleaf)~Enc(Kleaf, Kn8)의 디바이스 키 블록(DKB)이 저장된다. 이 구성은, 먼저 설명한 DKB와 마찬가지이지만, 리프 키에 의해 직접 암호화되어 저장된 데이터는 리프 키의 바로 상측의 노드 키 Kn47로부터 카테고리 노드 키인 Kn8까지의 키로서 구성된다.

<715> 또한, 디바이스는 자기 녹음, 재생용의 유효화 키 블록(EKB)을 보유하며, 자기의 디바이스에서의 콘텐츠 녹음, 재생시에는 이 자기 녹음, 재생용의 유효화 키 블록(EKB)과 디바이스 키 블록(DKB)과의 처리에 의해 콘텐츠 키 Kcon을 취득하여, 콘텐츠의 복호, 암호화를 실행한다.

<716> 도 46에는, 도 45의 (b)의 DKB, EKB를 구비한 디바이스에서의 콘텐츠 키의 취득 처리에 있어서 실행하는 단계가 도시되어 있다. 우선 단계 S4601에서, 디바이스는 리프 ID에 기초하여 스토리지 키 Kstd를 추출한다. 스토리지 키 Kstd는 리프 ID에 기초하여 디바이스 내의 시큐어 메모리로부터 추출하거나, 또는 상술한 바와 같이 마스터 키 Kmas와 리프 ID에 기초하여 산출한다.

<717> 이어서, S4602에서 스토리지 키 Kstd에 기초하여 디바이스 키 블록(DKB)의 처리 즉, Enc(Kstd, Kleaf)의 복호를 실행하고, 리프 키를 구한다. 이어서, S4603에 있어서, 리프 키 Kleaf에 기초하여 디바이스 키 블록(DKB)의 처리 즉, Enc(Kleaf, Kn8)의 복호를 실행하고, 카테고리 노드 키를 구한다. DKB는 리프 키에 의해 직접 암호화된 노드 키가 저장되어 있으므로, 고위의 노드 키를 직접 리프 키에 의한 복호 처리에 의해 취득하는 것이 가능하게 된다.

<718> 이어서, 단계 S4604에서 노드 키 Kn8로부터 EKB 처리를 실행하고, 순차 고위의 노드 키를 구하여, 최상위 키인 루트 키를 산출한다. 이어서, 단계 S4605에서, 유효화 키 블록(EKB)의 처리에 의해 구한 루트 키 Kroot를 이용하여 Enc(Kroot, KEK)의 복호 처리를 실행하여 키 암호화 키 KEK를 구한다. 마지막으로 단계 S4606에서 취득한 키 암호화 키 KEK를 이용하여 콘텐츠 데이터에 부수한 데이터 내에 저장된 Enc(KEK, Kcon)의 복호 처리를 실행하여 콘텐츠 키 Kcon을 취득한다.

<719> 도 45의 (b)에 도시된 유효화 키 블록(EKB)은 자기 녹음 재생용의 EKB 이지만, 여러가지 콘텐츠를 디바이스에 다운로드할 때에, 그 콘텐츠에 대응하는 EKB를 더불어 다운로드하고, 콘텐츠에 대응시켜 EKB를 메모리에 저장하며, 콘텐츠의 재생 시에 다운로드한 콘텐츠 대응의 EKB에 대해 도 46의 처리를 실행하는 것도 가능하다. 또한, 도 45의 (b)에 도시된 디바이스 키 블록(DKB)은 상위로부터 8단의 노드 Kn8의 노드 키까지를 직접 리프 키로 암호화한 데이터를 DKB의 암호화 키 데이터로 한 구성이지만, 저장하는 노드 키는 또한 상위, 또는 하위까지의 노드 키로 해도 좋다.

<720> 이상, 특정한 실시예를 참조하면서, 본 발명에 대해 상해해 왔다. 그러나, 본 발명의 요지를 일탈하지 않은 범위에서 당 업자가 그 실시예의 수정이나 대용을 할 수 있는 것은 자명하다. 즉, 예시라는 형태로 본 발명을 개시한 것으로, 한정적으로 해석되어서는 안된다. 본 발명의 요지를 판단하기 위해서는, 첫머리에 기재한 특허청구의 범위의 란을 참작해야 한다.

발명의 효과

<721> 이상, 설명한 바와 같이, 본 발명의 데이터 처리 장치 및 방법에 따르면, 복수의 디바이스를 리프로서 구성한 트리의 루트로부터 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 포함하는 패스 상의 갱신 키, 및 하위 키에 의한 상위 키의 암호화 처리 데이터를 포함하는 유효화 키 블록(EKB)에 의해 암호화된 인증 키를 제공하며, 선택된 정당한 디바이스에 있어서만 인증 키를 취득 가능한 구성으로 한 시큐러티가 높은 인증 키 배신 시스템이 실현된다.

<722> 또한, 본 발명의 데이터 처리 장치 및 방법에 따르면, 데이터 처리 장치는 기억 장치가 상호 인증 처리의 실행 기능을 갖지 않은 경우에도 상기 데이터 처리 장치에 구성한 가상 메모리 디바이스와의 상호 인증 처리의 성립을, 기억 장치로부터의 데이터 재생 처리 또는 기억 장치에 대한 데이터 기록 처리의 실행 조건으로 한다. 또한, 정당한 라이센스를 갖는 데이터 처리 장치에서만 복호 가능한 유효화 키 블록(EKB)에 의해 인증 키를 제공하는 구성에 의해서, 부정한 데이터 처리 장치에서는 정당한 인증 키가 취득되지 않고, 가상 메모리 디바이스와의 인증 성립이 불가능해져, 콘텐츠 이용의 제한이 가능한 시스템이 실현된다.

도면의 간단한 설명

<1> 도 1은 본 발명의 데이터 처리 장치의 사용 개념을 설명하는 도면.

<2> 도 2는 본 발명의 데이터 처리 장치의 시스템 구성예 및 데이터 경로예를 도시하는 도면.

<3> 도 3은 본 발명의 데이터 처리 장치에서의 각종 키, 데이터의 암호화 처리에 대하여 설명하는 트리 구성도.

<4> 도 4는 본 발명의 데이터 처리 장치에서의 각종 키, 데이터의 배포에 사용되는 유효화 키 블록(EKB)의 예를 도시하는 도면.

<5> 도 5는 본 발명의 데이터 처리 장치에서의 콘텐츠 키의 유효화 키 블록(EKB)을 사용한 배포예와 복호 처리예를 도시하는 도면.

<6> 도 6은 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)의 포맷예를 도시하는 도면.

<7> 도 7은 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)의 태그의 구성을 설명하는 도면.

<8> 도 8은 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)과, 콘텐츠 키, 콘텐츠를 더불어 배신하는 데이터 구성예를 도시하는 도면.

<9> 도 9는 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)과, 콘텐츠 키, 콘텐츠를 더불어 배신한 경우의 디바이스에서의 처리예를 도시하는 도면.

<10> 도 10은 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)과 콘텐츠를 기록 매체에 저장한 경우의 대응에 대해 설명하는 도면.

<11> 도 11은 본 발명의 데이터 처리 장치에서의 계층 트리 구조의 카테고리 분류의 예를 설명하는 도면.

<12> 도 12는 본 발명의 데이터 처리 장치에서의 간략화 유효화 키 블록(EKB)의 생성 과정을 설명하는 도면.

<13> 도 13은 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)의 생성 과정을 설명하는 도면.

<14> 도 14는 본 발명의 데이터 처리 장치에서의 간략화 유효화 키 블록(EKB)을 설명하는 도면.

<15> 도 15는 본 발명의 데이터 처리 장치에서의 재생 장치와 기억 장치의 구성을 도시하는 블록도.

<16> 도 16은 본 발명의 데이터 처리 장치에서의 기억 장치 내의 기억 유닛에 기억되어 있는 데이터를 설명하는 도면.

<17> 도 17은 본 발명의 데이터 처리 장치에서의 기억 장치의 플래시 메모리에 기억되는 데이터를 설명하는 도면.

<18> 도 18은 본 발명의 데이터 처리 장치에서의 재생 관리 파일의 데이터 구성을 개략적으로 도시하는 도면.

<19> 도 19는 본 발명의 데이터 처리 장치에서의 데이터 파일의 데이터 구성을 개략적으로 도시하는 도면.

<20> 도 20은 본 발명의 데이터 처리 장치에서의 재생 관리 파일의 데이터 구성을보다 상세히 도시하는 도면.

<21> 도 21은 본 발명의 데이터 처리 장치에서의 데이터 파일의 데이터 구성을 보다 상세히 도시하는 도면.

<22> 도 22는 본 발명의 데이터 처리 장치에서의 데이터 파일의 속성 헤더의 일부를 도시하는 도면.

<23> 도 23은 본 발명의 데이터 처리 장치에서의 데이터 파일의 속성 헤더의 일부를 도시하는 도면.

<24> 도 24는 본 발명의 데이터 처리 장치에서의 모드의 종류와 각 모드에 있어서의 녹음 시간 등을 도시하는 도면.

<25> 도 25는 본 발명의 데이터 처리 장치에서의 복사 제어 정보를 설명하는 도면.

<26> 도 26은 본 발명의 데이터 처리 장치에서의 데이터 파일의 속성 헤더의 일부를 도시하는 도면.

<27> 도 27은 본 발명의 데이터 처리 장치에서의 데이터 파일의 각 데이터 블록의 헤더를 도시하는 개략 선도.

<28> 도 28은 본 발명의 데이터 처리 장치에서의 데이터 기록 처리 플로우를 도시하는 도면.

<29> 도 29는 본 발명의 데이터 처리 장치에서 적용 가능한 상호 인증 처리를 도시하는 도면.

<30> 도 30은 본 발명의 데이터 처리 장치에서의 데이터 재생 처리 플로우를 도시하는 도면.

<31> 도 31은 본 발명의 데이터 처리 장치에서의 배신 키 허가 정보 파일의 포맷을 도시하는 도면.

<32> 도 32는 본 발명의 데이터 처리 장치에서의 데이터 저장 형태를 도시하는 도면.

<33> 도 33은 본 발명의 데이터 처리 장치에서의 키 유효화 블록(EKB)을 사용한 데이터 복호 처리 플로우를 도시하는 도면.

<34> 도 34는 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)과, 인증 키를 더불어 배신하는 데이터 구성과, 디바이스에서의 처리예를 도시하는 도면.

<35> 도 35는 본 발명의 데이터 처리 장치에서의 유효화 키 블록(EKB)과, 인증 키를 더불어 배신하는 데이터 구성과, 디바이스에서의 처리예를 도시하는 도면.

<36> 도 36은 본 발명의 데이터 처리 장치에서의 가상 메모리 카드를 적용한 인증 처리 시퀀스를 도시하는 도면.

<37> 도 37은 본 발명의 데이터 처리 장치에서 적용 가능한 엔티티 체크치(ICV)의 생성에 사용하는 MAC치 생성예를 도시하는 도면.

<38> 도 38은 본 발명의 데이터 처리 장치에서의 엔티티 체크치(ICV)의 저장 형태를 설명하는 도면.

<39> 도 39는 본 발명의 데이터 처리 장치에서의 MAC치를 저장하는 시퀀스 페이지 포맷을 도시하는 도면.

<40> 도 40은 본 발명의 데이터 처리 장치에서의 ICV를 저장하는 풀 페이지 포맷을 도시하는 도면.

<41> 도 41은 본 발명의 데이터 처리 장치에서의 ICV 체크 처리 플로우를 도시하는 도면.

<42> 도 42는 본 발명의 데이터 처리 장치에서 착용 가능한 확장 MAC의 생성, 저장 처리를 설명하는 도면.

<43> 도 43은 본 발명의 데이터 처리 장치에서의 키 유효화 블록(EKB)을 이용한 콘텐츠 키의 취득 처리 형태를 설명하는 도면.

<44> 도 44는 본 발명의 데이터 처리 장치에서 사용되는 디바이스 키 블록(DKB)의 구성에 대하여 설명하는 도면.

<45> 도 45는 본 발명의 데이터 처리 장치에서의 디바이스 키 블록(DKB), 키 유효화 블록(EKB)의 저장 구성예를 도시하는 도면.

<46> 도 46은 본 발명의 데이터 처리 장치에서의 디바이스 키 블록(DKB), 키 유효화 블록(EKB)을 이용한 콘텐츠 키의 취득 처리 형태를 설명하는 도면.

<47> <도면의 주요 부분에 대한 부호의 설명>

<48> 10 : 콘텐츠 배신 수단

<49> 11 : 인터넷

<50> 12 : 위성 방송

<51> 13 : 전화 회선

<52> 14 : 미디어

<53> 20 : 데이터 처리 수단

<54> 21 : 퍼스널 컴퓨터(PC)

<55> 22 : 포터블 디바이스(PD)

<56> 23 : 휴대 전화, PDA

<57> 24 : 기록 재생기, 게임 단말

<58> 25 : 재생 장치

<59> 30 : 기억 수단

<60> 100 : 퍼스널 컴퓨터(PC)

<61> 200 : 재생 장치

<62> 300 : 기억 장치

<63> 601 : 버전

<64> 602 : 깊이

<65> 603 : 데이터 포인터

<66> 604 : 태그 포인터

<67> 605 : 서명 포인터

<68> 606 : 데이터부

<69> 607 : 태그부

<70> 608 : 서명

<71> 33, 43 : 제어 모듈

<72> 50, 60 : 난수 발생 유닛

<73> 51, 61 : 기억 유닛

<74> 52, 62 : 키 생성/연산 유닛

<75> 53, 63 : 상호 인증 유닛

<76> 54, 74 : 암호화/복호 유닛

<77> 55, 65 : 제어 유닛

<78> 34 : 플래시 메모리

<79> 44 : 편집 모듈

<80> 45 : 압축/신장 모듈

<81> 46 : 스피커

<82> 49 : 메모리

<83> 800 : 기억 장치

<84> 801 : 제어 모듈

<85> 802 : 플래시 메모리

<86> 810 : 재생 장치 A

<87> 811 : 제어 모듈

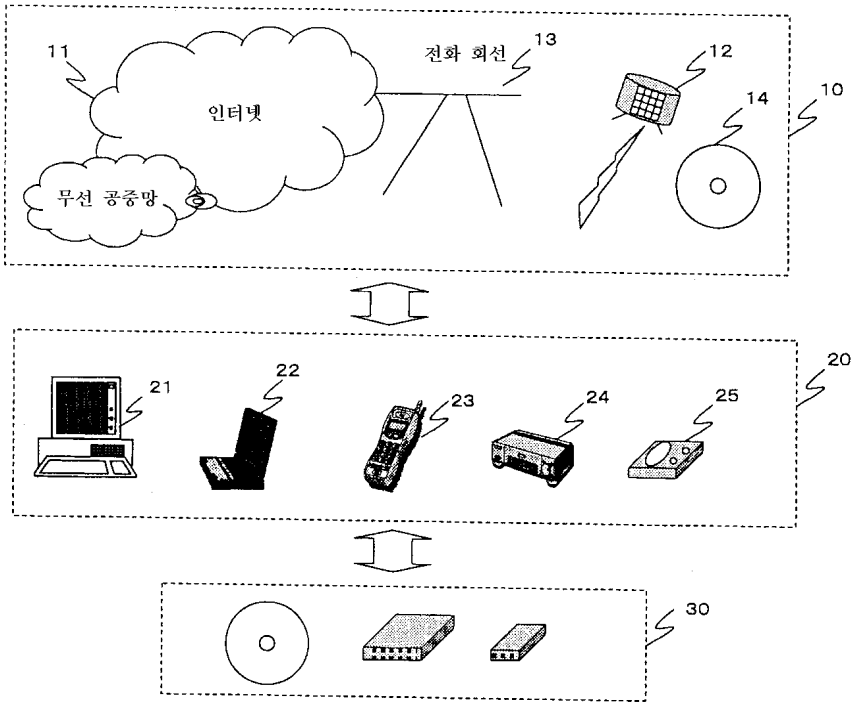
<88> 830 : 재생 장치 B

<89> 831 : 제어 모듈

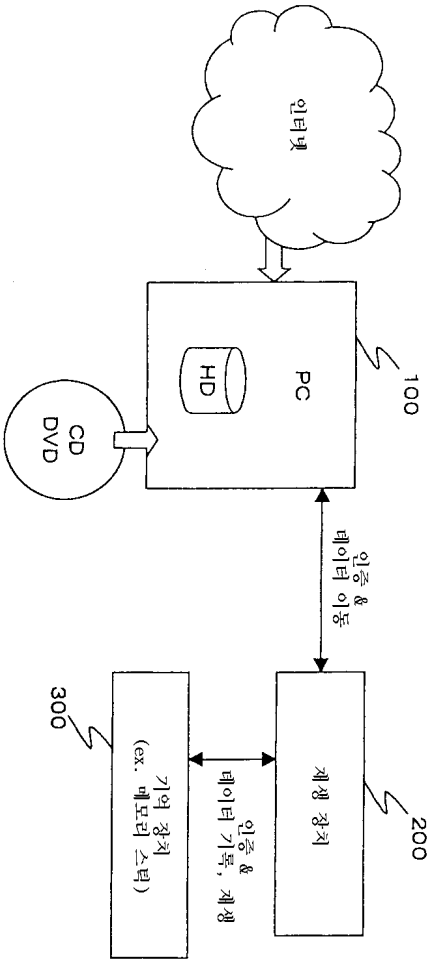
<90> 832 : 메모리

도면

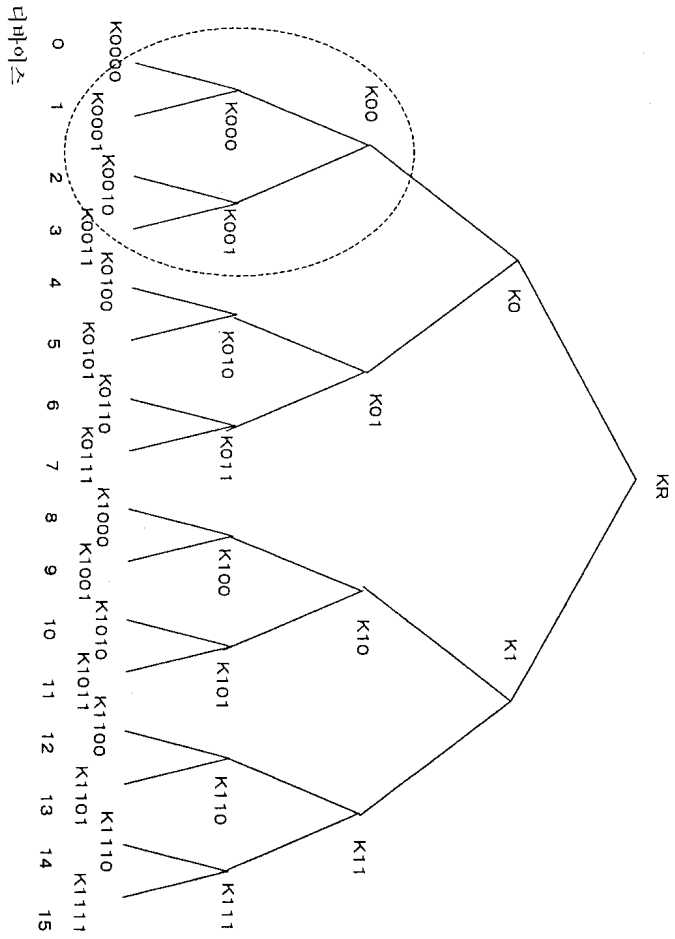
도면1



도면2


인터넷
PC
HD
CD
DVD
100
인증 &
데이터 이동
재생 장치
200
기억 장치
(ex. 메모리 스틱)
300
인증 &
데이터 기록, 재생

도면3


KR
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
K0000
K0001
K0010
K0011
K0100
K0101
K0110
K0111
K1000
K1001
K1010
K1011
K1100
K1101
K1110
K1111
디바이스
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

도면4

(A) 유효화 키 블럭(EKB:Enabling Key Block) 예 1

디바이스0, 1, 2에 버전:t의 노드키를 송부

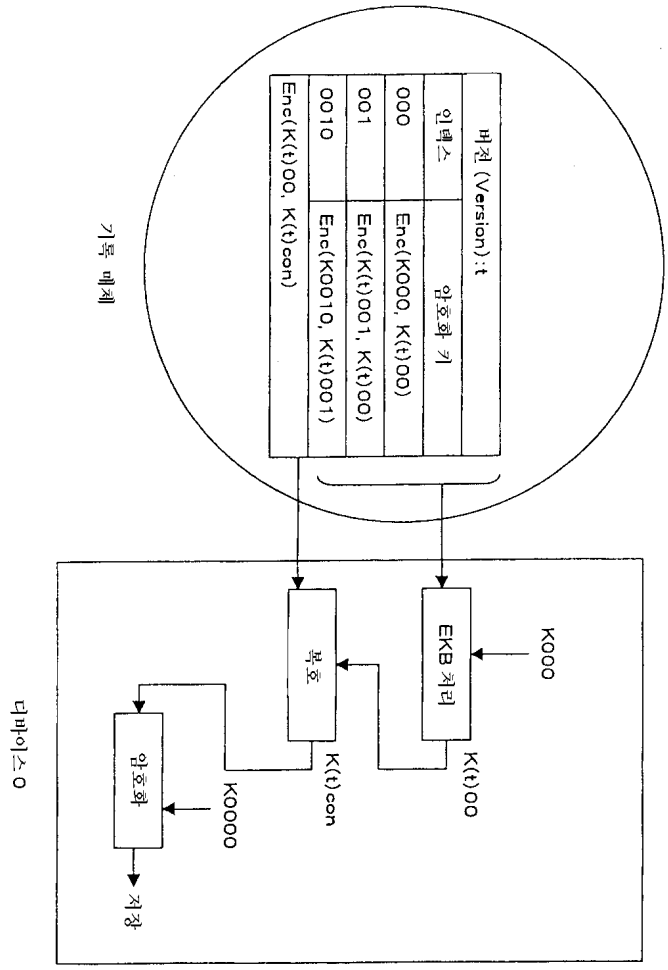
버전 (Version):t	
인덱스	암호화 키
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

(B) 유효화 키 블럭(EKB:Enabling Key Block) 예 2

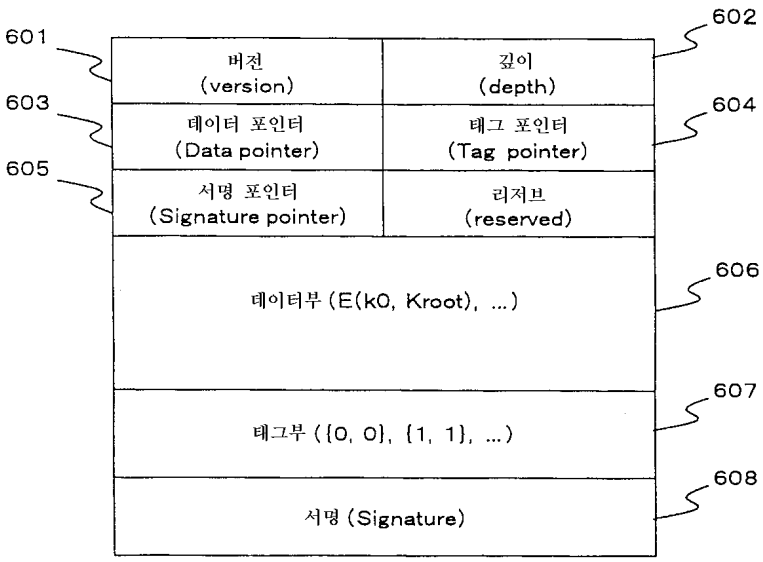
디바이스0, 1, 2에 버전:t의 노드키를 송부

버전 (Version):t	
인덱스	암호화 키
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

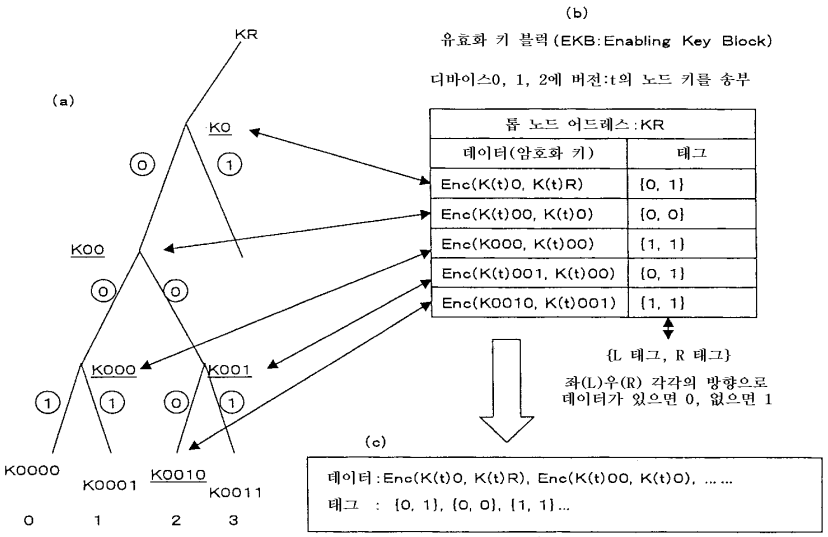
도면5


기록 매체
버전 (Version) : t
인덱스
암호화 키
000
Enc(K000, K(t)00)
001
Enc(K(t)001, K(t)00)
0010
Enc(K0010, K(t)001)
Enc(K(t)00, K(t)con)
디바이스 0
복호
EKB 처리
K000
K(t)con
K(t)00
암호화
K0000
저장

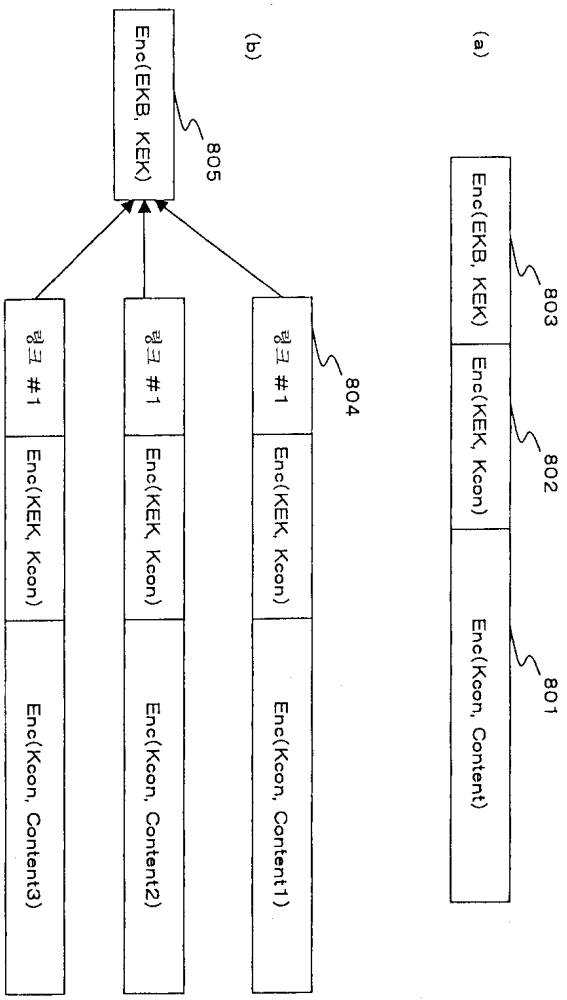
도면6


601
602
버전
(version)
깊이
(depth)
603
604
데이터 포인터
(Data pointer)
태그 포인터
(Tag pointer)
605
서명 포인터
(Signature pointer)
리저브
(reserved)
606
데이터부 (E(k0, Kroot), ...)
607
태그부 ({0, 0}, {1, 1}, ...)
608
서명 (Signature)

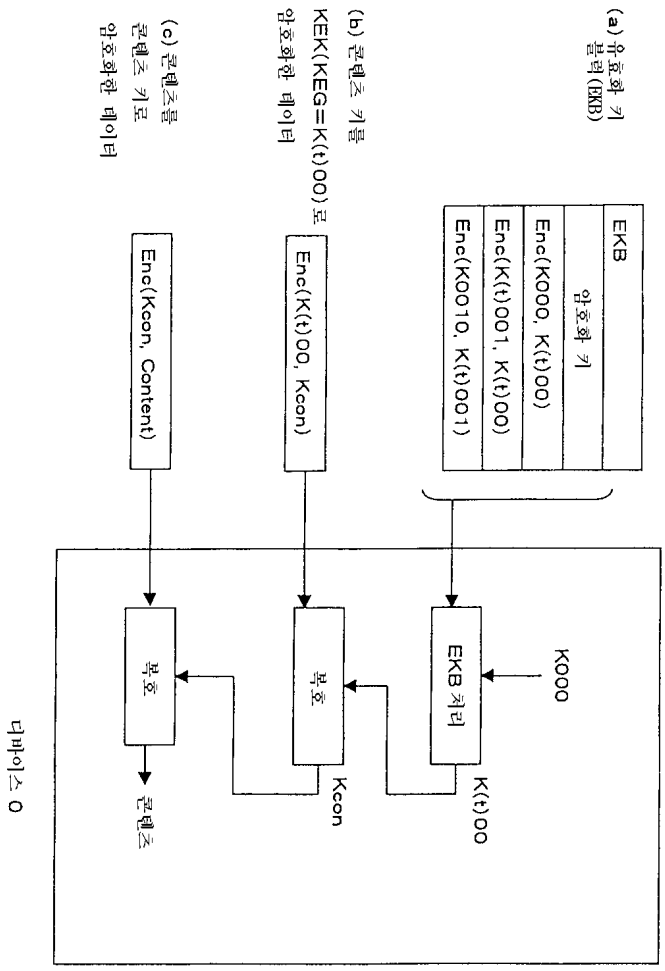
도면7


(a)
KR
K0
K00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
(b)
유효화 키 블럭 (EKB:Enabling Key Block)
디바이스0, 1, 2에 버전:t의 노드 키를 송부
톱 노드 어드레스 : KR
데이터(암호화 키)
태그
Enc(K(t)0, K(t)R)
{0, 1}
Enc(K(t)00, K(t)0)
{0, 0}
Enc(K000, K(t)00)
{1, 1}
Enc(K(t)001, K(t)00)
{0, 1}
Enc(K0010, K(t)001)
{1, 1}
{L 태그, R 태그}
좌(L)우(R) 각각의 방향으로
데이터가 있으면 0, 없으면 1
(c)
데이터 : Enc(K(t)0, K(t)R), Enc(K(t)00, K(t)0), ……
태그 : {0, 1}, {0, 0}, {1, 1}…

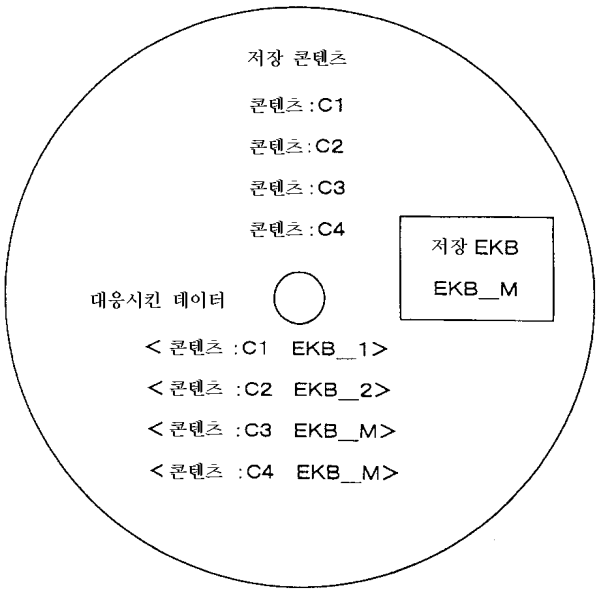
도면8


(a)
Enc(EKB, KEK)
803
Enc(KEK, Kcon)
802
Enc(Kcon, Content)
801
(b)
Enc(EKB, KEK)
805
링크 #1
Enc(KEK, Kcon)
Enc(Kcon, Content3)
링크 #1
Enc(KEK, Kcon)
Enc(Kcon, Content2)
링크 #1
804
Enc(KEK, Kcon)
Enc(Kcon, Content1)

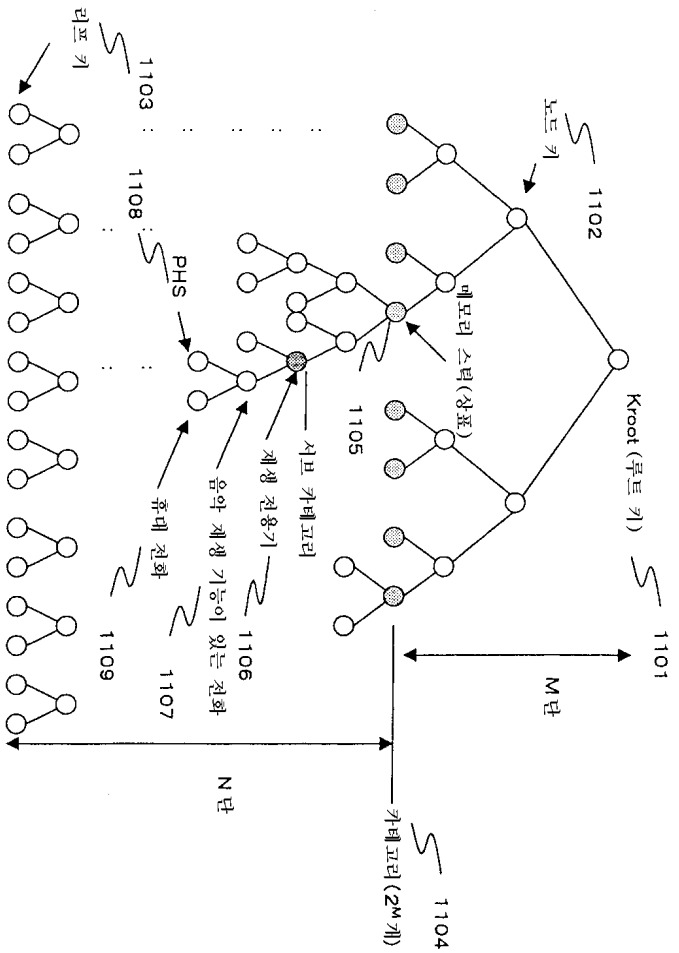
도면9


(a) 유효화 키 블럭 (EKB)
EKB
암호화 키
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
(b) 콘텐츠 키를 KEK(KEG=K(t)00)로 암호화한 데이터
Enc(K(t)00, Kcon)
(c) 콘텐츠를 콘텐츠 키로 암호화한 데이터
Enc(Kcon, Content)
디바이스 0
K000
EKB 처리
K(t)00
복호
Kcon
복호
콘텐츠

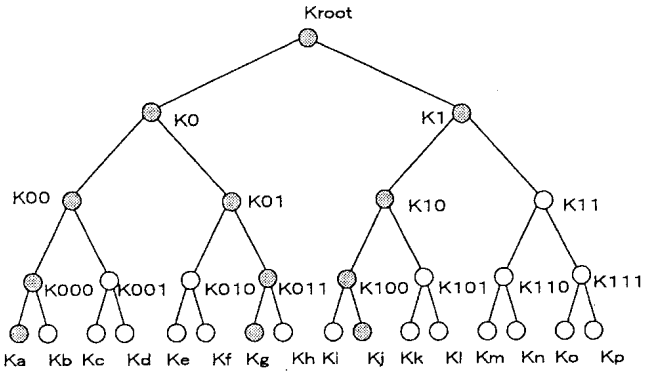
도면10


저장 콘텐츠
콘텐츠 : C1
콘텐츠 : C2
콘텐츠 : C3
콘텐츠 : C4
저장 EKB
EKB__M
대응시킨 데이터
< 콘텐츠 : C1 EKB__1>
< 콘텐츠 : C2 EKB__2>
< 콘텐츠 : C3 EKB__M>
< 콘텐츠 : C4 EKB__M>
기록 매체

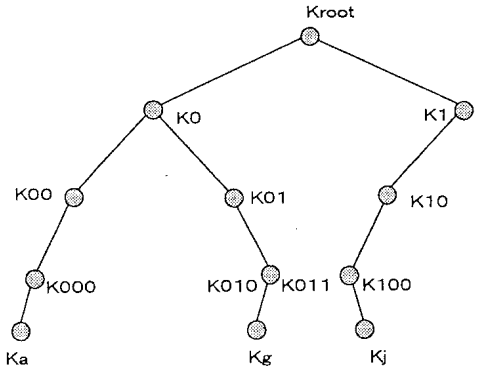
도면11


Kroot (루트 키)
1101
M 단
노드 키
1102
메모리 스틱(상표)
1105
서브 카테고리
재생 전용기
1106
음악 재생 기능이 있는 전화
1107
휴대 전화
1109
PHS
1108
리프 키
1103
N 단
카테고리 (2^M개)
1104

도면12


Kroot
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
Ka
Kb
Kc
Kd
Ke
Kf
Kg
Kh
Ki
Kj
Kk
Kl
Km
Kn
Ko
Kp

(a)

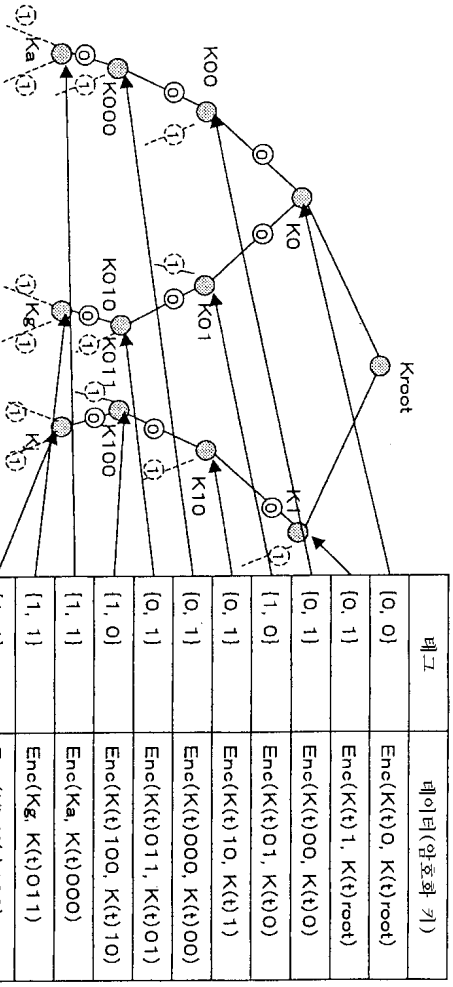

Kroot
K0
K1
K00
K01
K10
K000
K010
K011
K100
Ka
Kg
Kj

(b)

도면13

유효화 키 블럭(EKB: Enabling Key Block)을 이용한
디바이스 Ka, Kg, Kj 로의 버전 t의 콘텐츠 키 송부 처리

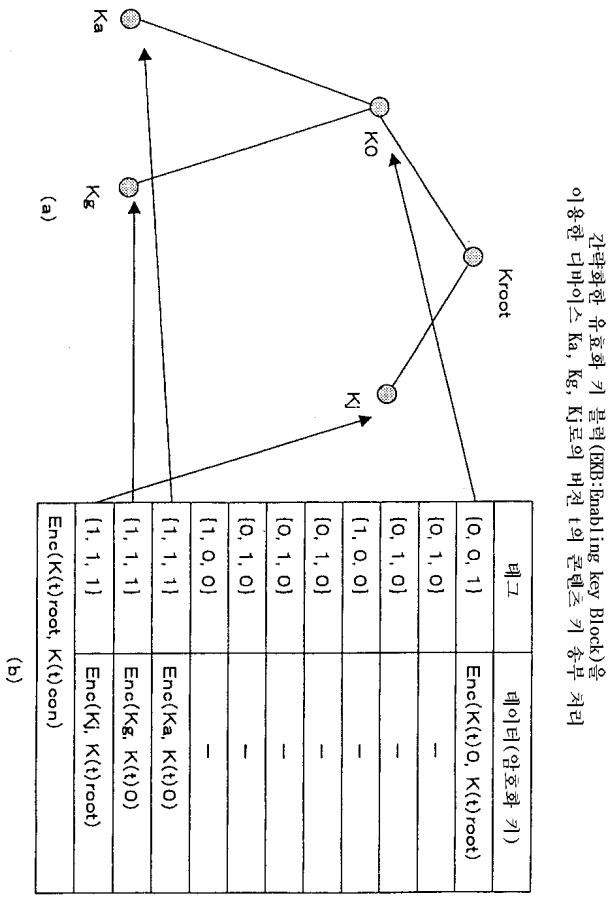
(a)



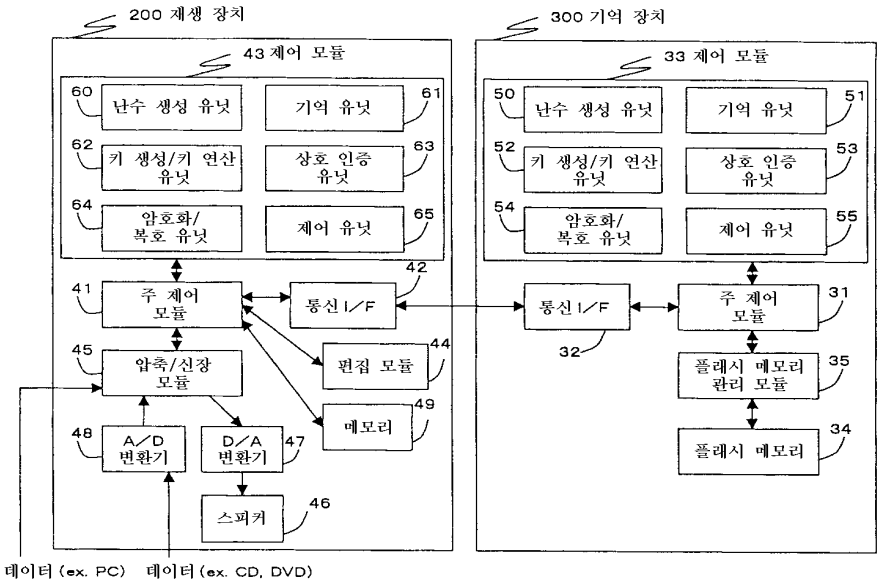
(b)

태그	데이터(암호화 키)
{0, 0}	Enc(K(t)0, K(t)root)
{0, 1}	Enc(K(t)1, K(t)root)
{0, 1}	Enc(K(t)00, K(t)0)
{1, 0}	Enc(K(t)01, K(t)0)
{0, 1}	Enc(K(t)10, K(t)1)
{0, 1}	Enc(K(t)000, K(t)00)
{0, 1}	Enc(K(t)011, K(t)01)
{1, 0}	Enc(K(t)100, K(t)10)
{1, 1}	Enc(Ka, K(t)000)
{1, 1}	Enc(Kg, K(t)011)
{1, 1}	Enc(Kj, K(t)100)
Enc(K(t)root, K(t)con)	

도면14

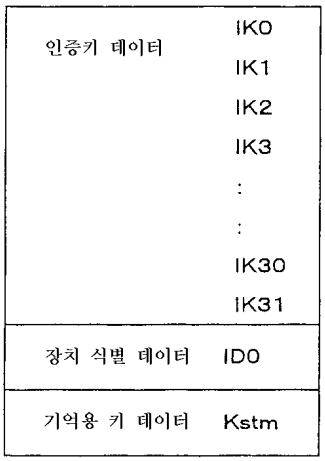

Ka
Kg
KO
Kroot
Ki
(a)
(b)
간략화한 유효화 키 블럭(EKB:Enabling key Block)을
이용한 디바이스 Ka, Kg, Kj로의 버전 t의 콘텐츠 키 송부 처리
태그
데이터(암호화 키)
{0, 0, 1}
Enc(K(t)O, K(t)root)
{0, 1, 0}
—
{0, 1, 0}
—
{1, 0, 0}
—
{0, 1, 0}
—
{0, 1, 0}
—
{0, 1, 0}
—
{1, 0, 0}
—
{1, 1, 1}
Enc(Ka, K(t)O)
{1, 1, 1}
Enc(Kg, K(t)O)
{1, 1, 1}
Enc(Ki, K(t)root)
Enc(K(t)root, K(t)con)

도면15

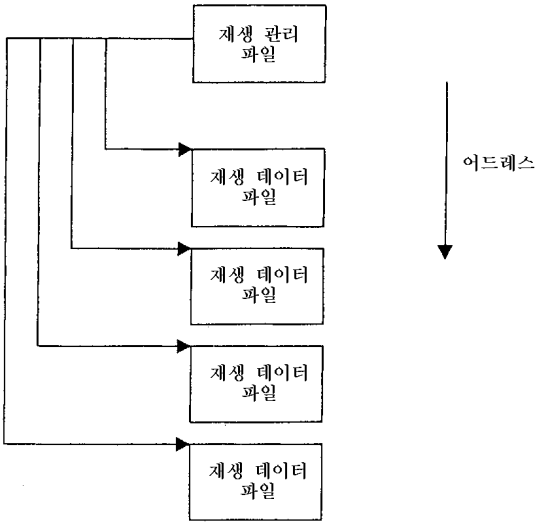

200 재생 장치
43 제어 모듈
60 난수 생성 유닛
61 기억 유닛
62 키 생성/키 연산 유닛
63 상호 인증 유닛
64 암호화/복호 유닛
65 제어 유닛
41 주 제어 모듈
42 통신 I/F
45 압축/신장 모듈
44 편집 모듈
48 A/D 변환기
47 D/A 변환기
49 메모리
46 스피커
데이터 (ex. PC)
데이터 (ex. CD, DVD)
300 기억 장치
33 제어 모듈
50 난수 생성 유닛
51 기억 유닛
52 키 생성/키 연산 유닛
53 상호 인증 유닛
54 암호화/복호 유닛
55 제어 유닛
32 통신 I/F
31 주 제어 모듈
35 플래시 메모리 관리 모듈
34 플래시 메모리

도면16

기억 장치의 기억 유닛에 저장되는 데이터

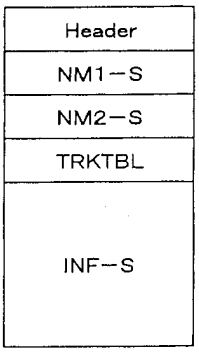


도면17

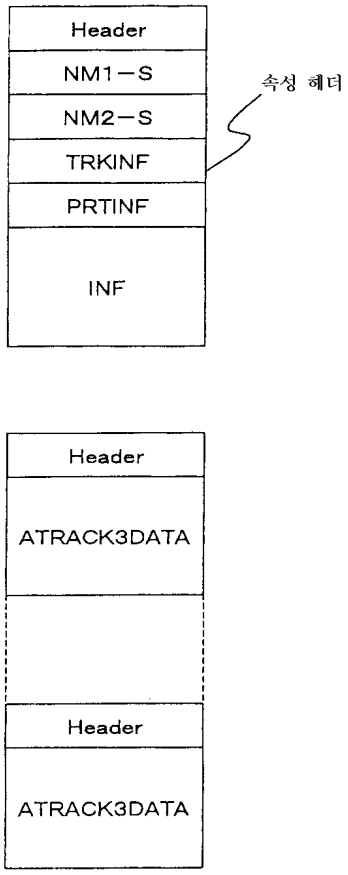


도면18

재생 관리 파일



도면19


Header
NM1-S
NM2-S
TRKINF
PRTINF
INF
속성 헤더
Header
ATRACK3DATA
Header
ATRACK3DATA

도면20

재생 관리 파일

A

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	BLKID-TL0				Reserved		Mcode		REVISION				Reserved			
0x0010	SN1C+L		SN2C+L		SINFSIZE		T-TRK		VerNo.		Reserved					

B

0x0020	NM1-S(256)							
0x0120	NM2-S(512)							
0x0310								
0x0320	Reserved(4)			EKB version		E(Kstm, Kcon)		
0x0330	E(KEKn, Kcon)					C_MAC[0]		
0x0340	Reserved(8)				Reserved(3)	MGR	S-YMDhms	
0x0350	TRK-001	TRK-002	TRK-003	TRK-004	TRK-005	TRK-006	TRK-007	TRK-008
0x0360	TRK-009	TRK-010	TRK-011	TRK-012	TRK-013	TRK-014	TRK-015	TRK-016
0x0660	TRK-393	TRK-394	TRK-395	TRK-396	TRK-397	TRK-398	TRK-399	TRK-400
0x0670	INF-S(14720)							
0x3FFF	BLKID-TL0		Reserved		Mcode		REVISION	Reserved

C

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
INF	0X00	ID	0X00	SIZE		Mcode		C+L		Reserved		DATA가변 길이			

도면21

ATRAC3 데이터 파일

	0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000	BLKID-HD0 | Reserved | Mcode | Reservrd | BLOCK SERIAL
0x0010	N1C+L | N2C+L | INFSIZE | T-PRT | T-SU | INX | XT
0x0020	NM1-S(256)
0x0120	NM2-S(512)
0x0310	
0x0320	Reserved(3) | EKi | EKB version | E(Kstm, Kcon)
0x0330	E(KEKn, Kcon) | C_MAC[n]
0x0340	Reserved(8) | INF_seq# | A | LT | FNo
0x0350	MG(D)SERIAL-nnn(Upper) | MG(D)SERIAL-nnn(Lower)
0x0360	CONNUM | YMDhms-S | YMDhms-E | XCC | CT | CC | CN
0x0370	PRTSIZE | PRTKEY | Reserved(8)
0x0380	CONNUM0 | PRTSIZE(0x0388) | PRTKEY
0x0390	Reserved(8) | CONNUM0
	INF(0x0400)
0x3FFF	BLKID-HD0 | Reserved | Mcode | Reservrd | BLOCK SERIAL
0x4000	BLKID-A3D | Reserved | Mcode | CONNUM0 | BLOCK SERIAL
0x4010	BLOCKSEED | INITIALIZATION VECTOR
0x4020	SU-000(Nbyte=384byte)
0x41A0	SU-001(Nbyte)
0x4320	SU-002(Nbyte)
0x04A0	SU-041(Nbyte)
0x7DA0	Reserved(Nbyte=208byte)
0x7F20	BLK SEED
0x7FF0	BLKID-A3D | Reserved | Mcode | CONNUM0 | BLOCK SERIAL

도면22

	0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000	BLKID-HD0 | Reserved | Mcode | Reservrd | BLOCK SERIAL
0x0010	N1C+L | N2C+L | INFSIZE | T-PRT | T-SU | INX | XT
0x0020	NM1-S(256)
0x0120	NM2-S(512)
0x0310	

도면23

0x0320	Reserved(3) | EKi | EKB version | E(Kstm, Kcon)
0x0330	E(KEKn, Kcon) | C_MAC[n]
0x0340	Reserved(8) | INF_seq# | A | LT | FNo
0x0350	MG(D)SERIAL-nnn(Upper) | MG(D)SERIAL-nnn(Lower)
0x0360	CONNUM | YMDhms-S | YMDhms-E | XCC | CT | CC | CN

도면24

bit7 : ATRACK3 모드 0 : Dual 1 : Joint

bit6, 5, 4 : 3bit N은 모드의 값

N	모드	시간	전송 레이트	SU	바이트
7	HQ	47min	176kbps	31SU	512
6		58min	146kbps	38SU	424
5	EX	64min	132kbps	42SU	384
4	SP	81min	105kbps	53SU	304
3		90min	94kbps	59SU	272
2	LP	128min	66kbps	84SU	192
1	mono	181min	47kbps	119SU	136
0	mono	258min	33kbps	169SU	96

bit3 : Reserved

bit2 : 데이터 구분 0 : 오디오 1 : 기타

bit1 : 재생 SKIP 0 : 통상 재생 1 : SKIP

bit0 : 엠파시스 0 : OFF 1 : ON(50／15μS)

도면25

bit7 : 복사 허가 0 : 복사 금지 1 : 복사 가능

bit6 : 세대 0 : 오리지널 1 : 제1 세대 이상

HCMS bit5－4 : 고속 디지털 복사에 관한 복사 제어

00 : 복사 금지 01 : 복사 제1 세대 10 : 복사 가능

복사 제1 세대의 복사한 아이는 복사 금지로 함

bit3－2 : MagicGate 인증 레벨

00 : Level10(Non－MG) 01 : Level1

02 : Level2 11 : Reserved

Level10 이외는 디바이드, 콘바인할 수 없음

bit1, 0 : Reserved

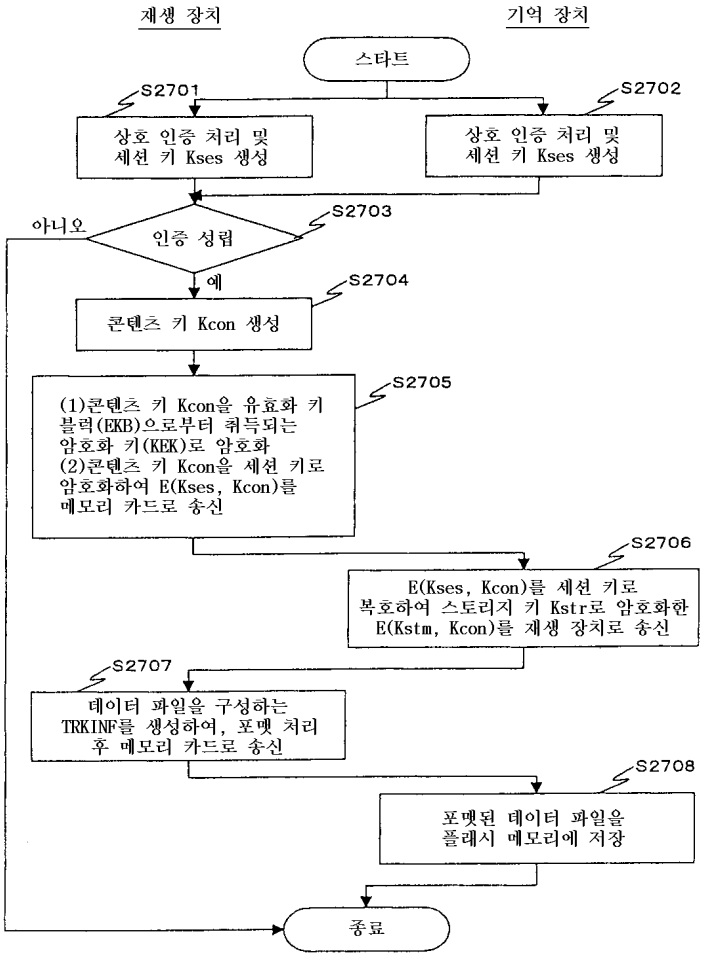
도면26

0x0370	PRTSIZE	PRTKEY		Reserved(8)
0x0380		CONNUM0	PRTSIZE(0x0388)	PRTKEY
0x0390		Reserved(8)		CONNUM0

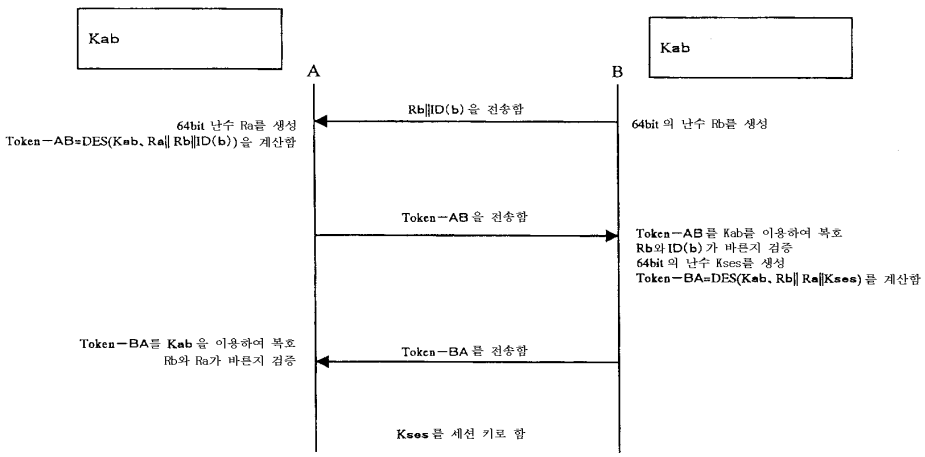
도면27

0x4000	BLKID－A3D	Reserved	Mcode	CONNUM0	BLOCK SERIAL
0x4010	BLOCKSEED			INITIALIZATION VECTOR	
0x4020	SU－000(Nbyte＝384byte)				

도면28

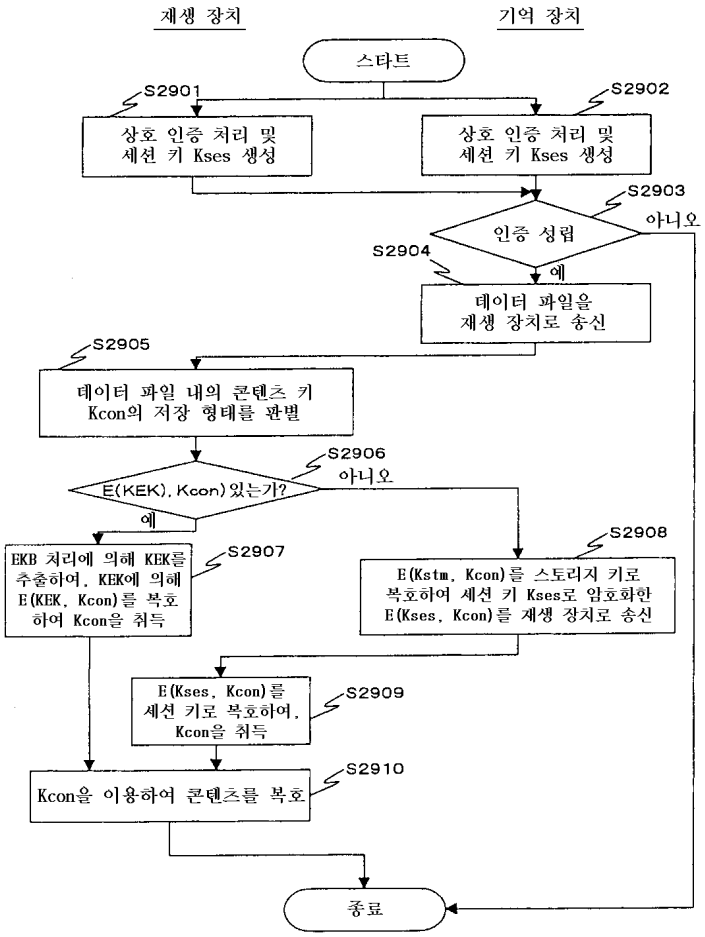


도면29

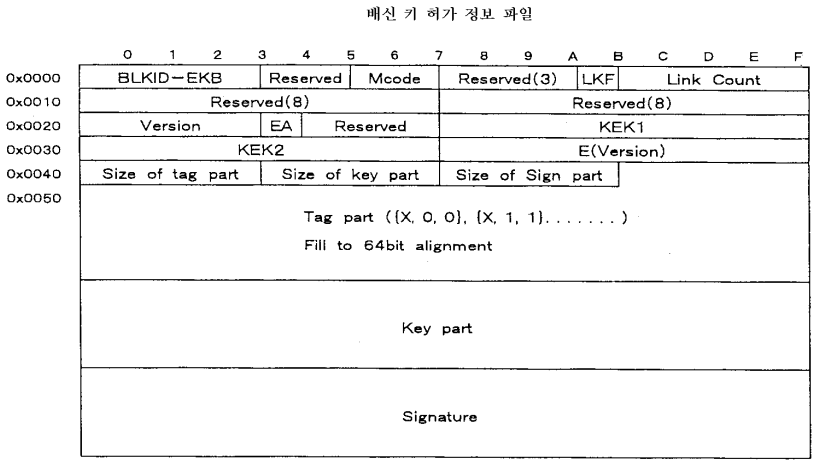


ISO/IEC 9798-2 대칭 키 암호 기술을 이용한 상호 인증 및 키 공유 방식

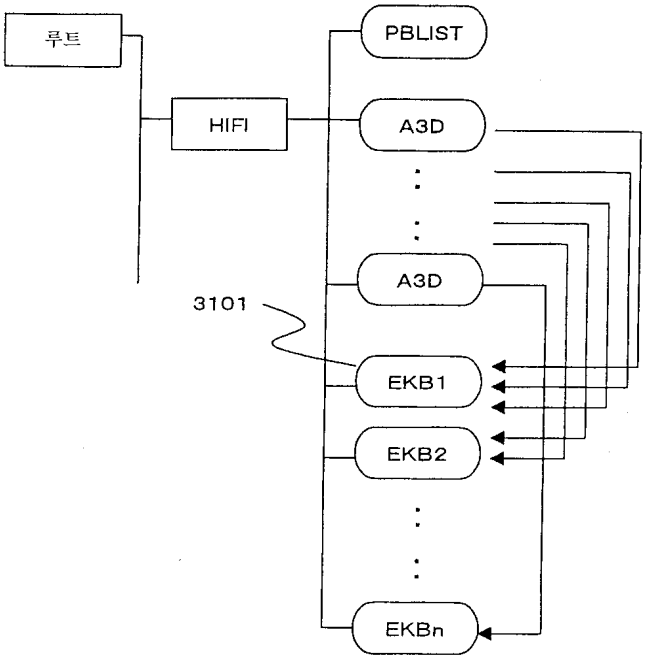
도면30


재생 장치
기억 장치
스타트
S2901
상호 인증 처리 및 세션 키 Kses 생성
S2902
상호 인증 처리 및 세션 키 Kses 생성
S2903
인증 성립
아니오
S2904
예
데이터 파일을 재생 장치로 송신
S2905
데이터 파일 내의 콘텐츠 키 Kcon의 저장 형태를 판별
S2906
E(KEK), Kcon) 있는가?
아니오
예
EKB 처리에 의해 KEK를 추출하여, KEK에 의해 E(KEK, Kcon)를 복호하여 Kcon을 취득
S2907
S2908
E(Kstm, Kcon)를 스토리지 키로 복호하여 세션 키 Kses로 암호화한 E(Kses, Kcon)를 재생 장치로 송신
E(Kses, Kcon)를 세션 키로 복호하여, Kcon을 취득
S2909
Kcon을 이용하여 콘텐츠를 복호
S2910
종료

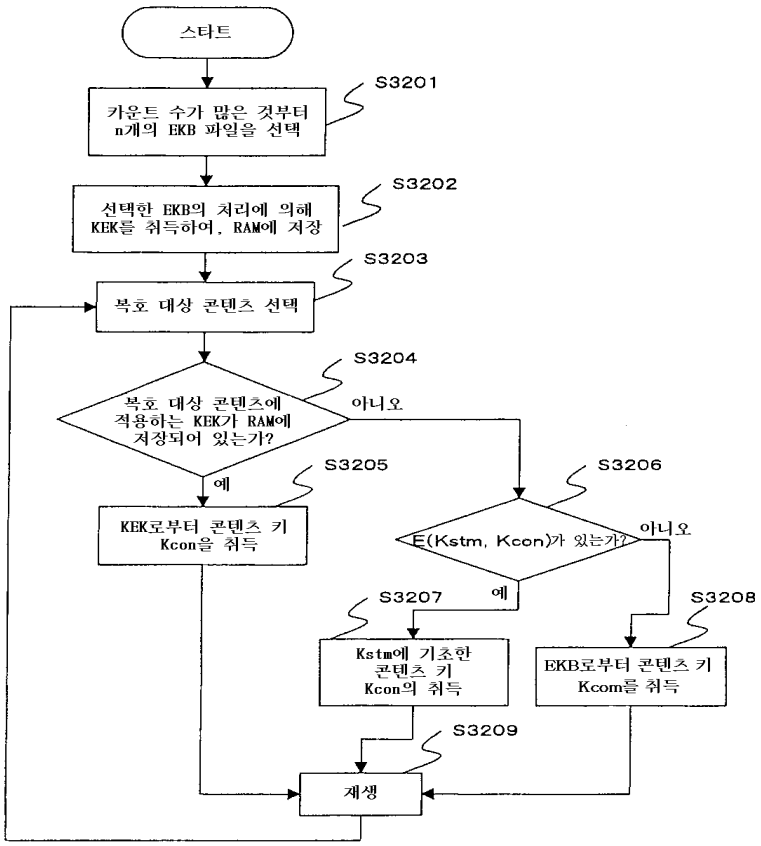
도면31


배신 키 허가 정보 파일
0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000
0x0010
0x0020
0x0030
0x0040
0x0050
BLKID-EKB
Reserved
Mcode
Reserved(3)
LKF
Link Count
Reserved(8)
Reserved(8)
Version
EA
Reserved
KEK1
KEK2
E(Version)
Size of tag part
Size of key part
Size of Sign part
Tag part ({X, 0, 0}, {X, 1, 1}.....)
Fill to 64bit alignment
Key part
Signature

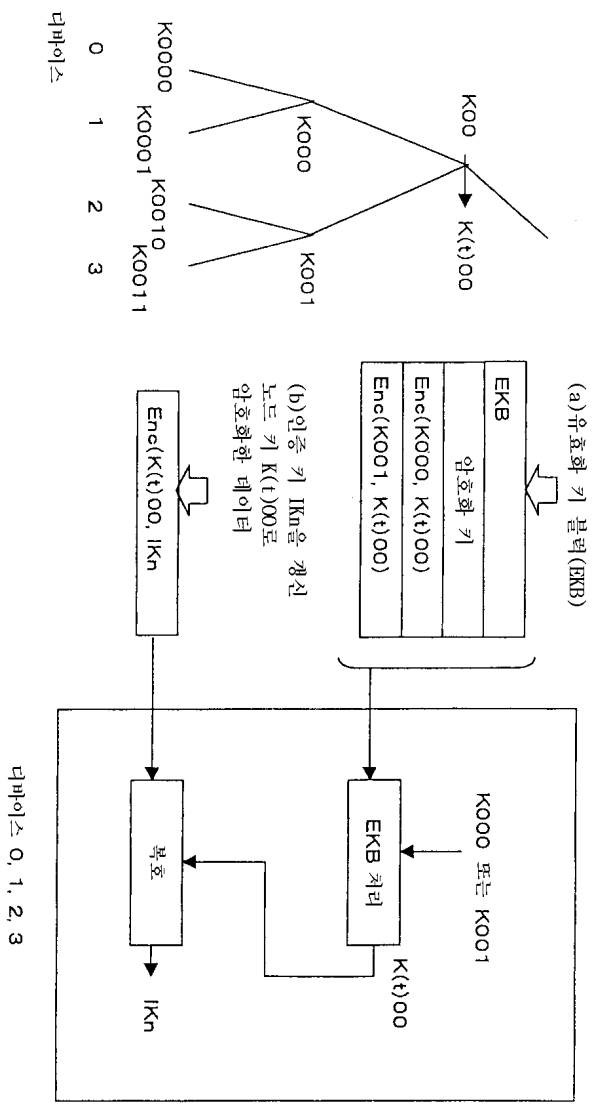
도면32


루트
PBLIST
HIFI
A3D
A3D
3101
EKB1
EKB2
EKBn

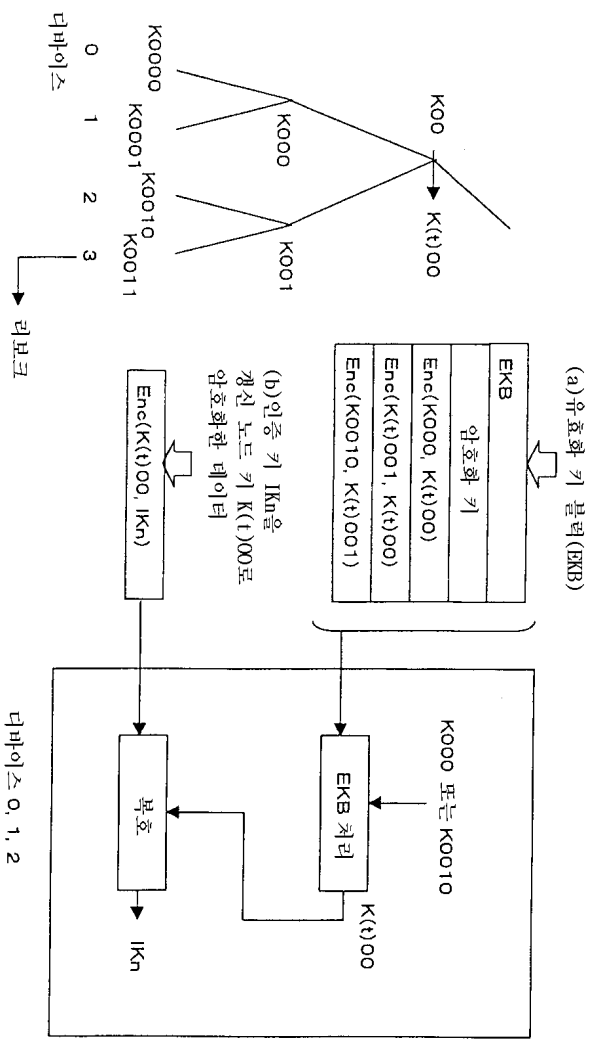
도면33


스타트
S3201
카운트 수가 많은 것부터 n개의 EKB 파일을 선택
S3202
선택한 EKB의 처리에 의해 KEK를 취득하여, RAM에 저장
S3203
복호 대상 콘텐츠 선택
S3204
복호 대상 콘텐츠에 적용하는 KEK가 RAM에 저장되어 있는가?
아니오
예
S3205
KEK로부터 콘텐츠 키 Kcon을 취득
S3206
E(Kstm, Kcon)가 있는가?
아니오
예
S3207
Kstm에 기초한 콘텐츠 키 Kcon의 취득
S3208
EKB로부터 콘텐츠 키 Kcom를 취득
S3209
재생

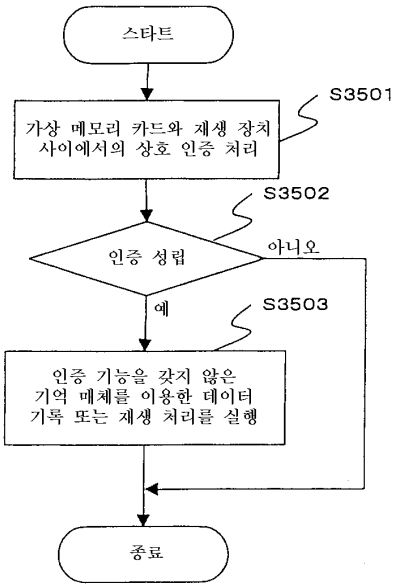
도면34


디바이스
0
1
2
3
K0000
K0001
K0010
K0011
K000
K001
K00
K(t)00
(a)유효화 키 블럭(EKB)
EKB
암호화 키
Enc(K000, K(t)00)
Enc(K001, K(t)00)
(b)인증 키 IKn을 갱신
노드 키 K(t)00로
암호화한 데이터
Enc(K(t)00, IKn
K000 또는 K001
EKB 처리
K(t)00
복호
IKn
디바이스 0, 1, 2, 3

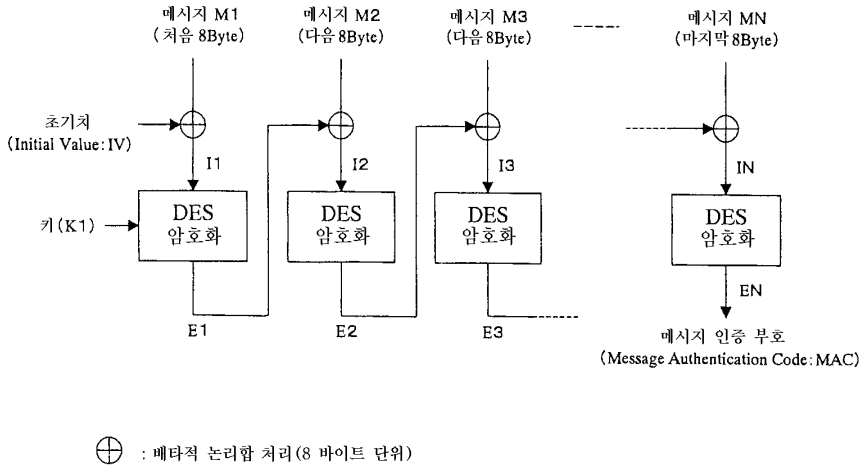
도면35


(a)유효화 키 블럭(EKB)
EKB
암호화 키
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
K000 또는 K0010
EKB 처리
K(t)00
복호
IKn
(b)인증 키 IKn을
갱신 노드 키 K(t)00로
암호화한 데이터
Enc(K(t)00, IKn)
디바이스 0, 1, 2
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
디바이스
리보크

도면36

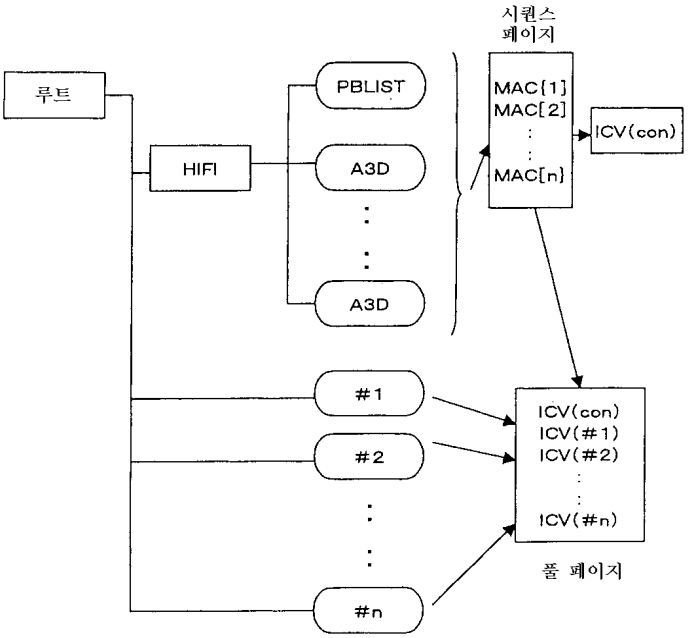


도면37

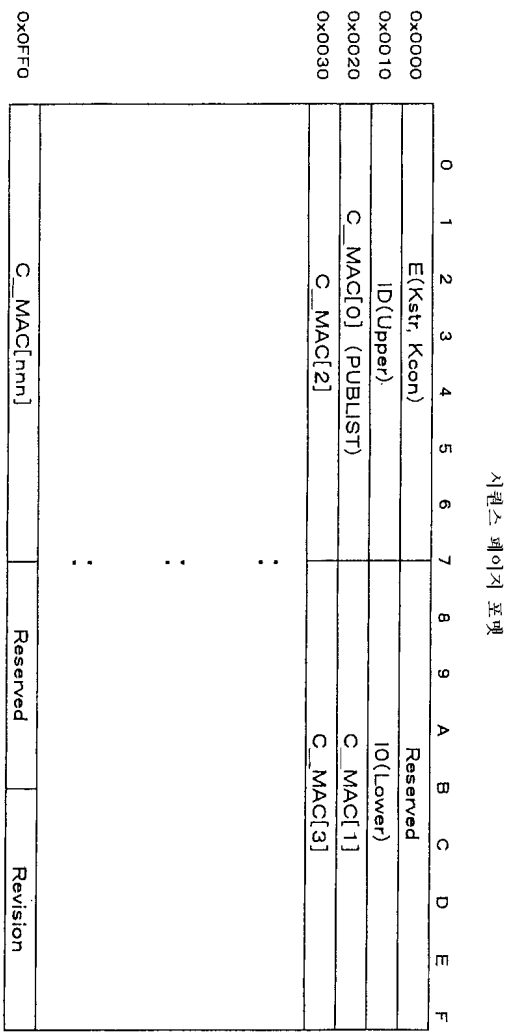


⊕ : 배타적 논리합 처리(8 바이트 단위)

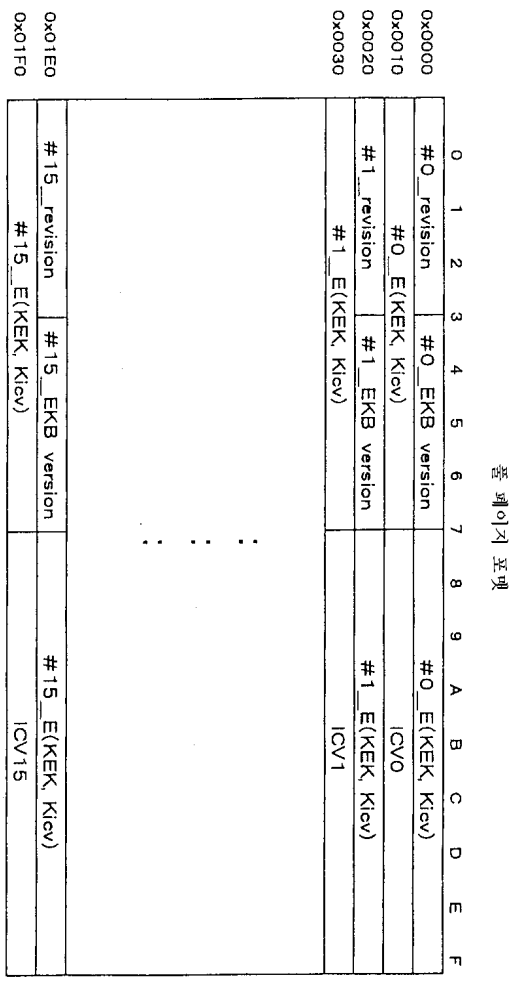
도면38


시퀀스
페이지
루트
PBLIST
HIFI
A3D
A3D
MAC[1]
MAC[2]
MAC[n]
ICV(con)
#1
#2
#n
ICV(con)
ICV(#1)
ICV(#2)
ICV(#n)
풀 페이지

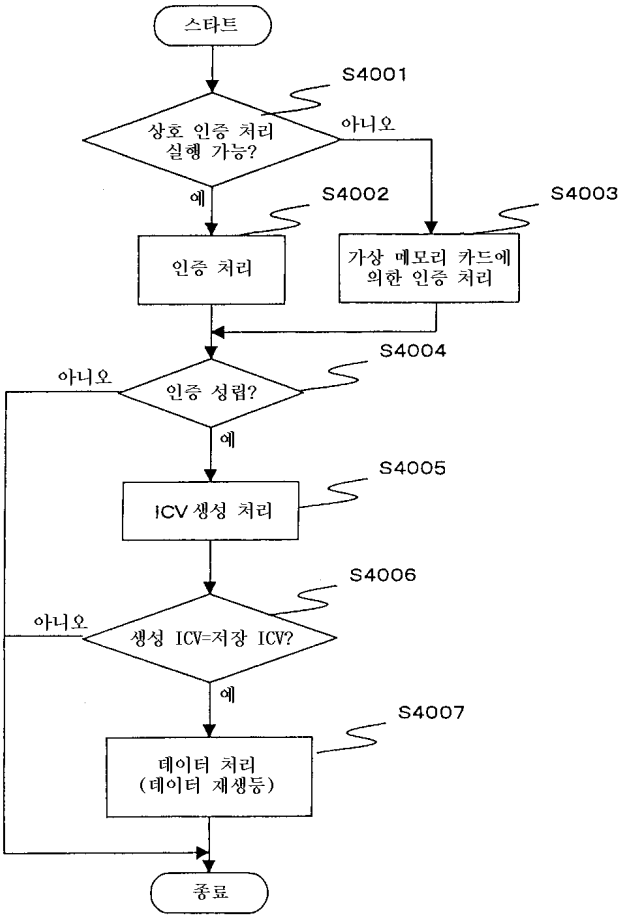
도면39


시퀀스 페이지 포맷
0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000
0x0010
0x0020
0x0030
0x0FF0
E(Kstr, Kcon)
Reserved
ID(Upper)
ID(Lower)
C_MAC[0] (PUBLIST)
C_MAC[1]
C_MAC[2]
C_MAC[3]
C_MAC[nnn]
Reserved
Revision

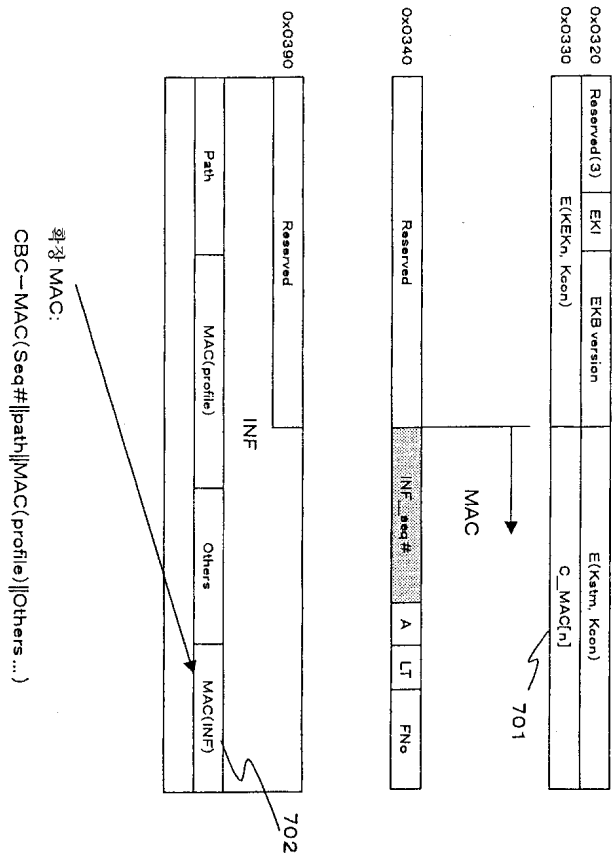
도면40


플 페이지 포맷
0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000
0x0010
0x0020
0x0030
0x01E0
0x01F0
#0_revision
#0_EKB version
#0_E(KEK, Kicv)
ICV0
#1_revision
#1_EKB version
#1_E(KEK, Kicv)
ICV1
#15_revision
#15_EKB version
#15_E(KEK, Kicv)
ICV15

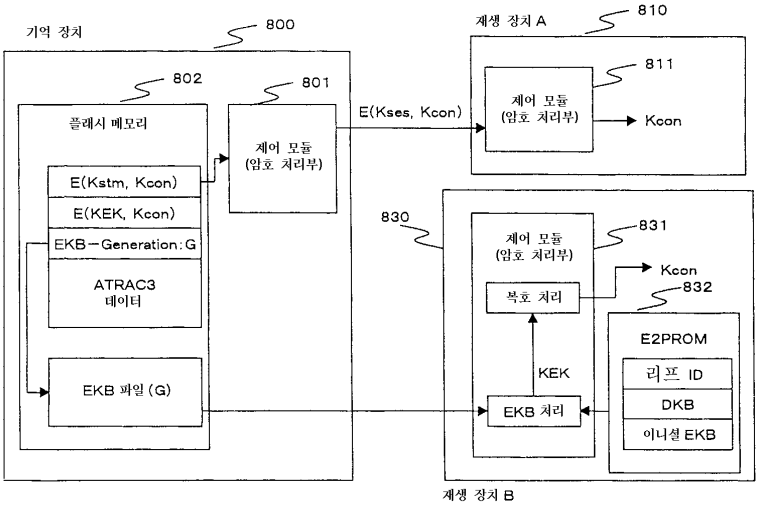
도면41


스타트
S4001
상호 인증 처리 실행 가능?
아니오
예
S4002
S4003
인증 처리
가상 메모리 카드에 의한 인증 처리
S4004
아니오
인증 성립?
예
S4005
ICV 생성 처리
S4006
아니오
생성 ICV=저장 ICV?
예
S4007
데이터 처리 (데이터 재생등)
종료

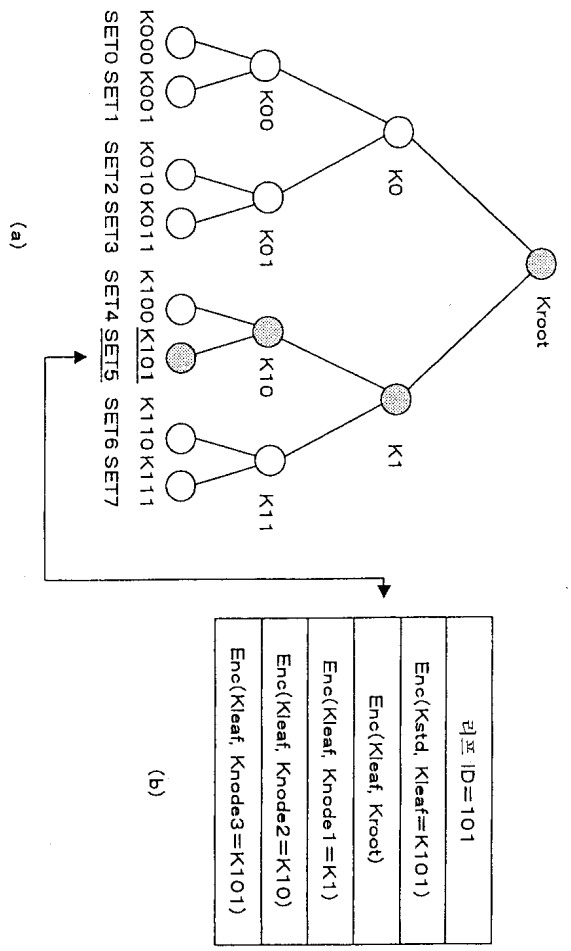
도면42


0x0390
0x0340
0x0330
0x0320
Path
MAC(profile)
Others
MAC(INF)
Reserved
INF
확장 MAC:
CBC-MAC(Seq#||path||MAC(profile)||Others...)
Reserved
INF_seq #
A
LT
FNo
MAC
Reserved(3)
EKI
EKB version
E(KEKn, Kcon)
C_MAC[n]
E(Kstm, Kcon)
701
702

도면43

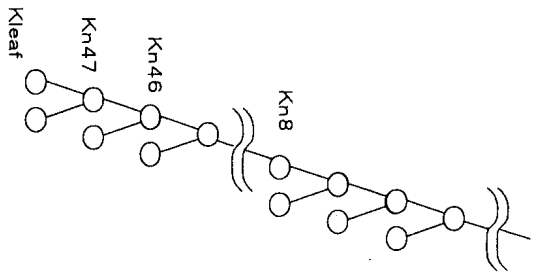

기억 장치
800
802
801
플래시 메모리
E(Kstm, Kcon)
E(KEK, Kcon)
EKB-Generation:G
ATRAC3 데이터
EKB 파일 (G)
제어 모듈 (암호 처리부)
E(Kses, Kcon)
재생 장치 A
810
811
제어 모듈 (암호 처리부)
Kcon
830
제어 모듈 (암호 처리부)
831
Kcon
복호 처리
832
KEK
EKB 처리
E2PROM
리프 ID
DKB
이니셜 EKB
재생 장치 B

도면44


Kroot
K0
K1
K00
K01
K10
K11
K000 K001 K010 K011 K100 K101 K110 K111
SET0 SET1 SET2 SET3 SET4 SET5 SET6 SET7
(a)
리프 ID=101
Enc(Kstd, Kleaf=K101)
Enc(Kleaf, Kroot)
Enc(Kleaf, Knode1=K1)
Enc(Kleaf, Knode2=K10)
Enc(Kleaf, Knode3=K101)
(b)

도면45

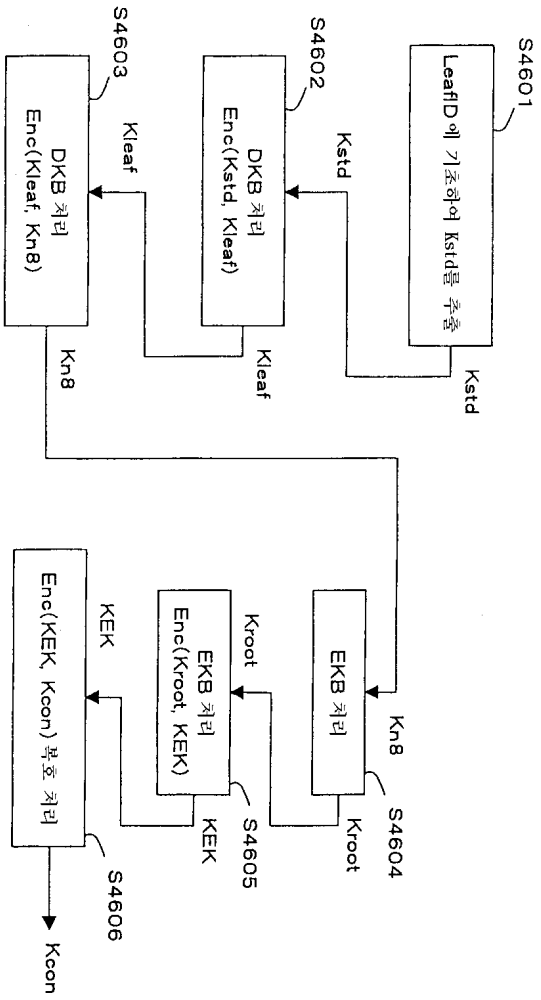
(a)



(b)

리프 ID=101
Enc(Kstd, Kleaf-1)
Enc(Kleaf, Kn47)
Enc(Kleaf, Kn46)
⋮
Enc(Kleaf, Kn8)
EKB

도면46


S4601
LeafID에 기초하여 Kstd를 추출
Kstd
S4602
DKB 처리
Enc(Kstd, Kleaf)
Kleaf
S4603
DKB 처리
Enc(Kleaf, Kn8)
Kn8
EKB 처리
Kn8
S4604
Kroot
EKB 처리
Enc(Kroot, KEK)
Kroot
S4605
KEK
Enc(KEK, Kcon) 복호 처리
KEK
S4606
Kcon