(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0052321
(43) 공개일자 2009년05월25일

(51) Int. Cl.
G11B 20/10 (2006.01) *G06F 11/30* (2006.01)
G06F 12/14 (2006.01) *H04L 9/32* (2006.01)
(21) 출원번호 10-2009-7002325
(22) 출원일자 2009년02월04일
심사청구일자 없음
번역문제출일자 2009년02월04일
(86) 국제출원번호 PCT/US2007/015431
국제출원일자 2007년06월28일
(87) 국제공개번호 WO 2008/008244
국제공개일자 2008년01월17일
(30) 우선권주장
11/557,049 2006년11월06일 미국(US)
(뒷면에 계속)

(71) 출원인
샌디스크 코포레이션
미합중국, 캘리포니아주 95035, 밀피타스, 맥카시 볼레바드 601
(72) 발명자
홀트즈만, 마이클
미국, 캘리포니아 95014, 쿠퍼티노, 반하트 플레이스 7602
바질라이, 론
이스라엘, 크파-브라딤 25147, 메론 스트리트 67
조강-쿨롬, 패브리스
미국, 캘리포니아 94070, 산 칼로스, 버클랜드 애비뉴 855
(74) 대리인
박경재, 송범엽

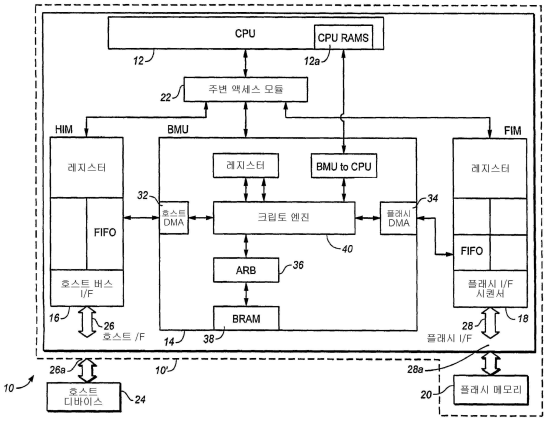
전체 청구항 수 : 총 63 항

(54) 다기능 제어 구조를 이용하는 콘텐트 제어 시스템과 방법

(57) 요 약

적어도 하나의 소프트웨어 애플리케이션은 메모리 장치에 저장되고, 보안 데이터 구조는 장치에 저장된 데이터로부터 얻을 수 있는 정보 및 적어도 하나의 소프트웨어 애플리케이션에 대한 액세스를 제어한다. 한 세트의 프로토콜들은 호스트 및 메모리 장치 사이의 통신을 제어한다. 메모리 장치에 저장된 적어도 하나의 소프트웨어 애플리케이션의 호출은 프로토콜을 변형한다. 보안 데이터 구조는 액세스 정책에 따라 메모리 장치에 저장된 데이터에 대한 액세스를 제어한다. 메모리 장치에 저장된 적어도 하나의 소프트웨어 애플리케이션의 호출은 데이터에 액세스하기 위한 액세스 정책 외에 적어도 하나의 조건을 부과한다. 객체에 액세스는 객체의 데이터를 처리하는 적어도 하나의 소프트웨어 애플리케이션을 호출할 것이다. 다수의 프로토콜들의 제 1 세트들 중 개별 세트들은 데이터가 데이터 객체에 제공 및 저장되게 하기 위하여 선택할 수 있다. 제 2 세트의 프로토콜들은 제 1 세트의 프로토콜들 중 어느 것이 객체에 데이터를 제공 및 저장하게 하기 위하여 사용되었는지에 무관하게, 데이터 객체로부터 데이터, 또는 상기 데이터로부터 유도된 데이터를 검색하기 위하여 사용된다.

대 표 도 - 도1



(30) 우선권주장
11/557,056 2006년11월06일 미국(US)
60/819,507 2006년07월07일 미국(US)

특허청구의 범위

청구항 1

호스트들에 데이터 처리 서비스들을 제공하는 데이터 저장 장치(data storage apparatus)로서,

호스트 중 개별 호스트에 제거 가능하게 접속되도록 구성되고 데이터를 저장할 수 있는 비휘발성 메모리 시스템과,

상기 비휘발성 메모리 시스템에 저장된 보안 데이터 구조(security data structure)와,

상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 소프트웨어 애플리케이션으로서, 상기 적어도 하나의 소프트웨어 애플리케이션은 상기 데이터의 처리를 수행하기 위하여 호스트에 의해 호출되고, 상기 보안 데이터 구조는 상기 데이터로부터 얻을 수 있는 정보 및 적어도 하나의 소프트웨어 애플리케이션에 대한 액세스를 인증 처리를 통하여 상기 메모리 시스템이 접속되어 있는 호스트들 중 하나의 호스트에 의해 제어하는, 적어도 하나의 소프트웨어 애플리케이션을

포함하는, 데이터 저장 장치.

청구항 2

제 1항에 있어서, 상기 데이터는, 상기 적어도 하나의 호스트가 상기 인증 처리에서 인증된 후, 호스트들 중 적어도 하나에 의해 호출된 적어도 하나의 소프트웨어 애플리케이션에 의해 액세스할 수 있는, 데이터 저장 장치.

청구항 3

제 1항에 있어서, 상기 적어도 하나의 소프트웨어 애플리케이션은 상기 정보를 얻기 위하여 상기 데이터 중 적어도 일부 데이터를 처리하는, 데이터 저장 장치.

청구항 4

제 3항에 있어서, 상기 정보는, 적어도 하나의 상기 호스트가 상기 인증 처리에서 인증된 후, 호스트들 중 적어도 하나에 표시될 수 있는, 데이터 저장 장치.

청구항 5

제 4항에 있어서, 상기 보안 데이터 구조는 제 1 및 제 2 제어 구조를 포함하고, 상기 제 1 제어 구조는 적어도 하나의 소프트웨어 애플리케이션과 연관되어 있으며, 상기 제 1 제어 구조는 상기 제 2 제어 구조에 상기 정보에 대한 액세스의 제어를 위임하는 권리를 갖고, 상기 제 2 제어 구조는 상기 인증 처리에 의해 호스트에 의한 상기 정보에 대한 액세스를 제어하는, 데이터 저장 장치.

청구항 6

제 4항에 있어서, 상기 적어도 하나의 호스트는 상기 데이터에 액세스하지 않는, 데이터 저장 장치.

청구항 7

제 3항에 있어서, 상기 데이터는 상기 적어도 하나의 소프트웨어 애플리케이션에 의해 일회용 패스워드(one time pass word)를 생성하기 위한 씨드 값(seed value)을 포함하고, 상기 정보는 상기 일회용 패스워드를 포함하는, 데이터 저장 장치.

청구항 8

제 3항에 있어서, 상기 데이터는 상기 비휘발성 메모리 시스템에 저장되거나 있는 인크립트 콘텐트(encrypted content)에 액세스하기 위한 적어도 하나의 라이센스에 관한 것이고, 상기 정보는 적어도 하나의 라이센스가 유효한지를 표시하는, 데이터 저장 장치.

청구항 9

제 8항에 있어서, 상기 비휘발성 메모리 시스템은 인크립트된 데이터를 저장하고, 상기 보안 데이터 구조는 상

기 정보에 응답하여 상기 인크립트된 데이터의 디크립션(decryption)을 제어하는, 데이터 저장 장치.

청구항 10

제 8항에 있어서, 상기 장치는 상기 비휘발성 메모리 시스템에 저장된 다수의 DRM 소프트웨어 애플리케이션들을 포함하고, 상기 DRM 소프트웨어 애플리케이션들은 상기 데이터를 처리하기 위한 호스트에 의해 선택 가능하고 호출될 수 있는, 데이터 저장 장치.

청구항 11

제 1항에 있어서, 상기 장치는 상기 비휘발성 메모리 시스템에 저장된 다수의 소프트웨어 애플리케이션을 포함하는, 데이터 저장 장치.

청구항 12

제 11항에 있어서, 상기 호스트들은 데이터 처리 요구(data processing request)에 의해 상기 다수의 소프트웨어 애플리케이션을 호출하고, 상기 장치는 상기 호스트들로부터 애플리케이션으로 상기 데이터 처리 요구를 패싱하기 위하여 상기 다수의 소프트웨어 애플리케이션 중 각각의 애플리케이션에 대응하는 통신 채널을 더 포함하고, 상기 보안 데이터 구조는 상기 하나의 애플리케이션에 대응하는 통신 채널을 통하여 상기 호스트 중 적어도 하나의 호스트로부터 애플리케이션 중 하나의 애플리케이션으로 상기 데이터 처리 요구 중 하나를 패싱하기 위하여 통신 채널을 제어하는, 데이터 저장 장치.

청구항 13

제 12항에 있어서, 상기 보안 데이터 구조는, 하나의 데이터 처리 요구가 상기 하나의 애플리케이션에 대응하는 통신 채널에 있을 때, 하나의 데이터 처리 요구에서 정보를 식별할 수 없는, 데이터 저장 장치.

청구항 14

제 12항에 있어서, 상기 보안 데이터 구조는 통신 채널 사이에 혼선이 없도록 통신 채널을 제어하는, 데이터 저장 장치.

청구항 15

제 12항에 있어서, 소프트웨어 애플리케이션 중 적어도 일부 애플리케이션 각각은 상기 각각의 소프트웨어 애플리케이션에 대응하는 통신 채널을 선택하여 호스트에 의한 선택이 가능한, 데이터 저장 장치.

청구항 16

제 15항에 있어서, 통신 채널 중 적어도 일부 통신 채널 각각은 호스트가 상기 인증 처리시 상기 보안 데이터 구조에 의해 인증된 후 호스트에 의한 선택이 가능한, 데이터 저장 장치.

청구항 17

제 12항에 있어서, 상기 보안 데이터 구조는 호스트들에 의해 통신 채널들 중 적어도 하나에 대한 액세스를 제어하는 적어도 하나의 제어 구조를 포함하는, 데이터 저장 장치.

청구항 18

제 1항에 있어서, 상기 보안 데이터 구조와 호스트 사이와, 상기 보안 데이터 구조와 적어도 하나의 소프트웨어 애플리케이션 사이의 인터페이스를 더 포함하는, 데이터 저장 장치.

청구항 19

제 18항에 있어서, 상기 적어도 하나의 소프트웨어 애플리케이션은 상기 보안 데이터 구조에 요구들을 전송하고, 상기 인터페이스는 호스트로부터의 요규와 적어도 하나의 소프트웨어 애플리케이션으로부터의 요구를 구분하지 못하므로, 상기 보안 데이터 구조는 상기 요구가 호스트들로부터 발생하였는지 또는 적어도 하나의 애플리케이션으로부터 발생하였는지 인식하지 못하는, 데이터 저장 장치.

청구항 20

제 1항에 있어서, 호스트들에 의해 액세스할 수 있고 상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 데이터 객체(data object)를 더 포함하고, 상기 적어도 하나의 데이터 객체는 데이터를 저장하고, 상기 적어도 하나의 데이터 객체와 상기 적어도 하나의 소프트웨어 애플리케이션 사이의 적어도 하나의 관계를 더 포함하여, 상기 정보에 대한 요구가 호스트 중 하나에 의해 상기 보안 데이터 구조로 전송될 때, 상기 적어도 하나의 소프트웨어 애플리케이션은 상기 적어도 하나의 관계를 통하여 호출되는, 데이터 저장 장치.

청구항 21

제 20항에 있어서, 상기 비휘발성 메모리 시스템에 저장된 인크립트된 콘텐트를 더 포함하고, 상기 데이터는 디크립션 키 값을 포함하고, 상기 정보는 적어도 하나의 소프트웨어 애플리케이션에 의해 상기 인크립트된 콘텐트와 상기 디크립션 키 값으로부터 얻어진 디크립트된 콘텐트를 포함하는, 데이터 저장 장치.

청구항 22

제 20항에 있어서, 상기 데이터는 상기 적어도 하나의 소프트웨어 애플리케이션에 의해 일회용 패스워드를 생성하기 위한 씨드 값을 포함하고, 상기 정보는 일회용 패스워드를 포함하는, 데이터 저장 장치.

청구항 23

제 1항에 있어서, 사적 키 및 공용 키를 포함하는 키 쌍을 포함하는 객체를 더 포함하고, 적어도 하나의 증명서는 공용 키를 포함하고, 상기 적어도 하나의 소프트웨어 애플리케이션은 공용 키가 진짜인 것을 호스트 중 적어도 하나에 증명하고, 상기 공용 키로 인크립트된 데이터를 얻기 위한 상기 적어도 하나의 증명서를 사용하는, 데이터 저장 장치.

청구항 24

제 1항에 있어서, 상기 비휘발성 메모리 시스템에 저장된 인크립트된 데이터를 더 포함하고, 상기 인크립트된 데이터는 상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 디크립션 키의 값에 의해 디크립트되고, 상기 보안 데이터 구조는 상기 적어도 하나의 디크립션 키의 값에 대한 액세스를 배타적으로 제어하는, 데이터 저장 장치.

청구항 25

제 24항에 있어서, 상기 적어도 하나의 소프트웨어 애플리케이션 및 상기 호스트들은 적어도 하나의 디크립션 키의 값에 액세스할 수 없는, 데이터 저장 장치.

청구항 26

호스트에 데이터 처리 서비스들을 제공하는 데이터 저장 장치로서,

호스트들 중 개별 호스트에 제거 가능하게 접속되도록 구성된 비휘발성 메모리 시스템과,

상기 비휘발성 메모리 시스템에 저장된 보안 데이터로서, 상기 보안 데이터 구조는 메모리 시스템이 접속된 호스트에 의해 비휘발성 메모리 시스템에 저장된 데이터에 대한 액세스를 제어하는, 보안 데이터와,

상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 소프트웨어 애플리케이션으로서, 상기 적어도 하나의 소프트웨어 애플리케이션은 상기 데이터의 처리를 수행하기 위하여 호스트들에 의해 호출되는, 소프트웨어 애플리케이션과,

호스트와 데이터 저장 장치 사이의 통신을 위해 상기 비휘발성 메모리 시스템에 저장된 한 세트의 프로토콜을

포함하고,

상기 프로토콜 중 적어도 하나는 상기 적어도 하나의 소프트웨어 애플리케이션의 호출에 의해 변형되는, 데이터 저장 장치.

청구항 27

제 26항에 있어서, 상기 적어도 하나의 소프트웨어 애플리케이션의 호출은 다른 프로토콜에 의해 적어도 하나의 프로토콜을 대체하는, 데이터 저장 장치.

청구항 28

제 27항에 있어서, 상기 다른 프로토콜은 증명서 취소 방법에 관련되는, 데이터 저장 장치.

청구항 29

호스트에 데이터 처리 서비스를 제공하는 데이터 저장 장치로서,

호스트 중 개별 호스트에 제거 가능하게 접속되도록 구성된 비휘발성 메모리 시스템과,

상기 비휘발성 메모리 시스템에 저장된 보안 데이터 구조로서, 상기 보안 데이터 구조는 비휘발성 메모리 시스템에 저장되거나 저장될 데이터에 대한 액세스를 제어하는, 보안 데이터 구조와,

상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 소프트웨어 애플리케이션으로서, 상기 적어도 하나의 소프트웨어 애플리케이션은 메모리 시스템이 접속되는 호스트 중 하나의 호스트에 의해 호출될 수 있는, 소프트웨어 애플리케이션을

포함하고,

상기 보안 데이터 구조는 액세스 정책을 실행함으로써 데이터에 대한 액세스를 제어하고, 상기 적어도 하나의 소프트웨어 애플리케이션의 호출은 호스트들에 의해 데이터에 대한 액세스에 대한 액세스 정책과 다른 적어도 하나의 부가적인 조건을 부과하는, 데이터 저장 장치.

청구항 30

제 29항에 있어서, 상기 적어도 하나의 부가적인 조건은 라이센스에 관련되는, 데이터 저장 장치.

청구항 31

제 29항에 있어서, 상기 호스트들은 데이터에 대한 액세스가 승인되기 전에 제 1 및 제 2 액세스 정책들 모두에 부합할 필요가 있는, 데이터 저장 장치.

청구항 32

호스트에 데이터 처리 서비스를 제공하는 데이터 저장 장치로서,

호스트 중 개별 호스트에 제거 가능하게 접속되도록 구성된 비휘발성 메모리 시스템과,

상기 비휘발성 메모리 시스템에 저장된 보안 데이터 구조로서, 상기 보안 데이터 구조는 비휘발성 메모리 시스템에 저장되거나 저장될 데이터에 대한 액세스를 제어하는, 보안 데이터 구조와,

상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 소프트웨어 애플리케이션으로서, 상기 적어도 하나의 소프트웨어 애플리케이션은 메모리 시스템이 상기 데이터의 처리를 수행하기 위하여 접속된 호스트에 의해 호출되는, 소프트웨어 애플리케이션과,

상기 비휘발성 메모리 시스템에 저장되고, 상기 데이터 중 적어도 일부를 포함하고, 적어도 하나의 데이터 객체와 상기 적어도 하나의 소프트웨어 애플리케이션 사이의 적어도 하나의 관계를 포함하며, 상기 메모리 시스템이 접속된 호스트에 의해 상기 적어도 하나의 데이터 객체가 액세스될 때, 적어도 하나의 상기 소프트웨어 애플리케이션은 정보를 얻기 위하여 상기 적어도 하나의 데이터 객체의 상기 적어도 일부 데이터를 처리하기 위하여 상기 적어도 하나의 관계를 통하여 호출되는, 적어도 하나의 데이터 객체를

포함하는, 데이터 저장 장치.

청구항 33

제 32항에 있어서, 상기 보안 데이터 구조는 인증 처리시 적어도 하나의 호스트에 의해 비휘발성 메모리 시스템에 저장된 데이터에 대한 액세스를 제어하고, 상기 정보는 적어도 하나의 호스트가 인증 처리를 통하여 상기 보안 데이터 구조에 의해 인증된 후, 상기 적어도 하나의 애플리케이션의 임의의 추가 호출 없이, 적어도 하나의 호스트에 표시되는, 데이터 저장 장치.

청구항 34

제 33항에 있어서, 상기 적어도 하나의 호스트는 상기 적어도 하나의 데이터 객체의 상기 적어도 일부 데이터에 대한 액세스를 가지지 않는, 데이터 저장 장치.

청구항 35

호스트에 데이터 처리 서비스를 제공하는 데이터 저장 장치로서,

호스트의 개별 호스트에 제거 가능하게 접속되도록 구성된 비휘발성 메모리 시스템과,

상기 비휘발성 메모리 시스템에 저장된 보안 데이터 구조로서, 상기 보안 데이터 구조는, 비휘발성 메모리 시스템에 저장되거나 저장될 데이터로부터 얻을 수 있는 정보에 대한 액세스를 메모리 시스템이 접속된 호스트에 의해 제어하기 위하여 제 1 제어 구조를 포함하는, 보안 데이터 구조와,

상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 소프트웨어 애플리케이션으로서, 상기 적어도 하나의 소프트웨어 애플리케이션은, 상기 정보를 얻기 위하여 상기 데이터의 처리를 수행하기 위하여 호스트들에 의해 호출될 수 있고, 상기 보안 데이터 구조는 상기 적어도 하나의 소프트웨어 애플리케이션의 호출을 제어하기 위한 제 2 제어 구조를 포함하며, 상기 제 1 및 제 2 제어 구조들은 동일한 제어 메카니즘을 실질적으로 사용하는, 소프트웨어 애플리케이션을

포함하는, 데이터 저장 장치.

청구항 36

호스트에 데이터 처리 서비스를 제공하는 데이터 저장 장치로서,

호스트 중 개별 호스트에 제거 가능하게 접속되고 데이터를 저장할 수 있도록 구성된 비휘발성 메모리 시스템과,

상기 비휘발성 메모리 시스템에 저장된 보안 데이터 구조와,

상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 데이터 객체와,

상기 비휘발성 메모리 시스템에 저장된 다수의 서로 다른 제 1 세트의 프로토콜로서, 상기 제 1 세트들은 상기 데이터로부터 유도된 유도 데이터 또는 호스트들로부터의 데이터가 상기 보안 데이터 구조의 제어하에서 상기 적어도 하나의 데이터 객체에 제공되고 저장되도록 하기 위하여 메모리 시스템이 접속된 호스트들 중 하나에 의해 개별적으로 선택할 수 있는, 제 1 세트의 프로토콜과,

상기 비휘발성 메모리 시스템에 저장되고 상기 데이터 또는 유도 데이터가 상기 보안 데이터 구조의 제어하에서 적어도 하나의 데이터 객체로부터 검색되도록 하는 제 2 세트의 프로토콜들을

포함하고,

상기 제 2 세트의 프로토콜들은 프로토콜들의 제 1 세트들 중 어느 것이 제공 및 저장되는지에 무관하게 상기 데이터 또는 유도 데이터의 검색을 수행할 수 있는, 데이터 저장 장치.

청구항 37

제 36항에 있어서, 상기 제 1 세트의 프로토콜들 사이의 차이는 상기 비휘발성 메모리 시스템의 인증 또는 상기 데이터의 인크립션에 관련되는, 데이터 저장 장치.

청구항 38

제 36항에 있어서, 상기 비휘발성 메모리 시스템에 저장된 다수의 다른 소프트웨어 애플리케이션들을 더 포함하고, 상기 다른 소프트웨어 애플리케이션들 중 적어도 일부의 각각은 제 1 세트의 프로토콜들 중 하나에 대응하여, 제 1 세트의 프로토콜들 중 하나가 선택되어 상기 데이터 또는 유도 데이터의 제공 및 저장을 수행할 때, 상기 하나의 제 1 세트에 대응하는 소프트웨어 애플리케이션은 상기 데이터 또는 유도 데이터를 처리하기 위하여 호출되는, 데이터 저장 장치.

청구항 39

제 38항에 있어서, 상기 적어도 일부의 다른 소프트웨어 애플리케이션들은 상기 데이터 또는 유도 데이터를 처

리하게 하는, 데이터 저장 장치.

청구항 40

제 38항에 있어서, 상기 데이터 또는 유도 데이터는 다수의 씨드 값들 중 하나를 포함하고, 상기 적어도 일부의 소프트웨어 애플리케이션들 각각은 대응하는 일회용 패스워드를 형성하기 위하여 상기 다수의 씨드 값들 중 대응하는 값을 처리하는, 데이터 저장 장치.

청구항 41

제 38항에 있어서, 상기 비휘발성 메모리 시스템은 인크립트된 데이터를 저장하고, 상기 데이터 또는 유도 데이터는 상기 인크립트된 콘텐트를 디크립트하기 위한 다수의 디크립션 키 중 하나를 포함하고, 상기 적어도 일부 소프트웨어 애플리케이션 중 각각은 상기 인크립트된 콘텐트를 디크립트하기 위한 상기 다수의 디크립션 키들 중 대응하는 키를 사용하는, 데이터 저장 장치.

청구항 42

그 안에 저장된 데이터, 상기 비휘발성 메모리 시스템에 저장된 보안 데이터 구조 및 상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 소프트웨어 애플리케이션을 가진 비휘발성 메모리를 포함하는 장치에 의해 호스트에 데이터 처리 서비스를 제공하는 방법으로서,

호스트에 비휘발성 메모리 시스템을 제거 가능하게 접속하는 단계와,

정보를 얻기 위하여 상기 데이터를 처리하기 위한 상기 적어도 하나의 소프트웨어 애플리케이션을 사용하는 것을 호출하는 단계와,

호스트에 의해 상기 정보에 액세스하는 단계로서, 상기 호출 및 상기 액세스 단계는 보안 데이터 구조에 의해 제어되는, 단계를

포함하는, 데이터 처리 서비스 제공 방법.

청구항 43

제 42항에 있어서, 상기 보안 데이터 구조는 제 1 및 제 2 제어 구조를 포함하고, 상기 제 1 제어 구조는 상기 적어도 하나의 소프트웨어 애플리케이션과 연관되고, 상기 제 2 제어 구조는 호스트에 의해 상기 정보에 대한 액세스를 제어하고, 상기 방법은 상기 제 2 제어 구조에 상기 정보에 대한 액세스의 제어를 위임하는 상기 제 1 제어 구조를 더 포함하는, 데이터 처리 서비스 제공 방법.

청구항 44

제 42항에 있어서, 상기 호스트는 상기 데이터에 액세스할 수 없는, 데이터 처리 서비스 제공 방법.

청구항 45

제 42항에 있어서, 상기 데이터는 씨드 값을 포함하고, 상기 정보는 상기 일회용 패스워드를 포함하고, 상기 호출 단계는 상기 적어도 하나의 소프트웨어 애플리케이션이 씨드로부터 일회용 패스워드를 생성하게 하는, 데이터 처리 서비스 제공 방법.

청구항 46

제 42항에 있어서, 상기 데이터는 비휘발성 메모리 시스템에 저장되거나 저장될 인크립트된 콘텐트에 액세스하기 위한 적어도 하나의 라이센스에 관련되고, 상기 호출 단계는 적어도 하나의 라이센스가 유효한지에 대한 표시를 상기 적어도 하나의 소프트웨어 애플리케이션이 생성하게 하는, 데이터 처리 서비스 제공 방법.

청구항 47

제 46항에 있어서, 상기 비휘발성 메모리 시스템은 인크립트된 데이터를 저장하고, 상기 방법은 상기 인크립트된 데이터의 디크립션이 상기 정보에 응답하여 허용되는지를 상기 보안 데이터 구조가 결정하는 단계를 포함하는, 데이터 처리 서비스 제공 방법.

청구항 48

비휘발성 메모리 장치에 의해 호스트들에 데이터 처리 서비스들을 제공하기 위한 방법으로서,

상기 장치는 다수의 소프트웨어 애플리케이션들을 그 내부에 저장하고,

데이터 소스로부터 호스트들 중 하나를 통하여 비휘발성 메모리 장치에서 데이터를 수신하는 단계와,

상기 하나의 호스트로부터 요구에 응답하여, 비휘발성 메모리 장치의 데이터 객체를 생성하고 데이터 객체에 상기 데이터 및 상기 데이터로부터 유도된 유도 데이터를 저장하기 위하여 상기 다수의 소프트웨어 애플리케이션들 중 제 1 소프트웨어 애플리케이션을 호출하는 단계와,

상기 다수의 소프트웨어 애플리케이션들의 제 2 소프트웨어 애플리케이션과 상기 데이터 객체를 연관시켜, 상기 데이터 객체가 액세스될 때, 상기 제 2 소프트웨어 애플리케이션이 호출되는 단계를

포함하는, 데이터 처리 서비스 제공 방법.

청구항 49

제 48항에 있어서, 상기 제 1 소프트웨어 애플리케이션 및 상기 제 2 소프트웨어 애플리케이션은 동일한 소프트웨어 애플리케이션인, 데이터 처리 서비스 제공 방법.

청구항 50

제 48항에 있어서, 상기 비휘발성 메모리 장치는 호스트들 각각에 제거 가능하게 접속되도록 구성되고, 상기 방법은 비휘발성 메모리 장치 및 호스트들 중 하나를 제거 가능하게 접속하는 단계를 더 포함하는, 데이터 처리 서비스 제공 방법.

청구항 51

제 48항에 있어서, 상기 비휘발성 메모리 장치는 보안 데이터 구조를 포함하고, 상기 보안 데이터 구조는 제 1 및 제 2 제어 구조를 포함하고, 상기 제 1 제어 구조는 상기 제 1 소프트웨어 애플리케이션과 연관되고 상기 데이터 객체에 대한 액세스를 제어하고, 상기 방법은 상기 제 1 제어 구조가 제 2 제어 구조에 상기 데이터 객체에 대한 액세스의 제어를 위임하는 단계를 더 포함하는, 데이터 처리 서비스 제공 방법.

청구항 52

제 51항에 있어서, 상기 호스트들 중 하나를 통하여 상기 데이터 객체를 액세스하는 단계를 더 포함하고, 상기 액세스 단계는 상기 제 2 제어 구조에 의해 제어되는, 데이터 처리 서비스 제공 방법.

청구항 53

제 48항에 있어서, 상기 데이터는 일회용 패스워드를 생성하기 위한 씨드 값에 관련되고, 상기 방법은 상기 제 1 소프트웨어 애플리케이션이 데이터 객체를 생성하고 데이터 객체내에 상기 씨드 값을 저장하는 단계를 더 포함하는, 데이터 처리 서비스 제공 방법.

청구항 54

제 53항에 있어서, 소스와 다른 엔티티에 의해 상기 데이터 객체에 액세스하는 단계를 더 포함하고, 상기 제 2 소프트웨어 애플리케이션은 일회용 패스워드를 생성하기 위하여 씨드 값을 사용하여 데이터 처리를 수행하기 위하여 호출되고, 상기 방법은 엔티티에 상기 일회용 패스워드를 제공하는 단계를 더 포함하는, 데이터 처리 서비스 제공 방법.

청구항 55

제 48항에 있어서, 상기 데이터는 라이센스 객체를 포함하고, 상기 방법의 상기 제 1 소프트웨어 애플리케이션이 데이터 객체를 생성하고 데이터 객체에 디크립션 키를 저장하는 단계를 더 포함하고, 상기 디크립션 키는 비휘발성 메모리 장치에 저장되거나 저장될 인크립트된 콘텐트를 디크립트하기 위하여 사용될 수 있는, 데이터 처리 서비스 제공 방법.

청구항 56

제 55항에 있어서, 상기 라이센스 객체는 디크립션 키를 포함하고, 상기 저장 단계는 데이터 객체의 상기 라이센스 객체에 디크립션 키를 저장하는, 데이터 처리 서비스 제공 방법.

청구항 57

제 55항에 있어서, 상기 비휘발성 메모리 장치는 보안 데이터 구조를 포함하고, 상기 라이센스 객체는 디크립션 키를 포함하지 않고, 상기 방법은 상기 제 1 소프트웨어 애플리케이션이 디크립션 키를 생성하기 위하여 상기 보안 데이터 구조에 요구하는 단계를 더 포함하고, 상기 저장 단계는 데이터 객체에 상기 보안 데이터 구조에 의해 생성된 디크립션 키를 저장하는, 데이터 처리 서비스 제공 방법.

청구항 58

데이터 저장 장치를 사용하여 호스트들에 데이터 처리 서비스들을 제공하기 위한 방법으로서, 상기 데이터 저장 장치는

호스트들 중 개별 하나에 제거 가능하게 접속되고 데이터를 저장할 수 있도록 구성된 비휘발성 메모리 시스템과,

상기 비휘발성 메모리 시스템에 저장된 보안 데이터 구조와,

상기 비휘발성 메모리 시스템에 저장된 적어도 하나의 데이터 객체와,

상기 비휘발성 메모리 시스템에 저장된 다수의 서로 다른 제 1 세트의 프로토콜과,

상기 비휘발성 메모리 시스템에 저장되고 상기 데이터 또는 상기 데이터로부터 유도된 유도 데이터가 상기 보안 데이터 구조의 제어하에서 상기 적어도 하나의 데이터 객체로부터 검색되게 하는 제 2 세트의 프로토콜을

포함하고,

상기 방법은,

메모리 시스템이 접속된 호스트로부터 데이터, 또는 유도 데이터가 상기 보안 데이터 구조의 제어하에서 상기 적어도 하나의 데이터 객체에 제공 또는 저장되게 하는 제 1 세트들 중 하나를 선택하는 단계와,

프로토콜들의 제 1 세트들 중 어느 것이 제공 및 저장되는지에 무관하게 상기 데이터 또는 유도 데이터의 검색을 실행하기 위하여 프로토콜들의 상기 제 2 세트를 사용하는 단계를

포함하는, 데이터 처리 서비스 제공 방법.

청구항 59

제 58항에 있어서, 상기 제 1 세트의 프로토콜들 사이의 차이는 상기 비휘발성 메모리 시스템의 인증 또는 상기 데이터의 인크립션에 관련되는, 데이터 처리 서비스 제공 방법.

청구항 60

제 58항에 있어서, 상기 데이터 저장 장치는 상기 비휘발성 메모리 시스템에 저장된 다수의 다른 소프트웨어 애플리케이션들을 더 포함하고, 적어도 일부의 상기 다른 소프트웨어 애플리케이션들 각각은 제 1 세트의 프로토콜들 중 하나에 대응하여, 프로토콜들의 제 1 세트들 중 하나를 선택하는 단계는 상기 데이터 또는 유도 데이터를 처리하기 위하여 상기 하나의 제 1 세트에 대응하는 소프트웨어 애플리케이션을 호출하는, 데이터 처리 서비스 제공 방법.

청구항 61

제 60항에 있어서, 상기 적어도 일부의 다른 소프트웨어 애플리케이션들은 상기 데이터 또는 유도 데이터를 처리하여 다른 결과들을 형성하는, 데이터 처리 서비스 제공 방법.

청구항 62

제 60항에 있어서, 상기 데이터 또는 유도 데이터는 다수의 씨드 값들 중 하나를 포함하고, 상기 호출된 소프트

웨어 애플리케이션은 대응하는 일회용 패스워드를 형성하기 위하여 상기 다수의 씨드 값들 중 대응하는 하나를 처리하는, 데이터 처리 서비스 제공 방법.

청구항 63

제 60항에 있어서, 상기 비휘발성 메모리 시스템은 인크립트된 데이터를 저장하고, 상기 데이터 또는 유도 데이터는 상기 인크립트된 콘텐트를 디크립트하기 위한 다수의 디크립션 키들 중 하나를 포함하고, 상기 호출된 소프트웨어 애플리케이션은 상기 인크립트된 콘텐트를 디크립트하기 위하여 상기 다수의 디크립션 키들 중 대응하는 하나를 사용하는, 데이터 처리 서비스 제공 방법.

명 세 서

기 술 분 야

<1> 이 출원은 2006년 7월 7일 출원된 미국 예비 출원 60/819,507의 장점을 주장한다.

<2> 이 출원은 2005년 12월 20일 출원된 미국 출원 11/313,870에 관한 것이고; 상기 출원은 2004년 12월 21일 출원된 미국 예비 출원 60/638,804의 장점을 주장한다. 이 출원은 추가로 2005년 12월 20일 출원된 미국 특허 출원 11/314,411에 관한 것이고; 이 출원은 2005년 12월 20일 출원된 미국 특허 출원 11/314,410에 추가로 관한 것이고; 이 출원은 2005년 12월 20일 출원된 미국 특허 출원 11/313,536에 추가로 관한 것이고; 이 출원은 2005년 12월 20일 출원된 미국특허 출원 11/313,538에 관한 것이고; 이 출원은 2005년 12월 20일에 출원된 미국 특허 출원 11/314,055에 관한 것이고; 이 출원은 2005년 12월 20일에 출원된 미국 특허 출원 11/314,052에 관한 것이고; 이 출원은 2005년 12월 20일에 출원된 미국 특허 출원 11/314,053에 관한 것이다.

<3> 본 출원은 2006년 11월 6일 출원된 Holtzman 등에 의한 발명의 명칭이 "Content Control Method Using Certificate Chains"인 미국 출원 11/557,028, 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Content Control System Using Certificate Chains"인 미국 출원 11/557,010, 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Content Control Method Using Revocation Lists"인 미국 출원 11/557,006, 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Content Control System Using Certificate Revocation Lists"인 미국 출원 11/557,026, 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Content Control Method Using Versatile Control Structure"인 미국 출원 11/557,049, 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Content control system Using Versatile Control Structure"인 미국 출원 11/557,056, 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Method for Controlling Information Supplied From Memory Device"인 미국 출원 11/557,052, 2006년 11월 6일 출원된 Holtzman 등에 의한 발명의 명칭이 "System for Controlling Information Supplied From Memory Device"인 미국 출원 11/557,051, 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Control Method Using Identity Objects"인 미국 출원 11/557,041 및 2006년 11월 6일 출원되고 Holtzman 등에 의한 발명의 명칭이 "Control System Using Identity Objects"인 미국 출원 11/557,039에 관한 것이다.

<4> 상술된 출원들은 여기에 전체적으로 나타난 바와 같이 참조로써 여기에 완전히 통합된다.

<5> 본 발명은 일반적으로 메모리 시스템, 특히 다기능 콘텐트 제어 특징들을 가진 메모리 시스템에 관한 것이다.

배 경 기 술

<6> 플래시 메모리 카드들 같은 저장 장치들은 사진들 같은 디지털 콘텐트를 저장하기 위한 선택 저장 매체가 되었다. 플래시 메모리 카드들은 또한 다른 형태의 미디어 콘텐트를 분배하기 위하여 사용될 수 있다. 게다가, 컴퓨터들, 디지털 카메라들, 셀룰러 전화들, 퍼스널 디지털 어시스탄트들(PDA) 및 MP3 플레이어들 같은 미디어 플레이어들 같은 증가하는 다양한 호스트 장치들은 플래시 메모리 카드들에 저장된 미디어 콘텐트를 렌더링하는 능력을 가진다. 따라서 플래시 메모리 카드들뿐 아니라 다른 형태의 모바일 저장 장치들이 디지털 콘텐트를 분배하기 위하여 폭넓게 사용되는 매개물이 될 큰 가능성이 있다.

<7> 다양한 목적들을 위한 스마트 카드들 같은 저장 장치들의 사용이 증가하여, 저장 장치가 많은 제어 및 처리 능력들을 가지는 것이 필요하다. 따라서, 몇몇 애플리케이션들에서, 특정 제어 구조가 저장 장치들에 저장되는 것은 바람직할 수 있다. 이들 제어 구조들은 저장 장치들이 장치들에 저장된 데이터에 대한 액세스를 제어하게 한다. 예를 들어, 스마트 카드들은 은행 계정들 또는 건강 보험 관련 정보에 액세스를 위한 증명서들 같은 은행

관련 정보를 저장하기 위하여 사용된다. 제어 구조들은 상기 정보에 대한 인증되지 않은 액세스를 방지하기 위하여 스마트 카드들에 설치된다. 스마트 카드들 상 정보는 2003년 영국 Rankl 및 Effing, John Wiley & Sons, Ltd에 의한 제 3 편집본인 스마트 카드들 핸드북에서 발견될 수 있다. 현재, 스마트 카드들은 선불 전화 카드들, 은행 카드들 또는 건강 보험 카드들 형태 같은 단일 사용 또는 목적들에 주로 사용되었다.

<8> 다른 애플리케이션들에서, 소프트웨어 애플리케이션들이 저장 장치들에 저장되는 것은 바람직할 수 있다. 이들 소프트웨어 애플리케이션들은 저장 장치들이 장치들에 저장된 데이터를 처리하게 한다. 예를 들어, 자바 카드들로서 공지된 몇몇 스마트 카드들은 은행 서비스들 같은 서비스들을 지원하기 위한 소프트웨어 애플리케이션들을 포함한다. 자바 카드들상 정보는 Sun Developer Network상에서 2003년 5월 29일 공개된 C. Enrique Ortiz에 의한 "An Introduction to Java Card Technology - Part 1."에서 발견될 수 있다. 그들의 설계에 의해, 자바 카드들을 포함하는 스마트 카드들은 데이터, 또는 애플리케이션들, 양쪽 모두에 아닌 액세스를 제어하기 위하여 사용된다.

<9> 상술된 다양한 문제점들 및 문제들로 인해, 저장 및 호스트 장치들에 현재 사용되는 시스템들 중 어느 것도 완전히 만족스럽지 않다. 그러므로 보다 우수한 특성들을 가진 개선된 시스템들을 제공하는 것이 바람직하다.

발명의 상세한 설명

<10> 많은 애플리케이션들에서, 저장 장치들상에 데이터 처리 작용들을 운용하는 것은 바람직하다. 결과적인 시스템은 데이터 처리 임무들 모두가 호스트상에서 실행되는 경우의 해결책들보다 안전하고, 보다 효과적이고 덜 호스트 의존적일 것이다. 일 실시예에서, 적어도 하나의 소프트웨어 애플리케이션은 비휘발성 메모리 시스템에 저장되고, 적어도 하나의 소프트웨어 애플리케이션은 메모리 시스템의 데이터 처리를 수행하기 위하여 호스트들에 의해 호출될 수 있다. 메모리 시스템에 저장된 보안 데이터 구조는 데이터로부터 얻을 수 있는 정보 및 적어도 하나의 소프트웨어 애플리케이션들에 대한 액세스를 제어한다.

<11> 다른 실시예에서, 데이터 저장 장치는 호스트들에 데이터 처리 서비스들을 제공하기 위하여 사용된다. 데이터 저장 장치의 비휘발성 메모리 시스템에 저장된 적어도 하나의 소프트웨어 애플리케이션은 메모리 시스템내 데이터의 처리를 수행하기 위하여 호스트에 의해 호출될 수 있다. 메모리 시스템에 저장된 한 세트의 프로토콜들은 호스트들 및 데이터 저장 장치 사이의 통신을 위한 것이다. 적어도 하나의 소프트웨어 애플리케이션이 호출될 때, 적어도 하나의 프로토콜들은 변형된다. 이런 특성은 호스트들 및 데이터 저장 장치들 사이의 통신이 적어도 하나의 소프트웨어 애플리케이션의 호출에 의해 융통성 있게 제어되게 한다.

<12> 다른 실시예에서, 비휘발성 메모리 시스템에 저장된 보안 데이터 구조는 액세스 정책에 따라 호스트들에 의해 메모리 시스템에 저장된 데이터에 대한 액세스를 제어한다. 메모리 시스템에 저장된 소프트웨어 애플리케이션이 호출될 때, 액세스 정책과 다른 적어도 부가적인 조건은 호스트들에 의하여 데이터에 대한 액세스에 부과된다.

<13> 본 발명의 일 실시예에서, 적어도 하나의 소프트웨어 애플리케이션 및 적어도 하나의 데이터 객체는 비휘발성 메모리 시스템에 저장된다. 적어도 하나의 데이터 객체 및 적어도 하나의 소프트웨어 애플리케이션 사이의 관계는 설정되어, 적어도 하나의 데이터 객체가 액세스될 때, 적어도 하나의 소프트웨어 애플리케이션은 적어도 하나의 데이터 객체의 데이터를 처리하기 위하여 호출된다.

<14> 다른 실시예에서, 다수의 소프트웨어 애플리케이션들은 비휘발성 메모리 장치에 저장된다. 다수의 애플리케이션들의 제 1 소프트웨어 애플리케이션은 소스로부터 수신된 데이터 또는 유도 데이터를 저장하기 위하여 데이터 객체를 생성하기 위한 호스트 요구에 응답하여 호출된다. 데이터 객체는 다수의 소프트웨어 애플리케이션들의 제 2 소프트웨어 애플리케이션과 연관된다. 데이터 객체가 액세스될 때, 제 2 소프트웨어 애플리케이션은 호출된다.

<15> 다른 실시예에서, 적어도 하나의 소프트웨어 애플리케이션은 비휘발성 메모리 시스템에 저장되고 여기서 적어도 하나의 소프트웨어 애플리케이션은 저장된 데이터의 처리를 수행하기 위하여 호스트들에 의해 호출되거나 정보를 얻기 위하여 메모리 시스템에 저장될 수 있다. 메모리 시스템에 저장된 보안 데이터 구조는 상기 정보에 대한 액세스를 호스트들에 의해 제어하기 위한 제 1 제어 구조 및 적어도 하나의 소프트웨어 애플리케이션의 호출을 제어하기 위한 제 2 제어 구조를 포함하고, 제 1 및 제 2 제어 구조들은 동일한 제어 메카니즘을 실질적으로 사용한다.

<16> 본 발명의 일 실시예에서, 적어도 하나의 데이터 객체는 비휘발성 메모리 시스템에 저장된다. 다수의 다른 프로토콜들 중 제 1 세트들은 메모리 시스템에 저장되고, 제 1 세트들의 개별 세트들은 호스트들로부터 데이터 또는

데이터로부터 유도된 유도 데이터가 적어도 하나의 데이터 객체에 제공 및 저장되게 하기 위하여 호스트들에 의해 선택 가능하다. 메모리 시스템에 저장된 제 2 세트의 프로토콜들은 데이터 또는 유도 데이터가 적어도 하나의 데이터 객체로부터 검색되게 한다 제 2 세트의 프로토콜들은 제 1 세트의 어느 프로토콜들이 제공 및 저장될 수 있는지 무관하게 데이터 또는 유도 데이터의 검색을 수행할 수 있다.

<17> 상술된 특징들은 콘텐트 소유자에 대한 제어 및/또는 보호의 보다 큰 기능성을 제공하기 위하여 저장 시스템들에서 개별적으로 사용될 수 있거나, 임의의 결합으로 결합될 수 있다.

<18> 여기에 참조된 모든 특허들, 특허 출원들, 논문들, 책들, 명세서들, 표준들, 다른 공개물들, 도큐먼트들 및 기타 등등은 모든 목적을 위하여 전체적으로 참조로써 통합된다. 통합된 공개물들, 도큐먼트들 또는 본 도큐먼트의 것들 및 텍스트의 임의의 사이에서 용어의 정의 또는 사용의 임의의 불일치 또는 충돌 범위에 대해, 본 도큐먼트의 용어의 사용상 정의는 널리 사용된다.

실 시 예

<62> 도면들은 본 발명의 측면들의 다양한 실시예들의 특징들을 도시한다. 상세한 설명에서 간략화를 위하여, 동일한 구성요소들은 이 애플리케이션에서 동일한 번호들로 표시된다.

<63> 본 발명의 다양한 측면들이 구현되는 예시적인 메모리 시스템은 도 1의 블록도에 의해 도시된다. 도 1에 도시된 바와 같이, 메모리 시스템(10)은 중앙 처리 유니트(CPU)(12), 버퍼 관리 유니트(BMU)(14), 호스트 인터페이스 모듈(HIM)(16) 및 플래시 인터페이스 모듈(FIM)(18), 플래시 메모리(20) 및 주변 액세스 모듈(PAM)(22)을 포함한다. 메모리 시스템(10)은 호스트 인터페이스 버스(26) 및 포트(26a)를 통하여 호스트 장치(24)와 통신한다. NAND 타입일 수 있는 플래시 메모리(20)는 디지털 카메라, 퍼스널 컴퓨터, 퍼스널 디지털 어시스탄트(PDA), MP-3 플레이어 같은 디지털 미디어 플레이어, 셀룰러 전화, 다른 디지털 장치 또는 기구상 셋톱 박스일 수 있는 호스트 장치(24)에 데이터 저장을 제공한다. CPU(12)에 대한 소프트웨어는 플래시 메모리(20)에 저장될 수 있다. FIM(18)은 플래시 인터페이스 버스(28) 및 포트(28a)를 통하여 플래시 메모리(20)에 접속한다. HIM(16)은 호스트 장치에 접속을 위해 적당하다. 주변 액세스 모듈(22)은 CPU(12)와 통신을 위한 FIM, HIM 및 BMU 같은 적당한 제어기 모듈을 선택한다. 일 실시예에서, 점선 박스내 시스템(10)의 모든 구성요소들은 메모리 카드 또는 스틱(10') 같은 단일 유니트에 밀봉되고 바람직하게 캡슐화된다. 메모리 시스템(10)은 호스트 장치(24)에 탈착 가능하게 접속되어, 시스템(10)의 콘텐트는 많은 다른 호스트 장치들 각각에 액세스될 수 있다.

<64> 하기 설명에서, 메모리 시스템(10)은 메모리 장치(10)라 불리거나 간단히 메모리 장치 또는 장치라 한다. 본 발명이 플래시 메모리들을 참조하여 여기에 도시되었지만, 본 발명은 자기 디스크들, 광학 CD들 같은 다른 타입의 메모리들뿐 아니라, 모든 다른 타입의 재기입 가능 비휘발성 메모리 시스템들에 응용할 수 있다.

<65> 버퍼 관리 유니트(14)는 호스트 다이렉트 메모리 액세스(HDMA)(32), 플래시 다이렉트 메모리 액세스(FDMA)(34), 조정자(36), 버퍼 랜덤 액세스 메모리(BRAM)(38) 및 크립토-엔진(40)을 포함한다. 조정자(36)는 공유된 버스 조정자이므로, 단지 하나의 마스터 또는 이니시에이터(HDMA 32, FDMA 34 또는 CPU 12일 수 있는)는 언제나 활성화될 수 있고 슬레이브 또는 타켓은 BRAM(38)이다. 조정자는 적당한 이니시에이터 요구를 BRAM(38)에 채널링한다. HDMA(32) 및 FDMA(34)는 HIM(16), FIM(18) 및 BRAM(38) 또는 CPU 랜덤 액세스 메모리(CPU RAM)(12a) 사이에서 데이터가 전달되게 한다. HDMA(32) 및 FDMA(34)의 동작은 일반적이고 여기에 상세히 기술되지 않는다. BRAM(38)은 호스트 장치(24) 및 플래시 메모리(20) 사이에서 통과된 데이터를 저장하기 위하여 사용된다. HDMA(32) 및 FDMA(34)는 HIM(16)/FIM(18) 및 BRAM(38) 또는 CPU RAM(12a) 사이에서 데이터를 전달하고 섹터 완료를 표시한다.

<66> 일 실시예에서, 메모리 시스템(10)은 인크립션 및/또는 디크립션에 사용되는 키 값(들)을 생성하고, 여기서 이 값(들)은 실질적으로 바람직하게 호스트 장치(24) 같은 외부 장치에 액세스할 수 없다. 선택적으로, 키 값은 라이센스 서버에 의한 것과 같은 시스템(10) 외부에 생성되고, 시스템(10)에 전송된다. 키 값이 생성되는 방법에 무관하게, 키 값이 시스템(10)에 저장되면, 유일하게 인증된 엔티티들은 키 값에 액세스할 것이다. 그러나, 인크립션 및 디크립션은 통상적으로 파일 단위로 행해지는데, 그 이유는 호스트 장치가 파일들 형태로 메모리 시스템(10)에 데이터를 독출 및 기입하기 때문이다. 많은 다른 타입의 저장 장치들과 같이, 메모리 장치(10)는 파일들을 관리하지 않는다. 파일들의 논리 어드레스들이 식별되는 파일 할당 테이블(FAT)을 메모리(20)가 저장하는 동안, FAT는 제어기(12)가 아닌 호스트 장치(24)에 의해 액세스되고 관리된다. 그러므로, 특정 파일의 데이터를 인크립트하기 위하여, 제어기(12)는 메모리(20)의 파일내 데이터의 논리 어드레스들을 전송하기 위한 호스트 장치에 의존하여야 하고, 그러므로 특정 파일의 데이터는 시스템(10)에만 이용할 수 있는 키 값(들)을 사용

하여 시스템(10)에 의해 발견 및 인크립트되고 및/또는 디크립트될 수 있다.

<67> 호스트 장치(24) 및 메모리 시스템(10) 모두가 파일들의 데이터를 암호적으로 처리하기 위해 동일한 키(들)를 참조하는 것을 다루기 위하여, 호스트 장치는 시스템(10)에 의해 생성되거나 전송된 각각의 키에 대한 레퍼런스를 제공하고, 여기서 상기 레퍼런스는 간단히 키 ID일 수 있다. 따라서, 호스트(24)는 키 ID를 가진 시스템(10)에 의해 암호적으로 처리된 각각의 파일을 연관시키고, 시스템(10)은 호스트에 의해 제공된 키 ID를 가진 데이터를 암호적으로 처리하기 위하여 사용된 각각의 키 값을 연관시킨다. 따라서, 이것은 데이터가 암호적으로 처리되는 것을 호스트가 요구할 때, 메모리(20)로부터 인출되거나 저장될 데이터의 논리 어드레스들과 함께 키 ID를 가진 요구를 시스템(10)에 전송할 것이다. 시스템(10)은 키 값을 생성 또는 수신하고 상기 값을 호스트(24)에 의해 제공된 키 ID와 연관시키고, 암호화 처리를 수행한다. 이런 방식으로, 키 값(들)에 배타적 액세스를 포함하는 키(들)을 사용하여 암호화 처리를 완벽하게 제어하면, 메모리 시스템(10)이 동작하는 방식에 대한 변화가 필요하지 않다. 다른 말로, 일단 키 값이 시스템(10)에 의해 저장되거나 생성되면, 시스템은 FAT의 배타적 제어에 의해 호스트(24)가 파일들을 계속 관리하게 하고, 상기 시스템은 암호화 처리에 사용된 키 값(들)의 관리를 위한 배타적 제어를 유지한다.

<68> 호스트(24)에 의해 제공된 키 ID 및 메모리 시스템에 전송되거나 생성된 키 값은 실시예들 중 하나의 "콘텐트 인크립션 키" 또는 CEK로서 하기에 참조되는 품질의 두 개의 속성들을 형성한다. 호스트(24)가 하나 또는 그 이상의 파일들과 각각의 키 ID를 연관시키는 동안, 호스트(24)는 구성되지 않은 데이터 또는 임의의 방식으로 구성된 데이터와 각각의 키 ID를 연관시킬 수 있고, 완전한 파일들로 구성된 데이터로 제한되지 않는다.

<69> 사용자 또는 애플리케이션이 시스템(10)의 보호되는 콘텐트 또는 영역에 대한 액세스를 얻도록 하기 위하여, 시스템(10)이 미리 저장된 증명서를 사용하여 인증될 필요가 있다. 증명서는 상기 증명서를 가진 특정 사용자 또는 애플리케이션에 승인된 액세스 권리들에 결합된다. 사전 등록 처리시, 시스템(10)은 사용자 또는 애플리케이션의 신원 및 증명서의 기록을 저장하고, 액세스 권리들은 사용자 또는 애플리케이션에 의해 결정되고 호스트(24)를 통하여 제공된 상기 신원 및 증명서와 연관된다. 사전 등록이 완료된 후, 사용자 또는 애플리케이션이 데이터를 메모리(20)에 기입하기를 요구할 때, 호스트 장치를 통하여 신원 및 증명서, 데이터를 인크립트하기 위한 키 ID, 및 인크립트된 데이터가 저장된 논리 어드레스들을 제공할 필요가 있을 것이다. 시스템(10)은 키 값을 생성 또는 수신하고 호스트 장치에 의해 제공된 키 ID와 상기 값을 연관시키고, 이런 사용자 또는 애플리케이상 레코드에 기입될 데이터를 인크립트하기 위하여 사용되는 키 값에 대한 키 ID를 저장한다. 그 다음 데이터를 인크립트하고 생성 또는 수신된 키 값뿐 아니라 호스트에 의해 설계된 어드레스들의 인크립트된 데이터를 저장한다.

<70> 사용자 또는 애플리케이션이 메모리(20)로부터 인크립트된 데이터를 독출하기를 요구할 때, 신원 및 증명서, 요구된 데이터를 인크립트하기 위하여 이전에 사용된 키에 대한 키 ID, 및 인크립트된 데이터가 저장된 논리 어드레스를 제공할 필요가 있을 것이다. 그들이 매칭하면, 시스템(10)은 메모리로부터 사용자 또는 애플리케이션에 의해 제공된 키 ID와 연관된 키 값을 인출하고, 키 값을 사용하여 호스트 장치에 의해 설계된 어드레스들에 저장된 데이터를 디크립트하고 사용자 또는 애플리케이션에 디크립트된 데이터를 전송할 것이다.

<71> 암호화 처리에 사용된 키들의 관리로부터 인증 증명서들을 분리함으로써, 증명서들을 공유하지 않고 데이터에 액세스하는 권리들을 공유하는 것은 가능하다. 따라서, 다른 증명서들을 가진 사용자들 또는 애플리케이션들의 그룹은 동일한 데이터에 액세스하기 위하여 동일한 키들에 액세스하고, 상기 그룹 외측 사용자들은 액세스할 수 없다. 그룹 내 모든 사용자들 또는 애플리케이션들이 동일한 데이터에 액세스할 수 있는 동안, 그들은 다른 권리들을 여전히 가질 수 있다. 따라서, 몇몇은 단지 독출 전용 액세스를 가지며, 다른 것들은 기입 액세스만을 가지며, 다른 것들은 모두를 가질 수 있다. 시스템(10)이 사용자들 또는 애플리케이션 신원들 및 증명서들의 레코드, 그들이 액세스하는 키 ID들, 및 키 ID들의 각각에 연관된 액세스 권리들을 유지하기 때문에, 시스템(10)이 적당한 인증된 호스트 장치에 의해 제어되는 바와 같이, 키 ID들을 부가하거나 삭제하고, 하나의 사용자 또는 애플리케이션이 다른 사용자 또는 애플리케이션으로 액세스 권리들을 위임하거나, 심지어 사용자들 또는 애플리케이션들에 대한 레코드들 또는 테이블들을 삭제 또는 부가하는 것은 가능하다. 저장된 레코드는 구조 채널이 특정 키들에 액세스하기 위하여 필요한 것을 지정할 수 있다. 인증은 대칭 또는 비대칭 알고리즘뿐 아니라 패스워드들을 사용하여 수행될 수 있다.

<72> 특히 메모리 시스템(10)에 보안된 콘텐트의 휴대성은 중요하다. 키 값에 대한 액세스가 메모리 시스템에 의해 제어되는 실시예들에서, 메모리 시스템 또는 상기 시스템을 통합한 저장 장치가 하나의 외부 시스템으로부터 다른 시스템으로 전달될 때, 내부에 저장된 콘텐트의 보안성은 유지된다. 키가 메모리 시스템에 의해 생성되든 메

모리 시스템 외측으로부터 발생하든, 외부 시스템들은 만약 그들이 메모리 시스템에 의해 완전히 제어되는 방식으로 인증되지 않으면 시스템(10)의 콘텐트에 액세스할 수 없다. 심지어 이렇게 인증된 후, 액세스는 메모리 시스템에 의해 전체적으로 제어되고, 외부 시스템들은 메모리 시스템의 미리 설정된 레코드들에 따라 제어되는 방식으로만 액세스할 수 있다. 만약 요구가 상기 레코드들에 부합하지 않으면, 요구는 거부될 것이다.

<73> 보호 콘텐트에서 보다 큰 융통성을 제공하기 위하여, 파티션들로서 하기에 언급되는 메모리의 특정 영역이 적당하게 인증된 사용자들 또는 애플리케이션들에 의해서만 평가될 수 있다는 것이 계획된다. 키 바탕 데이터 인크립션의 상기 기술된 특징들과 결합될 때, 시스템(10)은 보다 큰 데이터 보호 능력을 제공한다. 도 2에 도시된 바와 같이, 플래시 메모리(20)는 다수의 파티션들로 분할된 저장 능력을 가질 수 있다: 사용자 영역 또는 파티션 및 거스텀(custom) 파티션들. 사용자 영역 또는 파티션(P0)은 인증 없이 모든 사용자들 및 애플리케이션들에 액세스할 수 있다. 사용자 영역에 저장된 데이터의 모든 비트 값들이 임의의 애플리케이션 또는 사용자에 의해 독출되거나 기입되는 동안, 만약 독출된 데이터가 인크립트되면, 디크립트 없이 사용자 또는 애플리케이션은 사용자 영역에 저장된 비트 값들에 의해 표현된 정보에 액세스할 수 없다. 이것은 예를 들어 사용자 영역(P0)에 저장된 파일들(102 및 104)에 의해 도시된다. 사용자 영역에는 모든 애플리케이션들 및 사용자들에 의해 독출되고 이해될 수 있는 106 같은 인크립트되지 않은 파일들이 저장된다. 따라서, 심볼적으로, 인크립트된 파일들은 파일들(102 및 104) 같은 처리와 연관된 록들이 도시된다.

<74> 사용자 영역(P0)의 인크립트된 파일이 인증되지 않은 애플리케이션들 또는 사용자들에 의해 이해될 수 없는 동안, 상기 애플리케이션들 또는 사용자들은 몇몇 애플리케이션들에 바람직하지 않을 수 있는 파일을 삭제 또는 파손할 수 있다. 이런 목적을 위하여, 메모리(20)는 이전 인증 없이 액세스될 수 없는 파티션들(P1 및 P2) 같은 보호된 커스텀 파티션들을 포함한다. 이런 애플리케이션에서 실시예들에서 허용된 인증 처리는 하기에서 설명된다.

<75> 도 2에 도시된 바와 같이, 다양한 사용자들 또는 애플리케이션들은 메모리(20)의 파일들에 액세스할 수 있다. 따라서 사용자들(1 및 2) 및 애플리케이션들(1-4)(장치들 동조)은 도 2에 도시된다. 이들 엔티티들이 메모리(20)의 보호된 콘텐트에 액세스되기 전에, 상기 엔티티들은 하기 설명된 방식으로 인증 처리에 의해 우선 인증된다. 이런 처리에서, 액세스를 요구하는 엔티티는 임무 바탕 액세스 제어를 위하여 호스트측에서 식별될 필요가 있다. 따라서, 엔티티 요구 액세스는 우선 "I am application 2 and I wish to read file 1" 같은 정보를 제공하여 그 자체를 식별한다. 그 다음 제어기(12)는 신원, 인증 정보 및 메모리(20) 또는 제어기(12)에 저장된 레코드에 대한 요구를 매칭한다. 만약 모든 요구들이 부합하면, 그 다음 액세스는 상기 엔티티에 대해 승인된다. 도 2에 도시된 바와 같이, 사용자(1)는 파티션(p1)의 파일(101)로부터 독출 및 기입되지만, 제한되지 않은 권리들을 가진 사용자(1)가 P0에서 파일들(106)로부터 독출되고 기입되는 것 외에 파일들(102 및 104)만을 독출할 수 있다. 다른 한편 사용자(2)는 파일(101 및 104)에 액세스하는 것이 허용되지 않고 파일(102)을 독출 및 기입 액세스를 가진다. 도 2에 도시된 바와 같이, 사용자들(1 및 2)은 동일한 로그인 알고리즘(AES)을 가지며, 애플리케이션들(1 및 3)은 사용자(1 및 2)와 다른 로그인 알고리즘들(예를 들어 RSA 및 001001)을 가진다.

<76> 보안 저장 애플리케이션(SSA)은 메모리 시스템(10)의 보안 애플리케이션이고, 상술된 많은 특징들을 실행하기 위하여 사용될 수 있는 본 발명의 실시예를 도시한다. SSA는 CPU(12)의 메모리(20) 또는 비휘발성 메모리(도시되지 않음)에 저장된 데이터베이스를 가진 소프트웨어 또는 컴퓨터 코드로서 구현될 수 있고, RAM(12a)에 독출되고 CPU(12)에 의해 실행된다. SSA에 대한 참조에 사용되는 두문자어는 하기 테이블에 제공된다:

<77> 정의, 두문자어 및 생략어

<78>

ACR	액세스 제어 레코드들
AGP	ACR 그룹
CBC	체인 블록 암호
CEK	콘텐트 인크립션 키
ECB	전자 코드북
ACAM	ACR 속성들 관리
PCR	허용 제어 레코드
SSA	보안 저장 애플리케이션
엔티티	SSA에 로그인하고 그 기능들을 사용하는 실제 및 개별 존재(호스트 측)를 가진 임의의 것

<79> SSA 시스템 설명

<80> 데이터 보안, 보전 및 액세스 제어는 SSA의 주요 임무들이다. 데이터는 몇몇 종류의 대량 저장 장치에서 명백히 저장되는 파일들이다. SSA 시스템은 저장 시스템의 정상에 자리하고 저장된 호스트 파일들에 대한 보안 층을 부가하고, 하기 기술된 보안 데이터 구조들을 통하여 보안 기능들을 제공한다.

<81> SSA의 주 임무는 메모리의 저장된(및 보안된) 콘텐트와 연관된 다른 권리들을 관리하는 것이다. 메모리 애플리케이션은 다중 저장된 콘텐트에 대한 콘텐트 권리들 및 다중 사용자들을 관리할 필요가 있다. 그 측면에서 호스트 애플리케이션들은 상기 애플리케이션들에 가시적인 드라이브들 및 파티션들, 및 저장 장치상 저장된 파일들의 위치들을 관리 및 묘사하는 파일 할당 테이블들(FAT)을 안다.

<82> 저장 장치가 파티션들에 분할된 NAND 플래시 칩을 사용하는 경우, 비록 다른 모바일 저장 장치들이 사용될 수 있지만 본 발명의 범위내에 있다. 이들 파티션들은 연속적인 논리 어드레스들의 스레드들(thread)이고, 여기서 시작 및 종료 어드레스는 경계들을 정의한다. 제한들은 그러므로 만약 목표되면 상기 경계들 내 어드레스들 사이 제한들을 연관시키는 소프트웨어(메모리 20내에 저장된 소프트웨어 같은)에 의해 파티션들을 숨기기 위하여 액세스상에 부과될 수 있다. 파티션들은 이것에 의해 관리되는 논리 어드레스 경계들에 의해 SSA에 대해 완전히 인식할 수 있다. SSA 시스템은 인증되지 않은 호스트 애플리케이션들로부터 보안 데이터를 물리적으로 보안하기 위한 파티션들을 사용한다. 호스트에 대해, 파티션들은 어느 데이터 파일들을 저장하기 위하여 특성 공간들을 정의하는 메카니즘이다. 이들 파티션들은 저장 장치에 액세스하는 누군가가 볼 수 있고 장치상 파티션의 존재를 인식하는 공용이거나, 선택된 호스트 애플리케이션들만이 저장 장치에서 존재에 액세스하고 인식할 수 있는 사적 또는 숨겨진 것일 수 있다.

<83> 도 3은 메모리의 파티션들을 도시하는 메모리의 개략도이다: P0, P1, P2 및 P3(명백하게 4개 보다 적거나 많은 파티션들이 사용될 수 있다), 여기서 P0는 인증 없이 임의의 엔티티에 의해 액세스될 수 있는 공용 파티션이다.

<84> 사적 구획부(P1, P2, 또는 P3 같은)은 그 내부의 파일들에 액세스를 숨긴다. 파티션에 호스트가 액세스하는 것을 막음으로써, 플래시 장치(예를 들어, 플래시 카드)는 파티션 내부 데이터 파일들의 보호를 전달한다. 이런 종류의 보호는 파티션내의 논리 어드레스들에 저장된 데이터에 액세스 제한을 부과함으로써 숨겨진 파티션에 존재하는 모든 파일들을 숨긴다. 다른 말로, 제한들은 논리 어드레스들의 범위와 연관된다. 상기 파티션에 액세스하는 모든 종류의 사용자들/호스트들은 내부의 모든 파일들에 대한 제한되지 않은 액세스를 가질 것이다. 서로로부터 다른 파일들을 분리하기 위하여 - 또는 파일들의 그룹들 - SSA 시스템은 키들 및 키 레퍼런스들 또는 키 ID들을 사용하여 다른 레벨의 파일 당 보안 및 보전을 제공한다 - 또는 파일들의 그룹 -. 다른 메모리 어드레스들에서 데이터를 인크립트하기 위하여 사용된 특정 키 값의 키 레퍼런스 또는 키 ID는 인크립트된 데이터를 포함하는 컨테이너 또는 도메인과 유사해질 수 있다. 이런 이유 때문에, 도 4에서, 키 레퍼런스들 또는 키 ID들(예를 들어, "키 1" 및 "키 2")은 키 ID들과 연관된 키 값들을 사용하여 인크립트된 파일들을 둘러싸는 영역들처럼 그래프적으로 도시된다.

<85> 도 4를 참조하여, 예를 들어, 파일 A는 그것이 임의의 키 ID에 의해 동봉되지 않는 것으로 도시되기 때문에, 임의의 인증 없이 모든 엔티티들에 액세스할 수 있다. 공용 파티션내 파일 B 가 모든 엔티티들에 의해 독출 또는 과기입될 수 있지만, ID를 가진 키, "키 1"로 인크립트된 데이터를 포함하여, 파일 B에 포함된 정보는 상기 엔티티가 상기 키에 액세스하지 않으면 엔티티에 액세스할 수 있다. 이런 방식으로 키 값들 및 키 레퍼런스들 또는 키 ID들을 사용하는 것은 상술된 파티션에 의해 제공된 보호 타입과 반대일 때만 논리 보호를 제공한다. 따라서, 파티션(공용 또는 사적)에 액세스할 수 있는 임의의 호스트는 인크립트된 데이터를 포함하는 전체 파티션의 데이터를 독출 또는 기입한다. 그러나, 데이터가 인크립트되기 때문에, 인증되지 않은 사용자들은 그것을 전와시킬 수 있다. 그들은 바람직하게 검출 없이 데이터를 변경할 수 없다. 인크립션 및/또는 디크립션 키들에 액세스를 제한하지 않음으로써, 이런 특징은 인증된 엔티티들이 데이터를 사용하게 한다. 파일들(B 및 C)은 R0의 키 ID "키2"를 가진 키를 사용하여 인크립트된다.

<86> 데이터 신뢰성 및 보전은 콘텐트 인크립션 키들(CEK)을 사용하는 대칭 인크립션 방법들을 통하여 제공될 수 있다. CEK당 하나이다. SSA 실시예에서, CEK들의 키 값들은 플래시 장치(예를 들어, 플래시 카드)에 의해 생성 또는 수신되고, 내부적으로만 사용되고, 외측 세계로부터 비밀들로서 유지된다. 인크립트되거나 암호화된 데이터는 해시되거나 암호는 데이터 보전을 보장하기 위하여 차단된 체인이다.

<87> 파티션의 모든 데이터는 다른 키들에 의해 인크립트되고 다른 키 ID들과 연관된다. 공용 또는 사용자 파일들 또는 동작 시스템 영역(즉, FAT)내 특정 논리 어드레스들은 임의의 키 또는 키 레퍼런스과 연관되지 않을 수 있고, 따라서 파티션에 액세스할 수 있는 임의의 엔티티에 이용할 수 있다.

<88> 키들 및 파티션들을 생성할뿐 아니라 그들로부터 데이터를 기입 및 독출하거나 키들을 사용하는 능력을 요청하는 엔티티는 액세스 제어 레코드(ACR)를 통하여 SSA 시스템에 로그인할 필요가 있다. SSA 시스템의 ACR 특권은 작용들을 호출한다. 모든 ACR은 다음 3개의 카테고리들의 작용들을 수행하기 위한 하가들을 가질 수 있다: 파티션들 및 키들/키 ID들 생성, 파티션들 및 키들 액세스 및 다른 ACR들 생성/업데이팅.

<89> ACR들은 ACR 그룹들 또는 AGP들이라 불리는 그룹들로 구성된다. 일단 ACR이 성공적으로 인증되면, SSA 시스템은 임의의 ACR의 작용들이 실행될 수 있는 세션을 개방한다. ACR들 및 AGP들은 정책들에 따른 파티션들 및 키들에 대한 액세스를 제어하기 위하여 사용된 보안 데이터 구조들이다.

<90> 사용자 파티션(들)

<91> SSA 시스템은 사용자 파티션(들) 이라 불리는 하나 또는 그 이상의 공용 파티션들을 관리한다. 이런 파티션은 저장 장치상에 존재하고 저장 장치의 표준 독출 기입 명령들을 통하여 액세스될 수 있는 파티션 또는 파티션들이다. 파티션(들)의 크기 및 장치상 존재에 관한 정보를 모으는 것은 바람직하게 호스트 시스템으로부터 숨겨질 수 없다.

<92> SSA 시스템은 표준 독출 기입 명령들 또는 SSA 명령들을 통하여 이 파티션(들)에 액세스할 수 있다. 그러므로, 파티션에 액세스하는 것은 바람직하게 특정 ACR들로 제한될 수 없다. 그러나, SSA 시스템은 호스트 장치들이 사용자 파티션에 대한 액세스를 제한하게 할 수 있다. 독출 및 기입 액세스들은 개별적으로 인에이블/디스에이블될 수 있다. 모두 4개의 결합들(예를 들어, 가입 전용, 독출 전용(기입 보호), 독출 및 기입 및 액세스 방지)은 허용된다.

<93> SSA 시스템은 ACR들이 사용자 파티션내의 파일들을 키 ID들에 연관되게 하고 상기 키 ID들과 연관된 키들을 사용하여 개별 파일들을 인크립트하게 한다. 사용자 파티션들내에서 인크립트된 파일들 액세스 및 파티션들에 대한 액세스 권리 설정은 SSA 명령 세트를 사용하여 행해질 것이다. 상기 특징들은 파일들에 구성되지 않은 데이터에 적용한다.

<94> SSA 파티션들

<95> SSA 명령들을 통해서만 액세스될 수 있는 숨겨진(인증되지 않은 파티션들) 파티션들이 있다. SSA 시스템은 ACR상 로깅에 의해 설정된 세션(하기에 기술된)을 통하는 것과 달리, 호스트 장치가 SSA 파티션에 액세스하게 할 것이다. 유사하게, 바람직하게 SSA는, 만약 요구가 설정된 세션을 통하여 발생하지 않으면, SSA 파티션의 존재, 크기 및 액세스 허용에 관한 정보를 제공하지 않을 것이다.

<96> 파티션들에 대한 액세스 권리들은 ACR 허용들로부터 유도된다. 일단 ACR이 SSA 시스템에 로그되면, 다른 ACR들(이하에서 기술된)과 파티션을 공유할 수 있다. 파티션이 생성될 때, 호스트는 레퍼런스 이름 또는 파티션에 대한 ID(예를 들어, 도 3 및 4의 P0-P3)를 제공한다. 이런 레퍼런스는 파티션에 대한 추가 독출 및 기입 명령들에 사용된다.

<97> 저장 장치의 구획화

<98> 장치의 모든 이용 가능한 저장 능력은 사용자 파티션 및 현재 구성된 SSA 파티션들에 바람직하게 할당된다. 그러므로, 임의의 재구획 동작은 기존 파티션들의 재구성을 포함할 수 있다. 장치 용량에 대한 순수 변화(모든 파티션들의 크기들의 합)는 영일 것이다. 장치 메모리 공간에서 파티션들의 ID들은 호스트 시스템에 의해 정의된다.

<99> 호스트 시스템은 두 개의 보다 작은 것들로 기존 파티션들을 재구획하거나, 두 개의 기존 파티션들(인접하거나 인접하지 않을 수 있음)을 하나로 합병한다. 분할되거나 합병된 파티션들의 데이터는 호스트의 분별부에서 소거되거나 건드려지지 않는다.

<100> 저장 장치의 재구획화가 데이터의 손실(저장 장치의 논리 어드레스 공간에서 소거 또는 이동되기 때문에)을 유발할 수 있기 때문에 재구획화에 대한 심한 제한들은 SSA 시스템에 의해 관리된다. 루트 AGP(하기에서 설명됨)에 잔류하는 ACR만이 재구획화 명령을 발생하고 그것에 의해 소유된 파티션들만을 참조할 수 있다. SSA 시스템이 파티션들(FAT 또는 다른 파일 시스템 구조)에 데이터를 조직하는 방법을 인식하지 못하기 때문에, 호스트는 장치가 재구획화되는 임의의 시간에 이들 구조들을 재구성할 책임이 있다.

<101> 사용자 파티션의 재구획화는 호스트 OS에 의해 알 수 있는 바와 같이 이런 파티션의 크기 및 다른 속성들을 변화시킬 것이다.

<102> 재구획화 후, SSA 시스템의 임의의 ACR이 비존재 파티션들을 참조하지 못하는 것을 보장하기 위한 호스트 시스템의 책임이 있다. 만약 이들 ACR들이 적당하게 삭제되거나 업데이트하지 못하면, 이들 ACR들 대신 비존재 파티션들에 액세스하기 위한 미래 시도들은 검출되고 시스템에 의해 거절된다. 삭제된 키들 및 키 ID들에 관한 유사한 관심이 취해진다.

<103> 키들, 키 ID들 및 논리적 보호

<104> 파일이 특정 숨겨진 파티션에 기입될 때, 상기 파일은 일반적인 공용부로부터 숨겨진다. 그러나, 일단 엔티티(반대이거나 아닌)가 이 파티션에 대한 지식을 얻고 액세스하면, 파일은 이용 가능하고 명백히 보인다. 파일을 추가로 안전하게 하기 위하여, SSA는 숨겨진 파티션에서 이를 인크립트할 수 있고, 여기서 파일을 디크립트하기 위한 키에 액세스하기 위한 증명서들은 바람직하게 파티션에 액세스하기 위한 파티션과 다르다. 파일들이 호스트에 의해 전체적으로 제어 및 관리된다는 사실로 인해, 파일과 CEK를 연관시키는 것은 문제이다. 어떤 SSA 지식들 - 키 ID -에 대해 파일을 링크하는 것은 이를 수정한다. 따라서, 키가 SSA에 의해 생성될 때, 호스트는 SSA에 의해 생성된 키를 사용하여 인크립트된 데이터와 이 키에 대한 키 ID를 연관시킨다. 만약 키가 키 ID와 함께 SSA에 전송되면, 키 및 키 ID는 서로 쉽게 연관될 수 있다.

<105> 키 값 및 키 ID는 논리적 보안을 제공한다. 위치와 무관하게 주어진 키 ID와 연관된 모든 데이터는 콘텐트 인크립션 키(CEK)에서 동일한 키 값으로 암호화되고, 상기 콘텐트 인크립션 키의 레퍼런스 이름 또는 키 ID는 호스트 애플리케이션에 의해 생성시 유일하게 제공된다. 만약 엔티티가 숨겨진 파티션에 액세스하면(ACR을 통하여 인증함으로써) 그리고 이 파티션내의 인크립트된 파일을 독출하거나 기입하기를 원하면, 파일과 연관된 키 ID에 액세스할 필요가 있다. 이 키 ID에 대한 키에 액세스를 승인할 때, SSA는 이런 키 ID와 연관된 CEK의 키 값을 로드하고 그것을 호스트에 전송하기 전에 데이터를 디크립트하거나 플래시 메모리(20)에 기입하기 전에 데이터를 인크립트한다. 일 실시예에서, 키 ID와 연관된 CEK의 키 값은 SSA 시스템에 의해 일단 임의로 생성되고 그것에 의해 유지된다. SSA 시스템 외측의 것은 CEK의 이런 키 값에 대한 지식 또는 액세스를 가지지 않는다. 외측 세계는 CEK의 키 값이 아닌 레퍼런스 또는 키 ID만을 제공하고 사용한다. 키 값은 완전히 관리되고 바람직하게 SSA에 의해 액세스할 수 있다. 선택적으로, 키는 SSA 시스템에 제공될 수 있다.

<106> SSA 시스템은 다음 암호화 모드들(사용된 실제 암호화 알고리즘들, 및 CEK의 키 값들은 시스템 제어되고 외부 세계에 드러나지 않는다) 다음의 임의의 하나(사용자 정의됨)를 사용하여 키 ID와 연관된 데이터를 보호한다.

<107> 블록 모드 - 데이터는 블록들로 분할되고, 그들 중 하나는 개별적으로 인크립트된다. 이 모드는 일반적으로 사전 공격들에 덜 민감하고 민감한 것으로 고려된다, 그러나 사용자가 데이터 블록들 중 임의의 하나에 랜덤하게 액세스하게 한다.

<108> 체인 모드 - 데이터는 인크립션 처리 동안 체인화된 블록들로 분할된다. 모든 블록은 다음 하나의 인크립션 처리에 대한 입력들 중 하나로서 사용된다. 이 모드에서, 비록 보다 많은 보안이 고려되지만, 데이터는 시작부터 끝으로 순차적으로 기입 및 독출되고, 사용자들에게 허용되지 않는 과부하를 생성한다.

<109> 해시 - 데이터 보전을 유효화하기 위해 사용될 수 있는 데이터 다이제스트의 부가적인 생성을 가진 체인 모드.

<110> ACR들 및 액세스 제어

<111> SSA는 다중 애플리케이션들을 처리하기 위하여 설계되고 이들 각각은 시스템 데이터베이스의 노드들의 트리로서 표현된다. 애플리케이션들 사이의 상호 배제는 트리 브랜치들 사이의 비혼선을 보장함으로써 달성된다.

<112> SSA 시스템에 액세스를 얻기 위하여, 엔티티는 시스템의 ACR들 중 하나를 통한 접속을 설정할 필요가 있다. 로그인 과정들은 사용자 접속하기를 선택한 ACR에 내장된 정의들에 따라 SSA 시스템에 의해 관리된다.

<113> ACR은 SSA 시스템에 대한 개별 로그인 포인트이다. ACR은 로그인 증명들 및 인증 방법을 홀드한다. 또한 레코드 안에는 SSA 시스템 내의 로그인 허용들이 존재하고, 이들 중 독출 및 기입 특권들이 있다. 이것은 동일한 AGP에서 n ACR들을 도시하는 도 5에 도시된다. 이것은 n ACR들 중 적어도 몇몇이 동일한 키에 대한 액세스를 공유할 수 있는 것을 의미한다. 따라서, ACR#1 및 ACR#n은 키 ID "키 3"을 가진 키에 대한 액세스를 공유하고, 여기서 ACR#1 및 ACR#n은 ACR ID들이고, "키 3"은 "키 3"과 연관된 데이터를 인크립트하기 위하여 사용된 키에 대한 키 ID이다. 동일한 키는 다중 파일들, 또는 데이터의 다중 세트들을 인크립트 및/또는 디크립트하기 위하여 사용될 수 있다.

<114> SSA 시스템은 시스템에 몇몇 종류의 로그인을 지원하고, 인증 알고리즘들 및 사용자 증명서들은 일단 성공적으로 로그인 하면 시스템에 사용자의 특권일 수 있는 바와 같이 가변할 수 있다. 도 5는 다른 로그인 알고리즘들

및 증명서들을 도시한다. ACR#1은 증명서로서 패스워드 로그인 알고리즘 및 패스워드를 지정하는 반면 ACR#2는 증명서로서 PKI(공용 키 인프라구조) 로그인 알고리즘 및 공용 키를 지정한다. 따라서, 로그인하기 위하여, 엔티티는 유효 ACR ID를 제공할 뿐 아니라, 올바른 로그인 알고리즘 및 증명서를 제공할 필요가 있을 것이다.

<115> 일단 엔티티가 SSA 시스템의 ACR에 로그되면, 그의 허용 - SSA 명령들을 사용할 권리들 -은 ACR과 연관된 허용 제어 레코드(PCR)에서 정의된다. 도 5에서, ACR#1은 "키 3"과 연관된 데이터에 대한 독출 전용 허용을 승인하고, ACR#2는 도시된 PCR에 따라 "키 5"와 연관된 데이터를 독출 및 기입하기 위한 허용을 승인한다.

<116> 다른 ACR들은 독출 및 기입하기 위한 것을 가진 키들 처럼 시스템의 공통 관심사들 및 특권을 공유할 수 있다. 이것을 달성하기 위하여, 공통의 무언가를 가진 ACR들은 AGP들으로 그룹화된다 - ACR 그룹들. 따라서, ACR#1 및 ACR#n은 키 ID "키 3"을 가진 키에 대한 액세스를 공유한다.

<117> AGP들 및 그 안쪽에 있는 ACR들은 민감한 데이터 보안을 유지하는 보안 키들을 생성하는 것을 제외하고 계층 트리들로 구성되고 ; ACR은 바람직하게 그의 키 ID/파티션들에 대응하는 다른 ACR 엔트리들을 생성한다. 이들 ACR 칠드런(children)은 그들의 파더(farther)-생성기와 동일하거나 작은 허용들을 가질 것이고, 생성된 파더 ACR 그 자체 키들에 대하여 허용들이 제공된다. 부가할 필요없이, 칠드런 ACR들은 그들이 생성하는 임의의 키에 액세스 허용들을 얻는다. 이것은 도 6에 도시된다. 따라서, AGP(120)에서 ACR들 모두는 ACR(122)에 의해 생성되었고 상기 ACR들 중 둘은 "키 3"과 연관된 데이터에 액세스하기 위한 허용(들)을 ACR(122)로부터 받는다.

<118> AGP

<119> SSA 시스템상에 로깅은 AGP 및 AGP내의 ACR을 지정함으로써 수행된다.

<120> 모든 AGP는 SSA 데이터베이스의 엔트리에 대한 인덱스로서 사용되는 유일한 ID(레퍼런스 이름)를 가진다. AGP 이름은 ACP가 생성될 때 SSA 시스템에 제공된다. 만약 제공된 AGP 이름이 시스템에 이미 존재하면, SSA는 생성 동작을 거절할 것이다.

<121> AGP들은 다음 섹션들에 기술될 바와 같이 액세스 및 관리 허용들의 위임에 대한 제한들을 관리하기 위하여 사용된다. 도 6의 두 개의 트리들에 의해 사용된 기능들 중 하나는 두 개의 애플리케이션들, 또는 두 개의 다른 컴퓨터 사용자들 같은 전체적으로 분리된 엔티티들에 의한 액세스를 관리하는 것이다. 상기 목적을 위해, 두 개의 액세스 처리들이 비록 양쪽이 동시에 발생하지만, 서로 실질적으로 무관하게(즉, 실질적으로 혼선 없이) 되는 것은 중요할 수 있다. 이것은 각각의 트리에서 인증, 허용뿐 아니라 부가적인 ACR들 및 AGP들의 생성이 다른 트리의 것에 접속되지 않고 의존하지 않는 것을 의미한다. 따라서, SSA 시스템이 메모리(10)에 사용될 때, 이것은 메모리 시스템(10)이 동시에 다중 애플리케이션들을 사용하게 한다. 또한 두 개의 애플리케이션들이 두 개의 독립된 세트의 데이터(예를 들어, 한 세트의 사진들 및 한 세트의 노래들)를 서로 무관하게 액세스하게 한다. 이것은 도 6에 도시된다. 따라서, 도 6의 상부 부분의 트리에서 노드들(ACR들)을 통하여 애플리케이션 또는 사용자가 액세스하기 위한 "키 3", "키 X" 및 "키 Z"와 연관된 데이터는 사진들을 포함할 수 있다. 애플리케이션 또는 사용자가 도 6의 바닥 부분의 트리의 노드들(ACR들)을 통하여 액세스하기 위한 "키 5" 및 "키 Y"와 연관된 데이터는 노래들을 포함할 수 있다. AGP가 생성된 ACR은 AGP가 ACR 엔트리들의 빔일 때만 그것을 삭제하기 위한 허용을 가진다.

<122> 엔티티의 SSA 엔트리 포인트: 액세스 제어 레코드(ACR)

<123> SSA 시스템의 ACR은 엔트리가 시스템에 로그되도록 허용되는 방식을 기술한다. 엔티티가 SSA 시스템에 로그될 때, 수행하고자 하는 인증 처리에 대응하는 ACR을 지정할 필요가 있다. ACR은 도 5에 도시된 바와 같이 ACR에서 정의된 바와 같이 인증되면 사용자가 실행할 수 있는 승인 작용들을 도시하는 허용 제어 레코드(PCR)를 포함한다. 호스트 측 엔티티는 모든 ACR 데이터 필드들을 제공한다.

<124> 엔티티가 ACR상에 성공적으로 로그될 때, 엔티티는 ACR의 구획 및 키 액세스 허용들 및 ACAM 허용들(하기 설명됨) 모두에 대한 질문을 수행할 것이다.

<125> ACR ID

<126> SSA 시스템 엔티티가 로그인 처리를 시작할 때, 로그인 방법에 대응하는 ACR ID(ACR이 생성될 때 호스트에 의해 제공된 바와 같이)를 지정할 필요가 있기 때문에 SSA는 올바른 알고리즘들을 설정할 것이고 모든 로그인 요구들이 부합될 때 올바른 PCR을 선택한다. ACR ID는 ACR이 생성될 때 SSA 시스템에 제공된다.

<127> 로그인/인증 알고리즘

<128> 인증 알고리즘은 어떤 종류의 로그인 과정이 엔티니에 의해 사용되고, 어떤 종류의 증명서들이 사용자의 신원 증명을 제공하기 위하여 부합되는지를 지정한다. SSA 시스템은 대칭 또는 비대칭 암호화를 바탕으로 두 가지 방식 인증 프로토콜들에 대한 비과정(및 증명서 없음) 및 패스워드 바탕 과정으로부터 범위를 설정한 몇몇 표준 로그인 알고리즘들을 지원한다.

<129> 증명서들

<130> 엔티티의 증명서들은 로그인 알고리즘에 대응하고 사용자를 검증 및 인증하기 위하여 SSA에 의해 사용된다. 증명서에 대한 예는 패스워드 인증을 위한 패스워드/PIN 번호, AES 인증을 위한 AES 키, 등등일 수 있다. 증명서들(즉, PIN, 대칭 키, 등등...)의 타입/포맷은 인증 모드로부터 미리 정의되고 유도된다; 그들은 ACR이 생성될 때 SSA 시스템에 제공된다. SSA 시스템은 장치(예를 들어, 플래시 카드)가 RSA 또는 다른 타입의 키 쌍을 생성하기 위하여 사용되고 공용 키가 증명서 생성을 위하여 익스포트될 수 있는 PKI 바탕 인증을 제외하고, 이들 증명서들을 정의, 분배 및 관리 부분을 가지지 않는다.

<131> 허용 제어 레코드(PCR)

<132> PCR은 SSA 시스템에 로깅 및 연속적으로 ACR의 인증 처리를 패스한 후 엔티티에 승인된 것을 도시한다. 3개의 타입의 허용 카테고리들이 있다: 구획 및 키들에 대한 생성 허용들, 파티션들에 대한 액세스 허용들 및 엔티티 ACR 속성들에 대한 관리 허용들.

<133> 액세스 파티션들

<134> PCR의 이런 섹션은 파티션들(SSA 시스템에 제공된 바와 같은 ID들을 사용하여)의 리스트를 포함하고, 엔티티는 연속적으로 ACR 단계를 완료 후 액세스할 수 있다. 각각의 파티션에 대해 액세스 타입은 기입 전용 또는 독출 전용으로 제한될 수 있거나 전체 기입/독출 액세스 권리들을 지정할 수 있다. 따라서, 도 5의 ACR#1은 파티션#1이 아닌 파티션#2에 액세스된다. PCR에 지정된 제한들은 SSA 파티션들 및 공용 파티션에 적용한다.

<135> 공용 파티션은 SSA 시스템을 호스팅하는 장치(예를 들어, 플래시 카드)에 대한 정규 독출 및 기입 명령들, 또는 SSA 명령들에 의해 액세스될 수 있다. 루트 ACR(이하에서 설명됨)이 공용 파티션을 제한하기 위한 허용이 생성될 때, 그는 그의 칠드런에게 이를 패스할 수 있다. ACR은 바람직하게 정규 독출 및 기입 명령들이 공용 파티션에 액세스하는 것을 제한할 수 있다. SSA 시스템의 ACR들은 생성 후에만 바람직하게 제한될 수 있다. 일단 ACR이 공용 파티션으로부터/상기 파티션에 독출/기입하기 위한 허용을 가지면, 바람직하게 제거될 수 없다.

<136> 액세스 키 ID들

<137> PCR의 이런 섹션은 키 ID들(호스트에 의하여 SSA 시스템에 제공된 바와 같이)의 리스트와 연관된 데이터를 포함하고, 엔티티는 ACR 정책들이 엔티티의 로그인 처리에 의해 부합될 때 액세스할 수 있다. 지정된 키 ID는 PCR에 나타나는 파티션에 존재하는 파일/파일들과 연관된다. 키 ID들이 장치(예를 들어, 플래시 카드)의 논리 어드레스들과 연관되지 않기 때문에, 하나 이상의 파티션이 특정 ACR과 연관될 때, 파일들은 파티션들 중 어느 하나일 수 있다. PCR에 지정된 키 ID들은 각각 다른 세트의 액세스 권리들을 가질 수 있다. 키 ID들에 의해 지적된 데이터 액세스는 기입 전용 또는 독출 전용으로 제한될 수 있거나 전체 기입/독출 액세스 권리들을 지정할 것이다.

<138> ACR 속성 관리(ACAM)

<139> 이 섹션은 특정 경우들에서 ACR의 시스템 속성들이 변화될 수 있는 방법을 기술한다.

<140> SSA 시스템에서 허용될 수 있는 ACAM 작용들은 하기와 같다:

<141> 1. AGP들 및 ACR 생성/삭제/업데이트.

<142> 2. 파티션들 및 키들 생성/삭제.

<143> 3. 키들 및 파티션들에 액세스 권리들 위임.

<144> 파더 ACR은 바람직하게 ACAM 허용들을 편집할 수 없다. 이것은 ACR의 삭제 및 재생성을 요구한다. 또한 ACR에 의해 생성된 키 ID에 대한 액세스 허용들은 바람직하게 삭제될 수 없다.

<145> ACR은 다른 ACR들 및 AGP들에 대한 능력을 가질 수 있다. ACR들을 생성하는 것은 생성기에 의해 소유된 몇몇 또는 모든 ACAM 허용들을 그들에게 위임하는 것을 의미할 수 있다. ACR들을 생성하기 위한 허용을 가지는 것은 다

음 작용들에 대한 허용을 가지는 것을 의미한다:

<146> 1. 차일드의 증명서들 정의 및 편집 - 인증 방법은 바람직하게 생성한 ACR에 의해 일단 설정되면 편집될 수 없다. 인증서들은 차일드에 대해 이미 정의된 인증 알고리즘의 경계내에서 변경될 수 있다.

<147> 2. ACR 삭제.

<148> 3. 차일드 ACR에 대한 생성한 허용 위임(따라서 그랜드칠드런을 가짐).

<149> 다른 ACR들을 생성하기 위한 허용들을 가진 ACR은 생성한 ACR들에 대한 차단해제된 허용을 위임하기 위한 허용을 가진다(비록 ACR들을 실패시키기 위한 허용을 가지지 않지만). 파더 ACR은 그의 차단 방지기에 대한 레퍼런스를 차일드 ACR에 배치할 것이다.

<150> 파더 ACR은 그의 차일드 ACR을 삭제하기 위한 허용을 가지는 ACR만이다. ACR이 생성된 하위 레벨 ACR을 삭제할 때, 하위 레벨 ACR에 의해 생성된 모든 ACR들은 자동으로 삭제된다. ACR이 삭제될 때, 생성된 모든 키 ID들 및 파티션들은 삭제된다.

<151> ACR이 자신의 레코드를 업데이트할 수 있는 두 개의 예외들이 있다.

<152> 1. 비록 생성기 ACR에 의해 설정되더라도 패스워드들/PIN들은 그것들을 포함하는 ACR에 의해서만 업데이트될 수 있다.

<153> 2. 루트 ACR은 그 자체 및 존재하는 AGP를 삭제할 수 있다.

<154> 키들 및 파티션들에 대한 액세스 권리 위임

<155> ACR들 및 AGP들은 루트 AGP 및 그 내부의 ACR들이 트리의 상부에 있는 경우(예를 들어, 도 6의 루트 AGP들 130 및 132) 계층 트리들에서 어셈블리된다. 그들이 비록 전체적으로 서로 분리되지만 SSA 시스템에 몇몇의 AGP 트리들이 있을 수 있다. AGP 내의 ACR은 내부에 있는 동일한 AGP내의 모든 ACR들, 및 그들에 의해 생성된 모든 ACR들에 대한 키들에 액세스 허용을 위임할 수 있다. 키들을 생성하기 위한 허용은 바람직하게 키들을 사용하기 위한 액세스 허용들을 위임하기 위한 허용을 포함한다.

<156> 키들에 대한 허용은 3개의 카테고리들로 분할된다:

<157> 1. 액세스 - 이것은 키에 대한 액세스 허용들, 즉 독출, 기입을 정의한다.

<158> 2. 소유권 - 키를 생성한 ACR은 정의에 의한 소유자이다. 이런 소유권은 하나의 ACR로부터 다른 ACR로 위임될 수 있다(만약 그들이 동일한 AGP 또는 차일드 AGP에 있다면). 키의 소유권은 허용들을 위임할뿐 아니라 삭제하기 위한 허용을 제공한다.

<159> 3. 액세스 권리 위임 - 이런 허용은 ACR이 그가 소유한 권리들을 위임하게 한다.

<160> ACR은 그가 생성한 액세스 허용들뿐 아니라 그가 액세스 허용들을 가진 다른 파티션들을 위임할 수 있다.

<161> 허용 위임은 파티션들의 이름들 및 키 ID들을 지정된 ACR의 PCR에 부가함으로써 수행된다. 키 액세스 허용들을 위임하는 것은 키 ID에 의한 것이거나 액세스 허용이 위임한 ACR의 생성된 키들 모두에 대한 것을 언급함으로써 이루어질 수 있다.

<162> ACR들의 차단 및 차단 해제

<163> ACR은 시스템을 가진 엔티티의 ACR 위임 처리가 성공하지 못할 때 증가하는 차단 카운터를 가질 수 있다. 성공하지 못한 인증의 특정 최대 수(MAX)에 도달될 때, ACR은 SSA 시스템에 의해 차단될 것이다.

<164> 차단된 ACR은 차단된 ACR에 의해 참조된 다른 ACR에 의해 차단 해제될 수 있다. 차단 해제 ACR에 대한 레퍼런스는 생성기에 의해 설정된다. 차단 해제 ACR은 바람직하게 차단된 ACR의 생성과 동일한 AGP에 있고 "차단 해제" 허용을 가진다.

<165> 시스템의 다른 ACR은 차단된 ACR을 차단 해제할 수 있다. ACR은 차단해제 ACR 없이 차단 카운터로 구성될 수 있다. 이 경우, 만약 이런 ACR이 차단되면, 차단 해제될 수 있다.

<166> 루트 AGP - 애플리케이션 생성

<167> SSA 시스템은 다중 애플리케이션들을 처리하기 위하여 설계되고 그들 각각의 데이터를 분리한다. AGP 시스템의 트리 구조는 애플리케이션 특정 데이터를 식별하고 분리하기 위하여 사용된 메인 툴이다. 루트 AGP는 애플리케

이션 SSA 데이터베이스 트리의 팁이고 다소 다른 작용 룰들을 고수한다. 몇몇의 루트 AGP들은 SSA 시스템에 구성될 수 있다. 두 개의 루트 AGP들(130 및 132)은 도 6에 도시된다. 분명히 보다 적거나 많은 AGP들은 사용될 수 있고 본 발명의 범위내에 있다.

<168> 새로운 애플리케이션에 대한 장치 및/또는 장치에 대한 새로운 애플리케이션들의 발행 증명서들 등록은 장치에 새로운 AGP/ACR 트리를 부가하는 처리를 통하여 수행된다.

<169> SSA 시스템은 3 개의 다른 모드들의 루트 AGP 생성을 지원한다(루트 AGP의 ACR들 및 그의 허용 모두):

<170> 1. 개방: 임의의 종류의 인증을 요구하지 않는 임의의 사용자 또는 엔티티, 또는 시스템 ACR(하기에 설명됨)을 통하여 인증된 사용자들/엔티티들은 새로운 루트 AGP를 생성할 수 있다. 개방 모드는 모든 데이터 전달이 개방 채널(즉, 발행 기관의 보안 환경에서)에서 수행되는 동안 임의의 보안 조치들 없이, 또는 시스템 ACR 인증을 통하여 설정된 보안 채널(즉, 오버 더 에어(OTA) 및 포스트 발행 과정들)을 통하여 루트 AGP들의 생성을 수행할 수 있다.

<171> 만약 시스템 ACR이 구성되지 않고(이것은 선택적 특징이다) 루트 AGP 생성 모드가 개방으로 설정되면, 개방 채널 옵션만이 이용 가능하다.

<172> 2. 제어: 시스템 ACR을 통하여 인증된 엔티티들만이 새로운 루트 AGP를 생성할 수 있다. SSA 시스템은 만약 시스템 ACR이 구성되지 않으면 이 모드로 설정될 수 없다.

<173> 3. 로깅: 루트 AGP들의 생성은 디스에이블되고 부가적인 루트 AGP들이 시스템에 부가되지 않을 수 있다.

<174> 두 개의 SSA 명령들은 이런 특징(이들 명령들은 인증 없이 임의의 사용자/엔티티에 이용할 수 있다)을 제어한다:

<175> 1. 방법 구성 명령 - 3 개의 루트 AGP 생성 모드들 중 임의의 하나를 사용하기 위하여 SSA 시스템을 구성하기 위해 사용됨. 다음 모드 변화들만이 허용된다: 개방 -> 제어, 제어 -> 록킹(즉, 만약 SSA 시스템이 현재 제어된 것으로 구성되면, 록킹으로만 변화될 수 있다).

<176> 2. 방법 구성 록 명령 - 방법 구성 명령을 디스에이블하고 현재 선택된 방법을 영구히 록킹하기 위하여 사용됨.

<177> 루트 AGP가 생성될 때, ACR들의 생성 및 구성할 수 있는(루트 AGP의 생성에 적용된 동일한 액세스 제한들을 사용함) 특정 시작 모드에 있다. 루트 AGP 구성 처리의 종료 시, 엔티티가 개방 모드로 명백히 스위칭될 때, 기존 ACR들은 더 이상 업데이트될 수 없고 부가적인 ACR들은 더 이상 생성될 수 없다.

<178> 일단 루트 AGP가 표준 모드에 놓이면, 그것은 루트 AGP를 삭제하기 위한 허용이 양도된 ACR들 중 하나를 통하여 시스템에 로깅함으로써 삭제될 수 있다. 이것은 특정 시작 모드 외에 루트 AGP의 다른 예외이다; 바람직하게 그것은 다음 트리 레벨의 AGP들과 반대인 바와 같이 자신의 AGP를 삭제하기 위한 허용을 가진 ACR을 포함할 수 있는 AGP이다.

<179> 루트 ACR 및 표준 ACR 사이의 제 3 및 최종 차는 파티션들을 생성 및 삭제하기 위한 허용을 가질 수 있는 시스템의 ACR이다.

<180> SSA 시스템 ACR

<181> 시스템 ACR은 다음 두 개의 SSA 동작들에 사용될 수 있다:

<182> 1. 적의의 환경 내에서 보안 채널의 보호하에서 ACR/AGP 트리 생성.

<183> 2. SSA 시스템을 호스팅하는 장치를 식별 및 인증.

<184> 바람직하게 SSA에 단지 하나의 시스템 ACR이 있고 일단 변화되지 않는 것이 정의된다. 시스템 ACR을 생성할 때 시스템 인증이 필요하지 않다; 단지 SSA 명령은 요구된다. 생성-시스템-ACR 특징은 디스에이블될 수 있다(유사하게 생성-루트-AGP 특징). 시스템 ACR이 생성된 후, 생성-시스템-ACR 명령은, 바람직하게 단지 하나의 시스템 ACR이 허용되기 때문에 효과를 가지지 않는다.

<185> 생성 과정 동안, 시스템 ACR은 동작하지 않는다. 마무리 후, 특정 명령은 시스템 ACR이 생성되고 진행될 준비를 가리키는 것을 발행할 필요가 있다. 이런 시점 후 시스템 ACR은 바람직하게 업데이트 또는 대체될 수 없다.

<186> 시스템 ACR은 SSA에서 루트 ACR/AGP를 생성한다. 그것은 호스트가 만족되고 차단하는 시간까지 루트 레벨을 부가/변화시키기 위한 허용을 가진다. 루트 AGP를 차단하는 것은 시스템 ACR에 대한 접속을 자르고 증거를 조화시

킨다. 이 시점에서, 루트 AGP 및 그 내부의 ACR들을 변화/편집할 수 없다. 이것은 SSA 명령을 통하여 수행된다. 루트 AGP들의 생성을 디스에이블하는 것은 영구적인 효과를 가지며 리버스될 수 없다. 시스템 ACR을 포함하는 상기 특징들은 도 7에 도시된다. 시스템 ACR은 3개의 다른 루트 AGP들을 생성하기 위하여 사용된다. 이들이 생성된 후 특정 시간에서, SSA 명령은 시스템 ACR로부터 루트 AGP들을 차단하기 위하여 호스트로부터 전송되어, 도 7의 루트 AGP들에 시스템 ACR을 접속시키는 점선들에 의해 표시된 바와 같이 생성-루트-AGP 특징을 디스에이블한다. 이것은 3 개의 루트 AGP들의 조화 증거를 렌더한다. 3 개의 루트 AGP들은 루트 AGP들이 차단되기 전 또는 후에 3 개의 독립된 트리들을 형성하기 위하여 칠드런 AGP들을 생성하기 위하여 사용될 수 있다.

<187> 상술된 특징들은 콘텐트를 가진 보안 생산물들을 구성할 때 콘텐트 소유자에게 큰 융통성을 제공한다. 보안 생산물들은 "발행"될 필요가 있다. 발행은 장치가 호스트를 식별할 수 있는 식별 키들을 넣는 처리이고 그 반대도 가능하다. 장치(예를 들어 플래시 카드)를 식별하는 것은 호스트가 그의 비밀들을 신뢰할 수 있는지를 가리키게 한다. 다른 한편, 호스트를 식별하는 것은 만약 호스트가 허용되면 장치가 보안 정책들(특정 호스트 명령 승인 및 실행)을 강화하게 한다.

<188> 다중 애플리케이션들을 사용하기 위하여 설계된 생산물들은 몇몇 식별 키들을 가질 것이다. 생산물은 "사전 발행"될 수 있고 - 선적 전 제조 동안 저장된 키들 -, 또는 "포스트 발행"될 수 있다 - 새로운 키들을 선적 후 부가된다. 포스트 발행을 위하여, 메모리 장치(예를 들어, 메모리 카드)는 장치에 애플리케이션들을 부가하도록 허용된 엔티티들을 식별하기 위하여 사용된 몇몇 종류의 마스터 또는 장치 레벨 키들을 포함할 필요가 있다.

<189> 상술된 특징들은 생산물이 포스트 발행을 인에이블/디스에이블하도록 구성되게 한다. 게다가, 포스트 발행 구성은 선적 후 보안적으로 수행될 수 있다. 장치는 상술된 마스터 또는 장치 레벨 키들 외에 그 위에 키들을 가지지 않은 소매 생산물로서 구매되고, 그 다음 추가 포스트 발행 애플리케이션들을 인에이블하거나 디스에이블하기 위하여 새로운 사용자에 의해 구성될 수 있다.

<190> 따라서, 시스템 ACR 특징은 상기 목적들을 달성하기 위한 능력을 제공한다:

<191> - 시스템 ACR을 가지지 않는 메모리 장치들은 애플리케이션들의 제한되지 않고 제어되지 않은 변경을 허용할 것이다.

<192> - 시스템 ACR 없는 메모리 장치들은 시스템 ACR 생성을 디스에이블하기 위하여 구성될 수 있고, 이것은 새로운 애플리케이션들의 부가를 제어하는 방식이 없다는 것을 의미한다(만약 새로운 루트 AGP를 생성하는 특징이 디스에이블되면).

<193> - 시스템 ACR을 가진 메모리 장치들은 보안 채널을 통한 애플리케이션들의 제어된 부가가 시스템 ACR 증명서를 사용한 인증 과정을 통하여 설정되게 할 것이다.

<194> - 시스템 ACR을 가진 메모리 장치들은 애플리케이션들이 부가되기 전 또는 후에, 애플리케이션 부가 특징을 디스에이블하기 위하여 구성될 수 있다.

<195> 키 ID 리스트

<196> 키 ID들은 특정 ACR 요구에 따라 생성된다; 그러나, 메모리 시스템(10)에서, 그들은 SSA 시스템에 의해 유일하게 사용된다. 키 ID가 생성될 때, 다음 데이터는 생성한 ACR에 의해 또는 상기 ACR에 제공된다:

<197> 1. 키 ID. ID는 호스트를 통하여 엔티티에 의해 제공되고 모든 추가 독출 또는 기입 액세스들에서 키를 사용하여 인크립트 또는 디크립트된 키 및 데이터를 참조하기 위하여 사용된다.

<198> 2. 키 암호화 및 데이터 보전 모드(상기 및 이하에 설명되는 차단, 체인화 및 해시 모드들).

<199> 속성들이 제공된 호스트 외에, 다음 데이터는 SSA 시스템에 의해 유지된다:

<200> 1. 키 ID 소유자. 소유자인 ACR의 ID. 키 ID가 생성될 때, ACR은 소유자이다. 키 ID 소유권은 그러나 다른 ACR에 전달될 수 있다. 바람직하게 키 ID 소유자만이 키 ID의 소유권을 전달, 및 위임할 수 있게 된다. 연관된 키에 액세스 허용 위임, 및 이들 권리들 취소는 위임 허용들이 할당된 키 ID 소유자 또는 임의의 다른 ACR에 의해 관리될 수 있다. 이들 동작들 중 임의의 하나를 실행하기 위한 시도가 이루어질 때마다, SSA 시스템은 만약 요구한 ACR이 인증되면 승인할 것이다.

<201> 2. CEK. 이것은 키 값이 키 ID에 의해 연관되거나 지적된 콘텐트를 암호화하기 위하여 사용된 CEK이다. 키 값은 SSA 시스템에 의해 생성된 128 비트 AES 랜덤 키일 수 있다.

<202> 3. MAC 및 IV 값들. 체인 차단 암호화(CBC) 인크립션 알고리즘들에 사용된 동적 정보(메시지 인증 코드들 및 시작 벡터들).

<203> SSA의 다양한 특징들은 도 8a ~ 16의 흐름도를 참조하여 도시되고, 여기서 단계의 좌측 'H'는 동작이 호스트에 의해 수행되는 것을 의미하고, 'C'는 동작이 카드에 의해 수행되는 것을 의미한다. 이들 SSA 특징들이 메모리 카드들을 참조하여 도시되는 동안, 이들 특징들이 다른 물리적 형태들의 메모리 장치들에 적용할 수 있다는 것이 이해될 것이다. 시스템 ACR을 생성하기 위하여, 호스트는 메모리 장치(10)의 SSA에 시스템 ACR을 생성하기 위한 명령(블록 202)을 발행한다. 장치(10)는 시스템 ACR이 이미 존재하는지(블록 204, 다이아몬드 206)를 검사하여 응답한다. 만약 이미 존재하면, 장치(10)는 결함을 리턴하고 정지한다(직사각형 208). 만약 그렇지 않으면, 메모리(10)는 시스템 ACR 생성이 허용되는지(다이아몬드 210)를 알기 위하여 검사하고, 만약 허용되지 않으면(블록 212) 결함 상태를 리턴한다. 따라서, 시스템 ACR이 필요하지 않도록 필요한 보안 특징들이 미리 결정되는 경우 같은 시스템 ACR의 생성을 장치 발행자가 허용하지 않는 예들이 있을 수 있다. 만약 이것이 허용되면, 장치(10)는 OK 상태를 리턴하고 호스트로부터 시스템 ACR 증명서들을 기다린다(블록 214). 호스트는 SSA 상태 및 시스템 ACR의 생성이 허용되는 것을 장치(10)가 가리키는지(블록 216 및 다이아몬드 218)를 검사한다. 만약 생성이 허용되지 않거나 만약 시스템 ACR이 이미 존재하면, 호스트는 정지한다(직사각형 220). 시스템 ACR의 생성이 허용되는 것을 만약 장치(10)가 가리키면, 호스트는 로그인 증명서를 정의하기 위하여 SSA 명령을 발행하고 이를 장치(10)에 전송한다(블록 222). 장치(10)는 수신된 증명서로 시스템 ACR 레코드를 업데이트하고 OK 상태를 리턴한다(블록 224). 상태 신호에 응답하여, 호스트는 시스템 ACR이 준비된 것을 가리키는 SSA 명령을 발행한다(블록 226). 장치(10)는 시스템 ACR을 록킹하여 응답함으로써 업데이트 또는 대체될 수 없다(블록 228). 이것은 호스트에 대해 장치(10)를 식별하기 위하여 시스템 ACR 및 그의 신원의 특징들을 록킹한다.

<204> 새로운 트리들(새로운 루트 AGP들 및 ACR)을 생성하기 위한 과정은 이들 기능들이 장치에 구성되는 방식에 의해 결정된다. 도 9는 상기 과정들을 설명한다. 호스트(24) 및 메모리 시스템(10) 모두는 이를 따른다. 만약 새로운 루트 AGP를 부가하는 것이 함께 디스에이블되면, 새로운 루트 AGP들은 부가될 수 없다(다이아몬드 246). 만약 인에이블되지만 시스템 ACR이 요구되면, 호스트는 시스템 ACR을 통하여 인증하고 생성 루트_AGP 명령을 발행하기 전에(블록 254) 보안 채널을 설정한다(다이아몬드 250, 블록 252). 만약 시스템 ACR이 필요하지 않으면(다이아몬드 248), 호스트(24)는 인증 없이 생성 루트 AGP 명령을 발생하고 블록(254)으로 진행한다. 만약 시스템 ACR이 존재하면, 호스트는 필요하지 않을 때에서 이를 사용할 수 있다(흐름도에 도시되지 않음). 장치(예를 들어, 플래시 카드)는 만약 기능이 디스에이블되면 새로운 루트 AGP를 생성하기 위한 임의의 시도를 거절할 것이고, 만약 시스템 ACR이 필요하면(다이아몬드들 246 및 250) 인증 없이 새로운 루트 AGP를 생성하기 위한 시도를 거절할 것이다. 블록(254)에서 새롭게 생성된 AGP 및 ACR은 동작 모드로 스위칭되어 상기 AGP들의 ACR들은 업데이트될 수 없거나 변화될 수 없고, ACR들은 그들에 부가될 수 없다(블록 256). 그 다음 시스템은 선택적으로 록킹되어 부가적인 루트 AGP들은 생성될 수 없다(블록 258). 점선 박스(258)는 이런 단계가 선택적 단계인 것을 가리키는 협정이다. 이 출원의 도면의 점선 흐름도의 모든 박스들은 선택적 단계들이다. 이것은 콘텐트 소유자가 합법 콘텐트를 가진 진짜 메모리 장치를 모사할 수 있는 다른 불법 목적을 위한 장치(10)의 사용을 차단하게 한다.

<205> ACR들을 생성하기 위하여(상술된 바와 같이 루트 AGP의 ACR들과 다른), 도 10에 도시된 바와 같이 ACR을 생성하기 위한 권리를 가진 임의의 ACR에서 시작할 수 있다(블록 270). 엔티티는 엔트리 포인트 ACR 신원, 생성하기를 원하는 모든 필요한 속성들을 가진 ACR을 제공함으로써 호스트(24)를 통하여 진입하기를 시도한다(블록 272). SSA는 ACR 신원에 대한 매칭 및 상기 신원을 가진 ACR이 ACR을 생성하기 위한 허용을 가지는지를 검사한다(다이아몬드 274). 만약 상기 요구가 인증되는 것으로 검증되면, 장치(10)의 SSA는 ACR을 생성한다.

<206> 도 11은 도 10의 방법을 사용하는 보안 애플리케이션들에 유용한 트리를 도시하는 두 개의 AGP들을 도시한다. 따라서, 판매중 AGP의 신원(m1)을 가진 ACR은 ACR을 생성하기 위한 허용을 가진다. ACR(m1)은 키 ID "판매 정보"와 연관된 데이터 및 키 ID "가격 리스트"와 연관된 데이터를 독출 및 기입하기 위한 키를 사용하기 위한 허용을 가진다. 도 10의 방법을 사용하여, 두 개의 ACR들: 키 ID "판매중 정보"와 연관된 데이터에 액세스하기 위하여 필요한 키가 아닌, 키 ID "가격 리스트"와 연관된 가격 데이터에 액세스하기 위한 키에 대한 독출 허용만을 가진 s1 및 s2를 가진 판매 AGP를 생성한다. 이런 방식으로, ACR들(s1 및 s2)을 가진 엔티티들은 독출되지만 가격 데이터를 변화시킬 수 없고, 판매 데이터에 액세스하지 못할 것이다. 다른 한편 ACR(m2)은 ACR들을 생성하기 위한 허용을 가지지 않으며, 단지 키 ID "가격 리스트" 및 키 ID "판매 정보"와 연관된 데이터에 액세스하기 위한 키들에 대한 독출 허용만을 가진다.

<207> 따라서, 액세스 권리들은 m1이 가격 데이터를 독출할 권리를 s1 및 s2에 위임하는 경우 상술된 방식으로 위임될

수 있다. 이것은 특히 큰 마케팅 및 판매 그룹들이 포함되는 경우 유용하다. 한명 또는 몇몇 판매인들이 있는 경우, 도 10의 방법을 사용할 필요가 없을 수 있다. 대신, 액세스 권리들은 도 12에 도시된 바와 같이 동일한 AGP 내의 하위 또는 동일 레벨의 사람에게 ACR에 의해 위임될 수 있다. 첫째, 엔티티는 호스트를 통해 트리의 상술된 방식으로 ACR을 지정함으로써 상기 AGP에 대한 트리에 진입한다(블록 280). 다음 호스트는 위임하기 위한 ACR 및 권리들을 지정할 것이다. SSA는 ACR에 대한 트리(들) 및 특정 다른 ACR에 권리들을 위임하기 위한 허용을 가지는지(다이아몬드 282)를 검사한다. 만약 가지면, 권리들은 위임되고(블록 284); 만약 그렇지 않으면 중단된다. 결과는 도 13에 도시된다. 이 경우 ACR m1은 독출 허용을 ACR s1에 위임하기 위한 허용을 가지므로, s1은 위임 후 가격 데이터에 액세스하기 위한 키를 사용할 것이다. 이것은 만약 m1이 위임하기 위한 가격 데이터 및 허용에 액세스하기 위한 동일하거나 보다 큰 권리들을 가지면 수행될 수 있다. 일 실시예에서, m1은 위임 후 액세스 권리들을 유지한다. 바람직하게 액세스 권리들은 제한된 시간, 제한된 액세스들, 등등 같은 제한된 조건들(오히려 영구적으로) 하에서 위임될 수 있다.

<208> 키 및 키 ID를 생성하기 위한 처리는 도 14에 도시된다. 엔티티는 ACR을 통하여 인증한다(블록 302). 엔티티는 호스트에 의해 지정된 ID를 가진 키의 생성을 요구한다(블록 304). SSA는 만약 지정된 ACR이 그렇게 하기 위한 위임을 가지는지를 검사 및 살펴본다(다이아몬드 306). 예를 들어, 만약 키가 특정 파티션의 데이터에 액세스하기 위하여 사용되면, SSA는 ACR이 상기 파티션에 액세스할 수 있는지를 검사 및 살펴본다. 만약 ACR이 인증되면, 메모리 장치(10)는 호스트에 의해 제공된 키 ID와 연관된 키 값을 생성하고(블록 308), ACR에 키 ID를 저장하고, 키 값을 메모리(제어기 연관 메모리 또는 메모리 20)에 저장하고 엔티티에 의해 지정된 정보에 따라 권리들 및 허용을 할당하고(블록 310) 및 상기 할당된 권리들 및 허용들을 가진 ACR의 PCR을 변형한다(블록 312). 따라서, 키의 생성기는 독출 및 기입 허용들, 동일한 AGP의 다른 ACR들 또는 하위 레벨의 ACR에 위임 및 공유하기 위한 권리, 및 키의 소유권을 전달할 권리 같은 모든 이용 가능한 권리들을 가진다.

<209> ACR은 도 15에 도시된 바와 같이 SSA 시스템의 다른 ACR의 허용들(또는 함께 존재)을 변경할 수 있다. 엔티티는 이전과 같이 ACR을 통한 트리이다; 하나의 경우 엔티티는 인증되고 그 다음 ACR을 지정한다(블록 330, 332). 이것은 타켓 ACR의 삭제 또는 타켓 ACR의 허용을 요구한다(블록 334). 만약 지정된 ACR 또는 상기 시간에서의 하나의 작용이 그렇게 하기 위한 권리를 가지면(다이아몬드 336), 타켓 ACR은 삭제되거나, 타켓 ACR의 PCR은 상기 허용을 삭제하기 위하여 변경된다(블록 338). 만약 이것이 인증되지 않으면, 시스템은 정지한다.

<210> 상술된 처리 후, 타켓은 처리 이전에 수행할 수 있는 데이터에 액세스를 더 이상 수행할 수 없을 것이다. 도 16에 도시된 바와 같이, 엔티티는 타켓 ACR에 진입하기 위하여 시도할 수 있고(블록 350) 인증 처리 결함들을 발견하는데, 그 이유는 이전에 존재하는 ACR ID가 SSA에 더 이상 제공되지 않기 때문이고, 이에 따라 액세스 권리들은 거절된다(다이아몬드 352). ACR ID가 삭제되지 않은 것을 가정하여, 엔티티는 ACR(블록 354) 및 특정 파티션의 키 ID 및/또는 데이터(블록 356)를 지정하고, SSA는 상기 ACR의 PCR에 따라 허용된 키 ID 또는 구획 액세스 요구를 찾기 위하여 검사한다(다이아몬드 358). 만약 허용이 삭제되거나 만료되면, 요구는 다시 거절된다. 그렇지 않으면, 요구는 승인된다(블록 360).

<211> 상기 처리는 ACR 및 그의 PCR이 다른 ACR에 의해 막 변경되었는지 시작하기 위하여 구성되었는지에 무관하게, 보호될 데이터에 대한 액세스가 장치(예를 들어 플래시 카드)에 의해 관리되는 방법을 기술한다.

<212> 세션들

<213> SSA 시스템은 동시에 로그인된 다중 사용자들을 처리하기 위하여 설계된다. 이 특징이 사용될 때, SSA에 의해 수신된 모든 명령은 특정 엔티티와 연관되고 만약 이런 엔티티를 인증하기 위하여 사용된 ACR이 요구된 작용에 대한 허용들을 가지면 실행된다.

<214> 다중 엔티티들은 세션 개념을 통하여 지원된다. 세션은 인증 처리 동안 설정되고 SSA 시스템에 의해 세션-id가 할당된다. 세션-id는 시스템에 로그하기 위하여 사용된 ACR과 내부적으로 연관되고 모든 다른 SSA 명령들에 사용될 엔티티에 익스포트된다.

<215> SSA 시스템은 두 가지 타입들을 지원한다: 세션들 중 : 개방, 및 보안 세션들. 특정 인증 처리와 연관된 세션 타입은 ACR에서 정의된다. SSA 시스템은 인증 자체를 강화하는 방식과 유사한 방식으로 세션 설정을 강화할 것이다. ACR이 엔티티 허용들을 정의하기 때문에, 이런 메카니즘은 시스템 설계자들이 특정 키에 액세스하거나 특정 ACR 관리 동작들(즉, 새로운 ACR들 생성 및 증명서들 설정)을 호출하는 것과 보안 터널링을 연관시킬 수 있게 한다.

<216> 개방 세션

<217> 개방 세션은 버스 인크립션 없는 세션 id로 식별된 세션이고, 모든 명령들 및 데이터는 명확하게 패스된다. 이런 동작 모드는 엔티티들이 협박 모델의 일부가 아니고, 버스상 도청도 아닌 바람직하게 다중 사용자 또는 다중 엔티티 환경에 사용된다.

<218> 비록 데이터의 전송을 보호하거나 호스트 측상에서 애플리케이션들 사이의 효율적인 방화벽을 수행하지 않지만, 개방 세션 모드는 SSA 시스템이 현재 인증된 ACR들에 대해 허용된 정보에만 액세스되게 할 수 있다.

<219> 개방 세션은 파티션 또는 키가 보호될 필요가 있는 경우들에 사용될 수 있다. 그러나, 유효 인증 처리 후, 액세스는 호스트상 모든 엔티트들에 대해 승인된다. 인증된 ACR의 허용을 얻기 위하여 다양한 호스트 애플리케이션들이 공유할 필요가 있는 것만이 세션 id이다. 이것은 도 17a에 도시된다. 라인 위 단계들(400)은 호스트(24)에 의해 얻어진 것이다. 엔티티가 ACR1에 대해 인증된 후(블록 402), 메모리 장치(10)의 키 ID X와 연관된 파일에 액세스를 요구한다(블록 404,406 및 408). 만약 ACR 1의 PCR이 상기 액세스를 허용하면, 장치(10)는 요구를 승인한다(다이아몬드 410). 만약 그렇지 않으면, 시스템은 블록(402)으로 리턴한다. 인증이 완료된 후, 메모리 시스템(10)은 할당된 세션 id(및 ACR 증명서들이 아닌)에 의한 명령을 발행하는 엔티티를 식별한다. 일단 ACR(1)이 PCR의 키 ID들과 연관된 데이터에 액세스를 얻으면, 개방 세션에서, 임의의 다른 애플리케이션 또는 사용자는 호스트(24) 상 다른 애플리케이션들 사이에 공유된 올바른 세션 ID를 지정함으로써 동일한 데이터에 액세스할 수 있다. 이런 특징은 사용자가 일단 한번 로그인 할 수 있는데 보다 편리한 애플리케이션들에서 바람직하고, 로그인이 다른 애플리케이션들에 대해 수행되는 계정에 묶여진 모든 데이터에 액세스할 수 있다. 따라서, 셀룰러 전화 사용자는 저장된 이메일들에 액세스할 수 있고, 여러번 로그인하지 않고 메모리(20)의 저장된 음악을 청취한다. 다른 한편, ACR1에 의해 포함되지 않은 데이터는 액세스 가능하지 않을 것이다. 따라서, 동일한 셀룰러 전화 사용자는 독립된 계정 ACR2를 통하여 액세스할 수 있는 게임들 및 사진들 같은 값어치 있는 콘텐트를 가질 수 있다. 이것은 그가 액세스하기 위한 다른 것들을 원하지 않는 데이터이다: 그의 전화를 차용한 다른 사람들인 그가 비록 그의 제 1 계정 ACR1을 통하여 이용할 수 있는 데이터에 액세스할 수 있는 다른 것들을 신경쓰지 않을 수 있지만. 개방 세션에서 ACR1에 대한 액세스를 허용하면서 두 개의 독립된 계정들로 데이터에 대한 액세스를 분리하는 것은 용이하게 사용되고 값진 데이터 보호를 제공한다.

<220> 호스트 애플리케이션들 중에서 세션 id를 공유하는 처리를 보다 용이하게 하기 위하여, ACR이 개방 세션을 요구할 때, 세션이 "0(제로)" id를 할당하는 것을 특히 요구할 수 있다. 이런 방식에서, 애플리케이션들은 미리 정의된 세션-id를 사용하기 위하여 설계된다. 유일한 제한은 분명한 이유들로 인해 세션 0을 요구하는 단지 하나의 ACR만이 특정 라인에서 인증될 수 있다는 것이다. 다른 ACR 요구 세션 0을 인증하기 위한 시도는 거절될 것이다.

<221> 보안 세션

<222> 보안 층을 부가하기 위하여, 세션은 도 17b에 도시된 바와 같이 사용될 수 있다. 메모리(10)는 작용 세션들의 세션 id들을 저장한다. 도 17b에서, 예를 들어, 키 ID X와 연관된 파일에 액세스하기 위하여, 엔티티는 파일에 액세스하도록 허용되기 전에 세션 id "A" 같은 세션 id를 제공할 필요가 있을 것이다(블록 404,406,412 및 414). 이런 방식으로, 만약 요구한 엔티티가 올바른 세션 id를 인식하지 않으면, 메모리(10)에 액세스할 수 없다. 세션이 오버한 후 세션 id가 삭제되고 각각의 세션과 다를 것이기 때문에, 엔티티는 세션 번호를 제공할 때만 액세스를 얻을 수 있다.

<223> SSA 시스템은 명령이 세션 번호를 사용하여 올바른 인증된 엔티티로부터 쉽게 발생할지를 추적한다. 공격자들이 부당한 명령들을 전송하기 위하여 개방 채널을 사용할 것이 의심되는 애플리케이션들 및 용도의 경우에 대해, 호스트 애플리케이션은 보안 세션(보안 채널)을 사용한다.

<224> 보안 채널을 사용할 때, 세션 id 및 전체 명령은 보안 채널 인크립션(세션) 키로 인크립트되고 보안 레벨은 호스트측 실행과 같이 높다.

<225> 세션 종료

<226> 세션은 종료되고, ACR은 다음 시나리오들 중 임의의 하나에서 로그 오프된다:

<227> 1. 엔티티는 명백한 말단 세션 명령을 발행한다.

<228> 2. 통신 타임아웃. 특정 엔티티는 ACR 파라미터들 중 하나로서 정의된 시간 기간 동안 명령을 발행하지 않는다.

<229> 3. 모든 개방 세션들은 장치(예를 들어, 플래시 카드)가 리셋되고 및/또는 전력이 사이클된 후 종료된다.

<230> 데이터 보전 서비스들

<231> SSA 시스템은 SSA 데이터베이스(모든 ACR들, PCR들, 등등...)의 보전을 검증한다. 게다가 데이터 보전 서비스들은 키 ID 메카니즘을 통하여 전체 데이터에 대해 제공된다.

<232> 만약 키 ID가 인크립션 알고리즘들로서 해시되어 구성되면, 해시 값들은 CEK 및 CEK의 IV와 함께 저장된다. 해시 값들은 기입 동작 동안 계산 및 저장된다. 해시 값들은 다시 독출 동작들 동안 계산되고 이전 기입 동작들 동안 저장된 값들과 비교된다. 엔티티가 키 ID를 액세스하는 매 시간, 부가적인 데이터는 홀수 데이터에 연결되고(암호적으로) 적당한 해시 값(독출 또는 기입을 위해)은 업데이트된다.

<233> 호스트만이 키 ID와 연관되거나 지적된 데이터를 알기 때문에, 호스트는 다음 방식으로 데이터 보전 기능의 몇몇 측면들을 관리한다:

<234> 1. 키 ID와 연관되거나 지적될 데이터 파일은 시작부터 끝까지 기입 또는 독출된다. 파일 부분들에 액세스하기 위한 임의의 시도는 SSA 시스템이 CC 인트립션 방법을 사용하고 전체 데이터의 해시된 메시지 다이제스트를 생성하기 때문에 방지될 것이다.

<235> 2. 중간 해시 값들이 SSA 시스템에 의해 유지되기 때문에, 인접한 스트림의 데이터를 보호할 필요가 없다(데이터 스트림은 다른 키 Id들의 데이터 스트림과 인터리빙될 수 있고 다중 세션들상에서 분할될 수 있다). 그러나, 엔티티는 만약 데이터 스트림이 재시작되면 해시 값들을 리셋하도록 SSA 시스템에게 명확하게 명령할 필요가 있을 것이다.

<236> 3. 독출 동작이 완료될 때, 호스트는 기입 동작 동안 계산된 해시 값과 비교하여 SSA 시스템이 독출 해시를 유효화하는 것을 요구한다.

<237> 4. SSA 시스템은 "더미 독출" 동작들을 제공한다. 이런 특징은 인크립션 엔진들을 통하여 데이터를 스트림하지만 호스트 밖으로 전송하지 않을 것이다. 이런 특징은 장치(예를 들어, 플래시 카드)의 밖으로 실제로 독출되기 전에, 데이터 보전을 검증하기 위하여 사용될 수 있다.

<238> 랜덤 수 생성

<239> SSA 시스템은 내부 랜덤 수 생성기를 외부 엔티티들이 이용하게 할 것이고 SSA 시스템의 외부에 사용될 랜덤 수들을 요구한다. 이 서비스는 임의의 호스트에 이용할 수 있고 인증할 필요없다.

<240> RSA 키 쌍 생성

<241> SSA 시스템은 내부 RSA 키 쌍 생성 특징을 외부 사용자들에 이용하게 하고 SSA 시스템의 외부에 사용될 키 쌍을 요구한다. 이런 서비스는 임의의 호스트에 이용할 수 있고 인증할 필요없다.

<242> 다른 실시예

<243> 계층적 방법을 사용하는 대신, 유사한 결과들은 도 18에 도시된 바와 같이 데이터 베이스 방법을 사용하여 달성될 수 있다.

<244> 도 18에 도시된 바와 같이, 엔티티들의 증명서들 리스트, 인증 방법들, 결함 시도들의 최대 수, 및 증명서들의 최대 수는 메모리(10)의 제어기(12)에 의해 수행된 데이터베이스의 정책(독출, 키들 및 파티션들에 대한 기입 액세스, 보안 채널 요구)에 대한 상기 증명서 요구들에 관련된 제어기(12) 및 메모리(20)에 저장된 데이터베이스에 입력될 수 있다. 또한 데이터베이스에는 키들 및 파티션들에 대한 제한들 및 제약들이 저장된다. 따라서, 몇몇 엔티티들(예를 들어 시스템 관리자)은 이들 엔티티들이 모든 키들 및 파티션들에 액세스할 수 있는 것을 의미하는 기입 리스트상에 있을 수 있다. 다른 엔티티들은 블랙 리스트상에 있고, 임의의 정보에 액세스하기 위한 시도들은 차단될 것이다. 제한은 총체적이거나, 키 및/또는 파티션 특정일 수 있다. 이것은 특정 엔티티들만이 특정 키들 및 파티션들에 액세스하고, 특정 엔티티들은 그렇지 않다는 것을 의미한다. 제한들은 인크립트 또는 디크립트하기 위하여 사용된 키인 파티션에 무관하게 콘텐트 자체상에 놓일 수 있다. 따라서, 특정 데이터(예를 들어, 노래들)는 그들에 액세스하는 제 1 5개의 호스트 장치들에 의해 액세스될 수 있거나, 다른 데이터(예를 들어, 영화들)니 엔티티들이 액세스되었는지에 무관하게 제한된 횟수 동안 독출될 수 있는 속성들을 가질 수 있다.

<245> 인증

<246> 패스워드 보호

<247> - 패스워드 보호는 보호된 영역에 액세스하기 위하여 패스워드가 제공될 필요가 없는 것을 의미한다. 만약 하나 이상의 패스워드일 수 없으면 패스워드들은 독출 액세스 또는 독출/기입 액세스 같은 다른 권리들과 연관될 수 있다.

<248> - 패스워드 보호는 장치(예를 들어, 플래시 카드)가 호스트에 의해 제공된 패스워드를 검증될 수 있고, 즉 즉 장치는 장치 관리 보안 메모리 영역에 저장된 패스워드를 가지는 것을 의미한다.

<249> 발행 및 제한들

<250> - 패스워드들은 재연 공격에 민감하다. 패스워드가 각각의 프리젠테이션 후 변화하지 않기 때문에, 패스워드는 동일하게 재전송될 수 있다. 이와 같은 패스워드는 만약 보호될 데이터가 값어치 있으면 사용되지 않아야 하고, 통신 버스는 쉽게 액세스할 수 있는 것을 의미한다.

<251> - 패스워드는 저장된 데이터에 대한 액세스를 보호할 수 있지만 데이터(키가 아님)를 보호하기 위하여 사용되지 않는다.

<252> - 패스워드들과 연관된 보안 레벨을 증가시키기 위하여, 패스워드들은 마스터 키를 사용하여 다양화되고, 그 결과 해킹은 전체 시스템을 크랙하지 못한다. 세션 키 바탕 보안 통신 채널은 패스워드를 전송하기 위하여 사용될 수 있다.

<253> 도 19는 패스워드를 사용한 인증을 도시하는 흐름도이다. 엔티티는 시스템(10)(예를 들어, 플래시 메모리 카드)에 계정 id 및 패스워드를 전송한다. 시스템은 패스워드가 메모리의 패스워드가 일치하는지를 찾기 위하여 검사한다. 만약 일치하면, 인증된 상태는 리턴된다. 그렇지 않으면, 에러 카운터는 계정에 대해 증가되고, 엔티티는 계정 id 및 패스워드를 재입력하기를 요구받는다. 만약 카운터가 넘치면, 시스템은 액세스가 거부된 상태로 리턴한다.

<254> 대칭 키

<255> 대칭 키 알고리즘은 SAME(동일)한 키가 인크립트 및 디크립트하기 위하여 양쪽 측면들에 사용되는 것을 의미한다. 이것은 키가 통신 전에 사전 할당되는 것을 의미한다. 또한 각각의 측면은 서로의 리버스 알고리즘, 한쪽 측면상 인크립트 알고리즘 및 다른 측면에서 디크립트 알고리즘을 실행하여야 한다. 양쪽 측면들은 양쪽 알고리즘들이 통신을 실행할 필요가 없다.

<256> 인증

<257> - 대칭 키 인증은 장치들(예를 들어, 플래시 카드) 및 호스트가 동일한 키를 공유하고 동일한 암호화 알고리즘(다이렉트 및 리버스 예를 들어 DES 및 DES-1)을 공유하는 것을 의미한다.

<258> - 대칭 키 인증은 챌린지-응답(재연 공격에 대한 보호)을 의미한다. 보호된 장치는 다른 장치에 대한 챌린지를 생성하고 응답을 계산한다. 인증 장치는 응답을 거꾸로 전송하고 보호된 장치는 응답을 검사하고 이에 따라 인증을 유효화한다. 그 다음 인증과 연관된 권리들은 승인될 수 있다.

<259> 인증은:

<260> - 외부: 장치(예를 들어, 플래시 카드)가 외부 세계를 인증하고, 즉 장치가 주어진 호스트 또는 애플리케이션의 증명서들을 유효화한다.

<261> - 상호: 챌린지는 양쪽 측면들에서 생성된다.

<262> - 내부: 호스트 애플리케이션은 장치(예를 들어, 플래시 카드)를 인증하고, 즉 호스트는 장치가 이 애플리케이션에 대해 진짜인지를 검사한다.

<263> 전체 시스템의 보안 레벨을 증가시키기 위하여(즉, 하나의 차단은 모두를 차단하지 않음),

<264> - 대칭 키는 마스터 키를 사용하여 다양화와 함께 결합될 수 있다.

<265> - 상호 인증은 실제 챌린지인지를 보장하기 위하여 양쪽 측면으로부터 챌린지를 사용한다.

<266> 인크립션

<267> 대칭 키 암호화는 매우 효과적인 알고리즘이기 때문에, 암호화를 처리하기 위하여 강력한 CPU를 필요로 하지 않기 때문에 인크립션에 사용될 수 있다.

<268> 통신 채널을 보안하기 위하여 사용될 때:

<269> - 양쪽 장치들은 채널을 보안하기 위하여 사용된 세션 키를 알아야 한다(즉, 인크립트는 모든 배출 데이터를 인크립트하고 모든 인입 데이터를 디크립트한다). 이런 세션 키는 일반적으로 사전 공유된 비밀 대칭 키 또는 PKI를 사용하여 설정된다.

<270> - 양쪽 장치들은 동일한 암호화 알고리즘들 시그네이쳐를 알고 실행하여야 한다.

<271> 대칭 키는 데이터를 사인하기 위하여 사용될 수 있다. 이 경우 시그네이쳐는 인크립션의 부분 결과이다. 결과적 파티션을 유지하는 것은 키 값을 노출시키지 않고 필요한 만큼 다수번 사인하게 한다.

<272> 발행 및 제한들

<273> 대칭 알고리즘들은 매우 효과적이고 안전하지만, 그들은 사전 공유된 비밀을 바탕으로 한다. 발행은 동적 방식 및 랜덤하게(세션 키와 같이) 이런 비밀을 공유한다. 상기 생각은 공유된 비밀이 장기간 안전을 유지하기 힘들고 다중 사람들과 공유하는 것을 거의 불가능하게 한다.

<274> 이런 동작을 용이하게 하기 위하여, 공용 키 알고리즘은 그들을 공유하지 않고 비밀들의 교환을 허용하기 때문에 발전되었다.

<275> 비대칭 인증 과정

<276> 비대칭 키 바탕 인증은 보안 채널 통신을 위한 세션 키를 구성하는 일련의 데이터 패싱 명령들을 사용한다. 기본적인 프로토콜은 SSA 시스템에게 사용자를 인증한다. 프로토콜 변형들은 상호 인증을 위하여 허용하고, 여기서 사용자는 그가 사용하고자 하는 ACR, 및 두 개의 팩터 인증을 검증한다.

<277> SSA의 대칭 인증 프로토콜들은 바람직하게 공용 키 인트라구조(PKI) 및 RSA 알고리즘들을 사용한다. 이들 알고리즘에 의해 정의된 바와 같이, 인증 처리의 각각의 파티는 자신의 RSA 키 쌍을 생성하게 한다. 각각의 쌍은 공용 및 사적 키들로 구성된다. 키들이 익명이기 때문에, 상기 키들은 신원의 증명을 제공할 수 없다. PKI 층은 공용 키들 중 각각 하나를 사인하는 제 3 진짜 파티를 호출한다. 신뢰적인 파티의 공용 키는 서로를 인증하고 파티들의 공용 키들을 검증하기 위하여 사용된 파티들 사이에서 사전 공유된다. 일단 진짜가 설정되면(양쪽 파티들은 다른 파티에 의해 제공된 공용 키가 진실일 수 있다는 것이 결정된다), 프로토콜은 인증(각각의 파티가 매칭 사적 키를 홀딩하는 것을 검증) 및 키 교환을 계속한다. 이것은 하기된 도 22 및 23에 도시된 챌린지 응답 메카니즘을 통하여 수행될 수 있다.

<278> 사인 공용 키를 포함하는 구조는 증명서라 한다. 증명서들이 표시된 신뢰적인 파티는 증명서 인증국(CA)이라 한다. 파티가 인증되도록 하기 위하여 공용 키의 인증을 인증하는 RSA 키 쌍 및 증명서를 가진다. 증명서는 다른 (인증) 파티에 의해 신뢰성있는 증명서 인증국에 의해 사인된다. 인증 파티는 신뢰적인 CA의 공용 키를 소유한 것으로 예상된다.

<279> SSA는 증명서 체이닝을 허용한다. 이것은 식별된 파티의 공용 키가 다른 - 식별 타이에 의한 하나의 신뢰성으로부터 - CA에 의해 사인될 수 있다는 것을 의미한다. 이 경우 식별된 파티는 자신의 증명서 이에, 공용 키를 사인하는 CA의 증명서를 제공한다. 만약 이런 제 2 레벨 증명서가 다른 파티에 의해 신뢰되지 않으면(신뢰성있는 CA에 의해 사인되지 않음), 제 3 레벨 증명서는 제공될 수 있다. 이런 증명서 체인의 알고리즘에서, 각각의 파티는 공용 키를 인증하기 위한 필요한 증명서들의 완전한 리스트를 소유할 것이다. 이것은 도 23 및 24에 도시된다. 이런 타입의 ACR에 의한 상호 인증에 필요한 증명서들은 선택된 길이의 RSA 키 쌍들이다.

<280> SSA 증명서들

<281> SSA는 [X.509] 버젼 3 디지털 증명서들을 사용한다. [X.509]는 범용 표준이다; 여기에 기술된 SSA 증명서 프로파일은 증명서의 정의된 필드들의 콘텐트들을 지정 및 제한한다. 증명서 프로파일은 증명서 체인의 관리, SSA 증명서들의 유효화 및 증명서 취소 리스트(CRL) 프로파일에 대해 정의된 신뢰성 계층을 정의한다.

<282> 증명서는 공용 정보(내부에 있는 공용 키로서)로 생각되고 그러므로 인크립트되지 않는다. 그러나, 공용 키뿐 아니라 모든 다른 정보 필드들이 변경되지 않는 것을 검증하는 RSA 시그네이쳐를 포함한다.

<283> [X.509]는 각각의 필드가 ASN 1 표준을 사용하여 포맷화되고, 그 다음 데이터 인코딩을 위하여 DER 포맷을 사용한다.

<284> SSA 증명서 개요

<285> 도 20 및 21에 도시된 SSA 증명서 관리 아키텍처의 일 실시예는 비록 3개 이상의 계층의 보다 크거나 작은 수의 레벨들이 장치에 사용될 수 있지만, 호스트에 대한 계층의 제한되지 않은 레벨 및 장치에 대해 3개의 레벨 까지로 구성된다.

<286> 호스트 증명서 계층

<287> 장치는 두 개의 팩터들을 바탕으로 호스트들을 인증한다: 장치(ACR의 생성 중 저장된 ACR 증명서 같은)에 저장된 루트 CA 증명서 및 증명서/증명서 체인은 장치(특정 ACR에 대해)에 액세스하기 위한 엔티티에 의해 공급된다.

<288> 각각의 ACR에 대해 호스트 증명서 인증국은 루트 CA(이것은 ACR 증명서들에 존재하는 증명서이다)로서 사용한다. 예를 들어: 하나의 ACR에 대해 루트 CA는 "호스트 1 CA(레벨 2) 확실성"일 수 있고 다른 ACR에 대해 "호스트 루트 CA 확실성"일 수 있다. 각각의 ACR에 대해, 루트 CA에 의해 사인된 증명서(또는 루트 CA를 최종 엔티티 증명서에 접속하는 증명서 체인)를 홀딩하는 모든 엔티티는 ACR이 제공된 것에 로그인될 수 있고, 이것은 최종 엔티티 증명서에 대한 대응 사적 키를 가진다. 상술된 바와 같이, 증명서들은 공용 지식이고, 비밀로 유지되지 않는다.

<289> 루트 CA에 의해 발행된 모든 증명서 홀더들(및 대응 사적 키)이 ACR에 로그인된다는 사실은 특정 ACR에 대한 인증이 ACR 증명서에 저장된 루트 CA의 발행자에 의해 결정되는 것을 의미한다. 다른 말로, 루트 CA의 발행자는 ACR의 인증 방법을 관리하는 엔티티일 수 있다.

<290> 호스트 루트 증명서

<291> 루트 증명서는 SSA가 로그인(호스트)하기 시도하는 엔티티의 공용 키를 검증하기 시작하기 위하여 사용하는 신뢰성 있는 CA 증명서이다. 이런 증명서는 ACR이 ACR 증명서들의 일부로서 생성될 때 제공된다. 이것은 PKI 시스템에 대한 신뢰성의 근원이고, 그러므로 신뢰성 있는 엔티티(파더 ACR 또는 제조/구성 신뢰성있는 환경)에 의해 제공될 것이 가정된다. SSA는 증명서 시그네이쳐를 검증하기 위하여 공용 키를 사용하여 이 증명서를 검증한다. 호스트 루트 증명서는 시스템(10)의 도 1의 CPU(12)에 의해서만 바람직하게 액세스할 수 있는 장치의 비밀 키들을 이용하여 비휘발성 메모리(도 1에 도시되지 않음)에서 인크립되어 저장된다.

<292> 호스트 증명서 체인

<293> 인증 동안 SSA에 제공되는 증명서들이 있다. 호스트 증명서 체인의 재수집은 체인의 처리가 완료된 후 장치에 저장되어야 한다.

<294> 도 20은 다수의 다른 호스트 증명서 체인들을 도시하는 호스트 증명서 레벨 계층의 개략도이다. 도 20에 도시된 바와 같이, 호스트 증명서는 단지 3개만이 도시되지만 많은 다른 증명서 체인들을 가질 수 있다.

<295> A1. 호스트 루트 CA 증명서(502), 호스트 11 CA(레벨 2) 증명서(504) 및 호스트 증명서(506);

<296> B1. 호스트 루트 CA 증명서(502), 호스트 n CA(레벨 2) 증명서(508), 호스트 1 CA(레벨 3) 증명서(510), 호스트 증명서 512;

<297> C1. 호스트 루트 CA 증명서(502), 호스트 m CA(레벨 2) 증명서(508) 및 호스트 증명서(514).

<298> 상기 3개의 증명서 체인들(A1,B1 및 C1)은 호스트의 공용 키가 진짜인 것을 증명하기 위하여 사용될 수 있는 3개의 가능한 호스트 증명서 체인들을 도시한다. 상기 증명서 체인(A1) 및 도 20을 참조하여, 호스트 1 CA(레벨 2) 증명서(504)의 공용 키는 호스트 루트(CA)의 사적 키에 의해 사인되고(즉, 공용 키의 다이제스트를 인크립트하여), 상기 호스트 루트의 공용 키는 호스트 루트 CA 증명서(502) 내에 있다. 호스트 증명서(506)내 호스트 공용 키는 공용 키가 호스트 1 CA(레벨 2) 증명서(504)에 제공된 호스트 1 CA(레벨 2)의 사적 키에 의해 차례로 사인된다. 따라서, 호스트 루트 CA의 공용 키를 가진 엔티티는 상기 증명서 체인 A1의 인증을 검증할 수 있을 것이다. 제 1 단계로서, 엔티티는 호스트에 의해 전송된 호스트 1 CA(레벨 2) 증명서(504)내에 사인된 공용 키를 디크립트하기 위하여 소유물에서 호스트 루트 CA의 공용 키를 사용하고 디크립트된 사인 공용 키를 호스트에 의해 전송된 호스트 1 CA(레벨 2) 증명서(504)의 사인되지 않은 공용 키의 다이제스트와 비교한다. 만약 두 개가 매칭하면, 호스트 1 CA(레벨 2)의 공용 키는 인증되고, 엔티티는 호스트에 의해 전송된 호스트 증명서(506)내 호스트 1 CA(레벨 2)의 사적 키에 의해 사인된 호스트의 공용 키를 디크립트하기 위하여 호스트 1 CA(레벨 2)의 인증된 공용 키를 사용할 것이다. 만약 이런 디크립트된 사인 값이 호스트에 의해 전송된 호스트 증명서(506)의 공용 키의 다이제스트를 오프하는 것과 매칭하면, 호스트의 공용 키는 또한 인증된다. 증명서 체인들

B1 및 C1은 유사한 방식으로 인증에 사용될 수 있다.

<299> 체인 A1을 포함하는 상기 처리로부터 주지될 바와 같이, 엔티티에 의해 검증될 필요가 있는 호스트로부터의 제 1 공용 키는 호스트 루트 CA 증명서가 아닌 호스트 1 CA(레벨 2)의 키이다. 그러므로, 엔티티에 전송될 필요가 있는 모든 호스트는 호스트 1 CA(레벨 2) 증명서(504) 및 호스트 증명서(506)이므로, 호스트 1 CA(레벨 2) 증명서는 전송될 필요가 있는 체인내 제 1 증명서일 것이다. 상술된 바와 같이, 증명서 검증 시퀀스는 다음과 같다. 증명서 엔티티, 이 경우 메모리 장치(10)는 우선 루트 CA 아래 CA의 증명서(504)인 체인내 제 1 증명서의 공용 키의 진위를 검증한다. 상기 증명서의 공용 키가 진짜인 것으로 검증된 후, 장치(10)는 다음 증명서, 이 경우 호스트 증명서(506)를 검증하기 위하여 진행한다. 동일한 토큰에 의해, 검증의 유사한 시퀀스는 적용될 수 있고 여기서 증명서 체인은 둘 이상의 증명서들을 포함하고, 루트 증명서 바로 아래 증명서에서 시작하고 인증될 엔티티의 증명서에서 끝난다.

<300> 장치 증명서 계층

<301> 호스트는 두 개의 팩터들을 바탕으로 장치를 인증한다: 호스트에 저장된 장치 루트 CA 및 장치에 의해 호스트에 공급되는 증명서/증명서 체인(이것은 증명서들로서 ACR의 생성 후 장치에 공급된다). 호스트에 의해 장치를 검증하기 위한 처리는 상술된 호스트를 인증하는 장치와 유사하다.

<302> 장치 증명서 체인

<303> ACR의 키 쌍의 증명서들이 있다. 상기 증명서들은 ACR이 생성될 때 카드에 제공된다. SSA는 이들 증명서들을 개별적으로 저장하고 인증 동안 하나씩 호스트에 그들을 제공할 것이다. SSA는 호스트에 인증하기 위한 이들 증명서들을 사용한다. 장치는 비록 3과 다른 다수의 증명서들이 사용될 수 있지만, 3 증명서의 체인을 처리할 수 있다. 증명서들의 수는 하나의 ACR에서 다른 ACR로 가변할 수 있다. ACR이 생성될 때가 결정된다. 장치는 호스트에 증명서 체인을 전송할 수 있지만, 증명서 체인 데이터를 사용하지 않기 때문에 이들을 분석할 필요가 없다.

<304> 도 21은 저장 장치들 같은 SSA를 사용하는 장치들에 대한 n개의 다른 증명서 체인들을 통하여 1을 도시하기 위한 장치 증명서 레벨 계층을 도시하는 개략도이다. 도 21에 도시된 n개의 다른 증명서 체인들은 다음과 같다:

<305> A2. 장치 루트 CA 증명서(520), 장치 1 CA(제조자) 증명서(522) 및 장치 증명서(524);

<306> B2. 장치 루트 CA 증명서(520), 장치 n CA(제조자) 증명서(526) 및 장치 증명서(528).

<307> SSA 장치는 다른 제조자들을 통하여 1에 의해 제조될 수 있고, 각각은 그 자신의 장치 CA 증명서를 가진다. 그러므로, 특정 장치에 대한 장치 증명서의 공용 키는 제조자의 사적 키에 의해 사인될 것이고, 제조 시 공용 키는 장치 루트 CA의 사적 키에 의해 사인된다. 장치의 공용 키가 검증되는 방식은 상술된 호스트의 공용 키의 경우와 유사하다. 호스트를 위하여 상술된 체인 A1의 검증의 경우 처럼, 장치 루트 CA 증명서를 전송할 필요는 없고, 전송될 필요가 없는 체인들의 제 1 증명서는 장치 증명서 다음 장치 i CA(제조자) 증명서이고, i는 1 내지 n의 정수이다.

<308> 도 21에 도시된 실시예에서, 장치는 두 개의 증명서들을 제공할 것이다: 장치 i CA(제조자) 증명서는 자신의 장치 증명서를 뒤따른다. 장치 i CA(제조자) 증명서는 상기 장치를 제조한 제조자의 증명서이고 장치의 공용 키를 사인하기 위하여 사적 키를 제공하는 제조자이다. 장치 i CA(제조자) 증명서가 호스트에 의해 수신될 때, 호스트는 장치 i CA(제조자) 공용 키를 디크립트 및 검증하기 위하여 소유물에서 루트 CA의 사적 키를 사용한다. 만약 이런 검증이 실패하면, 호스트는 상기 처리를 중지하고 인증이 실패한 것을 장치에 통지한다. 만약 인증이 성공하면, 호스트는 다음 증명서에 대한 요구를 장치에 전송한다. 그 다음 장치는 유사한 방식으로 호스트에 의해 검증될 자신의 장치 증명서를 전송한다.

<309> 상술된 검증 처리들은 도 22 및 23에 보다 상세히 도시된다. 도 22에서, "SSM 시스템"은 여기에 기술된 SSA 시스템뿐 아니라 하기된 다른 기능들을 실행하는 소프트웨어 모듈이다. SSM은 CPU(12)의 메모리(20) 또는 비휘발성 메모리(도시되지 않음)에 저장된 데이터베이스를 가진 소프트웨어 또는 컴퓨터 코드로서 구현될 수 있고, RAM(12a)에 독출되고 CPU(12)에 의해 실행된다.

<310> 도 22에 도시된 바와 같이, 장치(10)의 SSM 시스템(542)이 호스트 시스템(540)을 인증하는 처리의 3개의 단계들이 있다. 제 1 공용 키 검증 단계에서, 호스트 시스템(540)은 SSM 시스템(542)에 SSM 명령의 호스트 증명서 체인을 전송한다. SSM 시스템(542)은 ACR(550)의 호스트 루트 증명서(548)에 배치된 루트 증명서 인증 공용 키를 사용하여 호스트 증명서(544) 및 호스트 공용 키(546)의 진위를 검증한다(블록 552). 여기서 루트 증명서 인증국 및 호스트 사이의 중간 증명서 인증국은 포함되고, 중간 증명서(549)는 블록(552)의 검증을 위해 사용된다.

증명서 또는 처리(블록 552)가 성공적인 것이 가정되면, SSM 시스템(542)은 제 2 단계로 진행한다.

<311> SSM 시스템(542)은 랜덤 수(554)를 생성하고 이를 호스트 시스템(540)에 챌린지로서 전송한다. 시스템(540)은 호스트 시스템의 사적 키(547)를 사용하여 랜덤 수(554)를 사인하고(블록 556) 챌린지에 대한 응답으로서 사인된 랜덤 수를 전송한다. 응답은 호스트 공용 키(546)를 사용하여 디크립트되고(블록 558) 랜덤 수(554)와 비교된다(블록 560). 디크립트된 응답이 랜덤 수(554)와 매칭하는 것이 가정되면, 챌린지 응답은 성공적이다.

<312> 제 3 단계에서, 랜덤 수(562)는 호스트 공용 키(546)를 사용하여 인크립트된다. 이런 랜덤 수(562)는 세션 키이다. 호스트 시스템(540)은 SSM 시스템(542)으로부터 인크립트된 수(562)를 디크립트(블록 564)하기 위하여 사적 키를 사용하여 세션 키를 얻을 수 있다. 이런 세션 키에 의해, 호스트 시스템(540) 및 SSM 시스템(542) 사이의 보안 통신은 시작될 수 있다. 도 22는 호스트 시스템(540)이 장치(10)의 SSM 시스템(542)에 의해 인증되는 경우 일방향 비대칭 인증을 도시한다. 도 23은 도 22의 일방향 인증 프로토콜과 유사한 이 방향 상호 인증 처리를 도시하는 프로토콜 도면이고, 여기서 도 23의 SSM 시스템(542)은 호스트 시스템(540)에 의해 인증된다.

<313> 도 24는 본 발명의 일실시예를 도시하는데 사용된 증명서 체인(590)의 도면이다. 상술된 바와 같이, 검증을 위해 제공될 필요가 있는 증명서 체인은 다수의 증명서들을 포함할 수 있다. 따라서 도 24의 증명서 체인은 총 아홉(9) 증명서들을 포함하고, 그 모두는 인증을 위하여 검증될 필요가 있을 수 있다. 배경 섹션에서 상술된 바와 같이, 증명서 검증을 위한 종래 시스템에서, 불안전한 증명서 체인이 전송되거나, 만약 전체 증명서가 전송되면, 증명서들은 임의의 특정 순서로 전송되지 않으므로, 수신부는 증명서들의 전체 그룹이 수신 및 저장될 때까지 증명서들을 분석할 수 없을 것이다. 체인에서 증명서들의 수가 미리 공지되지 않기 때문에, 이것은 문제를 제공할 수 있다. 다량의 저장 공간은 불특정 길이의 증명서 체인을 저장하기 위하여 비축될 필요가 있을 수 있다. 이것은 검증을 수행하는 저장 장치들에 대한 문제일 수 있다.

<314> 본 발명의 일 실시예는 증명서 체인이 저장 장치에 의해 검증될 순서와 동일한 순서로 호스트 장치들이 증명서 체인을 전송하는 시스템에 의해 문제가 제거된다는 인식을 바탕으로 한다. 따라서 도 24에 도시된 바와 같이, 증명서들의 체인(590)은 호스트 증명서 바로 아래 증명서인 증명서 체인(590)(1)에서 시작하고 호스트 증명서인 증명서(590)(9)에서 종료한다. 그러므로, 장치는 증명서(590)(9)의 호스트 공용 키가 검증될 때까지 증명서(590)(1)의 공용 키를 우선 검증하고, 증명서(590)(2)의 공용 키의 검증을 수행하고 및 기타 등등이 이루어진다. 그 다음 이것은 전체 증명서 체인(590)의 검증 처리를 완료한다. 따라서 만약 호스트 장치가 메모리 장치(10)에 증명서 체인이 검증될 순서와 동일한 순서 또는 시퀀스로 증명서 체인(590)을 전송하면, 메모리 장치(10)는 체인(590)의 전체 9 증명서들이 수신될 때까지 기다리지 않고 수신될 때 각각의 증명서 검증을 시작할 수 있다.

<315> 따라서, 일 실시예에서, 호스트 장치는 메모리 장치(10)에 한번에 체인(590)의 하나의 증명서를 전송한다. 메모리 장치(10)는 한번에 단일 증명서를 저장할 것이다. 증명서가 검증된 후, 체인의 최종 증명서를 제외하고 호스트에 의해 전송된 다음 증명서에 의해 겹쳐써질 수 있다. 이런 방식으로, 메모리 장치(10)는 언제라도 단일 증명서만을 저장하기 위한 공간을 비축할 필요가 있을 것이다.

<316> 메모리 장치는 전체 체인(590)이 수신될 시기를 알 필요가 있을 것이다. 따라서, 바람직하게 최종 증명서(590)(9)는 이것이 체인의 최종 증명서인 표시자 또는 표시를 포함한다. 이런 특징은 도 25에 도시되고, 호스트에 의해 메모리 장치(10)에 전송된 증명서 버퍼에 앞서는 제어 섹터에 정보를 도시하는 테이블이다. 도 25에 도시된 바와 같이, 증명서(590)(9)의 제어 섹터는 인수 이름, 즉 최종 플래그를 포함한다. 메모리 장치(10)는 수신된 증명서가 체인의 최종 증명서인지를 결정하기 위해, "최종" 플래그가 설정되었는지를 검사함으로써 증명서(590)(9)가 체인내 최종 증명서인지를 검증할 수 있다.

<317> 다른 실시예에서, 체인(590)의 증명서들은 하나씩 전송되지 않고, 하나, 둘, 또는 세 개의 증명서들의 그룹으로 전송될 수 있다. 명백히, 그룹들에 다른 수의 증명서들, 또는 동일한 수의 증명서들을 가진 그룹들은 사용될 수 있다. 따라서, 체인(590)은 증명서들(591,593,595,597 및 599)의 다섯(5) 개의 연속 문자열들을 포함한다. 각각의 문자열들은 적어도 하나의 증명서를 포함한다. 증명서들의 연속적인 문자열은 체인(시작 증명서)의 발행시 하나의 문자열 이전 문자열 다음 증명서, 체인(종료 증명서)의 하나의 문자열을 뒤따르는 문자열 바로 다음의 증명서, 및 시작 및 종료 증명서들 사이의 모든 증명서들을 포함하는 것이다. 예를 들어, 문자열(593)은 모두 3개의 증명서들 590(2), 590(3), 및 590(4)를 포함한다. 증명서들의 5개의 문자열들은 다음 시퀀스: 591,593,595,597에서 메모리 장치(10)에 의해 검증되고, 599에서 종료한다. 그러므로, 만약 5개의 문자열들이 메모리 장치(10)에 의해 수행되는 검증과 동일한 시퀀스로 전송 및 수신되면, 메모리 장치는 그들이 검증된 후 임의의 문자열들을 저장할 필요가 없을 것이고 최종 문자열을 제외한 모든 문자열들은 호스트로부터 도달하는

다음 문자열에 의해 겹쳐써질 수 있다. 종래 실시예에서 처럼, 체인의 최종 증명서인 것을 표시하기 위하여 특정 값으로 설정된 플래그 같은 표시기를 체인내 최종 증명서가 포함하는 것은 바람직하다. 이 실시예에서, 메모리 장치는 5개의 문자들의 증명서들 중 가장 큰 수를 저장하기에 적당한 공간을 비축할 필요가 있을 것이다. 따라서 만약 호스트가 우선 전송하고자 하는 가장 긴 문자열의 메모리 장치(10)를 통지하면, 메모리 장치(10)는 가장 긴 문자열에 대해 충분한 공간을 비축할 필요가 있을 것이다.

<318> 바람직하게, 호스트에 의해 전송된 체인의 각각의 증명서 길이는 증명서에 의해 증명된 공용 키의 길이의 4개보다 작다. 유사하게, 메모리 장치의 공용 키를 증명하기 위하여 메모리 장치(10)에 의해 호스트 장치에 전송된 증명서들의 길이는 바람직하게 증명서에 의해 증명된 공용 키의 길이의 4배보다 작다.

<319> 증명서 체인들의 검증을 위한 상술된 실시예는 도 26의 흐름도에 도시되고, 여기서 간략화를 위하여, 각각의 그룹의 증명서들의 수는 1인 것으로 가정된다. 도 26에 도시된 바와 같이, 호스트는 카드에 순차적으로 체인의 증명서들을 전송한다. 체인(통상적으로 상기 설명된 바와 같이 루트 증명서 다음 증명서)내 제 1 증명서에서 시작하여, 카드는 인증된 호스트로부터 증명서를 순차적으로 수신한다(블록 602). 그 다음 카드는 수신된 증명서들 각각을 검증하고 증명서 중 임의의 하나가 검증되면 처리를 중단한다. 만약 증명서들 중 임의의 하나가 실패로 검증되면, 카드는 호스트를 변형한다(블록들 604,606). 카드는 최종 증명서가 수신되고 검증되었는지(다이아몬드 608)를 검출할 것이다. 만약 최종 증명서가 수신 및 검증되지 않으면, 카드는 호스트로부터 증명서들을 계속 수신 및 검증하기 위하여 블록(602)으로 리턴한다. 만약 최종 증명서가 수신 및 검증되면, 카드는 증명서 검증(610) 후 다음 단계로 진행한다. 도 26의 특징들 및 하기 추후 도면들이 실시예들로서 메모리 카드들을 참조하지만, 이들 특징들이 메모리 카드들이 아닌 물리적 형태들을 가진 메모리 장치들에 또한 응용할 수 있다는 것은 이해될 것이다.

<320> 상기 처리는 카드가 인증될 때 호스트에 의해 수행되고 호스트는 도 27에 도시된다. 도 27에 도시된 바와 같이, 호스트는 카드에 체인의 다음 증명서를 전송한다(블록 620)(통상적으로 루트 증명서 다음 증명서에서 시작한다). 호스트는 인증 실패를 표시하는 중단 통지가 카드로부터 수신되었는지(다이아몬드 622)를 결정한다. 만약 중단 통지가 수신되면, 호스트는 중단된다(블록 624). 만약 중단 통지가 수신되지 않으면, 호스트는 "최종 플래그"가 전송된 최종 증명서에 설정되었는지(다이아몬드 626)를 검사하여 체인의 최종 증명서가 전송되었는지를 알기 위해 검사한다. 만약 최종 증명서가 전송되었다면, 호스트는 증명서 검증 후(블록 628) 다음 단계로 진행한다. 도 22 및 23에 도시된 바와 같이, 다음 단계는 세션 키 생성 다음 챌린지 응답일 수 있다. 만약 체인의 최종 증명서가 전송되지 않았다면, 호스트는 체인의 다음 증명서를 전송하기 위하여 블록(620)으로 리턴한다.

<321> 카드가 인증될 때 카드 및 호스트에 의해 취해지는 작용들은 도 28 및 29에 도시된다. 도 28에 도시된 바와 같이, 시작 후, 카드는 체인에 증명서를 전송하기 위한 호스트로부터의 요구를 기다린다(블록 630, 다이아몬드 632). 만약 호스트로부터의 요구가 수신되지 않으면, 카드는 다이아몬드(632)로 리턴할 것이다. 만약 호스트로부터의 요구가 수신되면, 카드는 전송되어야 하는 제 1 증명서에서 시작하여(통상적으로 루트 증명서 다음 증명서에서 시작)(블록 634), 체인의 다음 증명서를 전송할 것이다. 카드는 실패 통지가 호스트로부터 수신되었는지(다이아몬드 636)를 결정한다. 만약 실패 통지가 수신되면, 카드는 중단한다(블록 637). 만약 결함 통지가 수신되지 않으면, 카드는 최종 증명서가 전송되었는지(다이아몬드 638)를 결정한다. 만약 최종 증명서가 전송되지 않으면, 카드는 다이아몬드(632)로 리턴하고 체인의 다음 증명서를 전송하기 위한 호스트로부터의 다음 요구를 수신할 때까지 기다린다. 만약 최종 증명서가 전송되면, 카드는 다음 단계로 진행한다(블록 639).

<322> 도 29는 카드가 인증될 대 호스트에 의해 취해지는 작용들을 도시한다. 호스트는 제 1 증명서가 전송될 요구에서 시작하여(블록 640) 카드에 체인의 다음 증명서에 대한 요구를 전송한다. 그 다음 호스트는 각각의 수신된 증명서를 검증하고, 만약 검증이 실패하면(블록 642) 처리를 중단하고 카드를 변형한다. 만약 검증이 패스하면, 호스트는 최종 증명서가 수신되었고 성공적으로 검증되었는지(다이아몬드 644)를 찾는다. 만약 최종 증명서가 수신되지 않고 성공적으로 검증되지 않으면, 호스트는 체인의 다음 증명서에 대한 요구를 전송하기 위하여 블록(640)으로 리턴한다. 만약 최종 증명서가 수신되고 성공적으로 검증되면, 호스트는 증명서 검증 후(블록 646) 다음 단계로 진행한다.

<323> 증명서 취소

<324> 증명서가 발행될 때, 전체 유효 기간 동안 사용될 것이 예상된다. 그러나, 다양한 환경들은 유효 기간의 만료 이전에 증명서가 무효화되게 할 수 있다. 상기 환경들은 이름 변경, 서브젝트 및 CA(예를 들어, 사용인은 구성의 사용을 종료한다) 사이의 연관성 변경, 및 대응하는 사적 키의 타협 또는 의심되는 타협을 포함한다. 상기 환경들에서, CA는 증명서를 취소할 필요가 있다.

<325> SSA는 다른 방식을 증명서 취소를 수행하고, 각각의 ACR은 증명서들을 취소하기 위한 특정 방법을 위하여 구성될 수 있다. ACR은 취소 방법을 지원하지 않도록 구성될 수 있다. 이 경우, 각각의 증명서는 만료일 까지 유효한 것을 고려된다. 또는 증명서 취소 리스트들(CRL)은 사용될 수 있다. 다른 대안으로서, 취소 방법은 특정 애플리케이션, 또는 지정 애플리케이션에 특정될 수 있고, 하기에 설명될 것이다. ACR은 3 개의 취소 방법들 중 어느 것이 취소 값을 지정함으로써 중단될지를 지정한다. 만약 ACR이 취소 방법 없이 생성되면, ACR 소유자에 의해 작동될 수 있는 취소 방법을 적용하는 것이 가능하다. 메모리 장치 증명서들의 취소는 SSA 보안 시스템이 아닌 호스트에 의해 강화된다. ACR 소유자는 호스트 루트 증명서의 취소를 관리하고, 수행되는 메카니즘은 ACR의 증명서들을 업데이트하여 이루어진다.

<326> 증명서 취소 리스트(CRL)

<327> SSA 시스템은 증명서 취소 리스트(CRL)라 불리는 사인 데이터 구조를 주기적으로 발행하는 각각의 CA를 포함하는 취소 방법을 사용한다. CRL은 CA에 의해 사인되고 공용으로 자유롭게 이용 가능한 취소된 증명서들을 식별하는 시간 스탬프 리스트이다. 각각의 취소 증명서는 증명서 일련 번호에 의해 CRL에서 식별된다. CRL의 크기는 임의적이고 취소된 비 만료 증명서들의 수에 따른다. 장치가 증명서를 사용할 때(예를 들어, 호스트의 신원을 검증하기 위해), 장치는 증명서 시그네이쳐를 검사할뿐 아니라(및 유효성) CRL을 통하여 수신된 일렴 번호의 리스트에 대해 이를 검증한다. 만약 증명서의 일련 번호 같은 신원 확인 증명서를 발행한 CA에 의해 발행된 CRL에서 이루어지면, 이것은 증명서가 취소되었고 더 이상 유효하지 않은 것을 가리킨다.

<328> CRL은 증명서들을 유효화하기 위하여 사용하기 위해 진짜인 것으로 검증될 필요가 있을 것이다. CRL들은 CRL을 발행한 CA의 사적 키를 사용하여 사인되고 CA의 공용 키를 사용하여 사인된 CRL을 디크립트하여 진짜인 것으로 검증될 수 있다. 만약 디크립트된 CRL이 사인되지 않은 CRL의 다이제스트아 매칭하면, 이것은 CRL이 변경되지 않고 진짜인 것을 의미한다. CRL들은 해싱 알고리즘을 사용하여 다이제스트들을 얻기 위하여 주로 해시되고 다이제스트들은 CA의 사적 키에 의해 인크립트된다. CRL이 유효한지를 검증하기 위하여, 사인된 CRL(즉, 해시 및 인크립트된 CRL)은 디크립트 및 해시된 CRL(즉, CRL의 다이제스트)를 형성하기 위하여 CA의 공용 키를 사용하여 디크립트된다. 이것은 해시 CRL과 비교된다. 따라서, 검증 처리는 디크립트 및 해시된 CRL과 비교를 위하여 CRL을 해싱하는 단계를 주로 포함할 수 있다.

<329> CRL 방법의 특정들 중 하나는 증명서의 검증(CRL에 대해)이 CRL을 얻는 것과 분리되어 수행될 수 있다는 것이다. CRL들은 적당한 증명서들의 발행자들에 의해 사인되고, 상술된 방식으로 CRL들을 발행한 CA들의 공용 키들을 사용하여 증명서들의 검증과 유사한 방식으로 검증된다. 메모리 장치는 시그네이쳐가 CRL이고 CRL의 발행자가 증명서의 발행자와 일치한 것을 검증한다. CRL 방법의 다른 특성은 CRL들이 즉 비신뢰적인 서버들 및 신뢰적이지 않은 통신들을 통하여 증명서들 자체와 동일한 수단에 의해 분배되는 것이다. CRL들 및 그들의 특성들은 X.509 표준에 상세히 설명된다.

<330> CRL에 대한 SSA 인프라 구조

<331> SSA는 CRL 방법을 사용하는 호스트들의 취소를 위한 인프라구조를 제공한다. CRL 취소 방법을 가진 ACR을 바탕으로 RSA에 대해 인증할 때, 호스트는 설정 증명서 명령에 부가적인 필드로서 하나의 CRL(잠재적으로, 만약 증명서들이 발행자(CA-빈 장소)에 의해 취소되지 않으면)을 부가한다. 이런 필드는 증명서의 발행자에 의해 사인된 CRL을 포함할 것이다. 이 필드가 제공될 때, 메모리 장치(10)는 우선 설정 증명서 명령의 증명서를 검증한다. CRL 저장소를 얻고 액세스하는 것은 완전히 호스트의 책이다. CRL들은 그들이 유효한 동안의 시간 기간들(CRL 만료 시간 기간들 또는 CET)이 발행된다. 검증 동안, 만약 현재 시간이 이런 시간 기간 내에 아니라는 것이 발견되면, CRL은 결함있는 것으로 간주되고, 증명서 검증에 사용될 수 없다. 결과는 증명서의 인증이 실패하는 것이다.

<332> 종래 증명서 검증 방법들에서, 엔티티 인증 또는 검증은 증명서 인증국들(CA)로부터 증명서 취소 리스트들을 소유하거나 검색하는 것을 예상하고 제공된 증명서가 취소되었는지를 결정하기 위하여 리스트에 대한 인증을 위해 제공된 증명서의 일렴 번호들을 검사한다. 인증 또는 검증 엔티티가 메모리 장치인 경우, 메모리 장치는 CA들로부터 증명서 취소 리스들을 검색하기 위하여 자체적으로 사용될 수 없다. 만약 증명서 취소 리스트가 장치에 미리 저장되면, 상기 리스트는 낡아지게 되어 설치일 후 취소된 증명서들은 리스트에 나타나지 않을 것이다. 이것은 사용자들이 취소된 증명서를 사용하여 저장 장치에 액세스하게 할 것이다. 이것은 바람직하지 않다.

<333> 상기 문제는 인증되기를 원하는 엔티티가 메모리 장치(10)일 수 있는 인증 엔티티에 인증될 증명서와 함께 증명서 취소 리스트를 제공하는 경우 시스템에 의한 일 실시예에서 해결될 수 있다. 그 다음 인증 엔티티는 증명서

및 수신된 증명서 취소 리스트의 인증을 검증한다. 인증 엔티티는 증명서의 일련 번호 같은 증명서의 식별이 리스트에 제공되는지를 검사하여 인증서가 취소 리스트상에 있는지를 검사한다.

<334> 상기 측면에서, 비대칭 인증 방법은 호스트 장치 및 메모리 장치(10) 사이의 상호 인증을 위해 사용될 수 있다. 메모리 장치(10)에 인증되기를 원하는 호스트 장치는 증명서 체인 및 대응 CRL들 모두를 제공할 필요가 있을 것이다. 다른 한편 호스트 장치들은 CRL들을 얻기 위하여 CA들에 접속하기 위해 사용되어, 메모리 장치(10)가 호스트 장치들에 의해 인증될 때, 메모리 장치는 증명서들 또는 증명서 체인들과 함께 호스트 장치들에 CRL들을 제공될 필요가 없다.

<335> 최근에, 다른 구현되거나 독립형 뮤직 플레이어들, mp3 플레이어들, 셀룰러 전화들, 퍼스널 디지털 어시스탄트들, 및 노트북 컴퓨터들 같은 콘텐트를 재생하기 위하여 사용될 수 있는 다수의 다른 타입의 휴대용 장치들은 증가하고 있다. 증명서 인증국들로부터 증명서 검증 리스트들을 액세스하기 위하여 월드 와이드 웹에 상기 장치들을 접속하는 것은 가능하고, 많은 사용자들은 하루를 바탕으로 웹에 접속하지 않고, 새로운 콘텐트를 얻거나 몇주들 같이 가입들을 갱신한다. 그러므로, 상기 사용자들이 보다 빈번하게 증명서 인증국들로부터 증명서 취소 리스트들을 얻는 것은 귀찮다. 상기 사용자들에 대해, 증명서 취소 리스트 및 선택적으로 보호된 콘텐트에 액세스하기 위한 저장 장치에 제공될 필요가 있는 호스트 증명서는 저장 장치 자체의 보호되지 않은 영역에 바람직하게 저장될 수 있다. 많은 타입의 저장 장치들(예를 들어, 플래시 메모리들)에서, 저장 장치들의 보호되지 않은 영역들은 저장 장치들 자체가 아닌 호스트 장치들에 의해 관리된다. 이런 방식으로, 보다 많은 날들을 증명서 취소 리스트들을 얻기 위하여 사용자가 웹에 접속할 필요가 없다. 호스트 장치는 저장 장치의 보호되지 않은 영역으로부터 상기 정보를 검색하고 그 다음 저장 장치의 보호 콘텐트에 액세스하기 위한 저장 또는 메모리 장치에 상기 증명서 및 리스트를 제출한다. 보호 콘텐트 및 대응 증명서 취소 리스트에 액세스하기 위한 증명서가 특정 시간 기간들 동안 통상적으로 유효하지 않기 때문에, 그들이 유효한 한, 사용자는 최근 증명서들 또는 증명서 취소 리스트를 얻지 못한다. 상기 특징은 업데이트된 정보를 위해 증명서 인증국에 접속하지 않고, 양쪽이 유효한 동안 사용자들이 합리적 장기간들 동안 증명서 취소 리스트 및 증명서에 편리한 액세스를 가지게 한다.

<336> 상술된 처리들은 도 30 및 31의 흐름도에 도시된다. 도 30에 도시된 바와 같이, 호스트(24)는 호스트가 인증을 위한 메모리 장치에 제공할 증명서에 속하는 CRL을 메모리 장치(10)의 보안되지 않은 공용 영역으로부터 독출한다(블록 652). CRL이 메모리의 보안되지 않은 영역에 저장되기 때문에, CRL이 호스트에 의해 얻어질 수 있기 전에 인증이 필요하지 않다. CRL이 메모리 장치의 공용 영역에 저장되기 때문에, CRL의 독출은 호스트 장치(24)에 의해 제어된다. 호스트는 차례로 메모리 장치에 검증될 증명서를 CRL에 전송하고(블록 654) 만약 메모리 장치(10)로부터 실패 통지를 받지 않으면(블록 656) 다음 단계로 진행한다. 도 31을 참조하여, 메모리 장치는 호스트로부터 CRL 및 증명서를 수신하고(블록 658) 증명서 일련 번호가 CRL에 있는지(블록 660)뿐 아니라, 다른 측면들(예를 들어, CRL이 만료되었는지)에 있는지를 검사한다. 만약 증명서 일련 번호가 CRL에서 발견되거나 다른 이유들로 인해 실패하면, 메모리 장치는 호스트에 실패 통지를 전송한다(블록 662). 이런 방식으로, 다른 호스트들은 동일한 CRL이 다른 호스트들의 인증에 사용될 수 있기 때문에, 메모리 장치의 공용 영역에 저장된 CRL을 얻을 수 있다. 상술된 바와 같이, CRL을 사용하여 검증될 증명서는 사용자의 편리를 위하여 메모리 장치(10)의 보안되지 않은 영역에서 바람직하게 CRL와 함께 저장될 수 있다. 그러나, 증명서는 증명서가 발행된 호스트에 의해서만 메모리 장치에 인증을 위하여 사용할 수 있다.

<337> CRL이 도 32에 도시된 바와 같이 다음 업데이트를 위한 시간에 필드들에 포함되는 경우, 장치(10) 내 SSA는 현재 시간이 이 시간 이후 인지를 알기 위하여 이 시간에 대해 현재 시간을 검사한다; 만약 그렇다면, 인증은 실패한다. 따라서 SSA는 바람직하게 다음 업데이트를 위한 시간뿐 아니라 현재 시간에 대한 CET(또는 CRL이 메모리 장치 10에 의해 수신될 때 시간에 대해) 모두를 검사한다.

<338> 상술된 바와 같이, 만약 CRL이 취소된 증명서들의 긴 식별 리스트들을 포함하면, 호스트에 의해 제공된 증명서의 일련 번호에 대한 리스트를 처리(예를 들어, 해싱) 및 검색은 특히 처리 및 검색이 순차적으로 수행되면 장기간 걸릴 수 있다. 따라서, 처리를 가속하기 위하여, 이들은 동시에 수행될 수 있다. 게다가, 만약 전체 CRL이 처리 및 검색 전 수신될 필요가 있으면, 처리는 시간 소비적일 수 있다. 출원자들은 그들이 수신될 때(온 더 플라이) 상기 처리가 CRL의 부분들을 처리 및 검색함으로써 촉진될 수 있다는 것을 인식하였고, 이에 따라 CRL의 최종 부분들이 수신될 때, 처리는 완료된다.

<339> 도 33 및 34는 취소 방법들의 상기 특징들을 도시한다. 엔티티 인증시(예를 들어, 메모리 카드 같은 메모리 장치), 증명서 및 CRL은 인증되기를 원하는 엔티티로부터 수신된다(블록 702). 인크립트되지 않은 CRL 부분들은 처리되고(예를 들어 해시되고) 검색은 제공된 증명서의 신원 식별(예를 들어, 일련 번호)을 위하여 동시에 상기

부분들에서 수행된다. 처리된(예를 들어, 해시) CRL 부분들은 인증되기를 원하는 엔티티로부터 수신된 부분들로부터 디크립트된 CRL 부분들을 컴파일함으로써 형성된 완전한 디크립트 및 해시된 CRL과 비교되는 해시된 완성된 CRL에 컴파일된다. 인증은 만약 비교가 비교시 매칭이 없다는 것을 가리키면 실패한다. 인증한 엔티티는 또한 다음 업데이트에 대한 시간뿐 아니라 현재 시간에 대한 CET 모두를 검사한다(블록 706,708). 인증은 만약 제공된 인증서의 신원 식별부가 CRL에 있는 것으로 발견되거나, 만약 현재 시간이 CET내에 있거나, 만약 다음 업데이트된 CRL에 대한 시간이 패스되면(블록 710) 실패한다. 해시 CRL 부분들 및 몇몇 실행들에서 컴파일을 위한 디크립트된 해시 CRL 부분들을 저장하는 것은 다량의 메모리 공간을 요구하지 않을 수 있다.

<340> 엔티티(예를 들어, 호스트)가 인증되고자 할 때, 증명서 및 CRL을 인증 엔티티에 전송할 것이고(블록 722), 다음 단계(블록 724)로 진행한다. 이것은 도 34에 도시된다.

<341> 상기와 유사한 처리는 만약 엔티티가 인증을 위한 증명서 체인을 제공하면 실행될 수 있다. 상기 경우, 상술된 처리는 대응 CRL과 함께 체인의 각각의 증명을 위해 반복된다. 각각의 증명서 및 CRL은 그들이 증명서 체인의 나머지 및 대응 CRL들의 수신을 기다리지 않고 수신되기 때문에 처리될 수 있다.

<342> 신원 객체(IDO)

<343> 신원 객체는 플래시 메모리 카드 같은 메모리 장치(10)가 RSA 키 쌍 또는 다른 타입의 암호화 ID들을 저장하게 하도록 설계된 보호된 객체가다. 신원 객체는 신원들을 사인하고 검증하며, 데이터를 인크립트 및 디크립트하기 위하여 사용될 수 있는 임의의 타입의 암호화 ID를 포함한다. 신원 객체는 키 쌍의 공용 키가 진짜인 것을 증명하는 CA(또는 다중 CA들로부터의 증명서 체인)로부터의 증명서를 포함한다. 신원 객체는 외부 엔티티 또는 내부 카드 엔티티(즉, 신원 객체의 소유자로 불리는 장치 자체, 내부 애플리케이션, 등등)의 신원 증거를 제공하기 위하여 사용될 수 있다. 그러므로, 카드는 챌린지 응답 메카니즘을 통하여 호스트를 인증하기 위하여 RSA 키 쌍 또는 다른 타입의 암호화 ID들을 사용하는 것이 아니고, 오히려 그것에 제공된 사인한 데이터 스트림들을 통한 신원 증거를 사용한다. 다른 말로, 신원 객체는 소유자의 암호화 ID를 포함한다. 신원 객체의 암호화 ID에 액세스하기 위하여, 호스트는 우선 인증될 필요가 있다. 하기된 바와 같이, 인증 처리는 ACR에 의해 제어된다. 호스트가 성공적으로 인증된 후, 암호화 ID는 인증 파티에 소유자의 신원을 설정하기 위하여 신원 객체 소유자에 의해 사용될 수 있다. 예를 들어, 암호화 ID(예를 들어, 공용-사적 키 쌍의 사적 키)는 다른 파티에 의해 호스트를 통하여 제공된 데이터를 사인하기 위하여 사용될 수 있다. 사인된 데이터 및 신원 객체의 증명서는 신원 객체 대신 다른 파티에게 제공된다. 증명서의 공용-사적 키 쌍의 공용 키는 CA(즉, 신뢰적인 인증국)에 의해 진짜로 증명되어, 다른 파티는 이 공용 키가 진짜인 것을 신뢰할 수 있다. 그 다음 다른 파티는 증명서의 공용 키를 사용하여 사인된 데이터를 디크립트하고, 다른 파티에 의해 전송된 데이터와 디크립트된 데이터를 비교한다. 만약 디크립트된 데이터가 다른 파티에 의해 전송된 데이터와 매칭하면, 이것은 신원 객체의 소유자가 진짜 사적 키에 액세스하고, 그러므로 존재를 나태는 엔티티가 진짜인 것을 나타낸다.

<344> 신원 객체의 제 2 사용은 RSA 키 자체 같은 암호화 ID를 사용하여 IDO 소유자에게 지정된 데이터를 보호하는 것이다. 데이터는 IDO 공용 키를 사용하여 인크립트될 것으로 예상된다. 메모리 카드 같은 메모리 장치(10)는 데이터를 디크립트하기 위하여 사적 키를 사용할 것이다.

<345> IDO는 임의의 타입의 ACR에 대해 생성될 수 있는 객체가다. 일 실시예에서, ACR은 단지 하나의 IDO 객체를 가질 수 있다. 양쪽 데이터 사이닝 및 보호 특징들은 SSA 시스템이 ACR을 인증할 수 있는 임의의 엔티티에 제공하는 서비스들이다. IDO의 보호 레벨은 ACR의 로그인 인증 방법과 같이 높다. 임의의 인증 알고리즘은 IDO를 가지도록 한정된 ACR을 위해 선택될 수 있다. IDO를 가진 ACR은 IDO 공용 키를 얻기 위한 명령에 응답하여 증명서 체인을 제공한다.

<346> IDO가 데이터 보호를 위해 사용될 때, 카드로부터 출력된 디크립트된 데이터는 추가 보호를 요구할 수 있다. 상기 경우, 호스트는 이용할 수 있는 인증 알고리즘들 중 임의의 것을 통하여 설정된 보안 채널을 사용하기 위하여 조장된다.

<347> IDO를 생성할 때, 키 길이뿐 아니라 PKCS#1 버젼은 선택된다. 일 실시예에서, 공용 및 사적 키들은 PKCS#1 v2.1에서 정의된 바와 같은 (지수, 계수) 표현을 사용한다.

<348> 일 실시예에서, IDO의 생성 동안 포함된 데이터는 선택된 길이의 RSA 키 쌍, 및 반복적으로 공용 키의 인증을 증명하는 증명서들의 체인이다.

<349> IDO를 소유하는 ACR은 사용자 데이터의 사인을 허용할 것이다. 이것은 두 개의 SSA 명령들을 통하여 수행된다:

<350> - 사용자 데이터 설정: 사인될 자유 포맷 데이터 버퍼 제공.

<351> - SSA 시그네이쳐 얻음, 카드는 RSA 시그네이쳐를 제공할 것이다(ACR 사적 키 사용). 시그네이쳐의 포맷 및 크기는 객체 타입에 의존하는 PKCS#1 V1.5 또는 V2.1에 따라 설정될 수 있다.

<352> IDO를 사용한 동작은 도 35-37에 도시되고, 여기서 메모리 장치(10)는 플래시 메모리 카드이고, 카드는 IDO의 소유자이다. 도 35는 호스트에 전송된 데이터를 사인시 카드에 의해 수행된 처리를 도시한다. 도 35를 참조하여, 호스트가 상술된 트리 구조의 모드에서 ACR에 의해 제어되는 바와 같이 인증된 후(블록 802), 카드는 증명서에 대한 호스트 요구를 기다린다(다이아몬드 804). 요구를 수신한 후, 카드는 증명서를 전송하고 다음 호스트 요구에 대한 다이아몬드(804)로 리턴한다(블록 806). 만약 증명서들의 체인이 카드에 의해 소유된 IDO의 공용 키를 증명하기 위하여 전송될 필요가 있으면, 상기 작용들은 체인의 모든 증명서들이 호스트에 전송될 때까지 반복된다. 각각의 증명서가 호스트에 전송된 후, 카드는 호스트로부터 다른 명령들을 기다린다(다이아몬드 808). 만약 명령이 미리 설정된 기간 내에 호스트로부터 수신되면, 카드는 다이아몬드(804)로 리턴한다. 호스트로부터 데이터 및 명령을 수신한 후, 카드는 명령이 데이터를 사인하는지(다이아몬드 810)를 알기 위하여 검사한다. 만약 명령이 데이터를 사인하기 위한 것이면, 카드는 IDO에 사적 키를 사용하여 데이터를 사인하고 그 다음 사인된 데이터를 호스트로 전송하고(블록 812) 다이아몬드(804)로 리턴한다. 만약 호스트로부터의 명령이 호스트로부터 데이터를 사인하는 것이 아니면, 카드는 수신된 데이터를 디크립트하기 위하여 IDO의 사적 키를 사용하고(블록 814), 다이아몬드(804)로 리턴한다.

<353> 도 36은 호스트에 전송될 데이터의 카드 사이닝시 호스트에 의해 수행된 처리를 도시한다. 도 36을 참조하여, 호스트는 카드에 인증 정보를 전송한다(블록 822). 상술된 트리 구조 모드에서 ACR에 의해 제어되는 바와 같이 성공적인 인증 후, 호스트는 증명서 체인에 대한 요구들을 카드에 전송하고 체인을 수신한다(블록 824). 카드의 공용 키가 검증된 후, 호스트는 사인을 위한 데이터를 카드에 전송하고 카드의 사적 키에 의해 사인된 데이터를 수신한다(블록 826).

<354> 도 37은 호스트가 카드의 공용 키를 사용하여 데이터를 인크립트하고 카드에 인크립트된 데이터를 전송할 때 호스트에 의해 수행된 처리를 도시한다. 도 37을 참조하여, 호스트는 카드에 인증 정보를 전송한다(블록 862). ACR에 의해 제어되는 인증이 성공적으로 수행된 후, 호스트는 IDO의 카드 공용 키를 검증하기 위하여 필요한 증명서 체인에 대한 요구들을 카드에 전송하고(블록 8640, 데이터에 대한 요구들을 카드에 전송한다. IDO의 카드의 공용 키가 검증된 후, 호스트는 카드의 검증된 공용 키를 사용하여 카드로부터 데이터를 인크립트하고 이를 카드에 전송한다(블록들 866,868).

<355> 질문들

<356> 호스트들 및 애플리케이션들은 그들이 시스템 동작을 실행하기 위하여 함께 작동하는 메모리 장치 또는 카드와 관련한 특정 정보를 소유할 필요가 있다. 예를 들어, 호스트들 및 애플리케이션들은 메모리 카드에 저장된 애플리케이션들이 청원을 위해 이용할 수 있는지를 알 필요가 있다. 호스트에 의해 필요한 정보는 때대로 모두가 그것을 소유할 권리를 가지지 않는 것을 의미하는 공용 지식이 아니다. 따라서 인증 및 비인증 사용자들 사이를 구별하기 위하여, 호스트에 의해 사용될 수 있는 두 개의 질문 방법들을 제공할 필요가 있다.

<357> 일반 정보 질문

<358> 이 질문은 제한들 없이 시스템 공용 정보를 제공한다. 메모리 장치들에 저장된 기밀 정보는 두 개의 부분들을 포함한다: 공유 부분, 및 비공유 부분. 기밀 정보의 일 부분은 개별 엔티티들에 대한 소유권일 수 있는 정보를 포함하여, 각각의 엔티티는 다른 것들의 소유권 기밀 정보에 액세스하지 않고 그 또는 그녀 자신의 소유권 정보를 액세스하게 된다. 이런 타입의 기밀 정보는 공유되지 않고 공유되지 않은 부분 또는 기밀 정보 부분을 형성한다.

<359> 일반적으로 공용인 것으로 생각된 특정 정보는 몇몇 경우들에서 카드에 존재하는 애플리케이션들의 이름 및 수명 사이클 상태 같은 기밀로서 간주될 수 있다. 이것에 대한 다른 실시예는 공용인 것으로 고려되지만 몇몇 SSA 사용 경우들에 대해 기밀일 수 있는 루트 ACR 이름들일 수 있다. 이들 경우들에 대해, 시스템은 일반적인 정보 질문에 응답하여 인증되지 않은 사용자들이 아닌 모든 인증된 사용자들에게만 이용 가능하게 이 정보를 유지하기 위한 옵션을 제공한다. 상기 정보는 기밀 정보의 공유 부분을 구성한다. 기밀 정보의 공유 부분의 예는 루트 ACR 리스트 - 장치상 현재 제공된 모든 루트 ACR들의 리스트를 포함할 수 있다.

<360> 일반적인 정보 질물은 통한 공용 정보에 대한 액세스는 호스트/사용자가 ACR에 로그되게 할 필요가 없다. 따라

서 SSA 표준으로 알 수 있는 누군가는 정보를 실행 및 수신할 수 있다. SSA 측면들에서 이런 질문 명령은 세션 번호 없이 다루어진다. 그러나,만약 엔티티에 의한 기밀 정보의 공유 부분에 대한 액세스가 요구되면, 엔티티는 메모리 장치의 데이터에 대한 액세스를 제어하는 임의의 제어 구조들(예를 들어, ACR들 중 임의의 것)을 통하여 우선 인증될 필요가 있다. 성공적인 인증 후, 엔티티는 생성된 정보 질문을 통하여 기밀 정보의 공유 부분에 액세스할 수 있을 것이다. 상기 설명된 바와 같이, 인증 처리는 SSA 세션 번호 또는 액세스에 대한 id를 유발할 것이다.

<361> 분별있는 정보 질문

<362> 개별 ACR들 및 시스템 액세스 및 애셋들에 관련한 사적 정보는 분별있는 것으로 생각되고 명확한 인증을 필요로 한다. 따라서 이런 종류의 질문은 정보 질문에 대한 인증을 수신하기 전에 ACR 로그인 및 인증(만약 인증이 ACR에 의해 지정되면)을 호출한다. 이런 질문은 SSA 세션 번호를 필요로 한다.

<363> 두 가지 타입의 질문들이 상세하게 기술되기 전에, 질문들을 실행하기 위한 실제적인 해결책으로서 인덱스 그룹들의 개념을 우선 기술하는 것은 유용할 것이다.

<364> 인덱스 그룹들

<365> 잠재적인 SSA 호스트들에서 운용하는 애플리케이션들은 독출될 섹터들의 번호를 지정하기 위하여 호스트상 연산 시스템(OS) 및 시스템 구동기들에 의해 요구된다. 이것은 차례로 많은 섹터들이 매 SSA 독출 동작을 위하여 독출될 필요가 있는 방법을 호스트 애플리케이션이 아는 것을 요구하는 것을 의미한다.

<366> 질문 동작들의 성질이 그것을 요구하는 사람에게 일반적으로 공지되지 않은 정보를 공급하기 때문에, 호스트 애플리케이션이 질문을 발행하고 이런 동작에 필요한 섹터들의 양을 추측하는 것은 어렵다.

<367> 이런 문제를 해결하기 위하여 SSA 질문 출력 버퍼는 질문 요구에 따라 단지 하나의 섹터(512 바이트들)만으로 구성된다. 출력 정보의 일부인 객체들은 인덱스 그룹들이라 불리는 것으로 구성된다. 각각의 타입의 객체는 단일 섹터에 적합할 수 있는 객체들의 수의 원인이 되는 다른 바이트 크기를 가질 수 있다. 이것은 이런 객체의 인덱스 그룹을 정의한다. 만약 객체가 20 바이트 크기를 가지면, 이런 객체의 인덱스 그룹은 25 오브젝터들 까지를 포함한다. 만약 총 56 개의 오브제트들이 있으면 그들은 3 인덱스 그룹들로 구성되었고, 여기서 객체 '0'(제 1 객체)는 제 1 인덱스 그룹을 시작하고, 객체 '25'는 제 2 인덱스 그룹을 시작하고 인덱스 50은 제 3 및 최종 인덱스 그룹을 시작한다.

<368> 시스템 질문(일반 정보 질문)

<369> 이 질문은 장치에서 운용하는 애플리케이션들 및 다른 트리들 같이 설정된 현재 시스템 및 장치의 지원된 SSA 시스템에 관한 일반적인 공용 정보를 제공한다. 하기된 ACR 질문(분별있는 질문)과 유사하게, 시스템 질문은 몇몇 질문 옵션들을 제공하기 위하여 구성된다:

<370> - 일반 - SSA 지원 버젼.

<371> - SSA 애플리케이션들 - 운용 상태를 포함하는 장치상에 현재 제공된 모든 SSA 애플리케이션들 리스트.

<372> 상술된 정보는 공용 정보이고, ACR 질문에 대해, 얼마나 많은 섹터들이 질문 출력 버퍼를 위해 독출되는가를 호스트가 알기 위한 필요성을 제거하기 위하여, 호스트가 부가 인덱스 그룹들에 추가 질문하는 동안 하나의 섹터가 장치로부터 다시 전송될 것이다. 따라서 만약 루트 ACR 객체의 수가 인덱스 그룹 '0'에 대한 출력 버퍼 크기를 초과하면, 호스트는 다음 인덱스 그룹('1')에 다른 질문 요구를 전송할 수 있다.

<373> ACR 질문(분별있는 정보 질문)

<374> SSA ACR 질문 명령은 키 및 애플리케이션 ID들, 파티션들 및 차일드 ACR들 같은 ACR의 시스템 리소스들에 관한 정보를 ACR 사용자에게 공급하고자 한다. 질문 정보는 ACR에 로그되고 시스템 트리상 다른 ACR들에 관한 것은 없다. 다른 말로, 액세스는 ACR의 허용하에서 액세스할 수 있는 기밀 정보 부분이 포함되는 것으로 제한된다.

<375> 사용자가 질문할 수 있는 3개의 다른 ACR 객체들이 있다:

<376> - 파티션들의 이름 및 액세스 권리들(소유자, 독출, 기입).

<377> - 키 ID들 및 애플리케이션 ID들 - 이름 및 어드레스 권리들(소유자, 독출, 기입).

<378> - 차일드 ACR들 - 다이렉트 차일드 ACR의 ACR 및 AGP 이름.

<379> - IDO들 및 비밀 데이터 객체들(하기된) - 이름 및 액세스 권리들(소유자, 독출, 기입).

<380> ACR과 접속된 객체들의 수가 가변할 수 있기 때문에, 정보는 512 바이트들 이상일 수 있다 - 하나의 섹터. 객체들의 수를 미리 알지 못하고, 사용자는 얼마나 많은 섹터들이 전체 리스트를 얻기 위하여 장치의 SSA 시스템으로부터 독출될 필요가 있는지 알 방법이 없다. 따라서 SSA 시스템에 의해 제공된 각각의 객체 리스트는 상술된 시스템 질문들의 경우와 유사하게 인덱스 그룹들로 분할된다. 인덱스 그룹은 섹터상에 적합한 객체들의 수, 즉 얼마나 많은 객체들이 장치의 SSA 시스템으로부터 호스트로 하나의 섹터로 전송될 수 있는지이다. 이것은 장치의 SSA 시스템이 요구된 인덱스 그룹의 하나의 섹터를 전송하게 한다. 호스트/사용자는 질문된 객체들의 버퍼, 버퍼내 객체들의 수를 수신할 것이다. 만약 버퍼가 풀이면, 사용자는 다음 객체 인덱스 그룹에게 질문할 수 있다.

<381> 도 38은 일반적인 정보 질문을 포함하는 동작을 도시하는 흐름도이다. 도 38을 참조하여, SSA 시스템이 엔티티로부터 일반적인 정보 질문을 수신할 때(블록 902), 시스템은 엔티티가 인증되었는지(다이아몬드 904)를 결정한다. 만약 인증되면, 시스템은 엔티티에 공용 정보 및 기밀 정보의 공유 정보를 공급한다(블록 906). 만약 인증되지 않으면, 시스템은 엔티티에게 공용 정보만을 공급한다(블록 908).

<382> 도 39는 분별있는 정보 질문을 포함하는 동작을 도시하는 흐름도이다. 도 39를 참조하여, SSA 시스템이 엔티티로부터 분별있는 정보 질문을 수신할 때(블록 922), 시스템은 엔티티가 인증되었는지(다이아몬드 924)를 결정한다. 만약 인증되면, 시스템은 기밀 정보를 엔티티에게 공급한다(블록 926). 만약 인증되지 않으면, 시스템은 기밀 정보에 대한 엔티티의 액세스를 거절한다(블록 928).

<383> 특성 설정 확장(FSE)

<384> 많은 경우들에서 카드상 SSA 내의 데이터 처리 작용들(예를 들어, DRM 라이센스 객체 유효화)을 운용하는 것은 매우 바람직하다. 결과적인 시스템은 보다 안전하고, 보다 효과적이고, 대안 해결책에 관련하여 덜 의존하는 호스트일 것이고, 모든 데이터 처리 임무들은 호스트에서 실행된다.

<385> SSA 보안 시스템은 메모리 카드에 의해 저장, 관리, 및 보호된 객체들의 수집부에 액세스, 및 사용을 관리하기 위해 설계된 한 세트의 인증 알고리즘들 및 인증 정책을 포함한다. 일단 호스트가 액세스하면, 호스트는 메모리 장치에 저장된 데이터상 처리들을 수행할 것이고, 여기서 메모리 장치에 대한 액세스는 SSA에 의해 제어된다. 그러나, 데이터가 자연적으로 매우 특정한 애플리케이션이고 그러므로 데이터 포맷이나 데이터 처리 어느 것도 SSA에서 정의되지 않고, 이것은 장치들상에 저장된 데이터로 취급되지 않는 것은 가정된다.

<386> 본 발명의 일 실시예는 SSA 시스템이 메모리 카드의 호스트들에 의해 일반적으로 수행되는 몇몇 기능들을 호스트들이 실행하게 하기 위하여 강화될 수 있다는 인식을 바탕으로 한다. 따라서 호스트들의 몇몇 소프트웨어기능들은 두 부분들로 분할될 수 있다: 하나의 부분은 호스트들에 의해 수행되고 다른 부분은 카드에 의해 수행된다. 이것은 많은 애플리케이션들에 대한 데이터 처리의 보안성 및 효율성을 강화시킨다. 이런 목적을 위해, FSE로서 공지된 메카니즘은 SSA의 능력들을 강화하기 위하여 부가될 수 있다. 이런 방식으로 카드에 의해 실행되는 FSE의 호스트 애플리케이션은 내부 애플리케이션들, 또는 장치 내부 애플리케이션들이라 한다.

<387> 강화된 SSA 시스템은 카드 애플리케이션의 도입을 통하여 카드의 인증 및 액세스 제어를 제공하는 기본 SSA 명령 세트를 확장하기 위한 메카니즘을 제공한다. 카드 애플리케이션은 SSA의 것외에 서비스들(예를 들어, DRM 방법들, 전자 우편 서비스 트랜잭션들)을 실행하는 것으로 가정된다. SSA 특징 설정 확장부(FSE)는 소유권일 수 있는 데이터 처리 소프트웨어/하드웨어 모듈들로 표준 SSA 보안 시스템을 강화하기 위하여 설계된 메카니즘이다. SSA FSE 시스템에 의해 정의된 서비스들은 호스트 장치들이 이용 가능한 애플리케이션에 대한 카드를 질문하게 하고, 상술된 질문들을 사용하여 얻어질 수 있는 정보 외에 특정 애플리케이션을 선택 및 통신하게 한다. 상술된 일반적이고 분별있는 질문들은 이런 목적에 사용될 수 있다.

<388> SSA FSE에 설정된 카드 피쳐를 확장하기 위한 두 가지 방법들은 사용된다:

<389> - 서비스들 제공 - 이런 특징은 인증된 엔티티들이 소유권일 수 있는 통신 파이프로서 공지된 명령 채널을 사용하여 내부 애플리케이션과 직접 통신하게 할 수 있다.

<390> - SSA 표준 액세스 제어 정책들의 확장 - 이것은 내부 카드 애플리케이션들과 내부 보호 데이터 객체들(예를 들어, CEK들, 보안 데이터 객체들 또는 하기된 SDO들)을 연관시킴으로써 수행된다. 상기 객체가 액세스될 때마다, 만약 정의된 표준 SSA 정칙들이 만족되면, 연관된 애플리케이션은 호출되어 표준 SSA 정책들 외에 적어도 하나의 조건을 부과한다. 이런 조건은 바람직하게 표준 SSA 정책들과 충돌하지 않을 것이다. 액세스는 만약 이런 부

가적인 조건이 만족되면 승인된다. FSE의 능력들이 추가로 고쳐지기 전에, FSE의 아키텍처 측면들 및 통신 파이프 및 SDO는 지금 처리될 것이다.

<391> SSM 모듈 및 관련된 모듈들

<392> 도 40a는 본 발명의 실시예를 도시하기 위하여 호스트 장치(24)에 접속된 메모리 장치(10)(플래시 메모리 카드 같은)의 시스템 아키텍처(1000)의 기능 블록도이다. 카드(20)의 메모리 장치내 소프트웨어 모듈들의 주 구성요소들은 다음과 같다:

<393> SSA 전달 층(1002)

<394> SSA 전달 층은 카드 프로토콜에 종속된다. 이것은 카드(10)의 프로토콜 층상 호스트측 SSA 요구들(명령들)을 처리하고 그 다음 그들을 SSM API에 릴레이한다. 모든 호스트 카드 동기화 및 SSA 명령 식별은 이런 모듈에서 수행된다. 전달 층은 호스트(24) 및 카드(10) 사이에서 모든 SSA 데이터 전달을 책임진다.

<395> 보안 서비스들 모듈 코어(SSM 코어)(1004)

<396> 이 모듈은 SSA 실행의 중요 부분이다. SSM 코어는 SSA 아키텍처를 실행한다. 보다 특히 SSM 코어는 SSA 트리 및 ACR 시스템 및 시스템을 형성하는 상술된 대응 룰들 모두를 실행한다. SSM 코어 모듈은 인크립션, 디크립션 및 해싱 같은 암호화 특징들 및 SSA 보안을 지원하기 위하여 암호화 라이브러리(1012)를 사용한다.

<397> SSM 코어 API(1006)

<398> 이것은 호스트 및 내부 애플리케이션들이 SSA 동작들을 수행하기 위하여 SSM 코어와 인터페이스하는 층이다. 도 40a에 도시된 바와 같이, 양쪽 호스트(24) 및 내부 장치 애플리케이션들(1010)은 동일한 API를 사용할 것이다.

<399> 소스 애플리케이션 관리자 모듈(SAMM)(1008)

<400> SAMM은 SSA 시스템의 일부가 아니고 SSA 시스템과 인터페이스하는 내부 장치 애플리케이션들을 제어하는 카드내 중요 모듈이다.

<401> SAMM은 하기를 포함하는 애플리케이션들을 운용하는 모든 내부 장치를 관리한다:

<402> 1. 애플리케이션 라이프사이클 모니터 및 제어.

<403> 2. 애플리케이션 시작.

<404> 3. 애플리케이션/호스트/SSM 인터페이스.

<405> 장치 내부 애플리케이션들(1010)

<406> 카드측상에서 운용하기 위해 승인된 애플리케이션들이 있다. 상기 애플리케이션들은 SAMM에 의해 관리되고 SSA 시스템에 액세스할 수 있다. SSM 코어는 호스트 애플리케이션들 및 내부 애플리케이션들 사이의 통신 파이프를 제공한다. 상기 내부 운용 애플리케이션들에 대한 예들은 DRM 애플리케이션들 및 하기에 추가로 설명되는 바와 같은 일회용 패스워드(OTP) 애플리케이션들이다.

<407> 장치 관리 시스템(DMS)(1011)

<408> 이것은 추후 선적(일반적으로 포스트 발행이라 함) 모드에서 카드의 시스템 및 애플리케이션 펌웨어를 업데이트할 뿐 아니라 서비스들을 부가/제거하기 위하여 필요한 처리들 및 프로토콜들을 포함하는 모듈이다.

<409> 도 40b는 SSM 코어(1004)의 내부 소프트웨어 모듈들의 기능 블록도이다. 도 40b에 도시된 바와 같이, 코어(1004)는 SSA 명령 핸들러(1022)를 포함한다. 핸들러(1022)는 명령들이 SSA 관리자(1024)로 패스되기 전에 호스트로부터 또는 장치 내부 애플리케이션들(1010)로부터 발생하는 SSA 명령들을 분석한다. AGP들 및 ACR 같은 SSA 보안 데이터 구조들뿐 아니라 SSA 룰들 및 정책들은 SSA 데이터베이스(1026)에 저장된다. SSA 관리자(1024)는 ACR들 및 AGP들 및 데이터베이스(1026)에 저장된 다른 제어 구조들에 의해 가해진 제어를 실행한다. IDO들 같은 다른 객체들, 및 보안 데이터 객체들은 SSA 데이터베이스(1026)에 저장된다. SSA 관리자(1024)는 ACR들 및 AGP들 및 데이터베이스(1026)에 저장된 다른 제어 구조들에 의해 가해진 제어를 실행한다. SSA를 포함하지 않는 비보안 동작들은 SSA 비보안 동작들 모듈(1028)에 의해 조정된다. SSA 아키텍처 하의 보안 동작들은 SSA 보안 동작들 모듈(1030)에 의해 조정된다. 모듈(1032)은 모듈(1030)을 암호화 라이브러리(1012)에 접속하는 인터페이스이다. 1034는 도 1의 플래시 메모리(20)에 모듈들(1026 및 1028)을 접속하는 층이다.

<410> 통신(또는 패스) 파이프

<411> 패스 파이프 객체들은 인증된 호스트 측 엔티티들이 SSM 코어 및 SAMM에 의해 제어되는 바와 같이 내부 애플리케이션들과 통신하게 한다. 호스트 및 내부 애플리케이션 사이의 데이터 전달은 전송 및 수신 명령들(이하에 정의됨)을 통해 수행된다. 실제 명령들은 특정 애플리케이션이다. 파이프를 생성하는 엔티티(ACR)는 파이프 이름 및 채널을 개방할 애플리케이션의 ID를 제공할 필요가 있을 것이다. 모든 다른 보호된 객체들로 인해, ACR은 소유자가 되고 사용 권리들뿐 아니라 소유권을 표준 위임 룰들 및 제한들에 따라 다른 ACR에 위임하게 한다.

<412> 인증된 엔티티는 만약 생성_파이프 허용들이 ACAM으로 설정되면 파이프 객체들을 생성하게 할 것이다. 내부 애플리케이션과 통신은 기입 또는 독출 파이프 허용들이 PCR에서 설정되면 허용될 것이다. 소유권 및 액세스 권리들 위임은 만약 엔티티가 파이프 소유자이거나 위임 액세스 권리들이 PCR에서 설정되면 허용된다. 소유권을 다른 ACR에 위임할 때 모든 다른 허용으로 인해, 본래 소유자가 모든 허용들로부터 이런 장치 애플리케이션으로 스트립되게 할 것이다.

<413> 바람직하게 단지 하나의 통신 파이프는 특정 애플리케이션을 위해 생성된다. 제 2 파이프를 생성하고, 이를 이미 접속된 애플리케이션에 접속하기 위한 시도는 바람직하게 SSM 시스템(1000)에 의해 거절될 것이다. 따라서, 바람직하게 장치 내부 애플리케이션들(1010) 중 하나 및 통신 파이프 사이에 일 대 일 관계가 있다. 그러나, 다중 ACR들은 하나의 장치 내부 애플리케이션(위임 메카니즘을 통해)과 통신할 수 있다. 단일 ACR은 몇몇 장치 애플리케이션들과 통신할 수 있다(다른 애프리케이션들에 접속된 다중 파이프들의 위임 또는 소유권을 통해). 다른 파이프들을 제어하는 ACR들은 바람직하게 전체적으로 분리된 트리들의 노드에 배치되어, 통신 파이프들 사이의 혼선은 없다.

<414> 호스트 및 특정 애플리케이션 사이에서 데이터 전달은 다음 명령들을 사용하여 행해진다:

<415> - 기입 패스 - 호스트로부터 장치 내부 애플리케이션으로 인증되지 않은 데이터 버퍼 전달.

<416> - 독출 패스 - 호스트로부터 장치 내부 애플리케이션으로 인증되지 않은 데이터 버퍼 전달 및 일단 내부 처리가 수행되면 호스트에 인증되지 않은 데이터 버퍼를 다시 출력.

<417> 기입 및 독출 패스 명령들은 호스트들이 통신하고자 하는 장치 내부 애플리케이션(1008)의 ID를 파라미터로서 제공한다. 엔티티들 허용은 유효화되고 만약 요구한 엔티티(즉, 이 엔티티가 사용중인 세션을 호스팅하는 ACR)가 요구된 애플리케이션에 접속된 파이프를 사용하기 위한 허용을 가지면, 데이터 버퍼는 해석되고 명령은 실행될 것이다.

<418> 이런 통신 방법은 호스트 애플리케이션이 SSA ACR 세션 채널을 통하여 내부 장치 애플리케이션에 판매자/소유권 지정 명령들을 패스하게 한다.

<419> 보안 데이터 객체(SDO)

<420> FSE와 관련하여 사용될 수 있는 유용한 객체는 SDO이다.

<421> SDO는 민감한 정보의 보안 저장을 위한 범용 컨테이너로서 사용한다. CEK 객체들과 유사하게, 이것은 ACR에 의해 소유되고 액세스 권리들 및 소유권은 ACR들 사이에서 위임될 수 있다. 이것은 미리 정의된 정책 제한들에 따라 보호 및 사용된 데이터를 포함하고, 선택적으로 장치 내부 애플리케이션(1008)에 링크를 가진다. 민감한 데이터는 바람직하게 SSA 시스템에 의해, 그러나 객체의 소유자 및 사용자들에 의해 사용되지 않고, 해석되지 않는다. 다른 말로, SSA 시스템은 이것에 의해 조정된 데이터 내 정보를 판별하지 않는다. 이런 방식으로, 객체내 데이터의 소유자들 및 사용자들은 데이터가 호스트들 및 데이터 객체들 사이에서 패스될 때, SSA 시스템과의 인터페이스로 인한 민감한 정보의 손실에 덜 관련될 수 있다. 따라서, SDO 객체들은 호스트 시스템(또는 내부 애플리케이션들)에 의해 생성되고, CEK들이 생성되는 것과 유사한 방식으로 문자열 ID가 할당된다. 생성 후 호스트는 이름 외에, SDO에 링크된 애플리케이션에 대한 애플리케이션 ID 및 SSA에 의해 저장되고, 보전 검증되고, 검색될 데이터 블록을 제공한다.

<422> CEK들과 유사하게, SDO(들)은 SSA 세션내에서만 생성된다. 세션을 개방하기 위하여 사용된 ACR은 SDO의 소유자가 되고 이를 삭제하고, 민감한 데이터를 기입하고 독출하고, 또한 SDO를 다른 ACR(동일한 AGP내 또는 차일드)에 액세스하기 위한 소유권 및 허용을 위임하기 위한 권리를 가진다.

<423> 기입 및 독출 동작들은 SDO의 소유자에 대해 배타적으로 비축된다. 기입 동작은 제공된 데이터 버퍼를 가진 기존 SDO 객체 데이터를 겹쳐쓴다. 독출 동작은 SDO의 완전한 데이터 레코드를 검색할 것이다.

<424> SDO 액세스 동작들은 적당한 액세스 허용들을 가진 비 소유자 ACR들에 대해 허용된다. 다음 동작들은 정의된다:

<425> - SDO 설정, 애플리케이션 ID는 정의된다: 데이터는 애플리케이션 ID를 가진 내부 SSA 애플리케이션에 의해 처리될 것이다. 애플리케이션은 SDO와 연관하여 호출된다. 선택적인 결과로서, 애플리케이션은 SDO객체를 기입할 것이다.

<426> - SDO 설정, 애플리케이션 ID는 널(null)이다: 이런 옵션은 유효하지 않고 불법 명령 에러를 프롬프트할 것이다. 설정 명령은 카드내에서 운용하는 내부 애플리케이션을 요구한다.

<427> - SDO 얻음, 애플리케이션 ID는 정의된다: 상기 요구는 애플리케이션 ID를 가진 장치 내부 애플리케이션에 의해 처리될 것이다. 애플리케이션은 SDO와 관련하여 호출된다. 출력은 비록 정의되지 않지만 상기 요구자에게 다시 전송될 것이다. 애플리케이션은 선택적으로 SDO 객체를 독출할 것이다.

<428> - SDO 얻음, 애플리케이션 ID는 널이다: 이런 옵션은 유효하지 않고 불법 명령 에러를 프롬프트할 것이다. 얻음 명령은 카드에서 운용하는 내부 애플리케이션을 요구한다.

<429> - SDO 관련 허용들: ACR은 SDO 소유자이거나 액세스 허용들을 가진다(설정, 얻음 또는 둘다). 게다가, ACR은 그의 액세스 권리들을 소유하지 않은 SDO, 다른 ACR에 전달할 수 있게 될 수 있다. ACR은 SDO(들)을 생성하고 만약 ACAM 허용을 가지면 액세스 권리들을 위임하기 위하여 명백하게 허용된다.

<430> 내부 ACR

<431> 내부 ACR은 장치(10)에 대한 외부 엔트리들이 이런 ACR에 로그인될 수 없는 것을 제외하고 PCR을 가진 임의의 ACR과 유사하다. 대신, 도 40b의 SSA 관리자(1024)는 자동으로 내부 ACR에 로그인한다. 제어하의 객체들 또는 그것과 연관된 애플리케이션들은 호출된다. 액세스를 얻고자하는 엔티티가 카드 또는 메모리 장치에 대한 내부 엔티티일 때, 인증할 필요가 없다. SSA 관리자(1024)는 내부 통신을 수행하도록 내부 ACR에 세션 키를 간단히 패스할 것이다.

<432> FSE의 능력들은 두 개의 예들을 사용하여 도시될 것이다: 일회용 패스워드 생성 및 디지털 권리들 관리. 일회용 패스워드 생성 예가 기술되기 전에, 이중 팩터 인증 발행이 우선 처리될 것이다.

<433> OTP 실시예

<434> 이중 팩터 인증(DFA)

<435> DFA는 표준 사용자 인증서들(즉 사용자 이름 및 패스워드)에 부가적인 비밀, "제 2 팩터"를 부가함으로써 예로서 웹 서비스 서버에 개인 로그인들의 비밀을 강화하기 위하여 설계된 인증 프로토콜이다. 제 2 보안은 통상적으로 사용자가 그의 소유물에 있는 물리적 보안 토큰에 저장된 무언가이다. 로그인 처리 동안 사용자는 로그인 증명서의 일부로서 소유물의 증거를 제공할 필요가 있다. 소유물을 증명하기 위한 일반적으로 사용된 방식은 일회용 패스워드(OTP), 보안 토큰에 의해 생성되고, 출력되는 단일 로그인만을 위한 패스워드 상품을 사용하는 것이다. 만약 사용자가 올바른 OTP를 제공할 수 있으면, 토큰 없이 OTP를 계산하는 것을 암호적으로 실행할 수 없기 때문에 토큰의 소유물의 충분한 증거로서 생각된다. OTP가 하나의 로그인만에 대해 우수하기 때문에, 사용자가 로그인 시간에 토큰을 가져야 하는데, 그 이유는 이전 로그인으로부터 캡쳐된 기존 패스워드의 사용이 그 이상 임의의 우수함을 수행하지 않을 것이다.

<436> 다음 섹션들에서 기술된 제품은 OTP 시리즈의 다음 패스워드를 계산하기 위해 SSA 보안 데이터 구조, 플러스 하나의 FSE 설계를 이용하여, 다중 "가상" 보안 토큰들을 가진 플래시 메모리 카드를 실행하고, 각각 하나는 다른 시리즈의 패스워드들을 생성한다(이것은 다른 웹 사이트들로 로그인을 위해 사용될 수 있다). 이런 시스템의 블록도는 도 41에 도시된다.

<437> 완전한 시스템(1050)은 인증 서버(1052), 인터넷 서버(1054) 및 토큰(1058)을 가진 사용자(1056)를 포함한다. 제 1 단계는 인증 서버 및 사용자 사이의 공유된 비밀을 승인하는 것이다(씨드 제공이라 함). 사용자(1056)는 발행될 비밀 또는 씨드를 요구할 것이고 이를 보안 토큰(1058)에 저장할 것이다. 다음 단계는 특정 웹 서비스 서버와 발행된 비밀 또는 씨드를 결합하는 것이다. 일단 이것이 행해지면, 인증은 발생할 수 있다. 사용자는 토큰에게 OTP를 생성할 것을 명령한다. 사용자 이름 및 패스워드를 가진 OTP는 인터넷 서버(1054)에 전송된다. 인터넷 서버(1054)는 사용자 신원을 검증하기를 요청하는 인증 서버(1052)에 OTP를 전송한다. 인증 서버는 OTP를 생성하고, 토큰을 가진 공유된 비밀로부터 이것이 생성되기 때문에, 토큰으로부터 생성된 OTP를 매칭하여야 한다. 만약 매칭이 발견되면, 사용자 신원은 검증되고 인증 서버는 사용자 로그인 처리를 완료할 인터넷 서버

(1054)에 긍정적인 수신응답을 리턴할 것이다.

<438> OTP 생성을 위한 FSE 실행은 다음 특성들을 가진다:

<439> - OTP 씨드는 카드에 안전하게 저장된다(인크립트된).

<440> - 패스워드 생성 알고리즘은 카드 내부에서 실행된다.

<441> - 장치(10)는 다중 가상 토큰들을 대리 실행하고, 상기 토큰들 각각은 다른 씨드를 저장하고 다른 패스워드 생성 알고리즘들을 사용할 수 있다.

<442> - 장치(10)는 인증 서버로부터 장치로 씨드를 전달하기 위하여 보안 프로토콜을 제공한다.

<443> OTP 씨드 제공 및 OTP 생성을 위한 SSA 특성들은 도 42에 도시되고, 실선 화살표들은 소유권 및 액세스 권리들을 도시하고, 파선 화살표들은 연관성들 또는 링크를 도시한다. 도 42에 도시된 바와 같이, SSA FSE 시스템(1100)에서, 소프트웨어 프로그램 코드 FSE(1102)는 N 애플리케이션 ACR들(1106)에 의해 제어되는 하나 또는 그 이상의 통신 파이프들(1104)을 통하여 액세스될 수 있다. 하기된 실시예들에서, 단지 하나의 FSE 소프트웨어 애플리케이션은 도시되고, 각각의 FSE 애플리케이션을 위해, 단지 하나의 통신 파이프가 있다. 그러나, 하나 이상의 FSE 애플리케이션이 사용될 수 있다는 것이 이해될 것이다. 단지 하나의 통신 파이프가 도 42에 도시었지만, 다수의 통신 파이프들이 사용될 수 있는 것은 이해될 것이다. 모든 상기 변형들은 가능하다. 도 40a, 40b 및 42를 참조하여, FSE(1102)는 OTP제공을 위하여 사용된 애플리케이션일 수 있고 도 40a의 장치 내부 애플리케이션들(1010)의 서브세트를 형성한다. 제어 구조들(ACR들 1101,1103,1106,1110)은 SSA의 보안 데이터 구조들의 일부이고 SSA 데이터베이스(1026)에 저장된다. IDO(1120) 같은 데이터 구조들, SDO 객체들(1122), 및 통신 파이프(1104)는 또한 SSA 데이터베이스(1026)에 저장된다.

<444> 도 40a 및 40b를 참조하여, ACR들 및 데이터 구조들을 포함하는 보안 관련 동작들(예를 들어, 세션들에서 데이터 전달, 및 인크립션, 디크립션 및 해싱 같은 동작들)은 인터페이스(1032) 및 암호화 라이브러리(1012)의 도움으로 모듈(1030)에 의해 조정된다. SSM 코어 API(1006)는 호스트들(외부 ACR들)과 상호작용하는 ACR들 및 수행하지 않고 따라서 호스트들 대 장치 내부 애플리케이션들(1010)을 포함하는 동작들 사이를 구별하지 못하는 내부 ACR들 사이를 구별하지 못한다. 이런 방식으로, 동일한 제어 메카니즘은 호스트측 엔티티들에 의한 액세스 및 장치 내부 애플리케이션들(1010)에 의한 액세스를 제어하기 위하여 사용된다. 이것은 호스트 측 애플리케이션들 및 장치 내부 애플리케이션들(1010) 사이의 데이터 처리를 분할하기 위한 융통성을 제공한다. 내부 애플리케이션들(1010)(예를 들어, 도 42의 FSE 1102)은 내부 ACR들(예를 들어, 도 42의 ACR 1103)과 연관되고 상기 ACR들의 제어를 통하여 호출된다.

<445> 게다가, ACR들 및 연관된 SSA 룰들 및 정책들을 가진 AGP들 같은 보안 데이터 구조들은 바람직하게 SDO들의 콘텐트로부터 유도될 수 있는 콘텐트 또는 정보 같은 중용한 정보에 대한 액세스를 제어하여, 외부 또는 내부 애플리케이션들은 SSA 룰들 및 정책들에 따라 이런 콘텐트 또는 정보에만 액세스할 수 있다. 예를 들어, 만약 두 개의 다른 사용자들이 데이터를 처리하기 위하여 장치 내부 애플리케이션들(1010) 중 개별 하나를 호출할 수 있으면, 독립된 계층 트리들에 배치된 ACR들은 두 개의 사용자들에 의한 액세스를 제어하기 위하여 사용되어, 그들 사이에 혼선이 없다. 이런 방식으로, 양쪽 사용자들은 콘텐트 또는 정보의 제어를 잃는 SDO들 내 콘텐트 또는 정보의 소유자들의 일부상 걱정없이 데이터를 처리하기 위한 장치 내부 애프리케이션들(1010)의 공통 세트에 액세스할 수 있다. 예를 들어, 장치 내부애플리케이션들(1010)에 의해 액세스되는 데이터를 저장한 SDO들에 대한 액세스는 독립된 계층 트리들에 배치된 ACR들에 의해 제어되어, 그들 사이에 혼선이 벗다. 이런 방식의 제어는 SSA가 상술된 데이터에 대한 액세스를 제어하는 방식과 유사하다. 이것은 콘텐트 소유자들 및 사용자들에게 데이터 객체에 저장된 데이터의 보안을 제공한다.

<446> 도 42를 참조하여, OTP 관련 호스트 애플리케이션에 필요한 소프트웨어 애플리케이션 코드의 부분이 FSE(1102)의 애플리케이션으로서 메모리 장치(10)에 저장되게 하는 것을 가능하게 한다(예를 들어, 메모리 카드 발행 전 사전 저장 또는 저장 후 로딩). 상기 코드를 실행하기 위하여, 호스트는 N 인증 ACR들(1106) 중 하나를 통하여 우선 인증할 필요가 있을 것이고, N은 파이프(1104)에 액세스를 얻기 위한 양의 정수이다. 호스트는 호출하고자 하는 OTP 관련 애플리케이션을 식별하기 위해 애플리케이션 ID를 제공할 필요가 있을 것이다. 성공적인 인증 후, 상기 코드는 OTP 관련 애플리케이션과 연관된 파이프(1104)를 통하여 실행을 위해 액세스될 수 있다. 상기에서 주의된 바와 같이, OTP 관련 내부 애플리케이션 같은 특정 애플리케이션 및 파이프(1104) 사이의 일 대 일 관계가 있다. 도 42에 도시된 바와 같이, 다중 ACR들(1106)은 공통 파이프(1104)의 제어를 공유할 수 있다. ACR은 하나 이상의 파이프를 제어할 수 있다.

<447> 집합적으로 객체들(1114)이라 불리는 안전한 데이터 객체들(SDO 1, SDO 2 및 SDO 3)은 도 42에 도시되고, 각각은 OTP 생성용 씨드 같은 데이터를 포함하고, 상기 씨드는 값있고 바람직하게 인크립트된다. 3개의 데이터 객체들 및 FSE(1102) 사이의 링크들 또는 연관성(1108)은 객체들의 속성을 도시하고, 여기서 객체들 중 임의의 하나가 액세스될 때, SDO의 속성내 애플리케이션 ID를 가진 FSE(1102)의 애플리케이션은 호출될 것이고, 애플리케이션은 임의의 추가 호스트 명령들(도 1)의 수신을 요구하지 않고 메모리 장치의 CPU(12)에 의해 실행될 것이다.

<448> 도 42를 참조하여, 사용자가 OTP처리를 시작하기 위한 위치에 있기 전에, 보안 데이터 구조들(ACR들 1101, 1103, 1106 및 1110)은 OTP 처리를 제어하기 위한 PCT들이 미리 생성된다. 사용자는 인증 서버 ACR들(1106) 중 하나를 통하여 OTP 장치 내부 애플리케이션(1102)을 호출하기 위한 액세스 권리들을 가질 것을 요구할 것이다. 사용자는 N 사용자 ACR들(1110) 중 하나를 통하여 생성될 OTP에 액세스 권리들을 가질 것을 필요로 한다. SDO들(1114)은 OTP 씨드 제공 처리 동안 생성될 수 있다. IDO(1116)는 바람직하게 내부 ACR(1103)에 의해 생성 및 제어된다. 내부 ACR(1103)는 그들이 생성된 후 SDO들(1114)을 제어한다. SDO들(1114)이 액세스될 때, 도 40b의 SSA 관리자(1024)는 ACR(1103)에 자동으로 로그인 한다. 내부 ACR(1103)은 FSE(1102)와 연관된다. SDO들(1114)은 파선들(1108)에 의해 도시된 바와 같이 OTP 씨드 제공 처리 동안 FSE와 연관된다. 연관이 발생 후, SDO들이 호스트에 의해 액세스될 때, 연관성(1108)은 FSE(1102)가 호스트로부터 추가 요구없이 호출되게 할 것이다. 도 40b의 SSA 관리자(1024)는, 통신 파이프(1104)가 N ACR들(1106) 중 하나를 통하여 액세스될 때 ACR(1103)에 자동으로 로그인 한다. 양쪽 경우들(SDO 1114 및 파이프 1104에 액세스)에서, SSA 관리자는 FSE(1102)에 세션 번호를 패스할 것이고, 상기 세션 번호는 내부 ACR(1103)에 대한 채널을 식별할 것이다.

<449> OTP 동작은 두 개의 단계들을 포함한다: 도 43에 도시된 씨드 제공 단계 및 도 44에 도시된 OTP 생성 단계. 도 40 ~ 42에 대한 참조는 설명을 돕고자 이루어진다. 도 43은 씨드 제공 처리를 도시하는 프로토콜 도면이다. 도 43에 도시된 바와 같이, 다양한 작용들은 호스트(24) 같은 호스트 및 카드에 의해 취해진다. 다양한 작용들을 취하는 카드상 하나의 엔티티는 SSM 코어(1004)를 포함하는 도 40a 및 40b의 SSM 시스템이다. 다양한 작용들을 수행하는 카드상 다른 엔티티는 도 42에 도시된 FSE(1102)이다.

<450> 이중 팩터 인증시, 사용자는 발행될 씨드를 요구하고 일단 씨드가 발행되면, 씨드는 보안 토큰에 저장된다. 이 실시예에서, 보안 토큰은 메모리 장치 또는 카드이다. 사용자는 SSM 시스템에 대한 액세스를 얻기 위하여(화살표 1122) 도 42의 인증 ACR들(1106) 중 하나를 인증한다. 인증이 성공적인 것이 가정되면(화살표 1124), 사용자는 씨드를 요구한다(화살표 1126). 호스트는 씨드 요구를 사인하기 위한 특정 애플리케이션(1102)을 선택하여 카드에 씨드 요구를 사인하기 위한 요구를 전송한다. 만약 사용자가 호출될 필요가 있는 특정 애플리케이션 I.D를 인식하지 못하면, 이 정보는 예를 들어 장치에 대한 분별있는 질문을 통하여 장치(10)로부터 얻어질 수 있다. 그 다음 사용자는 호출되어야 하는 애플리케이션의 애플리케이션 ID를 요구하여, 애플리케이션에 대응하는 통신 파이프를 선택한다. 사용자 명령은 대응하는 통신 파이프를 통하여 사용자로부터 애플리케이션 I.D.에 의해 지정된 애플리케이션에 패스 명령에 전송된다(화살표 1128). 호출된 애플리케이션은 도 42의 IDO(1112) 같은 특정 IDO의 공용 키에 의해 시그네이쳐를 요구한다.

<451> SSM 시스템은 IDO의 공용 키를 사용하여 씨드 요구를 사인하고 사인이 완료된 것을 애플리케이션에게 통지한다(화살표 1132). 호출된 애플리케이션은 IDO의 증명서 체인을 요구한다(화살표 1134). 응답시, SSM 시스템은 ACR(1103)에 의해 제어되는 바와 같은 IDO의 증명서 체인을 제공한다(화살표 1136). 호출된 애플리케이션은 호스트에 동일한 것을 전송하는 SSM 시스템에 통신 파이프를 통하여 IDO의 증명서 체인 및 사인된 씨드 요구를 제공한다(화살표 1138). 통신 파이프를 통한 사인된 씨드 요구 및 IDO 증명서 체인의 전송은 도 40a의 SAMM(1088) 및 SSM 코어(1004) 사이에서 설정된 콜백(callback) 기능을 통하여 이루어지고, 여기서 콜백 기능은 하기에 설명될 것이다.

<452> 사인된 씨드 요구 및 호스트에 의해 수신된 IDO 증명서 체인은 도 41에 도시된 인증 서버(1052)에 전송된다. 증명서 체인은 사인된 씨드 요구가 신뢰성 있는 토큰으로부터 발생하는 것이 증명된 카드에 의해 제공되어 인증 서버(1052)는 비밀 씨드를 가진 카드를 제공한다. 인증 서버(1052)는 호스트에 사용자 ACR 정보와 함께 IDO의 공용 키로 인크립된 씨드를 전송한다. 사용자 정보는 사용자가 생성될 OTP에 액세스하기 위한 권리들을 가지는 N 사용자 ACR들 중 어느 하나를 가리킨다. 호스트는 애플리케이션 I.D.를 공급하여 FSE(1102) 내 OTP 애플리케이션을 호출하고, 이에 따라 애플리케이션에 대응하는 통신 파이프를 선택하고 사용자 ACR 정보를 SSM 시스템에 전송한다(화살표 1140). 인크립트된 씨드 및 사용자 ACR 정보는 통신 파이프를 통하여 선택된 애플리케이션에 전송된다(화살표 1142). 호출된 애플리케이션은 IDO의 사적 키를 사용하여 씨드의 디크립션을 위한 SSM 시스템에 하나의 요구를 전송한다(화살표 1144). SSM 시스템은 씨드를 디크립트하고 디크립션이 완료된 애플리케이션에 통지를 전송한다(화살표 1146). 그 다음 호출된 애플리케이션은 보안된 데이터 객체의 생성 및 보안된 데이

터 객체에 씨드의 저장을 요구한다. 또한 SDO가 일회용 패스워드를 생성하기 위한 OTP 애플리케이션(요구를 수행하는 동일한 애플리케이션)의 ID와 연관되는 것을 요구한다(화살표 1148). SSM 시스템은 SDO들(1114) 중 하나를 생성하고 SDO 내부에 씨드를 저장하고 OTP 애플리케이션의 ID와 SDO를 연관시키고, 완료될 때 애플리케이션에 통지를 전송한다(화살표 1150). 그 다음 애플리케이션은 호스트에 의해 공급된 사용자 정보를 바탕으로 적당한 사용자 ACR에 SDO(1114)를 액세스하기 위하여 내부 ACR(1103)에 의해 액세스 권리들을 위임하기 위하여 SSM 시스템에게 요구한다(화살표 1152). 위임이 완료된 후, SSM 시스템은 애플리케이션을 통지한다(화살표 1154). 애플리케이션은 통신 파이프를 통하여 SDO(슬롯 ID)의 이름을 콜 백 기능을 통해 SSM 시스템에 전송한다(화살표 1156). SSM 시스템은 동일한 것을 호스트에 전송한다(화살표 1158). 호스트는 SDO의 이름을 사용자 ACR에 결합하여, 사용자는 SDO에 액세스할 수 있다.

<453> OTP 생성 처리는 도 44의 프로토콜 도면을 참조하여 기술될 것이다. 일회용 패스워드를 얻기 위하여, 사용자는 액세스 권리들을 가진 사용자 ACR에 로그인할 것이다(화살표 1172). 인증이 성공적인 것을 가정하여, SSM 시스템은 호스트에게 통지하고 호스트는 SSM에 "SDO 얻음" 명령을 전송한다(화살표 1174,1176). 상술된 바와 같이, 씨드를 저장하는 SDO는 OTP를 생성하기 위한 애플리케이션과 연관된다. 그러므로 이전 처럼 통신 파이프를 통한 애플리케이션 선택 대신, OTP 생성 애플리케이션은 화살표 1176의 명령에 의해 액세스된 SDO 및 OTP 생성 애플리케이션 사이의 관계에 의해 호출된다(화살표 1178). OTP 생성 애플리케이션은 SDO로부터 콘텐트(즉, 씨드)를 독출하기 위하여 SSM 시스템에 요구한다(화살표 1180). 바람직하게, SSM은 SDO의 콘텐트에 포함된 정보를 인식하지 못하고, FSE에 의해 명령받은 바와 같이 SDO의 데이터를 간단히 처리할 것이다. 만약 씨드가 인크립트되면, 이것은 FSE에 의해 명령된 바와같이 독출 전에 씨드를 디크립트하는 것을 포함할 수 있다. SSM 시스템은 SDO로부터 씨드를 독출하고 OTP 생성 애플리케이션에 씨드를 제공한다(화살표 1182). OTP 생성 애플리케이션은 OTP를 생성하고 이를 SSM 시스템에 제공한다(화살표 11840. OTP는 SSM에 의해 호스트로 전송되고(화살표 1186) 그 다음 OTP를 인증 서버(1052)에 전송하여 이중 팩터 인증 처리를 완료한다.

<454> 콜백 기능

<455> 일반적인 콜백 기능은 도 40a의 SSA 코어(1004) 및 SAMM(1008) 사이에 설정된다. 다른 장치 내부 애플리케이션들 및 통신 파이프들은 상기 기능이 등록될 수 있다. 따라서 장치 내부 애플리케이션이 호출될 때, 애플리케이션은 처리 후 데이터를 애플리케이션에 호스트 명령을 패스하기 위하여 사용된 동일한 통신 파이프를 통하여 SSM 시스템에 패스하기 위한 이런 콜백 기능을 사용할 수 있다.

<456> DRM 시스템 실시예

<457> 도 45는 DRM 기능들을 실행하기 위한 기능들을 제어하기 위하여 FSE 애플리케이션들(1102') 및 제어 구조들(1101',1103',1106')에 대한 링크들(1108')을 가진 통신 파이프(1104'), CEK들(1114')을 사용하는 DRM 시스템을 도시하는 기능 블록도이다. 주의될 바와 같이, 도 45의 아키텍쳐는 보안 데이터가 인증 서버 ACR들 및 사용자 ACR들 대신 라이센스 서버 ACR들(1106') 및 재생 ACR들(1110'), 및 SDO들 대신 CEK들(1114')을 포함하는 것을 제외하고 도 42와 매우 유사하다. 게다가, IDO는 포함되지 않고 따라서 도 45에서 생략된다. CEK들(1114')은 라이센스 제공 처리시 생성될 수 있다. 도 46의 프로토콜 도면은 라이센스 제공 및 콘텐트 다운로드 처리를 도시하고, 여기서 키는 라이센스 객체에 제공된다. OTP 실시예에서 처럼, 라이센스를 획득하고자 하는 사용자는 우선 N ACR들(1106') 중 하나 및 N ACR들(1110') 중 하나 하에서 액세스 권리들을 획득하는 것을 필요로 하여, 콘텐트는 미디어 플레이어 소프트웨어 애플리케이션 같은 미디어 플레이어에 의해 렌더될 수 있다.

<458> 도 46에 도시된 바와 같이, 호스트는 라이센스 서버 ACR(1106')에 인증한다(화살표 1202). 인증이 성공적인 것을 가정하여(화살표 1204), 라이센스 버서는 CEK(키 ID 및 키 값)과 함께 라이센스 파일을 호스트에 제공한다. 호스트는 또한 카드상 SSM 시스템에 애플리케이션 ID를 공급함으로써 호출된 애플리케이션을 선택한다. 호스트는 플레이어 정보를 전송한다(예를 들어, 미디어 플레이어 소프트웨어 애플리케이션 정보)(화살표 1206). 플레이어 정보는 플레이어가 권리들을 액세스하는 N 재생 ACR들(1110') 중 어느 하나를 가리킬 것이다. SSM 시스템은 선택된 애플리케이션에 대응하는 통신 파이프를 통하여 라이센스 파일 및 CEK를 DRM 애플리케이션에 전송한다(화살표 1208). 호출된 애플리케이션은 숨겨진 파티션에 라이센스 파일을 기입하기 위하여 SSM 시스템에게 요구한다(화살표 1210). 라이센스 파일이 기입될 때, SSM 시스템은 애플리케이션에 통지한다(화살표 1212). DRM 애플리케이션은 생성되고 이를 라이센스 파일로부터 키 값에 저장한다. DRM 애플리케이션은 CEK 객체가 제공된 키와 연관된 라이센스를 검사하는 DRM 애플리케이션의 ID와 연관되는 것을 요구한다(화살표 1214). SSM 시스템은 이들 임무들을 완료하고 따라서 애플리케이션에 통지한다(화살표 12160. 그 다음 애플리케이션은 플레이어가 호스트에 의해 전송된 플레이어 정보를 바탕으로 콘텐트에 액세스하기 위한 허용을 가지는 재생 ACR에 위임될

CEK(1114')에 대한 독출 액세스 권리들을 요구한다(화살표 1218). SSM 시스템은 위임을 수행하고 애플리케이션에 통지한다(화살표 1220). 라이센스의 저장이 완료된 메시지는 애플리케이션에 의해 통신 파이프를 통하여 SSM 시스템에 전송되고 SSM 시스템은 이를 라이센스 서버에 전송한다(화살표들 1222 및 1224). 콜백 기능은 통신 파이프를 통하여 이런 작용에 사용된다. 이런 통지를 수신한 후, 라이센스 서버는 카드에 제공된 CEK의 키 값을 인크립트된 콘텐트에 제공한다. 인크립트된 콘텐트는 공용 카드 영역에 호스트에 의해 저장된다. 인크립트된 콘텐트 파일의 저장은 보안 기능들을 포함하지 않기 때문에, SSM 시스템은 저장시 포함되지 않는다.

<459> 재생 동작은 도 47에 도시된다. 사용자는 호스트를 통하여 적당한 재생 ACR(즉, 독출 권리들이 화살표들 1152 및 1154에서 상기 위임된 재생 ACR)을 인증한다(화살표 1242). 인증이 성공적인 것을 가정하여(화살표 1244) 사용자는 키 ID와 연관된 콘텐트를 독출하기 위한 요구를 전송한다(화살표 1246). 요구를 수신한 후, SSM 시스템은 DRM 애플리케이션 ID가 액세스되는 CEK 객체와 연관되고 따라서 식별된 DRM 애플리케이션이 식별되게 하는 것을 발견한다(화살표 1248). DRM 애플리케이션은 SSM 시스템이 키 ID와 연관된 데이터(즉, 라이센스)를 독출하게 한다(화살표 1250). SSM은 독출하기 위하여 요구된 데이터의 정보를 인식하지 못하고, 데이터 독출 처리를 수행하기 위하여 FSE로부터 상기 요구를 간단히 처리한다. SSM 시스템은 숨겨진 구획부으로부터 데이터(즉, 라이센스)를 독출하고 상기 데이터를 DRM 애플리케이션에 제공한다(화살표 1252). DRM 애플리케이션은 데이터를 해석하고 라이센스가 유효한지를 알기 위하여 데이터의 라이센스 정보를 검사한다. 만약 라이센스가 여전히 유효하면, DRM 애플리케이션은 콘텐트 디크립션이 승인되는 것을 SSM 시스템에게 알릴 것이다(화살표 1254). SSM 시스템은 CEK 객체의 키 값을 사용하여 요구된 콘텐트를 디크립트하고 디크립트된 콘텐트를 재생을 위해 호스트에 공급한다(화살표 1256). 만약 라이센스가 더 이상 유효하지 않으면, 콘텐트 액세스에 대한 요구는 거부된다.

<460> 키가 라이센스 서버로부터 라이센스 파일에 제공되지 않은 경우, 라이센스 제공 및 콘텐트 다운로드는 도 46에 도시된 것과 다소 다를 것이다. 상기 다른 방법은 도 48의 프로토콜 도면에 도시된다. 도 46 및 48 사이의 동일한 단계들은 동일한 수들에 의해 식별된다. 따라서 호스트 및 SSM 시스템은 우선 인증을 시작한다(화살표 1202,1204). 라이센스 서버는 호스트에 대한 키 값이 라이센스 파일 및 키 ID를 제공하고, 호스트는 SSM 시스템에 호출하기를 원하는 DRM 애플리케이션의 애플리케이션 ID와 함께 동일한 것을 전송할 것이다. 호스트는 플레이어 정보를 따라 전송한다(화살표 1206'). SSM 시스템은 선택된 애플리케이션에 대응하는 통신 파이프를 통하여 라이센스 파일 및 키 ID를 선택된 DRM 애플리케이션에 전송한다(화살표 1208). DRM 애플리케이션은 라이센스 파일이 숨겨진 파티션에 기입되는 것을 요구한다(화살표 1210). 라이센스 파일이 기입될 때, SSM 시스템은 DRM 애플리케이션에게 통지한다(화살표 1212). DRM 애플리케이션은 SSM 시스템이 키 값을 생성하고, CEK 객체를 생성하고, 그 내부에 키 값을 저장하고 CEK 객체와 DRM 애플리케이션의 ID를 연관시키는 것을 요구한다(화살표 1214'). 상기 요구가 승인된 후, SSM 시스템은 통지를 DRM 애플리케이션에 전송한다(화살표 1216). DRM 애플리케이션은 호스트로부터 플레이어 정보를 바탕으로 재생 ACR에 CEK 객체에 대한 독출 액세스 권리들을 위임하기 위하여 SSM 시스템에게 요구한다(화살표 1218). 이것이 완료될 대, SSM 시스템은 DRM 애플리케이션에게 통지한다(화살표 1220). DRM 애플리케이션은 라이센스가 저장된 것을 SSM 시스템에게 통지하고, 여기서 통지는 콜백 기능에 의해 통신 파이프를 통하여 전송된다(화살표 1222). 이런 통지는 SSM 시스템에 의해 라이센스 서버에 전송된다(화살표 1224). 라이센스 서버는 SSM 시스템에 키 ID와 연관된 콘텐트 파일을 전송한다(화살표 1226). SSM 시스템은 임의의 애플리케이션들을 포함하지 않고 키 ID에 의해 식별된 키 값을 가진 콘텐트 프로파일을 인크립트한다. 카드상에 인크립트되고 저장된 콘텐트는 도 47의 프로토콜을 사용하여 재생될 수 있다.

<461> OTP 및 DRM 실시예들에서, FSE(1102 및 1102')는 호스트 장치들에 의한 선택을 위해 많은 다른 OTP 및 DRM 애플리케이션들을 포함할 수 있다. 사용자들은 목표된 장치 내부 애플리케이션을 선택 및 호출하는 선택을 가진다. 그럼에도 불구하고, SSM 모듈 및 FSE 사이의 전체 관계는 동일하게 유지되어, 사용자들 및 데이터 제공자들은 SSM 모듈과 상호작용하고 FSE를 호출하기 위한 표준 프로토콜들의 세트를 사용할 수 있다. 사용자들 및 제공자들은 많은 다른 장치 내부 애플리케이션들의 특성들에 포함되지 않아야 하고, 그 중 몇몇은 소유권이다.

<462> 게다가, 제공한 프로토콜들은 도 46 및 48의 경우에서 처럼 다소 다를 수 있다. 라이센스 객체는 도 46의 경우의 키 값을 포함하지만, 도 48의 경우의 키 값을 포함하지 않는다. 이 차이는 상술된 바와 같이 약간 다른 프로토콜들을 호출한다. 그러나, 도 47의 재생은 라이센스가 제공되는 방법과 무관하게 동일하다. 따라서, 이런 차이는 콘텐트 제공자들 및 분배자들에게 문제이지만, 통상적으로 재생 단계에만 포함된 소비자들에게 문제가 아니다. 따라서 이런 아키텍처는 소비자들이 용이하게 사용하면서, 프로토콜들을 맞추기 위해 콘텐트 제공자들 및 분배자들에 큰 융통성을 제공한다. 명백히 둘 이상의 프로토콜 제공 세트에 의해 제공된 데이터로부터 유도된 정보는 여전히 제 2 프로토콜을 사용하여 액세스할 수 있다.

<463> 상기 환경들에 의해 제공된 다른 장점은 사용자들 같은 외측 엔티티들 및 장치 내부 애플리케이션들이 보안 데

이터 구조에 의해 제어되는 데이터의 사용을 공유할 수 있고, 사용자가 저장 데이터로부터 장치 내부 애플리케이션들에 의해 유도된 결과들에만 액세스할 수 있다는 것이다. 따라서, OTP 실시예에서, 호스트 장치들을 통한 사용자는 씨드 값이 아닌 OTP만을 얻을 수 있다. DRM 실시예에서, 호스트 장치들을 통한 사용자는 렌더된 콘텐트만을 얻을 수 있고, 라이센스 파일 또는 암호화 키에 액세스할 수 없다. 이런 특성은 보안을 타협하지 않고 소비자들에게 편리성을 허용한다.

<464> 하나의 DRM 실시예에서, 장치 내부 애플리케이션들과 호스트들 모두는 암호 키들에 액세스할 수 없다; 단지 보안 데이터 구조는 액세스를 가진다. 다른 실시예들에서, 보안 데이터 구조와 다른 엔티티들은 암호화 키들에 액세스할 수 있다. 키들은 장치 내부 애플리케이션들에 의해 생성되고, 그 다음 보안 데이터 구조에 의해 제어된다.

<465> 장치 내부 애플리케이션들 및 정보(예를 들어, OTP 및 렌더된 콘텐트)에 대한 액세스는 동일한 보안 데이터 구조에 의해 제어된다. 이것은 제어 시스템들의 복잡성 및 비용들을 감소시킨다.

<466> 장치 내부 애플리케이션들에 대한 액세스를 제공하는 내부 ACR로부터 호스트들에 의한 장치 내부 애플리케이션들을 호출하는 것으로부터 얻어진 정보에 대한 액세스를 제어하는 ACR로 액세스 권리들을 위임하기 위한 능력을 제공함으로써, 이런 특성은 상기 특성들 및 기능들을 달성할 수 있게 한다.

<467> 애플리케이션 지정 취소 방법

<468> 보안 데이터 구조의 액세스 제어 프로토콜은 장치 내부 애플리케이션이 호출될 때 변형될 수 있다. 예를 들어, 증명서 취소 프로토콜은 CRL을 사용한 표준, 또는 소유권 프로토콜일 수 있다. 따라서, FSE를 호출함으로써, 표준 CRL 취소 프로토콜은 FSE 소유권 프로토콜에 의해 대체될 수 있다.

<469> CRL 취소 방법을 지원하는 것 외에, SSA는 장치 내부 애플리케이션 및 CA 또는 임의의 다른 취소 인증국 사이의 사적 통신 채널을 통하여 호스트들을 취소하기 위하여 장치에 존재하는 특정 내부 애플리케이션을 실행할 수 있다. 내부 애플리케이션 소유권 취소 방법은 호스트 애플리케이션의 관계에 결합된다.

<470> 애플리케이션 지정 취소 방법이 구성될 때, SSA 시스템은 CRL(만약 제공되면)을 거절할 것이고 ELSE는 주어진 증명서가 취소될지 여부를 결정하기 위해 증명서 및 소유권 애플리케이션 데이터(애플리케이션 지정 통신 파이프를 통해 이전에 제공됨)를 사용할 것이다.

<471> 상술된 바와 같이, ACR은 3개의 취소 방법들 중 어느 것(취소 방법 없음, 표준 CRL 방법, 및 애플리케이션 지정 취소 방법)이 취소 값을 지정하여 적용되는가를 지정한다. 애플리케이션 지정 취소 방법 옵션이 선택될 때, ACR은 취소 방법의 변화시 내부 애플리케이션 ID를 위한 ID를 지정할 것이고, CET/APP_ID 필드의 값은 취소 방법의 변화시 내부 애플리케이션 ID에 대응할 것이다. 장치에 인증할 때, SSA 시스템은 내부 애플리케이션의 소유권 방법을 고수할 것이다.

<472> 프로토콜 중 하나의 세트를 다른 것으로 교체하는 대신, 장치 내부 애플리케이션의 호출은 SSA에 의해 이미 가해진 액세스 제어에 부가적인 액세스 조건들을 부과할 수 있다. 예를 들어, CEK의 키 값을 액세스하기 위한 권리는 추가로 FSE에 의해 구성될 수 있다. ACR이 키 값에 대한 액세스 권리들을 가지는 것을 SSA 시스템이 결정한 후, FSE는 액세스가 승인되기 전 참고될 것이다. 이런 특성은 콘텐트에 액세스를 제어하기 위한 큰 융통성을 콘텐트 소유자에게 허용한다.

<473> 본 발명이 다양한 실시예들을 참조하여 상기되었지만, 변화들 및 변형들이 첨부된 청구항들 및 그의 등가물에 의해서만 정의된 본 발명의 범위에서 벗어나지 않고 이루어질 수 있다는 것이 이해될 것이다.

산업상 이용 가능성

<474> 상술한 바와 같이, 본 발명은, 일반적으로 메모리 시스템, 특히 다기능 콘텐트 제어 특징을 갖는 메모리 시스템을 제공하는데 사용된다.

도면의 간단한 설명

<19> 도 1은 본 발명을 도시하는데 유용한 호스트 장치와 통신하는 메모리 시스템의 블록도이다.

<20> 도 2는 특정 분할부들 및 인크립트된 파일들에 대한 액세스가 본 발명의 다른 실시예들을 도시하는데 유용한 액세스 정책들 및 인증 과정들에 의해 제어되는 경우 메모리의 다른 분할부들 및 다른 분할부들에 저장된 언인크

립트 및 인크립트된 파일들의 개략도이다.

<21> 도 3은 메모리의 다른 분할부들을 도시하는 메모리의 개략도이다.

<22> 도 4는 본 발명의 다른 실시예들을 도시하는데 유용한 분할들의 파일들 중 몇몇이 인크립트된 도 3에 도시된 메모리의 다른 분할부들에 대한 파일 위치 테이블들의 개략도이다.

<23> 도 5는 본 발명의 다른 실시예들을 도시하는데 유용한 액세스 제어 레코드 그룹 및 연관된 키 레퍼런스내 액세스 제어 레코드들의 개략도이다.

<24> 도 6은 본 발명의 다른 실시예들을 도시하는데 유용한 액세스 제어 레코드 그룹들 및 액세스 제어 레코드들에 의해 형성된 트리 구조들의 개략도이다.

<25> 도 7은 상기 트리들의 형성 처리를 도시하기 위하여 액세스 제어 레코드 그룹들의 이들 계층 트리를 도시하는 트리의 개략도이다.

<26> 도 8a 및 8b는 시스템 액세스 제어 레코드를 생성 및 사용하기 위하여 메모리 카드 같은 메모리 장치 및 호스트 장치에 의해 수행되는 처리들을 도시하는 흐름도이다.

<27> 도 9는 다른 실시예들을 도시하는데 유용한 액세스 제어 레코드 그룹을 생성하기 위하여 시스템 액세스 제어 레코드를 사용한 처리를 도시하는 흐름도이다.

<28> 도 10은 액세스 제어 레코드를 생성하기 위한 처리를 도시하는 흐름도이다.

<29> 도 11은 계층 트리의 특정 애플리케이션을 도시하는데 유용한 두 개의 액세스 제어 레코드 그룹들의 개략도이다.

<30> 도 12는 특정 권리들의 위임을 위한 처리를 도시하는 흐름도이다.

<31> 도 13은 도 12의 위임 처리를 도시하기 위한 액세스 제어 레코드 그룹 및 액세스 제어 레코드의 개략도이다.

<32> 도 14는 인크립션 및/또는 디크립션을 위한 키를 생성하기 위한 처리를 도시하는 흐름도이다.

<33> 도 15는 액세스 제어 레코드에 따라 데이터 액세스를 위한 액세스 권리들 및/또는 허가를 제거하기 위한 처리를 도시하는 흐름도이다.

<34> 도 16은 액세스 권리들 및/또는 액세스에 대한 허가가 삭제되었거나 만료될 때 액세스를 요구하기 위한 처리를 도시하는 흐름도이다.

<35> 도 17a 및 17b는 본 발명의 다른 실시예들을 도시하는데 유용한 암호화 키들에 대한 액세스를 승인하기 위한 정책들 및 인증을 위한 룰 구조의 조직을 도시하는 개략도이다.

<36> 도 18은 정책들에 따라 보호된 정보에 대한 액세스를 제어하기 위한 다른 방법을 도시하는 데이터베이스 구조의 블록도이다.

<37> 도 19는 패스워드들을 사용하는 인증 처리를 도시하는 흐름도이다.

<38> 도 20은 다수의 인증 증명서 체인들을 도시하는 도면이다.

<39> 도 21은 다수의 장치 증명서 체인들을 도시하는 도면이다.

<40> 도 22 및 23은 일방향 및 상호 인증 방법들에 대한 처리를 도시하는 프로토콜 도면들이다.

<41> 도 24는 본 발명의 일 실시예를 도시하는 인증서 체인의 도면이다.

<42> 도 25는 인증서가 본 발명의 다른 실시예를 도시하기 위하여 인증서 체인의 최종 인증인 표시를 나타내는 최종 인증서를 메모리 장치에 전송하기 위하여 호스트에 의해 전송된 인증서 버퍼 앞에 있는 제어 섹터내 정보를 도시하는 테이블이다.

<43> 도 26 및 27은 메모리 카드가 호스트 장치를 인증하는 인증 방법들에 대해 각각 카드 및 호스트 처리들을 도시하는 흐름도들이다.

<44> 도 28 및 29는 호스트 장치가 메모리 카드를 인증하는 인증 방법들에 대해 각각 카드 및 호스트 처리들을 도시하는 흐름도이다.

<45> 도 30 및 31은 메모리 장치에 저장된 증명서 취소 리스트가 본 발명의 하나 이상의 실시예를 도시하기 위하여 호스트 장치에 의해 검색되는 각각 호스트 장치 및 메모리 장치에 의해 수행되는 처리들을 도시하는 흐름도이다.

<46> 도 32는 본 발명의 다른 실시예를 도시하기 위하여 리스트의 필드들을 도시하는 증명서 취소 리스트의 도면이다.

<47> 도 33 및 34는 증명서 취소 리스트들을 사용하여 증명서들을 검증하기 위한 카드 및 호스트 처리들을 각각 도시하는 흐름도들이다.

<48> 도 35는 호스트에 전송된 데이터를 카드 사이닝(signing)하고 호스트로부터 데이터를 디크립트하기 위한 카드 처리들을 도시하는 흐름도이다.

<49> 도 36은 카드가 호스트에 전송된 데이터를 사인하는 호스트 처리들을 도시하는 흐름도이다.

<50> 도 37은 호스트가 인크립트된 데이터를 메모리 카드에 전송하는 호스트 처리들을 도시하는 흐름도이다.

<51> 도 38 및 39는 일반적인 정보 및 분별있는 정보질문들에 대한 처리들을 각각 도시하는 흐름도이다.

<52> 도 40a는 본 발명의 실시예를 도시하기 위하여 호스트 장치에 접속된 메모리 장치(플래시 메모리 카드 같은)에서 시스템 아키텍처의 기능 블록도이다.

<53> 도 40b는 도 40a의 SSM 코어의 내부 소프트웨어 모듈들의 기능 블록도이다.

<54> 도 41은 일회용 패스워드를 생성하기 위한 시스템의 블록도이다.

<55> 도 42는 일회용 패스워드(OTP) 시드(seed) 제공 및 OTP 생성을 도시하는 기능 블록도이다.

<56> 도 43은 시드 제공 단계를 도시하는 프로토콜 도면이다.

<57> 도 44는 일회용 패스워드 생성 단계를 도시하는 프로토콜 도면이다.

<58> 도 45는 DRM 시스템을 도시하는 기능 블록도이다.

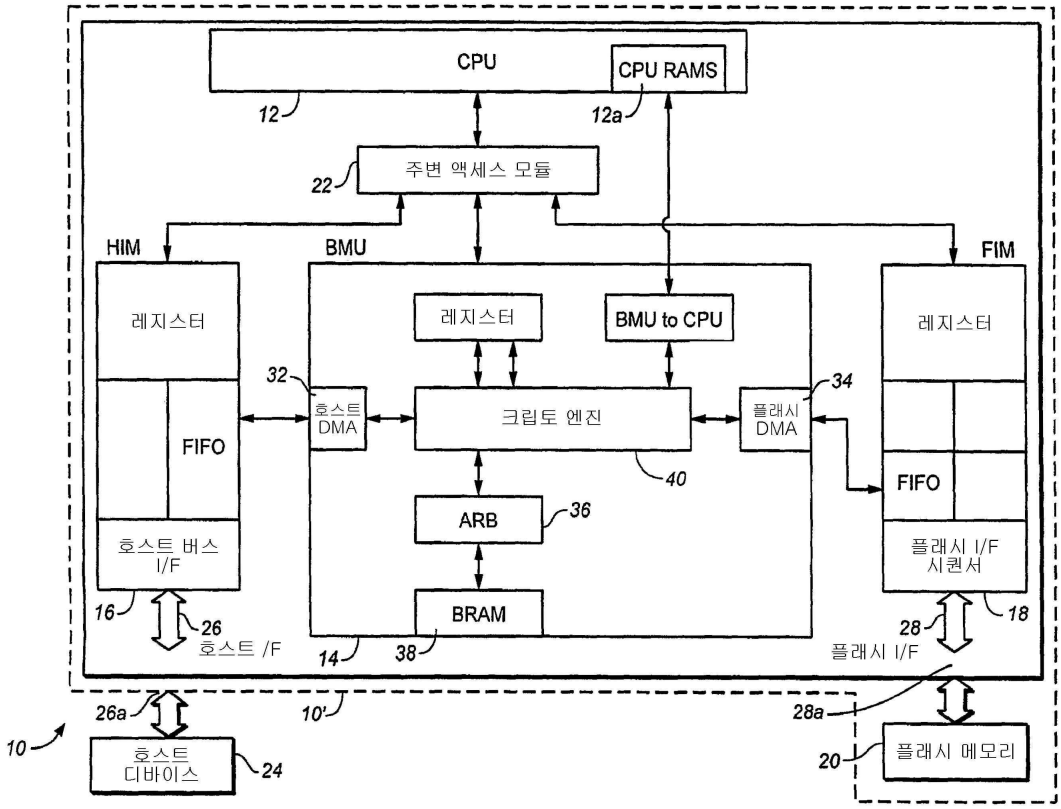
<59> 도 46은 키가 라이센스 객체에 제공되는 라이센스 제공 및 콘텐트 다운로드를 위한 처리를 도시하는 프로토콜 도면이다.

<60> 도 47은 재생 동작 동안 처리를 도시하는 프로토콜 도면이다.

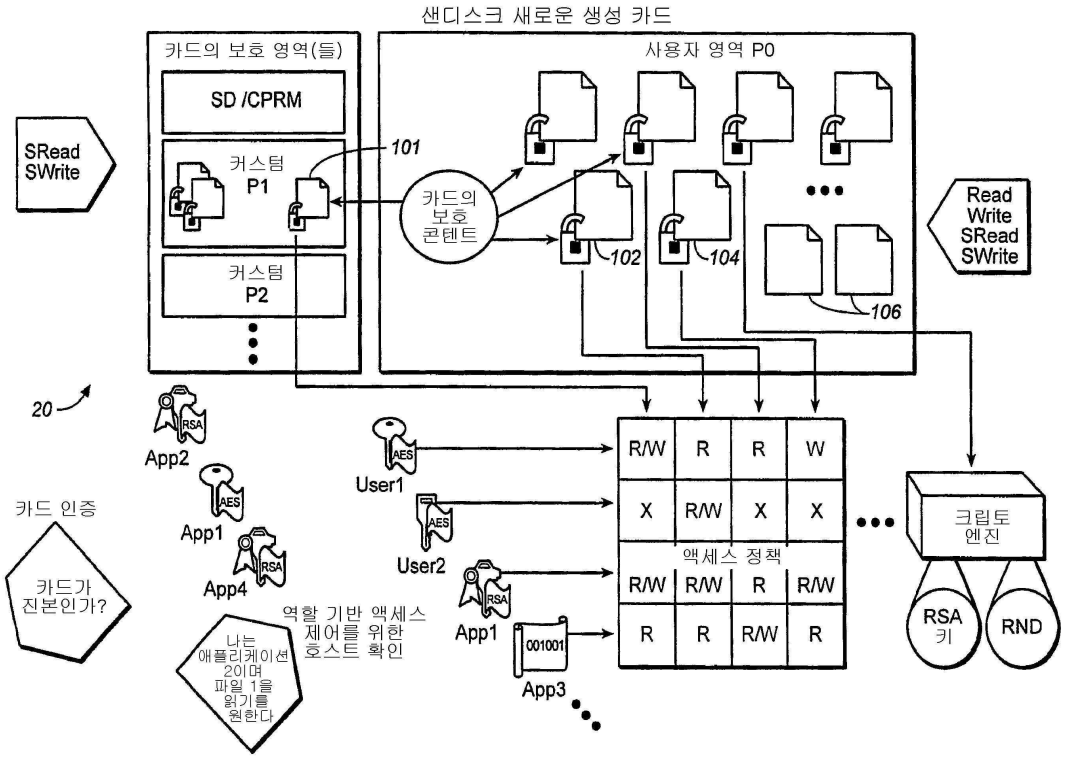
<61> 도 48은 키가 라이센스 객체에 제공되지 않은 경우 라이센스 제공 및 콘텐트 다운로드를 위한 처리를 도시하는 프로토콜 도면이다.

도면

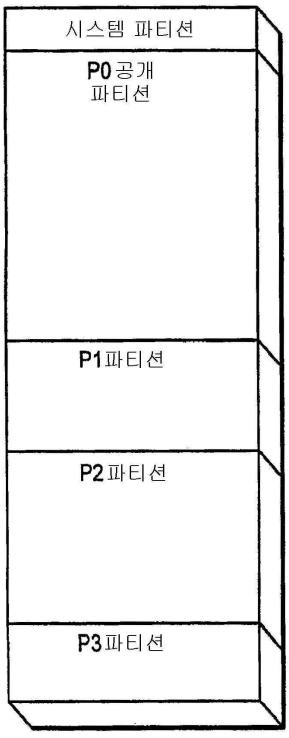
도면1


CPU
CPU RAMS
12
12a
22
주변 액세스 모듈
HIM
BMU
FIM
레지스터
레지스터
BMU to CPU
레지스터
32
34
호스트 DMA
크립토 엔진
플래시 DMA
FIFO
40
FIFO
ARB
36
호스트 버스 I/F
플래시 I/F 시퀀서
16
26
호스트 /F
BRAM
14
38
28
18
플래시 I/F
10
10'
26a
28a
호스트 디바이스
24
20
플래시 메모리

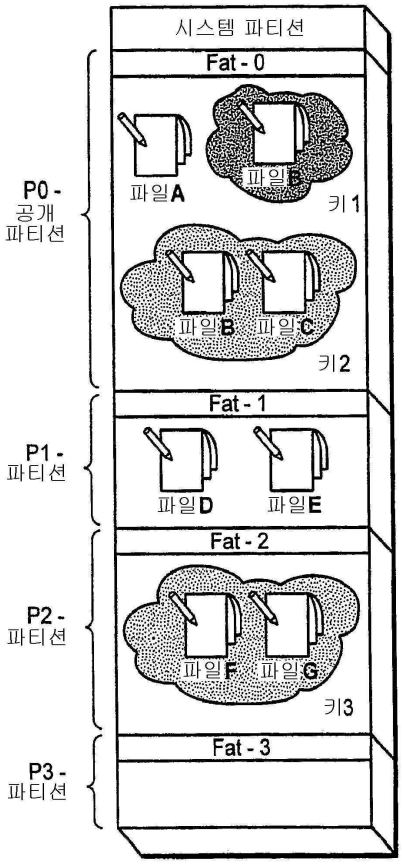
도면2


샌디스크 새로운 생성 카드
카드의 보호 영역(들)
SD /CPRM
커스텀 P1
커스텀 P2
사용자 영역 P0
SRead SWrite
101
카드의 보호 콘텐트
102
104
106
Read Write SRead SWrite
20
App2
App1
App4
User1
User2
App1
App3
001001
R/W R R W
X R/W X X
액세스 정책
R/W R/W R R/W
R R R/W R
크립토 엔진
RSA 키
RND
카드 인증
카드가 진본인가?
역할 기반 액세스 제어를 위한 호스트 확인
나는 애플리케이션 20이며 파일 1을 읽기를 원한다

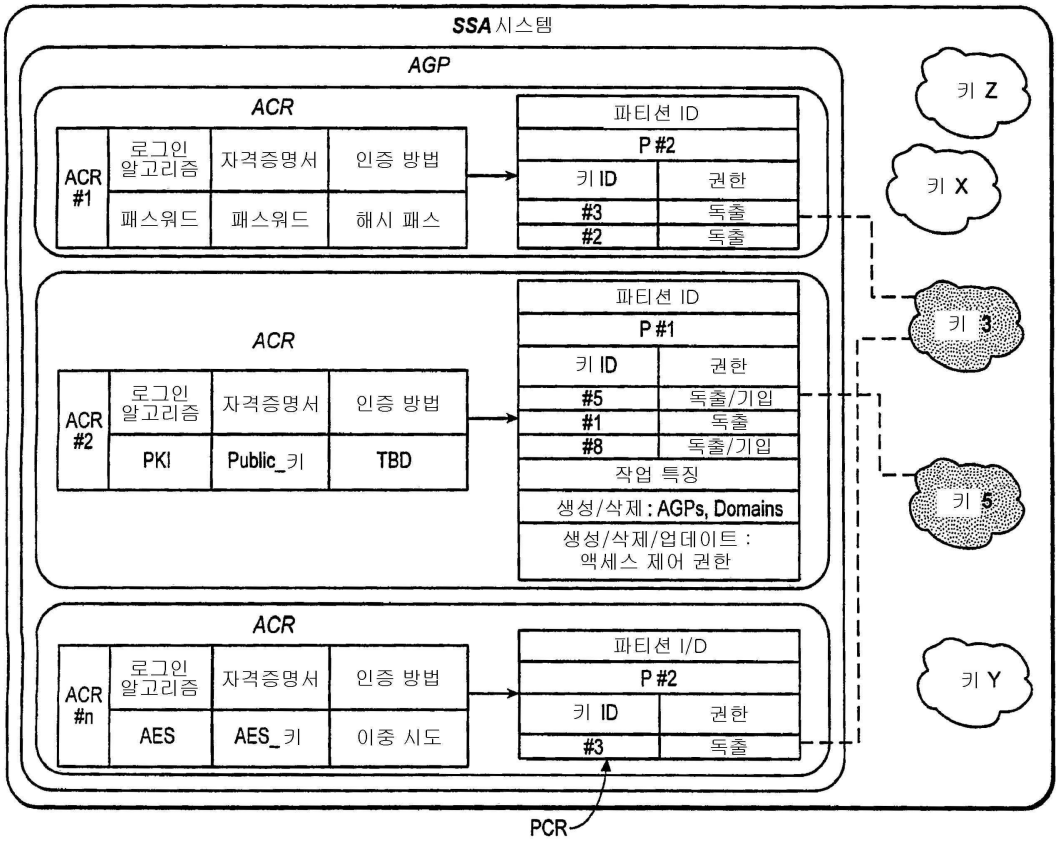
도면3


시스템 파티션
P0 공개 파티션
P1 파티션
P2 파티션
P3 파티션

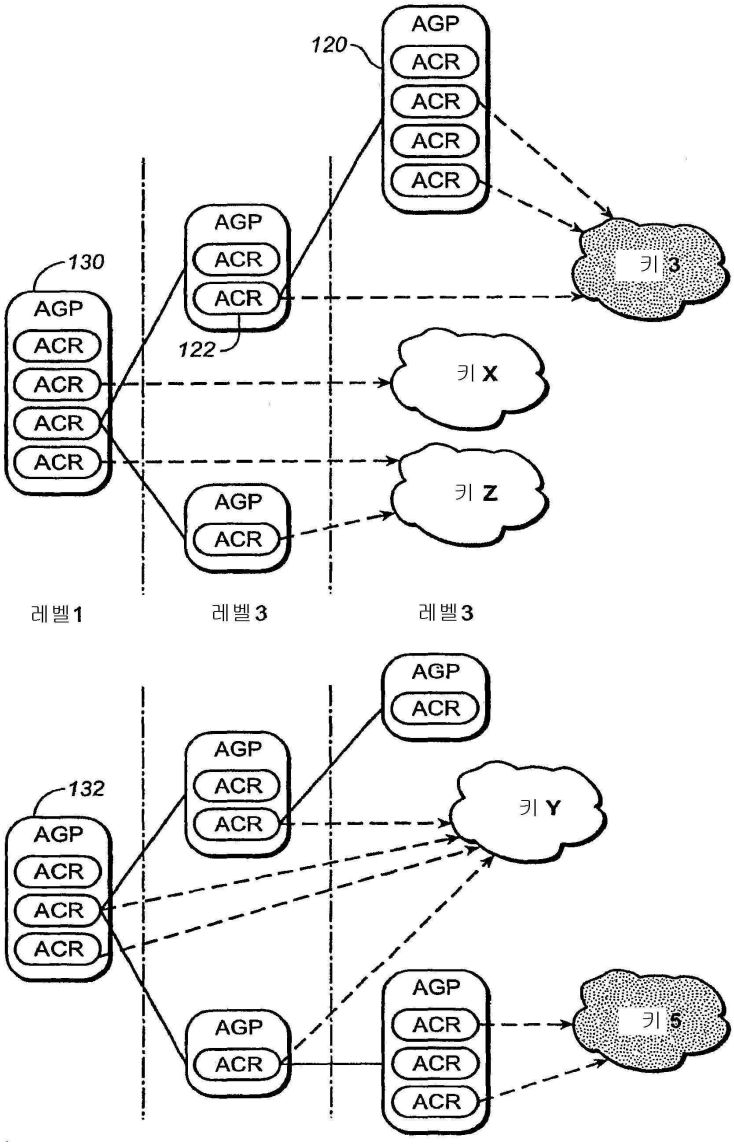
도면4


시스템 파티션
Fat - 0
파일A
파일B
키1
파일B
파일C
키2
Fat - 1
파일D
파일E
Fat - 2
파일F
파일G
키3
Fat - 3
P0 - 공개 파티션
P1 - 파티션
P2 - 파티션
P3 - 파티션

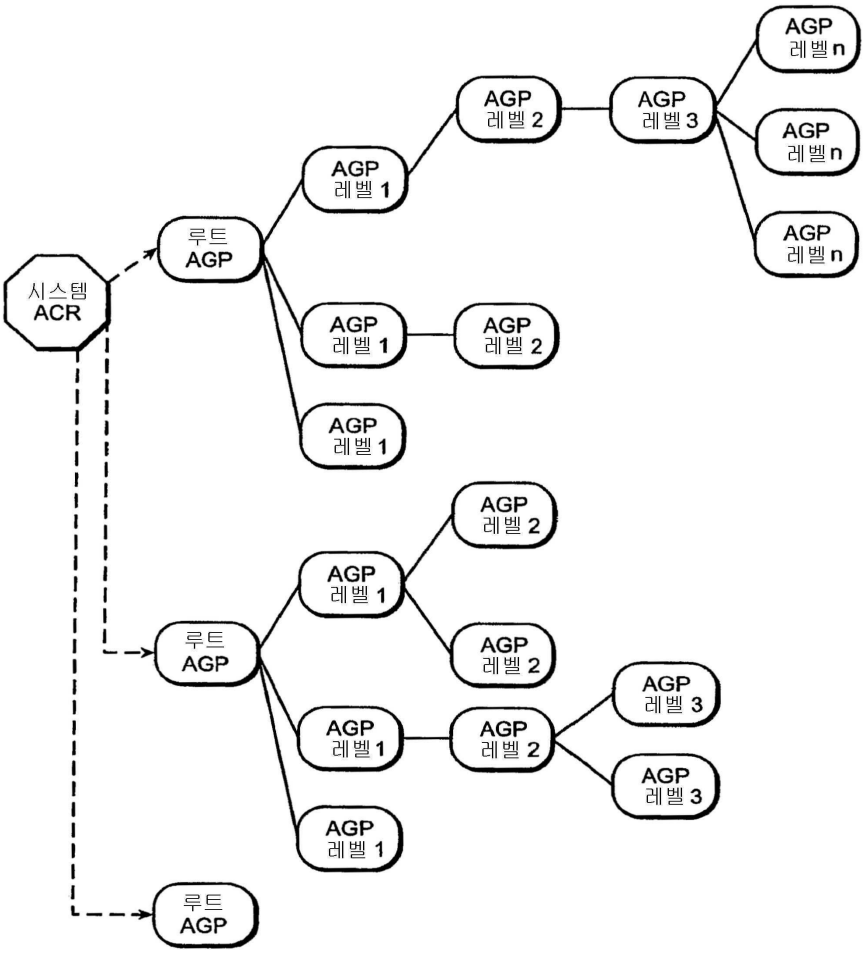
도면5


SSA 시스템
AGP
ACR
ACR #1
로그인 알고리즘
자격증명서
인증 방법
패스워드
패스워드
해시 패스
파티션 ID
P #2
키 ID
권한
#3
독출
#2
독출
ACR
ACR #2
로그인 알고리즘
자격증명서
인증 방법
PKI
Public_키
TBD
파티션 ID
P #1
키 ID
권한
#5
독출/기입
#1
독출
#8
독출/기입
작업 특징
생성/삭제 : AGPs, Domains
생성/삭제/업데이트 : 액세스 제어 권한
ACR
ACR #n
로그인 알고리즘
자격증명서
인증 방법
AES
AES_키
이중 시도
파티션 I/D
P #2
키 ID
권한
#3
독출
PCR
키 Z
키 X
키 3
키 5
키 Y

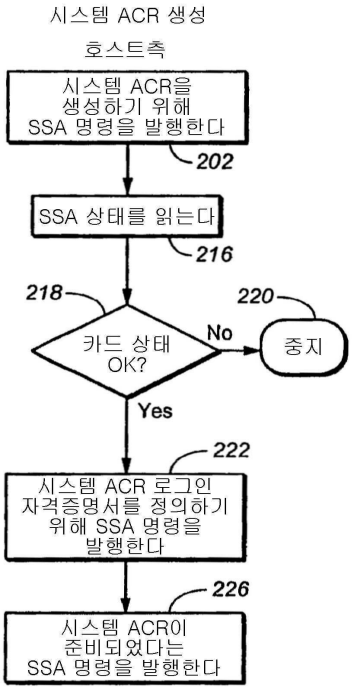
도면6


120
AGP
ACR
ACR
ACR
ACR
130
AGP
ACR
ACR
ACR
ACR
AGP
ACR
ACR
122
AGP
ACR
키 3
키 X
키 Z
레벨1
레벨3
레벨3
132
AGP
ACR
ACR
ACR
AGP
ACR
ACR
AGP
ACR
키 Y
AGP
ACR
ACR
ACR
키 5

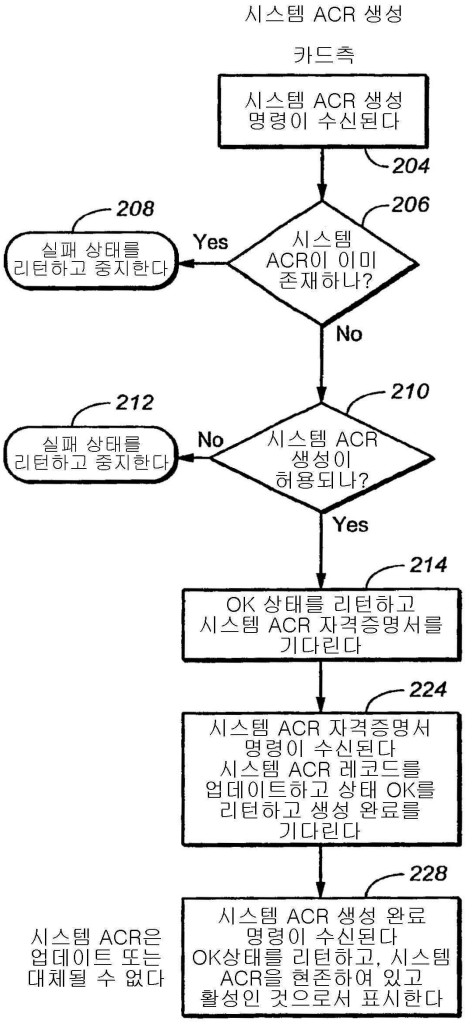
도면7


시스템
ACR
루트
AGP
AGP
레벨 1
AGP
레벨 2
AGP
레벨 3
AGP
레벨 n
AGP
레벨 n
AGP
레벨 n
AGP
레벨 1
AGP
레벨 2
AGP
레벨 1
AGP
레벨 2
루트
AGP
AGP
레벨 1
AGP
레벨 2
AGP
레벨 2
AGP
레벨 1
AGP
레벨 2
AGP
레벨 3
AGP
레벨 3
AGP
레벨 1
루트
AGP

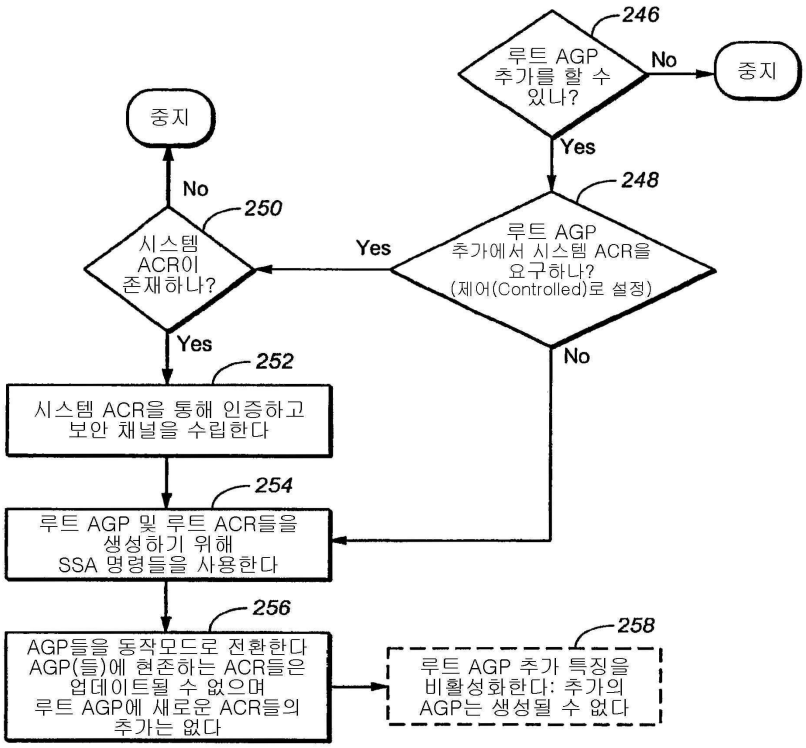
도면8a


시스템 ACR 생성
호스트측
시스템 ACR을
생성하기 위해
SSA 명령을 발행한다
202
SSA 상태를 읽는다
216
218
220
카드 상태
OK?
No
중지
Yes
222
시스템 ACR 로그인
자격증명서를 정의하기
위해 SSA 명령을
발행한다
226
시스템 ACR이
준비되었다는
SSA 명령을 발행한다

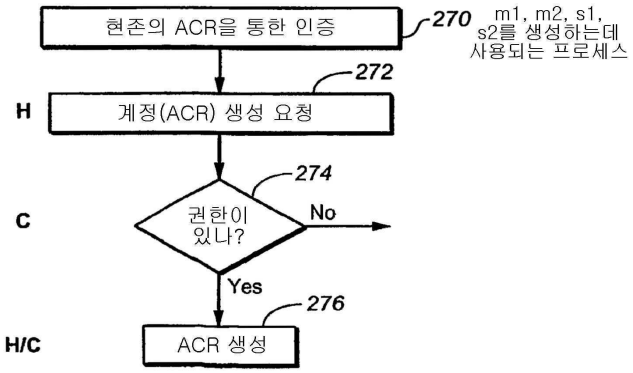
도면8b


시스템 ACR 생성
카드측
시스템 ACR 생성 명령이 수신된다
204
206
시스템 ACR이 이미 존재하나?
Yes
208
실패 상태를 리턴하고 중지한다
No
210
시스템 ACR 생성이 허용되나?
No
212
실패 상태를 리턴하고 중지한다
Yes
214
OK 상태를 리턴하고 시스템 ACR 자격증명서를 기다린다
224
시스템 ACR 자격증명서 명령이 수신된다 시스템 ACR 레코드를 업데이트하고 상태 OK를 리턴하고 생성 완료를 기다린다
228
시스템 ACR 생성 완료 명령이 수신된다 OK상태를 리턴하고, 시스템 ACR을 현존하여 있고 활성인 것으로서 표시한다
시스템 ACR은 업데이트 또는 대체될 수 없다

도면9

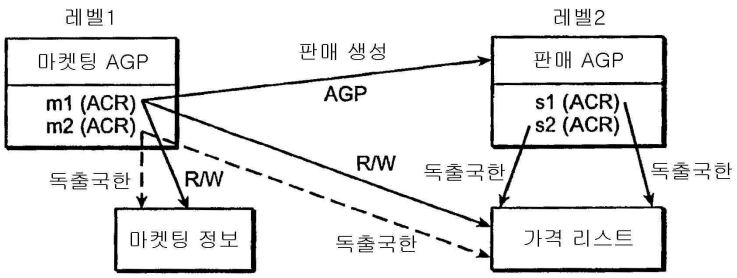


도면10

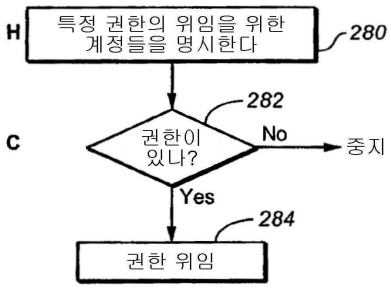


도면11

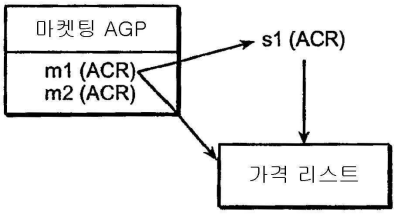
마켓팅 AGP에서 2개의 ACR들(m1,m2)과, 판매 AGP에서 2개의 ACR들(s1,s2)을 생성



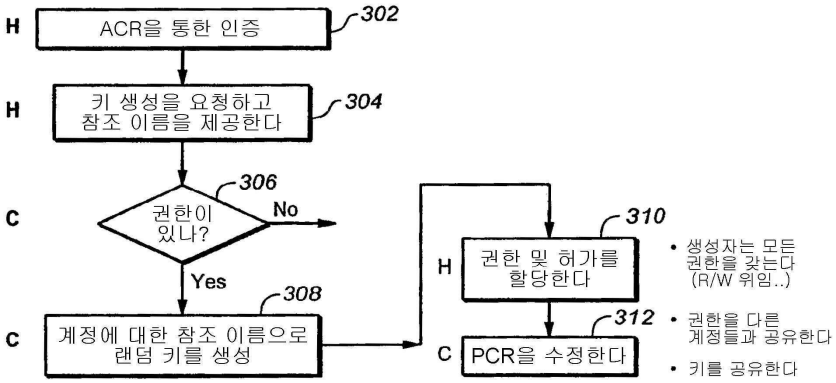
도면12



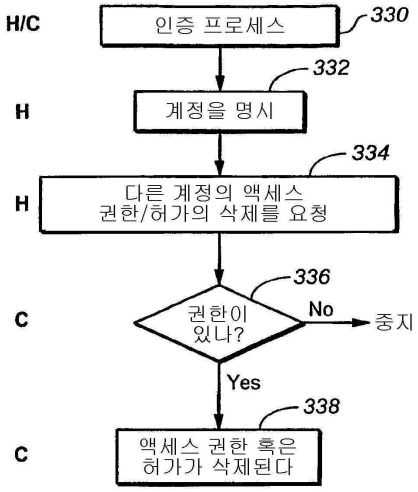
도면13



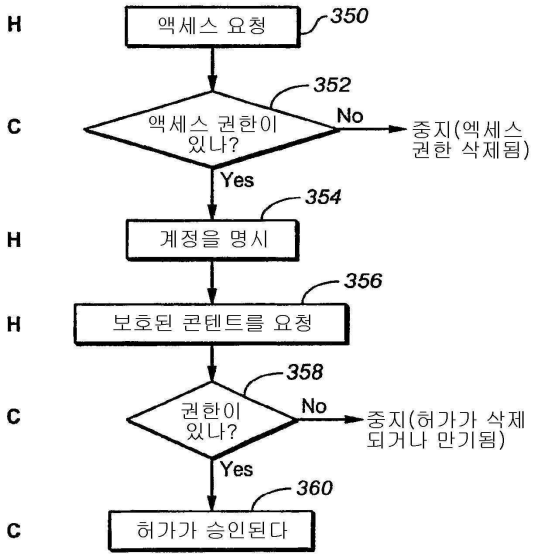
도면14



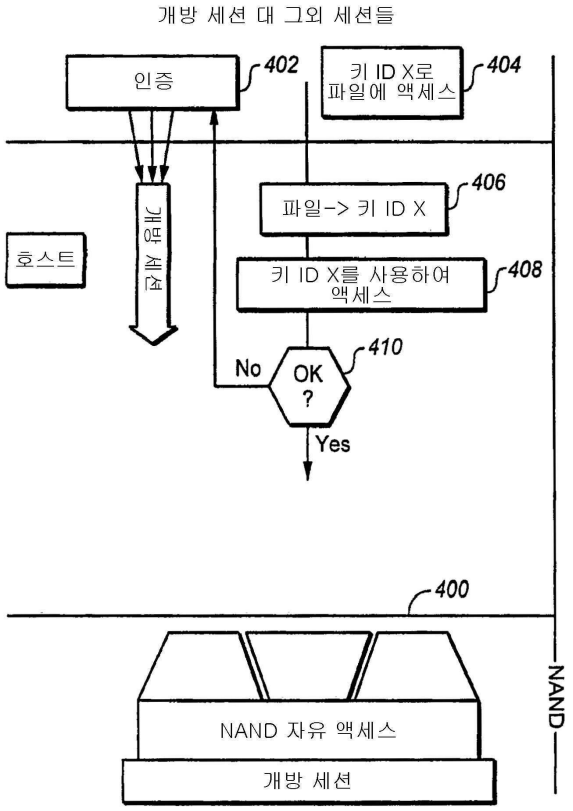
도면15


H/C
인증 프로세스
330
H
계정을 명시
332
H
다른 계정의 액세스 권한/허가의 삭제를 요청
334
C
권한이 있나?
336
No
중지
Yes
C
액세스 권한 혹은 허가가 삭제된다
338

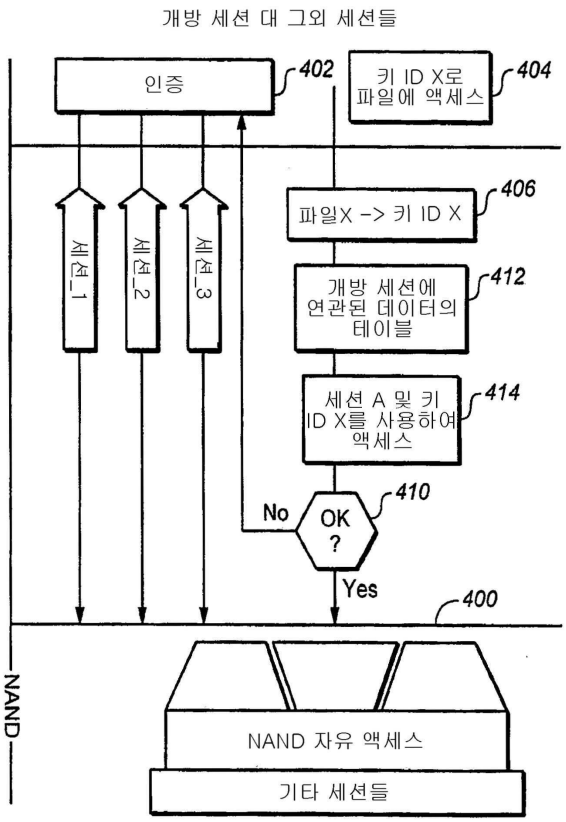
도면16


H
액세스 요청
350
C
액세스 권한이 있나?
352
No
중지(액세스 권한 삭제됨)
Yes
H
계정을 명시
354
H
보호된 콘텐트를 요청
356
C
권한이 있나?
358
No
중지(허가가 삭제되거나 만기됨)
Yes
C
허가가 승인된다
360

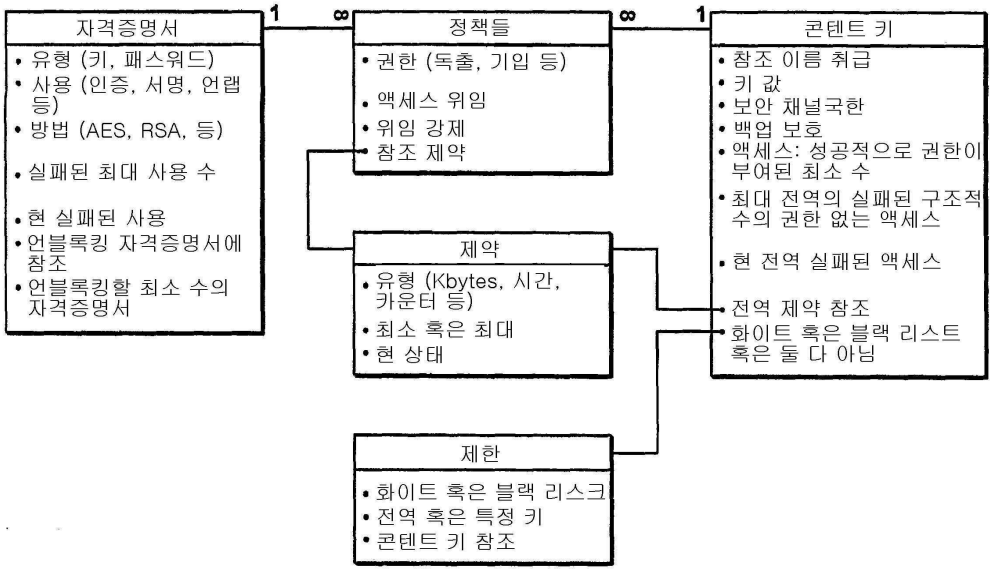
도면17a


개방 세션 대 그외 세션들
인증
402
키 ID X로 파일에 액세스
404
파일-> 키 ID X
406
호스트
개방 세션
키 ID X를 사용하여 액세스
408
No
OK ?
410
Yes
400
NAND 자유 액세스
개방 세션
NAND

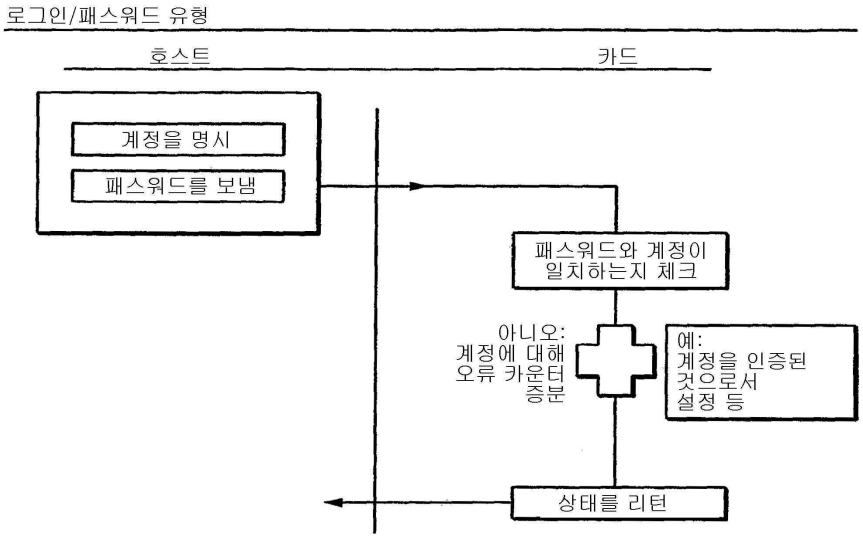
도면17b


개방 세션 대 그외 세션들
인증
402
키 ID X로 파일에 액세스
404
파일X -> 키 ID X
406
세션_1
세션_2
세션_3
개방 세션에 연관된 데이터의 테이블
412
세션 A 및 키 ID X를 사용하여 액세스
414
No
OK ?
410
Yes
400
NAND 자유 액세스
기타 세션들
NAND

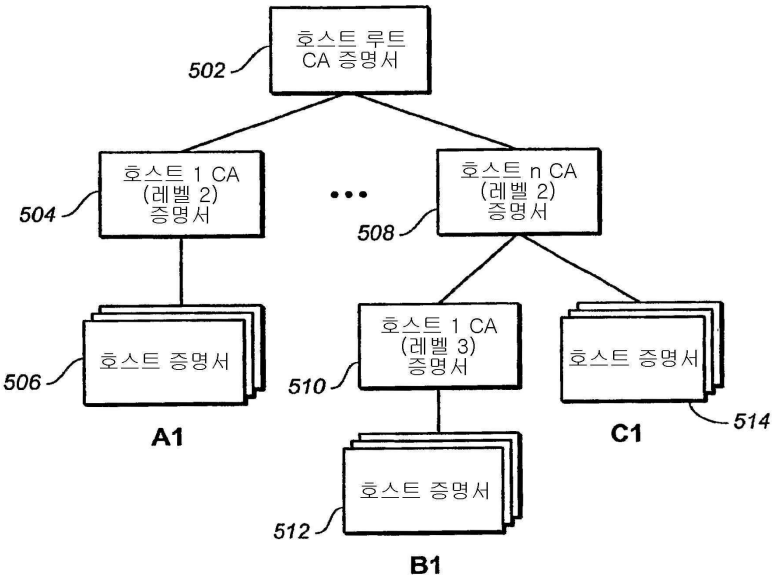
도면18


자격증명서
• 유형 (키, 패스워드)
• 사용 (인증, 서명, 언랩 등)
• 방법 (AES, RSA, 등)
• 실패된 최대 사용 수
• 현 실패된 사용
• 언블록킹 자격증명서에 참조
• 언블록킹할 최소 수의 자격증명서
1
∞
정책들
• 권한 (독출, 기입 등)
• 액세스 위임
• 위임 강제
• 참조 제약
∞
1
콘텐트 키
• 참조 이름 취급
• 키 값
• 보안 채널국한
• 백업 보호
• 액세스: 성공적으로 권한이 부여된 최소 수
• 최대 전역의 실패된 구조적 수의 권한 없는 액세스
• 현 전역 실패된 액세스
제약
• 유형 (Kbytes, 시간, 카운터 등)
• 최소 혹은 최대
• 현 상태
• 전역 제약 참조
• 화이트 혹은 블랙 리스트 혹은 둘 다 아님
제한
• 화이트 혹은 블랙 리스크
• 전역 혹은 특정 키
• 콘텐트 키 참조

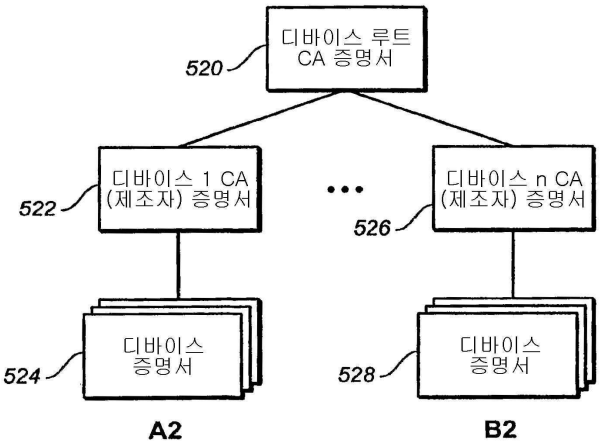
도면19


로그인/패스워드 유형
호스트
카드
계정을 명시
패스워드를 보냄
패스워드와 계정이 일치하는지 체크
아니오:
계정에 대해
오류 카운터
증분
예:
계정을 인증된
것으로서
설정 등
상태를 리턴

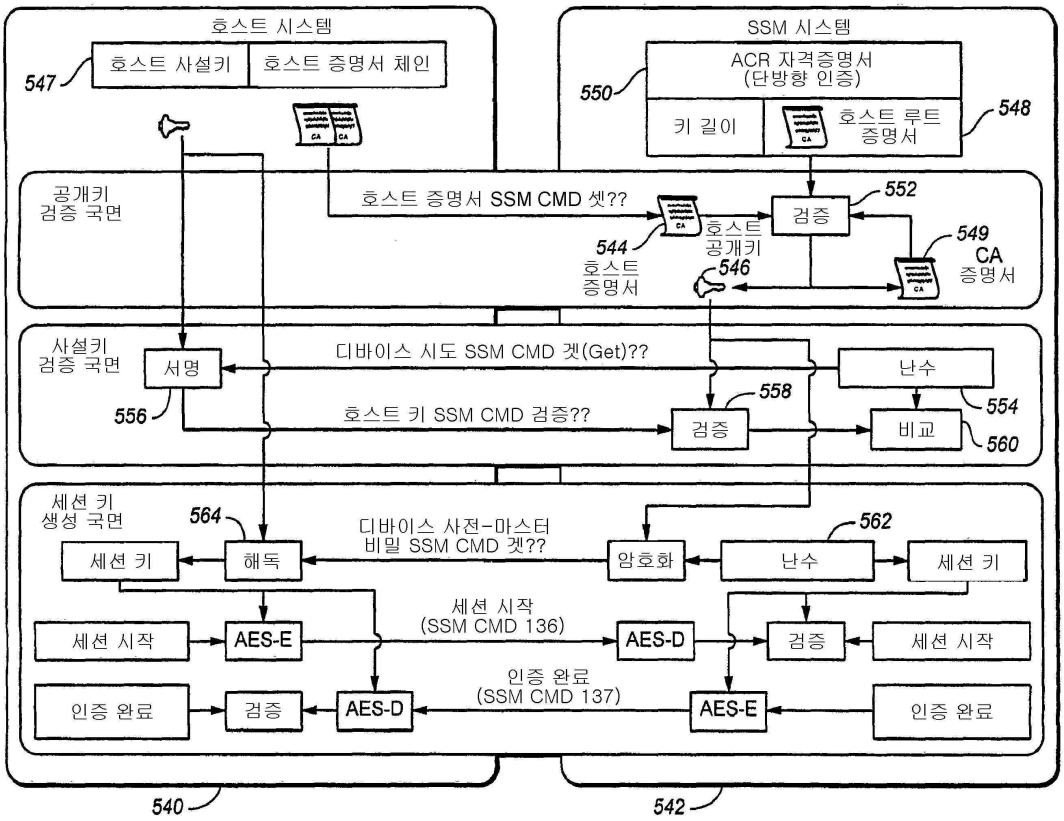
도면20


호스트 루트
CA 증명서
502
호스트 1 CA
(레벨 2)
증명서
504
호스트 n CA
(레벨 2)
증명서
508
호스트 증명서
506
A1
호스트 1 CA
(레벨 3)
증명서
510
호스트 증명서
514
C1
호스트 증명서
512
B1

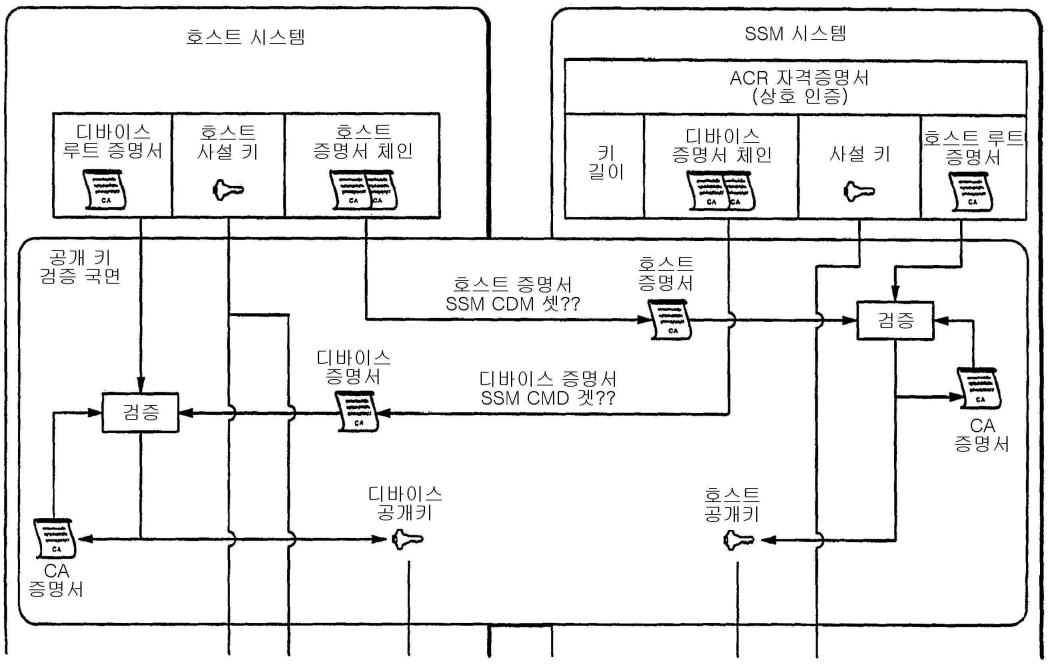
도면21


디바이스 루트
CA 증명서
520
디바이스 1 CA
(제조자) 증명서
522
디바이스 n CA
(제조자) 증명서
526
디바이스
증명서
524
A2
디바이스
증명서
528
B2

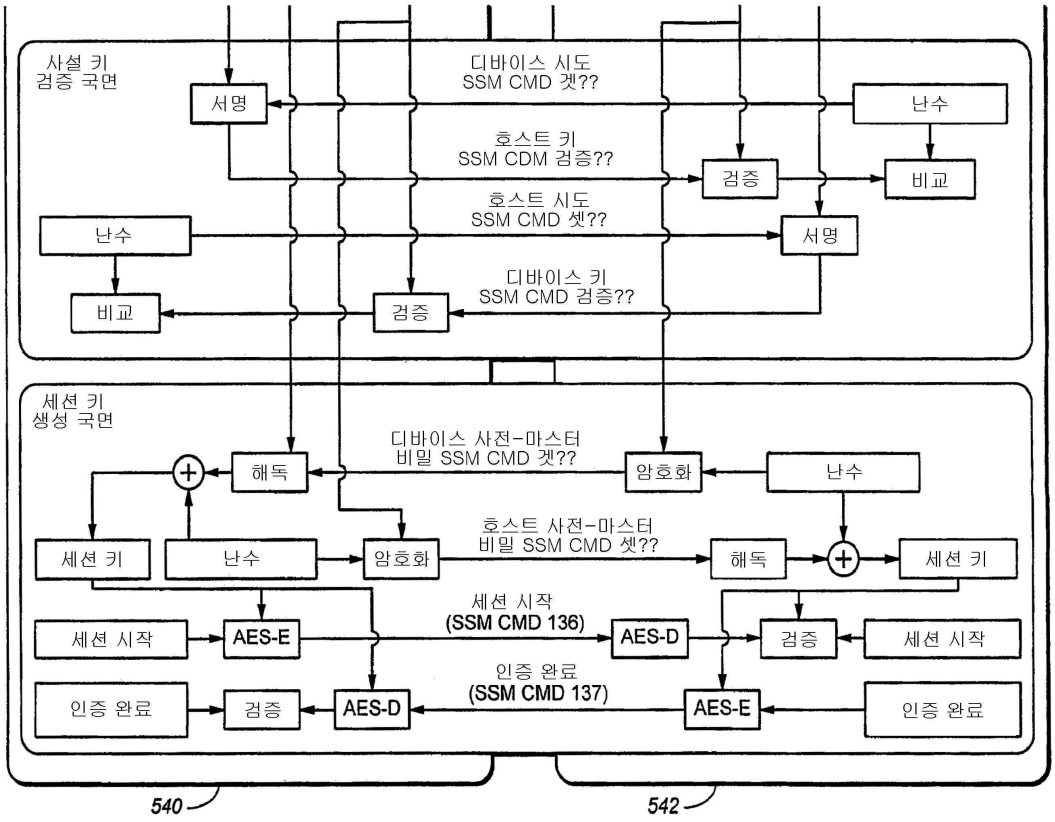
도면22


호스트 시스템
SSM 시스템
547
호스트 사설키
호스트 증명서 체인
550
ACR 자격증명서
(단방향 인증)
키 길이
호스트 루트
증명서
548
공개키
검증 국면
호스트 증명서 SSM CMD 셋??
552
검증
544
호스트
증명서
호스트
공개키
546
549
CA
증명서
사설키
검증 국면
서명
디바이스 시도 SSM CMD 겟(Get)??
난수
554
556
호스트 키 SSM CMD 검증??
558
검증
비교
560
세션 키
생성 국면
564
디바이스 사전-마스터
비밀 SSM CMD 겟??
562
세션 키
해독
암호화
난수
세션 키
세션 시작
(SSM CMD 136)
세션 시작
AES-E
AES-D
검증
세션 시작
인증 완료
(SSM CMD 137)
인증 완료
검증
AES-D
AES-E
인증 완료
540
542

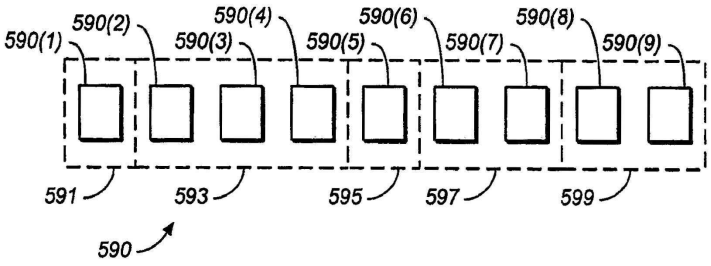
도면23a


호스트 시스템
SSM 시스템
디바이스
루트 증명서
호스트
사설 키
호스트
증명서 체인
ACR 자격증명서
(상호 인증)
키
길이
디바이스
증명서 체인
사설 키
호스트 루트
증명서
공개 키
검증 국면
호스트 증명서
SSM CDM 셋??
호스트
증명서
검증
디바이스
증명서
디바이스 증명서
SSM CMD 겟??
검증
CA
증명서
디바이스
공개키
호스트
공개키
CA
증명서

도면23b



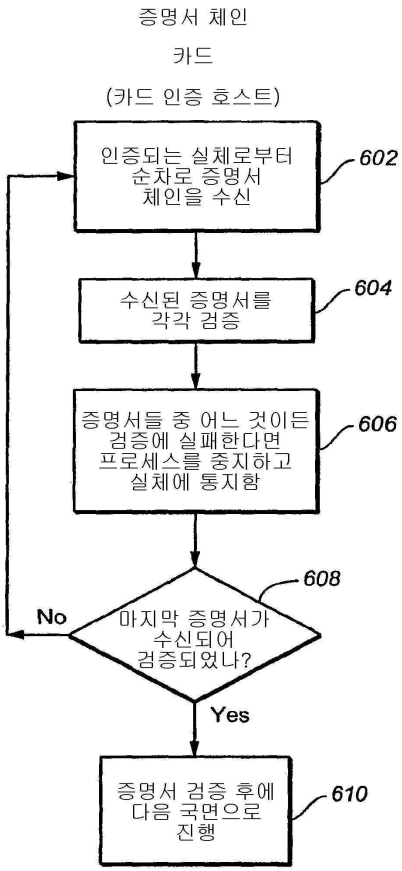
도면24



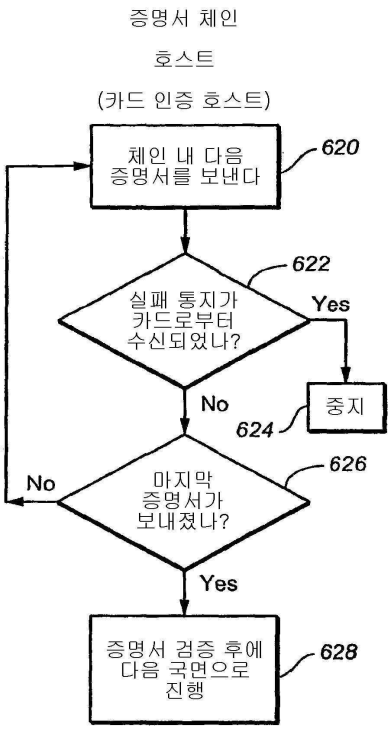
도면25

바이트 오프셋	인수 길이	인수 이름	인수 유형	주석
0-1	2	바이트로 증명서 크기	정수	바이트로 증명서 키의 길이
2	1	"마지막 임" 플래그	디스크리트	이 플래그는 체인에 현재 증명서가 마지막 증명서인지를 나타낸다

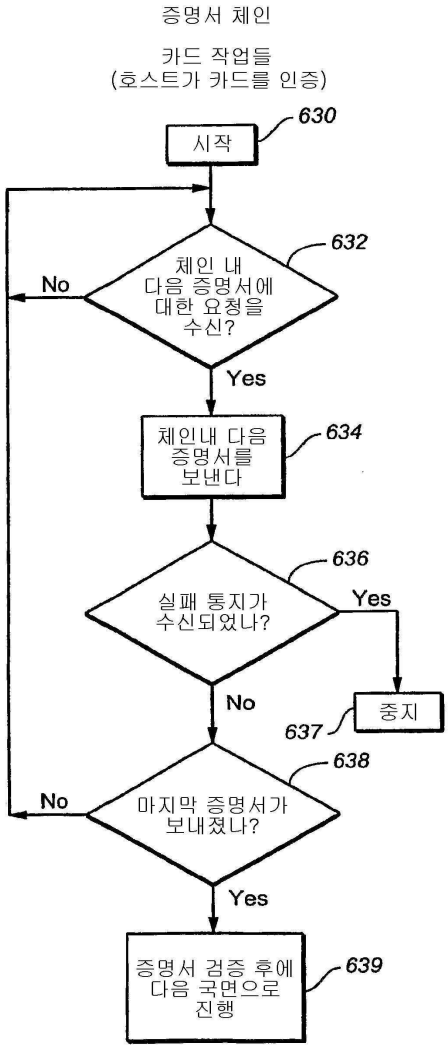
도면26


증명서 체인
카드
(카드 인증 호스트)
인증되는 실체로부터 순차로 증명서 체인을 수신
602
수신된 증명서를 각각 검증
604
증명서들 중 어느 것이든 검증에 실패한다면 프로세스를 중지하고 실체에 통지함
606
608
No
마지막 증명서가 수신되어 검증되었나?
Yes
증명서 검증 후에 다음 국면으로 진행
610

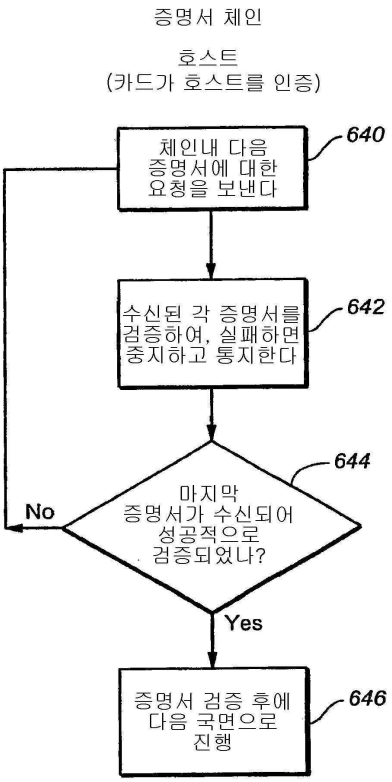
도면27


증명서 체인
호스트
(카드 인증 호스트)
체인 내 다음 증명서를 보낸다
620
622
실패 통지가 카드로부터 수신되었나?
Yes
No
중지
624
626
No
마지막 증명서가 보내졌나?
Yes
증명서 검증 후에 다음 국면으로 진행
628

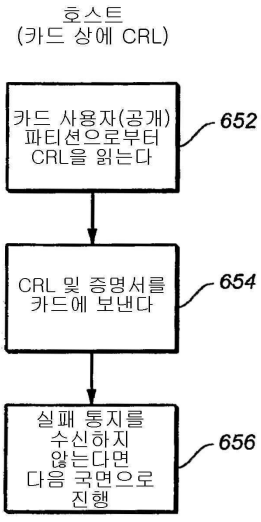
도면28


증명서 체인
카드 작업들
(호스트가 카드를 인증)
630
시작
632
체인 내 다음 증명서에 대한 요청을 수신?
No
Yes
634
체인내 다음 증명서를 보낸다
636
실패 통지가 수신되었나?
Yes
No
637
중지
638
마지막 증명서가 보내졌나?
No
Yes
639
증명서 검증 후에 다음 국면으로 진행

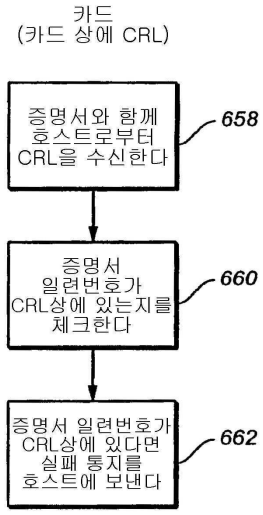
도면29


증명서 체인
호스트
(카드가 호스트를 인증)
체인내 다음 증명서에 대한 요청을 보낸다
640
수신된 각 증명서를 검증하여, 실패하면 중지하고 통지한다
642
마지막 증명서가 수신되어 성공적으로 검증되었나?
644
No
Yes
증명서 검증 후에 다음 국면으로 진행
646

도면30


호스트
(카드 상에 CRL)
카드 사용자(공개) 파티션으로부터 CRL을 읽는다
652
CRL 및 증명서를 카드에 보낸다
654
실패 통지를 수신하지 않는다면 다음 국면으로 진행
656

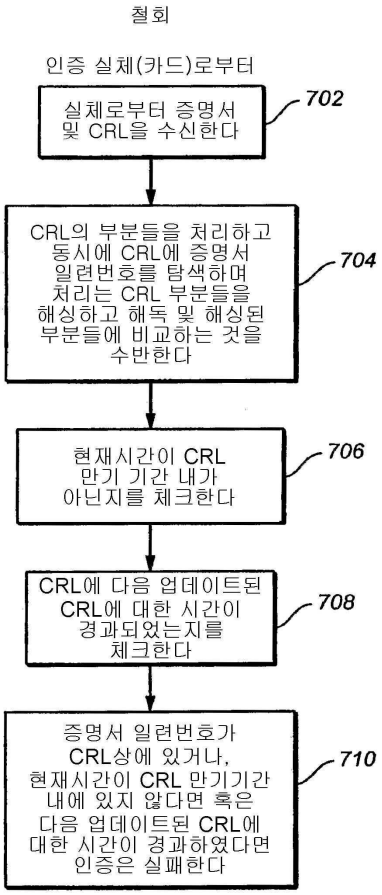
도면31


카드
(카드 상에 CRL)
증명서와 함께
호스트로부터
CRL을 수신한다
658
증명서
일련번호가
CRL상에 있는지를
체크한다
660
증명서 일련번호가
CRL상에 있다면
실패 통지를
호스트에 보낸다
662

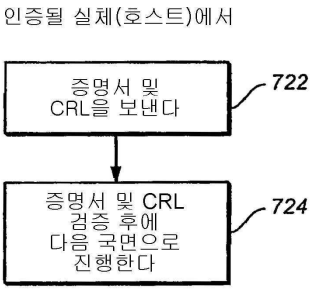
도면32


일련번호
Exp.
일자&기타
공개 키
서명
알고리즘
서명
다음 업데이트
시간
서명
알고리즘
일련번호의
리스트
서명
Certificate
CRL

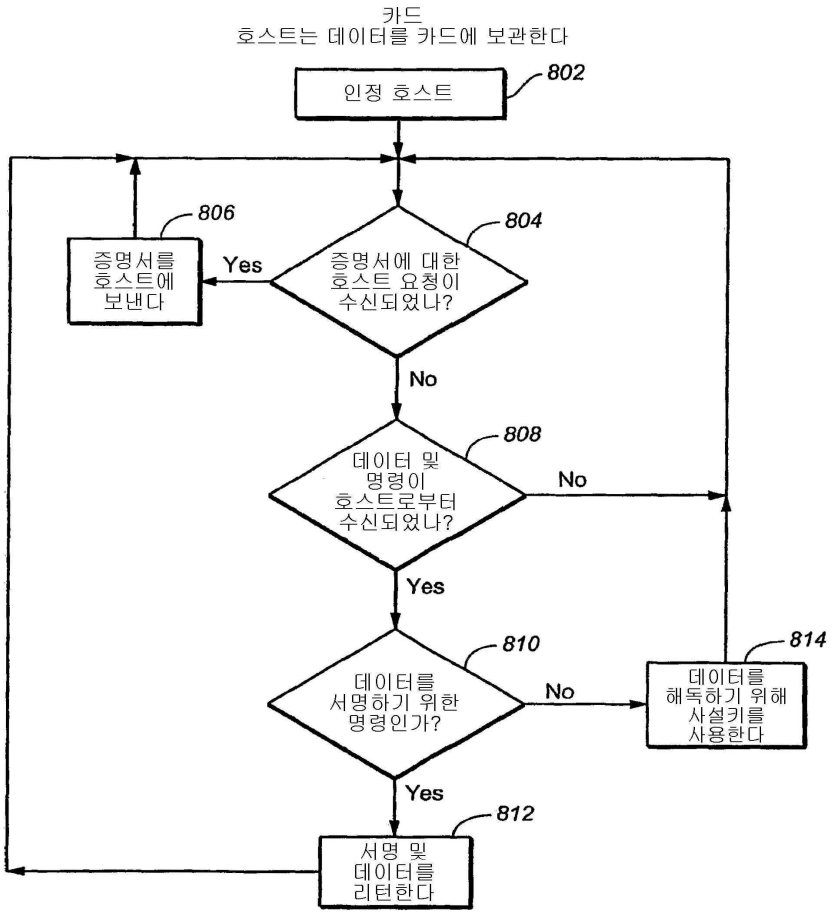
도면33


철회
인증 실체(카드)로부터
실체로부터 증명서 및 CRL을 수신한다
702
CRL의 부분들을 처리하고 동시에 CRL에 증명서 일련번호를 탐색하며 처리는 CRL 부분들을 해싱하고 해독 및 해싱된 부분들에 비교하는 것을 수반한다
704
현재시간이 CRL 만기 기간 내가 아닌지를 체크한다
706
CRL에 다음 업데이트된 CRL에 대한 시간이 경과되었는지를 체크한다
708
증명서 일련번호가 CRL상에 있거나, 현재시간이 CRL 만기기간 내에 있지 않다면 혹은 다음 업데이트된 CRL에 대한 시간이 경과하였다면 인증은 실패한다
710

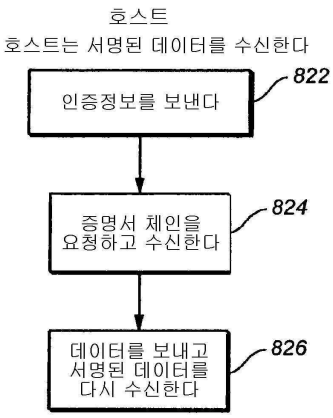
도면34


인증될 실체(호스트)에서
증명서 및 CRL을 보낸다
722
증명서 및 CRL 검증 후에 다음 국면으로 진행한다
724

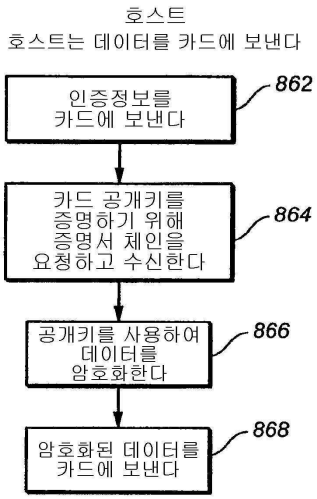
도면35


카드
호스트는 데이터를 카드에 보관한다
인정 호스트
802
804
증명서에 대한 호스트 요청이 수신되었나?
Yes
806
증명서를 호스트에 보낸다
No
808
데이터 및 명령이 호스트로부터 수신되었나?
No
Yes
810
데이터를 서명하기 위한 명령인가?
No
814
데이터를 해독하기 위해 사설키를 사용한다
Yes
812
서명 및 데이터를 리턴한다

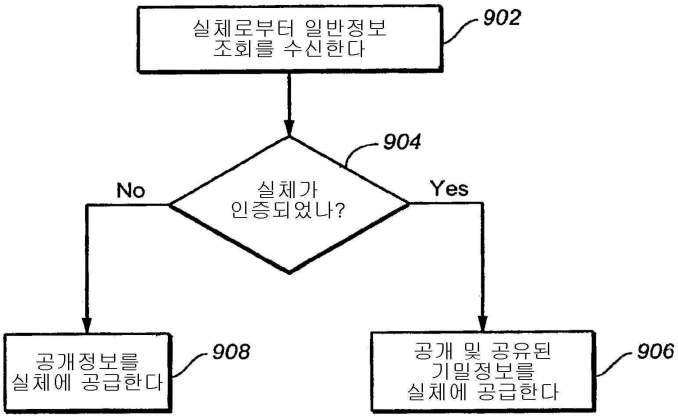
도면36


호스트
호스트는 서명된 데이터를 수신한다
822
인증정보를 보낸다
824
증명서 체인을 요청하고 수신한다
826
데이터를 보내고 서명된 데이터를 다시 수신한다

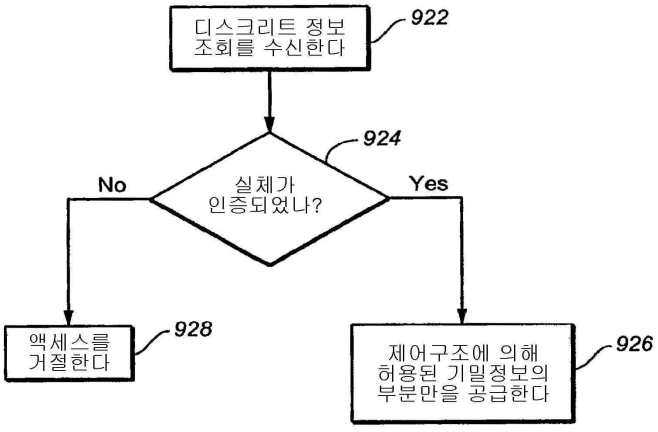
도면37



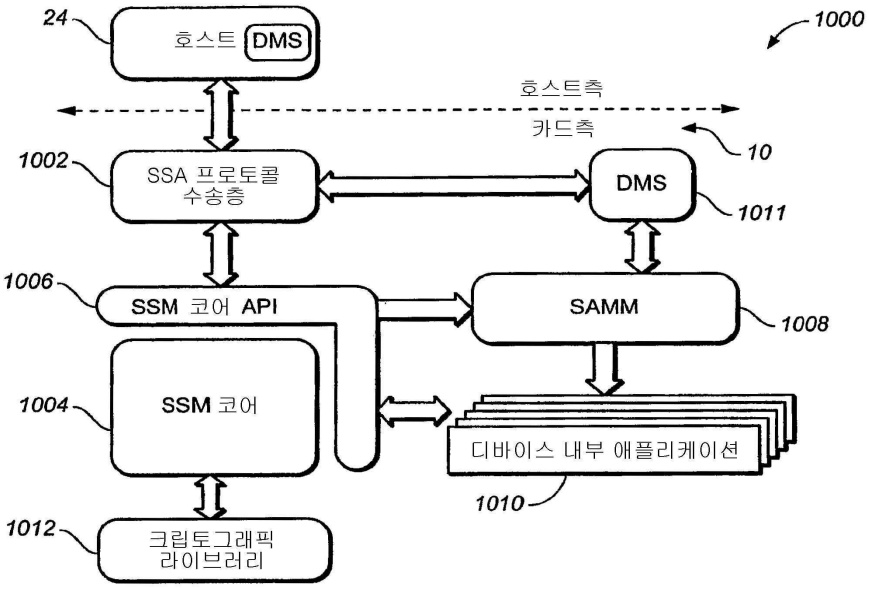
도면38



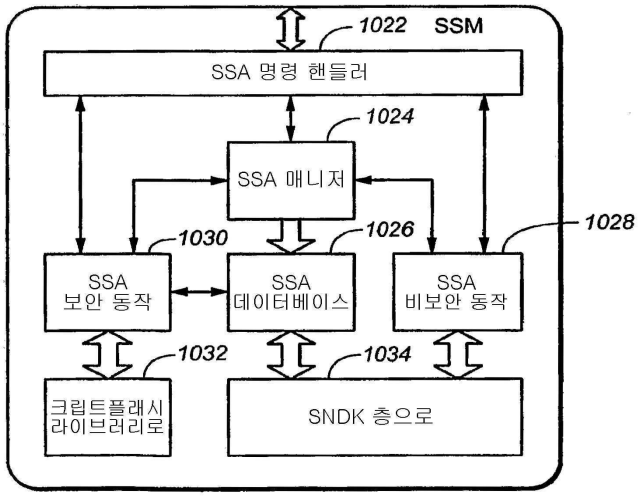
도면39



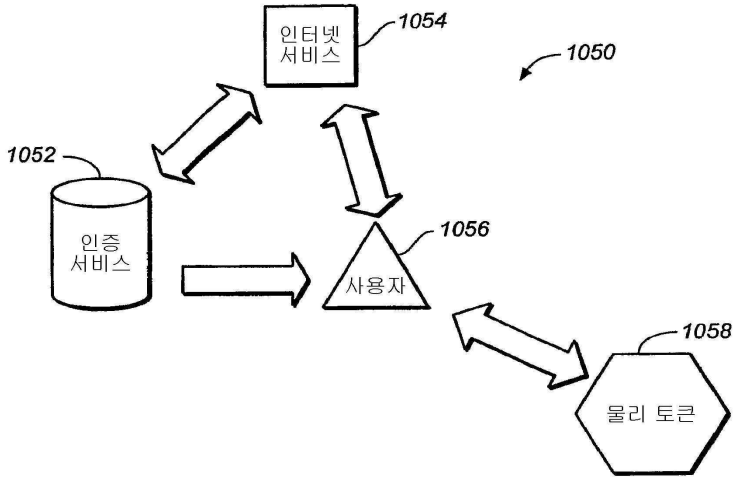
도면40a


24
호스트 DMS
1000
호스트측
카드측
10
1002
SSA 프로토콜
수송층
DMS
1011
1006
SSM 코어 API
SAMM
1008
1004
SSM 코어
디바이스 내부 애플리케이션
1010
1012
크립토그래픽
라이브러리

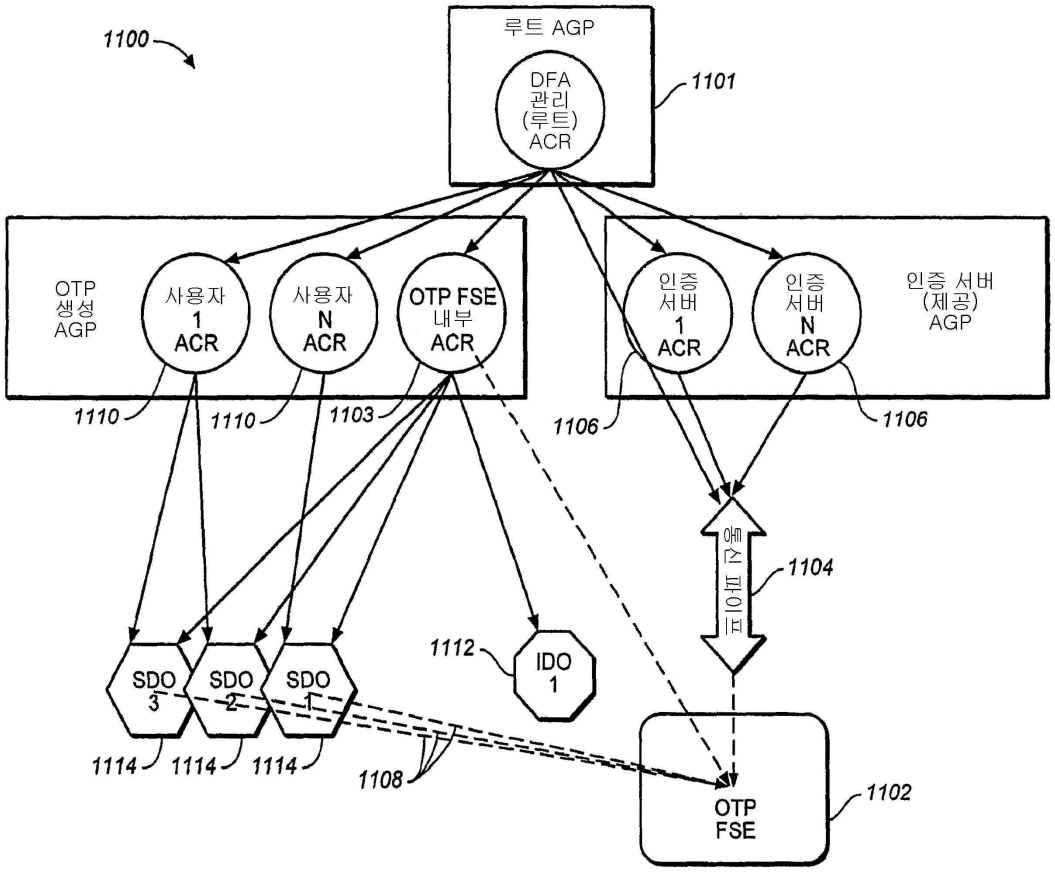
도면40b


1022
SSM
SSA 명령 핸들러
1024
SSA 매니저
1030
1026
1028
SSA
보안 동작
SSA
데이터베이스
SSA
비보안 동작
1032
1034
크립트플래시
라이브러리로
SNDK 층으로

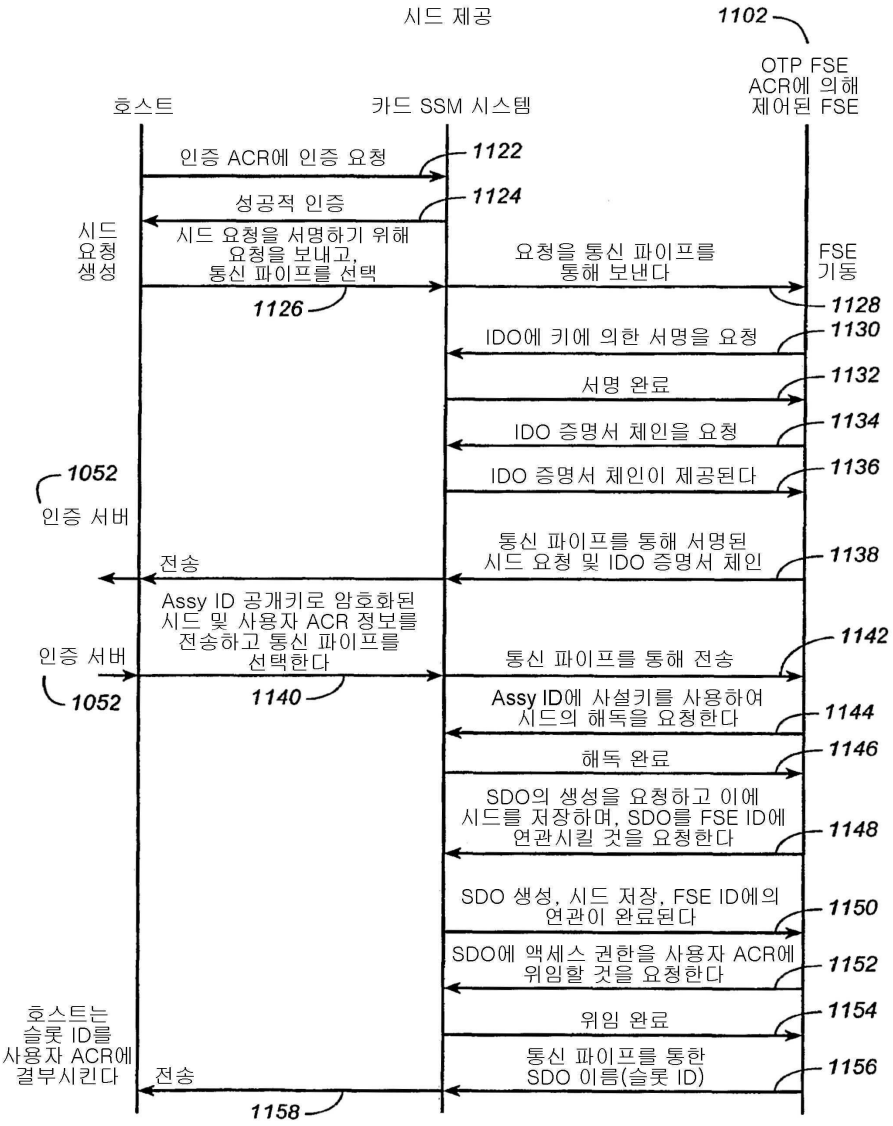
도면41


인터넷 서비스
1054
1050
1052
인증 서비스
1056
사용자
1058
물리 토큰

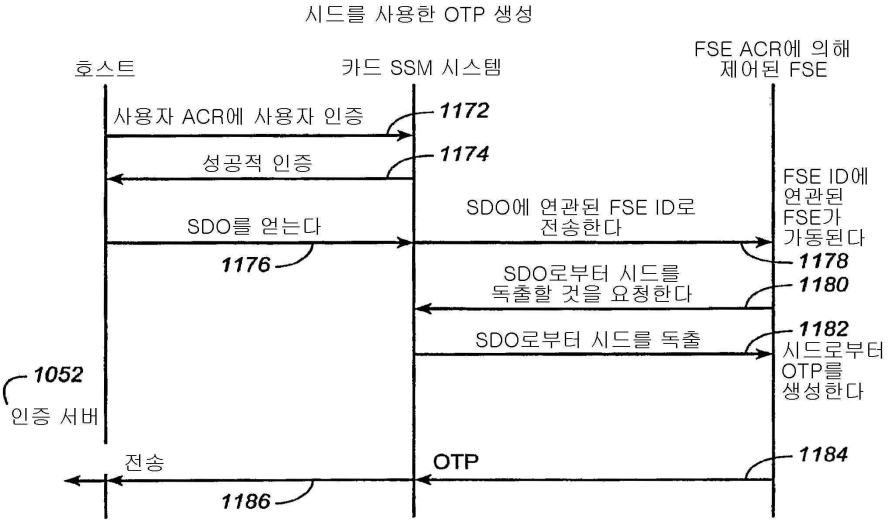
도면42


1100
루트 AGP
DFA 관리 (루트) ACR
1101
OTP 생성 AGP
사용자 1 ACR
사용자 N ACR
OTP FSE 내부 ACR
인증 서버 1 ACR
인증 서버 N ACR
인증 서버 (제공) AGP
1110
1110
1103
1106
1106
프로비저닝
1104
SDO 3
SDO 2
SDO 1
1112
IDO 1
1114
1114
1114
1108
OTP FSE
1102

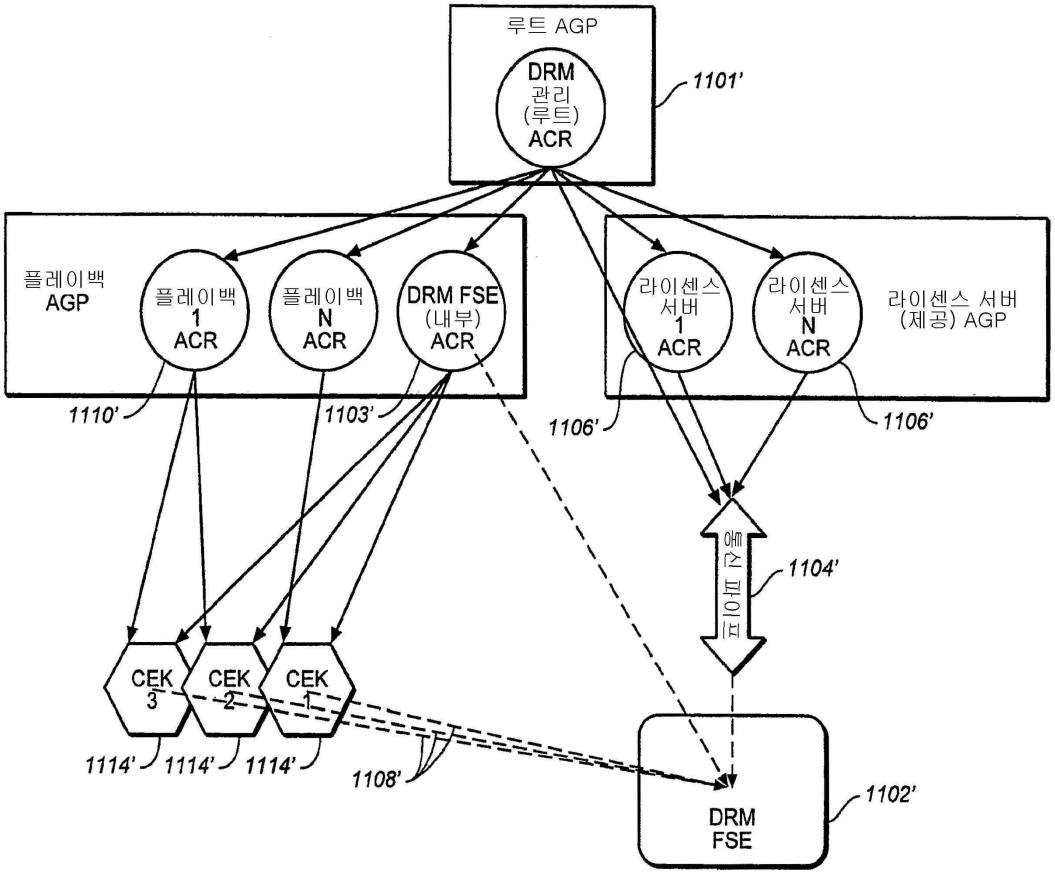
도면43


시드 제공
1102
OTP FSE
ACR에 의해
제어된 FSE
호스트
카드 SSM 시스템
인증 ACR에 인증 요청
1122
성공적 인증
1124
시드
요청
생성
시드 요청을 서명하기 위해
요청을 보내고,
통신 파이프를 선택
요청을 통신 파이프를
통해 보낸다
FSE
기동
1126
1128
IDO에 키에 의한 서명을 요청
1130
서명 완료
1132
IDO 증명서 체인을 요청
1134
1052
인증 서버
IDO 증명서 체인이 제공된다
1136
통신 파이프를 통해 서명된
시드 요청 및 IDO 증명서 체인
1138
전송
Assy ID 공개키로 암호화된
시드 및 사용자 ACR 정보를
전송하고 통신 파이프를
선택한다
인증 서버
통신 파이프를 통해 전송
1142
1052
1140
Assy ID에 사설키를 사용하여
시드의 해독을 요청한다
1144
해독 완료
1146
SDO의 생성을 요청하고 이에
시드를 저장하며, SDO를 FSE ID에
연관시킬 것을 요청한다
1148
SDO 생성, 시드 저장, FSE ID에의
연관이 완료된다
1150
SDO에 액세스 권한을 사용자 ACR에
위임할 것을 요청한다
1152
위임 완료
1154
호스트는
슬롯 ID를
사용자 ACR에
결부시킨다
통신 파이프를 통한
SDO 이름(슬롯 ID)
1156
전송
1158

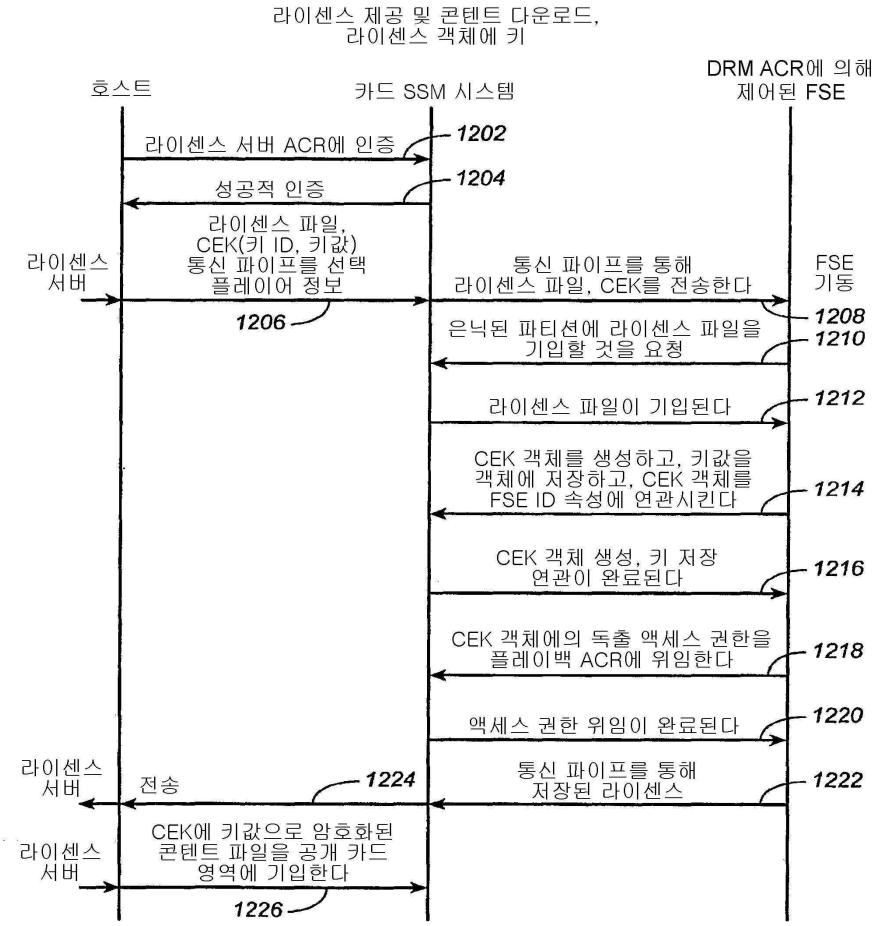
도면44


시드를 사용한 OTP 생성
호스트
카드 SSM 시스템
FSE ACR에 의해 제어된 FSE
사용자 ACR에 사용자 인증
1172
성공적 인증
1174
SDO를 얻는다
1176
SDO에 연관된 FSE ID로 전송한다
FSE ID에 연관된 FSE가 가동된다
1178
SDO로부터 시드를 독출할 것을 요청한다
1180
SDO로부터 시드를 독출
1182
시드로부터 OTP를 생성한다
1052
인증 서버
전송
1186
OTP
1184

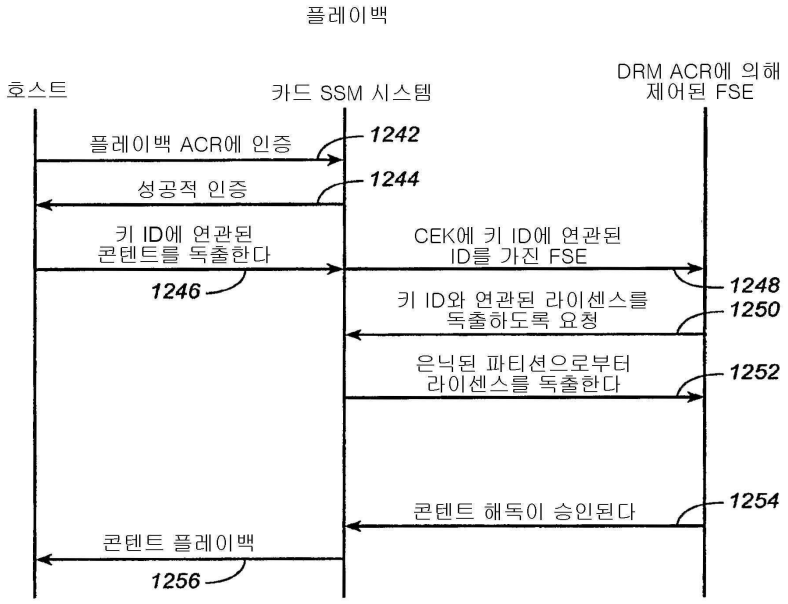
도면45


루트 AGP
DRM 관리 (루트) ACR
1101'
플레이백 AGP
플레이백 1 ACR
플레이백 N ACR
DRM FSE (내부) ACR
라이센스 서버 1 ACR
라이센스 서버 N ACR
라이센스 서버 (제공) AGP
1110'
1103'
1106'
1106'
프라이즈
1104'
CEK 3
CEK 2
CEK 1
1114'
1114'
1114'
1108'
DRM FSE
1102'

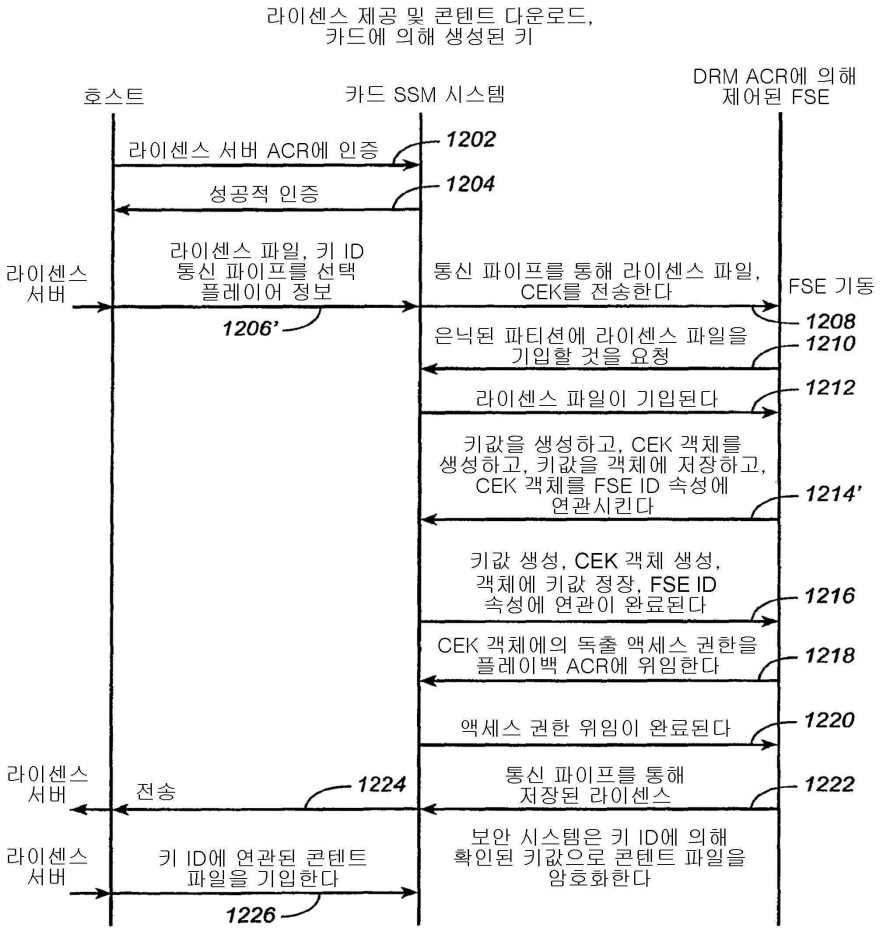
도면46


라이센스 제공 및 콘텐트 다운로드,
라이센스 객체에 키
호스트
카드 SSM 시스템
DRM ACR에 의해
제어된 FSE
라이센스 서버 ACR에 인증
1202
성공적 인증
1204
라이센스 파일,
CEK(키 ID, 키값)
통신 파이프를 선택
플레이어 정보
라이센스
서버
1206
통신 파이프를 통해
라이센스 파일, CEK를 전송한다
FSE
기동
1208
은닉된 파티션에 라이센스 파일을
기입할 것을 요청
1210
라이센스 파일이 기입된다
1212
CEK 객체를 생성하고, 키값을
객체에 저장하고, CEK 객체를
FSE ID 속성에 연관시킨다
1214
CEK 객체 생성, 키 저장
연관이 완료된다
1216
CEK 객체에의 독출 액세스 권한을
플레이백 ACR에 위임한다
1218
액세스 권한 위임이 완료된다
1220
라이센스
서버
전송
1224
통신 파이프를 통해
저장된 라이센스
1222
라이센스
서버
CEK에 키값으로 암호화된
콘텐트 파일을 공개 카드
영역에 기입한다
1226

도면47


플레이백
호스트
카드 SSM 시스템
DRM ACR에 의해
제어된 FSE
플레이백 ACR에 인증
1242
성공적 인증
1244
키 ID에 연관된
콘텐트를 독출한다
1246
CEK에 키 ID에 연관된
ID를 가진 FSE
1248
키 ID와 연관된 라이센스를
독출하도록 요청
1250
은닉된 파티션으로부터
라이센스를 독출한다
1252
콘텐트 해독이 승인된다
1254
콘텐트 플레이백
1256

도면48


라이센스 제공 및 콘텐트 다운로드,
카드에 의해 생성된 키
호스트
카드 SSM 시스템
DRM ACR에 의해
제어된 FSE
라이센스 서버 ACR에 인증
1202
성공적 인증
1204
라이센스
서버
라이센스 파일, 키 ID
통신 파이프를 선택
플레이어 정보
1206'
통신 파이프를 통해 라이센스 파일,
CEK를 전송한다
FSE 기동
1208
은닉된 파티션에 라이센스 파일을
기입할 것을 요청
1210
라이센스 파일이 기입된다
1212
키값을 생성하고, CEK 객체를
생성하고, 키값을 객체에 저장하고,
CEK 객체를 FSE ID 속성에
연관시킨다
1214'
키값 생성, CEK 객체 생성,
객체에 키값 정장, FSE ID
속성에 연관이 완료된다
1216
CEK 객체에의 독출 액세스 권한을
플레이백 ACR에 위임한다
1218
액세스 권한 위임이 완료된다
1220
라이센스
서버
전송
1224
통신 파이프를 통해
저장된 라이센스
1222
라이센스
서버
키 ID에 연관된 콘텐트
파일을 기입한다
1226
보안 시스템은 키 ID에 의해
확인된 키값으로 콘텐트 파일을
암호화한다