(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0031505
(43) 공개일자 2011년03월28일

(51) Int. Cl.
H04N 7/16 (2011.01) *H04N 5/44* (2011.01)
(21) 출원번호 10-2011-7005684(분할)
(22) 출원일자(국제출원일자) 2004년12월15일
심사청구일자 2011년03월10일
(62) 원출원 특허 10-2006-7011148
원출원일자(국제출원일자) 2004년12월15일
심사청구일자 2009년09월29일
(85) 번역문제출일자 2011년03월10일
(86) 국제출원번호 PCT/JP2004/019126
(87) 국제공개번호 WO 2005/060256
국제공개일자 2005년06월30일
(30) 우선권주장
60/530,663 2003년12월19일 미국(US)
JP-P-2003-421616 2003년12월18일 일본(JP)

(71) 출원인
파나소닉 주식회사
일본 오오사카후 가도마시 오오아자 가도마 1006 반치
(72) 발명자
구스도 다다오
일본 오오사카후 가도마시 오오아자 가도마 1006 반치 파나소닉 주식회사 내
시오미 다카카즈
일본 오오사카후 가도마시 오오아자 가도마 1006 반치 파나소닉 주식회사 내
(74) 대리인
한양특허법인

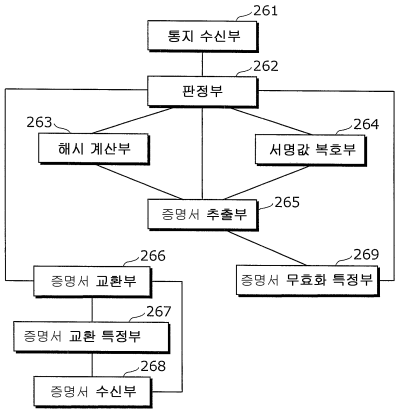
전체 청구항 수 : 총 4 항

(54) 애플리케이션 프로그램을 인증 및 실행하는 방법

(57) 요 약

종래 기술에 따르면, 프로그램이 비휘발성 메모리에 일단 저장된 후 기동되는 경우에는, 프로그램의 인증이 그러한 기동의 직전에 실행된다. 그러나, 프로그램의 기동이 개시하기 전에 암호화된 값의 복호와 같은 계산이 필요하므로, 계산에 필요한 시간에 비례하여 응답성이 저하되는 문제점이 일어난다. 이러한 문제점을 해결하기 위해, 프로그램의 인증은 그러한 프로그램이 저장되기 직전에 실행되어, 프로그램 기동 시에 인증이 실행되지 않거나 증명서의 유효성을 검증하기 위해 인증의 일부만 실행된다.

대 표 도 - 도26



특허청구의 범위

청구항 1

전송 스트림에 포함된 프로그램의 데이터 파일의 저장에 관한 정보에 따라, 상기 전송 스트림에 포함된 프로그램을 인증하여 방송 수신기에 저장하는 인증·저장 단계와, 상기 인증·저장한 프로그램을 실행하는 실행 단계를 가지는 인증 프로그램 실행 방법으로서,

상기 인증·저장 단계는,

상기 프로그램에 포함된 데이터 파일의 해시값이 상기 데이터 파일에 대응하는 해시 파일에 저장된 해시값과 일치하는지를 체크하는 제1 단계와,

상기 프로그램에 포함된 증명서 파일의 유효성을 검증하는 제2 단계와,

상기 프로그램의 증명서 파일에 포함된 리프 증명서의 공개키를 사용하여 상기 프로그램에 포함된 서명 파일의 서명값을 복호한 값과, 상기 프로그램 중 최상위 디렉토리에 위치하는 해시 파일의 해시값이 일치하는지를 검증하는 제3 단계와,

상기 제1 단계에서 해시값이 일치하고, 상기 제2 단계에서 상기 증명서 파일이 유효하다고 판정되며, 상기 제3 단계에서 해시값이 일치한다고 판정되었을 경우에 상기 프로그램을 인증하고, 상기 저장에 관한 정보에 따라 상기 프로그램의 데이터 파일을 저장하는 제4 단계를 포함하고,

상기 실행 단계는,

상기 저장한 프로그램에 포함된 증명서 파일의 유효성을 검증하는 제5 단계를 포함하고,

상기 제5 단계에서, 상기 저장한 프로그램에 포함된 증명서 파일이 유효하다고 판정되었을 경우에만 상기 저장한 프로그램을 다시 인증하고, 실행하며,

상기 제2 단계는,

상기 프로그램에 포함된 증명서 파일 중의 루트 증명서가 상기 방송 수신기 내의 루트 증명서와 일치하는지를 검증하는 제6 단계를 가지고,

상기 프로그램에 포함된 증명서 파일 중의 루트 증명서가 상기 방송 수신기 내의 루트 증명서와 일치하는 경우, 상기 증명서 파일이 유효하다고 판정하는것을 특징으로 하는 인증 프로그램 실행 방법.

청구항 2

청구항 1에 있어서,

상기 제2 단계는, 또한,

상기 프로그램에 포함된 증명서 파일 중의 증명서의 유효 기간을 체크하는 제7 단계를 가지고,

상기 프로그램에 포함된 증명서 파일 중의 루트 증명서가 상기 방송 수신기 내의 루트 증명서와 일치하고, 인증 일시가 상기 증명서 파일 중의 증명서의 유효 기간 내에 있는 경우, 상기 증명서 파일이 유효하다고 판정하는 것을 특징으로 하는 인증 프로그램 실행 방법.

청구항 3

전송 스트림에 포함된 프로그램의 데이터 파일의 저장에 관한 정보에 따라, 상기 전송 스트림에 포함된 프로그램을 인증하고, 저장하는 인증·저장 수단과, 상기 인증·저장한 프로그램을 실행하는 실행 수단을 가지는 인증 프로그램 실행 장치로서,

상기 인증·저장 수단은,

상기 프로그램에 포함된 데이터 파일의 해시값이 상기 데이터 파일에 대응하는 해시 파일에 저장된 해시값과 일치하는지를 체크하는 제1 검증부와,

상기 프로그램에 포함된 증명서 파일의 유효성을 검증하는 제2 검증부와,

상기 프로그램의 증명서 파일에 포함된 리프 증명서의 공개키를 사용하여 상기 프로그램에 포함된 서명 파일의 서명값을 복호한 값과, 상기 프로그램 중 최상위 디렉토리에 위치하는 해시 파일의 해시값이 일치하는지를 검증하는 제3 검증부와,

상기 제1 검증부에서 해시값이 일치하고, 상기 제2 검증부에서 상기 증명서 파일이 유효하다고 판정되며, 상기 제3 검증부에서 해시값이 일치한다고 판정되었을 경우에 상기 프로그램을 인증하고, 상기 저장에 관한 정보에 따라 상기 프로그램의 데이터 파일을 저장하는 기억부를 포함하고,

상기 실행 수단은,

상기 저장한 프로그램에 포함된 증명서 파일의 유효성을 검증하는 제4 검증부를 구비하고,

상기 제4 검증부에서, 상기 저장한 프로그램에 포함된 증명서 파일이 유효하다고 판정되었을 경우에만 상기 저장한 프로그램을 다시 인증하고, 실행하며,

상기 제2 검증부는,

상기 프로그램에 포함된 증명서 파일 중의 루트 증명서가 상기 방송 수신기 내의 루트 증명서와 일치하는지를 검증하는 제5 검증부를 가지고,

상기 프로그램에 포함된 증명서 파일 중의 루트 증명서가 상기 방송 수신기 내의 루트 증명서와 일치하는 경우, 상기 증명서 파일이 유효하다고 판정하는것을 특징으로 하는 인증 프로그램 실행 장치.

청구항 4

전송 스트림에 포함된 프로그램의 데이터 파일의 저장에 관한 정보에 따라, 상기 전송 스트림에 포함된 프로그램을 인증하여 방송 수신기에 저장하는 인증·저장 단계와, 상기 인증·저장한 프로그램을 실행하는 실행 단계를 컴퓨터에 실행시키는 프로그램으로서,

상기 인증·저장 단계는,

상기 프로그램에 포함된 데이터 파일의 해시값이 상기 데이터 파일에 대응하는 해시 파일에 저장된 해시값과 일치하는지를 체크하는 제1 단계와,

상기 프로그램에 포함된 증명서 파일의 유효성을 검증하는 제2 단계와,

상기 프로그램의 증명서 파일에 포함된 리프 증명서의 공개키를 사용하여 상기 프로그램에 포함된 서명 파일의 서명값을 복호한 값과, 상기 프로그램 중 최상위 디렉토리에 위치하는 해시 파일의 해시값이 일치하는지를 검증하는 제3 단계와,

상기 제1 단계에서 해시값이 일치하고, 상기 제2 단계에서 상기 증명서 파일이 유효하다고 판정되며, 상기 제3 단계에서 해시값이 일치한다고 판정되었을 경우에 상기 프로그램을 인증하고, 상기 저장에 관한 정보에 따라 상기 프로그램의 데이터 파일을 저장하는 제4 단계를 포함하고,

상기 실행 단계는,

상기 저장한 프로그램에 포함된 증명서 파일의 유효성을 검증하는 제5 단계를 포함하고,

상기 제5 단계에서, 상기 저장한 프로그램에 포함된 증명서 파일이 유효하다고 판정되었을 경우에만 상기 저장한 프로그램을 다시 인증하고, 실행하며,

상기 제2 단계는,

상기 프로그램에 포함된 증명서 파일 중의 루트 증명서가 상기 방송 수신기 내의 루트 증명서와 일치하는지를 검증하는 제6 단계를 가지고,

상기 프로그램에 포함된 증명서 파일 중의 루트 증명서가 상기 방송 수신기 내의 루트 증명서와 일치하는 경우, 상기 증명서 파일이 유효하다고 판정하는 것을 특징으로 하는 프로그램.

명 세 서

기 술 분 야

[0001] 본 발명은 다운로드된 프로그램의 신뢰성을 검증하고 신뢰할 수 있다고 검증된 프로그램을 실행하는 인증된 프로그램 실행 방법에 관한 것이다.

배 경 기 술

[0002] 프로그램을 다운로드하고 그러한 프로그램의 신뢰성을 체크/보증하는 디지털 텔레비전의 기능은 DVB-MHP 사양 "ETSI TS 101 812 V1.2.1 DVB-MHP 사양 1.0.2" 등에 기재되어 있다. 이러한 DVB-MHP 사양은, 수신되는 방송파상에 중첩된 프로그램이 부정 변경되었는지의 여부뿐만 아니라 그러한 프로그램이 신뢰할 수 있는 기구에 의해 발행되었는지의 여부를 검증하는 기능을 정의한다. 이러한 기능은 원래 필요로 했던 바와 같이 동작하지 않으므로 디지털 텔레비전에 손상을 가하는 재기입 프로그램 및 제3자를 스푸핑하는 프로그램이 기동되는 것을 방지할 수 있게 한다.

[0003] 또한, 일본 공개특허 2000-29833호 공보는 데이터를 축적 및 송신하는 서버 장치와 네트워크를 통해 데이터를 수신하는 단말 장치로 구성되어, 단말 장치에 수신된 데이터를 축적함으로써 축적된 데이터가 부정하게 사용되는 것을 방지하는 기술을 개시한다. 일본 공개특허 2000-29833호의 도 1은, 서버 장치(10)가 단말 장치(20)로부터의 요구에 응답하여 기억부(15)에 저장된 데이터를 기억부(23)로 복제하는 기술을 예시하며, 기억부(23)에 저장된 데이터가 사용되기를 원할 때, 조회부(26)가 서버 장치(10)에 조회를 행하고, 인증부(13)가 데이터의 사용에 대한 감사를 행하며, 문제가 없는 경우, 단말 장치(20)는 데이터를 사용한다. 상기 장치는 전원의 온/오프 시에도 비휘발성 메모리에 저장된 데이터의 신뢰성을 확인한 후에 데이터를 로딩할 수 있다. 프로그램 및 데이터의 신뢰성을 확인하는 것은 이하 인증이라고 한다.

[0004] 그러나, 종래의 기술에 따르면, 장치의 전원이 온/오프된 후에 그러한 프로그램을 기동시키기 위해 비휘발성 메모리로 프로그램을 1회만 저장하는 경우에, 프로그램의 인증은 프로그램이 기동된 직후에 실행된다. 이 경우에, 프로그램의 기동이 개시되기 전에 암호화된 값의 복호와 같은 계산을 실행하는 것이 필요하므로, 계산에 필요한 시간이 길어지는 만큼 응답성이 더욱 저하되는 문제점이 있다. 특히 프로그램이 빈번하게 기동되거나 프로그램의 용량이 큰 경우에, 계산량이 기동 빈도 및 용량에 비례하여 증가하기 때문에, 응답성은 더욱 더 저하된다.

발명의 내용

해결하려는 과제

[0005] 상기 문제점을 감안하여, 프로그램의 신뢰성을 보증하면서, 프로그램이 기동하기 전에 필요한 시간을 단축할 수 있는 응답성이 향상한 디지털 텔레비전과 같은 프로그램 인증 장치를 제공하는 것이 요구된다.

과제의 해결 수단

[0006] 본 발명은 프로그램이 저장되기 직전에 인증을 실행하고, 프로그램 기동 시에 인증을 실행하지 않거나 인증의 일부만을 실행함으로써, 응답성을 향상시키고 신뢰성을 보증할 수 있는 인증 프로그램 실행 방법을 제공하는 것을 목적으로 한다.

[0007] 상기 종래의 문제점을 해결하기 위해, 본 발명에 따르는 인증 프로그램 실행 방법은 전송 스트림에 포함된 프로그램을 인증하고, 상기 프로그램의 각 데이터 파일의 저장에 관한 정보에 따라 방송 수신기에 상기 인증된 프로그램을 저장하는 인증 및 기억 단계; 및 상기 인증 저장된 프로그램을 실행하는 실행 단계를 포함하고, 상기 인증 및 기억 단계는, 2개의 해시값이 일치되는지의 여부를 검증하는 단계로서, 상기 해시값 중 하나는 상기 프로그램에 포함되는 각 데이터 파일로부터 계산되고, 다른 하나의 해시값은 상기 각 데이터 파일에 대응하는 해시 파일에 저장된, 제1 단계; 상기 프로그램에 포함된 증명서 파일이 유효인지의 여부를 검증하는 제2 단계; 상기 프로그램의 상기 증명서 파일에 포함된 리프 증명서의 공개키를 사용하여 상기 프로그램에 포함된 서명 파일의 서명값을 복호함으로써 획득되는 복호값과, 상기 프로그램의 최상위 디렉토리에 위치하는 해시 파일로부터 계산되는 해시값이 일치하는지의 여부를 검증하는 제3 단계; 및 상기 2개의 해시값이 상기 제1 단계에서 일치되는 것으로 검증되고, 상기 증명서 파일이 상기 제2 단계에서 유효한 것으로 검증되며, 상기 복호값 및 상기 해시값이 상기 제3 단계에서 일치되는 것으로 검증될 것을 모두 충족하는 경우에, 상기 프로그램을 인증하고, 상기 저장에 관한 정보에 따라 상기 인증된 프로그램의 각 데이터 파일을 저장하는 제4 단계를 포함하며, 상기 실행 단

계는, 상기 저장된 프로그램에 포함된 상기 증명서 파일이 유효인지의 여부를 검증하는 제5 단계를 포함하고, 상기 실행 단계에서, 상기 저장된 프로그램은 다시 인증되어, 상기 저장된 프로그램에 포함된 상기 증명서 파일이 상기 제5 단계에서 유효하다고 검증된 경우에만 실행된다.

[0008] 따라서, 프로그램이 기동되기 전에 필요한 시간을 단축시키면서 프로그램의 신뢰성을 보증할 수 있게 된다.

[0009] 또한, 상기 프로그램이 디렉토리 구조를 갖는 경우에, 각 디렉토리에 포함되는 각 데이터 파일과 상기 각 데이터 파일에 대응하는 상기 해시 파일은 동일한 디렉토리 내에 위치되고, 상기 제1 단계는 각 디렉토리에 포함되는 각 데이터 파일에 대해 실행되어도 된다.

[0010] 따라서, 각 디렉토리에 포함되는 각 데이터 파일에 대해 상기 데이터 파일로부터 계산된 해시값과 상기 데이터 파일에 대응하는 해시 파일에 저장된 해시값이 일치하는지의 여부를 체크할 수 있게 된다.

[0011] 또한, 상기 제2 단계는, 2개의 루트 증명서가 일치하는지의 여부를 검증하는 단계로서, 상기 루트 증명서 중 하나는 상기 프로그램에 포함된 상기 증명서 파일 내에 있고, 다른 하나의 루트 증명서는 상기 방송 수신기에 설치된, 제6 단계를 포함해도 되고, 상기 제2 단계에서, 상기 2개의 루트 증명서가 일치하는 경우에, 상기 증명서 파일이 유효하다고 검증되어도 된다.

[0012] 여기에서, 상기 제2 단계는, 상기 프로그램에 포함된 상기 증명서 파일 내의 각 증명서의 유효 기간을 검증하는 제7 단계를 더 포함해도 되고, 상기 제2 단계에서, 상기 2개의 루트 증명서가 일치하고, 상기 인증이 실행되는 시간이 상기 증명서 파일 내의 각 증명서의 상기 유효 기간 내에 있을 것을 모두 충족하는 경우에, 상기 증명서 파일이 유효하다고 검증되어도 된다.

[0013] 따라서, 루트 증명서가 일치하지 않고 증명서의 유효 기간이 만료된 경우에 프로그램이 저장되는 것을 방지할 수 있게 된다.

[0014] 또한, 상기 제5 단계는, 2개의 루트 증명서가 일치하는지의 여부를 검증하는 단계로서, 상기 루트 증명서 중 하나는 상기 저장된 프로그램에 포함된 상기 증명서 파일 내에 있고, 다른 하나의 루트 증명서는 상기 방송 수신기에 설치된, 제8 단계를 포함해도 되고, 상기 제5 단계에서, 상기 2개의 루트 증명서가 일치하는 경우에, 상기 저장된 프로그램에 포함된 상기 증명서 파일이 유효하다고 검증되어도 된다.

[0015] 여기에서, 상기 제5 단계는, 상기 저장된 프로그램에 포함된 상기 증명서 파일 내의 각 증명서의 유효 기간을 검증하는 제9 단계를 더 포함해도 되고, 상기 제5 단계에서, 상기 2개의 루트 증명서가 일치하고, 상기 실행이 행해지는 시간이 상기 증명서 파일 내의 각 증명서의 상기 유효 기간 내에 있을 것을 모두 충족하는 경우에, 상기 저장된 프로그램에 포함된 상기 증명서 파일이 유효하다고 검증되어도 된다.

[0016] 따라서, 루트 증명서가 일치하지 않고 증명서의 유효 기간이 만료된 경우에 프로그램이 저장되는 것을 방지할 수 있게 된다.

[0017] 본 발명을 이상과 같은 인증된 프로그램 실행 방법으로서 구현할 수 있을 뿐만 아니라, 인증된 프로그램 실행 방법에 포함되는 특정 단계들을 그 유닛으로서 포함하고 이들 단계를 컴퓨터로 실행하게 하는 프로그램으로서 포함하는 인증된 프로그램 실행 장치로서 구현할 수도 있다. 또한, 그러한 프로그램은 CD-ROM과 같은 기록 매체로 및 인터넷과 같은 전송 매체를 통해 배포될 수 있음을 유념해야 한다.

[0018] 이상의 설명으로부터 명백해지는 바와 같이, 본 발명에 따르는 인증 프로그램 실행 방법은 프로그램의 신뢰성을 보증하면서 프로그램이 기동되기 전에 필요한 시간을 단축시킬 수 있다.

[0019] 명세서, 도면 및 청구범위를 포함하는 2003년 12월 18일에 출원된 일본 특허출원 2003-421616호의 개시물 및 2003년 12월 19일에 출원된 미국 임시 출원 60/530663호의 개시물은 참고로 본 명세서에 모두 일체화되어 있다.

[0020] 본 발명의 이들 목적, 이점 및 특징들은 발명의 특정 실시예를 예시하는 첨부하는 도면과 관련하여 행해지는 아래의 설명으로부터 명백해진다.

발명의 효과

[0021] 프로그램의 신뢰성을 보증할 뿐만 아니라 응답성을 향상시킬 수 있는 본 발명에 따르는 인증된 프로그램 실행 방법은 디지털 텔레비전 수신기의 기능을 시간적으로 향상시키는데 뿐만 아니라 그에 기능을 부가하는데도 유용하다. 또한, 본 발명은 디지털 텔레비전에 뿐만 아니라 퍼스날 컴퓨터 및 휴대 전화와 같은 소프트웨어에 의해 제어되는 정보 기기에 기능의 시간적 부가 및 기능성의 시간적 향상과 같은 사용에 적용 가능하다.

도면의 간단한 설명

[0022] 도 1은 본 발명의 제1 실시예에 따르는 케이블 텔레비전 시스템의 구성을 도시하는 도면이다.

도 2는 본 발명에 따르는 케이블 텔레비전 시스템에서의 헤드 엔드(head end)와 단말 장치들간의 통신에 사용되는 주파수 대역을 사용하는 일례를 도시하는 도면이다.

도 3은 본 발명에 따르는 케이블 텔레비전 시스템에서의 헤드 엔드와 단말 장치들간의 통신에 사용되는 주파수 대역을 사용하는 일례를 도시하는 도면이다.

도 4는 본 발명에 따르는 케이블 텔레비전 시스템에서의 헤드 엔드와 단말 장치들간의 통신에 사용되는 주파수 대역을 사용하는 일례를 도시하는 도면이다.

도 5는 본 발명에 따르는 케이블 텔레비전 시스템에서의 단말 장치의 구성을 도시하는 도면이다.

도 6은 본 발명에 따르는 케이블 텔레비전 시스템에서의 단말 장치의 외관의 일례를 도시하는 도면이다.

도 7은 본 발명에 따르는 POD(504)의 하드웨어 구성을 도시하는 도면이다.

도 8은 본 발명에 따르는 POD(504)에 저장되는 프로그램의 구성을 도시하는 도면이다.

도 9는 MPEG 표준으로 정의된 패킷의 구성을 도시하는 도면이다.

도 10은 MPEG2 전송 스트림의 일례를 도시하는 도면이다.

도 11은 입력부(513)가 정면 패널의 형태로 구성되어 있는 경우의 입력부(513)의 외관의 일례를 도시하는 도면이다.

도 12는 본 발명에 따르는 단말 장치(500)에 저장되는 프로그램의 구성을 도시하는 도면이다.

도 13A는 본 발명에 따르는 디스플레이(509)에 의해 표시되는 표시 스크린의 일례를 도시하는 도면이고, 도 13B는 본 발명에 따르는 디스플레이(509)에 의해 표시되는 표시 스크린의 일례를 도시하는 도면이다.

도 14는 본 발명에 따르는 2차 기억부(510)에 저장되는 정보의 일례를 도시하는 도면이다.

도 15A, 도 15B 및 도 15C는 본 발명에 따르는 1차 기억부(511)에 저장되는 정보의 일례를 각각 도시하는 도면이다.

도 16은 본 발명에 따르는 MPEG2 표준으로 규정된 PAT의 콘텐츠를 도시하는 개략도이다.

도 17은 본 발명에 따르는 MPEG2 표준으로 규정된 PMT의 콘텐츠를 도시하는 개략도이다.

도 18은 본 발명에 따르는 DVB-MHP 표준으로 규정된 AIT의 콘텐츠를 도시하는 개략도이다.

도 19는 본 발명에 따르는 DSMCC 포맷으로 송신되는 파일 시스템을 도시하는 개략도이다.

도 20은 본 발명에 따르는 XAIT의 콘텐츠를 도시하는 개략도이다.

도 21은 본 발명에 따르는 2차 기억부(510)에 저장되는 정보의 일례를 도시하는 도면이다.

도 22A, 도 22B 및 도 22C는 본 발명에 따르는 파일 또는 디렉토리의 해시값을 포함하는 파일의 일례를 각각 도시하는 도면이다.

도 23은 본 발명에 따르는 증명서 체인의 구성을 도시하는 도면이다.

도 24는 본 발명에 따르는 X.509 증명서의 구성을 도시하는 도면이다.

도 25는 본 발명에 따르는 서명 파일의 구성을 도시하는 도면이다.

도 26은 본 발명에 따르는 시큐리티 모듈의 구성 요소를 도시하는 도면이다.

도 27은 본 발명에 따르는 파일 시스템이 인증될 때 실행되는 동작을 도시하는 플로우차트이다.

도 28은 본 발명에 따라 프로그램 기동전 통지가 수신될 때 인증이 실행되지 않는 경우의 플로우차트이다.

도 29는 본 발명에 따르는 파일 시스템에 대해 부정 변경 확인이 실행될 때 실행되는 동작을 도시하는 플로우차트이다.

도 30은 본 발명에 따르는 서명 파일을 사용하여 부정 변경 확인이 실행될 때 실행되는 동작을 도시하는 플로우차트이다.

도 31은 본 발명에 따라 리프(leaf) 증명서와 중간 증명서 사이의 체인 관계가 확인될 때 실행되는 동작을 도시하는 플로우차트이다.

도 32는 본 발명에 따라 중간 증명서와 루트(root) 증명서 사이의 체인 관계가 확인될 때 실행되는 동작을 도시하는 플로우차트이다.

도 33은 본 발명에 따라 루트 증명서 내의 서명이 확인될 때 실행되는 동작을 도시하는 플로우차트이다.

도 34는 본 발명에 따라 저장되는 파일을 특정하는데 사용되는 파일의 일례를 도시하는 도면이다.

도 35는 본 발명에 따라 파일 시스템의 인증이 실행될 때 실행되는 동작을 도시하는 플로우차트이다.

도 36은 본 발명에 따라 프로그램 기동전 통지가 수신될 때 X.509 증명서의 유효성을 확인할 때 실행되는 동작을 도시하는 플로우차트이다.

도 37은 본 발명에 따르는 다운로드 모듈로부터 수신되는 코드 파일의 간략화된 구성을 도시하는 도면이다.

도 38A, 도 38B 및 도 38C은 본 발명에 따라 단말 장치가 소유하는 증명서가 교환되는 것을 각각 도시하는 도면이다.

도 39는 본 발명에 따라 증명서 교환이 실행될 때 실행되는 동작을 도시하는 플로우차트이다.

도 40은 본 발명에 따라 프로그램 기동전 통지가 수신될 때 루트 증명서의 비교 시에 실행되는 동작을 도시하는 플로우차트이다.

도 41은 본 발명에 따르는 CRL의 구성을 도시하는 도면이다.

도 42는 본 발명에 따르는 CRL 내의 무효화된 증명서 리스트를 도시하는 개략도이다.

도 43은 본 발명에 따르는 CRL을 포함하는 파일 시스템의 일례를 도시하는 도면이다.

도 44는 본 발명에 따르는 서명값 및 해시값에 기초하여 CRL의 유효성이 확인될 때 실행되는 동작을 도시하는 플로우차트이다.

도 45는 본 발명에 따라 루트 증명서들 사이의 비교와 증명서들 간의 체인 관계에 기초하여 CRL의 유효성이 확인될 때 실행되는 동작을 도시하는 플로우차트이다.

도 46은 본 발명에 따르는 파일 또는 디렉토리들의 해시값을 포함하는 파일의 일례를 도시하는 도면이다.

도 47은 본 발명에 따라 CRL이 프로그램 기억 시에 존재하는 경우에 인증을 실행하는 동작을 도시하는 플로우차트이다.

도 48은 CRL이 프로그램 기동 시에 존재하는 경우에 인증을 실행하는 동작을 도시하는 플로우차트이다.

도 49는 본 발명에 따르는 무효화된 증명서의 데이터베이스를 도시하는 개략도이다.

도 50은 본 발명에 따라 저장될 파일들을 특정하는데 사용되는 파일들을 포함하는 파일 시스템의 일례를 도시하는 도면이다.

도 51은 본 발명에 따라 저장될 파일들을 특정하는데 사용되는 파일의 일례를 도시하는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0023] 이하 도면을 참조하여 본 발명의 실시예들을 설명한다.

[0024] (제1 실시예)

[0025] 도면을 참조하여 본 발명에 따르는 케이블 텔레비전 시스템의 바람직한 실시예를 설명한다. 도 1은 헤드 엔드(101)와 3개의 단말 장치들, 즉, 단말 장치 A(111), 단말 장치 B(112) 및 단말 장치 C(113)인 케이블 시스템을 구성하는 장치들 사이의 관계를 도시하는 블록도이다. 본 실시예에서, 3개의 단말 장치들은 하나의 헤드 엔드에 접속되어 있지만, 임의의 수의 단말 장치(들)가 헤드 엔드에 접속되어 있는 경우에 본 발명을 실시하는 것이

가능하다.

[0026] 헤드 엔드(101)는 비디오, 오디오 및 데이터와 같은 방송 신호를 복수의 단말 장치에 송신하고, 단말 장치로부터 송신된 데이터를 수신한다. 이를 실현하기 위해, 헤드 엔드(101)와 단말 장치 A(111), 단말 장치 B(112) 및 단말 장치 C(113)와의 사이의 데이터 송신의 사용을 위해 주파수 대역이 분할된다. 도 2는 분할된 주파수 대역의 일례를 도시하는 테이블이다. 대역외(OOB로 축약)와 대역내의 크게 2 종류의 주파수 대역이 존재한다. 5~130㎒의 주파수 대역이 헤드 엔드(101)와 단말 장치 A(111), 단말 장치 B(112) 및 단말 장치 C(113)와의 사이의 데이터 교환을 위해 주로 사용되도록 OOB에 할당된다. 130㎒~864㎒의 주파수 대역은 비디오 및 오디오를 포함하는 방송 채널용으로 주로 사용되도록 대역내에 할당된다. QPSK는 OOB용으로 채용되지만, QAM64는 변조 방식으로서 대역내용으로 채용된다. 변조 방식의 상세한 설명은 본 발명과 관련이 적은 공지된 기술이므로 여기에서 생략한다. 도 3은 OOB 주파수 대역이 어떻게 사용되는지의 더욱 구체적인 예를 도시한다. 70㎒~74㎒의 주파수 대역은 헤드 엔드(101)로부터 데이터를 송신하는데 사용된다. 이 경우에, 모든 단말 장치 A(111), 단말 장치 B(112), 및 단말 장치 C(113)는 헤드 엔드(101)로부터 동일한 데이터를 수신한다. 한편, 10.0㎒~10.1㎒의 주파수 대역은 단말 장치 A(111)로부터 헤드 엔드(101)에 데이터를 송신하는데 사용된다. 10.1㎒~10.2㎒의 주파수 대역은 단말 장치 B(112)로부터 헤드 엔드(101)에 데이터를 송신하는데 사용된다. 10.2㎒~10.3㎒의 주파수 대역은 단말 장치 C(113)로부터 헤드 엔드(101)에 데이터를 송신하는데 사용된다. 따라서, 단말 장치 A(111), 단말 장치 B(112) 및 단말 장치 C(113)로부터 헤드 엔드(101)에 각 단말 장치에 고유한 데이터를 송신할 수 있게 된다. 도 4는 대역내 주파수 대역을 사용하는 하나의 예를 도시한다. 150~156㎒ 및 156~162㎒의 주파수 대역이 각각 텔레비전 채널 1 및 텔레비전 채널 2에 할당되고, 이후의 주파수들이 6㎒ 간격으로 텔레비전 채널들에 할당된다. 310㎒ 및 이후의 주파수들은 1㎒ 간격으로 라디오 채널들에 할당된다. 상기 채널들은 각각 아날로그 방송이나 디지털 방송 중 어느 하나에 사용될 수 있다. 디지털 방송의 경우에, 데이터가 MPEG2 규격에 부합하는 전송 패킷 포맷으로 송신되고, 그러한 경우에 오디오 및 비디오 데이터에 부가하여 여러 가지 데이터 방송 시스템용 데이터가 송신될 수 있다.

[0027] 헤드 엔드(101)에는 각각의 주파수 범위에 적합한 방송 신호를 송신하기 위해 QPSK 변조부, QAM 변조부 등이 설비된다. 또한, 헤드 엔드(101)에는 단말 장치로부터 데이터를 수신하기 위한 QPSK 복조부가 설비된다. 또한, 헤드 엔드(101)에는 상기 변조부들과 복조부에 관련된 여러 가지 디바이스들이 설비되는 것으로 가정된다. 그러나, 본 발명은 주로 단말 장치에 관한 것이므로, 그들의 상세한 설명은 생략한다.

[0028] 단말 장치 A(111), 단말 장치 B(112), 및 단말 장치 C(113)는 헤드 엔드(101)로부터 송신된 방송 신호를 수신하여 재생한다. 또한, 단말 장치 A(111), 단말 장치 B(112), 및 단말 장치 C(113)는 각 단말 장치에 고유한 데이터를 헤드 엔드(101)에 송신한다. 본 실시예에서는, 이들 3개의 단말 장치는 동일한 구성을 가져야 한다.

[0029] 도 5는 각 단말 장치의 하드웨어 구성을 도시하는 블록도이다. 500은 QAM 복조부(501), QPSK 복조부(502), QPSK 변조부(503), TS 디코더(505), 오디오 디코더(506), 스피커(507), 비디오 디코더(508), 디스플레이(509), 2차 기억부(510), 1차 기억부(511), ROM(512), 입력부(513) 및 CPU(514)로 구성되어 있는 단말 장치이다. 또한, POD(504)는 단말 장치(500)에 탈착 가능하다.

[0030] 도 6은 단말 장치(500)의 외관의 일례인 박형 텔레비전을 도시한다. 단말 장치는 다양한 구성으로 실현할 수 있지만, 본 실시예에서는 OpenCable(TM) 및 OCAP에 기초하여 구성되는 단말 장치가 일례로서 설명된다.

[0031] 601은 박형 텔레비전의 강철 케이스이며, 여기에 POD(504)를 제외한 단말 장치(500)의 모든 구성요소들이 내장된다.

[0032] 602는 도 5의 디스플레이(509)에 대응하는 디스플레이이다.

[0033] 603은 도 5의 입력부(513)에 대응하고 복수의 버튼으로 구성되는 정면 패널부이다.

[0034] 604는 헤드 엔드(101)로 및 로부터 신호들을 송/수신하기 위해 케이블선이 접속되는 신호 입력 단자이다. 신호 입력 단자는 도 5에 도시되는 QAM 복조부(501), QPSK 복조부(502), 및 QPSK 변조부(503)에 접속된다.

[0035] 605는 도 5의 POD(504)에 대응하는 POD 카드이다. POD(504)는 단말 장치(500)와 무관하게 구현되고, 도 6의 POD 카드(605)의 경우에서와 같이, 단말 장치(500)에 탈착 가능하다. POD(504)의 상세한 설명은 후술한다.

[0036] 606은 POD 카드(605)가 삽입되는 삽입 슬롯이다.

[0037] 도 5를 참조하면, QAM 복조부(501)는 CPU(514)에 의해 특정되는 주파수를 포함하는 튜닝(tuning) 정보에 따라

헤드 엔드(101)에서 QAM 변조되어 송신된 신호를 복조하여, 그 결과를 POD(504)에 전달한다.

[0038] QPSK 복조부(502)는 CPU(514)에 의해 특정되는 주파수를 포함하는 튜닝 정보에 따라 헤드 엔드(101)에서 QPSK 변조되어 송신된 신호를 복조하여, 그 결과를 POD(504)에 전달한다.

[0039] QPSK 변조부(503)는 CPU(514)에 의해 특정되는 주파수를 포함하는 복조 정보에 따라 POD(504)로부터 전달된 신호를 QPSK 변조하여, 그 결과를 헤드 엔드(101)에 송신한다.

[0040] 도 6에 도시된 바와 같이, POD(504)는 단말 장치(500)의 본체로부터 분리 가능하다. 단말 장치(500)의 본체와 POD(504) 사이의 접속 인터페이스의 정의는 OpenCable(TM) CableCARD(TM) Interface Specification (OC-SP-CC-IF-I15-031121) 및 이러한 사양을 참조하는 사양으로 부여된다. 이러한 사양에서의 CableCARD를 POD라고 한다. 여기에서, 상세한 설명은 생략하고 본 발명에 관련된 구성요소들에 대해서만 설명한다.

[0041] 도 7은 POD(504)의 내부 구성을 도시하는 블록도이다. POD(504)는 제1 디스크램블러(descrambler)부(701), 제2 디스크램블러부(702), 스크램블러(scrambler)부(703), 1차 기억부(704), 2차 기억부(705), 및 CPU(706)로 구성된다.

[0042] 제1 디스크램블러부(701)는 CPU(706)로부터의 지시 하에 단말 장치(500)의 QAM 복조부(501)로부터 스크램블된 신호를 수신하여, 그 신호를 디스크램블한다. 그 후, 제1 디스크램블러부(701)가 단말 장치(500)의 TS 디코더(505)에 디스크램블된 신호를 송신한다. 키와 같은 디스크램블러에 필요한 정보가 필요에 따라 CPU(706)에 의해 제공된다. 더욱 구체적으로는, 헤드 엔드(101)는 여러 개의 유료 채널을 방송하고, 사용자가 이들 유료 채널의 시청권을 구매했을 때, 제1 디스크램블러부(701)는 CPU(706)로부터 키와 같은 필요한 정보를 수신하여, 디스크램블러를 실행한다. 따라서, 사용자는 이들 유료 채널을 시청할 수 있다. 키와 같은 필요한 정보가 제공되지 않을 때, 제1 디스크램블러부(701)는 디스크램블을 실행하지 않고 수신된 신호를 TS 디코더(505)에 바로 전달한다.

[0043] 제2 디스크램블러부(702)는 CPU(706)로부터의 지시 하에 단말 장치(500)의 QPSK 복조부(502)로부터 스크램블된 신호를 수신하여, 그 신호를 디스크램블한다. 그 후, 제2 디스크램블러부(702)는 디스크램블된 신호를 CPU(706)에 전달한다.

[0044] 스크램블러부(703)는 CPU(706)로부터의 지시 하에 CPU(706)로부터 수신된 데이터를 스크램블하여, 그 결과를 단말 장치(500)의 QPSK 변조부(503)에 전송한다.

[0045] 구체적으로는 RAM과 같은 1차 메모리를 구성요소로 하는 1차 기억부(704)는 CPU(706)가 처리를 실행할 때 일시적으로 데이터를 저장하기 위한 것이다.

[0046] 구체적으로는 플래시 ROM과 같은 2차 기억 메모리를 구성요소로 하는 2차 기억부(705)는 CPU(706)에 의해 실행되는 프로그램을 저장할 뿐만 아니라 전원이 오프될 때에도 소거되지 않아야 하는 데이터를 저장하기 위한 것이다.

[0047] CPU(706)는 2차 기억부(705)에 저장된 프로그램을 실행한다. 이 프로그램은 복수의 서브 프로그램으로 구성된다. 도 8은 2차 기억부(705)에 저장된 프로그램의 일례를 도시한다. 도 8에서, 프로그램(800)은 메인 프로그램(801), 초기화 서브 프로그램(802), 네트워크 서브 프로그램(803), 재생 서브 프로그램(804), 및 PPV 서브 프로그램(805)을 포함하는 복수의 서브 프로그램으로 구성된다.

[0048] 여기에서, PPV는 Pay Per View의 약어로서, 사용자가 영화와 같은 특정 프로그램을 유료로 시청할 수 있게 하는 서비스이다. 사용자가 자신의 개인 식별 번호를 입력할 때, 사용자가 프로그램의 시청권을 구매한 사실이 헤드 엔드(101)에 통지되고, 프로그램이 디스크램블된다. 따라서, 사용자는 그 프로그램을 시청할 수 있다. 이러한 프로그램의 시청으로 인해, 사용자는 훗날 구입대금을 지불해야 한다.

[0049] 전원이 온될 때 메인 프로그램(801)에 의해 첫 번째로 기동되는 서브 프로그램인 메인 프로그램(801)은 다른 서브 프로그램을 제어한다.

[0050] 초기화 서브 프로그램(802)은 전원이 온될 때 메인 프로그램(801)에 의해 기동되어 단말 장치(500)와의 정보 교환 등을 수행하여 초기화 처리를 실행한다. 이 초기화 처리는 OpenCable(TM) CableCARD(TM) Interface Specification(OC-SP-CC-IF-I15-031121) 및 그러한 사양을 참조하는 사양에 상세히 정의된다. 또한, 초기화 서브 프로그램(802)은 이들 사양에서 정의되지 않은 초기화 처리도 실행한다. 여기에서, 그러한 초기화 처리의 일부를 소개한다. 전원이 온될 때, 초기화 서브 프로그램(802)은 단말 장치(500)의 CPU(514)를 통해 2차 기억

부(705)에 저장된 제1 주파수를 QPSK 복조부(502)에 통지한다. QPSK 복조부(502)는 제공된 제1 주파수를 사용하여 튜닝을 실행하여, 결과 신호를 2차 디스크램블러부(702)에 송신한다. 또한, 초기화 서브 프로그램(802)은 2차 기억부(705)에 저장된 제1 키와 같은 디스크램블링 정보를 2차 디스크램블러부(702)에 제공한다. 그 결과, 2차 디스크램블러부(702)는 디스크램블을 실행하고, 그 결과를 초기화 서브 프로그램(802)을 실행하는 CPU(706)에 전달한다. 따라서, 초기화 서브 프로그램(802)은 정보를 수신할 수 있다. 본 실시예에서, 초기화 서브 프로그램(802)은 네트워크 서브 프로그램(803)을 통해 정보를 수신한다. 이에 대한 상세한 설명은 후술한다.

[0051] 또한, 초기화 서브 프로그램(802)은 단말 장치(500)의 CPU(514)를 통해 2차 기억부(705)에 저장된 제2 주파수를 QPSK 변조부(503)에 통지한다. 초기화 서브 프로그램(802)은 2차 기억부(705)에 저장된 스크램블링 정보를 스크램블러부(703)에 제공한다. 초기화 서브 프로그램(802)이 네트워크 서브 프로그램(803)을 통해 스크램블러부(703)에 전송되는데 필요한 데이터를 제공할 때, 스크램블러부(703)는 제공된 스크램블링 정보를 사용하여 데이터를 스크램블하여, 스크램블된 데이터를 QPSK 변조부(503)에 제공한다. QPSK 변조부(503)는 수신한 스크램블된 정보를 변조하여, 변조된 정보를 헤드 엔드(101)에 전송한다.

[0052] 그 결과, 초기화 서브 프로그램(802)이 단말 장치(500), 2차 디스크램블러부(702), 스크램블러부(703), 및 네트워크 서브 프로그램(803)을 통해 헤드 엔드(101)와 양방향 통신을 수행할 수 있게 된다.

[0053] 메인 프로그램(801) 및 초기화 서브 프로그램(802)과 같은 복수의 서브 프로그램에 의해 사용되는 네트워크 서브 프로그램(803)은 헤드 엔드(101)와 양방향 통신을 수행하기 위한 서브 프로그램이다. 더욱 구체적으로는, 네트워크 서브 프로그램(803)은, 네트워크 서브 프로그램(803)을 사용하는 다른 서브 프로그램들이 TCP/IP에 따라 헤드 엔드(101)와 양방향 통신을 행하는 것처럼 동작한다. TCP/IP의 상세한 설명은, 복수의 단말들 사이에서 정보를 교환할 때 사용되는 프로토콜을 특정하는 공재된 기술이기 때문에, 여기에서 생략한다. 전원의 온 시에 초기화 서브 프로그램(802)에 의해 기동될 때, 네트워크 서브 프로그램(803)은, 2차 기억부(705)에 미리 저장되어 있는 POD(504)를 식별하는 식별자인 MAC(Media Access Control의 약어) 어드레스를, 단말 장치(500)를 통해 헤드 엔드(101)에 통지하여, IP 어드레스의 취득을 요구한다. 헤드 엔드(101)는 단말 장치(500)를 통해 POD(504)에 IP 어드레스를 통지하고, 네트워크 서브 프로그램(803)은 그러한 IP 어드레스를 1차 기억부(704)에 저장한다. 그 후, 헤드 엔드(101) 및 POD(504)는 POD(504)의 식별자로서 그러한 IP 어드레스를 사용하여 서로 통신한다.

[0054] 재생 서브 프로그램(804)은 2차 기억부(705)에 저장된 제2 키와 같은 디스크램블링 정보뿐만 아니라 단말 장치(500)에 의해 제공되는 제3 키와 같은 디스크램블링 정보를 제1 디스크램블러부(701)에 제공하여, 디스크램블이 실행될 수 있게 한다. 또한, 재생 서브 프로그램(804)은, 제1 디스크램블러부(701)에 입력된 신호가 PPV 채널인 것을 나타내는 정보를 네트워크 서브 프로그램(803)을 통해 수신한다. 신호가 PPV 채널이라는 통지 시에, 재생 서브 프로그램(804)은 PPV 서브 프로그램(805)을 기동시킨다.

[0055] 기동 시에, PPV 서브 프로그램(805)은 사용자에게 프로그램을 구매하도록 촉구하는 메시지를 단말 장치(500) 상에 표시하고, 사용자로부터의 입력을 수취한다. 더욱 구체적으로는, 스크린 상에 표시되기를 원하는 정보가 단말 장치(500)의 CPU(514)에 전송될 때, 단말 장치(500)의 CPU(514) 상에서 실행하는 프로그램이 단말 장치(500)의 디스플레이(509) 상에 메시지를 표시한다. 그 후, 사용자가 단말 장치(500)의 입력부(513)를 통해 개인 식별 번호를 입력할 때, 단말 장치(500)의 CPU(514)는 그 번호를 수취하여, POD(504)의 CPU(706) 상에서 실행되는 PPV 서브 프로그램(805)에 전송한다. PPV 서브 프로그램(805)은 수취된 개인 식별 번호를 네트워크 서브 프로그램(803)을 통해 헤드 엔드(101)에 전송한다. 그러한 개인 식별 번호가 유효할 때, 헤드 엔드(101)는 제4 키와 같은 디스크램블링하는데 필요한 디스크램블링 정보를 네트워크 서브 프로그램(803)을 통해 PPV 서브 프로그램(805)에 통지한다. PPV 서브 프로그램(805)은 제4 키와 같은 수취된 디스크램블링 정보를 제1 디스크램블러부(701)에 제공한 후, 제1 디스크램블러부(701)가 입력 신호를 디스크램블한다.

[0056] 도 5를 참조하면, TS 디코더(505)는 POD(504)로부터 수취된 신호에 대해 필터링을 실행하고, 필요한 데이터를 오디오 디코더(506), 비디오 디코더(508) 및 CPU(514)에 전달한다. 여기에서, POD(504)로부터 전송된 신호는 MPEG2 전송 스트림이다. MPEG2 전송 스트림에 대한 상세한 설명은 MPEG 규격 ISO/IEC138181-1에서 제공되어 있으므로, 본 실시예에서 상세히 설명하지 않는다. MPEG2 전송 스트림은 복수의 고정 길이 패킷으로 구성되며, 패킷 ID가 각 패킷에 할당되어 있다. 도 9는 패킷의 구성을 도시하는 도면이다. 900은 고정 길이 188바이트를 갖는 패킷이다. 상위 4바이트는 패킷을 식별하기 위한 정보를 저장하는 헤더(901)이고, 나머지 184바이트는 실행되기를 원하는 정보를 저장하는 페이로드(902)이다. 903은 헤더(901)의 명세를 도시한다. 패킷 ID는 제1 비트로부터 제12~제24 비트까지의 13비트에 포함된다. 도 10은 송신될 복수의 패킷열을 도시하는 개략도이다.

패킷(1001)은 헤더에 패킷 ID "1"을 갖고, 페이로드에 비디오 A의 제1 정보를 포함한다. 패킷(1002)은 헤더에 패킷 ID "2"를 갖고, 페이로드에 오디오 A의 제1 정보를 포함한다. 패킷(1003)은 헤더에 패킷 ID "3"을 갖고, 페이로드에 오디오 B의 제1 정보를 포함한다.

[0057] 패킷(1004)은 헤더에 패킷 ID "1"을 갖고, 페이로드에 비디오 A의 제2 정보를 포함하며, 이것은 패킷(1001)의 후속 정보이다. 유사하게, 패킷들(1005, 1026, 1027)은 다른 패킷들의 후속 데이터를 갖고 있다. 상기 방법으로 동일한 패킷 ID를 갖는 패킷들의 페이로드의 콘텐츠를 연결함으로써, 연속적으로 비디오 및 오디오를 재생할 수 있게 된다.

[0058] 도 10을 참조하면, CPU(514)가 패킷 ID "1" 뿐만 아니라 출력 수신지로서 "비디오 디코더(508)"를 TS 디코더(505)에 나타낼 때, TS 디코더(505)는 POD(504)로부터 수신된 MPEG2 전송 스트림에서 패킷 ID "1"을 갖는 패킷들을 추출하여, 이들을 비디오 디코더(508)에 전달한다. 따라서, 도 10에서는, 비디오 데이터만이 비디오 디코더(508)로 전달된다. 동시에, CPU(514)가 패킷 ID "2" 뿐만 아니라 "오디오 디코더(506)"을 TS 디코더(505)에 나타낼 때, TS 디코더(505)는 POD(504)로부터 수신된 MPEG2 전송 스트림에서 패킷 ID "2"를 갖는 패킷들을 추출하여, 이들을 오디오 디코더(506)에 전달한다. 도 10에서는, 오디오 데이터만이 오디오 디코더(506)로 전달된다.

[0059] 이러한 패킷 ID에 따르는 필요한 패킷들만 추출하는 처리는 TS 디코더(505)에 의해 실행되는 필터링에 대응한다. TS 디코더(505)는 CPU(514)로부터의 지시 시에 동시에 하나 이상의 필터링 처리를 실행할 수 있다.

[0060] 도 5를 참조하면, 오디오 디코더(506)는 TS 디코더(505)에 의해 제공되는 MPEG2 전송 스트림의 패킷들에 매립된 오디오 데이터를 연결하고, 연결된 데이터에 대해 디지털 아날로그 변환을 실행하며, 그 결과를 스피커(507)에 출력한다.

[0061] 스피커(507)는 오디오 디코더(506)에 의해 제공되는 신호를 오디오로서 출력한다.

[0062] 비디오 디코더(508)는 TS 디코더(505)에 의해 제공되는 MPEG2 전송 스트림의 패킷에 매립된 비디오 데이터를 연결하고, 연결된 데이터에 대해 디지털 아날로그 변환을 실행하며, 그 결과를 디스플레이(509)에 출력한다.

[0063] 구체적인 구성요소가 CRT나 액정 등인 디스플레이(509)는 비디오 디코더(508)에 의해 제공되는 비디오를 출력하고, CPU(514) 등에 의해 특정된 메시지를 표시한다.

[0064] 구체적인 구성요소가 플래시 메모리, 하드 디스크 등인 2차 기억부(510)는 CPU(514)에 의해 특정된 데이터 및 프로그램들을 저장 및 삭제한다. 저장된 데이터 및 프로그램들은 CPU(514)에 의해 참조된다. 저장된 데이터 및 프로그램들은 단말 장치(500)가 전원 오프되는 동안에도 기억부 내에 유지된다.

[0065] 구체적인 구성요소가 RAM 등인 1차 기억부(511)는 CPU(514)에 의해 특정된 데이터 및 프로그램들을 일시적으로 저장하고 그들을 삭제한다. 저장된 데이터 및 프로그램들은 CPU(514)에 의해 참조된다. 저장된 데이터 및 프로그램들은 단말 장치(500)가 전원 오프될 때 삭제된다.

[0066] ROM(512)은 그 구체적인 구성요소가 ROM, CD-ROM 및 DVD인 판독 전용 메모리 디바이스이다. ROM(512)은 CPU(514)에 의해 실행되는 프로그램을 저장한다.

[0067] 구체적인 구성요소가 정면 패널 또는 원격 제어기인 입력부(513)는 사용자로부터의 입력을 수취한다. 도 11은 정면 패널의 형태로 구성되어 있는 경우의 입력부(513)의 일례를 도시한다. 1100은 도 6에 도시된 정면 패널부(603)에 대응하는 정면 패널이다. 그러한 정면 패널(1100)은 7개의 버튼: 즉, 업커서 버튼(1101), 다운커서 버튼(1102), 레프트커서 버튼(1103), 라이트커서 버튼(1104), OK 버튼(1105), 취소 버튼(1106), 및 EPG 버튼(1107)으로 구성된다. 사용자가 버튼을 아래로 누를 때, 그 눌러진 버튼의 식별자가 CPU(514)에 통지된다.

[0068] CPU(514)는 ROM(512)에 저장된 프로그램을 실행한다. 실행될 그러한 프로그램으로부터의 지시에 따라, CPU(514)는 QAM 복조부(501), QPSK 복조부(502), QPSK 변조부(503), POD(504), TS 디코더(505), 디스플레이(509), 2차 기억부(510), 1차 기억부(511), 및 ROM(512)을 제어한다.

[0069] 도 12는 ROM(512)에 저장되어 CPU(514)에 의해 실행되는 프로그램의 구성의 일례를 도시하는 도면이다.

[0070] 프로그램(1200)은 복수의 서브 프로그램으로 구성된다. 더욱 구체적으로는, 프로그램(1200)은 OS(1201), EPG(1202), JavaVM(1203), 서비스 관리자(1204) 및 Java 라이브러리(1205)로 구성된다.

[0071] OS(1201)는 단말 장치(500)가 전원 온될 때 CPU(514)에 의해 기동되는 서브 프로그램이다. OS(1201)는 운영체

계의 약어이고, 그 일례로서는 리눅스 등이 있다. OS(1201)는 다른 서브 프로그램과 병렬로 서브 프로그램을 실행하기 위한 커널(1201a) 및 라이브러리(1201b)로 구성되는 공지된 기술의 총칭이므로, 상세한 설명은 생략한다. 본 실시예에서는, OS(1201)의 커널(1201a)이 서브 프로그램으로서 EPG(1202) 및 JavaVM(1203)을 실행한다. 한편, 라이브러리(1201b)는 이들 서브 프로그램에 단말 장치(500)의 구성요소를 제어하는데 필요한 복수의 기능을 제공한다.

[0072] 여기에서, 그러한 기능의 일례로서 튜닝을 소개한다. 튜닝으로 기능으로서, 주파수를 포함하는 튜닝 정보가 다른 서브 프로그램으로부터 수신된 후 QAM 복조부(501)에 전달된다. 따라서, QAM 복조부(501)가 제공된 튜닝 정보에 기초하여 복조를 실행항, 복조된 데이터를 POD(504)에 전달하는 것이 가능하다. 그 결과, 다른 서브 프로그램들은 라이브러리(1201b)를 통해 QAM 복조부를 제어할 수 있다.

[0073] EPG(1202)는 사용자에게 프로그램의 리스트를 표시할 뿐만 아니라 사용자로부터의 입력을 수취하기 위한 프로그램 표시부(1202a)와, 채널을 선택하기 위한 재생부(1102b)로 구성된다. 여기에서, EPG는 Electric Program Guide의 약어이다. EPG(1202)는 단말 장치(500)의 전원이 온될 때 기동된다. 기동된 EPG(1202)에서, 프로그램 표시부(1202a)는 단말 장치(500)의 입력부(513)를 통해 사용자로부터의 입력을 대기한다. 여기에서, 입력부(513)가 도 11에 도시된 정면 패널의 형태를 취하는 경우에는, 사용자가 입력부(513) 상의 EPG 버튼(1107)을 누를 때, CPU(514)에 그러한 EPG 버튼의 식별자가 통지된다. CPU(514) 상에서 실행되는 서브 프로그램인 EPG(1202)의 프로그램 표시부(1202a)는 이러한 식별자를 수취하여 프로그램 정보를 디스플레이(509) 상에 나타낸다. 도 13A 및 도 13B는 디스플레이(509) 상에 표시된 프로그램 테이블의 예들을 도시한다. 도 13A를 보라. 프로그램 정보가 디스플레이(509) 상에 격자 패턴으로 표시된다. 열(1301)은 시간 정보를 기재한다. 열(1302)은 열(1301)에 기재된 각각의 시간에 대응하는 기간 동안 방송되는 채널명 "채널 1" 및 프로그램들을 기재한다. "채널 1"에서 9:00에서 10:30까지 "뉴스 9"가 방송되고, 10:30에서 12:00까지 "영화 AAA"가 방송되는 것이 도시되어 있다. 열(1303)은 열(1302)에서와 같이, 열(1301)에 기재된 각각의 시간에 대응하는 기간 동안 방송되는 채널명 "채널 2" 및 프로그램들을 기재한다. 9:00에서 11:00까지 "영화 BBB"가 방송되고, 11:00에서 12:00까지 "뉴스 11"이 방송된다. 1330은 커서이다. 커서(1330)는 정면 패널(1100) 상의 레프트커서(1103) 또는 라이트커서(1104)를 누를 때 이동한다. 라이트커서(1104)가 도 13A에 도시된 상태로 아래로 눌러질 때, 커서(1330)는 도 13B에 도시된 바와 같이 우측을 향해 이동한다. 한편, 레프트커서(1103)가 도 13B에 도시된 상태로 아래로 눌러질 때, 커서(1330)는 도 13A에 도시된 바와 같이 좌측을 향해 이동한다.

[0074] 정면 패널(1100) 상의 OK 버튼(1105)이 도 13A에 도시된 상태로 아래로 눌러질 때, 프로그램 표시부(1202a)는 재생부(1102b)에 "채널 1"의 식별자를 통지한다. 한편, 정면 패널(1100) 상의 OK 버튼(1105)이 도 13B에 도시된 상태로 아래로 눌러질 때, 프로그램 표시부(1202a)는 재생부(1102b)에 "채널 2"의 식별자를 통지한다.

[0075] 또한, 프로그램 표시부(1202a)는 POD(504)를 통해 헤드 엔드(101)로부터 1차 기억부(511)로 표시될 프로그램 정보를 정기적으로 저장한다. 일반적으로, 헤드 엔드로부터 프로그램 정보를 얻는데는 시간이 걸린다. 그러나, 입력부(513)의 EPG 버튼(1107)을 누를 때 1차 기억부(511)에 미리 저장되어 있는 프로그램 정보를 표시함으로써 프로그램 테이블을 빠르게 표시하는 것이 가능해진다.

[0076] 재생부(1102b)는 수신된 채널의 식별자를 사용하여 채널을 재생한다. 채널 식별자와 채널 사이의 관계는 2차 기억부(510)에 의해 채널 정보로서 미리 저장된다. 도 14는 2차 기억부(510)에 저장된 채널 정보의 일례를 도시한다. 채널 정보는 테이블 형태로 저장된다. 열(1401)은 채널들의 식별자들을 기재한다. 열(1402)은 채널명들을 기재한다. 열(1403)은 튜닝 정보를 기재한다. 여기에서, 튜닝 정보는 주파수, 전송 레이트 및 부호화율과 같은 QAM 복호부(501)에 제공되는 값에 의해 표시된다. 열(1404)은 프로그램 번호들을 기재한다. 프로그램 번호들은 MPEG2 표준에 의해 정의되는 PMT를 식별하는데 사용되는 번호이다. PMT에 대한 설명은 후술한다. 각각의 라인(1411~1414)은 각 채널의 식별자, 채널명 및 튜닝 정보의 세트를 나타낸다. 라인(1411)은 식별자로서 "1", 채널명으로서 "채널 1", 튜닝 정보로서 "312㎒"의 주파수 및 프로그램 번호로서 "101"을 포함하는 세트를 기재한다. 재생부(1102b)는 채널을 재생하기 위해 서비스 관리자에게 직접 수신된 채널의 식별자를 전달한다.

[0077] 또한, 재생이 행해지는 동안 사용자가 정면 패널(1100) 상의 업커서(1101) 및 다운커서(1102)를 누를 때, 재생부(1102b)는 CPU(514)를 통해 입력부(513)로부터 사용자에 의해 그러한 누름에 대한 통지를 수신하고, 재생 중인 채널을 다른 채널로 전환한다. 먼저, 재생부(1102b)는 현재 재생 중인 채널의 식별자를 1차 기억부(511)에 저장한다. 도 15A, B 및 C는 1차 기억부(511)에 저장된 채널의 식별자의 예를 도시한다. 도 15A는 식별자 "3"이 저장되는 것을 도시하고, 도 14를 참조하여 채널명 "TV3"을 갖는 채널이 재생되는 것을 도시한다. 사용자

가 도 15A에 도시된 상태로 업커서(1101)를 누를 때, 재생부(1102b)는 도 14에 도시된 채널 정보를 참조하여, "채널 2"의 채널명을 갖는 채널의 식별자 "2"를 서비스 관리자에게 전달하여, 테이블에서의 이전의 채널인 "채널 2"의 채널명을 갖는 채널을 새롭게 재생한다. 동시에, 재생부(1102b)는 식별자를 1차 기억부(511)에 저장된 채널 식별자 "2"로 재기입한다. 도 15B는 그러한 재기입된 채널 식별자를 도시한다. 한편, 사용자가 도 15A에 도시된 상태로 다운커서(1102)를 누를 때, 재생부(1102b)는 도 14에 도시된 채널 정보를 참조하여, "TV Japan"의 채널명을 갖는 채널의 식별자 "4"를 서비스 관리자에게 전달하여, 테이블에서의 이후의 채널인 "TV Japan"의 채널명을 가는 채널을 새롭게 재생한다. 동시에, 재생부(1102b)는 식별자를 1차 기억부(511)에 저장된 채널 식별자 "4"로 재기입한다. 도 15C는 그러한 재기입된 채널 식별자를 도시한다.

[0078] JavaVM(1203)은 Java(TM) 언어로 기입된 프로그램을 순차적으로 분석 및 실행하는 Java 가상 머신이다. Java 언어로 기입된 프로그램들은 하드웨어에 의존하지 않는 바이트 코드로 공지된 중간 코드로 컴파일된다. Java 가상 머신은 그러한 바이트 코드를 실행하는 번역기이다. Java 가상 머신의 일부는 CPU(514)에 의해 해석될 수 있는 실행 가능한 형태로 바이트 코드를 번역하여 그 결과를 CPU(514)에 전달하여 그것을 실행한다. JavaVM(1203)은 커널(1201a)에 의해 특정되는 실행될 Java 프로그램에 의해 기동된다. 본 실시예에서는, 커널(1201a)은 실행될 Java 프로그램으로 서비스 관리자(1204)를 특정한다. Java 언어의 상세한 해설은 "Java 언어 사양"(ISBN 0-201-63451-1)을 포함하는 다수의 서적에서 제공된다. 따라서, 그에 대한 상세한 설명은 여기에서 생략한다. 또한, JavaVM 자체의 동작에 대한 상세한 해설은 "Java 가상 머신 사양"(ISBN 0-201-63451-X)을 포함하는 다수의 서적에서 제공된다. 따라서, 그에 대한 상세한 설명은 여기에서 생략한다.

[0079] Java 언어로 기입된 Java 프로그램인 서비스 관리자(1204)는 JavaVM(1203)에 의해 순차적으로 실행된다. 서비스 관리자(1204)가 JNI(Java Native Interface)를 통해 Java 언어로기입되지 않은 다른 서브 프로그램을 호출하거나 호출되는 것이 가능하다. JNI에 대한 해설은 "Java Native Interface"를 포함하는 다수의 서적에서 제공된다. 따라서, 그에 대한 상세한 설명은 여기에서 생략하다.

[0080] 서비스 관리자(1204)는 JNI를 통해 재생부(1102b)로부터 채널의 식별자를 수취한다.

[0081] 먼저, 서비스 관리자(1204)는 Java 라이브러리(1205) 내의 튜너(1205c)에 채널의 식별자를 전달하여, 튜닝을 요구한다. 튜너(1205c)는 2차 기억부(510)에 저장된 채널 정보를 참조하여 튜닝 정보를 획득한다. 서비스 관리자(1204)는 채널의 식별자 "2"를 튜너(1205c)에 전달한다고 가정하면, 튜너(1205c)는 도 14에 도시된 열(1412)를 참조하여 채널에 대응하는 튜닝 정보 "156㎒"를 획득한다. 튜너(1205c)는 OS(1201)의 라이브러리(1201b)를 통해 튜닝 정보를 QAM 복조부(501)에 전달한다. QAM 복조부(501)는 QAM 복조부(501)에 제공되는 튜닝 정보에 따라 헤드 엔드(101)로부터 전송된 신호를 복조하여, 그 결과 신호를 POD(504)에 전달한다.

[0082] 이어서, 서비스 관리자(1204)는 Java 라이브러리(1205) 내부의 CA(1205d)에 디스크램블링을 실행하도록 요구한다. CA(1205d)는 OS(1201) 내의 라이브러리(1201b)를 통해 디스크램블링하는데 필요한 정보를 POD(504)에 제공한다. 그러한 제공된 정보에 기초하여, POD(504)는 QAM 복조부(501)에 의해 제공되는 신호를 디스크램블하여, 결과 신호를 TS 디코더(505)에 전달한다.

[0083] 이어서, 서비스 관리자(1204)는 Java 라이브러리(1205) 내부의 JMF(1205a)에 채널의 식별자를 제공하여, 비디오 및 오디오의 재생을 요구한다.

[0084] 먼저, JMF(1205a)는 PAT 및 PMT로부터 재생될 비디오 및 오디오를 특정하기 위해 사용되는 패킷 ID를 획득한다. PAT 및 PMT는 MPEG2 전송 스트림에 포함되는 프로그램 라인업을 나타내는 MPEG2 표준에 의해 규정된 테이블이다. PAT 및 PMT는 오디오 및 비디오와 함께 MPEG2 전송 스트림에 포함되는 패킷의 페이로드로 반송된다. PAT 및 PMT의 상세한 설명에 대해 사양을 참조하라. 여기에서는, PAT 및 PMT의 개관만을 설명한다. Program Association Table의 약어인 PAT는 패킷 ID "0"의 패킷으로 반송된다. PAT를 취득하기 위해, JMF(1205a)는 OS(1201)의 라이브러리(1201b)를 통해 패킷 ID "0"과 CPU(514)를 TS 디코더(505)에 나타낸다. 그 후, TS 디코더(505)는 패킷 ID "0"에 기초하여 필터링을 실행하여, 그 결과를 CPU(514)에 전달한다. 따라서, JMF(1205a)는 PAT 패킷들을 수집할 수 있다. 도 16은 수집된 PAT 정보의 일례를 개략적으로 도시하는 테이블을 도시한다. 열(1601)은 프로그램 번호를 기재한다. 열(1602)은 패킷 ID를 기재한다. 열(1602)에 도시된 패킷 ID는 PAT를 취득하기 위해 사용된다. 각각의 라인(1611~1613)은 채널의 프로그램 번호와 그에 대응하는 패킷 ID의 쌍이다. 여기에서는, 3개의 채널이 정의된다. 라인(1611)은 프로그램 번호 "101"과 패킷 ID "501"의 쌍을 정의한다. JMF(1205a)에 제공된 채널 식별자가 "2"라고 가정하면, JMF(1205a)는 도 14의 열(1412)를 참조하여, 그러한 채널 식별자에 대응하는 프로그램 번호 "102"를 취득한 후, 도 16에 도시된 PAT 내의 라인(1612)를 참조하여, 프로그램 번호 "102"에 대응하는 패킷 ID "502"를 취득한다. Program Map Table의

약어인 PMT는 PAT에 특정된 패킷 ID를 갖는 패킷으로 반송된다. PMT를 취득하기 위해, JMF(1205a)는 OS(1201)의 라이브러리(1201b)를 통해 패킷 ID와 CPU(514)를 TS 디코더(505)에 나타낸다. 여기에서, 특정되는 패킷 ID는 "502"이다. 그 후, TS 디코더(505)는 패킷 ID "502"에 기초하여 필터링을 실행하여, 그 결과를 CPU(514)에 전달한다. 따라서, JMF(1205a)는 PMT 패킷을 수집할 수 있다. 도 17은 수집된 PMT 정보의 일례를 개략적으로 도시하는 테이블을 도시한다. 열(1701)은 스트림 유형을 기재한다. 열(1702)은 패킷 ID를 기재한다. 각각의 스트림 유형에 특정된 정보는 열(1702)에 특정된 패킷 ID를 갖는 패킷의 페이로드로 반송된다. 열(1703)은 부가 정보를 기재한다. 각각의 라인(1711~1714)은 기본 스트림으로 공지되어 있는 패킷 ID와 송신되는 정보의 유형의 쌍이다. 스트림 유형 "오디오"와 패킷 ID "5011"의 쌍인 라인(1711)은 오디오 데이터가 패킷 ID "5011"을 갖는 패킷의 페이로드에 저장되어 있는 것을 나타낸다. JMF(1205a)는 PMT로부터 재생될 비디오 및 오디오의 패킷 ID를 획득한다. 도 17을 참조하면, JMF(1205a)는 라인(1711)으로부터 오디오 패킷 ID "5011"을 획득하고, 라인(1712)으로부터 비디오 패킷 ID "5012"를 획득한다.

[0085] 그 후, JMF(1205a)는 출력 수신지로서 획득된 오디오 패킷 ID와 오디오 디코더(506)의 쌍 뿐만 아니라 출력 수신지로서 비디오 패킷 ID와 비디오 디코더(508)를 OS(1201)의 라이브러리(1201b)를 통해 TS 디코더(505)에 제공한다. TS 디코더(505)는 그러한 제공된 패킷 ID와 출력 수신지에 기초하여 필터링을 실행한다. 여기에서는, 패킷 ID "5011"을 갖는 패킷이 오디오 디코더(506)에 전달되고, 패킷 ID "5012"를 갖는 패킷이 비디오 디코더(508)에 전달된다. 오디오 디코더(506)는 제공된 패킷에 대해 디지털 아날로그 변환을 실행하여, 스피커(507)를 통해 오디오를 재생한다. 비디오 디코더(508)는 제공된 패킷에 대해 디지털 아날로그 변환을 실행하여, 디스플레이(509)에 비디오를 표시한다.

[0086] 마지막으로, 서비스 관리자(1204)는 Java 라이브러리(1205) 내의 AM(1205b)에 채널 식별자를 제공하여, 데이터 방송 재생을 요구한다. 여기에서, 데이터 방송 재생은 MPEG2 전송 스트림에 포함되는 Java 프로그램을 추출하여 그것을 JavaVM이 실행하게 하는 것을 의미한다. Java 프로그램을 MPEG2 전송 스트림에 매립하는 방식으로서, MPEG 규격 ISO/IEC138181-6에 기술되어 있는 DSMCC로 공지되어 있는 방법이 사용된다. DSMCC의 상세한 설명은 여기에서는 생략한다. DSMCC 규격은 MPEG2 전송 스트림 내의 패킷에서 컴퓨터에 의해 사용되는 파일 및 디렉토리로 구성된 파일 시스템을 인코딩하는 방법을 규정하고 있다. 실행될 Java 프로그램에 대한 정보는 AIT의 형태로 MPEG2 전송 스트림의 패킷으로 반송된다. AIT는 Application Information Table의 약어이며, 그 정의가 DVB-MHP 표준(정식으로는 ETSI TS 101 812 DVB-MHP 규격 V1.0.2로 공지됨)의 10장에 제공되어 있다.

[0087] 먼저, AIT를 취득하기 위해, AM(1205b)은 JMF(1205a)의 경우에서와 같이, PAT 및 PMT를 획득하여, AIT를 저장하는 패킷의 패킷 ID를 획득한다. 제공된 채널 식별자가 "2"이고, 도 16에 도시된 PAT와 도 17에 도시된 PMT가 송신되고 있다고 가정하면, AM(1205b)은 JMF(1205a)와 동일한 절차에 따라 도 17에 도시된 PMT를 획득한다. 이어서, AM(1205b)은 PMT로부터 스트림 유형이 "데이터"이고 부가 정보로서 "AIT"를 갖는 기본 스트림의 패킷 ID를 추출한다. 도 17에 도시된 바와 같이, 라인(1713)의 기본 스트림은 그러한 기본 스트림에 대응하며, 그에 따라 AM(1205b)은 그러한 기본 스트림으로부터 패킷 ID "5013"을 획득한다.

[0088] AM(1205b)은 OS(1201)의 라이브러리(1201b)를 통해 출력 수신지로서 CPU(514)와 AIT의 패킷 ID를 TS 디코더(505)에 제공한다. 그 후, TS 디코더(505)는 그러한 제공된 패킷 ID에 기초하여 필터링을 실행하여, 그 결과를 CPU(514)에 전달한다. 그에 따라, AM(1205b)은 AIT의 패킷을 수집할 수 있다. 도 18은 수집된 AIT 정보의 일례를 개략적으로 도시하는 테이블이다. 열(1801)은 Java 프로그램의 식별자를 기재한다. MHP 사양에 따라, 이들 식별자는 애플리케이션 ID로 정의되고, 이것은 Java 프로그램이 단말 장치(500)의 시큐리티 관리자(1205f)에 의해 인증되어야 하는 프로그램인지의 여부를 식별한다. 식별자의 값이 0x0 내지 0x3fff의 범위 내에 있을 때는 인증이 필요없지만, 식별자의 값이 0x4000 내지 0x7fff의 범위에 있을 때는 인증이 필요하다. 식별자 값이 전자의 범위 내에 있는 Java 프로그램은 "미서명 프로그램(unsigned program)"이라고 하고, 식별자 값이 후자의 범위 내에 있는 Java 프로그램은 "서명 프로그램"이라고 한다. 열(1802)은 Java 프로그램을 제어하기 위한 제어 정보를 기재한다. 제어 정보는 "autostart", "present", 및 "kill"을 포함한다. "autostart"는 단말 장치(500)가 즉시 프로그램을 자동으로 실행하는 것을 의미한다. "present"는 프로그램이 자동으로 실행되지 않는 것을 의미한다. "kill"은 프로그램이 종료되는 것을 의미한다. 열(1803)은 Java 프로그램을 DSMCC 포맷으로 포함하는 패킷 ID를 추출하는데 사용되는 DSMCC 식별자를 기재한다. 열(1804)은 Java 프로그램의 프로그램명을 기재한다. 각각의 라인(1811 및 1812)은 Java 프로그램에 대한 정보의 세트이다. 라인(1811)에 정의된 Java 프로그램은 식별자 "301", 제어 정보 "autostart", DSMCC 식별자 "1" 및 프로그램명 "a/TopXlet"의 세트이다. 라인(1812)에 정의된 Java 프로그램은 식별자 "302", 제어 정보 "present", DSMCC 식별자 "1" 및 프로그램명 "b/GameXlet"의 세트이다. 여기에서, 이들 2개의 Java 프로그램은 동일한 DSMCC 식별자를 갖는다. 이것은 2개

의 Java 프로그램이 동일한 DSMCC 방식으로 인코딩된 파일 시스템에 포함되어 있는 것을 나타낸다. 여기에서, 단지 4개의 정보만이 각각의 Java 프로그램에 대해 특정되어 있지만, 실제로는 더 많은 정보가 특정된다. 상세에 대해서는 DVB-MHP 사양을 참조하라.

[0089] AM(1205b)은 AIT로부터 "autostart" Java 프로그램을 찾아내서, 대응하는 DSMCC 식별자 및 Java 프로그램명을 추출한다. 도 18을 참조하면, AM(1205b)은 라인(1811)에서 Java 프로그램을 추출하고, DSMCC 식별자 "1" 및 Java 프로그램명 "a/TopXlet"을 획득한다.

[0090] 이어서, AM(1205b)은 AIT로부터 획득된 DSMCC 식별자를 사용하여 Java 프로그램을 DSMCC 포맷으로 저장하는 패킷의 패킷 ID를 PMT로부터 획득한다. 더욱 구체적으로는, AM(1205b)은 스트림의 유형이 "데이터"이고 부가 정보 내의 DSMCC 식별자가 일치하는 기본 스트림에 포함되는 패킷 ID를 PMT로부터 획득한다.

[0091] 여기에서, 그러한 DSMCC 식별자가 "1"이고 PMT가 도 17에 도시된 것이라고 가정하면, 라인(1714) 내의 기본 스트림은 상기 조건을 충족시킨다. 따라서, 패킷 ID "5014"가 추출될 것이다.

[0092] AM(1205b)은 OS(1201)의 라이브러리(1201b)를 통해 데이터가 DSMCC 포맷으로 내장되어 있는 패킷의 패킷 ID와 출력 수신지로서의 CPU(514)를 TS 디코더(505)에 지정한다. 여기에서, 패킷 ID "5014"가 제공된다. 그 후, TS 디코더(505)는 제공된 패킷 ID에 기초하여 필터링을 실행하여, 그 결과를 CPU(514)에 전달한다. 그에 따라, AM(1205b)은 필요한 패킷을 수집할 수 있다. AM(1205b)은 DSMCC 방식에 따라 수집된 패킷으로부터 파일 시스템을 재구성하여, 재구성된 파일 시스템을 1차 기억부(511)에 저장한다. MPEG2 전송 중의 패킷으로부터 파일 시스템과 같은 데이터를 추출하여 추출된 데이터를 1차 기억부(511)와 같은 기억부에 저장하는 프로세스를 이하 다운로드라고 한다.

[0093] 도 19는 다운로드된 파일 시스템의 일례를 도시한다. 도 19에서, 원들은 디렉토리를 나타내고, 사각형은 파일을 나타내며, 여기에서 1901은 루트 디렉토리이고, 1902는 디렉토리 "a"이며, 1903은 디렉토리 "b"이고, 1904는 파일 "TopXlet. class"이며, 1905는 파일 "GameXlet. class"이다.

[0094] 다음에, AM(1205b)은 1차 기억부(511)에 다운로드된 파일 시스템 중에서 실행될 Java 프로그램을 JavaVM(1203)에 전달한다. 여기에서, 실행될 Java 프로그램명이 "a/TopXlet"이라고 가정하면, 상기 Java 프로그램명에 ". class"를 첨부한 결과인 파일 "a/TopXlet. class"가 실행될 파일이다. "/"은 디렉토리와 파일명 사이의 경계 기호이고, 도 19에 도시된 바와 같이, 파일(1904)은 실행될 Java 프로그램이다. 이어서, AM(1205b)은, Java 프로그램의 식별자를 기재하는 열(1801)이 미서명 프로그램을 나타내기 때문에, 그러한 Java 프로그램의 인증을 행하기 위해 시큐리티 관리자(1205f)를 요구할 필요가 없다는 것을 의미하는, 파일(1904)을 JavaVM(1203)에 전달한다.

[0095] JavaVM(1203)은 그러한 수신된 Java 프로그램을 실행한다.

[0096] 다른 채널의 식별자의 수신 시에, 서비스 관리자(1204)는 Java 라이브러리(1205)에 포함되는 각 라이브러리를 통해 실행되고 있는 비디오 및 오디오의 재생 뿐만 아니라 Java 프로그램의 실행을 동일한 Java 라이브러리(1205)에 포함되는 각 라이브러리를 통해 종료한 후, 새롭게 수신된 채널 식별자에 기초하여 비디오 및 오디오의 재생 뿐만 아니라 Java 프로그램의 실행을 행한다.

[0097] Java 라이브러리(1205)는 ROM(512)에 저장된 복수의 Java 라이브러리의 집합이다. 본 실시예에서는, Java 라이브러리(1205)는 JMF(1205a), AM(1205b), 튜너(1205c), CA(1205d), POD Lib(1205e), 시큐리티 관리자(1205f), 다운로드 모듈(1206) 등을 포함한다.

[0098] 서비스 관리자(1204) 및 다운로드 모듈(1206)은 Java 라이브러리(1205)에 포함된 POD Lib(1205e)를 통해 헤드 엔드(101)와의 양방향 통신을 실행한다. 이러한 양방향 통신은 OS(1201)의 라이브러리(1201b) 및 POD(504)를 통해 QPSK 복조부(502) 및 QPSK 변조부(503)를 사용하여 POD Lib(1205e)에 의해 실현될 수 있다.

[0099] 다운로드 모듈(1206)은 이러한 통신을 통해 헤드 엔드(101)로부터 코드 데이터를 수신할 수 있다. 코드 데이터는 단말 장치(500)의 펌웨어 및/또는 X.509 증명서를 포함하는 이진 데이터를 참조한다. 도 37은 본 발명에 관한 일부만을 기재하는 코드 데이터를 도시하는 개략도이다. 코드 데이터(37)를 수신할 때, 다운로드 모듈(1206)은 포함되어 있는 경우 루트 증명서(371)를 추출하여, 시큐리티 관리자(1205f)에 전달한다. 372는 펌웨어와 같은 다른 데이터를 나타낸다.

[0100] AM(1205b)은, 단말 장치(500)가 2차 기억부(510)에 저장해야 하는 Java 프로그램에 관한 정보를 헤드 엔드(101)로부터 수신한다. 그러한 정보는 XAIT 정보라고 한다. XAIT 정보는 헤드 엔드(101)와 POD(504) 사이에서 임

의의 형태로 송신된다. 본 발명은 XAIT로 필요한 정보가 포함되는 한, 송신 포맷에 무관하게 실행될 수 있다.

[0101] 도 20은 헤드 엔드(101)로부터 획득된 XAIT 정보의 일례를 개략적으로 도시하는 테이블을 도시한다. 열(2001)은 Java 프로그램의 식별자를 기재한다. 열(2002)은 Java 프로그램을 제어하기 위한 제어 정보를 기재한다. 제어 정보는 "autostart" 및 "present"를 포함한다. "autostart"는 단말 장치(500)가 전원이 온될 때 프로그램이 자동으로 실행되는 것을 의미하고, "present"는 프로그램이 자동으로 실행되지 않는 것을 의미한다. 열(2003)은 Java 프로그램을 DSMCC 포맷으로 포함하는 패킷 ID를 추출하는데 사용되는 DSMCC 식별자를 기재한다. 열(2004)은 Java 프로그램의 프로그램명을 기재한다. 열(2005)은 Java 프로그램의 우선도를 기재한다. 각각의 라인(2011 및 2012)은 각각의 Java 프로그램에 대한 정보의 세트이다. 라인(2011)에 정의된 Java 프로그램은 식별자 "0x7001", 제어 정보 "autostart", DSMCC 식별자 "1", 및 프로그램명 "a/PPV1X1et"의 세트이다. 이 Java 프로그램이 서명된 프로그램인 것은 그 Java 프로그램 애플리케이션 ID로부터 알 수 있다. 여기에서는, 각각의 Java 프로그램에 대해 5개의 정보만이 특정되어 있지만, 본 발명은 더 많은 정보가 정의될 때에도 실행될 수 있다.

[0102] XAIT 정보의 수취 시에, AM(1205b)은 AIT 정보로부터 Java 프로그램을 다운로드하는 절차와 동일한 절차에 따라, MPEG2 전송 스트림으로부터의 파일 시스템을 1차 기억부(511)에 저장한다. 이 후에, AM(1205b)은 파일 시스템을 2차 기억부(510)에 저장하기 직전에 시큐리티 관리자(105f)에게 기억전(pre-storage) 통지를 행한다. 이 때, 본 발명에 따라 시큐리티 관리자(1205f)에 의해 인증 동작이 개시되지만, 그에 대한 상세한 설명은 후술한다. 시큐리티 관리자(1205f)로부터 기동이 가능한 것이 통지될 때, AM(1205b)은 파일 시스템을 2차 기억부(510)에 저장한다. 이어서, AM(1205b)은 다운로드된 파일 시스템의 기억 위치와 XAIT 정보를 관련시킨 결과를 2차 기억부(510)에 저장한다. 도 21은 2차 기억부(510)에 서로 관련되어 저장되어 있는 다운로드된 파일 시스템과 XAIT 정보의 일례를 도시한다. 여기에서, OCAP 사양에 정의된 파일이 일례로서 기재된다. 도 20의 것과 동일한 도 21의 구성요소는 서로 동일하므로, 그러한 구성요소의 설명은 생략한다. 열(2101)은 다운로드된 파일 시스템의 기억 위치를 저장한다. 도 21에서, 그러한 기억 위치는 화살표로 나타낸다. 2110은 다운로드된 파일 시스템으로, 여기에는 최상위 디렉토리(2111), 디렉토리 "a"(2112), 디렉토리 "b"(2113), 파일 "PPV1X1et. class"(2114), 파일 "ocap. hashfile"(2116~2118), 파일 "ocap. certificate. 1"(2119) 및 파일 "ocap. signaturefile. 1"(2120)이 포함된다.

[0103] 파일(2116~2118)은 파일명이나 디렉토리명과 대응하는 해시값이 포함되는 해시 파일이다. 도 22(a), (b) 및 (c)는 "ocap. hashfiles"의 세부를 도시하는 개략도이다. 도 22(a)의 221은 "ocap. hashfile"(2116)을 도시하고, 도 22(b)의 222는 ""ocap. hashfile"(2117)을 도시하며, 도 22(c)의 223은 "ocap. hashfile"(2118)을 도시한다. "/" 디렉토리(2111)에 존재하는 221의 "ocap. hashfile"은 "ocap. certificate. 1" 파일(2119), "ocap. signaturefile. 1" 파일(2120), 동일한 디렉토리(2111) 내에 존재하는 "a" 디렉토리(2112) 및 "b" 디렉토리(2113)를 열(2211) 내에 포함한다. 열(2212)은 열(2213) 내에 기재된 각 값을 계산하는데 해시 알고리즘이 사용된 것을 나타낸다. 열(2211) 내의 파일이나 디렉토리와 관련된 열(2213)은 열(2212) 내에 특정된 해시 알고리즘의 사용에 의해 계산된 해시값을 포함한다. 현재, 주로 사용되는 해시 알고리즘은 SHA1(Secure Hash Algorithm 1) 및 MD5(Message Digest 5)이다. 이들은 이하의 특징: 즉, 변환된 후에는 원래의 데이터를 예측하는 것이 불가능하고, 파일이 파괴되거나 부정 변경되었는지를 체크하는데 사용되는 특징을 갖는 고정 길이 바이트값으로 임의의 길이를 갖는 데이터를 변환하는 공지된 알고리즘이다. 한편, 해시값은 해시 알고리즘의 사용에 의해 생성되는 의사 난수이다. 해시 알고리즘이 SHA1일 때, 해시값의 길이는 20바이트인 반면에, 해시 알고리즘이 MD5일 때, 해시값의 길이는 16바이트로 변환된다. SHA1 및 MD5에 대한 상세는 "FIPS-PUB 186-2 Secure Hash Standard" 및 "IETF RFC1321"을 각각 참조하라. 여기에서, 열(2211) 내에 기재된 각각의 디렉토리 "a" 및 "b"에 대응하는 해시값은 "a" 디렉토리에 존재하는 "ocap. hashfile" 파일(2117) 및 "b" 디렉토리에 존재하는 "ocap. hashfile" 파일(2118)에 대해 계산된 SHA1 해시값이다.

[0104] 221의 "ocap. hashfile"의 경우에서와 같이, 222의 "ocap. hashfile"은 동일한 디렉토리(2112) 내에 존재하는 "PPV1X1et. class" 파일(2114)의 파일명, 해시 알고리즘 및 해시값을 포함한다. 유사하게, 223에는 동일한 디렉토리(2113) 내에 존재하는 "PPV2X1et. class" 파일(2115)의 파일명, 해시 알고리즘 및 해시값이 포함된다.

[0105] 여기에서는, 본 발명에 관련된 속성만이 기재되어 있으므로, OCAP 사양 "OpenCable(TM) Application Platform Specification OCAP 1.0 Profile(OC-SP-OCAP1.0-IF-I09-031121)"이 "ocap. hashfile"에 대한 상세한 설명을 위해 참조되어야 한다.

[0106] 파일(2119)은 증명서 체인이다. 도 23은 "ocap. certificate. 1" 파일(2119)의 상세한 구성을 도시하는 도면

이다. "ocap. certificate. x"(x는 양의 정수)의 일반적인 구성을 도시하는 231은 루트 증명서(2311), 중간 증명서(2312) 및 리프 증명서(2313)을 포함한다. 이들은 예컨대, 루트 증명서(2311)의 소유자가 중간 증명서(2312)를 발행하고, 중간 증명서(2312)의 소유자가 리프 증명서(2313)를 발행하는 체인 관계에 있다. OCAP 사양에 따라, 서명 파일 "ocap. signaturefile. x"와 관련된 증명서 체인은 동일한 값 "x"를 갖는 "ocap. certificate. x"이다. 도 21의 경우에는, "ocap. signaturefile. 1"에 대응하는 증명서 체인은 "ocap. certificate. 1"이다. 또한, 루트 증명서(2311), 중간 증명서(2312) 및 리프 증명서(2313)는 동일한 X.509 증명서 포맷으로 구성된다. X.509 증명서는 IYU-T의 권고안으로서, 증명서 표현 포맷의 디팩토(de facto) 표준으로 정보 및 통신 산업의 여러 분야에서 널리 사용된다. 도 23에는, 3개의 증명서만이 도시되어 있지만, 복수의 중간 증명서가 존재하는 경우가 있다. 그러나, 이 경우에는, 이들 중간 증명서는 서로 관련되어 있는 체인 상태에 있어야 한다.

[0107] 도 24는 X.509 증명서의 구성을 도시하는 도면이다. 여기에서는, 본 발명을 설명하는데 필요한 속성들만 도시되어 있다. X.509 증명서에 대한 상세한 설명을 위해, IETF RFC3280 "Internet X.509 Public Key Infrastructure Certificate and CRL Profile"을 참조하라. 241은 X.509 증명서의 속성 영역을 나타내고, 242는 X.509 증명서의 서명값을 나타낸다. 일련 번호(2411)는 증명서를 식별하기 위한 번호를 나타내고, 서명 알고리즘(2412)은 서명값(242)을 판정하는데 사용되는 알고리즘을 나타내며, 금회 갱신 일시(2413)는 이 X.509 증명서가 유효로 될 때의 일시를 나타내고, 다음회 갱신 일시(2414)는 이 X.509 증명서가 만료될 때의 일시를 나타내며, 발행자명(2415)은 이 X.509 증명서를 발행한 기관명이고, 주체자명(2416)은 이 X.509 증명서의 소유자를 나타내며, 공개키(2417)는 주페자명(2416)의 공개키를 나타내고, 서명값(242)은 이 X.509 증명서의 발행자의 비밀키로 서명된(암호화된) 값을 나타낸다. 본 실시예에서는, 금회 갱신 일시(2413)와 다음회 갱신 일시(2414)는 일시의 정보를 필요로 하지만, 금회 갱신 일시(2413) 및 다음회 갱신 일시(2414)는 항상 시간 정보를 필요로 하지는 않는다. 공개키 및 비밀키를 이용하는 시스템으로서, 공개키 암호 시스템이 전자 상거래 등에 널리 사용되고 있다. 공개키 암호 시스템에서, 암호화된 텍스트는 평문을 암호화하는데 사용된 키와 상이한 키로 복호화된다. 암호화용 키와 복호용 키가 상이하기 때문에, 복호용 키로부터 암호화용 키를 추정하는 것이 불가능하다. 이러한 암호화용 키는 비밀키에 상당하고, 이러한 복호용 키는 공개키에 상당한다. 공개키 암호 시스템의 대표적인 예들은 RSA(Rivest-Shamir-Adleman) 및 DSA(Digital Signature Standard)를 포함한다.

[0108] 파일(2120)은 서명 파일이다. 도 25는 "ocap. signaturefile. 1" 파일(2120)을 도시하는 개략도이다. 251은 어떤 X.509 증명서가 관련되어 있는지를 식별하기 위한 증명서 식별자를 나타내고, 252는 해시 서명 알고리즘을 나타내며, 253은 252에 나타낸 해시 서명 알고리즘의 사용에 의해 "ocap. hashfile"(2116)로부터 계산된 서명값을 나타낸다.

[0109] Java 프로그램이 2차 기억부(510)에 저장되면, 단말 장치(500)의 전원 오프 및 채널 변경과 같은 원인으로 인해 1차 기억부(511)로부터 Java 프로그램이 삭제된 경우에도, AM(1205b)이 도 20에 도시된 XAIT를 수신하는 한 다운로드를 대기할 필요없이 그러한 Java 프로그램을 기동하는 것이 가능하다. 다시 말하면, 도 20에서, 프로그램 "/a/PPV1Xlet"의 제어 정보(2002)는 "autostart"이다. 따라서, 도 21의 2011에서, "/a/PPV1Xlet"에 대응하는 파일 시스템의 기억 위치(2101)에 대한 탐색이 이루어진 후 파일(2114)이 JavaVM(1203)에 전달될 때, 그러한 파일 시스템에 저장된 Java 프로그램 "PPV1Xlet"가 기동된다.

[0110] 이어서, 본 발명의 주요한 기능인 시큐리티 관리자(1205f)에 대해 설명한다.

[0111] 시큐리티 관리자(1205f)는 도 20의 2004에 나타낸 "/a/PPV1Xlet" 및 "/b/PPVXlet2"가 저장되려고 하는 것을 나타내는 기억전 통지를 서비스 관리자(1204)로부터 수신한다. 그러한 통지의 수취 시에, 시큐리티 관리자(1205f)는 미서명 프로그램인지 서명된 프로그램인지를 판정하기 위해 Java 프로그램 식별자(2001)의 값을 체크한다. 여기에서, Java 프로그램은 서명된 프로그램이기 때문에, 시큐리티 관리자(1205f)는 "/" 디렉토리보다 하위의 파일 시스템의 인증을 실행한다. 파일 시스템을 검증하기 위해, 도 21에 도시된 ocap. hashfiles(2116~2118), ocap. certificate.1(2119) 및 ocap.signaturefile.1(2120)을 사용하여 인증이 실행된다.

[0112] 도 26은 파일 시스템의 인증을 실행하는 시큐리티 관리자(1205f)의 구성요소를 도시한다.

[0113] 통지 수신부(261)는 AM(1205b)이 파일 시스템을 저장하려고 하기 직전에 기억전 통지를 수신하거나 그러한 사실을 판저부(262)에 통지하기 위한 것이다.

[0114] 판정부(262)는 인증 결과를 판정한다. 판정부(262)는 해시값을 수신하기 위해 파일 시스템에 대한 해시 계산을 행하도록 해시 계산부(263)에 요구한다. 판정부(262)는 "ocap. hashfile" 파일에 존재하는 해시값(2213, 2223

및 2233) 중에서 비교될 값을 추출하여, 그 값이 수신된 해시값과 일치하는지의 여부를 체크한다. 그들 값이 일치하지 않는 경우, 판정부(262)는 부정 변경이 있었다고 판정하고, 인증은 실패로 종료한다.

[0115] 또한, 판정부(262)는 증명서 추출부(265)를 사용하여 각각의 X.509 증명서를 추출하고, 현재 시간이 각각의 X.509 증명서의 금회 갱신 일시(2413) 이전이 아니고 각각의 X.509 증명서의 다음회 갱신 일시(2414)의 이후가 아닌지를(즉, 현재 시간이 각각의 X.509 증명서의 금회 갱신 일시(2413)와 다음회 갱신 일시(2414) 사이에 있는지를) 판정한다. 현재의 일시는 OS(1201)의 라이브러리(1201b)로부터 획득된다. 유효 기간이 "금회 갱신 일시＜현재 일시＜다음회 갱신 일시"를 충족하지 않으면, 판정부(262)는 인증이 실패라고 판정한다.

[0116] 또한, 증명서 체인을 인증하기 위해, 판정부(262)는 각각의 X.509 증명서의 속성 영역(241)에 대한 해시 계산을 행하도록 해시 계산부(263)에 요구한다. 그 후, 판정부(262)는 각각의 X.509 증명서에 포함된 서명값(242)을 복호하기 위한 계산을 행하도록 서명값 복호부(264)에 요구하고, 그 결과적인 복호값을 해시값 계산부(263)에 의해 획득된 해시값고 비교하여 증명서 체인의 상태를 체크한다. 이들 값이 일치하지 않는 경우, 증명서들이 체인 관계가 아닌 것을 의미하고, 따라서 인증이 실패한 것으로 판정된다. 한편, 그들 값이 일치하고 증명서들이 체인 관계에 있다고 검증되었을 때, 증명서 체인 내의 루트 증명서가 단말 장치(500)의 2차 기억부(510)에 포함되는지의 여부가 체크된다. 포함되지 않으면, 판정부(262)는, 비교를 실행하는 것이 불가능하다는 것과 관련하여, 인증이 실패인 것으로 판정한다.

[0117] 판정부(262)는 이하의 요건, 즉 (1) 부정 변경이 없었고, (2) 기간 유효성이 있으며, (3) 증명서들이 체인 관계에 있고, (4) 루트 증명서들이 일치하는 요건이 모두 충족될 때, 인증이 성공인 것으로 판정한다.

[0118] 판정부(262)에 의해 각각의 파일의 해시값을 계산하도록 요구했을 때, 해시 계산부(263)는 그들에 대한 해시 계산을 실행하기 위해 OS(1201)의 라이브러리(1201b)로부터 각각의 파일을 추출하여, 결과값을 판정부(262)에 전달한다. 또한, 해시 계산부(263)는 증명서 추출부(265)로부터 증명서 체인(231) 내의 각각의 X.509 증명서를 획득하여, 그들의 각각의 속성 영역(241)에 대한 해시 계산을 실행한다.

[0119] 서명값 복호부(264)는 판정부(262)에 의해 각 X.509 증명서 또는 "ocap.signaturefile.x" 중 하나의 서명값을 복호하기 위한 계산을 실행하도록 요구된다. 각 X.509 증명서의 서명을 복호하기 위한 계산을 실행할 때, 서명값 복호부(264)는 증명서 추출부(265)로부터 증명서 체인(231) 내의 각각의 X.509 증명서를 획득한 후, 그들의 각각의 서명을 복호하기 위한 계산을 실행하여, 그 결과를 판정부(262)에 반환한다.

[0120] 증명서 추출부(265)는 판정부(262), 해시 계산부(263) 및 서명값 복호부(264)에 의해 증명서 체인(231) 내의 각각의 X.509 증명서를 추출하도록 요구되고, X.509 증명서를 추출하여 반환한다.

[0121] 도 27은 파일 시스템의 인증을 실행할 때 시큐리티 관리자(1205f)에 의해 실행되는 동작을 요약한 플로우차트이다. 이 플로우차트에 기초하여, 파일 시스템이 도 21에 도시된 구성을 갖는 경우에 실행되는 동작에 대해 설명한다. AM(1205b)으로부터 파일 시스템에 대한 기억전 통지를 수취할 때(단계 S271), 시큐리티 관리자(1205f)는 파일 시스템의 최상위 레벨 "/" 디렉토리보다 하위의 파일 시스템에 대한 부정 변경 체크를 행한다(단계 272). 부정 변경 체크 시에, 해시값들을 비교함으로써, 파일 시스템의 각 디렉토리 내에 존재하는 파일에 변조나 변경이 없는 것이 검증된다.

[0122] 도 29 및 도 30은 단계 S272의 상세한 플로우차트이다. 먼저, 단계 S291에 도시된 바와 같이, "/" 디렉토리에 존재하는 각각의 디렉토리 "a" 및 "b"와, 각각의 파일 "ocap.certificate.1" 및 "ocap.signaturefile.1"에 대해 해시값이 계산된다. 디렉토리 "a" 및 "b"의 해시값은 각각 "/a/ocap.hashfile" 파일(222) 및 "/b/ocap.hashfile" 파일(223)로부터 계산된다. 단계 S293에서, 단계 S292에서 계산된 해시값들이 "/ocap.hashfile" 내의 2213에 기재된 해시값들과 비교된다. 단계 S294에서, 계산된 해시값들 중 어느 하나가 2213 내의 해시값들과 상이한 경우, 부정 변경된 것으로 판정된다(단계 S297). 한편, 계산된 해시값이 모두 2213 내의 해시값들과 일치할 때, 단계 S295로 전이가 행해진다. 단계 S295에서, 부정 변경 체크가 완료되지 않은 서브디렉토리가 존재하는지의 여부가 체크된다. 현 단계에서, 디렉토리 "a" 및 "b"는 "/" 디렉토리의 서브디렉토리로 존재하며, 이들에 대해서는 아직 부정 변경 체크가 실행되지 않았다. 따라서, 이들 디렉토리 "a" 및 "b"에 대해 부정 변경 체크가 실행될 필요가 있다. 먼저, 단계 S296에서 "a" 디렉토리에 주목하고, 여기에서 "/" 디렉토리에 대해 실행된 것과 등가의 프로세스가 실행된다. "a" 디렉토리에 대한 부정 변경 체크가 완료된 후에, "b" 디렉토리에 대한 부정 변경 체크가 실행된다. 디렉토리 "a" 및 "b"에 대한 부정 변경 체크가 완료되었을 때, "/" 디렉토리에 주목하여 도 30의 단계 S301에 대한 프로세스가 실행된다. 단계 S301에서, 리프 증명서(2313)이 증명서 체인(231)인 "/ocap.certificate.1" 파일(2119)로부터 추출된다. 그 후, 단계 S302

에서, 공개키(2417)가 추출된 리프 증명서(2313)로부터 취해진다. 이어서, 단계 S303에서, "/ocap.hashfile" 파일(221)의 해시값이 계산된다. 한편, 단계 S304에서, "/ocap.certificate.1" 파일(2119) 내의 리프 증명서(2313) 내에 존재하는 공개키(2417)를 사용하여, "/ocap.signaturefile.1" 파일(2120) 내의 서명값(242)에 대해 복호가 실행된다. 단계 S305에서, 서명값을 복호함으로써, 단계 S303에서 계산된 해시값이 단계 S304에서 획득된 값과 동일한지의 여부가 체크된다. 이들 계산된 값이 일치하는 경우, "/" 디렉토리보다 하위의 파일 시스템이 부정 변경되지 않았다고 판정할 수 있다(단계 S306). 한편, 계산된 값들이 일치하지 않는 경우, 파일 시스템이 변경된 것으로 판정할 수 있다(단계 S307). 최상위 레벨 "/" 디렉토리에서 시작하여 내림차순으로 서브디렉토리를 향해 부정 변경 체크가 실행되는 일례에 대해 설명하였지만, 본 발명은 이것에 한정되는 것은 아니다. 따라서, 최하위 레벨 디렉토리에서 시작하여 오름차순으로 최상위 레벨 디렉토리를 향해 프로세스가 실행될 수도 있다. 상기 프로세스들을 통해, 도 27의 단계 S272의 결과가 획득된다.

[0123] 단계 S273에서, 단계 S272에서의 결과가 "부정 변경이 있다"일 때, 인증이 실패한 것으로 판정되고, 그러한 사실에 대한 통지가 이루어지며(단계 S279), 그 후 프로세스가 종료된다. 단계 S272의 결과가 "부정 변경 없음"일 때, 단계 S274에 대한 프로세스가 실행된다.

[0124] 이어서, 도 31 내지 도 33을 참조하여, 증명서 체인 인증(단계 S274)에 대하여 상세히 설명한다. 중간 증명서(2313)와 리프 증명서(2313)에 대한 체크가 먼저 실행된다고 가정하면, 그에 대한 플로우차트는 도 31에 도시된다. 먼저, 중간 증명서(2312) 및 리프 증명서(2313)가 증명서 체인(231)로부터 추출된다(단계 S311). 그러한 추출된 리프 증명서(2313)로부터, 금회 갱신 일시(2413), 다음회 갱신 일시(2414) 및 발행자명(2415)가 추출된다(단계 S312). 그들 중에서, 현재 일시가, 그 동안 증명서가 유효한 상태로 유지될 수 있는, 상기 금회 갱신 일시(2413)와 다음회 갱신 일시(2414) 사이에 있는지의 여부가 판정된다(단계 S313). 증명서가 유효한 상태로 유지될 수 있는 기간을 현재 일시가 초과하는 경우, 증명서 체인에 대한 인증이 실패로 종료한다(단계 S319). 한편, 현재 일시가 증명서의 유효 기간 내에 있는 것으로 판정될 때, 중간 증명서(2312) 내의 주체자명(1416) 및 공개키(2417)가 추출되고(단계 S314), 중간 증명서(2312)의 주체자명(2416)과 리프 증명서(2313)의 발행자명(2415) 사이의 비교가 행해져서, 중간 증명서(2312)와 리프 증명서(2313)가 체인 관계에 있는지의 여부를 판정한다(단계 S315). 이들 증명서가 체인 관계에 있지 않은 경우, 증명서 체인의 인증은 실패이다. 한편, 그들 사이에 체인 관계가 있을 때, 리프 증명서(2313)의 속성 영역(241)에 대한 해시값이 계산된다(단계 S316). 또한, 리프 증명서(2313) 내의 서명값(242)은 중간 증명서(2312)의 공개키(2417)로 복호된다(단계 S317). 단계 S316 및 S317이 완료될 때, 각각의 단계에서 획득된 해시값과 복호된 서명값이 일치하는지의 여부가 체크된다(단계 S318). 그들 값이 일치하지 않는 경우, 증명서 체인의 인증은 실패로 종료한다(단계 S319).

[0125] 이어서, 루트 증명서(2311)와 중간 증명서(2312) 사이의 체크가 실행된다. 도 32는 이 프로세스를 도시하는 플로우차트이다. 루트 증명서(2311)와 중간 증명서(2312)는 증명서 체인(231)으로부터 추출되고(단계 S321), 중간 증명서(2312)와 리프 증명서(2313)에 대해 실행된 체크와 등가인 프로세스가 루트 증명서(2311)와 중간 증명서(2312)에 대해 실행된다(단계 S322~단계 S328).

[0126] 단계 S328에서 상기 값들이 일치한다고 판정될 때, 루트 증명서(2311)에 대해서만 체크가 실행된다. 도 33은 루트 증명서(2311)에 대해서만 실행되는 체크를 도시하는 플로우차트이다. 단계 S321에서 추출된 루트 증명서(2311)로부터, 금회 갱신 일시(2413), 다음회 갱신 일시(2414) 및 발행자명(2415)이 추출된다(단계 S331). 그들 중에서, 현재 일시가, 그 동안 증명서가 유효한 상태로 유지될 수 있는, 상기 금회 갱신 일시(2413)와 다음회 갱신 일시(2414) 사이에 있는지의 여부가 판정된다(단계 S332). 현재 일시가 증명서가 유효한 상태로 유지될 수 있는 기간을 초과한 경우, 증명서 체인의 인증은 실패로 종료한다. 한편, 현재 일시가 증명서의 유효 기간 내에 있다고 판정될 때, 루트 증명서(2311)의 속성 영역(241)에 대한 해시값이 계산된다(단계 S334). 또한, 루트 증명서(2311) 내의 서명값(242)은 루트 증명서(2311)의 공개키(2417)로 복호된다(단계 S335). 단계 S334 및 단계 S335가 완료될 때, 각각의 단계에서 획득된 해시값과 복호된 서명값이 일치하는지의 여부가 체크된다(단계 S336). 그들 값이 일치하는 경우, 증명서 체인의 인증은 성공이지만(S337), 그들 값이 일치하지 않는 경우, 증명서 체인의 인증은 실패로 종료한다(단계 S338). 이 점에서, 단계 S274의 프로세스가 종료한다.

[0127] 프로세스는 S274의 결과에 따라 단계 S275에서 상이하게 실행된다. 단계 S274의 결과가 "증명서 체인의 인증이 실패하였다"일 때, 인증이 실패한 것으로 판정되고 그에 대한 통지가 행해지며(단계 S279), 그 후 파일 시스템에 대한 인증이 종료된다. 한편, "증명서 체인의 인증이 성공하였다"의 경우에, 단계 S276의 프로세스가 실행된다.

[0128] 이어서, 단말 장치(500)의 2차 기억부(510)는 "/ocap.certificate.1" 파일(2119)의 루트 증명서(2311)와 동일

한 증명서를 탐색한다(단계 S276). 동일한 증명서가 2차 기억부(510) 내에 존재하지 않을 때, 단계 S277에서 증명서 체인(231)의 인증이 실패한 것으로 판정되고, 이러한 인증 실패에 관한 통지가 행해지며(단계 S279), 그 후 프로세스가 종료된다. 한편, 루트 증명서(2311)가 포함될 때, 파일 시스템의 인증이 성공한 것으로 판정되고, 이러한 인증 성공에 관한 통지가 AM(1205b)에 행해진다(단계 S278). 도 28을 참조하면, 그 후 Java 프로그램의 기동전 통지가 수신될지라도(단계 S281), 프로세스는 실행되지 않고 종료된다.

[0129] 제1 실시예에서, 저장된 Java 프로그램이 일정 기간 후에 기동될 때, 파일 시스템이 저장되기 직전에 미리 인증되었기 때문에, 인증을 실행할 필요가 없다.

[0130] 여기에서는, 도 34에 도시된 "application description file"이 파일 시스템 내에 존재하고 그것에 기재된 파일들만 저장되는 경우를 설명한다. OCAP 사양에 따르면, 예를 들어, "application description file"은 XML(eXtensible Markup Language) 포맷으로 기재된다. 도 34는 "application description file"의 일례를 도시한다. 도 34에서, 도 21에 도시된 "PPV2Xlet.class"(2115)가 기재되어 있지 않다. 따라서, 이 경우에, "PPV2Xlet.class"(2115)가 기억 대상으로 포함되지 않는다. 이 경우에, S292에서 "PPV2Xlet.class"(2115)에 대해 해시값이 계산되지 않으므로, S293에서 "ocap.hashfile" 파일(2118)에 기재된 2233의 해시값과의 비교가 행해지지 않다. 단계 S294에서, 기억 대상으로 포함되지 않은 파일이 적용 외인 것으로 규정함으로써 S295의 프로세스로의 전이가 행해질 수도 있다.

[0131] (제2 실시예)

[0132] 파일 시스템에 포함된 Java 프로그램(PPV1Xlet.class(2114) 또는 PPV2Xlet.class(2115))이 그러한 파일 시스템이 저장된 후 일정 기간 동안 기동될 때, "/ocap.certificate.1" 파일(2119)에 포함된 X.509 증명서 중 하나의 유효성이 만료될(즉, Java 프로그램의 기동 일시>다음회 갱신 일시(2414)) 가능성이 있다. 그러나, 제1 실시예에서는 이미 만료된 X.509 증명서가 증명서 체인(231)에 포함되어 있을지라도 Java 프로그램이 기동될 수 있다.

[0133] 따라서, 본 실시예는 제1 실시예에 Java 프로그램을 기동할 때 증명서 체인(231)에 포함된 각각의 증명서(2311, 2312, 2313)의 유효성이 만료되지 않은 것을 검증하는 기능을 추가함으로써 달성된다. 도 26은 본 실시예의 구성요소를 도시한다. 본 실시예에 필요한 구성요소들(261~265)은 제1 실시예에서 이미 설명하였으므로, 여기에서는 그 설명을 생략한다.

[0134] 플로우차트로서, 도 27의 플로우차트는 도 35의 플로우차트로 대체되고, 도 36의 플로우차트가 추가된다.

[0135] 도 35를 참조하면, 파일 시스템이 저장되기 직전에 실행되는 프로세스들(단계 S351 내지 단계 S357)은 제1 실시예에서 설명한 프로세스들(단계 S271 내지 단계 S277)과 동일하므로, 그 설명을 생략한다. 인증이 실패되지 않은 경우, 프로세스는 도 36에 도시된 플로우차트로 진행한다. 일정 기간 후에 Java 프로그램인 PPV1Xlet.class(2114)가 기동될 것이라는 통지의 수취 시에(단계 S361), 각각의 X.509 증명서, 즉, 루트 증명서(2311), 중간 증명서(2312) 및 리프 증명서(2313)가 "ocap.certificate.1" 파일(2119)로부터 추출된다(단계 S362). 그 후, 추출된 X.509 증명서는 리프 증명서에서 시작하여 루트 증명서까지의 순서로 하나씩 선택되고(단계 S363), 현재 일시가 각각의 X.509 증명서의 금회 갱신 일시(2413)와 다음회 갱신 일시(2414)의 사이에 있는지의 여부를 체크한다(단계 S364). 현재 일시가 금회 갱신 일시(2413)와 다음회 갱신 일시(2414) 사이에 있지 않은 경우, 인증이 실패한 것으로 판정되고 그러한 사실에 관한 통지가 행해진다(단계 S367). 다른 경우에는, 모든 X.509 증명서에 대해 체크가 실행되었는지의 여부가 체크된다(단계 S365). 모든 X.509 증명서에 대해 체크가 완료되지 않은 경우, 프로세스는 S363으로 리턴되어, 후속 프로세스들이 반복된다. 한편, 모든 X.509 증명서가 단계 S365에서 미리 체크된 경우, 인증이 성공한 것으로 판정되고, 이 인증 성공에 관한 통지가 행해지며(단계 S366), 그 후 프로세스가 종료된다. 도 36의 플로우차트에 도시된 프로세스들을 추가함으로써, 유효기간이 만료된 Java 프로그램이 기동되지 않도록 인증 실패를 AM(1205b)에 통지할 수 있게 된다. 시큐리티 관리자(1205f)에 의해 인증 실패를 통지했을 때, AM(1205b)은 그러한 Java 프로그램을 JavaVM(1203)에 전달하지 않고 기동을 중지한다.

[0136] (제3 실시예)

[0137] 제1 실시예에서 설명한 바와 같이, 2차 기억부(510)는 증명서 체인(231) 내의 루트 증명서(2311)와 비교되는 루트 증명서인 X.509 증명서를 포함한다. 2차 기억부(510)에 저장된 루트 증명서는 증명서의 신뢰성이 해킹 등으로 인해 저하되는 경우를 대비하여 새로운 X.509 증명서로 교환된다(이하 증명서 교환이라고 한다). 새로운 X.509 증명서는 헤드 엔드(101)로부터 단말 장치(500)에 송신되어, 다운로드 모듈(106)을 통해 시큐리티 관리자

(1205f)에 송달된다.

[0138] 도 38(a), (b) 및 (c)는 시큐리티 관리자(1205f)에 의해 교환된 2차 기억부(510) 내의 루트 증명서를 각각 도시하는 도면이다. 이 경우에는, 증명서 A(381)가 교환될 오래된 증명서이고, 증명서 B(382)가 새로운 증명서이다. 도 38(a)의 38-1은 증명서 교환이 실행되기 전의 2차 기억부(510)에 저장된 증명서를 도시하고, 도 38(b)의 38-2는 교환되는 도중의 증명서를 도시하며, 도 38(c)의 38-3은 증명서 교환이 실행된 후의 2차 기억부(510)에 저장된 증명서를 도시한다.

[0139] 제1 실시예 및 제2 실시예에서는, Java 프로그램이 저장된 후에 증명서 교환이 실행될 때에도, Java 프로그램의 기동 시에 새로운 증명서를 고려하지 않았다. 예를 들어, 증명서 체인(231) 내의 루트 증명서(2311)는, 시큐리티 관리자(1205f)가 그 기억전 통지에 응답하여 Java 프로그램을 인증 중일 때, 증명서 A(381)와 일치하고, 시큐리티 관리자(1205f)는 증명서 A(381)가 증명서 B(382)로 교환된 후에 Java 프로그램의 기동전 통지를 수신한다고 가정한다. 이 시점에서, 2차 기억부(510)는 증명서 체인(231) 내의 루트 증명서(2311)와 일치하는 증명서를 포함하지 않으며, 이것은 그러한 증명서가 신뢰할 수 없다는 것을 의미한다. 그러나, 제1 실시예 및 제2 실시예에서는, Java 프로그램의 기동 직전에 루트 증명서들간의 비교가 행해지지 않기 때문에(즉, 증명서 체인(231) 내의 루트 증명서(2311)가 증명서 B(382)와 비교되지 않기 때문에), 인증 실패에 관한 통지가 AM(1205b)에 행해지지 않는다. 그 결과, AM(1205b)은 Java 프로그램이 기동되게 한다.

[0140] 그러므로, 본 실시예는 Java 프로그램 기동 시에 증명서 교환을 고려하여 루트 증명서의 비교를 실행하는 기능이 추가된다.

[0141] 도 26은 본 실시예의 구성요소를 도시한다. 구성요소들(261~265)은 이미 설명하였으므로 그 설명은 생략한다. 증명서 교환부(266), 증명서 교환 특정부(267) 및 증명서 수신부(268)가 추가된다.

[0142] 증명서 교환 특정부(267)가, 수신된 증명서보다 더 오래된 증명서가 2차 기억부(510)에 저장되어 있다고 판정했을 때, 증명서 교환부(266)는 그러한 오래된 증명서를 새로운 증명서로 교환한다. 한편, 증명서 교환 특정부(267)가 오래된 증명서가 저장되지 않았다고 판정할 때, 증명서 교환부(266)는 새로운 증명서를 2차 기억부(510)에 저장한다.

[0143] 증명서 교환 특정부(267)는 증명서 수신부(268)에 의해 수신된 증명서를 수신한다. 그 후, 증명서 교환 특정부(267)는 OS(1201)의 라이브러리(1201b)를 사용하여, 수신된 증명서보다 더 오래되고 발행자가 동일한 증명서가 있는지를 확인하기 위해 2차 기억부(510)에 저장된 증명서를 체크한다.

[0144] 증명서 수신부(268)는 다운로드 모듈(1206)이 헤드 엔드(101)로부터 새로운 증명서를 수신할 때, 그러한 새로운 증명서를 수신한다. 증명서의 수취 시에, 증명서 수신부(268)는 증명서를 증명서 교환부(266) 및 증명서 교환 특정부(267)에 전달한다.

[0145] 또한, 도 39 및 도 40이 도 35의 플로우차트에 이어서 추가된다.

[0146] 도 39는 증명서 교환을 실행할 때의 플로우차트인 한편, 도 40은 증명서 교환이 실행된 후에 Java 프로그램을 기동할 때의 플로우차트이다. 도 39를 참조하면, 증명서 교환의 요구가 수신될 때(단계 S391), 그러한 수신된 증명서의 발행자명이 추출된다(단계 S392). 교환될 필요가 있는 오래된 증명서가 단말 장치(500)의 2차 기억부(510)에 존재하는지의 여부가 체크되고(단계 S393), 오래된 증명서가 존재할 때만, 그러한 증명서가 삭제된다. 그 후, 수신된 증명서가 2차 기억부(510)에 저장된다(단계 S395). Java 프로그램의 기동 통지가 일정 기간 후에 수신될 때(단계 S401), 증명서 체인(231) 내의 루트 증명서(2311)와 일치하는 증명서를 2차 기억부(510)에서 탐색하며(단계 S402), 일치하는 증명서가 존재하면(단계 S403), 인증이 성공한 것으로 판정되고 그러한 사실에 관한 통지가 행해진다(단계 S404). 일치하는 증명서가 존재하지 않으면(단계 S403), 인증이 실해한 것으로 판정되고, 그러한 사실에 관한 통지가 행해진다(단계 S405). 단계 S404에서 인증이 성공한 것으로 판정하기 전에, 증명서 체인 내의 각각의 X.509 증명서가 "금회 갱신 일시<현재 일시<다음회 갱신 일시"를 충족시키는 것을 검증한 후에 인증이 성공한 것으로 결론지을 수도 있다.

[0147] 또한, 루트 증명서들이 일치하는지를 체크하는 것에 부가하여, S402 이전에, 증명서 체인 내의 증명서들이 체인 관계에 있는지의 여부를 확인하기 위해, 도 31~도 33에 도시된 체크를 실행한 후에 인증이 성공인지 실패인지를 판정하는 것도 가능하다.

[0148] 더욱이, 상기 설명은 교환되어야 하는 증명서가 발행자명에 기초하여 특정되는 경우에 대해 설명하였지만, 증명서는 주체자명과 같은 다른 속성값에 기초하여 특정될 수도 있다.

[0149] (제4 실시예)

[0150] 파일 시스템에 포함된 Java 프로그램(PPV1Xlet.class(2114) 또는 PPV2Xlet.class(2115))이 그러한 파일 시스템이 저장된 후 일정 기간 동안 기동될 때, "/ocap.certificate.1" 파일(2119)에 포함되는 X.509 증명서 중 어느 하나의 유효 기간이 만료된 것 및 루트 증명서가 교환된 것과 다른 이유로 증명서가 무효화된 경우가 있다. 이러한 구성으로 인해 무효화된 증명서가 존재할 때에도 Java 프로그램이 기동될 수 있다.

[0151] 여기에서, CRL(Certificate Revocation List)은 증명서를 무효화하는 것으로 공지되어 있다. 도 41은 CRL의 구성을 도시하는 도면이다. 여기에서는, 본 발명을 설명하는데 필요한 속성만이 도시되어 있다. CRL의 상세한 설명을 위해, IETF RF C3280 "Internet X.509 Public Key Infrastructure Certificate and CRL Profile"을 참조하라. 411은 CRL의 속성 영역을 나타내고, 412는 서명값(413)의 서명 알고리즘을 나타내며, 413은 CRL의 서명값을 나타낸다. 발행자명(4111)은 이 CRL의 발행자를 나타내고, 금회 갱신 일시(4112)는 CRL이 유효로 될 때의 일시를 나타내며, 다음회 갱신 일시(4113)는 CRL의 유효 기간이 만료할 때의 일시를 나타내고, 무효화된 증명서 리스트(4114)는 무효화된 X.509 증명서에 관한 정보를 나타낸다. 도 42는 무효화된 증명서 리스트(4114)의 구성을 도시하는 도면이다. 본 발명을 설명하는데 필요한 속성만이 여기에도 도시되어 있다. 복수의 무효화된 X.509 증명서에 관한 정보가 무효화된 증명서 리스트(4114)에 저장된다. 도 42의 경우에는, 무효화된 "증명서 A"(421)에 관한 정보로서, 증명서를 고유하게 식별하는 일련 번호(4211) 및 "증명서 A"(421)가 무효화될 때의 일시(4212)가 포함된다. 다른 무효화된 리스트들은 421과 등가이다.

[0152] 도 43은 CRL을 포함하는 파일 시스템의 구성의 일례를 도시한다. "/" 디렉토리(431), "a" 디렉토리(432), "SimpleXlet.class" 파일(433), "ocap.hashfile" 파일(434~435), "ocap.certificate.1" 파일(436), "ocap.signaturefile.1" 파일(437), "ocap.crl.2" 파일(438) 및 "ocap.certificate.2" 파일(439)이 내부에 저장된다. CRL을 포함하지 않는 파일 시스템의 인증은 제1 실시예에서 설명한 바와 같다. 따라서, 본 실시예에서는 그러한 파일의 증명서 체인인 "ocap.certificate.2" 파일(439) 및 CRL 포맷으로 구성되어 있는 "ocap.crl.2" 파일(438)에 초점을 맞춘다. OCAP 사양에 따르면, "ocap.crl.x"의 증명서 체인은 "ocap.certificate.x"이다. 도 43의 경우에는, "ocap.crl.2"의 증명서 체인은 "ocap.certificate.2"이다.

[0153] 도 46은 "ocap.hashfile" 파일(434)을 도시하는 개략도이다. 461은 ocap.hashfile(434)의 세부를 도시한다. "/" 디렉토리(431) 내에 존재하는 461의 ocap.hashfile은 동일한 디렉토리(431) 내에 존재하는 "ocap.certificate.1" 파일(436), "ocap.signature.1" 파일(437), "a" 디렉토리(432), "ocap.crl.2" 파일(438) 및 "ocap.certificate.2" 파일(439)의 각각에 관련된 해시값들을 포함한다.

[0154] 도 44는 CRL의 인증을 설명하기 위한 플로우차트이다. 이하의 설명은 파일 시스템이 도 43에 도시된 구성을 갖는 일례에 대한 설명이다. 먼저, 금회 갱신 일시(4112) 및 다음회 갱신 일시(4113)가 CRL로부터 추출되고(단계 S441), 현재 일시가 상기 금회 갱신 일시(4112)와 다음회 갱신 일시(4113) 사이에 있는지의 여부가 체크된다(단계 S442). 있지 않은 경우, 이 CRL은 무효라고 판정된다(단계 S447). 현재 일시가 그들 일시 사이에 있으면, "ocap.crl.2" 파일(438)의 서명값을 검증하기 위해 속성 영역(411)의 해시값이 계산된다(단계 S443). 동시에, 리프 증명서(2313)의 공개키(2417)가 증명서 체인인 "ocap.certificate.2" 파일(439)로부터 추출되고(단계 S444), "ocap.crl.2" 파일(438)의 서명값(413)이 추출된 공개키(2417)로 복호된다(단계 S445). 그 후, 단계 S443에서 획득된 해시값이 단계 S445에서 획득된 복호값과 동일한지의 여부가 체크된다(단계 S446). 그들 값이 동일하지 않으면, CRL이 무효하고 판정된다(단계 S447). 그들 값이 동일하면, 도 45를 참조하여, 증명서 체인인 "ocap.certificate.2" 파일(439)에 대한 인증이 실행된다(단계 S451). 증명서 체인을 인증하는 방법은 도 31 내지 도 33에 도시된 것과 동일하므로, 여기에서 설명하지 않는다. 이어서, 증명서 체인의 인증이 성공인지의 여부가 판정되고(단계 S452), 인증이 실패이면, 이 CRL은 무효인 것으로 판정된다(단계 S456). 한편, 인증이 성공이면, 2차 기억부(510)에서 루트 증명서와 동일한 증명서를 탐색한다(단계 S453). 여기에서, 일치하는 루트 증명서가 존재하지 않으면, 인증이 실패이고 이 CRL은 무효라고 판정되지만(단계 S456), 일치하는 루트 증명서가 포함되면, 인증이 성공이고 CRL은 유효하다고 판정된다.

[0155] 이하 증명서가 CRL에 따라 무효화되는 것에도 불구하고 Java 프로그램이 기동되는 문제점에 대한 해결법을 설명한다. 이를 뒷받침하기 위해, 본 실시예는, Java 프로그램의 기동 통지가 행해질 때, 그러한 Java 프로그램을 인증하는데 사용된 증명서가 CRL 내의 무효화된 증명서인지의 여부를 판정하는 기능이 추가된다.

[0156] 도 26은 본 실시예의 구성요소를 도시한다. 일부 추가가 행해진 262와 아직 설명하지 않은 269를 제외하고, 상술한 구성요소들에 대해서는 설명하지 않는다.

[0157] CRL을 인증할 수 있는 판정부(262)는 증명서 무효화 특정부(269)에 CRL에 의해 무효화될 증명서를 특정하도록 요구한다. 그 후, 통지 수신부(261)가 증명서 무효화 특정부(269)에 의해 특정된 무효화된 증명서에 관련된 Java 프로그램의 기동전 통지를 수신할 때, 판정부(262)는 인증이 실패인 것으로 판정한다. 한편, 통지 수신부(261)가 판정부(262)가 CRL을 인증하는데 실패하였으므로 그러한 CRL이 무효라고 판정한 상태로 Java 프로그램의 기동전 통지를 수신할 때, 판정부(262)는 인증이 성공이라고 판정한다.

[0158] 판정부(262)가 CRL의 인증이 성공하였다고 인식할 때, 증명서 무효화 특정부(269)는 증명서 추출부(265)에 의해 추출된 X.509 증명서 중 하나가 무효화된 증명서라고 특정한다.

[0159] 플로우차트로서, 도 47 및 도 48이 추가된다. 이하 이들 플로우차트에 따라 설명을 행한다. 도 21에 도시된 파일 시스템의 기억전 통지가 지금 행해지고, 도 35의 플로우차트에 도시된 프로세스가 개시되며, 단계 S357의 프로세스가 적당한 때에 완료된다고 가정한다. 도 43에 도시된 다른 파일 시스템의 기억전 통지가 그후 수취된다고 가정하면, 도 44의 플로우차트에 도시된 프로세스의 완료 후에 단계 S471 내지 단계 S477이 실행된다. 단계 S471 내지 단계 S477의 프로세스들은 단계 S351 내지 단계 S357의 프로세스들과 동일하다. 단계 S478에 도달했을 때, "ocap.crl.2" 파일(438)의 인증(도 44 및 도 45의 플로우차트)이 성공이면, 그러한 파일에 포함되는 무효화된 증명서에 관한 정보가 무효화된 증명서의 데이터베이스에 기입된다. 도 49는 무효화된 증명서의 데이터베이스를 도시하는 개략도이다. 방행자명이 열(491)에 저장되고, 증명서 일련 번호가 열(492)에 저장되며, 무효 일시가 열(493)에 저장된다. 여기에서, "PPV1Xlet.class"(2114)의 기동전 통지가 수취될 때(단계 S481), "ocap.certificate.1" 파일(2119)의 증명서 체인(231)에 포함되는 X.509 증명서 중 어느 하나가 무효화된 증명서의 데이터베이스에 포함되는지의 여부가 체크된다(단계 S483). 해당하는 증명서가 있으면, 인증이 실패라고 판정하고 이에 관한 통지가 행해진다(단계 S486). 한편, 해당하는 증명서가 존재하지 않을 때, 전체 증명서 체인에 대해 체크가 실행되고(단계 S484), 인증이 성공인 것으로 판정하는 통지가 행해진다(단계 S485). 상기 프로세스를 통해, 검증 시에 파일의 인증이 유효였던 증명서를 갖는 파일 시스템에 대해 실패이지만, Java 프로그램이 기동되었을 때까지 CRL에 의해 무효화된다.

[0160] 제1 내지 제4 실시예에서는, Java 프로그램의 기동전 통지가 수신될 때, 각 디렉토리에 위치하는 "ocap.hashfile"을 사용하여, 파일 시스템의 트리 구조가 정확한지의 여부를 확인하기 위해 검증을 더 실행할 수도 있다.

[0161] 또한, 간략하기 위한 목적으로 증명서 체인 내에는 하나의 중간 증명서만 존재하지만, 복수의 중간 증명서가 존재할 수도 있다. 그러나, 그 인증 체인의 인증이 실행될 때, 모든 중간 증명서들은 체인 관계에 있는 것이 필요하다.

[0162] 또한, 이하의 프로세스는 언급한 순서로 설명하였지만, 본 발명은 그러한 순서: 부정 변경의 존부의 체크, 증명서 체인의 인증, 및 2차 기억부가 증명서 체인 내의 루트 증명서와 동일한 루트 증명서를 포함하는지를 확인하기 위한 체크의 순서에 한정되는 것은 아니다.

[0163] 또한, 파일 시스템의 기억에 관하여, 시큐리티 관리자(1205f)는 OS의 라이브러리(1201b)를 사용하여 파일 시스템을 저장할 수도 있다. 또한, 제1 내지 제4 실시예는, 파일 시스템의 최상위 디렉토리 "/"에 "application description file"이 제공되고 그 콘텐츠로서 파일 시스템의 일부만이 저장될 파일로 나타나는 경우에도 적용 가능하다. 따라서, 저장될 파일만이 취급되면 문제가 발생하지 않는다.

[0164] 또한, 기억 대상으로서 프로그램들을 설명하였지만, 프로그램 이외의 데이터가 기억 대상이 될 수도 있고, 이것은 제1 내지 제4 실시예가 데이터에도 적용 가능한 것을 의미한다.

[0165] 또한, 하나 이상의 "ocap.certificate.x"가 "ocap.signaturefile.x"에 대응할 가능성이 있으며, 이 경우에 적어도 하나의 "ocap.certificate.x" 파일의 인증이 성공되는 것이 필요하다.

[0166] 또한, 증명서 체인의 일례로서 "ocap.certificate.x"가 제시되고, 해시값을 갖는 파일의 일례로서 "ocap.hashfile"이 제시되며, "/" 디렉토리 내의 "ocap.hashfile"가 부정 변경되었는지의 여부를 체크하는 파일의 일례로서 "ocap.signaturefile.x"가 제시되어 있지만, 본 발명은 이들 파일명에 한정되는 것은 아니다.

[0167] 또한, 인증 실패의 경우에, 인증은 다시 다운로드한 후에 실행될 수도 있다.

[0168] 또한, 인증 실패의 경우에, 저장된 프로그램 뿐만 아니라 인증에 사용된 증명서 체인, 서명 파일, 해시 파일이 기억 영역에 대해 충분한 용량을 유지하기 위해 삭제될 수도 있다.

[0169] 여기에서는, 프로그램을 구성하는 파일 시스템이 도 50에 도시된 구성을 갖는 경우를 설명하고, 도 51에 도시된

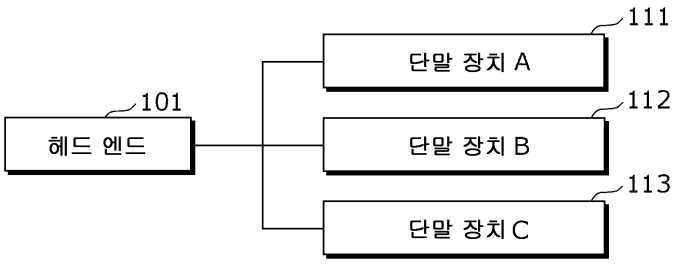
"application description file"의 경우에서와 같은 인증에 사용될 파일의 설명은 없다. 도 50에 도시된 5011 내지 5020은 도 21에 도시된 2111 내지 2120과 등가이다. 5021은 저장될 파일을 기재하는 "application description file"을 나타낸다. 도 51에 도시된 "application description file"에서는, 인증에 필요한 "ocap.hashfile"(5017), "ocap.signaturefile.1"(5020) 및 "ocap.certificate.1"(5019)의 설명은 없다. 이 경우에, 도 51에 도시된 바와 같이 파일이 저장되면, 인증을 실행하는데 필요한 파일들이 저장되지 않는다. 따라서, 제2, 제3 및 제4 실시예에서 제시된 인증은 기동 시에 실행될 수 없다. 저장된 프로그램이 기동될 때, 그러한 프로그램이 저장되기 전의 파일을 도시하는 도 50에 도시된 파일들이 다운로드를 위해 준비되고, 저장된 파일들은 프로그램을 구성하는 파일로서 사용될 수도 있으며, 인증용으로 사용되는 파일들은 인증용으로 다시 다운로드될 수도 있다.

[0170] 그러나, 프로그램이 저장되기 전의 파일을 도시하는 도 50에 도시된 파일이 다운로드될 수 없는 경우가 있을 수도 있다. 따라서, 인증에 필요한 파일이 "application description file"에 기재되지 않더라도, 인증에 필요한 파일은 프로그램 기동 시에 실행될 인증용으로 저장될 수도 있다.

[0171] 본 발명의 일부 예시적인 실시예들만 상세히 설명하였지만, 당업자라면 본 발명의 신규 특징 및 이점으로부터 실질적으로 벗어남없이 예시적인 실시예에서 다양한 변형이 가능하다는 것을 쉽게 이해할 것이다. 따라서, 그러한 모든 변형은 본 발명의 범위 내에 포함되는 것으로 간주된다.

도면

도면1



도면2

주파수 대역	용도	변조 방식
5~130MHz	대역외 (OOB) 헤드 엔드와 단말간의 데이터 교환	QPSK
130~864MHz	대역내 비디오 및 오디오를 포함하는 통상 텔레비전 방송	QAM

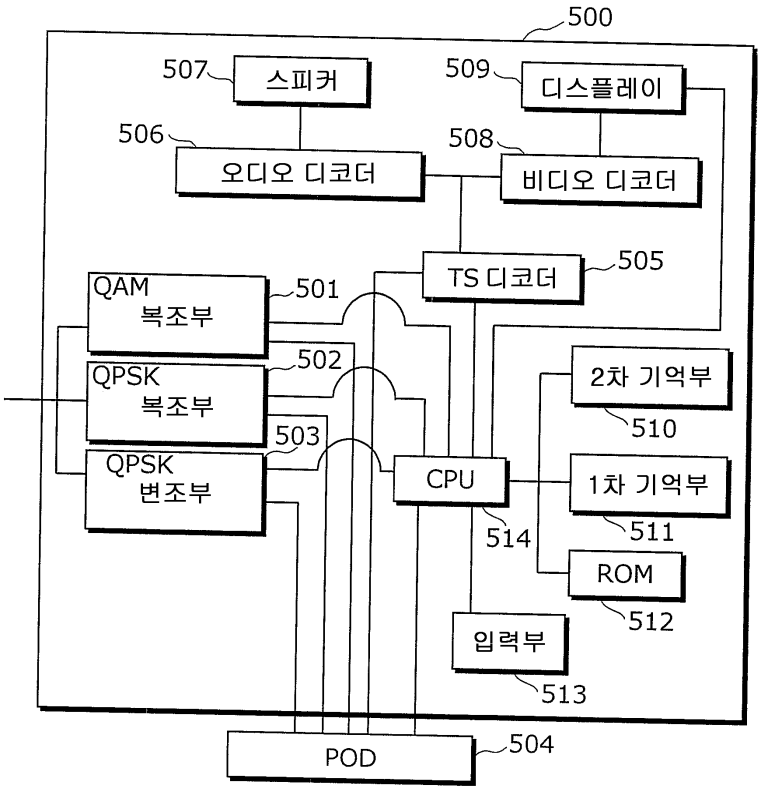
도면3

주파수 대역	용도
70~74MHz	헤드 엔드(101)로부터 단말 장치로의 데이터 송신
10.0~10.1MHz	단말 장치 A(111)로부터 헤드 엔드(101)로의 데이터 송신
10.1~10.2MHz	단말 장치 B(112)로부터 헤드 엔드(101)로의 데이터 송신
10.2~10.3MHz	단말 장치 C(113)로부터 헤드 엔드(101)로의 데이터 송신

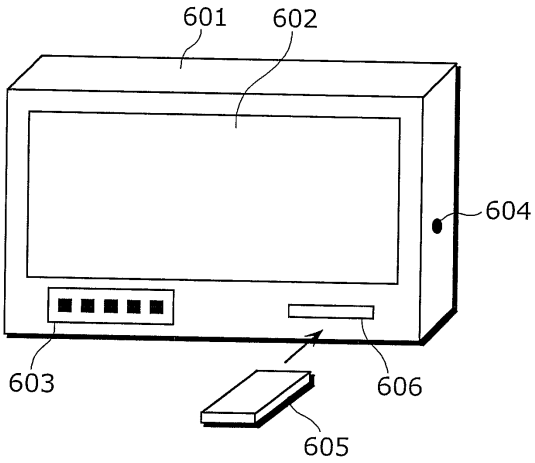
도면4

주파수 대역	용도
150~156MHz	텔레비전 채널 1
156~162MHz	텔레비전 채널 2
⋮	⋮
310~311MHz	라디오 채널 1
⋮	⋮

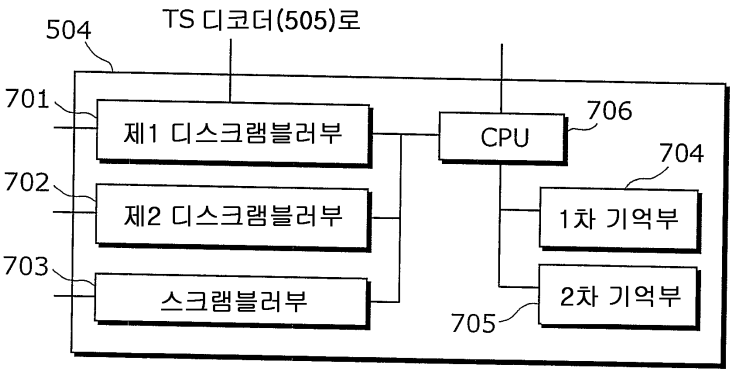
도면5


500
507
스피커
509
디스플레이
506
오디오 디코더
508
비디오 디코더
TS 디코더
505
QAM
복조부
501
QPSK
복조부
502
QPSK
변조부
503
2차 기억부
510
CPU
1차 기억부
511
514
ROM
512
입력부
513
POD
504

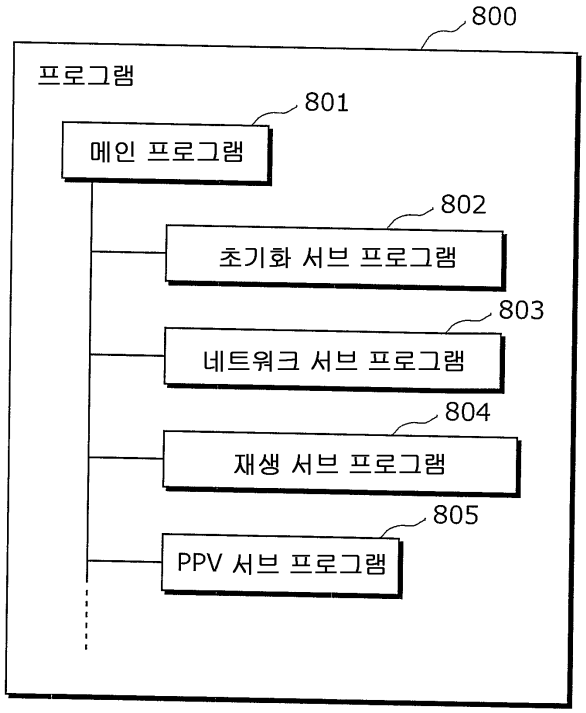
도면6


601
602
604
603
606
605

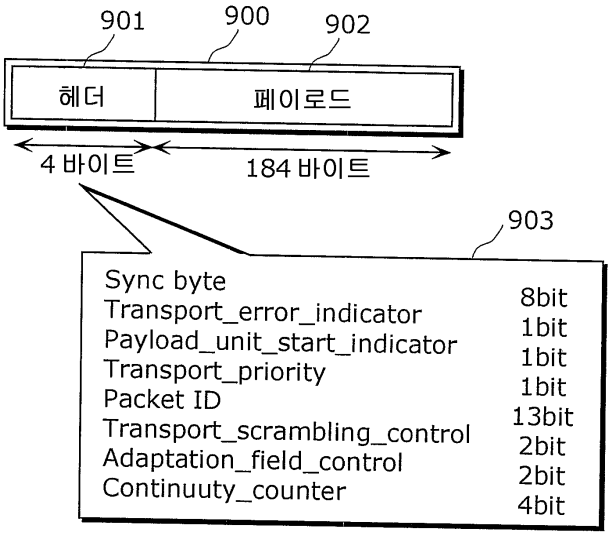
도면7


504
TS 디코더(505)로
701
제1 디스크램블러부
702
제2 디스크램블러부
703
스크램블러부
CPU
706
704
1차 기억부
2차 기억부
705

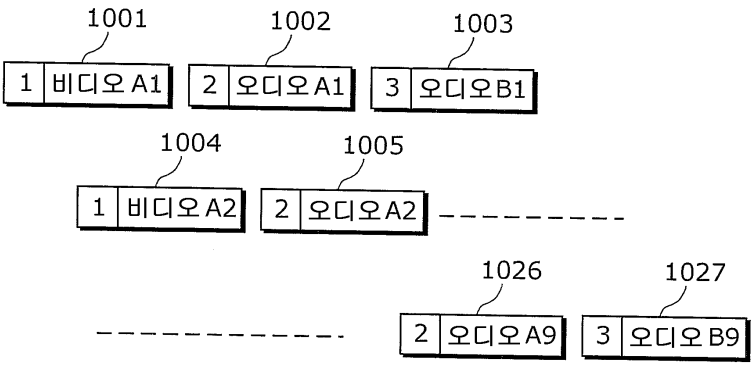
도면8


800
프로그램
801
메인 프로그램
802
초기화 서브 프로그램
803
네트워크 서브 프로그램
804
재생 서브 프로그램
805
PPV 서브 프로그램

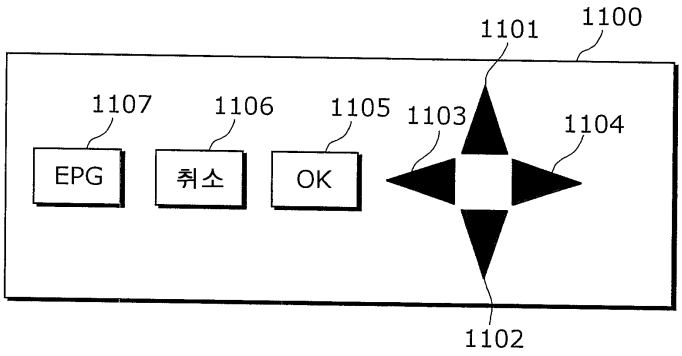
도면9



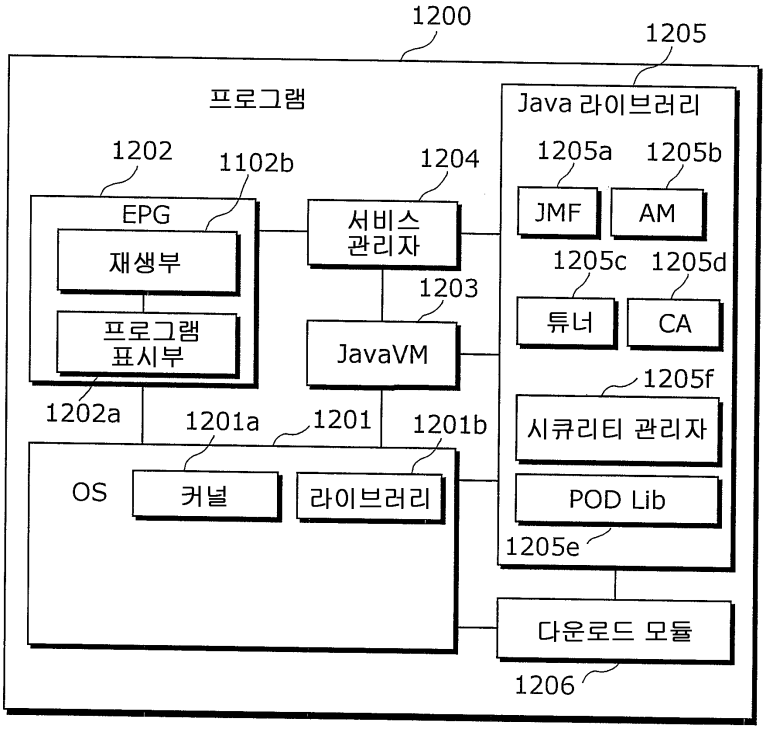
도면10



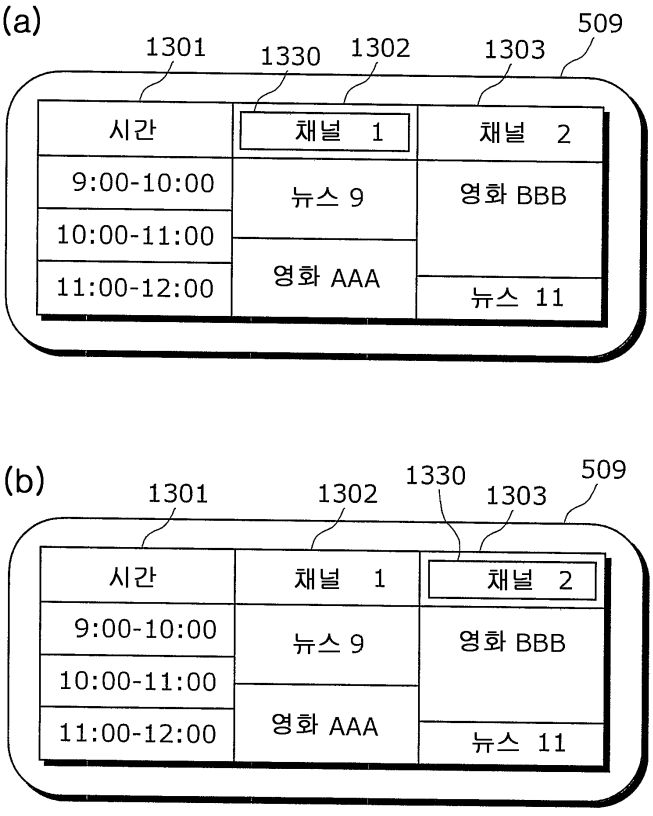
도면11



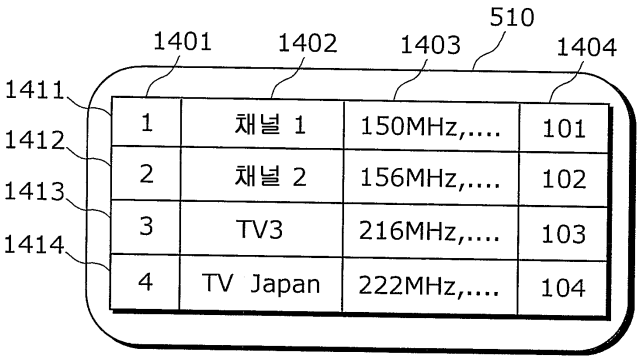
도면12


1200
1205
프로그램
Java 라이브러리
1202
1102b
1204
1205a
1205b
EPG
재생부
프로그램 표시부
서비스 관리자
JMF
AM
1205c
1205d
1203
튜너
CA
JavaVM
1205f
시큐리티 관리자
1202a
1201a
1201
1201b
OS
커널
라이브러리
POD Lib
1205e
다운로드 모듈
1206

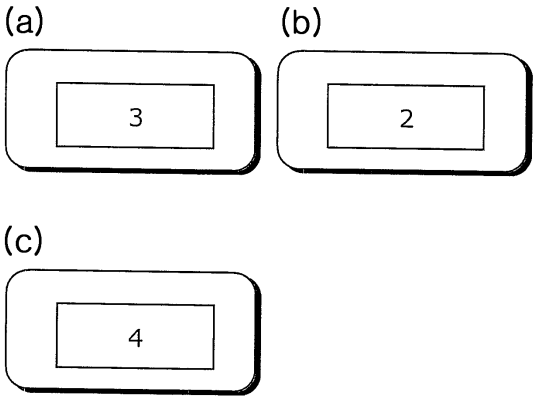
도면13


(a)
1301
1330
1302
1303
509
시간
채널 1
채널 2
9:00-10:00
뉴스 9
영화 BBB
10:00-11:00
11:00-12:00
영화 AAA
뉴스 11
(b)
1301
1302
1330
1303
509
시간
채널 1
채널 2
9:00-10:00
뉴스 9
영화 BBB
10:00-11:00
11:00-12:00
영화 AAA
뉴스 11

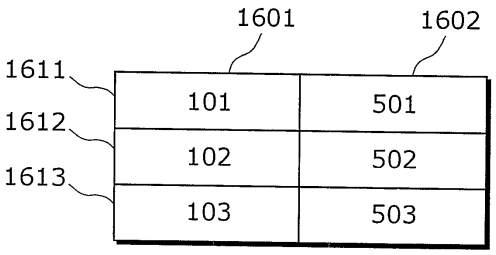
도면14



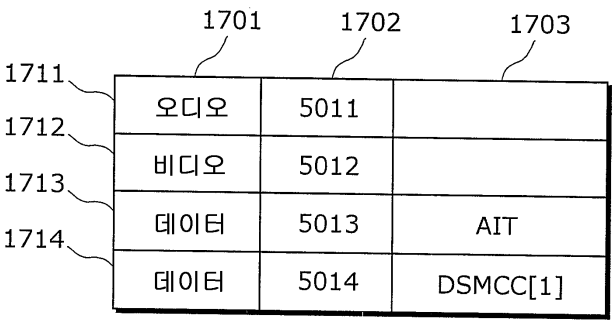
도면15



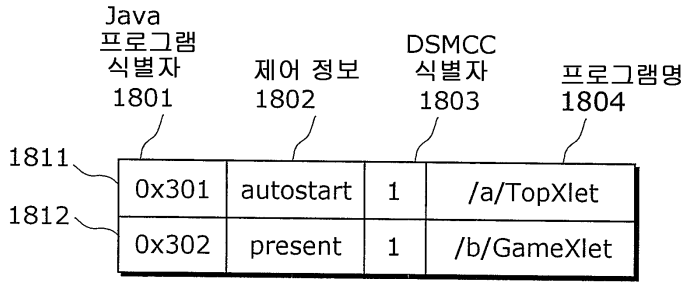
도면16



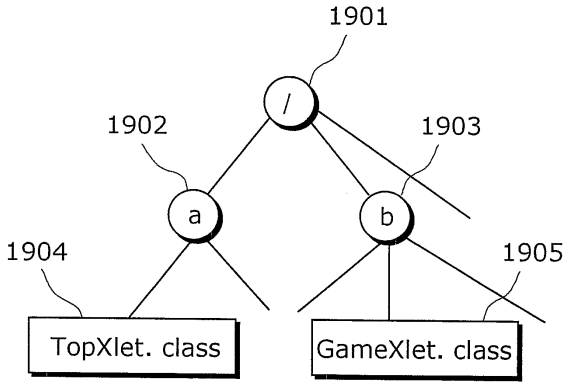
도면17



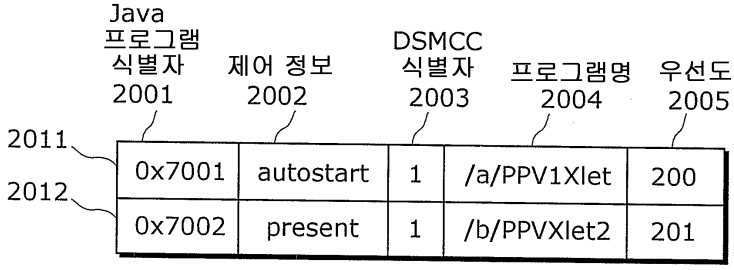
도면18



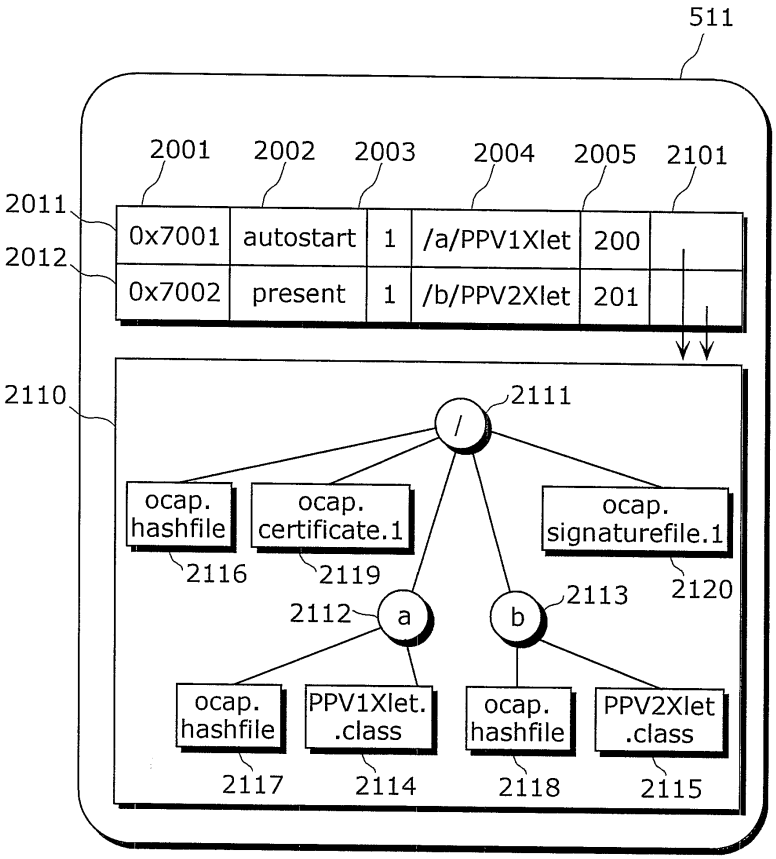
도면19



도면20



도면21


511
2001
2002
2003
2004
2005
2101
2011
2012
0x7001
autostart
1
/a/PPV1Xlet
200
0x7002
present
1
/b/PPV2Xlet
201
2110
2111
/
ocap.
hashfile
ocap.
certificate.1
ocap.
signaturefile.1
2116
2119
2120
2112
a
b
2113
ocap.
hashfile
PPV1Xlet.
.class
ocap.
hashfile
PPV2Xlet
.class
2117
2114
2118
2115

도면22

(a) 221

파일명 또는 디렉토리명 (2211)	해시 알고리즘 (2212)	해시값 (2213)
ocap.certificate.1	SHA1	e3 f4...3f
ocap.signaturefile.1	SHA1	03 98...35
a	SHA1	45 97...20
b	SHA1	a3 76...39

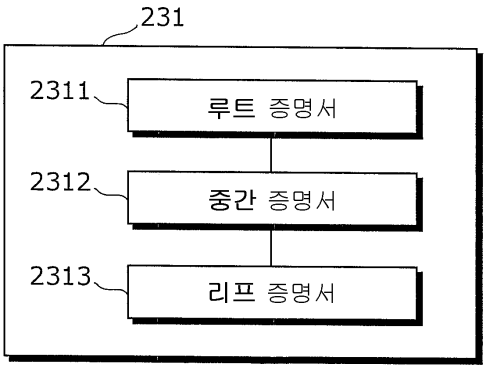
(b) 222

파일명 또는 디렉토리명 (2221)	해시 알고리즘 (2222)	해시값 (2223)
PPV1Xlet.class	SHA1	c8 38...59

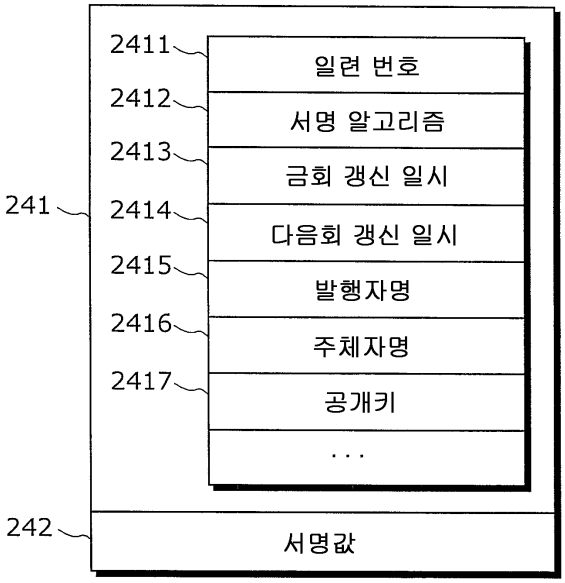
(c) 223

파일명 또는 디렉토리명 (2231)	해시 알고리즘 (2232)	해시값 (2233)
PPV2Xlet.class	SHA1	34 b4...56

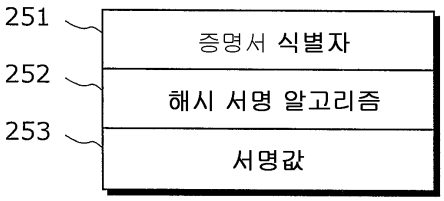
도면23



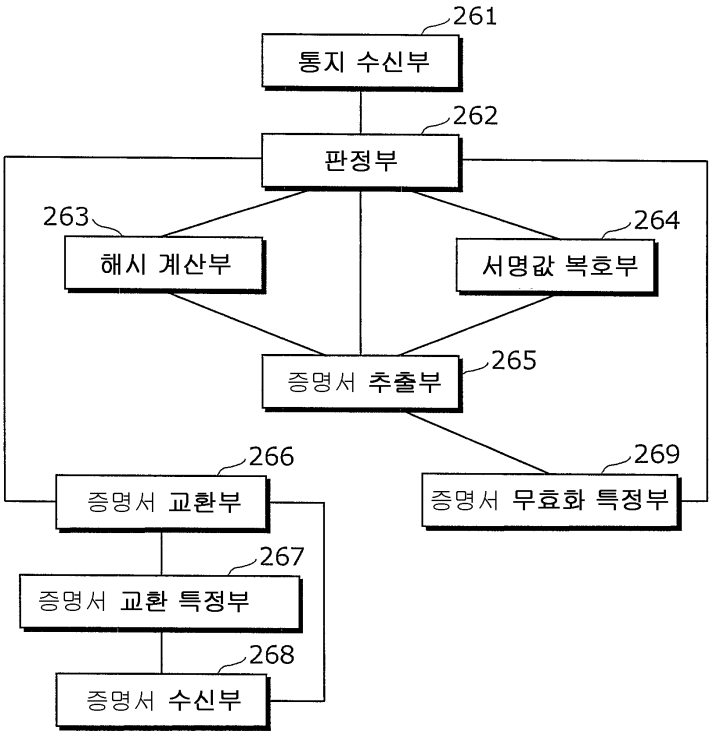
도면24



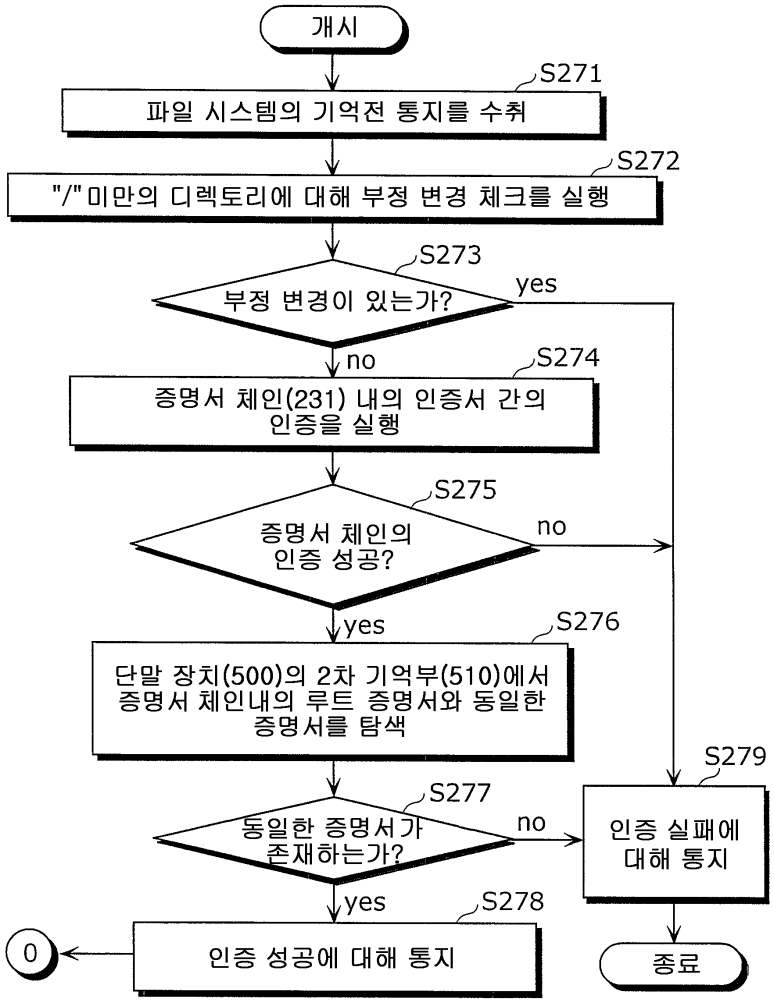
도면25



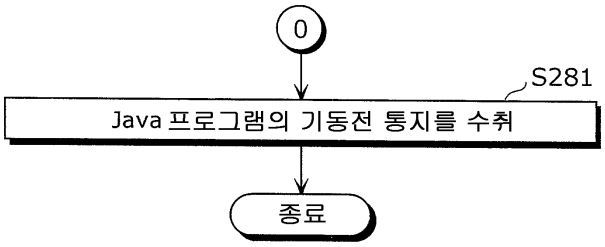
도면26



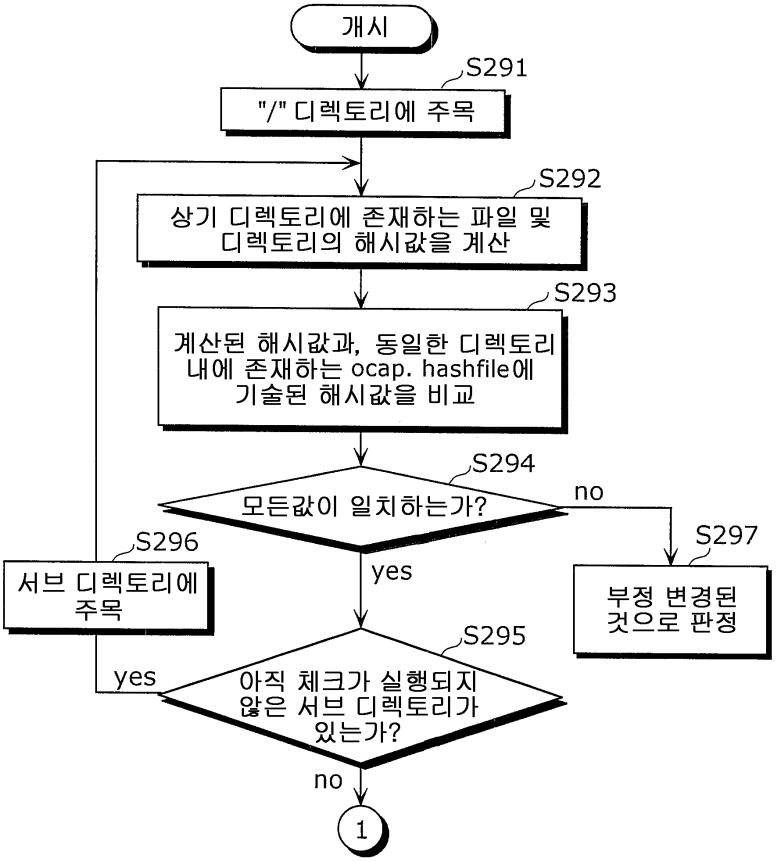
도면27


개시
S271
파일 시스템의 기억전 통지를 수취
S272
"/" 미만의 디렉토리에 대해 부정 변경 체크를 실행
S273
부정 변경이 있는가?
yes
no
S274
증명서 체인(231) 내의 인증서 간의 인증을 실행
S275
증명서 체인의 인증 성공?
no
yes
S276
단말 장치(500)의 2차 기억부(510)에서 증명서 체인내의 루트 증명서와 동일한 증명서를 탐색
S277
동일한 증명서가 존재하는가?
no
yes
S278
인증 성공에 대해 통지
S279
인증 실패에 대해 통지
종료
0

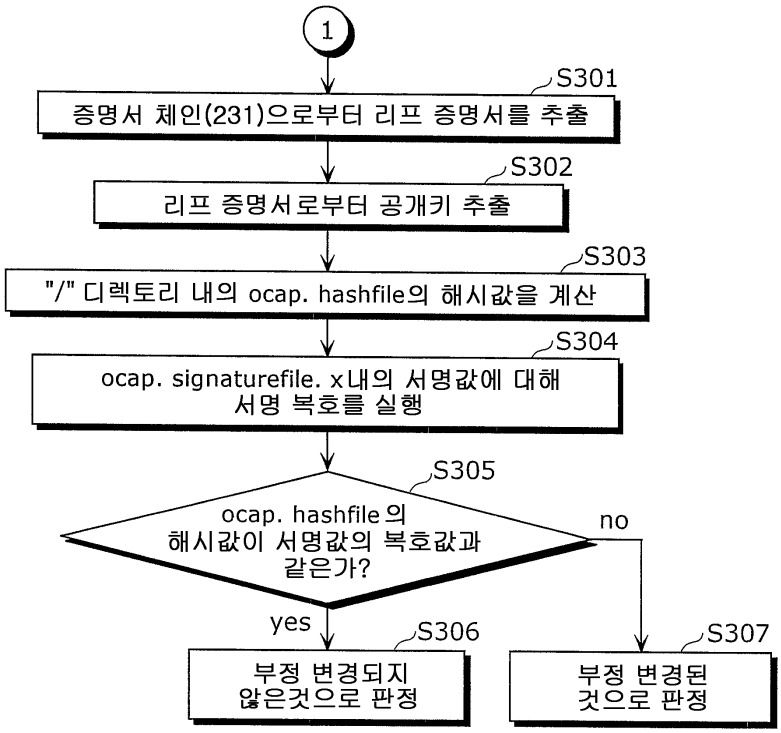
도면28


0
S281
Java 프로그램의 기동전 통지를 수취
종료

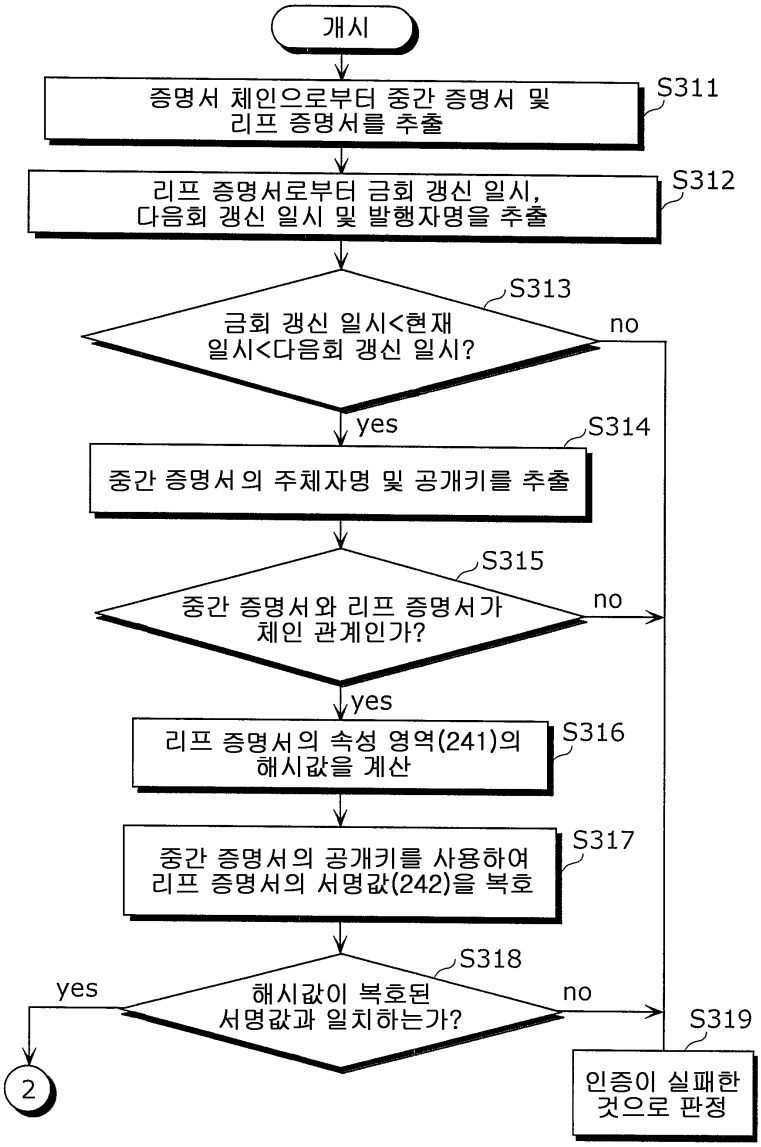
도면29


개시
S291
"/" 디렉토리에 주목
S292
상기 디렉토리에 존재하는 파일 및 디렉토리의 해시값을 계산
S293
계산된 해시값과, 동일한 디렉토리 내에 존재하는 ocap. hashfile에 기술된 해시값을 비교
S294
모든값이 일치하는가?
no
S297
부정 변경된 것으로 판정
yes
S296
서브 디렉토리에 주목
S295
yes
아직 체크가 실행되지 않은 서브 디렉토리가 있는가?
no
1

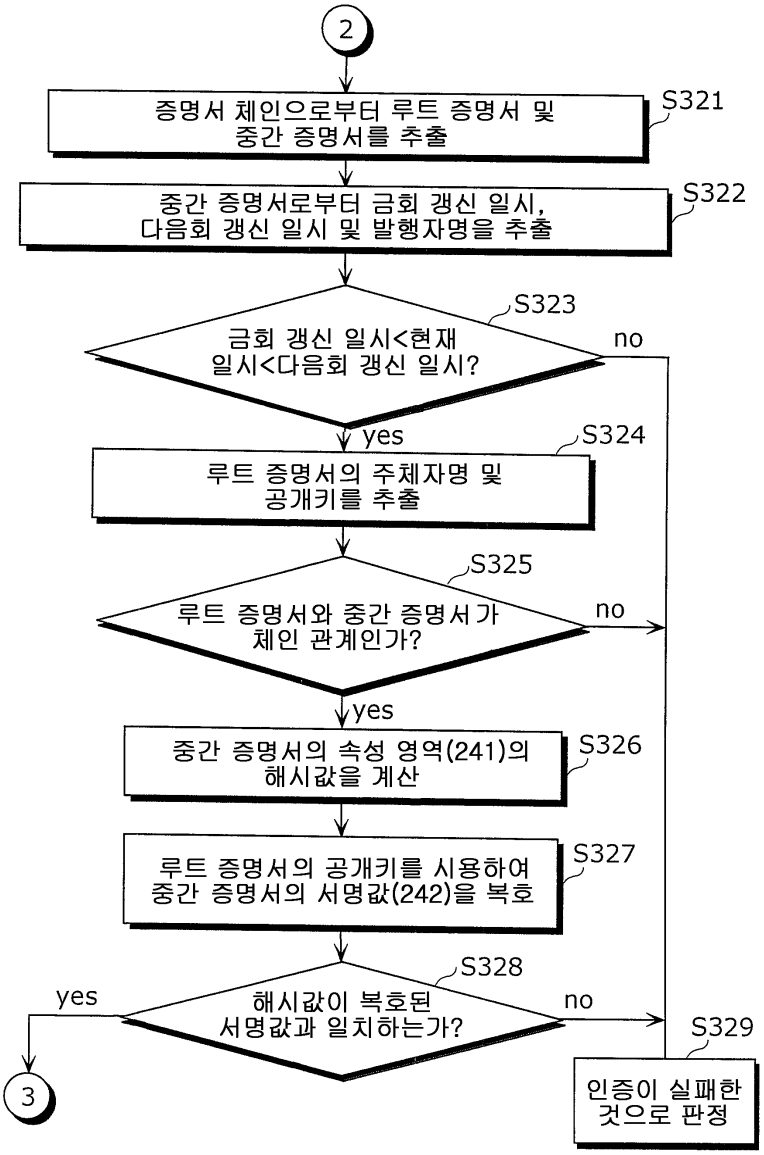
도면30


1
S301
증명서 체인(231)으로부터 리프 증명서를 추출
S302
리프 증명서로부터 공개키 추출
S303
"/" 디렉토리 내의 ocap. hashfile의 해시값을 계산
S304
ocap. signaturefile. x내의 서명값에 대해 서명 복호를 실행
S305
ocap. hashfile의 해시값이 서명값의 복호값과 같은가?
no
yes
S306
부정 변경되지 않은것으로 판정
S307
부정 변경된 것으로 판정

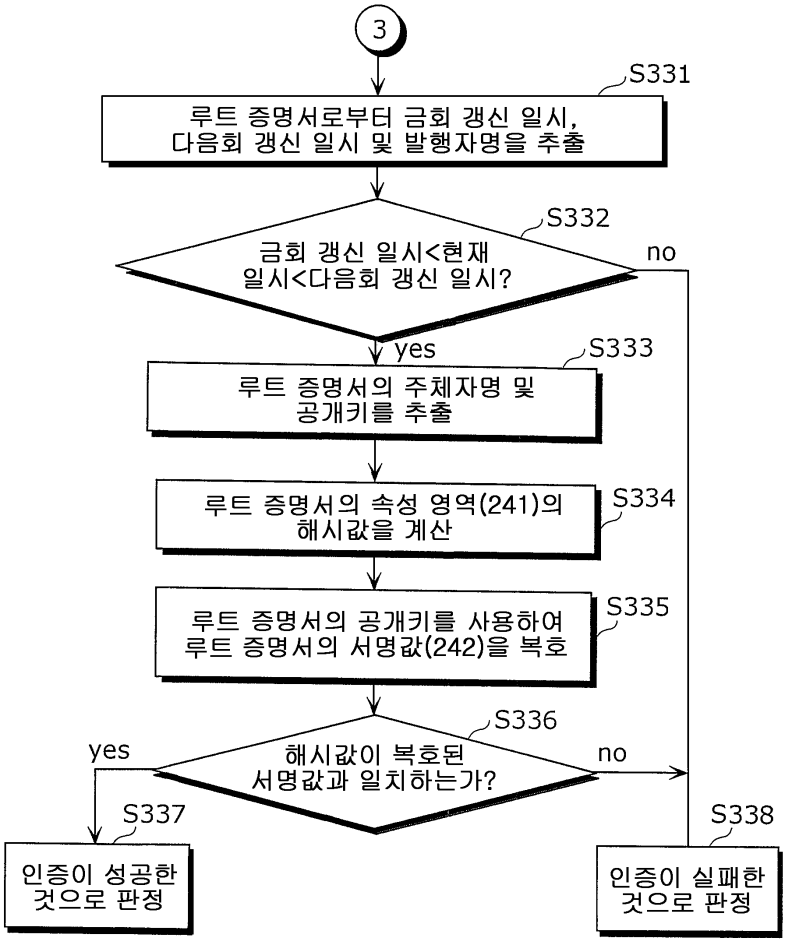
도면31


개시
증명서 체인으로부터 중간 증명서 및 리프 증명서를 추출
S311
리프 증명서로부터 금회 갱신 일시, 다음회 갱신 일시 및 발행자명을 추출
S312
금회 갱신 일시<현재 일시<다음회 갱신 일시?
S313
no
yes
중간 증명서의 주체자명 및 공개키를 추출
S314
중간 증명서와 리프 증명서가 체인 관계인가?
S315
no
yes
리프 증명서의 속성 영역(241)의 해시값을 계산
S316
중간 증명서의 공개키를 사용하여 리프 증명서의 서명값(242)을 복호
S317
해시값이 복호된 서명값과 일치하는가?
S318
yes
no
2
인증이 실패한 것으로 판정
S319

도면32


2
증명서 체인으로부터 루트 증명서 및 중간 증명서를 추출
S321
중간 증명서로부터 금회 갱신 일시, 다음회 갱신 일시 및 발행자명을 추출
S322
S323
금회 갱신 일시<현재 일시<다음회 갱신 일시?
no
yes
S324
루트 증명서의 주체자명 및 공개키를 추출
S325
루트 증명서와 중간 증명서가 체인 관계인가?
no
yes
중간 증명서의 속성 영역(241)의 해시값을 계산
S326
루트 증명서의 공개키를 사용하여 중간 증명서의 서명값(242)을 복호
S327
S328
yes
해시값이 복호된 서명값과 일치하는가?
no
S329
인증이 실패한 것으로 판정
3

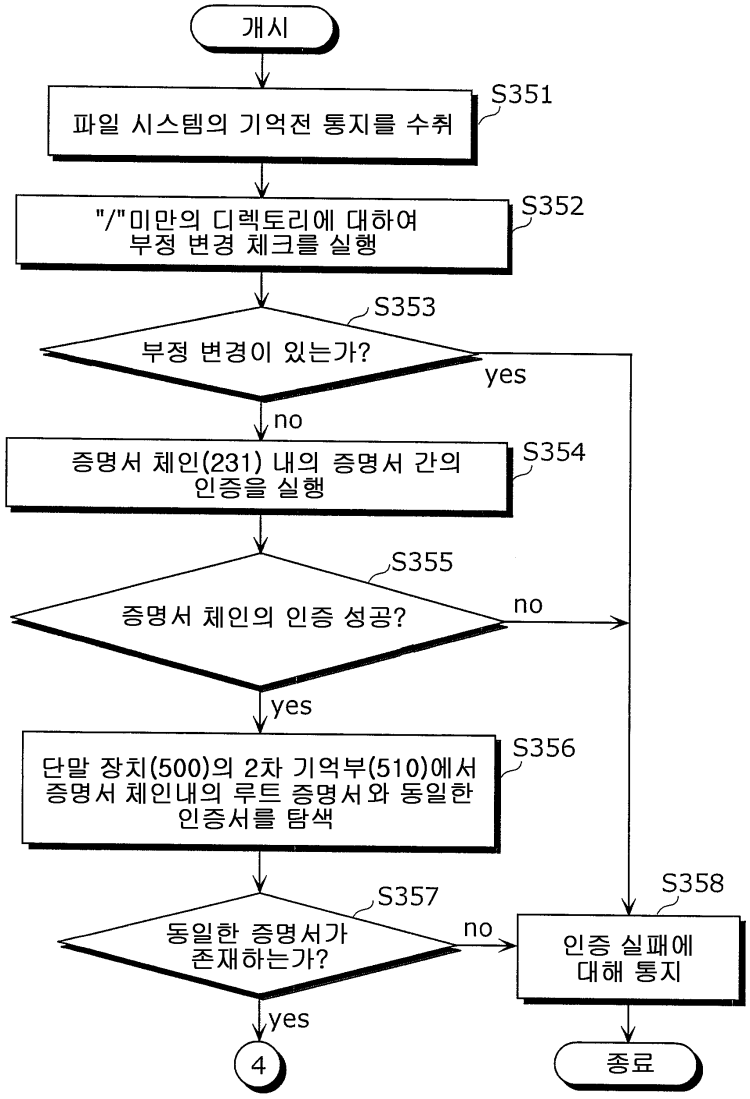
도면33



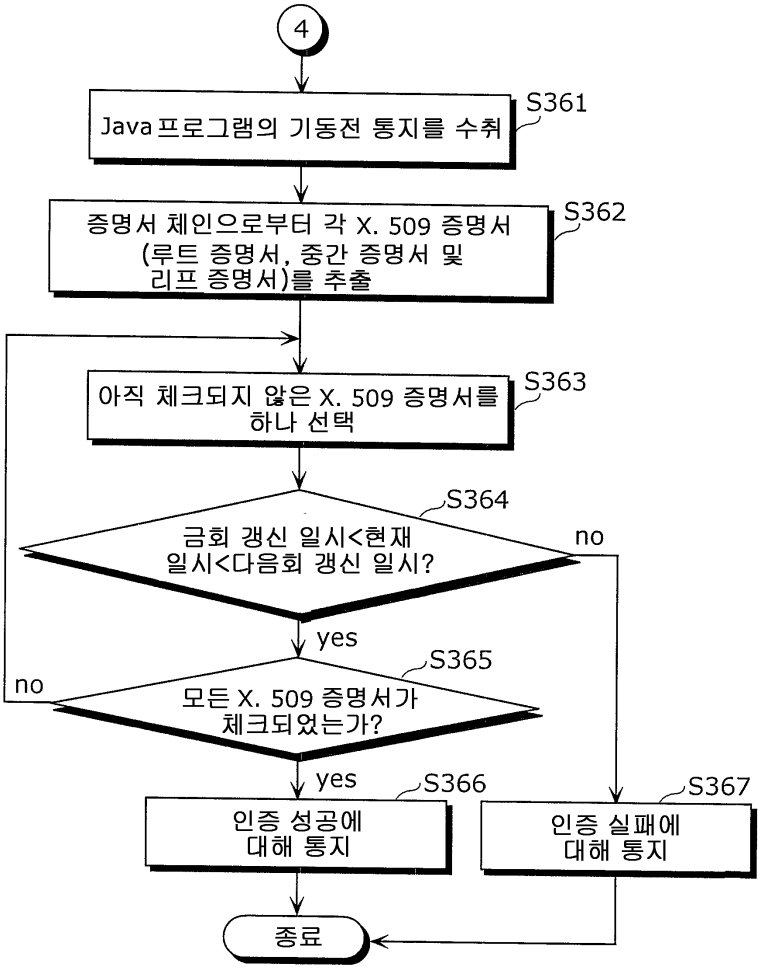
도면34

```
"-//OCAP//DTD Application Description File 1.0//EN"
"http://www.cablelabs.com/ocap/dtd/applicationdescriptionfile
-1-0.dtd"
<applicationdescription>
   <dir name="/">
      <file name="ocap.hashfile" size="25"/>
      <file name="ocap.certificate.1" size="100"/>
      <file name="ocap.signaturefile.1" size="30"/>
      <dir name="a">
         <file name="ocap.hashfile" size="15"/>
         <file name="PPV1Xlet.class" size="1000"/>
      </dir>
      <dir name="b">
         <file name="ocap.hashfile"/>
      </dir>
   </dir>
</applicationdescription>
```

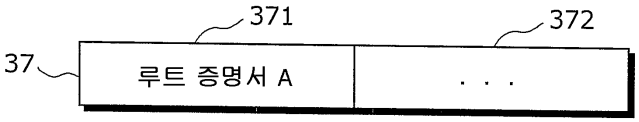
도면35


개시
파일 시스템의 기억전 통지를 수취
S351
"/"미만의 디렉토리에 대하여 부정 변경 체크를 실행
S352
부정 변경이 있는가?
S353
yes
no
증명서 체인(231) 내의 증명서 간의 인증을 실행
S354
증명서 체인의 인증 성공?
S355
no
yes
단말 장치(500)의 2차 기억부(510)에서 증명서 체인내의 루트 증명서와 동일한 인증서를 탐색
S356
동일한 증명서가 존재하는가?
S357
no
yes
4
인증 실패에 대해 통지
S358
종료

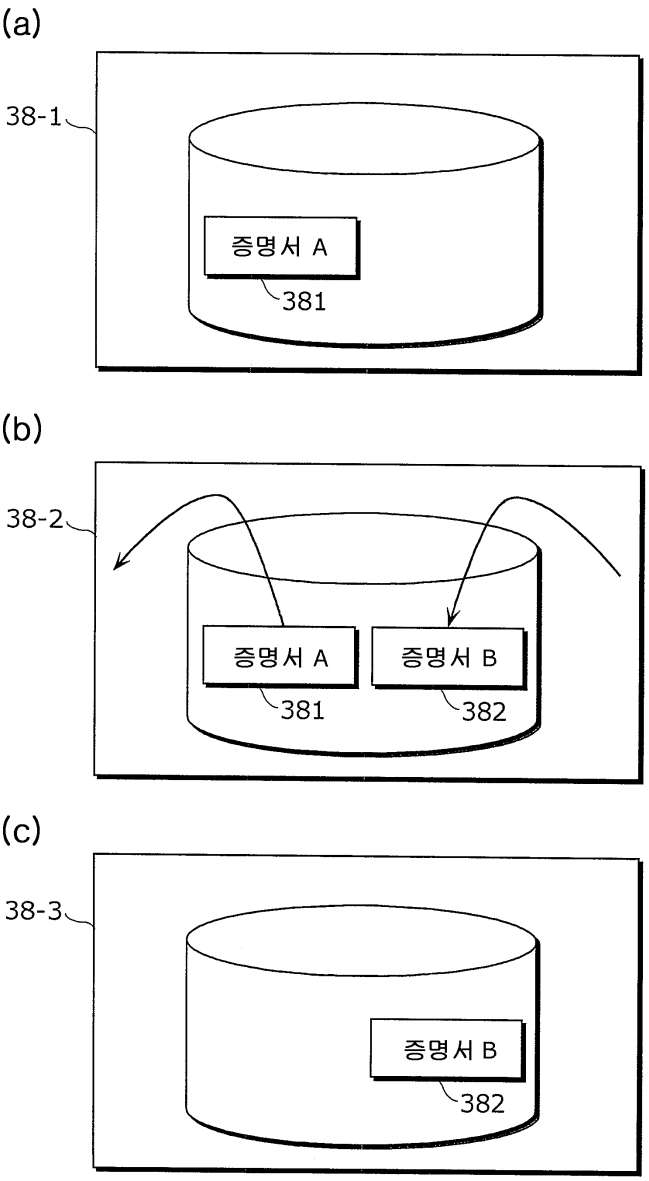
도면36


4
Java 프로그램의 기동전 통지를 수취
S361
증명서 체인으로부터 각 X. 509 증명서 (루트 증명서, 중간 증명서 및 리프 증명서)를 추출
S362
아직 체크되지 않은 X. 509 증명서를 하나 선택
S363
금회 갱신 일시<현재 일시<다음회 갱신 일시?
S364
no
yes
모든 X. 509 증명서가 체크되었는가?
S365
no
yes
인증 성공에 대해 통지
S366
인증 실패에 대해 통지
S367
종료

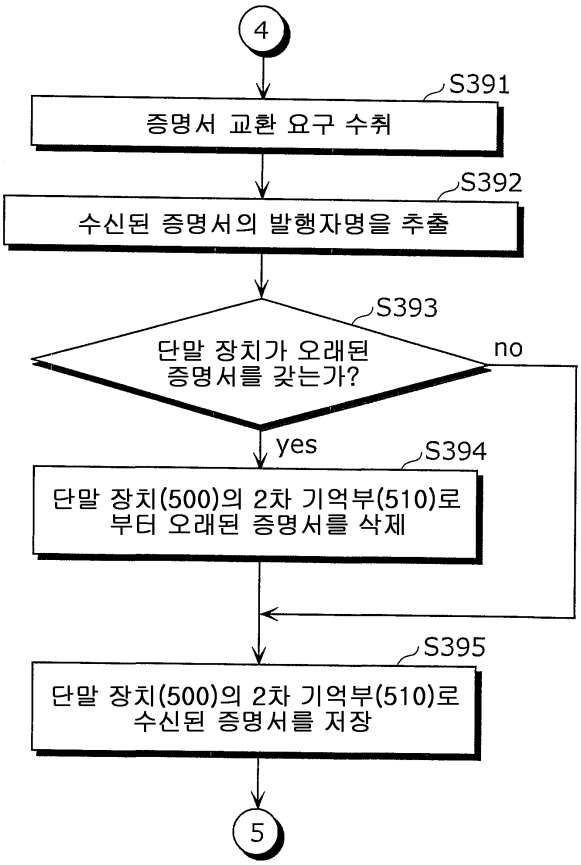
도면37


371
372
37
루트 증명서 A
. . .

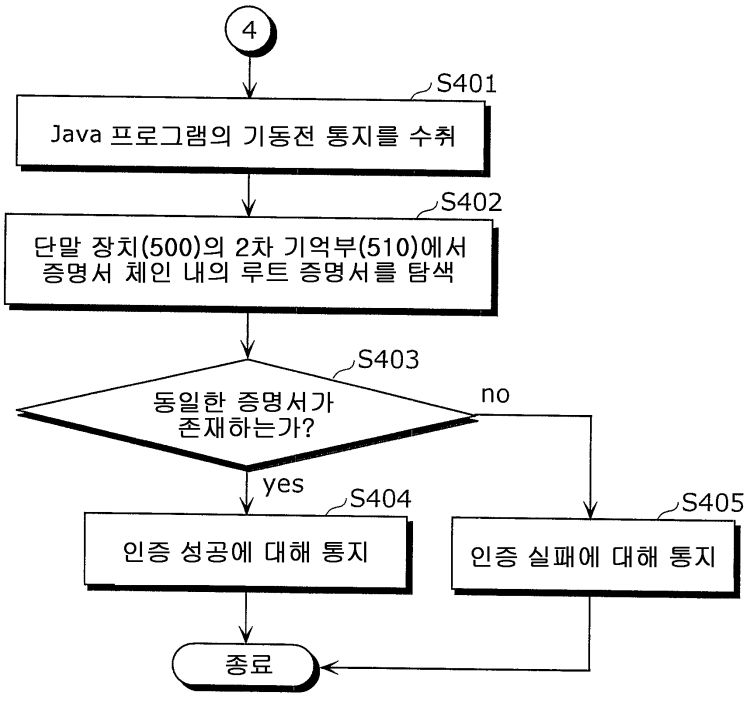
도면38



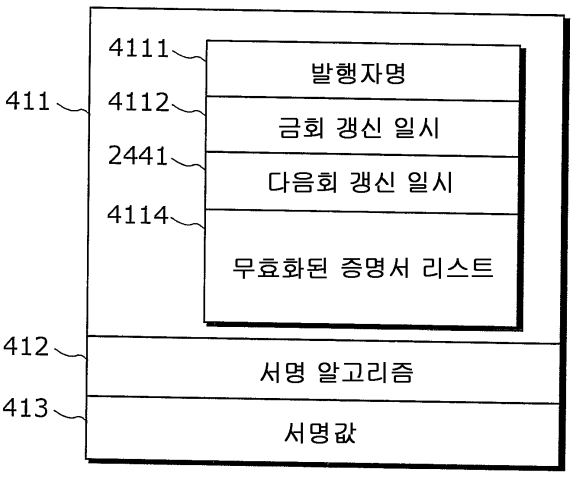
도면39


4
S391
증명서 교환 요구 수취
S392
수신된 증명서의 발행자명을 추출
S393
단말 장치가 오래된 증명서를 갖는가?
no
yes
S394
단말 장치(500)의 2차 기억부(510)로부터 오래된 증명서를 삭제
S395
단말 장치(500)의 2차 기억부(510)로 수신된 증명서를 저장
5

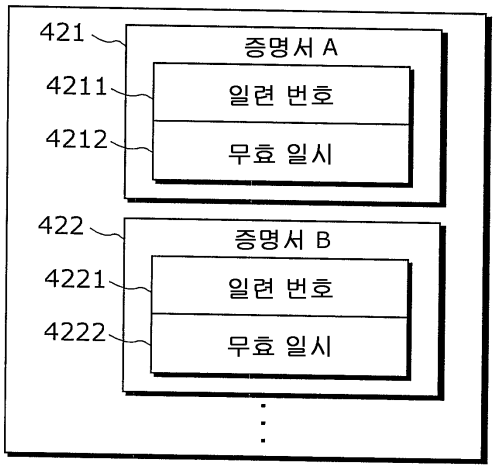
도면40


4
S401
Java 프로그램의 기동전 통지를 수취
S402
단말 장치(500)의 2차 기억부(510)에서 증명서 체인 내의 루트 증명서를 탐색
S403
동일한 증명서가 존재하는가?
no
yes
S404
인증 성공에 대해 통지
S405
인증 실패에 대해 통지
종료

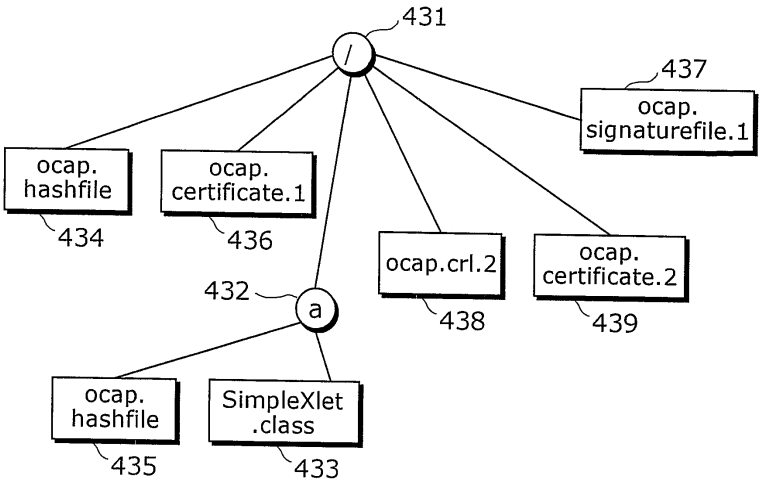
도면41



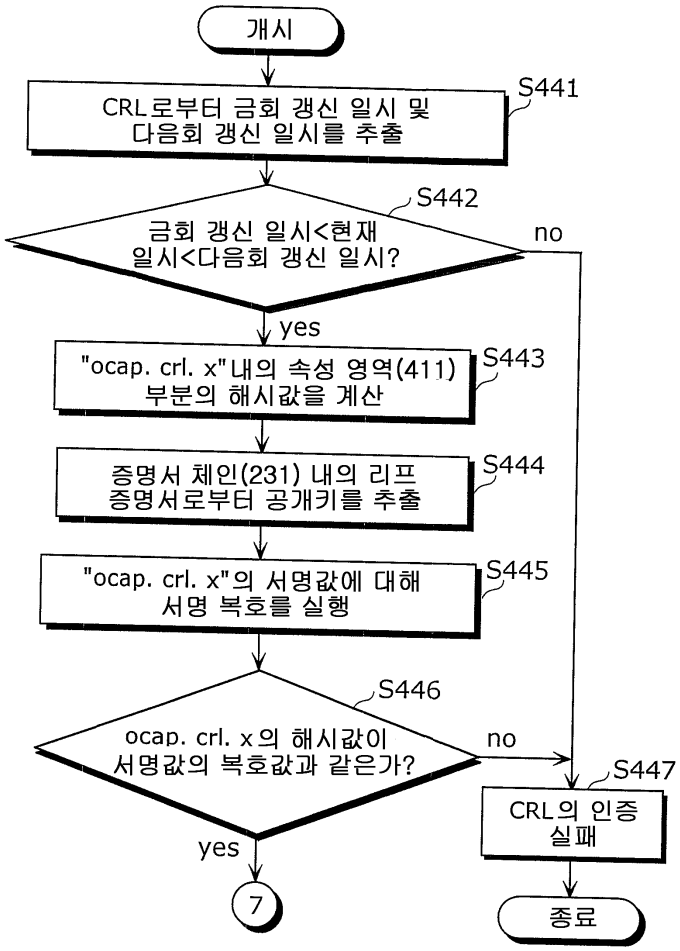
도면42



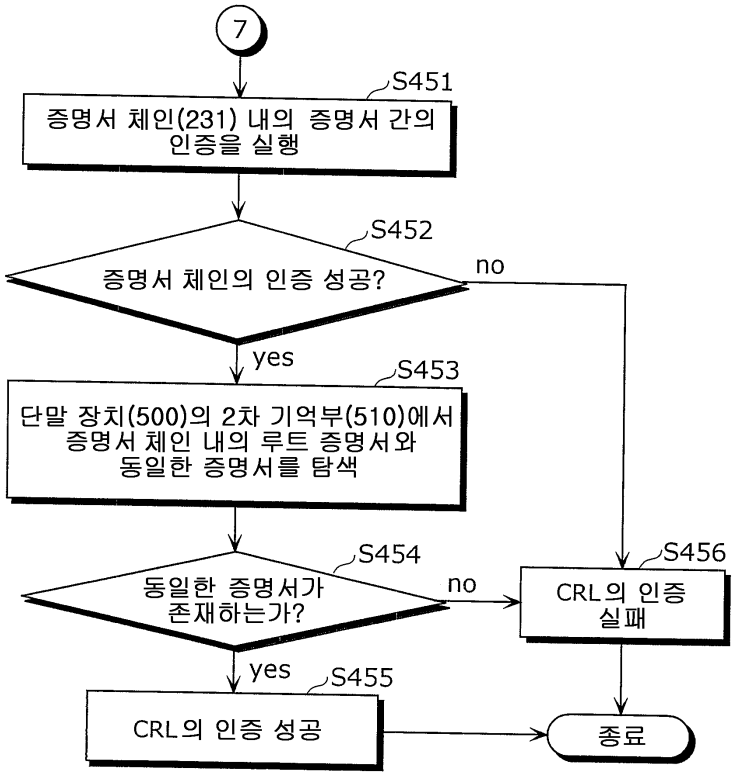
도면43



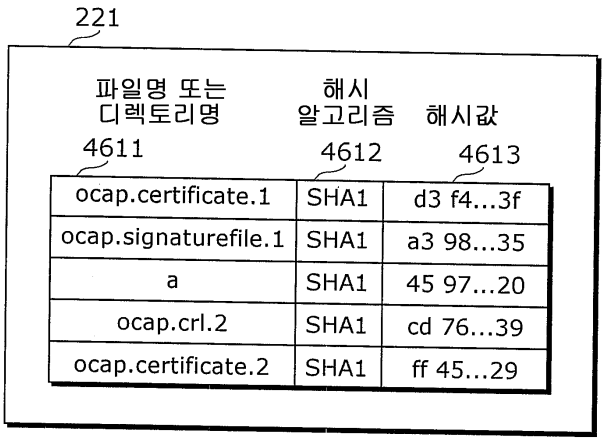
도면44


개시
CRL로부터 금회 갱신 일시 및 다음회 갱신 일시를 추출
S441
금회 갱신 일시<현재 일시<다음회 갱신 일시?
S442
no
yes
"ocap. crl. x"내의 속성 영역(411) 부분의 해시값을 계산
S443
증명서 체인(231) 내의 리프 증명서로부터 공개키를 추출
S444
"ocap. crl. x"의 서명값에 대해 서명 복호를 실행
S445
ocap. crl. x의 해시값이 서명값의 복호값과 같은가?
S446
no
yes
CRL의 인증 실패
S447
7
종료

도면45

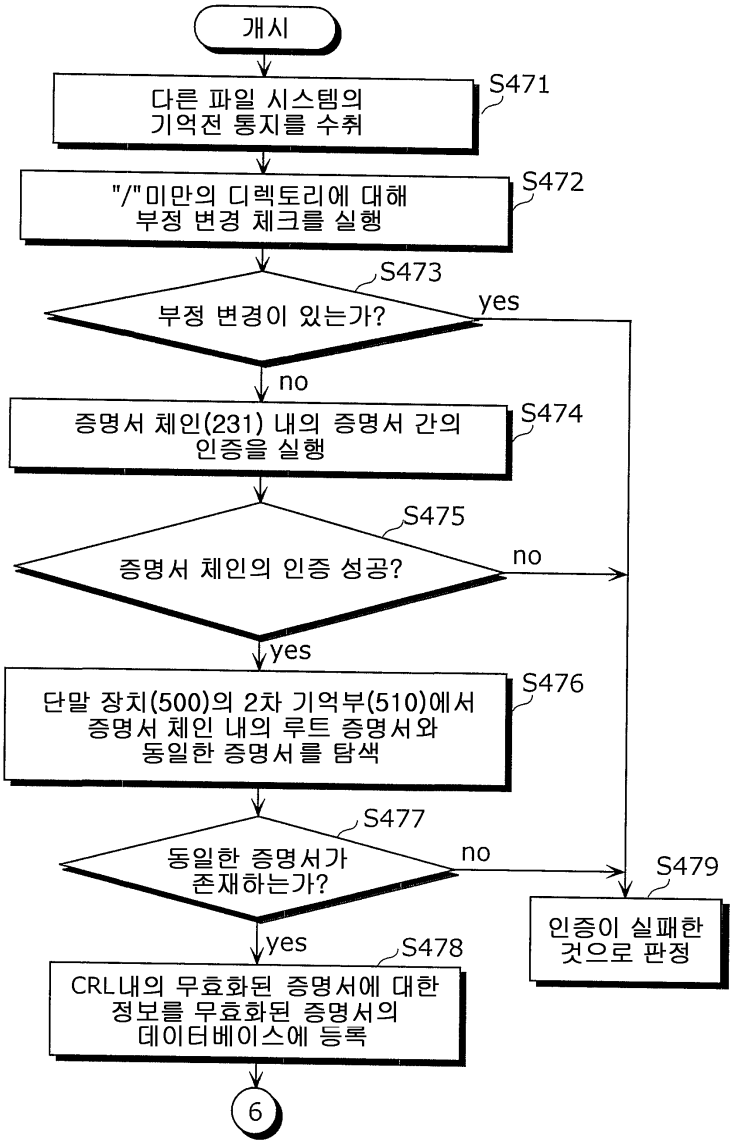


도면46

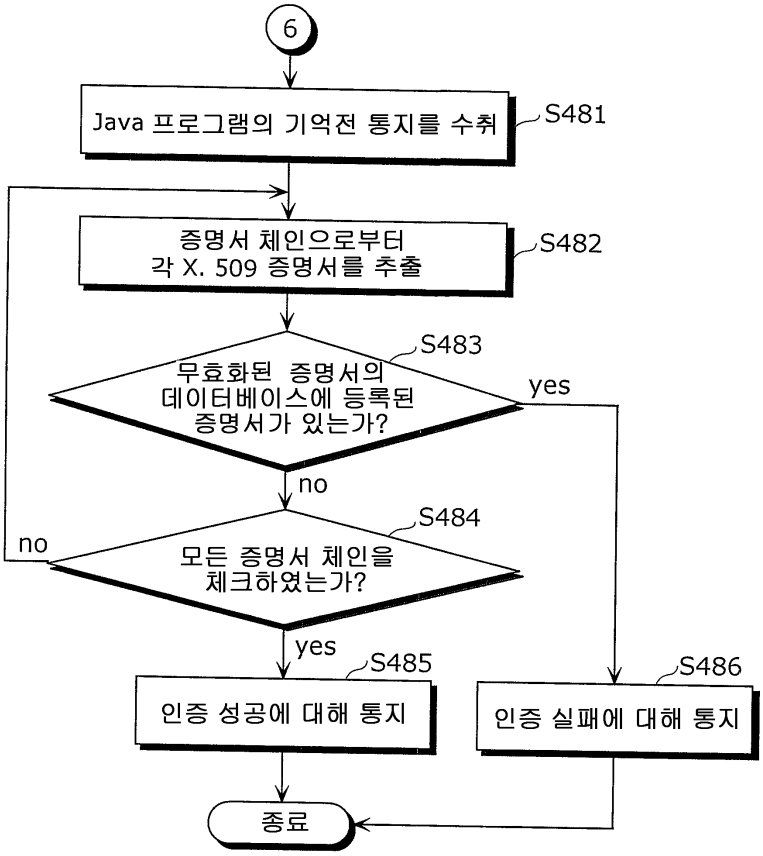


파일명 또는 디렉토리명 (4611)	해시 알고리즘 (4612)	해시값 (4613)
ocap.certificate.1	SHA1	d3 f4...3f
ocap.signaturefile.1	SHA1	a3 98...35
a	SHA1	45 97...20
ocap.crl.2	SHA1	cd 76...39
ocap.certificate.2	SHA1	ff 45...29

도면47


개시
다른 파일 시스템의 기억전 통지를 수취
S471
"/"미만의 디렉토리에 대해 부정 변경 체크를 실행
S472
부정 변경이 있는가?
S473
yes
no
증명서 체인(231) 내의 증명서 간의 인증을 실행
S474
증명서 체인의 인증 성공?
S475
no
yes
단말 장치(500)의 2차 기억부(510)에서 증명서 체인 내의 루트 증명서와 동일한 증명서를 탐색
S476
동일한 증명서가 존재하는가?
S477
no
yes
CRL내의 무효화된 증명서에 대한 정보를 무효화된 증명서의 데이터베이스에 등록
S478
인증이 실패한 것으로 판정
S479
6

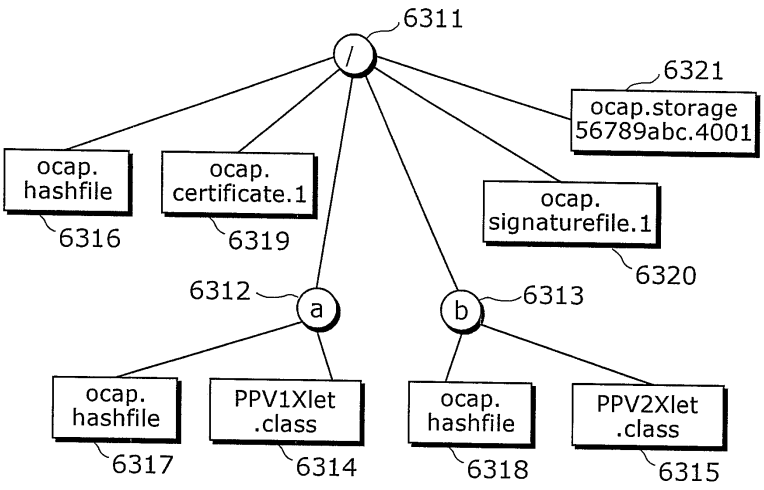
도면48



도면49

발행자명 (491)	일련 번호 (492)	무효 일시 (493)
P	3	2003-06-23 15:00 GMT
S	5	2003-04-12 23:00 GMT
D	1	2002-08-03 09:10 GMT
T	10	2003-12-02 05:00 GMT
K	13	2003-12-04 02:50 GMT

도면50



도면51

```
"-//OCAP//DTD Application Description File 1.0//EN"
"http://www.cablelabs.com/ocap/dtd/applicationdescriptionfile
-1-0.dtd"
<applicationdescription>
   <dir name="/">
      <file name="ocap.hashfile" size="25"/>
      <dir name="a">
         <file name="PPV1Xlet.class" size="1000"/>
      </dir>
      <dir name="b">
         <file name="ocap.hashfile" size="15"/>
         <file name="PPV2Xlet.class" size="1000"/>
      </dir>
   </dir>
</applicationdescription>
```