(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
H04N 7/08 (2006.01)

(45) 공고일자 2006년08월09일
(11) 등록번호 10-0610523
(24) 등록일자 2006년08월02일

(21) 출원번호 10-1999-7007180
(22) 출원일자 1999년08월09일
번역문 제출일자 1999년08월09일
(86) 국제출원번호 PCT/JP1998/003127
국제출원일자 1998년07월13일

(65) 공개번호 10-2000-0070915
(43) 공개일자 2000년11월25일
(87) 국제공개번호 WO 2000/03541
국제공개일자 2000년01월20일

(81) 지정국 국내특허 : 일본, 대한민국, 미국,

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

(73) 특허권자 소니 가부시끼 가이샤
일본국 도쿄도 시나가와쿠 키타시나가와 6쵸메 7반 35고

(72) 발명자 구보타다츠야
일본도쿄도시나가와쿠키타시나가와6쵸메7반35고소니가부시키가이샤내

와카츠키노리오
일본도쿄도시나가와쿠키타시나가와6쵸메7반35고소니가부시키가이샤내

(74) 대리인 이병호
정상구
신현문
이범래

심사관 : 최훈

(54) 프로그램 배포 시스템, 프로그램 전송 방법 및 조건부 액세스 시스템

요약

본 발명에 따른 전송 스트림 패킷 (transport stream packet)의 형태로 구성된 복수의 데이터 요소들로 구성되는 프로그램 데이터의 전송 스트림 패킷을 다중화하여 전송하는 데이터 다중화 장치, 프로그램 배포 시스템, 프로그램 전송 시스템, 유료 방송 시스템, 프로그램 전송 방법, 조건부 액세스 시스템, 및 데이터 수신 장치에서, 프로그램을 구성하는 복수의 데이터 요소들 중 하나 또는 그 이상의 데이터 요소에 대응하는 스크램블 키 (scramble key)(Ks)를 발생하고 각 데이터 요소를 스크램블링함으로서, 청취자는 각 데이터 요소에 대해 가입될 수 있다.

대표도

도 1

명세서

기술분야

본 발명은 데이터 다중화 장치, 프로그램 배포 시스템, 프로그램 전송 시스템, 유료 방송 시스템, 프로그램 전송 방법, 조건부 액세스 시스템, 및 데이터 수신 장치에 관한 것으로, 특히 예를 들면, 전송을 위해 비디오 데이터 및 오디오 데이터의 압축 부호화 및 다중화를 실행하는 디지털 방송 시스템에 대한 데이터 다중화 장치 및 데이터 수신 장치에 적용된다.

배경기술

최근에, 비디오 데이터 및 오디오 데이터를 압축하는데 MPEG2 (Moving Picture Experts Group Phase 2)를 사용하고 지상파나 위성파를 통해 부호화된 데이터 스트림 (data stream)을 방송하는 디지털 방송 시스템이 제안되어오고 있다. 이 디지털 방송 시스템은 각각이 각 프로그램의 비디오 데이터나 오디오 데이터와 같은 기본 데이터를 소정 수의 바이트로 분할하고 분할된 데이터의 각 블록 시작부에 헤더 (header)를 첨부함으로서 부호화된 비디오 스트림 및 부호화된 오디오 스트림으로 구성되는 복수의 프로그램들에 대해 복수의 전송 패킷 (transport packet)들을 발생한다. 이어서, 이들 전송 패킷은 지상파나 위성파를 통해 방송되도록 다중화된다.

수신 장치가 전송된 데이터, 즉 다중화된 전송 패킷을 수신할 때, 이는 수신된 데이터로부터 각 전송 패킷의 헤더 정보를 판독하고 헤더 정보에 기초하여 다중화되지 않은 원래 기본 데이터를 재저장함으로서 각 프로그램의 부호화된 비디오 스트림 및 부호화된 오디오 스트림을 구할 수 있다.

이러한 디지털 방송 시스템에서, 프로그램은 일반적으로 복수의 데이터 요소들 (복수의 채널들에 대한 비디오 데이터 및 오디오 데이터)로 구성된다. 그러므로, 이러한 디지털 방송 시스템에서는 청취자가 프로그램에 포함된 각 데이터 요소에 가입하는 것이 바람직하다. 예를 들어, 프로그램은 총 4가지 종류의 데이터 요소, 즉 비디오 데이터, 메인 오디오 (main audio) 데이터, 서브오디오 (subaudio) 데이터, 및 추가 데이터로 구성된다고 가정한다. 종래 디지털 방송 시스템에서는 청취자가 각 프로그램, 즉 모두 6개의 데이터 요소에 대해 가입되어야 한다. 따라서, 수신자가 비디오 데이터 및 메인 오디오 데이터에만 가입하고 싶더라도, 불필요한 요소 (서브오디오 및 추가 데이터)에 가입하여야 한다.

더욱이, 종래 스크램블 (scramble) 장치는 다중화 이전에 사용되므로, 장치는 각 프로그램에 대해 스크램블 장치를 포함하여야 하므로 불가피하게 커지게 된다.

부가하여, 종래 디지털 방송 시스템은 프로그램 가입 정보와 암호화 키를 특정한 간격으로 프로그램에 다중화하여 전송하였다. 다중화 스트림의 최고 비트 레이트가 제한되므로, 전송 레이트는 이러한 프로그램 가입 정보 및 암호화 키의 전송으로 인해 프로그램에 대해 충분히 높지 않다. 그러므로, 충분한 전송 레이트가 보장될 수 없으면, 프로그램 데이터를 버퍼링하는 전송 버퍼가 오버플로우 (overflow)되는 문제점이 있었다.

발명의 상세한 설명

본 발명은 상기를 고려해 구상되었고, 수신자가 프로그램을 구성하는 데이터 요소들 중 필요한 데이터 요소들만을 수신하도록 허용하는 즉, 즉 각 데이터 요소를 스크램블 (scramble)할 수 있는 디지털 방송 시스템을 제안한다.

본 발명은 각 프로그램에 대한 스크램블 장치의 제공을 방지함으로서 장치 구성을 최소화시킬 수 있는 디지털 방송 시스템을 제안한다.

본 발명은 어떠한 프로그램 데이터라도 버퍼링하는 전송 버퍼로부터 어떠한 오버플로우라도 방지할 수 있는 디지털 방송 시스템을 제안한다.

이러한 문제점을 해결하기 위해, 본 발명은 전송 스트림 패킷들의 형태로 구성된 복수의 데이터 요소들로 구성되는 프로그램 데이터의 전송 스트림 패킷들을 다중화하여 전송하는 데이터 다중화 장치를 제공한다. 데이터 다중화 장치는 데이터 요소에 대응하는 스크램블 키를 발생하기 위한 스크램블 키 발생 수단과, 스크램블 키 발생 수단에 의해 발생된 스크램블 키

를 사용해 대응하는 데이터 요소의 전송 스트림 패킷을 스크램블링하기 위한 스크램블 수단을 구비하므로, 데이터 다중화 장치는 프로그램을 구성하는 복수의 데이터 요소들 중에서 하나 또는 그 이상의 데이터 요소들에 대응하는 스크램블 키를 발생하고 각 데이터 요소를 스크램블링할 수 있다.

각 데이터 요소를 스크램블링함으로서, 청취자는 각 데이터 요소에 대해 가입할 수 있다.

본 발명은 복수의 데이터 요소들로 구성된 프로그램을 배포하기 위한 프로그램 배포 시스템을 제공하고, 이는 가입자 관리 시스템을 통해 각 프로그램이나 데이터 요소에 대한 가입자의 가입만을 관리하지 않고, 각 데이터 요소에 대해 스크램블 키를 발생하여 발생된 스크램블 키에 기초하여 프로그램에 포함된 데이터 요소를 디스크램블링(descrambling)하고 다중화된 스트림에 포함되는 부호화된 데이터 요소에 대해 각 데이터 요소를 선택적으로 스크램블링하므로, 청취자는 각 데이터 요소에 대해 가입할 수 있게 된다.

본 발명은 복수의 데이터 요소들로 구성된 프로그램을 전송하기 위한 프로그램 전송 시스템을 제공하고, 이는 가입자가 가입한 데이터 요소와 프로그램만을 보거나 듣기 위해, 프로그램에 포함된 복수의 데이터 요소들을 스크램블링하는데 사용되는 복수의 스크램블 키들을 발생하고, 발생된 스크램블 키에 기초하여 각 데이터 요소를 선택적으로 스크램블링하고, 또한 전송을 위해 스크램블링된 데이터 요소를 다중화하므로, 청취자는 각 데이터 요소에 대해 가입할 수 있고 그와 같이 가입된 데이터 요소만을 디스크램블링하여 보고 들을 수 있게 된다.

본 발명은 복수의 데이터 요소들로 구성된 프로그램을 방송하기 위한 유료 방송 시스템을 제공하고, 이는 가입자 관리 시스템을 통해 각 프로그램이나 데이터 요소에 대한 가입자의 가입만을 관리하지 않고, 각 데이터 요소에 대해 스크램블 키를 발생하여 발생된 스크램블 키에 기초하여 프로그램에 포함된 데이터 요소를 디스크램블링하고 다중화된 스트림에 포함되는 부호화된 데이터 요소에 대해 각 데이터 요소를 선택적으로 스크램블링하므로, 청취자는 각 데이터 요소에 대해 가입할 수 있게 된다.

본 발명은 프로그램을 구성하는 복수의 데이터 요소들이나 프로그램 배포 시스템에 의해 배포된 복수의 프로그램들 중에서 가입된 프로그램이나 데이터 요소에만 조건부 액세스를 제공하기 위한 조건부 액세스 시스템을 제공하고, 이는 복수의 암호화 스크램블 키들을 포함하는 복수의 전송 스트림 패킷들로부터, 수신자에 의해 가입된 프로그램이나 데이터 요소와 연관되는 암호화 스크램블 키를 포함하는 전송 스트림 패킷을 필터링하고, 필터링된 복수의 전송 스트림 패킷들에 포함된 복수의 암호화 스트림 키들을 해독하고, 해독된 복수의 스크램블 키들을 발생하고, 또한 복수의 데이터 요소들에 대응하는 해독된 복수의 스크램블 키들을 사용해 각 데이터 요소를 디스크램블링하므로, 청취자는 각 데이터 요소에 대해 가입할 수 있고 이와 같이 가입된 데이터 요소만을 디스크램블링하여 보고 듣을 수 있게 된다.

본 발명에 따라, 각각 대응하는 스크램블 키로 다중화된 각 전송 스트림 패킷을 스크램블링함으로서, 스크램블링을 위한 회로 구성은 다중화 이전에 각 전송 스트림 패킷을 스크램블링하는 것 보다 더 간단해질 수 있다.

본 발명은 복수의 데이터 요소들을 구성하는 복수의 데이터 패킷들을 저장하는 복수의 버퍼 메모리들, 버퍼 메모리를 스위칭하기 위한 스위치 수단을 갖고 스위치 수단으로 버퍼 메모리를 순차적으로 시분할 스위칭하여 출력을 제공하도록 복수의 패킷 데이터 스트링 (string)을 시분할 다중화하는 다중화 수단, 및 패킷 데이터 스트링에 대한 입력 레이트에 따라 스위치 수단에 의해 스위칭가능한 복수의 버퍼 메모리를 선택하는 스위치 제어 수단을 구비한다. 따라서, 입력 레이트가 기준 레이트 보다 더 높을 때, 복수의 버퍼 메모리 중에서 우선순위가 낮은 정보를 버퍼링하기 위한 버퍼 메모리를 배제하도록 스위치 수단을 스위칭가능하게 제어함으로서, 우선순위가 높은 데이터 요소를 버퍼링하기 위한 버퍼 메모리로부터의 오버플로우가 방지될 수 있다.

본 발명은 전송 스트림 패킷의 형태인 복수의 데이터 요소들로 구성된 프로그램 데이터의 전송 스트림 패킷을 다중화함으로서 얻어진 다중화된 데이터를 수신하는 데이터 수신 장치를 제공하고, 이는 각 데이터 요소에 대응하는 스크램블 키를 다중화된 데이터로부터 추출하기 위한 스크램블 키 추출 수단과, 스크램블 키 추출 수단에 의해 추출된 스크램블 키를 사용해 다중화된 데이터에 포함된 각 데이터 요소에 대한 전송 스트림 패킷을 디스크램블링하기 위한 디스크램블링 수단을 구비하므로, 각 데이터 요소는 스크램블 키를 사용해 스크램블 키에 대응하는 각 데이터 요소에 대한 전송 스트림 패킷을 디스크램블링함으로서 분리되어 디스크램블링될 수 있다.

도면의 간단한 설명

도 1은 본 발명에 따른 디지털 방송 시스템의 구조를 도시하는 블록도.

도 2는 각 프로그램에 포함된 기본 데이터와 스크램블 키 (scramble key) 사이의 대응관계를 도시하는 도면.

도 3은 전송 스트림 패킷 (transport stream packet)으로 기록된 정보와 그 PID 값 사이의 대응관계를 도시하는 도면.

도 4는 전송 패킷 종류와 그 PID 값 사이의 대응관계를 도시하는 도면.

도 5는 부호화 시스템의 구조를 도시하는 블록도.

도 6은 전송 패킷에 대한 구문 (syntax)을 도시하는 도면.

도 7은 전송 패킷에 대한 구문을 도시하는 도면.

도 8은 전송 패킷에 대한 구문을 도시하는 도면.

도 9는 전송 패킷에 대한 구문을 도시하는 도면.

도 10은 멀티플렉서 시스템의 구조를 도시하는 블록도.

도 11은 각 테이블과 그 PID 사이의 대응관계를 도시하는 도면.

도 12는 프로그램 연관 섹션에 대한 구문을 도시하는 도면.

도 13은 프로그램 연관 섹션에 대한 구문을 도시하는 도면.

도 14는 각 섹션과 그 PID 값 사이의 대응관계를 도시하는 도면.

도 15는 프로그램 맵 (map) 섹션에 대한 구문을 도시하는 도면.

도 16은 프로그램 맵 섹션에 대한 구문을 도시하는 도면.

도 17은 조건부 액세스 섹션에 대한 구문을 도시하는 도면.

도 18은 조건부 액세스 섹션에 대한 구문을 도시하는 도면.

도 19는 조건부 액세스 섹션에 대한 구문을 도시하는 도면.

도 20은 디스크립터의 내용과 각 테이블에 의해 표시되는 기본 데이터의 PID를 도시하는 도면.

도 21은 EMM 패킷에 대한 데이터 구조를 도시하는 도면.

도 22는 ECM 패킷에 대한 데이터 구조를 도시하는 도면.

도 23은 수신 장치의 구조를 도시하는 블록도.

실시예

본 발명을 수행하기 위한 최상의 모드
이하, 도면을 참조하여 본 발명의 실시예를 설명한다.

도 1을 참조하여, 본 발명에 따른 데이터 다중화 장치가 적용된 데이터 방송 시스템이 설명된다.

이 데이터 방송 시스템은 위성 디지털 방송 시스템이나 지상파 디지털 방송 시스템과 같은 유료 방송 시스템에 사용되는 시스템으로서, 도 1에 도시된 바와 같이, 방송 데이터 처리 시스템 (BDPS) (Broadcast_Data_Processing_System)(1),

가입자 관리 시스템 (SMS) (Subscriber_Management_System)(2), 가입자 시청허가 시스템 (SAS) (Subscriber_Authorization_System)(3), EPG (Electronic_Program_Guide) 시스템(4), 서버 시스템(5), 루트 시스템 (routing system)(6), 인코더 시스템(7), 멀티플렉서 시스템(8), 인코더/멀티플렉서 제어 유닛(9), 및 변조 회로(10)를 구비한다.

방송 데이터 처리 시스템(1)은 가입자 관리 시스템(2), 가입자 시청허가 시스템(3), EPG 시스템(4), 서버 시스템(5), 루트 시스템(6), 인코더 시스템(7), 멀티플렉서 시스템(8), 인코더/멀티플렉서 제어 유닛(9), 및 변조 회로(10)와 같이 방송국내에 제공되는 모든 시스템 및 장치를 제어하기 위한 시스템이다. 이 방송 데이터 처리 시스템(1)은 그에 등록된 프로그램 계획표를 갖고, 이는 프로그램 공급자에 의해 공급된 프로그램 및 판촉물과 방송국 자체에서 제작되는 프로그램 및 CM 내용물을 포함하는 모든 내용물에 대한 방송 시간을 관리하는데 사용된다. 방송 데이터 처리 시스템(1)은 프로그램 계획표에 따라 각 장치와 시스템을 제어한다. 이러한 프로그램 계획표는 프로그램 공급자에 대한 정보를 포함하는 서비스 정보 파일, 매시간이나 매일에 기초하여 기록된 프로그램 스케쥴을 포함하는 이벤트 정보 파일, 및 15초에 기초하여 기록된 프로그램 시간 스케쥴을 포함하는 동작 정보 파일로 구성된다.

가입자 관리 시스템(2)은 가입자 등록 정보 및 유인 정보와 같은 가입자 관리 정보를 관리하는데 사용된다. 특별히, 가입자 관리 시스템(2)은 가입, 거래, 및 요금청구와 같은 주요 기능을 갖는 유료 방송 시스템내의 핵심 시스템이다. 부가하여, 가입자 관리 시스템(2)은 데이터 요소를 스크램블링하기 위한 스크램블 키(Ks) 및 이러한 스크램블 키(Ks)를 암호화하기 위한 작업키 (work key)(Kw)와 같은 키에 대한 정보 뿐만 아니라 방송 계약자, 프로그램 공급자, 및 프로그래밍 중개자에 대한 정보를 관리하기 위한 또 다른 기능을 갖는다. 가입자 관리 시스템(2)은 또한 실시간으로 전화선을 통해 수신단에 제공된 IRD에 의해 공급되는 청취자 정보를 처리하기 위한 또 다른 기능을 갖는다. 가입자 관리 시스템(2)은 결과적으로 추후 설명될 가입자 시청허가 시스템(3)에게 가입자에 대한 정보 뿐만 아니라 작업키(Kw)를 포함하는 EMM (Entitlement_Management_Message) 데이터를 제공한다. 이러한 EMM 데이터는 추후 상세히 설명된다.

가입자 시청허가 시스템(3)은 마스터 키 (master key)(Km)를 사용한 소정의 암호화 알고리즘을 통해 가입자 관리 시스템(2)으로부터 수신된 EMM 데이터에 포함되는 작업키 (Kw)를 암호화함으로서 암호화된 작업키 (Kw')를 발생한다. 본 발명에 따른 디지털 방송 시스템에 사용되는 암호화 알고리즘은 본 출원인에 의해 개발된 CRYPS (SONY 상표)라는 암호화 알고리즘으로, 상공부에 의해 설정된 DES 시스템에 유사한 블록 암호화 기술을 적용함을 이해하여야 한다. 가입자 시청허가 시스템(3)은 가입자 관리 시스템(2)에 의해 공급되는 EMM 데이터에 포함된 작업키(Kw)를 암호화 알고리즘 CRYPS를 통해 암호화된 작업키(Kw')로 대치하고, 이어서 이와 같이 암호화된 EMM 데이터를 발생한다. 본 출원인의 고유의 암호화 알고리즘을 통해 암호화된 EMM 데이터를 위성을 통해 전송하기 위해, 가입자 시청허가 시스템(3)은 전송 스트림 패킷의 페이로드 섹션 (payload section)에 암호화된 EMM 데이터를 삽입함으로서 암호화된 EMM 데이터를 전송 스트림 패킷으로 번역한다. 다음 설명에서는 이러한 EMM 데이터를 포함하는 전송 스트림 패킷이 EMM 패킷이라 칭하여진다. 가입자 시청허가 시스템(3)은 네트워크를 통해 추후 설명될 인코더/멀티플렉서 제어 유닛(9)으로부터 EMM 패킷에 주어진 PID (packet identifier)를 수신함을 또한 이해하여야 한다.

가입자 시청허가 시스템(3)은 256개 작업키(Kw)가 작업키를 식별하기 위한 작업키 식별 번호 (Kw_No)와 연관된 작업키 테이블을 갖고, 이 작업키 테이블은 네트워크를 통해 멀티플렉서 시스템(8)의 메모리로 다운로드 (download)될 수 있다. 이러한 작업키 테이블이 멀티플렉서 시스템(8)으로 다운로드되는 이유는 암호화된 작업키(Kw')만이 가입자 시청허가 시스템(3)에 의해 멀티플렉서 시스템(8)에 전송되기 때문에 멀티플렉서 시스템(8)이 작업키 식별 번호 (Kw_No)로부터 암호화되지 않은 작업키(Kw)를 얻기 때문이다.

더욱이, 가입자 시청허가 시스템(3)은 전송된 프로그램에 포함되는 각 데이터 요소를 스크램블링하도록 스크램블 키(Ks)를 발생한다. 이에 대해, 가입자 시청허가 시스템(3)은 각 프로그램이나 그에 포함된 각 데이터 요소에 대해 다른 스크램블 키(Ks)를 발생한다. 예를 들어 도 2에 도시된 바와 같이, 스크램블 키(Ks1)는 각각 제 1 프로그램을 구성하는 비디오 데이터 및 메인 오디오 데이터를 스크램블링하도록 발생되고, 스크램블 키(Ks2)는 제 2 프로그램을 구성하는 비디오 데이터 및 메인 오디오 데이터를 스크램블링하도록 발생되고, 스크램블 키(Ks3)는 제 2 프로그램의 서브오디오 데이터를 스크램블링하도록 발생되고, 또한 스크램블 키(Ks4)는 제 2 프로그램의 전용 데이터를 스크램블링하도록 발생된다. 어느 스크램블 키가 어느 프로그램의 어느 요소에 지정되는가는 프로그램의 내용에 의존하여 임의적으로 가입자 시청허가 시스템(3)에 의해 결정될 수 있다. 예를 들어 제 4 프로그램에 대해, 다른 스크램블 키 (Ks7) 내지 (Ks10)는 각각 비디오 데이터, 메인 오디오 데이터, 서브오디오 데이터, 및 전용 데이터에 지정될 수 있고, 제 5 프로그램에 대해서는 같은 스크램블 키 (Ks11)가 비디오 데이터, 메인 오디오 데이터, 서브오디오 데이터, 및 전용 데이터에 모두 지정될 수 있다.

더욱이, 디지털 방송 시스템에 사용되는 암호화/해독화 시스템의 보안 레벨을 증진시키기 위해, 가입자 시청허가 시스템(3)은 가입자 시청허가 시스템(3)내의 난수 발생기에서 4초의 간격으로 스크램블 키 (Ks1) 내지 (Ks19)를 업데이트한다.

부가하여, 가입자 시청허가 시스템(3)은 디스크램블링에 사용되는 복수의 ECM (Entitlement_Control_Message) 데이터 항목을 발생한다. 이 ECM 데이터는 적어도 작업키 테이블에 목록화되는 작업키(Kw)를 지정하기 위한 작업키 번호, 데이터 스트림을 스크램블링하기 위한 스크램블 키(Ks), 및 가입자 시청허가 시스템(3)을 식별하기 위한 CA_system_ID로 구성된다. 그러므로, 도 2에 도시된 바와 같이 19개 스크램블 키 (Ks1) 내지 (Ks19)가 사용되는 예에서, 가입자 시청허가 시스템(3)은 스크램블 키만이 ECM 데이터 항목으로 등록되기 때문에 19개 ECM 데이터 항목을 발생한다.

발생된 복수의 ECM 데이터 항목을 전송하기 위해, 가입자 시청허가 시스템(3)은 또한 전송 스트림 패킷의 페이로드 섹션에 복수의 ECM 데이터 항목을 삽입함으로서 복수의 ECM 데이터 항목을 전송 스트림 패킷으로 번역한다. 다음의 설명에서, 이러한 ECM 데이터를 포함하는 전송 스트림 패킷은 ECM 패킷이라 칭하여진다. 도 1에서, ECM1은 스크램블 키(Ks1)를 포함하는 ECM 패킷을 나타내고, ECM2는 스크램블 키(Ks2)를 포함하는 ECM 패킷을 나타내고, 유사하게 ECM3 내지 ECM19는 각각 스크램블 키 (Ks3) 내지 (Ks19)를 포함하는 ECM 패킷을 나타낸다. 또한, 가입자 시청허가 시스템(3)은 네트워크를 통해 추후 설명될 인코더/멀티플렉서 제어 유닛(9)으로부터 복수의 ECM 패킷 각각에 주어진 PID (packet identifier)를 수신함을 이해하여야 한다. 또한, 도 1에서는 ECM 패킷이 간략하게 멀티플렉서 시스템에 바로 공급되는 것으로 도시되지만, 실제로는 ECM 패킷이 네트워크 및 인코더/멀티플렉서 제어 유닛(9)을 통해 멀티플렉서 시스템(8)에 공급됨을 이해하여야 한다.

이제는 상술된 작업키와 스크램블 키가 디지털 방송 시스템에 사용되어야 하는 이유가 이후 설명된다.

전형적인 디지털 방송 시스템은 일반적으로 가입자만이 프로그램을 보거나 듣도록 허가하고 그들의 가입 조건에 기초하여 이들 가입자에게 거래를 전달하는 유료 방송 시스템을 사용한다. 유료 방송 시스템을 이루기 위해, 이러한 디지털 방송 시스템은 전송 이전에 방송국에 의해 발생된 스크램블 키로 프로그램을 스크램블링하고 가입자만이 그 프로그램을 디스크램블링하여 보거나 듣도록 허용하여야 한다. 특별히, 가입자만이 수신단에서 이러한 스크램블링된 데이터 요소를 디스크램블링하도록 허용하기 위해서는 스크램블링에서 사용된 스크램블 키(Ks)가 디스크램블링에서도 또한 사용되어야 한다. 위성을 통해 전송된 프로그램을 자동적으로 디스크램블링하기 위해서는 이들 스크램블 키(Ks)가 수신단으로 전송되어야 한다.

그러나, 이들 스크램블 키(Ks)가 수신단으로 전송되면, 시청허가되지 않은 수신자도 이들 스크램블 키(Ks)를 구할 수 있고, 결과적으로 전송된 모든 프로그램을 무료로 보거나 들을 수 있게 된다. 그러므로, 제 1 보안 레벨로서, 본 발명에 따른 유료 방송 시스템은 난수 발생기에서 수 초의 간격으로 이들 스크램블 키(Ks)를 변화시키고 작업키(Kw)로 스크램블 키(Ks)를 암호화한다. 난수 발생기로 스크램블 키(Ks)를 암호화하는 것은 소프트웨어로 이루어질 수 있으므로, 암호화 기술에 대해 훨씬 더 높은 보안 레벨을 제공할 수 있다.

조건부 액세스 시스템의 보안 레벨을 더 증진시키기 위해, 본 발명에 따른 디지털 방송 시스템은 추후 설명될 멀티플렉서 시스템(8)의 암호화 회로(822)에서 작업키(Kw)를 사용해 이들 스크램블 키(Ks)를 암호화하고, 이어서 암호화된 스크램블 키(Ks')를 수신단으로 전송한다. 즉, 스크램블링 프로그램에서 사용된 스크램블 키(Ks) 그 자체를 전송하는 것 보다 암호화된 스크램블 키(Ks)를 전송함으로서, 보안 레벨을 최대로 증진시킬 수 있다. 수신단에서, 암호화된 스크램블 키(Ks')를 해독하기 위한 해독기는 수신기내의 보안 모듈 (IC 카드)에 제공되고, 모든 해독 동작은 보안 모듈로 실행된다. IC 카드로 구성된 보안 모듈은 그에 포함된 프로그램 코드가 사실상 복호화될 수 없도록 하는 구조를 가지므로, 외부사람들이 스크램블 키를 해독하기가 매우 어렵다.

EPG 시스템(4)은 전송된 프로그램과 연관된 프로그램 안내 데이터를 생성하는 시스템이다. 프로그램 안내 데이터 (EPG 데이터)는 예를 들면, 미래에 방송될 프로그램의 방송 시간을 알리는 데이터, 선택된 프로그램의 내용을 설명하는 문자 데이터, 및 프로그램에 대한 제목 데이터를 칭한다. EPG 데이터는 주로 방송국에 의해 생성되거나 프로그램 공급자가 자체 프로그램에 대해 특별한 EPG 데이터를 생성할 수 있다.

서버 시스템(5)은 다양한 내용물을 포함하는 디스크 드라이버를 어레이로 연결시킴으로서 구성된 서버와, 서버의 기록/재생 동작을 제어하기 위한 서버 제어 컴퓨터로 구성된다. 서버 시스템(5)은 이터넷 (Ethernet)과 같은 방송국내의 LAN을 통해 방송 데이터 처리 시스템(1)에 연결되므로, 프로그램 데이터 처리 시스템의 프로그램 계획 리스트에 따라 프로그램을 전송하도록 제어될 수 있다. 서버 시스템(5)은 CM 내용물을 공급하기 위한 CM 서버 기능, 텔레비젼 프로그램물과 뉴스 프로그램물을 공급하기 위한 매일 서버 기능, 및 텔레비젼 프로그램물과 영화물을 공급하기 위한 VOD (video-on-demand) 서버 기능과 같이, 기록된 내용물의 종류에 대응하는 수개의 서버 기능을 갖고, 방송 데이터 처리 시스템(1)의 제어하에서 특정한 채널을 통해 복수의 원하는 프로그램을 제공할 수 있다.

루트 시스템(6)은 서버 시스템(5)에 의해 공급된 다중채널 데이터를 적절한 채널로 전하기 위한 시스템이다. 루트 시스템(6)은 이터넷을 통해 방송 데이터 처리 시스템(1)에 연결되므로, 프로그램 데이터 처리 시스템의 프로그램 계획 리스트에 따라 적절한 채널을 통해 프로그램을 전송하도록 제어될 수 있다.

인코더/멀티플렉서 제어 유닛(9)은 인코더 시스템(7) 및 멀티플렉서 시스템(8)을 제어하기 위한 제어 명령을 네트워크를 통해 인코더 시스템(7) 및 멀티플렉서 시스템(8)에 공급한다. 인코더/멀티플렉서 제어 유닛(9)은 네트워크를 통해 EPG 시스템(4)으로부터 EPG 패킷을 수신할 뿐만 아니라 네트워크를 통해 가입자 시청허가 시스템(3)으로부터 ECM 패킷(ECM1 내지 ECM19) 및 EMM' 패킷을 수신하고, 수신된 EPG, EMM', 및 ECM 패킷을 멀티플렉서 시스템(8)에 공급한다. 부가하여, 인코더/멀티플렉서 제어 유닛(9)은 전용 데이터를 인코더 시스템(7)에 공급할 뿐만 아니라 프로그램 지정 정보(PSI: Program_Specific_Information)를 멀티플렉서(8)에 공급한다.

프로그램 지정 정보(PSI)는 프로그램 맵 (map) 테이블 (PMT)에 지정된 프로그램 번호와 그의 대응하는 전송 스트림 패킷을 도시하기 위한 프로그램 연관 테이블 (PAT: Program_Association_Table), 지정된 프로그램에 대한 데이터 요소가 설명되는 전송 스트림 패킷을 도시하기 위한 프로그램 맵 테이블 (PMT: Program_Map_Table), 및 EMM 데이터를 포함하는 전송 스트림 패킷을 지정하기 위한 조건부 액세스 테이블 (CAT: Conditional_Access_Table)로 구성된다. 인코더/멀티플렉서 제어 유닛(9)은 프로그램 연관 테이블(PAT), 프로그램 맵 테이블(PMT), 및 조건부 액세스 테이블(CAT)를 발생하고, 발생된 이들 테이블을 전송 스트림의 형태로 제공한다. 다음 설명에서, 이러한 프로그램 연관 테이블(PAT)을 포함하는 전송 스트림 패킷은 PAT 패킷이라 칭하여지고, 프로그램 맵 테이블(PMT)을 포함하는 전송 스트림 패킷은 PMT 패킷이라 칭하여지고, 또한 조건부 액세스 테이블(CAT)을 포함하는 전송 스트림 패킷은 CAT 패킷이라 칭하여진다.

EPG 시스템(4), 가입자 시청허가 시스템(3), 및 인코더 시스템(7)에 의해 생성된 전송 스트림 패킷에 적절한 패킷 식별자(PID)를 지정하기 위해, 인코더/멀티플렉서 제어 유닛(9)은 적절한 패킷 식별자(PID)를 발생하고, 이를 가입자 시청허가 시스템(3), EPG 시스템(4), 및 인코더 시스템(7)에 공급한다.

패킷 식별자 (PID: Packet Identification)는 전송 스트림 패킷을 식별하기 위한 식별자이다. 이러한 패킷 식별자(PID)는 전송 스트림 패킷의 페이로드에 저장된 데이터 종류에 따라 결정되는 13 비트의 고유의 데이터이다. 예를 들어, 도 3에 도시된 바와 같이, "0x0000"의 PID 값은 페이로드에 프로그램 연관 테이블(PAT)에 관한 정보를 저장하는 전송 스트림 패킷에 지정되고, "0x0001"의 PID 값은 페이로드에 조건부 액세스 테이블(CAT)에 대한 정보를 저장하는 전송 스트림 패킷에 지정된다. PID 값 "0x0020" 내지 "0x1FFE" 중 하나는 페이로드에 비디오 데이터 및 오디오 데이터와 같은 기본 데이터나 프로그램 맵 테이블(PMT)에 대한 정보를 저장하는 전송 스트림 패킷에 대한 PID 값으로 선택된다.

각 전송 스트림 패킷에 지정된 PID를 발생할 때, 인코더/멀티플렉서 제어 유닛(9)은 스크램블링에 사용되는 스크램블 키(Ks)와 전송 스트림 패킷에 지정된 PID 사이의 대응관계를 나타내는 PID 테이블을 발생한다. PID 테이블은 이전에 사용된 PID 값들을 저장하는 테이블이고, PID 테이블에 저장된 PID 값들을 참조하여, 인코더/멀티플렉서 제어 유닛(9)은 PID 값이 이전에 사용되었는가의 여부를 결정하고 이전 PID 값의 복제가 아닌 새로운 PID 값을 발생할 수 있다. PID 테이블은 또한 다중화된 전송 스트림 패킷이 스크램블링되어야 하는가의 여부를 결정하는데 사용된다.

예를 들어 도 4에 도시된 바와 같이, PID 테이블은 발생된 전송 스트림 패킷에 지정된 PID와 대응하는 전송 스트림 패킷에 저장된 데이터에 사용되는 스크램블 키(Ks) 사이의 대응관계를 나타낸다. 도 4에서, PAT 패킷에 대한 PID 값은 "0x0000"으로 고정되고, 제 1 프로그램에 대응하는 PMT 패킷에 지정된 PID 값은 "0x0100"이고, 제 2 프로그램에 대응하는 PMT 패킷에 지정된 PID 값은 "0x0101"이고, 제 1 프로그램의 비디오 데이터로 사용되는 Video[1] 패킷과 메인 오디오 데이터로 사용되는 Main_Audio[1] 패킷을 스크램블링하는 스크램블 키(Ks1)와 함께 ECM 패킷에 지정된 PID 값은 "0x0300"이고, 제 2 프로그램의 비디오 데이터로 사용되는 Video[2] 패킷을 스크램블링하는 스크램블 키(Ks2)와 함께 ECM 패킷에 지정된 PID 값은 "0x0301"이고, 제 2 프로그램의 메인 오디오 데이터로 사용되는 Main_Audio[2] 패킷을 스크램블링하는 스크램블 키(Ks2)와 함께 ECM 패킷에 지정된 PID 값은 "0x0302"이고, 제 2 프로그램의 서브오디오 데이터로 사용되는 Sub_Audio[2] 패킷을 스크램블링하는 스크램블 키(Ks3)와 함께 ECM 패킷에 지정된 PID 값은 "0x0303"이고, 제 2 프로그램의 전용 데이터로 사용되는 Private[2] 패킷을 스크램블링하는 스크램블 키(Ks4)와 함께 ECM 패킷에 지정된 PID 값은 "0x0304"이다. 제 1 프로그램의 비디오 데이터로 사용되는 Video[1] 패킷에 지정된 PID 값은 "0x0500"이고, 이 종류의 패킷은 스크램블 키(Ks1)로 스크램블링된다. 제 1 프로그램의 메인 오디오 데이터로 사용되는 Main_Audio[1] 패킷에 지정된 PID 값은 "0x0501"이고, 이 종류의 패킷은 스크램블 키(Ks1)로 스크램블링된다. 제 2 프로그램의 비디오 데이터로 사용되는 Video[2] 패킷에 지정된 PID 값은 "0x0502"이고, 이 종류의 패킷은 스크램블 키(Ks2)로 스크램블링된다. CAT 패킷에 지정된 PID 값은 "0x0001"로 고정되고, EMM 패킷에 지정된 PID 값은 "0x700"이다.

도 5에 도시된 바와 같이, 인코더 시스템(7)은 MPEG2 표준에 따라 복수의 채널을 통해 공급된 비디오 데이터를 부호화하기 위한 복수의 MPEG 비디오 인코더 (711V) 내지 (719V), MPEG2 표준에 따라 비디오 데이터 스트림에 대응하는 복수의 오디오 데이터 스트림을 부호화하기 위한 MPEG 오디오 인코더 (711A) 내지 (719A), 각 비디오 인코더 및 각 오디오 인코더에 의해 공급된 스트림과 인코더/멀티플렉서 제어 유닛(9)에 의해 공급된 전용 데이터 스트림을 다중화하기 위한 다중화 회로 (721) 내지 (729), 및 비디오/오디오 인코더 (711) 내지 (719)와 다중화 회로 (721) 내지 (729)를 제어하기 위한 부호화 제어기(70)를 구비한다. 도 5에 도시된 바와 같은 인코더 시스템은 9개 채널에 대한 프로그램을 부호화하도록 구성되지만, 물론 단지 9개 채널이 아니라 임의의 채널을 처리할 수 있음을 이해하여야 한다.

각 비디오 인코더 (711V) 내지 (719V)는 MPEG2 표준에 따라 비디오 데이터를 부호화함으로서 부호화된 비트 스트림을 발생한다. 이때, 비디오 인코더는 화상 유닛으로 부호화된 비트 스트림을 나누고 나뉜 비트 스트림에 헤더 (header)를 첨부함으로서 PES (Packetized Elementary Stream) 패킷을 발생한다. 다음에, 비디오 인코더는 PES 패킷을 184 바이트로 나누고 나뉜 184 바이트의 비트 스트림에 4 바이트의 헤더를 첨부함으로서 전송 스트림 패킷을 발생한다. 스트림으로 배열된 전송 스트림 패킷으로 구성되는 데이터는 전송 스트림이라 칭하여진다. 전송 스트림 패킷을 발생할 때, 인코더/멀티플렉서 제어 유닛(9)은 부호화된 비디오 데이터를 포함하는 전송 스트림 패킷을 식별하기 위한 패킷 식별자(PID)를 각 비디오 인코더 (711V) 내지 (719V)에 공급한다.

각 오디오 인코더 (711A) 내지 (719A)는 MPEG2 표준에 따라 메인 오디오 데이터와 서브오디오 데이터를 부호화함으로서 부호화된 비트 스트림을 발생한다. 비디오 인코더와 같이, 각 오디오 인코더 (711A) 내지 (719A)는 비트 스트림으로부터 PES 패킷을 발생하고, 이어서 PES 패킷을 184 바이트로 나누고 4 바이트의 헤더를 첨부함으로서 오디오 데이터를 포함한 전송 스트림 패킷을 발생한다. 전송 스트림 패킷을 발생할 때, 인코더/멀티플렉서 제어 유닛(9)은 부호화된 오디오 데이터를 포함하는 전송 스트림 패킷을 식별하기 위한 패킷 식별자(PID)를 각 오디오 인코더 (711A) 내지 (719A)에 공급한다.

각 다중화 회로 (721) 내지 (729)는 부호화된 비디오 데이터를 포함하는 전송 스트림, 부호화된 오디오 데이터를 포함하는 전송 스트림, 및 전용 데이터를 포함하는 전송 스트림을 다중화함으로서 전송 스트림을 발생한다. 특별히, 다중화 회로는 부호화된 비디오 데이터를 포함하는 전송 스트림, 부호화된 오디오 데이터를 포함하는 전송 스트림, 및 전용 데이터를 포함하는 전송 스트림 사이를 전송 스트림 패킷의 단위로 스위칭함으로서 이들 스트림을 다중화한다. 그러므로, 출력 전송 스트림에서는 부호화된 비디오 데이터를 포함하는 전송 스트림 패킷, 부호화된 오디오 데이터를 포함하는 전송 스트림 패킷, 및 전용 데이터를 포함하는 전송 스트림 패킷이 혼합된다. 이에 대해, 전용 데이터는 부호화 제어기(70)에 의해 전송 스트림 패킷에 놓이고, 이어서 네트워크를 통해 각 다중화 회로에 공급된다.

부호화 제어기(70)는 비디오 및 오디오 인코더 각각에 대해 적절한 부호화 비트 레이트를 지정한다. 예를 들어, 부호화 제어기(70)는 더 많은 부호화 비트를 요구하는 스포츠와 같은 프로그램을 부호화하기 위한 인코더에는 더 높은 비트 레이트를 지정하고, 더 적은 부호화 비트를 발생할 수 있는 뉴스와 같은 프로그램을 부호화하기 위한 인코더에는 더 낮은 비트 레이트를 지정한다. 즉, 다른 채널에 상대적으로 각 채널에 대해 비디오 데이터의 복잡성 (얼마나 많은 비트가 부호화에 의해 발생되는가에 대한 인덱스로 사용되는)을 포착하여, 최고 복잡성을 갖는 비디오 데이터를 위한 채널에는 최고 비트 레이트를 지정하고 내림 차순으로 더 낮은 다른 비트 레이트를 지정한다. 물론, 지정된 비트 레이트는 프로그램에 특정된 것이 아니고, 그 프로그램에서 비디오 데이터의 복잡성에 의존해 변할 수 있다.

다음에는 이 전송 스트림 패킷에 대한 구조와 구문 (syntax)이 도 6 내지 도 9를 참조하여 이후 상세히 설명된다.

전송 스트림 패킷은 4 바이트의 헤더와, 다양한 종류의 데이터 및 데이터 요소를 저장하기 위한 184 바이트의 페이로드 섹션으로 구성된다.

전송 스트림 패킷의 헤더는 다음의 필드로 구성된다: sync_byte, transport_error_indicator, payload_unit_start_indicator, transport_priority_PID, transport_scrambling_control, adaptation_field_control, continuity_counter, 및 adaptation_field.

sync_byte 필드는 8 비트의 고정 길이를 갖는 필드로서, 비트 스트림에서 동기화 패턴을 검출한다. 이 필드는 "01000111" (0x47)의 고정값을 갖도록 정의되고, 스트림에서 이 비트 패턴을 검출함으로서 동기화 조건을 검출할 수 있다.

transport_error_indicator 필드는 1 비트의 플래그로서, "1"로 설정될 때는 적어도 1 비트의 정정가능하지 않은 에러가 전송 스트림 패킷에 존재함을 나타낸다.

payload_unit_start_indicator 필드는 1 비트의 플래그로서, 비디오/오디오 데이터와 다른 기본 데이터 또는 프로그램 지정 정보 (PSI)를 운반하는 전송 스트림 패킷에 중요한 의미를 갖는다. 전송 스트림 패킷의 페이로드 섹션이 기본 데이터를 포함하면, payload_unit_start_indicator 필드는 다음의 의미를 갖는다: payload_unit_start_indicator 필드가 "1"일 때, 이는 기본 데이터가 이 전송 스트림 패킷의 페이로드 섹션 중 시작부에 삽입됨을 나타내고; payload_unit_start_indicator 필드가 "0"일 때, 이는 기본 데이터가 이 전송 스트림 패킷의 페이로드 섹션 중 시작부에 삽입되지 않음을 나타낸다. payload_unit_start_indicator 필드가 "1"로 설정되면, 이는 단 하나의 PES 패킷이 특정한 전송 스트림 패킷에서 시작됨을 나타낸다. 반대로, 전송 스트림 패킷의 페이로드 섹션이 PSI 데이터를 포함하면, payload_unit_start_indicator 필드는 다음의 의미를 갖는다: 전송 패킷이 PSI 부의 제 1 바이트를 운반할 때, payload_unit_start_indicator 필드는 "1"이고; 전송 스트림이 PSI 부의 제 1 바이트를 운반하지 않을 때, payload_unit_start_indicator 필드는 "0"이다. 전송 스트림 패킷이 널 패킷 (null packet)이면, payload_unit_start_indicator 필드는 또한 "0"이다.

transport_priority 필드는 전송 패킷의 우선순위를 나타내기 위한 1 비트의 식별자이다. transport_priority 필드가 "1"로 설정될 때, 이 전송 패킷은 같은 패킷 식별자 PID를 갖는 패킷으로, 이 패킷이 "1" 이외의 값으로 설정된 transport_priority 필드를 갖는 것 보다 더 높은 우선순위를 가짐을 나타낸다. 예를 들어, 기본 스트림내의 특정한 패킷은 transport_priority 필드의 패킷 식별자를 설정함으로서 우선적으로 지정될 수 있다.

transport_scrambling_control 필드는 전송 스트림 패킷 페이로드의 스크램블 모드를 나타내기 위한 2 비트의 데이터이다. 스크램블 모드는 페이로드에 저장된 데이터가 스크램블링되었는가의 여부와 스크램블의 종류를 나타낸다. 이는 전송 스트림 패킷 헤더와 적응 필드가 스크램블 키(Ks)로 스크램블링되지 말아야 한다는 표준에 의해 요구된다. 그러므로, 이 transport_scrambling_control 필드를 사용함으로서, 전송 스트림 패킷 페이로드에 저장된 데이터가 스크램블링되었는가의 여부를 결정할 수 있다.

adaptation_field_control 필드는 적응 필드 및/또는 페이로드가 이 전송 스트림의 패킷 헤더에 놓임을 나타내기 위한 2 비트의 데이터이다. 특별히, 페이로드 데이터만이 패킷 헤더에 놓이면, adaptation_field_control 필드는 "01"이고, 적응 필드만이 패킷 헤더에 놓이면, adaptation_field_control 필드는 "10"이고, 적응 필드와 페이로드가 모두 패킷 헤더에 놓이면, adaptation_field_control 필드는 "11"이다.

continuity_counter 필드는 같은 PID를 갖는 연속적으로 전송되는 패킷이 전송 동안 부분적으로 비거나 버려지는가의 여부를 나타내기 위한 데이터이다. 특별히, continuity_counter 필드는 같은 PID를 갖는 전송 스트림 패킷이 전송될 때마다 증가되는 4 비트의 필드이다. 그러나, 이 continuity_counter는 적응 필드가 패킷 헤더에 놓일 때에만 카운트됨을 이해하여야 한다.

adaptation_field는 분리된 스트림에 추가 정보를 삽입하거나 옵션으로 바이트를 채워넣기 위한 필드이다. 이 적응 필드를 사용함으로서, 분리된 스트림의 동적 상태 전이에 대한 모든 정보가 데이터와 함께 전송될 수 있다.

adaptation_field는 다음의 필드로 구성된다: adaptation_field_length, discontinuity_counter, random_access_indicator, elementary_stream_priority_indicator, PCR_flag, OPCR_flag, splicing_point_flag, splicing_point, transport_private_data_flag, adaptation_field_extension_flag, program_clock_reference (PCR), original_program_clock_reference (OPCR), splice_countdown, transport_private_data_length, private_data, adaptation_field_extension_length, ltw_flag (legal_time_window flag), piecewise_rate_flag, 및 seamless_splice_flag.

adaptation_field_length 필드는 이 adaptation_field_length 필드에 이어지는 적응 필드의 바이트수를 나타내는 데이터이다. adaptation_field_control 필드가 "11"이면, adaptation_field_length는 0 내지 182 비트이고; adaptation_field_control이 "10"이면, adaptation_field_length는 183 비트이다. 기본 스트림이 전송 스트림의 페이로드에 채워지기에 충분하지 않을 때는 이것을 일부 비트로 채우도록 주입하기가 요구됨을 이해하여야 한다.

discontinuity_counter 필드는 시스템 클럭 기준 (SCR)이 불연속적인 같은 PID를 갖는 복수의 패킷 도중에 재설정되는가의 여부를 나타내는 데이터이다. 시스템 클럭 기준이 불연속적이면, 이 discontinuity_counter 필드는 "1"이고; 시스템 클

럭 기준이 아직 연속적이면, 이 discontinity_counter 필드는 "0"이다. 시스템 클럭 기준은 비디오 및 오디오 데이터를 복호화하기 위한 MPEG 디코더가 디코더에 대한 시스템 시간 클럭을 인코더에 의해 의도된 타이밍 값으로 설정하는데 사용하는 기준 정보임을 이해하여야 한다.

random_access_indicator 필드는 비디오 순차 헤더나 오디오 프레임의 시작부를 나타내는 데이터이다. 즉, 이 random_access_indicator 필드는 데이터 요소가 무작위로 액세스될 때 비디오 또는 오디오 데이터에 대한 액세스점 (프레임의 시작부인)을 나타내는 데이터이다.

elementary_stream_priority_indicator 필드는 같은 PID를 갖는 패킷에서 이 전송 스트림 패킷의 페이로드에 운반되는 기본 스트림 데이터의 우선순위를 나타내는 데이터이다. 예를 들어, 기본 스트림의 비디오 데이터가 인트라-코드화 (intra-code)될 때, elementary_stream_priority_indicator 필드는 "1"로 설정된다. 반대로, 인터-코드화 (inter-code)화된 비디오 데이터를 포함하는 전송 스트림의 elementary_stream_priority_indicator는 "0"으로 설정된다.

PCR_flag 필드는 PCR (program_clock_reference) 데이터가 적응 필드에 존재하는가의 여부를 나타내는 데이터이다. PCR 데이터가 적응 필드에 존재하면, PCR_flag 필드는 "1"로 설정되고; PCR 데이터가 존재하지 않으면, PCR_flag 필드는 "0"으로 설정된다. PCR 데이터는 전송된 데이터를 복호화하기 위한 복호화 동작의 적절한 타이밍을 구하기 위해 수신단에서 디코더에 의해 사용됨을 이해하여야 한다.

OPCR_flag 필드는 OPCR (original_program_clock_reference) 데이터가 적응 필드에 존재하는가의 여부를 나타내는 데이터이다. OPCR 데이터가 적응 필드에 존재하면, OPCR_플래그 필드는 "1"로 설정되고; OPCR 데이터가 존재하지 않으면, OPCR_플래그 필드는 "0"으로 설정된다. OPCR 데이터는 복수의 원래 전송 스트림으로부터 전송 스트림을 재구성하는 접속 동작에 의해 사용되어 원래 전송 스트림에 대한 PCR 데이터를 나타냄을 이해하여야 한다.

splicing_point_flag 필드는 편집점 (접속점)을 전송 레벨로 나타내기 위해 splice_countdown 데이터가 적응 필드에 존재하는가의 여부를 나타내는 데이터이다. splice_countdown 데이터가 적응 필드에 존재하면, splicing_point_flag 필드는 "1"이고; splice_countdown 데이터가 적응 필드에 존재하지 않으면, splicing_point_flag 필드는 "0"이다.

transport_private_data_flag 필드는 특정한 사용자 데이터를 설명하는데 사용되는 전용 데이터가 적응 필드에 존재하는가의 여부를 나타내는 데이터이다. 전용 데이터가 적응 필드에 존재하면, transport_private_data_flag 필드는 "1"로 설정되고; 전용 데이터가 적응 필드에 존재하지 않으면, transport_private_data_flag 필드는 "0"으로 설정된다.

adaptation_field_extension_flag 필드는 확장 필드가 적응 필드에 존재하는가의 여부를 나타내는 데이터이다. 확장 필드가 적응 필드에 존재하면, adaptation_field_extension_flag 필드는 "1"로 설정되고; 확장 필드가 적응 필드에 존재하지 않으면, adaptation_field_extension_flag 필드는 "0"으로 설정된다.

program_clock_reference (PCR)는 수신단에 있는 클럭이 전송단에 있는 클럭과 동위상이 될 때 기준되는 기준 클럭이다. 이 PCR 데이터는 각 전송 패킷이 발생될 때의 시간 데이터를 포함한다. PCR 데이터는 42 비트의 길이로서, 33 비트의 program_clock_reference_base와 9 비트의 program_clock_reference_extension으로 구성된다. 시스템 클럭을 program_clock_reference_extension으로 0 내지 299의 범위에서 카운트하고 카운트 299가 0으로 재설정될 때 발생된 캐리 (carry)를 갖는 program_clock_reference_base에 한 비트를 추가함으로서 24 시간이 카운트될 수 있다.

original_program_clock_reference (OPCR) 필드는 특정한 전송 스트림으로부터 단일 프로그램 전송 스트림을 재구성하는데 사용되는 데이터이다. 이러한 프로그램 전송 스트림이 완전하게 재구성될 때, original_program_clock_reference는 program_clock_reference 필드로 복사된다.

splice_countdown 필드는 같은 PID를 갖는 전송 스트림 패킷에서 전송 스트림 패킷 레벨에 있는 편집가능한 (접속가능한) 지점에 이를 필요가 있는 패킷의 수를 나타내는 데이터이다. 그러므로, 편집가능한 접속점에 있는 전송 스트림 패킷은 "0"으로 설정된 splice_countdown 필드를 갖는다. "0"으로 설정된 splice_countdown 필드를 갖는 전송 패킷에서, 전송 스트림 패킷 페이로드의 최종 바이트는 이것을 부호화된 화상의 최종 바이트로 재배치함으로서 접속될 수 있다.

이에 대해, 접속 동작은 전송 레벨에서 두 개의 다른 기본 스트림을 결부시키도록 실행되어 새로운 전송 스트림을 형성한다. 접속 동작은 두 종류로 나뉠 수 있다: 복호화의 불연속성을 갖지 않는 심리스 접속 (seamless splice)과 복호화의 불연속성을 갖는 난심리스 접속 (nonseamless splice). 복호화의 불연속성이 일어나지 않는 경우는 새롭게 부가된 스트림의

액세스 유닛에 대한 복호화 시간이 접속 이전의 오래된 스트림의 액세스 유닛에 대한 복호화 시간과 일치함을 의미한다. 반대로, 복호화의 불연속성이 일어나는 경우는 새롭게 부가된 스트림의 액세스 유닛에 대한 복호화 시간이 접속 이전의 오래된 스트림의 액세스 유닛에 대한 복호화 시간과 일치하지 않음을 의미한다.

transport_private_data_length 필드는 적응 필드에서 전용 데이터의 바이트수를 나타내는 데이터이다.

private_data는 표준에 의해 특별히 요구되지는 않지만 적응 필드에서 특정한 사용자 데이터를 설명하는데 사용될 수 있는 필드이다.

adaptation_field_extension_length 필드는 적응 필드에서 적응 필드 확장 데이터의 길이를 나타내는 데이터이다.

ltw_flag (legal_time_window_flag) 필드는 디스플레이 윈도우의 오프셋 (offset) 값을 나타내는 ltw_offset 데이터가 적응 필드에 존재하는가의 여부를 나타내는 데이터이다.

piecewise_rate_flag 필드는 piecewise_rate 데이터가 적응 필드에 존재하는가의 여부를 나타내는 데이터이다.

seamless_splice_flag 필드는 접속점이 정상적인 접속점인가 심리스 접속점인가의 여부를 나타내는 데이터이다. seamless_splice_flag 필드가 "0"일 때, 이는 접속점이 정상적인 접속점임을 나타내고; seamless_splice_flag 필드가 "1"일 때, 이는 접속점이 심리스 접속점임을 나타낸다. 정상적인 접속점은 접속점이 PES 패킷내의 브레이크 (break)에 존재하고 이 접속점에 바로 선행하는 접속 패킷이 액세스 유닛으로 끝나고 같은 PID를 갖는 다음 전송 패킷이 PES 패킷의 헤더에서 시작됨을 의미한다. 반대로, 심리스 접속점은 접속점이 PES 패킷의 중간에 존재하고 새롭게 부가된 스트림의 액세스 유닛에 대한 복호화 시간이 접속 이전의 오래된 스트림의 액세스 유닛에 대한 복호화 시간과 일치하도록 하기 위해, 오래된 스트림의 특징이 부분적으로 새로운 스트림의 것으로 사용됨을 의미한다.

다음에는 도 10을 참조하여 멀티플렉서 시스템(8)이 이후 상세히 설명된다. 멀티플렉서 시스템(8)은 PAT 패킷, PMT 패킷, CAT 패킷, 부호화된 기본 데이터를 포함하는 전송 스트림 패킷, EPG 패킷, ECM 패킷, 및 EMM' 패킷을 다중화하여 전송 스트림을 발생하는 시스템이다.

특별히, 멀티플렉서 시스템(8)는 멀티플렉서 시스템(8)내의 모든 회로를 제어하기 위한 멀티플렉서 제어기(81), 가입자 시청허가 시스템(3)에 의해 공급된 ECM 패킷을 암호화하기 위한 암호화 블록(82), 각각 프로그램 지정 정보(PSI)로서 인코더/멀티플렉서 시스템(9)에 의해 공급된 PAT 패킷, PMT 패킷, 및 CAT 패킷을 버퍼링하는 FIFO 버퍼 (841) 내지 (843), 각각 복수의 프로그램들을 포함하는 전송 스트림 패킷을 버퍼링하는 FIFO 버퍼 (851) 내지 (859), 각각 ECM 패킷, EMM 패킷, 및 EPG 패킷을 버퍼링하는 FIFO 버퍼 (861) 내지 (863), 각 FIFO 버퍼의 판독/기록 동작을 제어할 뿐만 아니라 각 FIFO 버퍼의 자유 영역을 모니터하기 위한 FIFO 제어기(83), FIFO 버퍼에 의해 공급된 전송 스트림 패킷을 다중화하여 다중화된 전송 스트림을 발생하기 위한 다중화 회로(87), 및 다중화된 전송 스트림 패킷에 포함된 데이터 요소를 스크램블링하기 위한 스크램블 블록(88)을 구비한다.

멀티플렉서 제어기(81)는 인코더/멀티플렉서 제어 유닛(9)으로부터 EPG 패킷, EMM' 패킷, ECM 패킷 (ECM1 내지 ECM19), PAT 패킷, PMT 패킷, 및 CAT 패킷을 수신하여 각 패킷을 적절한 회로에 공급한다. 멀티플렉서 제어기(81)는 또한 인코더/멀티플렉서 제어 유닛(9)으로부터 PID 테이블을 수신하고 이를 스크램블 블록(88)에 공급한다.

암호화 블록(82)은 인코더/멀티플렉서 제어 유닛(9) 및 멀티플렉서 제어기(81)를 통해 가입자 시청허가 시스템(3)에 의해 공급된 복수의 ECM 패킷에 포함된 스크램블 키(Ks)를 암호화하기 위한 블록이다. 암호화 블록(82)은 암호화 동작 이전에 가입자 시청허가 시스템(3)으로부터 다운로드된 작업키 테이블을 저장하기 위한 RAM(821)과, 작업키(Kw)에 기초하여 ECM에 포함된 스크램블 키(Ks)를 암호화하기 위한 암호기(822)를 갖는다.

이제, 예를 들어 제 1 프로그램을 구성하는 기본 데이터를 스크램블링하기 위한 스크램블 키(Ks1)가 암호화되는 경우를 설명함으로서, 암호화 블록(82)의 암호화 동작이 이후 설명된다. 먼저, ECM 패킷(ECM1)에 포함된 작업키 번호(Kw_No)는 RAM(821)에 공급된다. RAM(821)은 가입자 시청허가 시스템으로부터 다운로드된 작업키 테이블을 저장하고, 이는 작업키 번호(Kw_No)와 작업키(Kw) 사이의 대응관계를 나타낸다. 패킷(ECM1)에 포함된 작업키 번호(Kw_No)는 RAM(821)에 공급되므로, 공급된 작업키 번호(Kw_No)에 대응하는 작업키(Kw)는 RAM(821)에 의해 제공된다. 암호화 회로(822)는 스크램블 키(Ks1)를 포함하는 ECM 패킷을 수신하고, 또한 RAM(821)으로부터 작업키(Kw)를 수신한다. 암호화 회로(822)는 작업키(Kw)로 소스 ECM에 포함된 스크램블 키(Ks1)를 암호화하고, 이어서 암호화된 스크램블 키(Ks1')를 발생한다. 암호화 회로(822)는 암호화되지 않은 스크램블 키(Ks) 보다는 암호화된 스크램블 키(Ks1')를 소스 ECM에 축적

하고, 암호화된 스크램블 키(Ks1)를 포함한 전송 스트림 패킷을 암호화된 ECM 패킷으로 제공한다. 암호화 알고리즘은 EMM 데이터에 포함된 작업키(Kw)를 암호화하기 위해 가입자 시청허가 시스템(3)에서 사용된 것, 즉 암호화 알고리즘 CRYSP와 같음을 이해하여야 한다.

암호화 블록(82)은 ECM 데이터로서 공급되는 스크램블 키(Ks1)를 스크램블 블록(88)으로 공급한다. 스크램블 블록(88)으로 공급되는 스크램블 키(Ks1)는 작업키(Kw)로 암호화되지 않았음을 이해하여야 한다.

상기의 예에서는 제 1 ECM 패킷(ECM)으로 축적되는 스크램블 키(Ks1)가 상술된 바와 같이 암호화되지만, 다른 스크램블 키 (Ks2) 내지 (Ks19)에 대한 암호화 동작은 동일하게 실행된다. 그러므로, 암호화된 ECM 패킷 (ECM1') 내지 (ECM19')은 암호화 블록(82)에 의해 FIFO 버퍼(861)에 제공되고, 암호화되지 않은 스크램블 키 (Ks1) 내지 (Ks19)은 또한 암호화 블록(82)에 의해 스크램블 블록(88)으로 제공된다.

다중화 회로(87)는 각 FIFO내의 데이터량에 의존해, FIFO (841) 내지 (843)에서 버퍼링된 PAT 패킷, PMT 패킷, 및 CAT 패킷으로부터 선택된 일부 패킷, FIFO (851) 내지 (859)에서 버퍼링된 복수의 프로그램들 Program1 내지 Program9을 포함한 전송 스트림 패킷, 및 FIFO (861) 내지 (863)에서 버퍼링된 ECM 패킷, EMM 패킷, 및 EPG 패킷을 스크램블 블록(88)에 공급한다. 이에 대해, FIFO 제어기(83)는 각 FIFO에 공급된 패킷 데이터에 대한 입력 레이트를 모니터하고, 각 FIFO에 대한 모든 입력 레이트의 합이 소정의 출력 레이트를 넘을 때 멀티플렉서 제어기(81)에 과도 입력 레이트 상태를 알린다. 그래서, 멀티플렉서 제어기(81)는 다중화 회로(87)에 의해 선택된 타켓 FIFO로부터의 EMM 패킷을 버퍼링하는 FIFO(862)를 제외시키고, 다른 FIFO를 선택 타켓으로 스위칭하여 스위칭된 FIFO로부터의 패킷 데이터를 스크램블 블록에 공급한다. 그러므로, 다중화 회로(87)에 의해 선택되는 FIFO의 수가 감소될 때, 타켓 FIFO에 대한 선택 동작의 수는 증가되고, 그에 의해 각 FIFO의 데이터 출력 레이트는 그에 따라 과도 입력 레이트 상태로 증가될 수 있다.

각 FIFO에 대한 모든 입력 레이트의 합이 소정의 출력 레이트 이하일 때, 멀티플렉서 제어기(81)는 FIFO(862)를 다중화 회로(87)에 의해 선택된 타켓 FIFO로 재저장함으로서 FIFO(862)에 버퍼링된 EMM 패킷을 스크램블 블록(88)에 공급한다. FIFO(862)에서 버퍼링된 가입 정보를 나타내는 EMM 데이터는 자주 바뀔 수 없어서 FIFO가 다중화 회로(87)에 의해 선택된 타켓 FIFO로부터 일시적으로 제외되더라도 실제로 불리한 상황이 발생되지 않음을 이해하여야 한다.

다중화 회로(87)에 의해 선택된 타켓 FIFO로부터 단일 FIFO(862)를 제외시키는 것만이 이러한 과도 입력 레이트 조건을 다룰 수 없도록 하는 범위로 각 FIFO에 대한 모든 입력 레이트의 합이 더 증가될 때, 멀티플렉서 제어기(81)는 EMM 데이터를 버퍼링하기 위한 FIFO(862)를 제외시킬 뿐만 이니라 EPG 데이터를 버퍼링하기 위한 FIFO(863)도 제외시킨다. 이러한 추가 제외는 다중화 회로(87)에 의해 선택되는 FIFO의 수를 더 감소시킬 수 있고, 타켓 FIFO에 대한 선택 동작의 수를 더 증가시켜, 그에 의해 각 FIFO의 데이터 출력 레이트는 그에 대응하여 과도 입력 레이트 상태로 더 증가될 수 있다. FIFO (863)에서 버퍼링된 프로그램 안내 정보를 나타내는 EPG 데이터는 자주 바뀔 수 없어서 FIFO가 다중화 회로(87)에 의해 선택된 타켓 FIFO로부터 일시적으로 제외되더라도 실제로 불리한 상황이 발생되지 않음을 이해하여야 한다.

스크램블 블록(88)은 PID 검출 회로(881)와 스크램블 회로(882)를 갖는다. PID 검출 회로(881)는 다중화 회로(87)로부터 전송 스트림 패킷을 수신할 뿐만 아니라 멀티플렉서 제어기(81)를 통해 인코더/멀티플렉서 제어 유닛(9)으로부터 PID 테이블 (도 4)을 수신한다. PID 테이블은 전송 스트림 패킷을 발생할 때 사용되는 스크램블 키와 PID 값 사이의 대응관계를 나타내므로, PID 검출 회로(881)는 PID 테이블에 저장된 PID 값을 참조하여 원하는 스크램블 키(Ks)를 검출하고, 검출된 스크램블 키(Ks)를 스크램블 회로(882)에 공급한다. 스크램블 회로(882)는 스크램블 키(Ks)와 연관된 프로그램 데이터를 포함하는 전송 스트림 패킷을 스크램블링하는데 PID 검출 회로(881)에 의해 공급된 스크램블 키(Ks)를 사용하고, 변조 회로(10) (도 1)에 출력을 제공한다.

이제는 프로그램 지정 정보가 이후 설명된다.

프로그램 지정 정보 (PSI: Program_Specific_Information)는 복수의 프로그램들과 데이터를 다중화함으로서 얻어진 전송 스트림 패킷 중에서 어느 데이터가 어느 패킷에 포함되는가를 나타내는 정보이다. 그러므로, 디코더는 이 프로그램 지정 정보를 참조하여 원하는 데이터를 복호화할 수 있다.

프로그램 지정 정보는 도 11에 도시된 바와 같은 4개의 테이블 구조로 분류될 수 있다. 프로그램 연관 테이블 (PAT: Program_Association_Table)은 지정된 프로그램 번호와 그에 대응하는 프로그램 맵 테이블 (PMT) PID를 나타내는 테이블이다. 프로그램 맵 테이블 (PMT: Program_Map_Table)은 지정된 프로그램의 요소가 설명되는 패킷의 PID를 나타내는 테이블이다. 네트워크 정보 테이블 (NIT: Network_Information_Table)은 네트워크 매개변수를 전송하는데 사용되지

만 특별히 표준에 의해 요구되지 않는 정보를 나타내는 전용 테이블이다. 조건부 액세스 테이블 (CAT: Conditional_ Access_Table)은 고유의 PID를 EMM 패킷에 지정하는 테이블이다. 다음의 설명에서는 프로그램 연관 테이블, 프로그램 맵 테이블, 및 조건부 액세스 테이블이 상세히 설명된다.

먼저, 도 12 및 도 13을 참조하여 프로그램 연관 테이블이 이후 설명된다.

프로그램 연관 테이블(PAT)은 프로그램의 내용을 지정하는 전송 스트림 패킷의 PID와 각 전송된 프로그램을 지정하는데 사용되는 테이블이다. 특별히, 프로그램 연관 테이블(PAT)은 다음으로 구성된다: table_id, section_syntax_indicator, section_length, transport_stream_id, version_number, current_next_indicator, section_number, last_section_ number, program_number, network_PID, 및 program_map_PID.

table_id는 도 14에 도시된 바와 같이 각 테이블에 지정된 고유의 식별 번호를 도시한다. 프로그램 연관 테이블(PAT)의 table_id는 "0x00"이고, 조건부 액세스 테이블(CAT)의 table_id는 "0x01"이고, 또한 프로그램 맵 테이블의 table_id는 "0x02"이다.

section_syntax_indicator는 "1"로 고정된 상수이다.

section_length는 이 section_length에 이어지는 비트로부터 CRC부의 최종 바이트까지의 길이를 바이트로 나타내는 필드이다.

transport_stream_id는 이 전송 스트림과 네트워크에서 다중화된 다른 전송 스트림간을 구별하는 식별 데이터이다.

version_number는 프로그램 연관 테이블(PAT)의 버전 수를 나타내는 데이터이다. 버전 수는 0 내지 31 범위의 정수이고, 프로그램 연관 테이블에서의 설정이 변할 때, 즉 전송 스트림의 특성이 변할 때 1 만큼 증가된다. 수신단에서의 디코더는 최종 버전의 섹션만이 유효함을 결정하는데 version_number를 참고한다.

current_next_indicator는 전송된 프로그램 연관 테이블이 현재 이용가능한가의 여부를 나타내는 데이터이다.

section_number는 프로그램 연관 테이블 섹션 번호를 나타내는 데이터이다. 예를 들면, 프로그램 연관 테이블에 포함된 program_association_section()의 section_number는 이것이 제 1 섹션임을 나타내도록 "0x00"으로 설정된다.

last_section_number는 프로그램 연관 테이블에서 지정된 모든 섹션 중에서 최종 섹션의 섹션 번호를 나타내는 데이터이다. 그러므로, last_section_number는 시스템이 프로그램 연관 테이블에 지정된 섹션의 수를 알 수 있게 한다.

program_number는 다중화되는 복수의 프로그램들 각각에 주어진 고유의 번호이다. 예를 들면, 본 발명에 따른 데이터 전송 장치에서, 프로그램 번호 (1) 내지 (9)는 각각 다중화되는 9개 프로그램에 지정된다. program_number는 사용자에 의해 임의로 정의될 수 있는 데이터이다. 그러나, program_number가 0으로 설정될 때, program_number는 네트워크 정보 테이블(NIT)이 스트림에 존재함을 나타낸다.

network_PID는 스트림에서 네트워크 정보 테이블(NIT)이 설명되는 PID를 나타내는 데이터이다. 네트워크 정보 테이블 (NIT)은 사용자에 의해 임의로 설정될 수 있고 주어진 장치에서는 사용되지 않는 테이블이다.

program_map_PID는 program_number에 의해 지정된 프로그램에 적용되는 프로그램 맵 테이블을 갖는 전송 스트림 패킷의 PID를 나타내는 데이터이다. 예를 들어, 프로그램이 비디오 데이터 1개와 오디오 데이터 4개로 구성되면, 그 프로그램의 비디오 데이터를 갖는 전송 스트림 패킷을 지정하는 한 개의 PID와 그 프로그램의 오디오 데이터를 갖는 전송 스트림 패킷을 지정하는 4개의 PID는 프로그램 연관 테이블에서 설명된다.

다음에는 도 15 및 도 16을 참조하여 프로그램 맵 테이블(PMT)이 이후 설명된다.

프로그램 맵 테이블은 프로그램 번호와 이들 프로그램을 구성하는 데이터 요소 사이의 맵핑 (mapping)을 도시하는데 사용되는 테이블이다. 즉, 프로그램 맵 테이블은 각 프로그램 번호에 대해, 비디오 데이터, 오디오 데이터, 및 그 프로그램을 구성하는 다른 추가 데이터와 같은 요소를 전송하도록 전송 스트림 패킷의 PID를 지정하는 섹션이다. 특별히, 프로그램 맵

테이블은 다음으로 구성된다: table_id, section_syntax_indicator, section_length, program_number, version_number, current_next_indicator, section_number, last_section_number, PCR_PID, program_info_length, descriptor(), stream_type, elementary_PID, ES_info_length, 및 descriptor().

table_id는 각 테이블을 식별하도록 지정된 고유의 식별 번호를 나타낸다. 프로그램 맵 테이블의 table_id는 "0x02"이다.

section_syntax_indicator는 "1"로 고정된 상수이다.

section_length는 이 section_length에 이어지는 비트로부터 CRC 섹션의 최종 바이트까지의 길이를 바이트로 나타내는 필드이다.

program_number는 다중화되는 복수의 프로그램들 각각에 주어진 고유의 번호이다. 예를 들면, 본 발명에 따른 데이터 전송 장치에서, 프로그램 번호 1 내지 9는 각각 다중화되는 9개의 프로그램에 지정된다.

version_number는 프로그램 맵 테이블을 구성하는 프로그램 맵 섹션의 버전 수를 나타내는 데이터이다. 버전 수는 프로그램 맵 섹션에서 전송되는 데이터가 변할 때 증가된다.

current_next_indicator는 전송되는 프로그램 연관 테이블이 현재 이용가능한가의 여부를 나타내는 데이터이다. 프로그램 연관 테이블이 현재 이용가능하면, current_next_indicator는 "1"로 설정되고, 프로그램 연관 테이블이 현재 이용가능하지 않으면, current_next_indicator는 "0"으로 설정된다.

section_number는 프로그램 맵 테이블에 포함된 섹션의 번호를 나타내는 데이터이고, 프로그램 맵 테이블을 구성하기 위해서는 단 하나의 프로그램 맵 섹션이 존재하기 때문에, 이는 항상 "0x00"으로 설정된다.

last_section_number는 프로그램 맵 테이블에 지정된 모든 섹션 중에서 최종 섹션의 섹션 번호를 나타내는 데이터이다. 그러므로, last_section_number는 항상 프로그램 맵 테이블을 구성하는 프로그램 맵 섹션에서 "0x00"으로 설정된다.

PCR_PID는 program_number에 의해 지정된 프로그램에 대해 유효한 PCR 데이터를 포함하는 전송 패킷의 PID를 나타내는 데이터이다.

program_info_length는 program_info_length 필드에 이어서 설명되는 descriptor()에 대한 바이트의 수를 지정하는 데이터이다.

descriptor()는 프로그램 요소와 프로그램의 정의를 확장하는데 사용되는 데이터 구조이다. 다양한 디스크립터(discriptor)가 가능하다. 예를 들면, 비디오 기본 스트림에 대한 부호화 매개변수를 식별하는 기본 정보를 설명하기 위한 video_stream_descriptor(); 오디오 기본 스트림에 대한 부호화 버전을 식별하는 기본 정보를 설명하기 위한 audio_stream_descriptor(); 복수의 스트림으로 다중화되는 계층구조로 부호화된 비디오 및 오디오 데이터를 포함하는 프로그램 요소를 식별하는 정보를 설명하기 위한 hierarchy_descriptor(); 전용 데이터를 유일하게 정의하여 식별하는 정보를 설명하기 위한 resistration_descriptor(); 서로 연관된 기본 스트림 사이에 존재하는 정렬 종류를 설명하기 위한 data_stream_alignment_descriptor(); 전송되는 비디오 데이터에 대한 디스플레이 윈도우의 배경에 디스플레이되는 배경 윈도우를 지정하는 정보를 설명하는 target_background_descriptor; 전송되는 비디오 데이터에 대한 디스플레이 윈도우의 디스플레이 위치를 지정하는 정보를 설명하기 위한 video_window_descriptor; 가입 정보 EMM, 프로그램 해독 정보 ECM, 및 다른 데이터를 설명하기 위한 CA_descriptor(); 연관된 프로그램 요소에 의해 사용되는 언어를 식별하는 정보를 설명하기 위한 language_descriptor(); 시간 스탬프 (stamp)를 발생하는데 사용되는 시스템 클럭에 대한 정보를 전송하기 위한 system_clock_descriptor(); video_buffer_verifier를 포함하는 STD (system_target_decoder) 다중화 버퍼에서 데이터 점유량의 언더플로우 및 오버플로우의 주요 레벨을 나타내는 데이터를 설명하기 위한 multiplex_buffer_utilization_descriptor(); 저작권을 보호하도록 오디오/비디오 작업의 식별을 허용하는 정보를 설명하기 위한 copyright_descriptor(); 전송되는 데이터 요소에 대한 최대 비트 레이트를 지정하는 정보를 설명하기 위한 maximun_bitrate_descriptor(); 전용 데이터의 전송을 지정하기 위한 private_data_indicator_descriptor(); 평활화 버퍼의 크기와 그 버퍼로부터의 출력 누설 비율에 대한 정보를 설명하기 위한 smoothing_buffer_descriptor(); STD 버퍼의 누설값을 지정하기 위한 STD_descriptor(); 및 부호화 종류에 대한 정보를 설명하기 위한 ibp_descriptor().

본 발명에 따른 데이터 전송 장치에서, 프로그램 맵 테이블에 사용되는 descriptor()의 데이터 구조는 프로그램 해독 정보 ECM를 포함하는 전송 스트림 패킷의 PID를 지정하는 정보를 설명하도록 의도됨을 이해하여야 한다.

stream_type은 이후 설명되는 elementary_PID에 의해 지정되는 PID를 갖는 패킷에 포함된 프로그램 요소의 종류를 지정하기 위한 데이터이다. 예를 들어, 패킷에 포함된 프로그램 요소가 ISO/IEC11172 표준에 일치하는 비디오 데이터이면, stream_type은 "0x01"로 설정되고, 프로그램 요소가 ISO/IEC13818-2 표준에 일치하는 비디오 데이터이면, "0x02"로 설정되고, 프로그램 요소가 ISO/IEC11172 표준에 일치하는 오디오 데이터이면 "0x03"으로 설정되고, 또한 프로그램 요소가 ISO/IEC13818-3 표준에 일치하는 오디오 데이터이면 "0x04"로 설정된다. 프로그램 맵 섹션에서, stream_type은 program_number에 의해 지정된 프로그램을 구성하는 요소의 수만큼 반복된다.

elementary_PID는 program_number에 의해 지정된 프로그램을 구성하는 요소를 전송하는 전송 스트림 패킷의 PID를 지정하기 위한 필드이다. elementary_PID는 스트림 요소의 종류를 나타내는 stream_type에 대응하여 설명되는 데이터이다.

ES_info_length는 ES_info_length 필드에 이어서 설명되는 디스크립터의 바이트 수를 지정하기 위한 데이터이다.

다음에는 도 17 및 도 18을 참조하여 조건부 액세스 테이블(CAT)이 이후 상세히 설명된다. 조건부 액세스 테이블은 유효 방송 시스템에서 비디오 데이터 및 오디오 데이터를 디스크램블링하는데 사용되는 가입 정보 EMM를 전송하는 패킷의 PID를 지정하기 위한 섹션이다. 조건부 액세스 테이블은 또한 가입자만이 프로그램을 디스크램블링하도록 허용하는 조건부 액세스 시스템 (CA 시스템: Conditional Access System)과 가입 정보 EMM 사이의 관계를 지정하기 위한 섹션이다.

조건부 액세스 테이블의 PID는 프로그램 맵 테이블과 같이 프로그램 연관 테이블에 의해 지정되지 않으므로, 조건부 액세스 테이블의 각 필드는 비트 스트림으로부터 조건부 액세스 테이블에 지정된 고유의 PID 값 "0x01"을 찾음으로서 복호화될 수 있다.

특별히, 조건부 액세스 테이블은 다음으로 구성된다: table_id, section_syntax_indicator, section_length, version_number, current_next_indicator, section_number, last_section_number, 및 descriptor().

table_id는 각 테이블에 지정된 고유의 식별 번호를 도시하고, 조건부 액세스 테이블(CAT)에 지정된 table_id는 "0x01"이다.

section_syntax_indicator는 "1"로 고정된 상수이다. section_length는 이 section_length에 이어지는 비트에서 CR 섹션의 최종 바이트까지의 길이를 바이트로 나타내는 필드이다. version_number는 조건부 액세스 테이블(CAT)의 버전 수를 나타내기 위한 데이터이다. version_number는 조건부 액세스 테이블의 설정이 변할 때 1 만큼 증가된다. current_next_indicator는 전송된 조건부 액세스 테이블이 현재 이용가능한가의 여부를 나타내기 위한 데이터이다. 조건부 액세스 테이블이 현재 이용가능하면, current_next_indicator는 "1"로 설정되고, 조건부 액세스 테이블이 현재 이용가능하지 않으면, current_next_indicator는 "0"으로 설정된다.

section_number는 조건부 액세스 테이블의 섹션 번호를 나타내기 위한 번호이다. 예를 들어, 조건부 액세스 테이블에 포함된 CA_section() 필드는 이것이 제 1 섹션임을 나타내도록 "0x00"으로 설정된다. section_number는 조건부 액세스 테이블에 포함된 섹션이 증가될 때마다 1 만큼 증가된다.

last_section_number는 조건부 액세스 테이블에 지정된 모든 섹션 중에서 최종 섹션의 섹션 번호를 나타내기 위한 데이터이다.

상술된 바와 같이, descriptor()는 프로그램 및 프로그램 요소의 정의를 확장하는데 사용되는 데이터 구조이다. 본 발명에 따른 데이터 전송 장치에서, 조건부 액세스 테이블에서 사용되는 descriptor()의 데이터 구조는 가입 정보 EMM를 포함하는 전송 스트림 패킷의 PID를 지정하는 정보를 설명하도록 의도됨을 이해하여야 한다.

다음에는 도 19를 참조하여 프로그램 맵 테이블과 조건부 액세스 테이블에서 사용되는 조건부 액세스 디스크립터 CA_descriptor()가 이후 설명된다.

조건부 액세스 디스크립터는 암호화된 ECM 데이터를 해독하도록 작업키(Kw) 및 가입에 대한 개인 정보를 포함하는 가입 정보 EMM을 지정할 뿐만 아니라 비디오 및 오디오와 같은 기본 스트림을 디스크램블링하도록 스크램블 키(Ks)를 포함하는 프로그램 해독 정보 ECM을 지정하는데 사용된다. 그러므로, 비디오 및 오디오와 같은 기본 스트림이 스크램블링되면, 조건부 액세스 디스크립터는 항상 비트 스트림에 존재한다.

조건부 액세스 디스크립터 CA_descriptor()는 다음의 데이터로 구성된다: descriptor_tag, descriptor_length, CA_system_ID, CA_PID, 및 private_data_byte. descriptor_tag는 상술된 바와 같이 복수의 디스크립터 각각을 지정하기 위한 고유의 식별 태그이다. 조건부 액세스 디스크립터 CA_descriptor()는 그에 지정된 "9"의 descriptor_tag를 갖는다. descriptor_length는 descriptor_length에 바로 이어지는 디스크립터의 데이터 바이트 수를 지정하기 위한 데이터이다. CA_system_ID는 연관된 ECM이나 EMM 데이터를 발생하여 적용시키는 조건부 액세스 시스템 (CA 시스템)의 종류를 나타내기 위한 데이터이다. CA_PID는 CA_system_ID에 의해 지정된 조건부 액세스 시스템 (CA 시스템)에서 ECM 또는 EMM 데이터를 포함하는 전송 스트림의 PID를 나타낸다.

조건부 액세스 디스크립터 CA_descriptor()의 CA_PID에 의해 지정된 패킷에서 설명되는 데이터 내용은 CA_descriptor()의 내용에 따라 변할 수 있다. 도 20을 참조하여, CA_PID에 의해 지정된 패킷이 이후 설명된다.

먼저, CA_PID에 의해 지정된 패킷에서 설명되는 데이터 내용은 CA_descriptor()의 사용, 즉 CA_descriptor()가 조건부 액세스 테이블(CAT)에서 또는 프로그램 맵 테이블에서 사용되는가의 여부에 따라 변할 수 있다. 특별히, CA_descriptor()가 조건부 액세스 테이블에 존재하면, CA_PID는 EMM 데이터를 포함하는 전송 스트림 패킷을 나타낸다. CA_descriptor()가 프로그램 맵 테이블에 존재하면, CA_PID는 ECM 데이터를 포함하는 전송 스트림 패킷을 나타낸다.

부가하여, CA_PID에 의해 지정된 패킷에서 설명되는 데이터 내용은 또한 조건부 액세스 디스크립터 CA_descriptor()의 프로그램 맵 테이블에서 문맥에 따라 변할 수 있다. 이에 대해, 조건부 액세스 디스크립터 CA_descriptor()의 프로그램 맵 테이블에서의 문맥은 CA_descriptor()가 제 1 FOR문 (도 15) 또는 제 2 FOR문 (도 15)에서 사용되는가의 여부를 의미한다.

조건부 액세스 디스크립터 CA_descriptor()가 제 1 FOR문에서 사용되면, CA_PID는 ECM 데이터를 포함하는 전송 스트림 패킷의 PID를 나타낸다. 이 경우의 제 1 CA_descriptor()는 program_number에 의해 지정된 프로그램을 포함하는 전송 스트림 패킷과 ECM 데이터를 포함하는 전송 스트림 패킷 사이의 대응관계를 설명하도록 구문 (syntax)에 따라 사용된다. 즉, 동일한 ECM 데이터는 program_number에 의해 지정된 프로그램에 포함되는 모든 데이터 요소에 지정된다. 다른 말로 하면, 프로그램에서 모든 데이터 요소는 동일한 스크램블 키로 스크램블링 및 디스크램블링된다.

반대로, 조건부 액세스 디스크립터 CA_descriptor()가 제 2 CA_descriptor()로 제 2 FOR문에서 사용되면, CA_PID는 또한 ECM 데이터를 포함하는 전송 스트림 패킷의 PID를 나타낸다. 그러나, 제 2 CA_descriptor()는 program_number에 의해 지정된 프로그램을 포함하는 전송 스트림 패킷과 ECM 데이터를 포함하는 전송 스트림 패킷 사이의 대응관계를 설명하도록 의도되지 않지만, 이 CA_descriptor()는 elementary_PID에 의해 지정된 데이터 요소를 포함하는 전송 스트림 패킷과 ECM 데이터를 포함하는 전송 스트림 패킷 사이의 대응관계를 설명한다. 즉, 다른 ECM 데이터는 elementary_PID에 의해 지정된 각 데이터 요소에 지정된다. 다른 말로 하면, 프로그램내의 다른 데이터 요소는 다른 스크램블 키로 스크램블링 및 디스크램블링된다. 프로그램이 한 채널의 비디오 데이터와 두 채널의 오디오 데이터, 즉 메인 오디오 및 서브오디오 데이터로 구성된다고 가정할 때, 이러한 방법으로 각 데이터 요소에 대해 다른 스크램블 키를 사용하는 것은 시스템이 다른 채널과 다른 스크램블 키로 서브오디오 데이터를 스크램블링하도록 허용한다.

다음에는 EMM 데이터와 ECM 데이터가 이후 설명된다.

EMM (Entitlement Management Message) 데이터는 스크램블 키(Ks)를 암호화하는 데 사용되는 작업키(Kw)와 가입자에 의해 가입된 프로그램을 나타내는 가입 정보를 포함하는 데이터이다. EMM 데이터는 프로그램 연관 테이블(PAT), 프로그램 맵 테이블(PMT), 및 조건부 액세스 테이블(CAT)과 같이 전송 스트림 패킷의 페이로드에서 전송된다. 다음의 설명에서, EMM 데이터를 포함하는 전송 스트림 패킷은 EMM 패킷이라 칭하여진다.

도 21에 도시된 바와 같이, EMM 패킷은 4 바이트의 헤더와 183 바이트의 페이로드 섹션으로 구성된다. 상술된 전송 스트림 패킷과 같이, EMM 패킷의 헤더는 다음의 데이터 필드로 구성된다: sync_byte, transport_error_indicator, payload_unit_start_indicator, transport_priority, PID, transport_scrambling_control, adaption_field_control, 및 continuity_counter. 183 바이트의 페이로드 섹션은 EMM 섹션 헤더, EMM 데이터, CRC, 및 여유 바이트로 구성된다.

EMM 섹션 헤더는 다음의 데이터 필드로 구성된다: table_id, section_syntax_indicator, reserved_0, reserved_1, section_length, table_id_extention, reserved_2, version_number, current_next_indicator, section_number, 및 last_section_number. table_id는 각 테이블 (각 섹션)에 지정된 고유의 식별 번호이다. EMM 섹션은 식별 번호로 지정된 사용자-정의가능 table_id "0x40" 내지 "0xFE"를 갖는다. section_syntax_indicator는 "1"로 고정된 상수이다. section_

length는 이 section_length에 이어지는 비트에서 CRC 섹션의 최종 바이트까지의 길이를 바이트로 나타내기 위한 데이터 필드이다. table_id_extention은 EMM 섹션의 확장 데이터가 있는가의 여부를 나타내기 위한 데이터이다. version_number는 EMM 섹션의 버전 수를 나타내기 위한 데이터이다. 버전 수는 EMM 섹션의 매개변수가 전송 스트림에서 변할 때 증가된다. current_next_indicator는 전송된 EMM 섹션이 현재 이용가능한가의 여부를 나타내기 위한 데이터이다. section_number는 EMM 섹션의 번호를 나타내기 위한 데이터로서, 항상 "1"이다. last_section_number는 EMM 섹션에서 최종 섹션의 섹션 번호를 나타내기 위한 데이터로서, 항상 "1"이다.

EMM 데이터는 다음의 데이터 필드로 구성된다: card_ID, EMM_type, CA_system_ID, Kw_No, Kw, authorize_type, service_ID, series_ID, event_ID, 및 component_map.

card_ID는 IRD에 설치된 고유의 IC 카드에 주어지는 식별 번호이고, 이는 항상 가입자의 관리를 위해 식별 번호로 사용된다. EMM_type은 EMM 데이터의 종류를 나타내는 데이터이다. CA_system_ID는 가입자 관리 시스템을 포함하는 CA 시스템에 주어진 식별 번호이다. Kw는 스크램블 키(Ks)를 암호화하는데 사용되는 작업키를 나타내고, Kw_No는 256개의 소정의 작업키 중에서 어느 작업키가 스크램블 키(Ks)를 암호화하는데 사용되었나를 나타내기 위한 번호이다. authorize_type, service_ID, series_ID, 및 event_ID는 가입자에 의해 어느 프로그램이 가입되었나를 나타내기 위한 가입 조건이다. component_map은 가입된 프로그램 중 어느 요소가 가입되었나를 나타내기 위한 데이터이다.

ECM (Entitlement_Control_Message) 데이터는 가입자가 가입한 스크램블링된 프로그램을 디스크램블링하는데 사용되는 스크램블 키(Ks)를 포함하는 데이터이다. ECM 데이터는 프로그램 연관 테이블(PAT), 프로그램 맵 테이블(PMT), 및 조건부 액세스 테이블(CAT)와 같이 전송 스트림 패킷의 페이로드에서 전송된다. 다음의 설명에서, ECM 데이터를 포함하는 전송 스트림 패킷은 ECM 패킷이라 칭하여진다. ECM 패킷은 대략 100 msec의 간격으로 전송되고, EMM 데이터로 전송되는 스크램블 키는 대략 4 sec의 간격으로 업데이트된다.

도 22에 도시된 바와 같이, ECM 패킷은 4 바이트의 헤더와 183 바이트의 페이로드 섹션으로 구성된다. 상술된 전송 스트림 패킷과 같이, ECM 패킷의 헤더는 다음의 데이터 필드로 구성된다: sync_byte, transport_error_indicator, payload_unit_start_indicator, transport_priority, PID, transport_scrambling_control, adaptation_field_control, 및 continuity_counter. 183 바이트의 페이로드 섹션은 ECM 섹션 헤더, ECM 데이터, CRC, 및 여유 바이트로 구성된다.

ECM 섹션 헤더는 다음의 데이터 필드로 구성된다: table_id, section_syntax_indicator, reserved_0, reserved_1, section_length, table_id_extention, reserved_2, version_number, current_next_indicator, section_number, 및 last_section_number. table_id는 각 테이블 (각 섹션)에 지정된 고유의 식별 번호이다. ECM 섹션은 식별 번호로 지정된 사용자-정의가능 table_id "0x40" 내지 "0xFE"를 갖는다. section_syntax_indicator는 "1"로 고정된 상수이다. section_length는 이 section_length에 이어지는 비트로부터 CRC 섹션의 최종 바이트까지의 길이를 바이트로 나타내기 위한 데이터 필드이다. table_id_extention은 ECM 섹션의 확장 데이터가 있는가의 여부를 나타내기 위한 데이터이다. version_number은 ECM 섹션의 버전 수를 나타내기 위한 데이터이다. 버전 수는 ECM 섹션의 매개변수가 전송 스트림에서 변할 때 증가된다. current_next_indicator는 전송된 ECM 섹션이 현재 이용가능한가의 여부를 나타내기 위한 데이터이다. section_number는 ECM 섹션의 번호를 나타내기 위한 데이터이고, 항상 "1"이다. last_section_number는 ECM 섹션에서 최종 섹션의 섹션 번호를 나타내기 위한 데이터이고, 항상 "1"이다.

ECM 데이터는 다음의 데이터 필드로 구성된다: ECM_type, CA_system_ID, Kw_No, service_mode, service_ID, series_ID, event_ID, component_map, Ks_Odd, 및 Ks_Even.

ECM_type은 ECM 데이터의 종류를 나타내기 위한 데이터이다. CA_system_ID는 가입자 관리 시스템을 포함하는 CA 시스템에 주어진 식별 번호이다. Kw_No는 256개의 소정의 작업키 중 어느 작업키가 스크램블 키(Ks)를 암호화하는데 사용되었나를 나타내기 위한 번호이다. service_mode, service_ID, series_ID, 및 event_ID는 가입자가 가입한 프로그램을 나타내기 위한 가입 조건이다. component_map은 가입된 프로그램 중 어느 요소가 가입되었나를 나타내기 위한 데이터이다. Ks_Odd 및 Ks_Even은 전송된 비디오 및 오디오 데이터를 스크램블링하는데 사용되는 스크램블 키이다. 스크램블 키는 홀수 키 Ks_Odd와 짝수 키 Ks_Even으로 구성되고, 홀수 키와 짝수 키는 4 sec의 간격으로 스크램블 키로서 번갈아 사용된다.

다음에는 도 23을 참조하여 수신기로서 제공되는 IRD (Integrated_Receiver_Decoder)(20)가 이후 상세히 설명된다.

IRD는 위성을 통해 전송된 변조 스트림을 복조하기 위한 복조 회로(21), 복조 회로(21)에 의해 복조된 스트림을 패킷형으로 나누기 위한 디멀티플렉서(22), 디멀티플렉서(22)에 의해 나뉜 PAT 패킷, PMT 패킷, CAT 패킷, EMM 패킷, 및 ECM

패킷을 수신하기 위한 CPU(23), 암호화된 작업키(Kw')와 암호화된 스크램블 키(Ks')를 해독하기 위한 보안 모듈(24), 스크램블링된 비디오 스트림, 오디오 스트림, 및 전용 데이터 스트림을 디스크램블링하기 위한 디스크램블러 (descrambler) (25V, 25A, 25P), 및 비디오 스트림과 오디오 스트림을 복호화하기 위한 디코더 (26V, 26A)를 구비한다.

디멀티플렉서(22)는 CPU(23)로부터 제어 명령을 수신하고 그 명령에 응답해 적절한 타이밍으로 CPU(23)에 적절한 전송 스트림 패킷을 공급하도록 동작된다.

CPU(23)는 디멀티플렉서(22)에 의해 공급된 PAT 패킷을 분석하기 위한 PAT 패킷 분석기(231), 디멀티플렉서(22)에 의해 공급된 PMT 패킷을 분석하기 위한 PMT 패킷 분석기(232), 디멀티플렉서(22)에 의해 공급된 CAT 패킷을 분석하기 위한 CAT 패킷 분석기(233), 디멀티플렉서(22)에 의해 공급된 EMM' 패킷을 분석하기 위한 EMM 패킷 분석기(234), 및 디멀티플렉서(22)에 의해 공급된 ECM' 패킷을 분석하기 위한 ECM 패킷 분석기(235)를 구비한다.

PAT 분석기(231)는 PAT 패킷에 포함된 program_number와 program_map_PID를 구하도록 디멀티플렉서(22)에 의해 공급된 전송 스트림 패킷을 PAT 패킷으로 분석한다. PAT 분석기(231)는 또한 가입자에 의해 어느 프로그램이 가입되었나를 나타내는 authorize_type을 보안 모듈(24)로부터 수신한다. 이어서, PAT 분석기(231)는 보안 모듈에 의해 공급된 authorize_type을 디멀티플렉서(22)에 의해 공급된 program_number와 비교하고, 가입자에 의해 가입된 프로그램과 일치하는 program_number에 대응하는 program_map_PID만을 선택한다. 그래서, PAT 분석기(231)는 그 프로그램에 대응하는 프로그램 맵 테이블의 PID만을 구할 수 있다.

이어서, 디멀티플렉서(22)는 PAT 분석기(231)에 의해 공급되는 program_map_PID에 의해 지정된 PID를 갖는 전송 스트림 패킷을 선택하고 선택된 전송 스트림 패킷을 PMT 분석기(232)에 PMT 패킷으로 제공한다.

PMT 분석기(232)는 program_map_PID에 의해 지정된 PID를 갖는 PMT 패킷에 포함되는 PMT 데이터를 분석한다. 특별히, PMT 분석기(232)는 PMT 패킷에서 PMT 데이터로 설명되는 elementary_PID로부터 program_number로 지정된 프로그램을 구성하는 데이터 요소를 포함한 전송 스트림 패킷의 PID를 구한다. 예를 들어 프로그램이 두 데이터 요소, 즉 비디오 데이터와 오디오 데이터로 구성된다고 가정하면, 제 1 elementary_PID는 비디오 스트림을 포함하는 전송 스트림 패킷을 지정하게 되고, 제 2 elementary_PID는 오디오 스트림을 포함한 또 다른 전송 스트림 패킷을 지정하게 된다.

부가하여, PMT 분석기(232)는 PMT 데이터에서 설명된 CA_descriptor를 descriptor() 함수로 참고한다. 이어서, PMT 분석기(232)는 CA_descriptor에서 설명된 CA_PID를 얻고, 이 CA_PID를 디멀티플렉서(22)에 피드백시킨다. CA_descriptor에서 설명된 CA_PID는 ECM 데이터를 포함하는 전송 스트림 패킷의 PID를 나타내기 위한 데이터이다.

이어서, 디멀티플렉서(22)는 PMT 분석기(232)에 의해 공급된 elementary_PID로 지정되는 PID를 갖는 전송 스트림 패킷을 적절한 처리 회로에 제공한다. 예를 들어, 데이터 요소가 비디오 스트림이면, 이는 비디오 스트림을 디스크램블링하도록 디스크램블러(25V)에 제공되고; 데이터 요소가 오디오 스트림이면, 이는 오디오 스트림을 디스크램블링하도록 디스크램블러(25A)에 제공하고; 또한 데이터 요소가 전용 데이터 스트림이면, 이는 전용 데이터 스트림을 디스크램블링하도록 디스크램블러(25P)에 제공된다.

부가하여, 디멀티플렉서(22)는 CA_PID에 의해 지정된 PID를 갖는 전송 스트림 패킷을 ECM 분석기(235)에 ECM 패킷으로 공급한다. 그래서, ECM 분석기(235)에 공급된 ECM 데이터는 가입자에 의해 가입된 프로그램에 관련되는 ECM 데이터만을 포함한다.

ECM 분석기(235)는 먼저 디멀티플렉서(22)로부터 수신된 ECM' 패킷을 필터링한다. 특별히, ECM 분석기(235)는 ECM 데이터내의 CA_system_ID를 보안 모듈(24)에 의해 공급된 CA_system_ID와 비교하고, 보안 모듈(24)에 의해 공급된 CA_system_ID를 정합시키는 CA_system_ID를 포함하는 ECM 패킷만을 선택한다. 이어서, ECM 분석기(235)는 선택된 ECM 패킷에 포함된 ECM 데이터를 분석함으로써 암호화된 스크램블 키(Ks')를 구할 수 있다. ECM 분석기(235)는 보안 모듈(24)의 마이크로프로세서(MPU)에서 암호화된 스크램블 키(Ks')를 해독 회로(242)에 공급한다.

디멀티플렉서(22)는 전송된 비트 스트림으로부터 CAT 패킷의 PID를 갖는 전송 스트림 패킷을 검출하고, 검출된 전송 스트림 패킷을 CAT 분석기(233)에 CAT 패킷으로 제공한다.

CAT 분석기(233)는 디멀티플렉서(22)로부터 수신된 CAT 패킷에 포함되는 CA_descriptor() 기능을 검출하고, CA_descriptor() 기능의 CA_PID로부터 CAT 패킷의 CA_descriptor()에 의해 지정된 전송 스트림 패킷의 PID를 구한다. CAT 패킷의 CA_descriptor()에 의해 지정된 전송 스트림 패킷은 EMM 정보를 포함하는 전송 스트림 패킷이다.

디멀티플렉서(22)는 전송된 비트 스트림으로부터 CAT 분석기(233)에서 수신된 CA_PID에 의해 지정되는 PID를 갖는 전송 스트림 패킷을 선택하고, 선택된 전송 스트림 패킷을 EMM 분석기(234)에 EMM' 패킷으로 공급한다.

EMM 분석기(234)는 먼저 디멀티플렉서(22)로부터 수신된 EMM' 패킷을 필터링하고, 보안 모듈에 대응하는 EMM' 패킷만을 선택한다. 특별히, 디멀티플렉서(22)로부터 수신된 EMM 패킷에 대해 CA_descriptor()의 CA_PID에 의해 지정되는 전송 스트림 패킷은 EMM 데이터를 포함하는 EMM' 패킷이기 때문에, EMM' 패킷에 포함된 EMM 데이터는 CA_PID를 참조하여 구해질 수 있다. EMM 분석기(234)는 EMM 데이터내의 Card_ID 및 CA_system_ID를 보안 모듈(24)의 메모리(2410에 의해 공급된 Card_ID 및 CA_system_ID와 비교하고, 각각 보안 모듈(24)에 의해 공급된 Card_ID 및 CA_system_ID를 정합시키는 Card_ID 및 CA_system_ID를 포함하는 EMM' 패킷만을 선택한다.

이어서, EMM 분석기(234)는 선택된 EMM 패킷에 포함된 101 바이트의 EMM 데이터를 보안 모듈(24)의 메모리(241)에 최종 EMM 데이터로 공급하여, 메모리(241)에서 오래된 EMM 데이터를 교체한다. 부가하여, EMM 분석기(234)는 보안 모듈(24)의 마이크로프로세서(MPU)에서 EMM 데이터에 포함되는 암호화된 작업키(Kw')를 해독 회로(242)에 공급한다.

보안 모듈(24)은 메모리(241)와, 제 1 해독 회로(242) 및 제 2 해독 회로(243)를 포함하는 마이크로프로세서로 구성된다. 보안 모듈(24)은 예를 들면, IRD로부터 제거가능한 IC 카드로 구성된다.

보안 모듈(24)의 제 1 해독 회로(242)는 EMM 분석기(234)로부터 암호화된 작업키(Kw')를 수신하여 암호화된 작업키(Kw')를 소정의 마스터 키(Km)로 해독한다. 이어서, 제 1 해독 회로(242)은 해독된 작업키(Kw)를 제 2 해독 회로(243)에 공급한다.

보안 모듈(24)의 제 2 해독 회로(243)는 ECM 회로(235)로부터 암호화된 스크램블 키(Ks')를 수신할 뿐만 아니라 제 1 해독 회로(242)로부터 해독된 작업키(Kw)를 수신하여 암호화된 스크램블 키(Ks')를 해독된 작업키(Kw)로 해독한다. 해독된 스크램블 키(Ks)는 디스크램블러 (25V, 25A, 25P)로 공급된다. 상기에서는 동일한 스크램블 키(Ks)가 디스크램블러 (25V, 25A, 25P)에 공급되지만, 다른 스크램블 키(Ks)가 각 데이터 요소에 정의되면, 다른 스크램블 키(Ks)가 다른 디스크램블러에 공급될 수 있음을 이해하여야 한다.

상기로부터, 복수의 데이터 요소들로 구성된 프로그램 데이터의 전송 스트림 패킷이 다중화되어 전송되는 구성의 데이터 다중화 장치에서는 프로그램을 구성하는 복수의 데이터 요소들 중에서 하나 또는 그 이상의 데이터 요소에 대응하는 스크램블 키가 발생되고 이어서 각 데이터 요소가 스크램블링되어, 각 데이터 요소는 가입자에 의해 가입될 수 있다.

다중화된 전송 스트림 패킷 각각은 대응하는 스크램블 키로 스크램블링되므로, 스크램블링을 위한 회로 구성은 다중화 이전에 각 전송 스트림 패킷을 스크램블링하는 것 보다 더 간단해질 수 있다.

우선순위가 높은 데이터 요소를 버퍼링하기 위한 버퍼 메모리로부터의 오버플로우는 각 버퍼 메모리에 공급된 데이터 요소에 대한 입력 레이트가 기준 레이트 보다 높을 때, 복수의 버퍼 메모리 중에서 우선순위가 낮은 정보를 버퍼링하기 위한 버퍼 메모리를 제외하도록 스위치 수단을 스위칭가능하게 제어함으로서 방지될 수 있다.

수신단에서, 다른 데이터 요소는 그 스크램블 키에 대응하는 각 데이터 요소의 전송 스트림 패킷을 스크램블 키로 디스크램블링함으로서 따로 디스크램블링될 수 있다.

상술된 구성에 따라, 가입자가 원하는 데이터 요소만을 보거나 들을 수 있게 허용하는 데이터 다중화 장치는 더 간단한 실행으로 이루어질 수 있다.

더욱이, 버퍼 메모리로부터의 오버플로우는 우선순위가 낮은 데이터 요소를 선택에서 제외하고 원하는 데이터 요소를 다른 것에 우선하여 다중화함으로서 효과적으로 방지될 수 있다.

산업상 이용 가능성

본 발명은 MPEG2에 따라 비디오 데이터 및 오디오 데이터의 압축 부호화를 실행하고 지상파나 위성파를 통해 부호화된 스트림을 전송하는 디지털 방송 시스템에 적용될 수 있다.

(57) 청구의 범위

청구항 1.
삭제

청구항 2.
삭제

청구항 3.
삭제

청구항 4.
삭제

청구항 5.
삭제

청구항 6.
삭제

청구항 7.
삭제

청구항 8.
삭제

청구항 9.
삭제

청구항 10.
삭제

청구항 11.
삭제

청구항 12.

복수의 데이터 요소들로 구성된 프로그램을 배포하는 프로그램 배포 시스템에 있어서:

각 프로그램이나 데이터 요소에 대한 가입자들의 가입들을 관리하는 가입자 관리 시스템;

상기 데이터 요소들 각각에 대해 상기 프로그램에 포함된 상기 데이터 요소들을 디스크램블링하는 데 사용될 스크램블 키를 발생하는 가입자 시청허가 시스템; 및

멀티플렉서 시스템으로서,

각 프로그램에 대해 부호화된 데이터 요소들로 구성된 부호화된 스트림들을 발생하도록 상기 프로그램에 포함된 상기 데이터 요소들의 각각을 부호화하는 부호화 시스템,

상기 부호화 시스템에 의해 각 프로그램에 대해 발생되는 상기 부호화된 스트림들을 다중화하는 다중화 수단, 및

상기 가입자 시청허가 시스템에 의해 발생된 상기 스크램블 키에 기초하여 상기 다중화된 스트림에 포함되는 상기 부호화된 데이터 요소들의 각각을 선택적으로 스크램블링하는 스크램블 수단을 포함하는, 상기 멀티플렉서 시스템을 포함하고,

상기 가입자 관리 시스템은 상기 스크램블 키를 암호화하기 위한 작업키를 발생하고,

상기 가입자 시청허가 시스템은 EMM 데이터로서 공급된 상기 작업키를 마스터 키로 암호화하여 암호화된 작업키를 출력으로서 제공하는 제 1 암호화 수단을 포함하고,

상기 멀티플렉서 시스템은 ECM 데이터에 포함된 상기 스크램블 키를 상기 작업키로 암호화하여 암호화된 스크램블 키를 출력으로서 제공하는 제 2 암호화 수단을 포함하고,

상기 스크램블 키를 암호화하기 위해 상기 제 2 암호화 수단에 의해 사용되는 상기 작업키는 상기 EMM 데이터에 포함된 상기 암호화된 작업키가 아니고, 상기 가입자 시청허가 시스템에 의해 공급된 작업키 테이블로부터 얻어진 암호화되지 않은 작업키인, 프로그램 배포 시스템.

청구항 13.
삭제

청구항 14.
삭제

청구항 15.
삭제

청구항 16.
삭제

청구항 17.
삭제

청구항 18.
삭제

청구항 19.
삭제

청구항 20.

제 12 항에 있어서,

상기 제 2 암호화 수단은 상기 가입자 시청허가 시스템에 의해 공급된 상기 작업키 테이블을 참조하여 상기 ECM 데이터에 포함된 상기 작업키 식별 번호로부터 상기 작업키를 얻고,

상기 제 2 암호화 수단은 상기 제 2 암호화 수단에 의해 암호화된 상기 암호화 스크램블 키를 암호화된 ECM 데이터로서 제공하는 프로그램 배포 시스템.

청구항 21.

제 20 항에 있어서,

상기 부호화 시스템에 의해 제공되는 상기 부호화된 스트림과 상기 가입자 시청허가 시스템에 의해 제공되는 상기 암호화된 EMM 데이터 및 ECM 데이터는 전송 스트림 패킷의 형태로 제공되고, 상기 전송 스트림 패킷 각각에는 상기 전송 스트림 패킷을 식별하기 위한 패킷 ID가 주어지는 프로그램 배포 시스템.

청구항 22.

제 21 항에 있어서,

상기 제 2 암호화 수단은 암호화된 ECM 데이터를 출력으로서 상기 다중화 수단에 제공하는 프로그램 배포 시스템.

청구항 23.

제 22 항에 있어서,

상기 스크램블 수단은 상기 프로그램을 구성하는 상기 복수의 데이터 요소들만을 스크램블링하는 프로그램 배포 시스템.

청구항 24.

제 22 항에 있어서,

상기 스크램블 수단은 상기 데이터 요소 각각을 포함하는 전송 스트림 패킷의 패킷 ID와 상기 데이터 요소에 대해 정의된 스크램블 키 사이의 대응관계를 나타내는 테이블에 기초하여 상기 데이터 요소들과 연관된 스크램블 키들을 사용해 상기 데이터 요소들을 스크램블링하는 프로그램 배포 시스템.

청구항 25.

제 22 항에 있어서,

상기 스크램블 수단은 상기 다중화 수단에 의해 상기 스크램블 수단에 공급된 모든 전송 스트림 패킷의 패킷 ID를 검출하고,

상기 스크램블 수단은 상기 패킷 ID들과 상기 스크램블들 키들 사이의 대응관계를 나타내는 테이블에 기초하여 스크램블 키들이 상기 검출된 패킷 ID에 대해 정의되는가의 여부를 결정하고,

일부 스크램블 키들이 상기 패킷 ID들에 대해 정의되면, 상기 스크램블 수단은 상기 패킷 ID들에 의해 나타내어지는 전송 스트림 패킷에 포함된 데이터 요소들을 상기 정의된 스크램블 키들로 스크램블링하고,

스크램블 키들이 상기 패킷 ID들에 대해 정의되어 있지 않으면, 상기 스크램블 수단은 상기 패킷 ID들에 의해 나타내어지는 전송 스트림 패킷들에 포함된 데이터를 스크램블링하지 않는 프로그램 배포 시스템.

청구항 26.

제 20 항에 있어서,

상기 멀티플렉서 시스템은 전송 스트림 패킷의 형태로 상기 멀티플렉서 시스템에 공급되는 데이터를 버퍼링하고 상기 전송 스트림 패킷들을 상기 다중화 수단에 제공하는 버퍼 수단을 더 포함하는 프로그램 배포 시스템.

청구항 27.

제 26 항에 있어서,

상기 멀티플렉서 시스템은 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷을 버퍼링하기 위한 복수의 버퍼들의 자유 영역을 모니터하고,

상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들을 버퍼링하기 위한 상기 복수의 버퍼들 중 어느 하나가 오버플로우 (overflow)되려고 하면, 상기 EMM 데이터를 포함하는 상기 전송 스트림 패킷은 상기 EMM 데이터를 포함하는 상기 전송 스트림 패킷을 버퍼링하기 위한 버퍼에 의해 상기 다중화 수단에 제공되지 않고, 그 대신 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들은 오버플로우될 것 같은 상기 버퍼에 의해 상기 다중화 수단에 제공되는 프로그램 배포 시스템.

청구항 28.

제 20 항에 있어서,

상기 멀티플렉서 시스템에 의해 제공되는 전송 스트림들을 전송선을 통해 수신단으로 배포하는 배포 시스템, 및

상기 전송선을 통해 전송된 상기 전송 스트림들을 수신하는 수신 시스템을 더 포함하는 프로그램 배포 시스템.

청구항 29.

제 28 항에 있어서,

상기 수신 시스템은:

상기 전송된 전송 스트림들을 역다중화하는 디멀티플렉서;

상기 스크램블링된 데이터 요소를 각각 상기 공급된 스크램블 키들로 디스크램블링하는 디스크램블러 (descrambler);

각 데이터 요소에 대해 상기 디스크램블링된 데이터를 복호화하는 디코더;

상기 전송 스트림을 구성하는 전송 스트림 패킷을 분석하는 CPU; 및

상기 전송 스트림에 포함된 상기 암호화된 스크램블 키를 해독하고 상기 해독된 스크램블 키를 상기 디스크램블러에 공급하는 보안 모듈을 포함하는 프로그램 배포 시스템.

청구항 30.

제 29 항에 있어서,

상기 보안 모듈은:

상기 전송된 전송 스트림에 포함된 상기 암호화 EMM 데이터에 포함되는 가입자의 가입 정보를 저장하는 메모리 수단;

상기 전송된 전송 스트림에 포함된 상기 암호화된 작업키 및 상기 가입자 관리 시스템에 의해 사용된 것과 같은 마스터 키를 수신하여 상기 암호화된 작업키를 상기 마스터 키로 해독하는 제 1 해독 수단; 및

상기 전송 스트림에 포함된 상기 암호화 스크램블 키 및 상기 제 1 해독 수단에 의해 공급된 상기 해독 작업키를 수신하여 상기 암호화된 스크램블 키를 상기 해독된 작업키로 해독하는 제 2 해독 수단을 포함하는 프로그램 배포 시스템.

청구항 31.

제 30 항에 있어서,

상기 CPU는 상기 디멀티플렉서에 의해 공급된 상기 암호화 ECM 데이터를 포함하는 상기 전송 스트림 패킷으로부터, 가입자에 의해 가입된 프로그램이나 데이터 요소에 대한 암호화된 ECM 데이터를 갖는 전송 스트림 패킷만을 필터링하고,

상기 CPU는 상기 필터링된 전송 스트림 패킷에 포함된 상기 암호화 ECM 데이터를 분석함으로써 상기 암호화 ECM 데이터로부터 상기 암호화된 스크램블 키를 얻는 프로그램 배포 시스템.

청구항 32.

제 31 항에 있어서,

상기 프로그램과 연관된 암호화 스크램블 키가 상기 CPU에 의해 공급되면, 상기 보안 모듈은 상기 공급된 암호화 스크램블 키를 해독하여 상기 프로그램을 구성하는 복수의 데이터 요소들에 대응하는 복수의 디스크램블러들에 동일한 스크램블 키를 각각 공급하고,

상기 복수의 데이터 요소들과 연관된 복수의 암호화 스크램블 키들이 상기 CPU에 의해 공급되면, 상기 보안 모듈은 상기 복수의 공급된 암호화 스크램블 키들을 각각 해독하여 상기 복수의 데이터 요소들 중에서 가입된 데이터 요소들에 대응하는 복수의 디스크램블러들에 상이한 스크램블 키들을 공급하는 프로그램 배포 시스템.

청구항 33.

복수의 데이터 요소들로 구성된 프로그램을 배포하는 프로그램 배포 시스템에 있어서:

각 프로그램 또는 데이터 요소에 대한 가입자의 가입들을 관리하는 가입자 관리 시스템;

상기 데이터 요소들 각각에 대한 상기 프로그램에 포함된 상기 데이터 요소들을 디스크램블링하는 데 이용될 스크램블 키를 발생하는 가입자 시청허가 시스템; 및

멀티플렉서 시스템으로서,

각 프로그램에 대해 부호화된 데이터 요소들로 구성되는 부호화 스트림들을 발생하도록 상기 프로그램에 포함된 상기 데이터 요소들의 각각을 부호화하는 부호화 시스템,

상기 부호화 시스템에 의해 각 프로그램에 대해 발생된 상기 부호화 스트림들을 다중화하는 다중화 수단, 및

상기 가입자 시청허가 시스템에 의해 발생된 상기 스크램블 키에 기초하여 상기 다중화된 스트림에 포함된 상기 부호화 데이터 요소들의 각각을 선택적으로 스크램블링하는 스크램블 수단을 포함하는, 상기 멀티플렉서 시스템을 포함하고;

상기 가입자 시청허가 시스템은 상기 스크램블 키를 암호화하는데 사용된 작업키를 마스터 키로 암호화하는 제 1 암호화 수단을 포함하고,

상기 가입자 시청허가 시스템은 상기 제 1 암호화 수단에 의해 암호화된 상기 암호화 작업키와 상기 가입자를 식별하기 위한 가입자 식별 번호를 암호화된 EMM 데이터로서 상기 멀티플렉서 시스템에 공급하고,

상기 멀티플렉서 시스템은 상기 다중화 수단 앞에, 상기 ECM 데이터에 포함된 스크램블 키를 암호화하는 제 2 암호화 수단을 더 포함하고,

상기 제 2 암호화 수단은 상기 작업키 테이블을 참조하여 상기 ECM 데이터에 포함된 상기 작업키 식별 번호로부터 작업키를 얻고,

상기 제 2 암호화 수단은 상기 작업키 테이블로부터 얻어진 상기 작업키를 사용해 상기 ECM 데이터에 포함된 상기 스크램블 키를 암호화하고,

상기 제 2 암호화 수단은 상기 제 2 암호화 수단에 의해 암호화된 상기 암호화 스크램블 키를 암호화된 ECM 데이터로서 상기 다중화 수단에 공급하는 프로그램 배포 시스템.

청구항 34.

삭제

청구항 35.

제 33 항에 있어서,

상기 데이터 배포 시스템에 의해 제공된 전송 스트림내에서 상기 프로그램을 구성하는 상기 복수의 데이터 요소들을 포함하는 전송 스트림 패킷, 상기 ECM 데이터를 포함하는 전송 스트림 패킷, 및 상기 EMM 데이터를 포함하는 전송 스트림 패킷의 패킷 ID를 식별하기 위한 프로그램 지정 정보를 발생하는 인코더/멀티플렉서 제어 시스템을 더 포함하고,

인코더/멀티플렉서 제어 시스템은 상기 프로그램 지정 정보에 따라 상기 프로그램을 구성하는 상기 복수의 데이터 요소들을 포함하는 상기 전송 스트림 패킷, 상기 ECM 데이터를 포함하는 상기 전송 스트림 패킷, 및 상기 EMM 데이터를 포함하는 상기 전송 스트림 패킷을 다중화하도록 상기 인코더 시스템과 상기 멀티플렉서 시스템을 제어하는 프로그램 배포 시스템.

청구항 36.

삭제

청구항 37.

삭제

청구항 38.

삭제

청구항 39.

삭제

청구항 40.

제 35 항에 있어서,

상기 인코더/멀티플렉서 제어 시스템은 전송 스트림 패킷의 형태로 상기 멀티플렉서 시스템에 공급된 모든 전송 스트림 패킷에 상기 전송 스트림 패킷을 식별하기 위한 패킷 ID를 할당하는 프로그램 배포 시스템.

청구항 41.

제 40 항에 있어서,

상기 프로그램 지정 정보는 적어도 프로그램 연관 테이블, 프로그램 맵 테이블, 및 조건부 액세스 테이블(conditional access table)로 구성되는 프로그램 배포 시스템.

청구항 42.

제 41 항에 있어서,

상기 인코더/멀티플렉서 제어 시스템은 PAT 패킷으로서 상기 프로그램 연관 테이블을 포함하는 전송 스트림 패킷을 상기 멀티플렉서 시스템에 공급하고,

상기 인코더/멀티플렉서 제어 시스템은 PMT 패킷으로서 상기 프로그램 맵 테이블을 포함하는 전송 스트림 패킷을 상기 멀티플렉서 시스템에 공급하고,

상기 인코더/멀티플렉서 제어 시스템은 상기 조건부 액세스 테이블을 포함하는 전송 스트림 패킷을 CAT 패킷으로서 상기 멀티플렉서 시스템에 공급하는 프로그램 배포 시스템.

청구항 43.

제 42 항에 있어서,

상기 프로그램 연관 테이블은 프로그램 번호와 상기 프로그램 번호에 대응하는 PMT 패킷의 패킷 ID를 지정하는 테이블이고,

상기 프로그램 맵 테이블은 프로그램을 구성하는 복수의 데이터 요소들 각각을 포함하는 전송 스트림 패킷의 패킷 ID를 지정하는 테이블이고,

상기 조건부 액세스 테이블은 암호화된 EMM 패킷의 상기 패킷 ID를 지정하는 테이블인 프로그램 배포 시스템.

청구항 44.

제 43 항에 있어서,

상기 프로그램 연관 테이블은 프로그램을 나타내는 프로그램 번호 및 상기 프로그램과 연관된 PMT 패킷의 패킷 ID를 기술하고,

상기 프로그램 맵 테이블은 상기 프로그램을 나타내는 프로그램 번호, 상기 프로그램을 구성하는 복수의 데이터 요소들을 포함하는 전송 스트림 패킷을 포함하는 복수의 패킷 ID들, 및 상기 프로그램이나 상기 데이터 요소와 연관된 암호화 ECM 패킷의 패킷 ID를 지정하는 디스크립터를 기술하는 프로그램 배포 시스템.

청구항 45.

제 44 항에 있어서,

상기 프로그램 맵 테이블내의 상기 디스크립터가 상기 프로그램 번호에 대응하는 위치에 기술되어 있으면, 상기 디스크립터는 상기 프로그램을 구성하는 상기 복수의 데이터 요소들의 모든 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 패킷의 패킷 ID를 지정하고,

상기 프로그램 맵 테이블내의 상기 디스크립터가 상기 프로그램의 상기 데이터 요소 각각에 대응하는 위치에 기술되어 있으면, 상기 디스크립터는 상기 프로그램을 구성하는 상기 복수의 데이터 요소들을 각각 스크램블링하기 위한 복수의 스크램블 키들을 포함하는 복수의 ECM 패킷들의 패킷 ID들을 지정하는 프로그램 배포 시스템.

청구항 46.

제 45 항에 있어서,

상기 프로그램이 제 1 데이터 요소 내지 제 n 데이터 요소를 갖고 상기 제 1 데이터 요소 내지 제 n 데이터 요소에 대해 동일한 스크램블 키가 지정되면,

상기 프로그램 맵 테이블은 상기 프로그램을 나타내는 프로그램 번호와 상기 제 1 데이터 요소 내지 상기 제 n 데이터 요소를 각각 스크램블링하기 위한 스크램블 키들을 포함하는 ECM 패킷들의 패킷 ID들 사이의 대응관계를 기술하는 프로그램 배포 시스템.

청구항 47.

제 45 항에 있어서,

상기 프로그램이 제 1 데이터 요소 내지 제 n 데이터 요소를 갖고 상기 제 1 데이터 요소 내지 제 n 데이터 요소에 대해 적어도 하나의 다른 스크램블 키가 지정되면,

상기 프로그램 맵 테이블은 상기 제 1 데이터 요소를 포함하는 전송 스트림 패킷의 패킷 ID와 상기 제 1 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 데이터를 포함하는 전송 스트림 패킷의 패킷 ID 사이의 대응관계를 기술하고,

상기 프로그램 맵 테이블은 상기 제 n 데이터 요소를 포함하는 전송 스트림 패킷의 패킷 ID와 상기 제 n 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 데이터를 포함하는 전송 스트림 패킷의 패킷 ID 사이의 대응관계를 기술하는 프로그램 배포 시스템.

청구항 48.

제 42 항에 있어서,

상기 인코더/멀티플렉서 제어 시스템은 상기 프로그램 맵 테이블과 상기 조건부 액세스 테이블에 대해 고유의 패킷 ID들을 지정하는 프로그램 배포 시스템.

청구항 49.

제 42 항에 있어서,

상기 스크램블 수단은 상기 프로그램 지정 정보, 상기 EMM 데이터, 및 상기 ECM 데이터를 스크램블링하지 않고, 상기 데이터 요소만을 스크램블링하는 프로그램 배포 시스템.

청구항 50.

제 42 항에 있어서,

상기 스크램블 수단은 상기 데이터 요소들 각각을 포함하는 전송 스트림 패킷의 패킷 ID와 상기 데이터 요소에 대해 지정된 스크램블 키 사이의 대응관계를 나타내는 테이블에 기초하여 상기 데이터 요소들에 대해 지정된 스크램블 키들을 이용해 상기 데이터 요소들을 스크램블링하는 프로그램 배포 시스템.

청구항 51.

제 42 항에 있어서,

상기 인코더/멀티플렉서 제어 시스템은 ECM 패킷, EMM 패킷, PSI 패킷, 및 상기 기본 패킷을 각각 식별하기 위해 패킷 ID들이 지정될 때 복수의 전송 스트림 패킷들에 대해 패킷 ID의 반복 할당이 회피될 수 있도록 이전 동작들에 사용되었던 패킷 ID들을 저장하는 프로그램 배포 시스템.

청구항 52.

제 42 항에 있어서,

상기 인코더/멀티플렉서 제어 시스템은 각 전송 스트림 패킷에 할당된 패킷 ID와 상기 전송 스트림 패킷에 포함된 데이터를 스크램블링하는데 사용되는 스크램블 키 사이의 대응관계를 나타내는 테이블을 발생하고,

상기 인코더/멀티플렉서 제어 시스템은 상기 패킷 ID와 상기 스크램블 키 사이의 대응관계를 나타내는 상기 테이블을 상기 멀티플렉서 시스템에 공급하는 프로그램 배포 시스템.

청구항 53.

제 52 항에 있어서,

상기 스크램블 수단은 상기 프로그램 지정 정보, 상기 EMM 데이터, 및 상기 ECM 데이터를 스크램블링하지 않고, 상기 패킷 ID들과 상기 스크램블 키들 사이의 대응관계를 나타내는 상기 테이블을 참조하여 상기 데이터 요소만을 스크램블링하는 프로그램 배포 시스템.

청구항 54.

제 52 항에 있어서,

상기 스크램블 수단은 상기 패킷 ID들과 상기 스크램블 키들 사이의 대응관계를 나타내는 상기 테이블을 참조하여 상기 데이터 요소들에 대해 지정된 스크램블 키들로 상기 데이터 요소들을 스크램블링하는 프로그램 배포 시스템.

청구항 55.

제 52 항에 있어서,

상기 스크램블 수단은 상기 다중화 수단에 의해 상기 스크램블 수단에 공급된 모든 전송 스트림 패킷의 패킷 ID들을 검출하고,

상기 스크램블 수단은 상기 패킷 ID들과 상기 스크램블들 키들 사이의 대응관계를 나타내는 상기 테이블에 기초하여 상기 검출된 패킷 ID들에 대해 스크램블 키들이 정의되었는가의 여부를 결정하고,

상기 패킷 ID들에 대해 일부 스크램블 키가 정의되어 있으면, 상기 스크램블 수단은 상기 패킷 ID들에 의해 나타내어지는 전송 스트림 패킷들에 포함된 데이터 요소들을 상기 정의된 스크램블 키들로 스크램블링하고,

상기 패킷 ID들에 대해 일부 스크램블 키가 정의되어 있지 않으면, 상기 스크램블 수단은 상기 패킷 ID들에 의해 나타내어지는 전송 스트림 패킷들에 포함된 데이터를 스크램블링하지 않는 프로그램 배포 시스템.

청구항 56.

제 33 항에 있어서,

상기 멀티플렉서 시스템은:

PAT 패킷들, PMT 패킷들, CAT 패킷들, 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들, 암호화된 EMM 패킷들, 및 암호화된 ECM 패킷들을 각각 버퍼링하고, 상기 전송 스트림 패킷들을 상기 다중화 수단에 제공하는 복수의 버퍼 수단들을 더 포함하는 프로그램 배포 시스템.

청구항 57.

제 56 항에 있어서,

상기 멀티플렉서 시스템은 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들을 버퍼링하기 위한 복수의 버퍼들의 자유 영역을 모니터하고,

상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷을 버퍼링하기 위한 상기 복수의 버퍼들 중 어느 하나가 오버플로우되려고 하면, 상기 EMM 패킷들은 상기 EMM 패킷들을 버퍼링하기 위한 버퍼에 의해 상기 다중화 수단에 제공되지 않고, 그 대신 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들은 오버플로우될 것 같은 상기 버퍼에 의해 상기 다중화 수단에 제공되는 프로그램 배포 시스템.

청구항 58.

제 47 항에 있어서,

상기 멀티플렉서 시스템에 의해 제공된 전송 스트림들을 전송선을 통해 상기 수신단에 배포하는 배포 시스템; 및

상기 전송선을 통해 전송된 상기 전송 스트림들을 수신하기 위한 수신 시스템을 더 포함하는 프로그램 배포 시스템.

청구항 59.

제 58 항에 있어서,

상기 수신 시스템은:

상기 전송된 전송 스트림들을 역다중화하는 디멀티플렉서;

상기 스크램블링된 데이터 요소들을 상기 공급된 스크램블 키들로 각각 디스크램블링하는 디스크램블러;

각 데이터 요소에 대해 상기 디스크램블링된 데이터를 복호화하는 디코더;

상기 전송 스트림을 구성하는 전송 스트림 패킷들을 분석하는 CPU; 및

상기 전송 스트림에 포함된 상기 암호화된 스크램블 키를 해독하고 상기 해독된 스크램블 키를 상기 디스크램블러에 공급하는 보안 모듈을 포함하는 프로그램 배포 시스템.

청구항 60.

제 59 항에 있어서,

상기 CPU는:

상기 전송 스트림에 포함된 프로그램 연관 테이블을 분석하기 위한 PAT 분석 수단;

상기 전송 스트림에 포함된 프로그램 맵 테이블을 분석하기 위한 PMT 분석 수단;

상기 전송 스트림에 포함된 조건부 액세스 테이블을 분석하기 위한 CAT 분석 수단;

상기 전송 스트림에 포함된 암호화 EMM 데이터를 분석하기 위한 EMM 분석 수단; 및

상기 전송 스트림에 포함된 암호화 ECM 데이터를 분석하기 위한 ECM 분석 수단을 포함하는 프로그램 배포 시스템.

청구항 61.

제 59 항에 있어서,

상기 보안 모듈은:

상기 EMM 데이터에 포함된 가입자의 가입 정보를 저장하는 메모리 수단;

상기 전송된 전송 스트림에 포함된 상기 암호화된 작업키 및 상기 가입자 관리 시스템에 의해 사용된 것과 동일한 마스터 키를 수신하여 상기 암호화된 작업키를 상기 마스터 키로 해독하는 제 1 해독 수단; 및

상기 전송 스트림에 포함된 상기 암호화된 스크램블 키 및 상기 제 1 해독 수단에 의해 공급된 상기 해독된 작업키를 수신하여 상기 암호화된 스크램블 키를 상기 해독된 작업키로 해독하는 제 2 해독 수단을 포함하는 프로그램 배포 시스템.

청구항 62.

제 61 항에 있어서,

상기 CPU는 상기 전송 스트림에 포함된 프로그램 연관 테이블 및 프로그램 맵 테이블을 분석함으로써 상기 프로그램을 구성하는 각 데이터 요소들을 포함하는 전송 스트림 패킷을 식별하고, 상기 디멀티플렉서를 제어하여 상기 데이터 요소를 포함하는 상기 전송 스트림 패킷을 상기 디스크램블러들 중 적절한 것에 제공하는 프로그램 배포 시스템.

청구항 63.

제 62 항에 있어서,

상기 CPU는 상기 전송 스트림에 포함된 조건부 액세스 테이블을 분석함으로써 EMM 데이터를 포함하는 전송 스트림 패킷을 검출하고,

상기 CPU는 상기 EMM 데이터를 포함하는 상기 전송 스트림으로부터, 가입자에 의해 가입된 프로그램에 대한 EMM 데이터를 갖는 전송 스트림 패킷만을 필터링하고,

상기 CPU는 상기 필터링된 전송 스트림 패킷에 포함된 상기 EMM 데이터를 분석함으로써 상기 EMM 데이터로부터 상기 암호화된 작업키를 얻는 프로그램 배포 시스템.

청구항 64.

제 61 항에 있어서,

상기 CPU는 상기 전송 스트림에 포함된 프로그램 연관 테이블과 상기 프로그램 연관 테이블에 의해 지정된 프로그램 맵 테이블을 분석함으로써, 상기 프로그램과 상기 ECM 데이터를 구성하는 복수의 데이터 요소들을 포함하는 전송 스트림 패킷들을 각각 검출하고,

상기 CPU는 상기 디멀티플렉서를 제어하여 상기 복수의 데이터 요소들을 포함하는 상기 전송 스트림 패킷들을 상기 디스크램블러에 각각 공급하고 상기 ECM 데이터를 포함하는 상기 전송 스트림 패킷을 수신하는 프로그램 배포 시스템.

청구항 65.

제 64 항에 있어서,

상기 CPU는 상기 디멀티플렉서에 의해 공급된 상기 암호화 ECM 데이터를 포함하는 상기 전송 스트림 패킷들로부터, 가입자에 의해 가입된 프로그램이나 데이터 요소에 대한 암호화 ECM 데이터를 갖는 전송 스트림 패킷만을 필터링하고,

상기 CPU는 상기 필터링된 전송 스트림 패킷에 포함된 상기 암호화 ECM 데이터를 분석함으로써 상기 암호화된 ECM 데이터로부터 상기 암호화 스크램블 키를 얻는 프로그램 배포 시스템.

청구항 66.

제 65 항에 있어서,

상기 프로그램 번호와 상기 암호화 ECM 패킷의 패킷 ID 사이의 대응관계가 상기 프로그램 맵 테이블의 구문(syntax)에 따라 기술되면,

상기 CPU는 상기 패킷 ID에 의해 지정된 상기 암호화 ECM 패킷에 포함된 암호화된 스크램블 키를 상기 프로그램에 대응하는 암호화된 스크램블 키로서 상기 보안 모듈에 공급하고,

상기 프로그램을 구성하는 복수의 데이터 요소들과 상기 복수의 암호화된 ECM 패킷들의 패킷 ID들 사이의 대응관계가 상기 프로그램 맵 테이블의 구문에 따라 기술되면,

상기 CPU는 상기 복수의 패킷 ID들에 의해 지정된 상기 암호화 ECM 패킷에 포함되는 복수의 다른 스크램블 키들을 상기 복수의 데이터 요소들에 대응하는 암호화된 스크램블 키들로서 상기 보안 모듈에 공급하는 프로그램 배포 시스템.

청구항 67.

제 66 항에 있어서,

상기 프로그램과 연관된 암호화 스크램블 키가 상기 CPU에 의해 공급되면, 상기 보안 모듈은 상기 공급된 암호화 스크램블 키를 해독하여 상기 프로그램을 구성하는 복수의 데이터 요소들에 대응하는 복수의 디스크램블러들에 동일한 스크램블 키를 각각 공급하고,

상기 복수의 데이터 요소들과 연관된 복수의 암호화 스크램블 키들이 상기 CPU에 의해 공급되면, 상기 보안 모듈은 상기 복수의 공급된 암호화 스크램블 키들을 각각 해독하여 상기 복수의 데이터 요소들 중에서 가입된 데이터 요소들에 대응하는 복수의 디스크램블러들에 상이한 스크램블 키들을 공급하는 프로그램 배포 시스템.

청구항 68.

삭제

청구항 69.

삭제

청구항 70.

삭제

청구항 71.

복수의 데이터 요소들로 구성된 프로그램을 전송하는 프로그램 전송 방법에 있어서:

가입자가 그 가입자에 의해 가입된 프로그램들이나 데이터 요소들만을 보고 및/또는 들을 수 있도록 상기 프로그램에 포함된 복수의 데이터 요소들을 스크램블링하는데 사용될 복수의 스크램블 키들을 발생하는 스크램블 키 발생 단계;

각 프로그램에 대해 부호화된 데이터 요소들로 구성된 부호화 스트림들을 발생하도록 상기 프로그램에 포함된 상기 데이터 요소들의 각각을 부호화하는 부호화 단계;

상기 부호화 단계에 의해 각 프로그램에 대해 제공된 상기 부호화 스트림들을 다중화하는 다중화 단계; 및

상기 발생된 스크램블 키에 기초하여 상기 다중화 스트림에 포함된 상기 부호화 데이터 요소들의 각각을 선택적으로 스크램블링하는 스크램블 단계를 포함하고,

상기 다중화 단계는 ECM 데이터에 포함된 스크램블 키를 암호화하기 위한 암호화 단계를 포함하고,

상기 스크램블 키 발생 단계는 작업키와 상기 작업키를 식별하기 위한 작업키 식별 번호 사이의 대응관계를 나타내는 작업키 테이블을 발생하고,

상기 다중화 단계에서의 상기 암호화 단계는 상기 작업키 테이블을 참조하여 상기 ECM 데이터에 포함된 상기 작업키 식별 번호로부터 작업키를 얻고,

상기 암호화 단계는 상기 작업키 테이블로부터 얻어진 상기 작업키를 사용해 상기 ECM 데이터에 포함된 상기 스크램블 키를 암호화하고,

상기 암호화 단계는 제 2 암호화 회로에 의해 암호화된 상기 암호화 스크램블 키를 암호화된 ECM 데이터로서 제공하는 프로그램 전송 방법.

청구항 72.

삭제

청구항 73.
삭제

청구항 74.
삭제

청구항 75.
삭제

청구항 76.
삭제

청구항 77.
삭제

청구항 78.

제 71 항에 있어서,

모든 전송 스트림 패킷에 상기 전송 스트림 패킷들을 식별하기 위한 패킷 ID들을 할당하는 프로그램 지정 정보 발생 단계를 더 포함하는 프로그램 전송 방법.

청구항 79.

제 78 항에 있어서,

상기 프로그램 지정 정보는 적어도 프로그램 연관 테이블, 프로그램 맵 테이블, 및 조건부 액세스 테이블로 구성되는 프로그램 전송 방법.

청구항 80.

제 79 항에 있어서,

상기 프로그램 지정 정보 발생 단계는 상기 프로그램 연관 테이블을 포함하는 전송 스트림 패킷을 PAT 패킷으로서 상기 다중화 단계에 공급하고,

상기 프로그램 지정 정보 발생 단계는 상기 프로그램 맵 테이블을 포함하는 전송 스트림 패킷을 PMT 패킷으로서 상기 다중화 단계에 공급하고,

상기 프로그램 지정 정보 발생 단계는 상기 조건부 액세스 테이블을 포함하는 전송 스트림 패킷을 CAT 패킷으로서 상기 다중화 단계에 공급하는 프로그램 전송 방법.

청구항 81.

제 80 항에 있어서,

상기 프로그램 연관 테이블은 프로그램 번호와 상기 프로그램 번호에 대응하는 PMT 패킷의 패킷 ID를 지정하는 테이블이고,

상기 프로그램 맵 테이블은 프로그램을 구성하는 복수의 데이터 요소들 각각을 포함하는 전송 스트림 패킷의 패킷 ID를 지정하는 테이블이고,

상기 조건부 액세스 테이블은 상기 암호화된 EMM 패킷의 패킷 ID를 지정하는 테이블인 프로그램 전송 방법.

청구항 82.

제 81 항에 있어서,

상기 프로그램 연관 테이블은 프로그램을 나타내는 프로그램 번호 및 상기 프로그램과 연관된 PMT 패킷의 패킷 ID를 기술하고,

상기 프로그램 맵 테이블은 상기 프로그램을 나타내는 프로그램 번호, 상기 프로그램을 구성하는 복수의 데이터 요소들을 포함하는 전송 스트림 패킷을 포함하는 복수의 패킷 ID들, 및 상기 프로그램이나 상기 데이터 요소와 연관된 암호화된 ECM 패킷의 상기 패킷 ID를 지정하는 디스크립터를 기술하는 프로그램 전송 방법.

청구항 83.

제 82 항에 있어서,

상기 프로그램 맵 테이블내의 상기 디스크립터가 상기 프로그램 번호에 대응하는 위치에 기술되어 있으면, 상기 디스크립터는 상기 프로그램을 구성하는 상기 복수의 데이터 요소들의 모든 데이터 요소들을 스크램블링하기 위한 스크램블 키를 포함하는 ECM 패킷의 패킷 ID를 지정하고,

상기 프로그램 맵 테이블내의 상기 디스크립터가 상기 프로그램의 상기 데이터 요소들 각각에 대응하는 위치에 기술되어 있으면, 상기 디스크립터는 상기 프로그램을 구성하는 상기 복수의 데이터 요소들을 각각 스크램블링하기 위한 복수의 스크램블 키들을 포함하는 복수의 ECM 패킷들의 패킷 ID들을 각각 지정하는 프로그램 전송 방법.

청구항 84.

제 83 항에 있어서,

상기 프로그램이 제 1 데이터 요소 내지 제 n 데이터 요소를 갖고 상기 제 1 데이터 요소 내지 상기 제 n 데이터 요소에 대해 적어도 하나의 다른 스크램블 키가 지정되면,

상기 프로그램 맵 테이블은 상기 제 1 데이터 요소를 포함하는 전송 스트림 패킷의 패킷 ID와 상기 제 1 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 데이터를 포함하는 전송 스트림 패킷의 패킷 ID 사이의 대응관계를 기술하고,

상기 프로그램 맵 테이블은 상기 제 n 데이터 요소를 포함하는 전송 스트림 패킷의 패킷 ID와 상기 제 n 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 데이터를 포함하는 전송 스트림 패킷의 패킷 ID 사이의 대응관계를 기술하는 프로그램 전송 방법.

청구항 85.

제 80 항에 있어서,

상기 스크램블 단계는 상기 프로그램 지정 정보, EMM 데이터, 및 상기 ECM 데이터를 스크램블링하지 않고, 상기 데이터 요소들 각각을 포함하는 전송 스트림 패킷의 패킷 ID와 상기 데이터 요소에 대해 지정된 스크램블 키 사이의 대응관계를 나타내는 테이블에 기초하여 상기 데이터 요소들에 대해 지정된 스크램블 키들을 사용해 상기 데이터 요소들만을 스크램블링하는 프로그램 전송 방법.

청구항 86.

제 80 항에 있어서,

상기 프로그램 지정 정보 발생 단계는 각 전송 스트림 패킷에 할당된 패킷 ID와 상기 전송 스트림 패킷에 포함된 데이터를 스크램블링하는데 사용되는 스크램블 키 사이의 대응관계를 나타내는 테이블을 발생하고,

상기 프로그램 지정 정보 발생 단계는 상기 패킷 ID들과 상기 스크램블 키들 사이의 대응관계를 나타내는 상기 테이블을 상기 다중화 단계에 공급하는 프로그램 전송 방법.

청구항 87.

제 86 항에 있어서,

상기 스크램블 수단은 상기 프로그램 지정 정보, EMM 데이터, 및 상기 ECM 데이터를 스크램블링하지 않고, 상기 패킷 ID들과 상기 스크램블 키들 사이의 대응관계를 나타내는 상기 테이블을 참조하여 상기 데이터 요소들만을 스크램블링하는 프로그램 전송 방법.

청구항 88.

제 86 항에 있어서,

상기 스크램블 단계는 상기 다중화 수단에 의해 상기 스크램블 수단에 공급된 모든 전송 스트림 패킷들의 패킷 ID들을 검출하고,

상기 스크램블 단계는 상기 패킷 ID들과 상기 스크램블들 키들 사이의 대응관계를 나타내는 상기 테이블에 기초하여 상기 검출된 패킷 ID들에 대해 스크램블 키들이 정의되었는가의 여부를 결정하고,

일부 스크램블 키들이 상기 패킷 ID들에 대해 정의되면, 상기 스크램블 단계는 상기 패킷 ID들로 나타내어지는 전송 스트림 패킷들에 포함된 데이터 요소들을 상기 정의된 스크램블 키들로 스크램블링하고,

스크램블 키들이 상기 패킷 ID들에 대해 정의되어 있지 않으면, 상기 스크램블 단계는 상기 패킷 ID들로 나타낸 전송 스트림 패킷들에 포함된 데이터를 스크램블링하지 않는 프로그램 전송 방법.

청구항 89.

제 71 항에 있어서,

상기 다중화 단계는 상기 스크램블 키들을 상기 작업키로 암호화하고,

상기 다중화 단계는 PAT 패킷들, PMT 패킷들, CAT 패킷들, 상기 데이터 요소를 포함하는 전송 스트림 패킷들, EMM 패킷들, 및 ECM 패킷들을 각각 복수의 버퍼 수단들에 버퍼링하는 프로그램 전송 방법.

청구항 90.

제 89 항에 있어서,

상기 다중화 단계는 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들을 버퍼링하기 위한 복수의 버퍼들의 자유 영역을 모니터하고,

상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들을 버퍼링하기 위한 상기 복수의 버퍼들 중 어느 하나가 오버플로우되려고 하면, 상기 EMM 패킷들은 상기 EMM 패킷들을 버퍼링하기 위한 버퍼에 의해 상기 다중화 단계에 제공되지 않고, 그 대신 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷은 오버플로우될 것 같은 상기 버퍼에 의해 제공되는 프로그램 전송 방법.

청구항 91.

프로그램 배포 시스템에 의해 배포된 복수의 프로그램들과 상기 프로그램들을 구성하는 복수의 데이터 요소들 중에서 가입된 프로그램들과 데이터 요소들에만 조건부 액세스를 제공하는 조건부 액세스 시스템에 있어서:

상기 프로그램 배포 시스템은:

가입자가 상기 가입자에 의해 가입된 데이터 프로그램이나 요소를 보고 및/또는 들을 수 있도록 상기 프로그램에 포함된 상기 데이터 요소들 각각을 스크램블링하는데 사용될 복수의 스크램블 키들을 발생하는 가입자 시청허가 시스템을 포함하고,

상기 가입자 시청허가 시스템은 상기 스크램블 키를 암호화하는데 사용될 작업키를 마스터 키로 암호화하는 제 1 암호화 수단을 포함하고,

상기 조건부 액세스 시스템은:

상기 전송 스트림으로부터, 상기 프로그램을 구성하는 복수의 스크램블링된 데이터 요소들을 포함하는 전송 스트림 패킷을 역다중화하고, 상기 복수의 데이터 요소들과 연관된 복수의 암호화 스크램블 키들을 포함하는 복수의 전송 스트림 패킷들을 역다중화하는 디멀티플렉서 수단;

상기 복수의 역다중화된 암호화 스크램블 키들을 포함하는 상기 복수의 전송 스트림 패킷들로부터, 가입자에 의해 가입된 상기 프로그램들 및 데이터 요소들과 연관된 암호화 스크램블 키를 포함하는 전송 스트림 패킷을 필터링하는 필터 수단;

상기 복수의 필터링된 전송 스트림 패킷들에 포함된 상기 복수의 암호화 스크램블 키들을 해독하여 복수의 해독된 스크램블 키들을 발생하는 해독 수단;

상기 복수의 데이터 요소들과 연관된 상기 복수의 해독된 스크램블 키들을 사용해 각 데이터 요소에 대해 상기 복수의 역다중화된 데이터 요소들을 디스크램블링하는 디스크램블 수단;

상기 디스크램블 수단에 의해 디스크램블링된 상기 복수의 데이터 요소들을 복호화하는 복호화 수단; 및

ECM 데이터에 포함된 스크램블 키를 암호화하는 제 2 암호화 수단을 가진 멀티플렉서 시스템을 포함하고,

상기 가입자 시청허가 시스템은 작업키와 상기 작업키를 식별하기 위한 작업키 식별 번호 사이의 대응관계를 나타내는 작업키 테이블을 상기 멀티플렉서 시스템의 상기 제 2 암호화 수단에 공급하고,

상기 제 2 암호화 수단은 상기 작업키 테이블을 참조하여 상기 ECM 데이터에 포함된 상기 작업키 식별 번호로부터 작업키를 얻고,

상기 제 2 암호화 수단은 상기 작업키 테이블로부터 얻어진 상기 작업키를 사용해 상기 ECM 데이터에 포함된 상기 스크램블 키를 암호화하고,

상기 제 2 암호화 수단은 상기 제 2 암호화 수단에 의해 암호화된 상기 암호화 스크램블 키를 암호화된 ECM 데이터로서 상기 멀티플렉서 시스템에 공급하는 조건부 액세스 시스템.

청구항 92.

삭제

청구항 93.

삭제

청구항 94.

삭제

청구항 95.

삭제

청구항 96.

삭제

청구항 97.

삭제

청구항 98.

삭제

청구항 99.

제 91 항에 있어서,

전송 스트림 패킷들의 형태로 상기 멀티플렉서 시스템에 공급된 모든 전송 스트림 패킷들에 상기 전송 스트림 패킷을 식별하기 위한 패킷 ID들을 할당하는 인코더/멀티플렉서 제어 시스템을 더 포함하는 조건부 액세스 시스템.

청구항 100.

제 99 항에 있어서,

적어도 프로그램 연관 테이블, 프로그램 맵 테이블, 및 조건부 액세스 테이블로 구성될 프로그램 지정 정보를 더 포함하는 조건부 액세스 시스템.

청구항 101.

제 100 항에 있어서,

상기 인코더/멀티플렉서 제어 시스템은 상기 프로그램 연관 테이블을 포함하는 전송 스트림 패킷을 PAT 패킷으로서 상기 멀티플렉서 시스템에 공급하고,

상기 인코더/멀티플렉서 제어 시스템은 상기 프로그램 맵 테이블을 포함하는 전송 스트림 패킷을 PMT 패킷으로서 상기 멀티플렉서 시스템에 공급하고,

상기 인코더/멀티플렉서 제어 시스템은 상기 조건부 액세스 테이블을 포함하는 전송 스트림 패킷을 CAT 패킷으로서 상기 멀티플렉서 시스템에 공급하는 조건부 액세스 시스템.

청구항 102.

제 101 항에 있어서,

상기 프로그램 연관 테이블은 프로그램 번호와 상기 프로그램 번호에 대응하는 PMT 패킷의 패킷 ID를 지정하는 테이블이고,

상기 프로그램 맵 테이블은 프로그램을 구성하는 복수의 데이터 요소들 각각을 포함하는 전송 스트림 패킷의 패킷 ID를 지정하는 테이블이고,

상기 조건부 액세스 테이블은 상기 암호화된 EMM 패킷의 패킷 ID를 지정하는 테이블인 조건부 액세스 시스템.

청구항 103.

제 102 항에 있어서,

상기 프로그램 연관 테이블은 프로그램을 나타내는 상기 프로그램 번호 및 상기 프로그램과 연관된 PMT 패킷의 상기 패킷 ID를 기술하고,

상기 프로그램 맵 테이블은 상기 프로그램을 나타내는 프로그램 번호, 상기 프로그램을 구성하는 복수의 데이터 요소들을 포함하는 전송 스트림 패킷을 포함하는 복수의 패킷 ID들, 및 상기 프로그램이나 상기 데이터 요소와 연관된 암호화 ECM 패킷의 패킷 ID를 지정하는 디스크립터를 기술하는 조건부 액세스 시스템.

청구항 104.

제 103 항에 있어서,

상기 프로그램 맵 테이블내의 상기 디스크립터가 상기 프로그램 번호에 대응하는 위치에 기술되어 있으면, 상기 디스크립터는 상기 프로그램을 구성하는 상기 복수의 데이터 요소들의 모든 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 패킷의 패킷 ID를 지정하고,

상기 프로그램 맵 테이블내의 상기 디스크립터가 상기 프로그램의 상기 데이터 요소 각각에 대응하는 위치에 기술되어 있으면, 상기 디스크립터는 상기 프로그램을 구성하는 상기 복수의 데이터 요소들을 각각 스크램블링하기 위한 복수의 스크램블 키들을 포함하는 복수의 ECM 패킷들의 패킷 ID들을 지정하는 조건부 액세스 시스템.

청구항 105.

제 104 항에 있어서,

상기 프로그램이 제 1 데이터 요소 내지 제 n 데이터 요소를 갖고 상기 제 1 데이터 요소 내지 제 n 데이터 요소에 대해 적어도 하나의 다른 스크램블 키가 지정되면,

상기 프로그램 맵 테이블은 상기 제 1 데이터 요소를 포함하는 전송 스트림 패킷의 패킷 ID와 상기 제 1 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 데이터를 포함하는 전송 스트림 패킷의 패킷 ID 사이의 대응관계를 기술하고,

상기 프로그램 맵 테이블은 상기 제 n 데이터 요소를 포함하는 전송 스트림 패킷의 패킷 ID와 상기 제 n 데이터 요소를 스크램블링하기 위한 스크램블 키를 포함하는 ECM 데이터를 포함하는 전송 스트림 패킷의 패킷 ID 사이의 대응관계를 기술하는 조건부 액세스 시스템.

청구항 106.

삭제

청구항 107.

제 101 항에 있어서,

상기 인코더/멀티플렉서 제어 시스템은 각 전송 스트림 패킷에 할당된 상기 패킷 ID와 상기 전송 스트림 패킷에 포함된 데이터를 스크램블링하는데 사용되는 스크램블 키 사이의 대응관계를 나타내는 테이블을 발생하고,

상기 인코더/멀티플렉서 제어 시스템은 상기 패킷 ID들과 상기 스크램블 키 들사이의 대응관계를 나타내는 상기 테이블을 상기 멀티플렉서 시스템에 공급하는 조건부 액세스 시스템.

청구항 108.

삭제

청구항 109.

삭제

청구항 110.

제 91 항에 있어서,

상기 멀티플렉서 시스템은:

PAT 패킷들, PMT 패킷들, CAT 패킷들, 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들, 암호화된 EMM 패킷들, 및 암호화된 ECM 패킷들을 각각 버퍼링하고, 상기 전송 스트림 패킷들을 상기 멀티플렉서 시스템에 제공하는 복수의 버퍼 수단들을 더 포함하는 조건부 액세스 시스템.

청구항 111.

제 110 항에 있어서,

상기 멀티플렉서 시스템은 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들을 버퍼링하기 위한 복수의 버퍼들의 자유 영역을 모니터하고,

상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들을 버퍼링하기 위한 상기 복수의 버퍼들 중 어느 하나가 오버플로우되려고 하면, 상기 EMM 패킷들은 상기 EMM 패킷들을 버퍼링하기 위한 버퍼에 의해 상기 멀티플렉서 시스템에 제공되지 않고, 그 대신 상기 데이터 요소들을 포함하는 상기 전송 스트림 패킷들은 오버플로될 것 같은 상기 버퍼에 의해 상기 멀티플렉서 시스템에 제공되는 조건부 액세스 시스템.

청구항 112.
삭제

청구항 113.
삭제

청구항 114.
삭제

청구항 115.
삭제

청구항 116.
삭제

청구항 117.
삭제

청구항 118.
삭제

청구항 119.

제 107 항에 있어서,

상기 프로그램과 연관된 암호화 스크램블 키가 상기 필터 수단에 의해 공급되면, 상기 해독 수단은 상기 공급된 암호화 스크램블 키를 해독하여 상기 프로그램을 구성하는 복수의 데이터 요소들에 대응하는 복수의 디스크램블러들에 동일한 스크램블 키를 각각 공급하고,

상기 복수의 데이터 요소들과 연관된 복수의 암호화된 스크램블 키들이 상기 필터 수단에 의해 공급되면, 상기 해독 수단은 상기 복수의 공급된 암호화 스크램블 키들을 각각 해독하여 상기 복수의 데이터 요소들 중에서 가입된 데이터 요소들에 대응하는 복수의 디스크램블러들에 상이한 스크램블 키들을 공급하는 조건부 액세스 시스템.

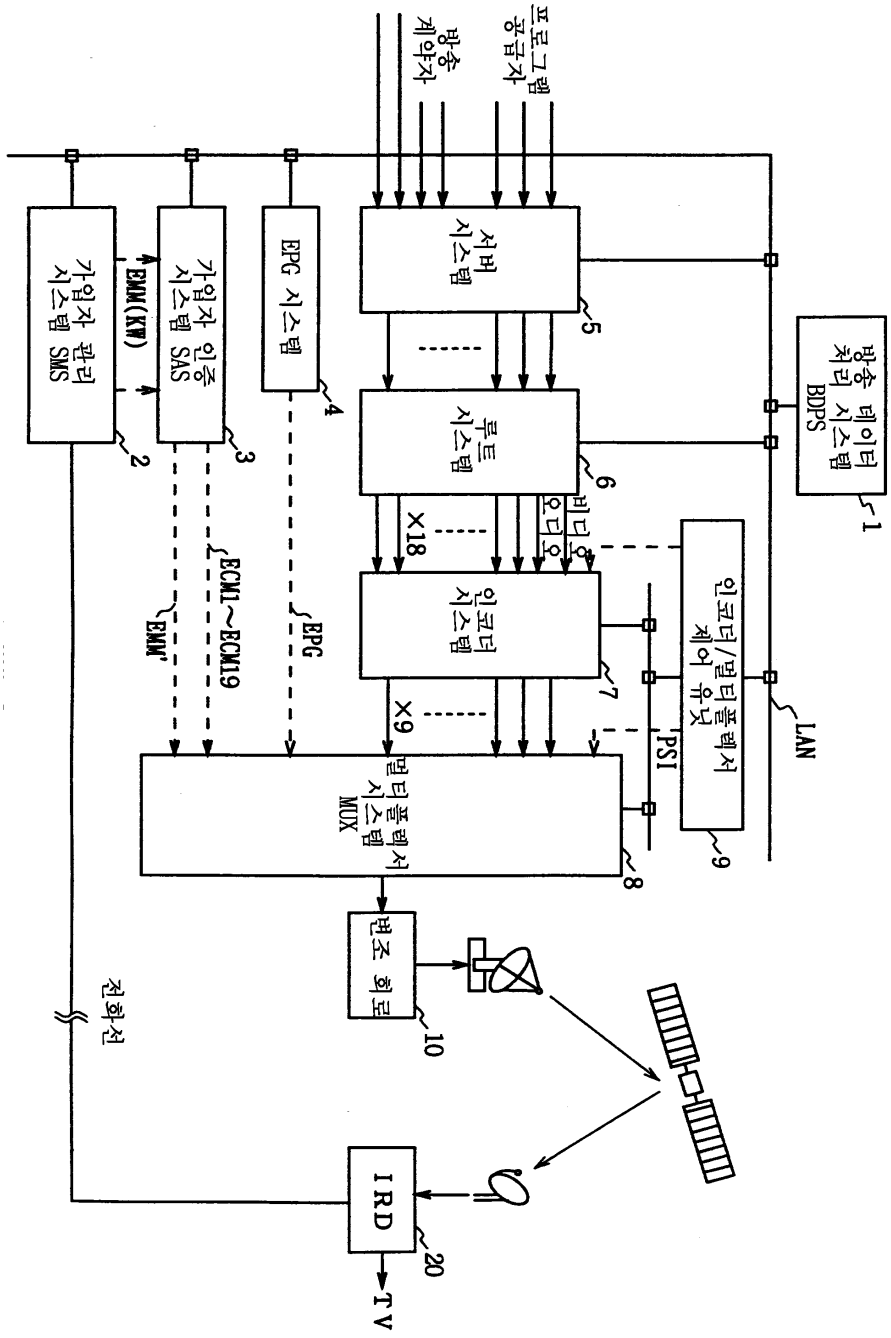
청구항 120.
삭제

청구항 121.
삭제

청구항 122.
삭제

도면

도면1



도면2

프로그램 번호	비디오	메인 오디오	서브 오디오	전용
1	Ks1		---	---
2	Ks2		Ks3	Ks4
3	Ks5	Ks6	---	---
4	Ks7	Ks8	Ks9	Ks10
5	Ks11			
6	Ks12		---	---
7	Ks13			Ks14
8	Ks15			
9	Ks16	Ks17	Ks18	Ks19

도면3

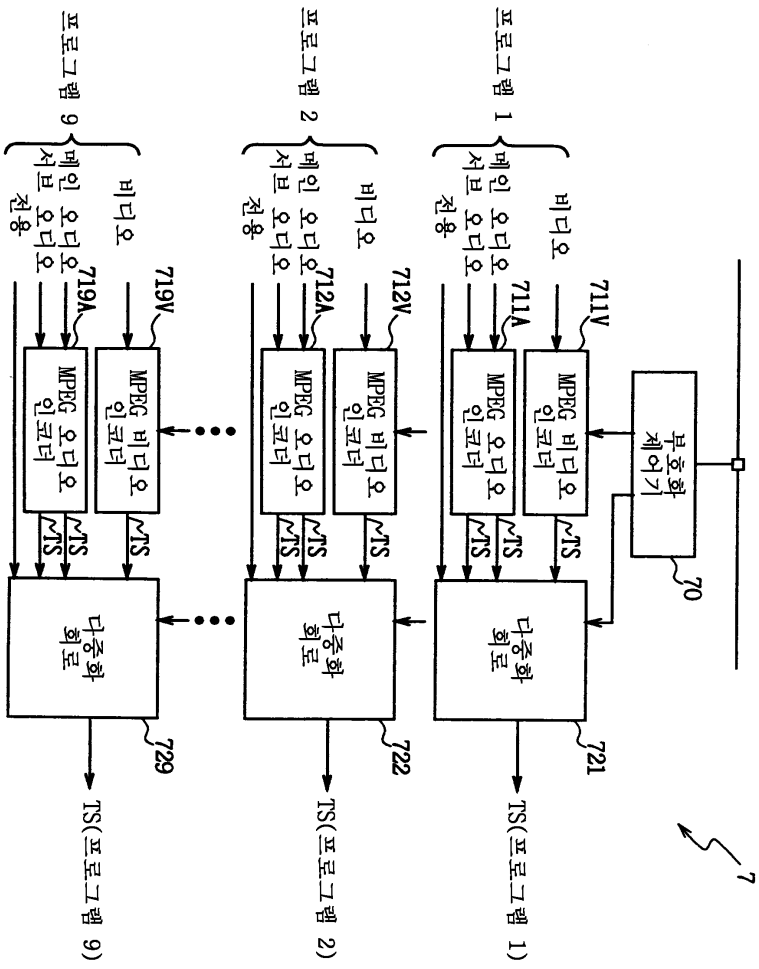
PID값	패킷에 기록된 정보
0x0000	PAT
0x0001	CAT
0x0002~0x000F	Reserved
0x0010	NIT、ST
0x0011	SDT、BAT、ST
0x0012	EIT、ST
0x0013	RST、ST
0x0014	TDT
0x0015~0x001F	예약
0x0020~0x1FFE	PMT, 비디오/오디오 데이터 스트림
0x1FFF	널 패킷

도면4

PID 테이블

패킷 종류	PID값	스크램블 키
PATPACKET	0×0000 고정값	---
•	•	
•	•	
•	•	
PMT1PACKET	0×0100	---
PMT2PACKET	0×0101	---
•	•	
•	•	
•	•	
•	•	
ECM1PACKET	0×0300	---
•	•	
•	•	
ECM2PACKET	0×0351	---
ECM2PACKET	0×0352	---
ECM3PACKET	0×0353	---
ECM4PACKET	0×0354	---
•	•	
•	•	
•	•	
•	•	
Video[1]PACKET	0×0500	Ks1
Main_Audio[1]PACKET	0×0501	Ks1
Video[2]PACKET	0×0502	Ks2
Main_Audio[2]PACKET	0×0503	Ks2
Sub_Audio[2]PACKET	0×0504	Ks3
Private[2]PACKET	0×0505	Ks4
•	•	
•	•	
•	•	
•	•	
CATPACKET	0×0001 (FIXED VALUE)	---
•	•	
•	•	
•	•	
•	•	
EMMPACKET	0×0700	---
•	•	
•	•	
•	•	

도면5



도면6

구문	비트수	니모닉
transport packet(){		
sync_byte	8	bslbf
transport_error_indicator	1	bslbf
payload_unit_start_indicator	1	bslbf
transport_priority	1	bslbf
PID	13	uimsbf
transport_scrambling_control	2	bslbf
adaptation_field_control	2	bslbf
continuity_counter	4	uimsbf
if(adaptation_field_control==' || adaptation_field_control=='11'{		
adaptation filed()		
}		
if(adaptation_field_control==' || adaptation_field_control=='11'{		
for(i=0;i<N;i++){		
data_byte	8	bslbf
}		
}		
}		

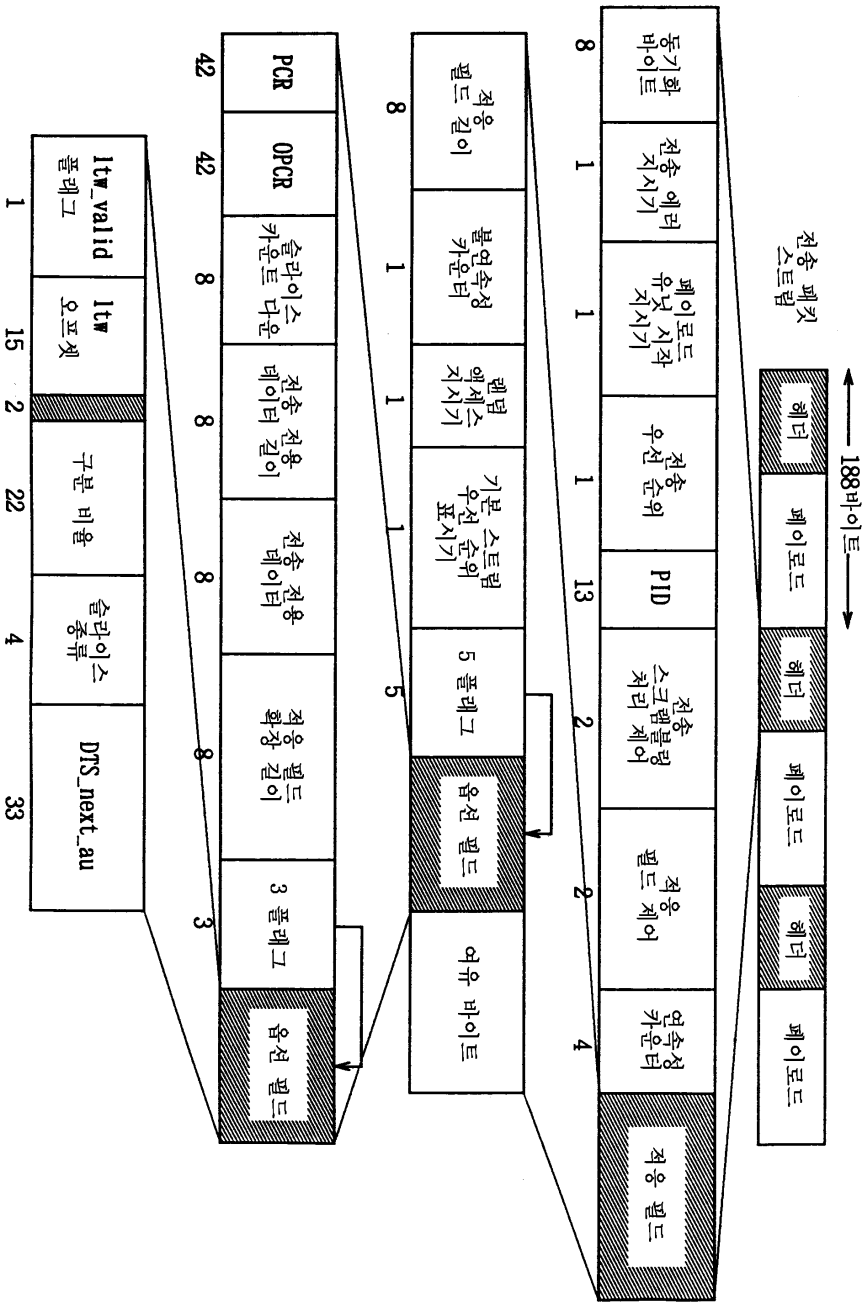
도면7

구문	비트수	니모닉
adaptation_field(){		
adaptation_filed_length	8	uimsbf
if(adaptation_field_length>0){		
discontinuity_indicator	1	bslbf
random access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_point_flag	1	bslbf
transport_private_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if(PCR_flag=='1'){		
program_clock_reference_base	33	uimsbf
reserved	6	bslbf
program_clock_reference_extension	9	uimsbf
}		
if(OPCR_flag=='1'){		
original_program_clock_reference_base	33	uimsbf
reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
if(splicing_point_flag=='1'){		
splice_countdown	8	tcimsbf
}		
if(transport_private_data_flag=='1'){		
transport_private_data_length	8	uimsbf
for(i=0;i<transport_private_data_length;i++;){		
private_data_byte	8	bslbf
}		
}		
if(adaptation_field_extension_flag=='1'){		
adaptation_field_extension_length	8	uimsbf
ltw_flag	1	bslbf
piecewise_rate_flag	1	bslbf
seamless_splice_flag	1	bslbf

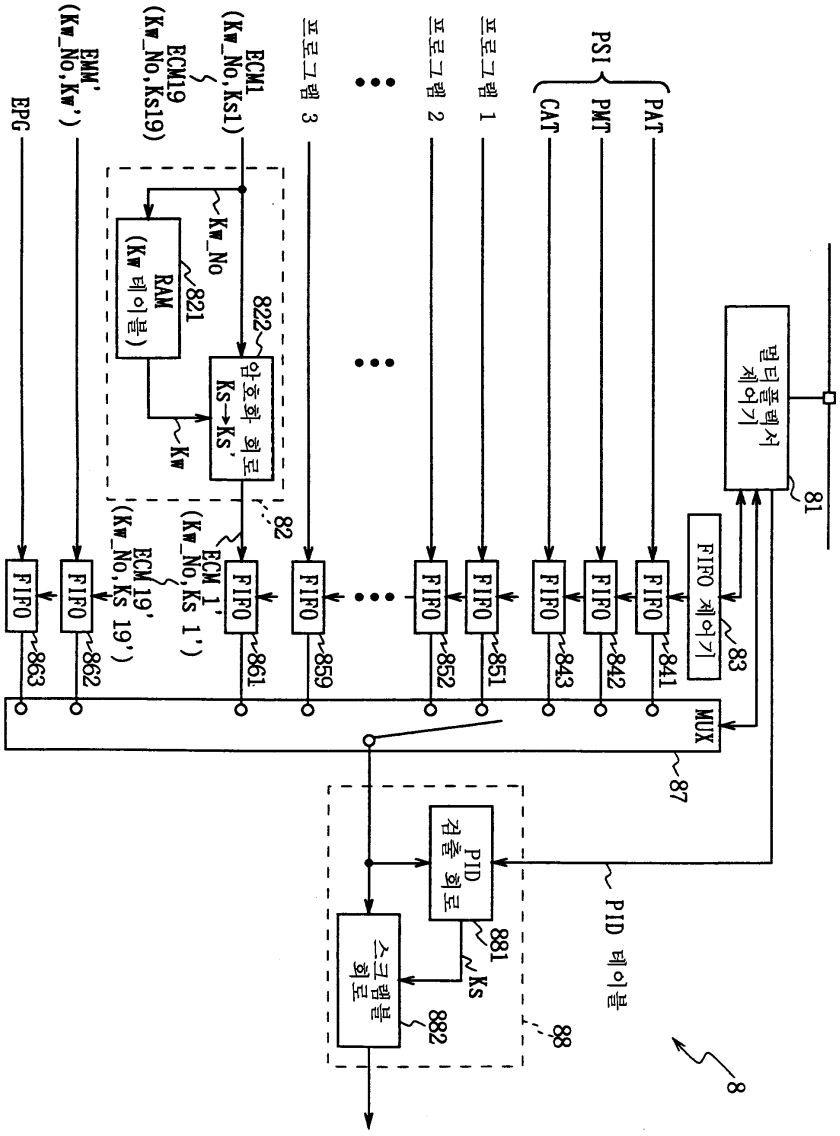
도면8

```
            reserved                                    5       bslbf
            if(ltw_flag=='1'){
                ltw_valid_flag                          1       bslbf
                ltw_offset                             15       uimsbf
            }
            if(piecewise_rate_flag=='1'){
                reserved                                2       bslbf
                piecewise_rate                         22       uimsbf
            }
            if(seamless_splice_flag=='1'){
                splice_type                             4       bslbf
                DTS_next_AU[32..30]                     3       bslbf
                market_bit                              1       bslbf
                DTS_next_AU[29..15]                    15       bslbf
                marker_bit                              1       bslbf
                DTS_next_AU[14..0]                     15       bslbf
                marker_bit                              1       bslbf
            }
            for(i=0;i<N;i++){
                reserved                                8       bslbf
            }
    }
    for(i=0;i<N;i++){
            stuffing_byte                               8       bslbf
    }
  }
}
```

도면9


전송 패킷 스트림
188바이트
헤더
페이로드
헤더
페이로드
헤더
페이로드
동기화 바이트
전송 에러 지시기
페이로드 유닛 시작 지시기
전송 우선 순위
PID
전송 스크램블링 처리 제어
적응 필드 제어
연속성 카운터
적응 필드
8
1
1
1
13
2
2
4
적응 필드 길이
불연속성 카운터
랜덤 액세스 지시기
기본 스트림 우선 순위 표시기
5 플래그
옵션 필드
여유 바이트
8
1
1
1
5
PCR
OPCR
슬라이스 카운트 다운
전송 전용 데이터 길이
전송 전용 데이터
적응 필드 확장 길이
3 플래그
옵션 필드
42
42
8
8
8
8
3
ltw_valid 플래그
ltw 오프셋
구분 비율
슬라이스 종류
DTS_next_au
1
15
2
22
4
33

도면10



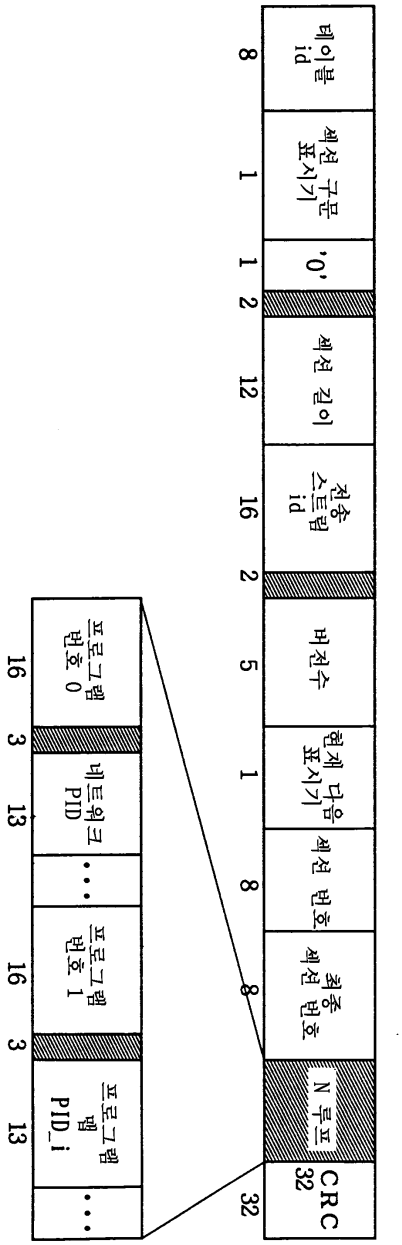
도면11

구조명	할당 PID#	설명
프로그램 연관 테이블 (PAT)	0x00	프로그램 번호 및 프로그램 맵 테이블 PID를 할당
프로그램 맵 테이블(PMT)	PAT에 의해 할당	하나 이상의 프로그램의 내용에 대해 PID를 지정
네트워크 정보 테이블(NIT)	PAT에 의해 할당	FOM 주파수 및 반복기 수와 같은 물리적인 네트워크 매개 변수
조건부 액세스 테이블(CAT)	0x01	하나 이상의(전용) EMM 스트림 각각에 고유의 PID 값을 할당

도면12

구문	비트수	니모닉
program_association section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0; i<N;i++) {		
program_number	16	uimsbf
reserved	3	bslbf
if(program number == '0')		
{		
network_PID	13	uimsbf
}		
else {		
program_map_PID	13	uimsbf
}		
}		
CRC32	32	rpchof
}		

도면13



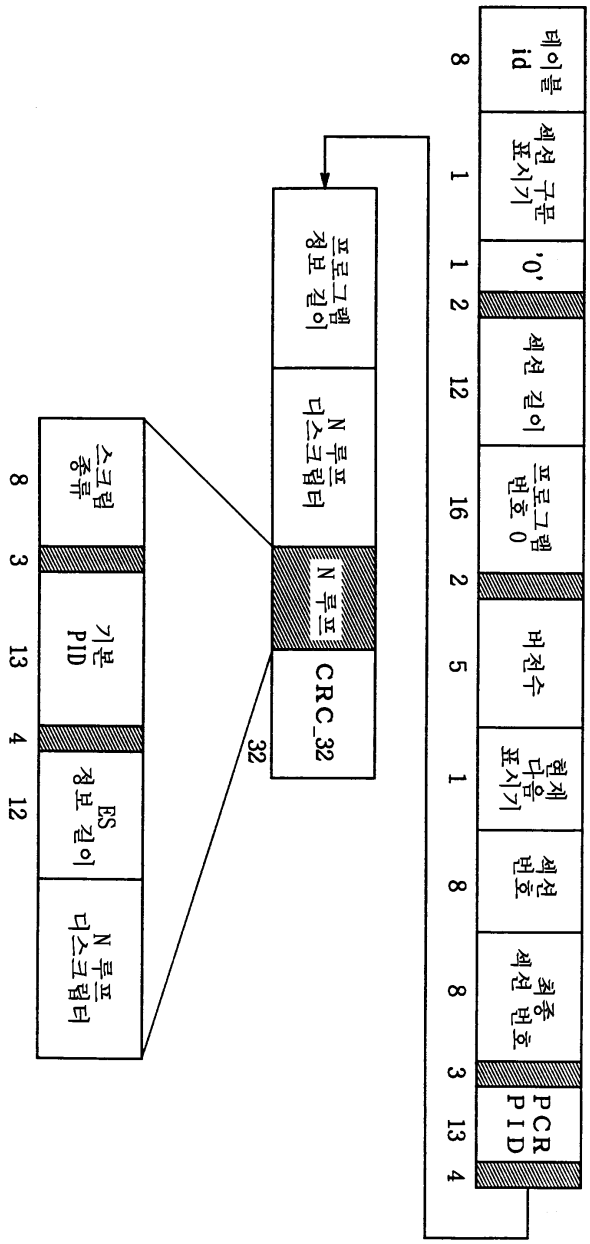
도면14

값	설명
0x00	프로그램 연관 섹션
0x01	조건부 액세스 섹션(CA 섹션)
0x02	프로그램 맵 섹션
0x03-0x3F	ITU-T 추천 H.222.0	ISO/IEC 13818 예약
0x40-0xFE	사용자 전용
0xFF	금지

도면15

구문	비트수	니모닉
TS_program_map_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
program_number	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved	3	bslbf
PCR_PID	13	uimsbf
reserved	4	bslbf
program_info_length	12	uimsbf
for(i=0; i<N;i++) {		
descriptor()		
}		
for(i=0; i<N;i++) {		
stream_type	8	uimsbf
reserved	3	bslbf
elementary_PID	13	uimsbf
reserved	4	bslbf
ES_info_length	12	uimsbf
for(i=0; i<N2; i++) {		
descriptor()		
}		
}		
CRC32	32	rpchof
}		

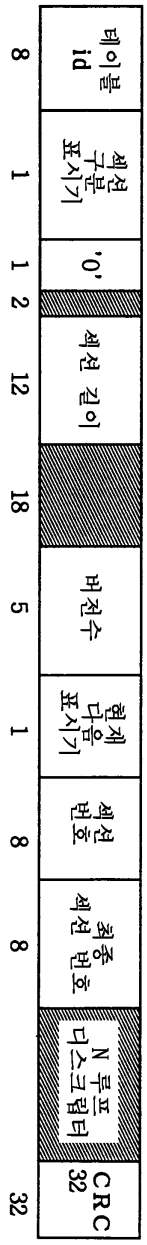
도면16



도면17

구문	비트수	니모닉
CA_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0; i<N;i++) {		
descriptor()		
}		
CRC32	32	rpchof
}		

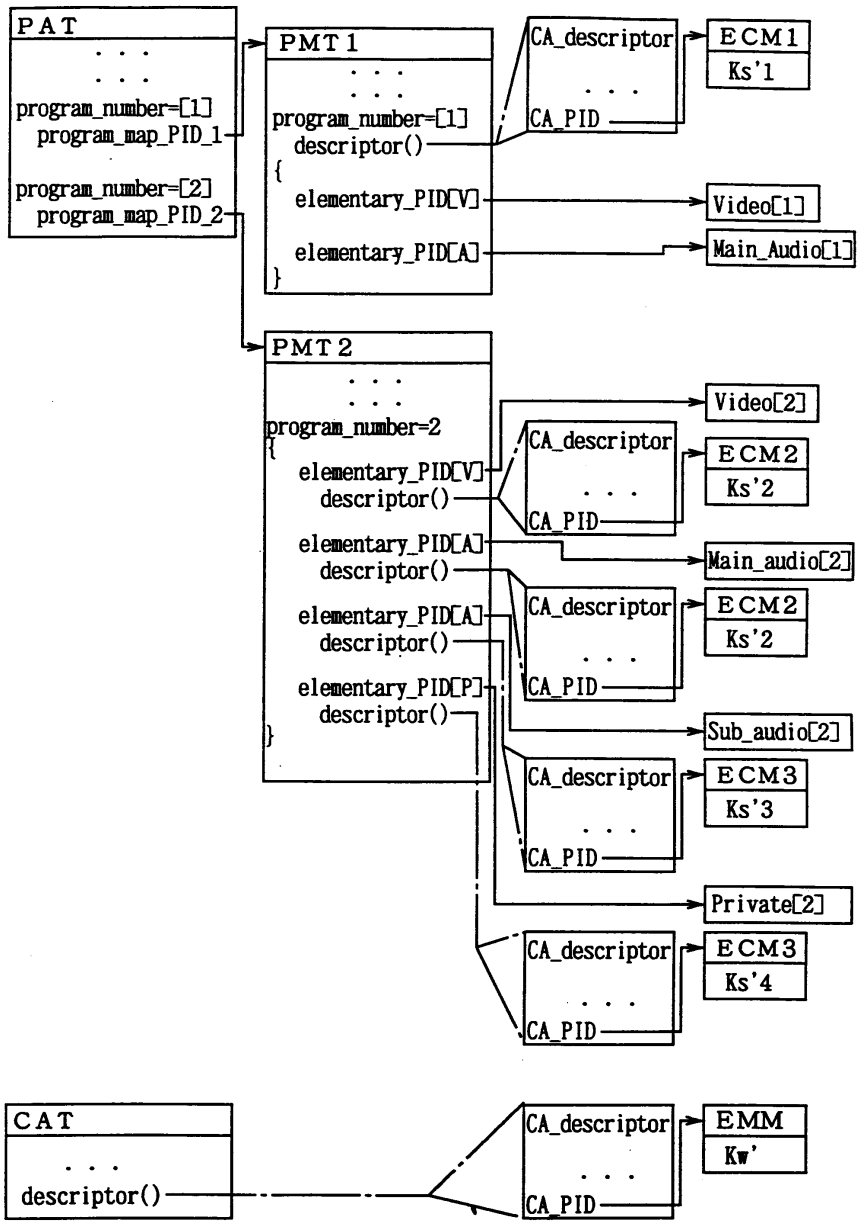
도면18



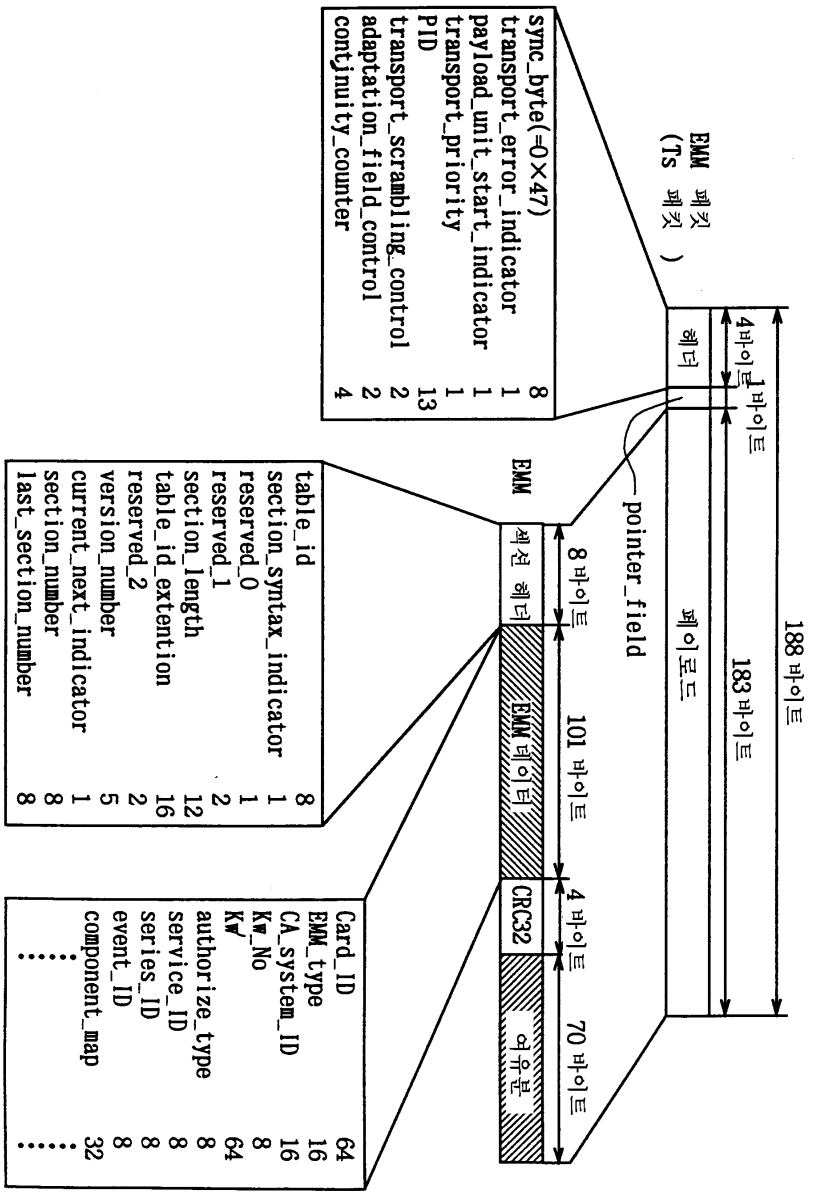
도면19

구문	비트수	니모닉
CA_descriptor() {		
descriptor_tag	8	uimbf
descriptor_length	8	uimsbf
CA_system_ID	16	uimsbf
reserved	3	bslbf
CA_PID	13	uimsbf
for(i=0; i<N;i++) {		
private_data_byte	8	uimsbf
}		
}		

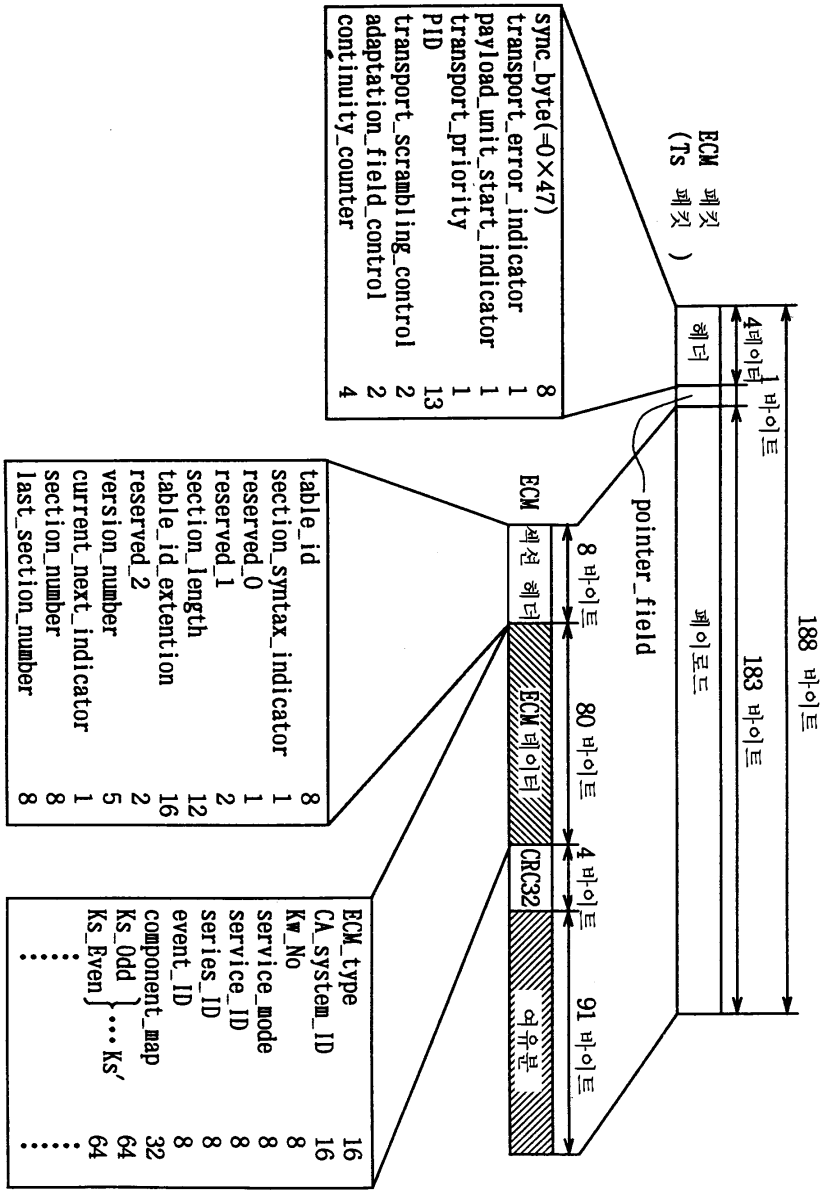
도면20


PAT
program_number=[1]
program_map_PID_1
program_number=[2]
program_map_PID_2
PMT 1
program_number=[1]
descriptor()
{
elementary_PID[V]
elementary_PID[A]
}
CA_descriptor
CA_PID
ECM1
Ks'1
Video[1]
Main_Audio[1]
PMT 2
program_number=2
{
elementary_PID[V]
descriptor()
elementary_PID[A]
descriptor()
elementary_PID[A]
descriptor()
elementary_PID[P]
descriptor()
}
Video[2]
ECM2
Ks'2
Main_audio[2]
Sub_audio[2]
ECM3
Ks'3
Private[2]
ECM3
Ks'4
CAT
descriptor()
EMM
Kw'

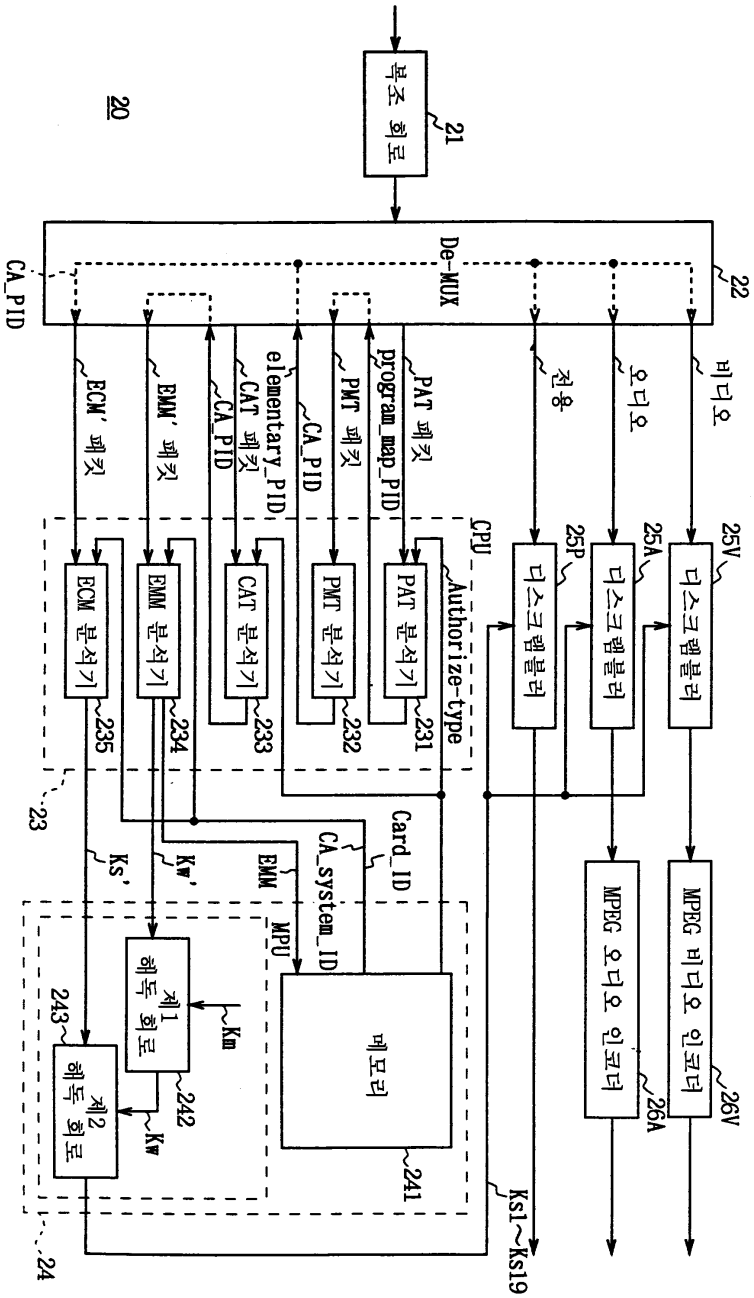
도면21


EMM 패킷
(Ts 패킷)
헤더
4바이트
1바이트
183 바이트
188 바이트
페이로드
pointer_field
EMM
섹션 헤더
8 바이트
EMM 데이터
101 바이트
CRC32
4 바이트
여유분
70 바이트
sync_byte(=0×47) 8
transport_error_indicator 1
payload_unit_start_indicator 1
transport_priority 1
PID 13
transport_scrambling_control 2
adaptation_field_control 2
continuity_counter 4
table_id 8
section_syntax_indicator 1
reserved_0 1
reserved_1 2
section_length 12
table_id_extention 16
reserved_2 2
version_number 5
current_next_indicator 1
section_number 8
last_section_number 8
Card_ID 64
EMM_type 16
CA_system_ID 16
Kw_No 8
Kw 64
authorize_type 8
service_ID 8
series_ID 8
event_ID 8
component_map 32

도면22


ECM 패킷
(Ts 패킷)
헤더
4데이터
1 바이트
183 바이트
188 바이트
페이로드
pointer_field
ECM 섹션 헤더
8 바이트
80 바이트
4 바이트
91 바이트
ECM 데이터
CRC32
여유분
sync_byte(=0×47) 8
transport_error_indicator 1
payload_unit_start_indicator 1
transport_priority 1
PID 13
transport_scrambling_control 2
adaptation_field_control 2
continuity_counter 4
table_id 8
section_syntax_indicator 1
reserved_0 1
reserved_1 2
section_length 12
table_id_extention 16
reserved_2 2
version_number 5
current_next_indicator 1
section_number 8
last_section_number 8
ECM_type 16
CA_system_ID 16
Kw_No 8
service_mode 8
service_ID 8
series_ID 8
event_ID 8
component_map 32
Ks_Odd } … Ks'
Ks_Even } 64
64

도면23


20
복조 회로
21
De-MUX
22
CA_PID
ECM' 패킷
EMM' 패킷
CAT 패킷
CA_PID
elementary_PID
PMT 패킷
CA_PID
PAT 패킷
program_map_PID
비디오
오디오
전용
CPU
Authorize-type
PAT 분석기
231
PMT 분석기
232
CAT 분석기
233
EMM 분석기
234
ECM 분석기
235
23
디스크랩블러
25V
디스크랩블러
25A
디스크랩블러
25P
MPEG 비디오 인코더
26V
MPEG 오디오 인코더
26A
Card_ID
CA_system_ID
메모리
241
Ks1~Ks19
Ks'
Kw'
EMM
MPU
제1 해독 회로
242
Km
Kw
제2 해독 회로
243
24