(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0017011
(43) 공개일자 2017년02월14일

(51) 국제특허분류(Int. Cl.)
H04L 12/26 (2006.01) *G06F 11/30* (2006.01)
H04L 29/06 (2006.01) *H04L 29/08* (2006.01)
H04W 12/04 (2009.01)
(52) CPC특허분류
H04L 43/10 (2013.01)
G06F 11/3051 (2013.01)
(21) 출원번호 10-2017-7003301(분할)
(22) 출원일자(국제) 2015년02월05일
심사청구일자 없음
(62) 원출원 특허 10-2016-7021392
원출원일자(국제) 2015년02월05일
심사청구일자 2016년08월04일
(85) 번역문제출일자 2017년02월06일
(86) 국제출원번호 PCT/US2015/014639
(87) 국제공개번호 WO 2015/120161
국제공개일자 2015년08월13일
(30) 우선권주장
61/935,967 2014년02월05일 미국(US)
14/614,914 2015년02월05일 미국(US)

(71) 출원인
애플 인크.
미합중국 95014 캘리포니아 쿠퍼티노 인피니트 루프 1
(72) 발명자
맥러플린, 케빈 피.
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 60-1 아이오에스 인피니트 루프 1
벅스, 앤드류
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 60-1 아이오에스 인피니트 루프 1
(뒷면에 계속)
(74) 대리인
장덕순, 백만기

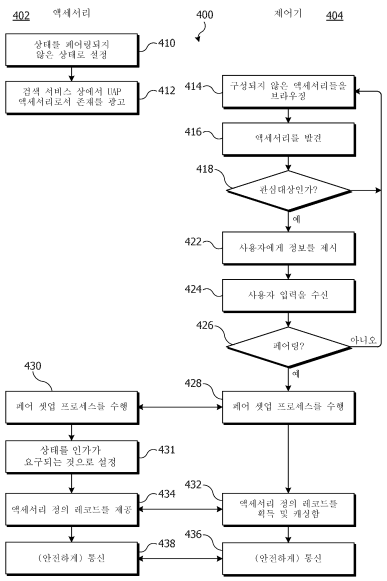
전체 청구항 수 : 총 25 항

(54) 발명의 명칭 **제어기와 액세서리 사이의 통신을 위한 균일한 통신 프로토콜**

(57) 요 약

균일한 프로토콜은 제어기 디바이스와, 제어기에 의해 제어되는 액세서리 디바이스 사이의 안전하고 인증된 통신을 용이하게 할 수 있다. 액세서리와 제어기는 페어링을 확립할 수 있으며, 그 존재는 추후에 검증되고 보안 통신 세션을 생성하는 데 사용될 수 있다. 액세서리는 서비스들의 집합으로서 액세서리를 정의하는 액세서리 정의 레코드를 제공할 수 있으며, 각각의 서비스는 하나 이상의 특성을 갖는다. 보안 통신 세션 내에서, 제어기는 특성들을 질의하여 액세서리 상태를 결정하고/하거나 특성들을 수정하여 그것의 상태를 변경하도록 액세서리에 지시할 수 있다.

대 표 도 - 도4



(52) CPC특허분류
H04L 63/061 (2013.01)
H04L 63/0823 (2013.01)
H04L 63/18 (2013.01)
H04L 67/303 (2013.01)
H04W 12/04 (2013.01)
(72) 발명자
라마, 스리니바스
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 60-1아이오에스 인피니트 루프 1
나다서, 아누쉬
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 60-1아이오에스 인피니트 루프 1
아부안, 조
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 302-4에이피피 인피니트 루프 1
브래들리, 밥
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 306-1에프엠 인피니트 루프 1
둘리, 크레이그
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 60-1아이오에스 인피니트 루프 1
골렘베스키 주니어, 그레그
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 81-2피티 인피니트 루프 1
마티아스, 아룬
미국 95014 캘리포니아주 쿠퍼티노 엠/에스 60-1아이오에스 인피니트 루프 1

명 세 서

청구범위

청구항 1

방법으로서,

제어기에 의해, 페어링(pairing)에 이용가능한 액세서리를 검출하는 단계;

상기 제어기에 의해, 상기 액세서리와 페어링을 확립하는 페어 셋업 동작을 수행하는 단계 - 상기 페어 셋업 동작은,

상기 액세서리와 상기 제어기 사이에서 교환되는 대역외 정보 항목을 사용하여 공유 비밀(shared secret)을 확립하는 것;

상기 액세서리의 장기간 공개 키(long-term public key) 및 상기 제어기의 장기간 공개 키를 안전하게 교환하기 위해 상기 공유 비밀을 사용하는 것; 및

상기 제어기에 의해, 상기 액세서리의 장기간 공개 키를 상기 액세서리의 식별자와 연관하여 안전하고 지속적으로 저장하는 것을 포함함 -;

상기 제어기에 의해, 상기 페어링된 액세서리로부터 액세서리 정의 레코드를 획득하는 단계 - 상기 액세서리 정의 레코드는 서비스들의 세트를 포함하며, 각각의 서비스는 특성들의 세트를 갖고, 각각의 특성은 액세서리 상태의 양상(aspect)을 표현함 -; 및

상기 제어기에 의해, 상기 액세서리 정의 레코드 내에 명시된 상기 특성들 중 하나 이상을 수정하려는 요청을 상기 페어링된 액세서리에 전달하는 단계 - 상기 요청은 상기 페어링된 액세서리의 동작을 호출하는 명령어로서 상기 페어링된 액세서리에 의해 해석가능함 - 를 포함하는, 방법.

청구항 2

제1항에 있어서, 상기 액세서리의 상기 동작을 호출하기 이전에,

상기 제어기에 의해, 상기 페어링된 액세서리와 페어 검증 동작을 수행하는 단계를 추가로 포함하며, 상기 페어 검증 동작은,

상기 제어기가 상기 페어링된 액세서리의 장기간 공개 키 및 상기 제어기의 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을, 상기 페어링된 액세서리에 송신하는 것; 및

상기 페어링된 액세서리로부터, 상기 페어링된 액세서리가 상기 제어기의 장기간 공개 키 및 상기 페어링된 액세서리의 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을 수신하는 것을 포함하며,

상기 제어기는 상기 페어 검증 동작이 성공하는 경우에만 상기 페어링된 액세서리의 상기 동작을 호출하는, 방법.

청구항 3

제2항에 있어서, 상기 페어 검증 동작을 수행하는 단계는 세션 키를 생성하는 단계를 추가로 포함하고, 상기 세션 키는 상기 액세서리 정의 레코드 내에 명시된 상기 특성 중 하나 이상을 수정하려는 상기 요청을 암호화하는 데 사용되는, 방법.

청구항 4

제1항에 있어서,

상기 제어기에 의해, 상기 액세서리 정의 레코드 내에 정의된 상기 특성들의 세트 중 특정 특성의 변화의 통지들에 가입하려는 요청을 상기 페어링된 액세서리에 송신하는 단계를 추가로 포함하며, 상기 가입하려는 요청은 상기 특정 특성을 명시하고 통지 모드를 추가로 명시하는, 방법.

청구항 5

제4항에 있어서, 상기 명시된 통지 모드는 이벤트-메시지 통지 모드이며, 상기 방법은,

상기 가입하려는 요청을 송신한 이후에, 상기 제어기에 의해, 상기 액세서리로부터 비요청(unsolicited) 응답 메시지를 수신하는 단계를 추가로 포함하며, 상기 비요청 응답 메시지는 상기 제어기에 의해 명시된 상기 특성의 변화의 통지를 포함하는, 방법.

청구항 6

제1항에 있어서, 상기 대역외 정보 항목은,

상기 액세서리에 의해 사용자에게 제공되고 상기 사용자에 의해 상기 제어기에 제공된 셋업 코드; 또는

상기 액세서리 이외의 소스로부터 상기 제어기에 의해 획득된 디지털 인증서 검증 정보

중 하나 이상을 포함하며, 상기 디지털 인증서 검증 정보는 상기 액세서리로부터 상기 제어기에 의해 수신된 디지털 인증서 및 서명을 검증하는 데 사용가능한, 방법.

청구항 7

제1항에 있어서,

상기 제어기에 의해, 다른 제어기를 대신해서 상기 페어링된 액세서리와 페어 추가 동작을 수행하는 단계를 추가로 포함하며, 상기 페어 추가 동작은 상기 다른 제어기의 장기간 공개 키 또는 상기 액세서리에 의해 상기 다른 제어기를 검증하는 데 사용가능한 인증서 중 하나 또는 둘 다를, 상기 제어기에 의해 상기 액세서리에 제공하는 것을 포함하는, 방법.

청구항 8

제어기로서,

하나 이상의 액세서리와 통신하는 통신 인터페이스;

하나 이상의 페어링된 액세서리에 대한 페어링 레코드들을 안전하게 저장하는 보안 저장 요소;

하나 이상의 페어링된 액세서리에 대한 액세서리 정의 레코드들을 저장하도록 구성된 데이터 저장 요소; 및

상기 통신 인터페이스, 상기 보안 저장 요소, 및 상기 데이터 저장 요소에 결합된 처리 서브시스템을 포함하며, 상기 처리 서브시스템은,

하나 이상의 액세서리와 페어링을 확립하고 - 액세서리와 페어링을 확립하는 것은,

대역외 정보 항목을 획득하는 것;

상기 액세서리로부터, 상기 액세서리의 장기간 공개 키 및 상기 액세서리가 상기 대역외 정보 항목을 갖는다는 증명을 안전하게 획득하는 것;

상기 제어기의 장기간 공개 키를 상기 액세서리에 안전하게 제공하는 것; 및

상기 액세서리에 대한 페어링 레코드를 생성하고 상기 보안 저장 요소에 지속적으로 저장하는 것 - 상기 페어링 레코드는 상기 액세서리의 장기간 공개 키를 포함함 - 을 포함함 -;

액세서리와 페어링을 확립한 이후에, 상기 페어링된 액세서리로부터 액세서리 정의 레코드를 획득하고 - 상기 액세서리 정의 레코드는 서비스들의 세트를 포함하며, 각각의 서비스는 특성들의 세트를 갖고, 각각의 특성은 액세서리 상태의 양상을 표현함 -;

상기 액세서리 정의 레코드 내에 명시된 상기 특성들 중 하나 이상을 수정하려는 요청을 상기 액세서리에 전달함으로써 상기 페어링된 액세서리의 동작을 호출하도록 구성된, 제어기.

청구항 9

제8항에 있어서, 상기 통신 인터페이스는 적어도 2개의 상이한 전송 프로토콜 스택을 포함하고, 상기 통신 인터

페이스는 특정 액세서리와 통신하는 데 사용될 상기 전송 프로토콜 스택들 중 하나를 선택하는, 제어기.

청구항 10

제9항에 있어서, 상기 적어도 2개의 상이한 전송 프로토콜 스택은 블루투스 LE 프로토콜 스택 및 HTTP/IP 프로토콜 스택을 포함하는, 제어기.

청구항 11

제8항에 있어서, 상기 처리 서브시스템은 페어링된 액세서리와 페어-검증된(pair-verified) 세션을 확립하도록 추가로 구성되고, 상기 페어 검증된 세션을 확립하는 것은,

상기 제어기가 상기 페어링된 액세서리의 장기간 공개 키 및 상기 제어기의 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을, 상기 페어링된 액세서리에 송신하는 것;

상기 페어링된 액세서리로부터, 상기 페어링된 액세서리가 상기 제어기의 장기간 공개 키 및 상기 페어링된 액세서리의 안전하게 저장된 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을 수신하는 것; 및

상기 제어기와 상기 페어링된 액세서리 사이에서 교환되는 후속 메시지들을 암호화하는 데 사용가능한 하나 이상의 세션 키를 확립하는 것을 포함하는, 제어기.

청구항 12

제11항에 있어서, 상기 처리 서브시스템은 확립된 페어-검증된 세션 내에서 페어 추가 동작을 수행하도록 추가로 구성되며, 상기 페어 추가 동작을 수행하는 것은 상기 페어링된 액세서리가 페어링을 확립해야 할 다른 제어기의 장기간 공개 키를 상기 페어링된 액세서리에 송신하는 것을 포함하는, 제어기.

청구항 13

방법으로서,

액세서리에 의해, 상기 액세서리가 페어링에 이용가능함을 광고하는 단계;

상기 액세서리에 의해, 제어기와 페어링을 확립하는 페어 셋업 동작을 수행하는 단계 - 상기 페어 셋업 동작은,

상기 액세서리와 상기 제어기 사이에서 교환되는 대역외 정보 항목을 사용하여 공유 비밀을 확립하는 것;

상기 액세서리의 장기간 공개 키 및 상기 제어기의 장기간 공개 키를 안전하게 교환하기 위해 상기 공유 비밀을 사용하는 것; 및

상기 액세서리에 의해, 상기 제어기의 장기간 공개 키를 상기 제어기의 식별자와 연관하여 안전하고 지속적으로 저장하는 것을 포함함 -;

상기 액세서리에 의해 상기 페어링된 제어기에, 상기 페어링된 액세서리에 대한 액세서리 정의 레코드를 제공하는 단계 - 상기 액세서리 정의 레코드는 서비스들의 세트를 포함하며, 각각의 서비스는 특성들의 세트를 갖고, 각각의 특성은 액세서리 상태의 양상에 대응함 -;

상기 액세서리에 의해, 상기 액세서리 정의 레코드 내에 명시된 상기 특성들 중 하나 이상을 수정하려는 요청을 상기 페어링된 제어기로부터 수신하는 단계; 및

상기 액세서리에 의해, 상기 수신된 요청에 응답하여 호출된 동작을 수행하는 단계를 포함하는, 방법.

청구항 14

제13항에 있어서,

상기 액세서리에서, 상기 액세서리가 페어링 모드로 진입해야 한다는 것을 나타내는 사용자 행동을 검출하는 단계를 추가로 포함하며,

상기 액세서리가 페어링에 이용가능함을 광고하는 단계는 상기 사용자 행동을 검출하는 단계에 응답하여 수행되는, 방법.

청구항 15

제13항에 있어서, 하나의 제어기와 상기 페어 셋업 동작을 수행한 이후에, 상기 액세서리는 후속적으로, 상기 페어링된 제어기의 장기간 공개 키가 안전하게 그리고 지속적으로 저장된 상태를 유지하는 동안은 임의의 다른 제어기와 상기 페어 셋업 동작을 수행하는 것을 거부하는, 방법.

청구항 16

제13항에 있어서, 상기 특성들 중 하나 이상을 수정하려는 요청을 상기 제어기로부터 수신하기 이전에,

상기 액세서리에 의해, 상기 페어링된 제어기와 페어 검증 동작을 수행하는 단계를 추가로 포함하며, 상기 페어 검증 동작은,

상기 페어링된 제어기로부터, 상기 페어링된 제어기가 상기 액세서리의 장기간 공개 키 및 상기 페어링된 제어기의 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을 수신하는 것; 및

상기 액세서리가 상기 페어링된 제어기의 장기간 공개 키 및 상기 액세서리의 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을, 상기 페어링된 제어기에 송신하는 것을 포함하며,

상기 액세서리는 상기 페어 검증 동작이 성공하는 경우에만 상기 요청에 응답하여 상기 호출된 동작을 수행하는, 방법.

청구항 17

제16항에 있어서, 상기 페어 검증 동작을 수행하는 단계는 세션 키를 생성하는 단계를 추가로 포함하고, 상기 세션 키는 상기 액세서리 정의 레코드 내에 명시된 상기 특성 중 하나 이상을 수정하려는 상기 수신된 요청을 복호화하는 데 사용되고,

상기 액세서리는 상기 수신된 요청이 성공적으로 복호화되는 경우에만 상기 호출된 동작을 수행하는, 방법.

청구항 18

제13항에 있어서,

상기 액세서리에 의해, 상기 액세서리 정의 레코드 내에 정의된 상기 특성들의 세트 중 특정 특성의 변화의 통지들에 가입하려는 요청을 페어링된 제어기로부터 수신하는 단계 - 상기 가입하려는 요청은 상기 특정 특성을 명시하고 통지 모드를 추가로 명시함 -; 및

상기 명시된 특성이 변화하는 경우, 상기 액세서리에 의해, 상기 명시된 통지 모드에 기초하여 상기 페어링된 제어기에 대한 업데이트 통지를 생성하는 단계를 추가로 포함하는, 방법.

청구항 19

제18항에 있어서, 상기 명시된 통지 모드는 이벤트-메시지 통지 모드이고, 상기 업데이트를 생성하는 단계는 비요청 응답 메시지를 상기 페어링된 제어기에 송신하는 단계를 포함하고, 상기 비요청 응답 메시지는 상기 명시된 특성의 변화의 통지를 포함하는, 방법.

청구항 20

제13항에 있어서, 상기 대역외 정보 항목은, 상기 액세서리에 의해 로컬로 저장되고 대역외 통신 채널을 사용하여 상기 제어기에 전달되는 셋업 코드를 포함하는, 방법.

청구항 21

제13항에 있어서,

상기 액세서리에 의해, 다른 제어기를 대신하여 상기 페어링된 제어기와 페어 추가 동작을 수행하는 단계를 추가로 포함하며, 상기 페어 추가 동작은,

상기 액세서리에 의해, 상기 페어링된 제어기로부터, 상기 다른 제어기의 장기간 공개 키를 수신하는 것; 및

상기 액세서리에 의해, 상기 다른 제어기의 장기간 공개 키를 상기 다른 제어기의 식별자와 연관하여 안전하고 지속적으로 저장하는 것을 포함하고,

상기 페어 추가 동작 이후에, 상기 다른 제어기는 제2 페어링된 제어기로서 상기 액세서리에 의해 취급되는, 방법.

청구항 22

액세서리로서,

하나 이상의 제어기와 통신하는 통신 인터페이스;

동작 컴포넌트;

페어링 레코드들을 안전하게 저장하는 보안 저장 요소;

액세서리 정의 레코드를 저장하도록 구성된 액세서리 데이터 저장 요소 - 상기 액세서리 정의 레코드는 서비스들의 세트를 포함하며, 각각의 서비스는 특성들의 세트를 갖고, 각각의 특성은 상기 동작 컴포넌트의 상태를 포함하는 액세서리 상태의 양상을 표현함 -; 및

상기 통신 인터페이스, 상기 동작 컴포넌트, 상기 보안 저장 요소, 및 상기 액세서리 데이터 저장 요소에 결합된 처리 서브시스템을 포함하며, 상기 처리 서브시스템은,

하나 이상의 제어기와 페어링을 확립하고 - 제어기와 페어링을 확립하는 것은,

대역외 정보 항목을 획득하는 것;

상기 제어기로부터, 상기 제어기의 장기간 공개 키 및 상기 제어기가 상기 대역외 정보 항목을 갖는다는 증명을 안전하게 획득하는 것;

상기 액세서리의 장기간 공개 키를 상기 제어기에 안전하게 제공하는 것; 및

상기 제어기에 대한 페어링 레코드를 생성하고 상기 보안 저장 요소에 지속적으로 저장하는 것 - 상기 페어링 레코드는 상기 제어기의 장기간 공개 키를 포함함 - 을 포함함 -;

제어기와 페어링을 확립한 이후에, 상기 저장된 액세서리 정의 레코드를 상기 페어링된 제어기에 제공하고;

상기 페어링된 제어기로부터, 상기 액세서리 정의 레코드 내에 명시된 상기 특성들 중 하나 이상을 수정하려는 요청을 수신하고;

상기 요청에 응답하여 상기 적어도 하나의 동작 컴포넌트의 동작을 달성하도록 구성된, 액세서리.

청구항 23

제22항에 있어서, 상기 처리 서브시스템은 페어링된 제어기와 페어-검증된 세션을 확립하도록 추가로 구성되며, 상기 페어-검증된 세션을 확립하는 것은,

상기 페어링된 제어기로부터, 상기 페어링된 제어기가 상기 액세서리의 장기간 공개 키 및 상기 페어링된 제어기의 안전하게 저장된 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을 수신하는 것;

상기 페어링된 제어기에, 상기 액세서리가 상기 페어링된 제어기의 장기간 공개 키 및 상기 액세서리의 장기간 공개 키에 대응하는 장기간 비밀 키를 갖는다는 증명을 송신하는 것; 및

상기 페어링된 제어기와 상기 액세서리 사이에서 교환되는 후속 메시지들을 암호화하는 데 사용가능한 하나 이상의 세션 키를 확립하는 것을 포함하는, 액세서리.

청구항 24

제23항에 있어서, 상기 처리 서브시스템은 확립된 페어-검증된 세션 내에서 페어 추가 동작을 수행하도록 추가로 구성되며, 상기 페어 추가 동작을 수행하는 것은,

상기 페어링된 제어기로부터, 상기 액세서리가 페어링을 확립해야 할 다른 제어기의 장기간 공개 키를 수신하는 것; 및

상기 다른 제어기에 대한 페어링 레코드를 생성하고 상기 보안 저장 요소에 지속적으로 저장하는 것 - 상기 페어링 레코드는 상기 다른 제어기의 상기 수신된 장기간 공개 키를 포함함 - 을 포함하는, 액세서리.

청구항 25

제22항에 있어서, 상기 적어도 하나의 동작 컴포넌트는,

잠금 메커니즘;

전구;

문 열림장치; 또는

카메라 중 하나 이상을 포함하는, 액세서리.

발명의 설명

기 술 분 야

[0001] 관련 출원에 대한 상호 참조

[0002] 본 출원은 2014년 2월 5일자로 출원된, 발명의 명칭이 "Protocols and Specifications for an Accessory Management System"인 미국 가특허 출원 제61/935,967호에 대한 우선권을 주장하며, 그 개시내용은 전체적으로 본 명세서에 참고로 포함된다. 본 출원은 또한 2015년 2월 5일자로 출원된, 발명의 명칭이 "Uniform Communication Protocols for Communication Between Controllers and Accessories"인 미국 특허 출원 제14/614,914호에 대한 우선권을 주장하며, 그 개시내용은 전체적으로 본 명세서에 참고로 포함된다.

배 경 기 술

[0003] 본 개시내용은 대체로 전자 디바이스들 사이의 통신에 관한 것으로, 보다 상세하게는 제어기와 액세서리 사이에서 통신하기 위한 균일한 통신 프로토콜(uniform communication protocol)에 관한 것이다.

[0004] 전자 디바이스들은 다양한 응용들에서 점점 더 인기가 많아지고 있다. 휴대 전화, 태블릿 컴퓨터, 홈 엔터테인먼트 시스템 등은, 사용자들이 정기적으로 상호작용하는 전자 디바이스들 중 단지 일부이다.

[0005] 점점 더 인기가 많아지고 있는 전자 디바이스들의 다른 카테고리는 온도조절기, 조명 디바이스, 가전제품 등과 같은 다양한 전자적으로 제어가능한 디바이스들을 포함한다.

발명의 내용

[0006] 현재는, 사용자가 다수의 전자적으로 제어가능한 디바이스들 또는 시스템들을 관리하는 것이 어려울 수 있다. 예를 들어, 사용자의 가정은 온도조절기, 전자적으로 제어가능한 조명 시스템, 가정 보안 시스템 등을 가질 수 있다. 각각의 그러한 시스템은 상이한 제조사에 의해 제조될 수 있고, 각 제조사는 전용 제어기 디바이스(예컨대, IR-기반 원격 제어 디바이스), 또는 사용자가 휴대 전화, 태블릿, 또는 가정용 컴퓨터 시스템과 같은 범용 컴퓨팅 디바이스 상에서 설치하고 실행시키는 제어기 앱(controller app)을 제공할 수 있다. 각각의 제어기 디바이스 또는 앱은 전형적으로 특정 제조사의 시스템들에 대해 맞춤화되어 있으며, 다른 제조사들의 시스템들과, 또는 동일한 제조사의 다른 시스템들과도 상호운용가능하지 않을 수도 있다. 그러한 개별적 접근법은 용이하게 확장가능하지 않다. 중앙에서 제어되거나 관리될 수 있는 시스템들의 상이한 어레이로 "스마트 홈" 환경 등을 생성하고자 하는 사용자는, 과잉의 제어기 디바이스들 및/또는 제어기 앱들을 축적할 필요성에 직면하게 된다.

[0007] 본 발명의 소정 실시예들은 제어기 디바이스(또는 "제어기")와, 제어되어야 하는 임의의 수의 다른 전자 디바이스들(본 명세서에서 "액세서리 디바이스들" 또는 단순히 "액세서리들"로 지칭됨) 사이의 통신을 위한 "균일한" 프로토콜에 관한 것이다. 제어기는 예를 들어, 데스크톱 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터, 휴대 전화, 다른 핸드헬드 또는 착용가능한 컴퓨팅 디바이스와 같은 범용 컴퓨팅 디바이스 상에서, 범용 컴퓨팅 디바이스에 적절한 실행가능한 프로그램 코드를 제공함으로써 구현될 수 있으며; 대안적으로, 제어기는 특수-목적 컴퓨팅 디바이스일 수 있다. 액세서리는 제어기에 의해 제어가능한 임의의 디바이스를 포함할 수 있다. 액세서리들의 예들에는 조명 기구, 온도조절기, 문 잠금장치, 자동 문 열림장치(예컨대, 차고문 열림장치), 스틸 또는 비디오 카메라 등이 포함된다. 액세서리들 및 제어기들은 Wi-Fi, 블루투스, 블루투스 LE 등과 같은 표준 전송 프로토콜들을 사용하여 유선 또는 무선 채널들을 통해 서로 통신할 수 있다.

[0008] 일부 실시예들에서, 균일한 액세서리 프로토콜은 액세서리를 서비스들의 집합으로서 정의하기 위한 단순하고 확장가능한 프레임워크를 정의할 수 있으며, 이때 각각의 서비스는 특성들의 세트로서 정의되고, 특성들 각각은

임의의 주어진 시간에 정의된 값을 갖는다. 특성들은 액세서리의 상태의 다양한 미세한(atomic) 양상(aspect)들을 표현할 수 있다. 예를 들어, 온도조절기의 경우, 특성들은 전원(온도조절기 유닛이 온인지 오프인지 여부), 현재 온도(온도조절기에 의해 측정된 실제 온도), 및 목표 온도(온도조절기가 유지하려고 하는 설정가능한 온도)를 포함할 수 있다. 프로토콜은 또한, 액세서리에 커맨드-및-제어(command-and-control) 메시지들(요청들)을 송신하기 위해 제어기에 의해 사용가능하고, 액세서리가 응답 메시지들을 송신하기 위한 메시지 포맷들을 정의할 수 있다. 요청들은, 제어기로 하여금 액세서리 특성을 질의(interrogate)(예컨대, 판독)할 수 있게 하고 일부 경우들에서 액세서리 특성들을 수정(예컨대, 그에 기록)할 수 있게 하며; 예를 들어, 제어기는 액세서리가 온인지 오프인지 여부를 결정하기 위해 전원 특성을 판독할 수 있고 액세서리를 오프 또는 온시키기 위해 전원 특성에 기록할 수 있다. 이와 같이, 임의의 유형의 액세서리는, 기능에 상관없이, 적절한 요청들을 송신함으로써 제어될 수 있다. 액세서리는 액세서리 정의 레코드를 제어기에 제공할 수 있다. 액세서리 정의 레코드는 액세서리의 모든 액세스가능한 특성들에 관한 완전한 정보를 포함할 수 있다. 제어기는 액세서리와 상호작용하는 방법을 결정하기 위해 액세서리 정의 레코드를 사용할 수 있다. 예를 들어, 액세서리 정의 레코드로부터의 정보는, 액세서리를 동작시키기 위한 사용자 인터페이스를 구성할 뿐만 아니라 액세서리로의 요청 메시지들을 구성하기 위해, 제어기에 의해 사용될 수 있다.

[0009] 일부 실시예들에서, 프로토콜은 또한, 특성이 변화할 때 액세서리가 제어기에 통지하기 위해 사용할 수 있는, 통지 메커니즘들을 정의할 수 있다. 예들은, 임의의 특성들이 변화했는지 여부에 관해 제어기가 액세서리에 질의할 수 있는, 수동적 통지(passive notification) 메커니즘들뿐만 아니라; 특정한 특성이 변할 때 액세서리가 선택적으로 하나 이상의 제어기에 대한 메시지들을 생성할 수 있는, 능동적 또는 이벤트-기반 통지(active or event-based notification) 메커니즘들을 포함한다. 다수의 통지 메커니즘이 동시에 지원될 수 있고, 제어기는 특정 액세서리, 서비스, 또는 특성에 사용될 통지 메커니즘을 선택할 수 있다.

[0010] 일부 실시예들에서, 프로토콜은 인가되지 않은 제어기들이 액세서리를 동작시키는 것을 방지하기 위해 사용될 수 있는 보안 조치들을 정의할 수 있다. 예를 들어, 액세서리는, 이전에 액세서리와 페어링(pairing)을 확립했고 따라서 액세서리에 의해 인식되는 제어기로부터의 요청들만을 수락하도록 구성될 수 있다. 페어링을 확립하는 것은 안전한 방식으로 액세서리와 제어기 사이에서 장기간 공개 키(long-term public key)들을 교환하는 것을 포함할 수 있으며, 이때 각각의 디바이스는 키들을 지속적으로 저장한다. 프로토콜은, 액세서리의 소유자/운영자의 승인이 없이 페어링이 확립되는 위험을 감소시키기 위해, 페어 셋업 절차를 명시할 수 있다. 예를 들어, 페어 셋업 프로세스 동안, 사용자는, 하나의 디바이스(예컨대, 액세서리)에 의해 제시된 셋업 코드를 판독하고 셋업 코드를 다른 디바이스(예컨대, 제어기)로 입력하거나, 또는 디바이스들을 서로 물리적으로 근접한 상태로 위치시키도록 요구될 수 있다. 페어링이 확립되면, 페어링은 단대단(end-to-end) 메시지 암호화를 제공하기 위해 활용되어, 페어링된 제어기 및 액세서리만이 그들 사이에서 교환되는 메시지들을 판독할 수 있게 한다. 예를 들어, 이전에 페어링을 확립한 액세서리와 제어기가 재접속할 때, 이들은 (예컨대, 각각이 다른 것의 장기간 공개 키를 소유한다는 것을 증명함으로써) 이전의 페어링을 검증하고 세션-특정 암호화 키들을 생성할 수 있다.

[0011] 일부 실시예들에서, 프로토콜은, 제어기가 그 근방의 호환가능한 액세서리들을 검색하고 구성하기 위한, 절차들을 정의할 수 있다. 이러한 절차들은, 제어기에 의해 관리되는 자동화 제어 시스템에 새로운 액세서리들을 추가하는 작업을 단순화할 수 있다.

[0012] 첨부 도면들과 함께 하기의 상세한 설명은 본 발명의 본질 및 이점들에 대한 보다 양호한 이해를 제공할 것이다.

도면의 간단한 설명

[0013] 도 1은 본 발명의 일 실시예에 따른 홈 환경을 도시한다.

도 2a 내지 도 2d는 본 발명의 일 실시예에 따른 액세서리 특성들의 예시적인 정의들을 도시한다.

도 2e는 본 발명의 일 실시예에 따른 특성에 대해 정의될 수 있는 속성들의 예들을 도시한다.

도 2f는 본 발명의 일 실시예에 따른 정의될 수 있는 확장 특성들의 예들을 도시한다.

도 2g 및 도 2h는 본 발명의 일 실시예에 따른 액세서리 서비스들의 예시적인 정의들을 도시한다.

도 2i는 본 발명의 일 실시예에 따른 정의될 수 있는 액세서리 정보 서비스의 예를 도시한다.

도 2j는 본 발명의 일 실시예에 따른 정의될 수 있는 액세서리 정보 서비스에 대한 특성들의 예들을 도시한다.

도 3a 내지 도 3c는 본 발명의 일 실시예에 따른 액세서리 정의 레코드의 예를 도시한다.

도 4는 본 발명의 일 실시예에 따른 제어기에 의해 액세서리를 검색하기 위한 프로세스의 흐름도이다.

도 5a 내지 도 5k는 본 발명의 일 실시예에 따른 요청 및 응답 메시지들의 예들을 도시한다.

도 6a 내지 도 6e는 본 발명의 일 실시예에 따른 요청 및 응답 메시지들의 추가적인 예들을 도시한다.

도 7은 본 발명의 일 실시예에 따른 수동적 통지 프로세스의 예를 도시한다.

도 8은 본 발명의 일 실시예에 따른 광고형(advertised) 통지 프로세스의 예를 도시한다.

도 9는 본 발명의 일 실시예에 따른 능동적 통지 프로세스의 예를 도시한다.

도 10은 본 발명의 일 실시예에 따른 이벤트 통지 프로세스의 예를 도시한다.

도 11a는 본 발명의 일 실시예에 따른 통지들에 가입하려는 요청 메시지의 예를 도시한다.

도 11b는 본 발명의 일 실시예에 따른 이벤트 메시지의 예를 도시한다.

도 12는 본 발명의 일 실시예에 따른 액세서리를 위한 페어링 프로파일에 대한 예시적인 특성들을 도시한다.

도 13a 내지 도 13c는 본 발명의 일 실시예에 따른 셋업-코드-기반 페어 셋업 프로세스의 예를 도시한다.

도 14a 내지 도 14c는 본 발명의 일 실시예에 따른 인증 칩 및 보안 인증서를 사용하는 페어 셋업 프로세스의 예를 도시한다.

도 15a 내지 도 15f는 본 발명의 일 실시예에 따른 셋업 코드 및 보안 인증서를 사용하는 페어 셋업 프로세스의 예를 도시한다.

도 16은 본 발명의 일 실시예에 따른 일반화된 페어 셋업 프로세스의 예를 도시한다.

도 17a 내지 도 17c는 본 발명의 일 실시예에 따른 페어 검증 프로세스의 예를 도시한다.

도 18a 및 도 18b는 본 발명의 일 실시예에 따른 페어 추가 프로세스의 예를 도시한다.

도 19a 및 도 19b는 본 발명의 일 실시예에 따른 페어 제거 프로세스의 예를 도시한다.

도 20은 본 발명의 일 실시예에 따른 문 잠금 액세서리를 위한 예시적인 동작 환경을 도시한다.

도 21은 본 발명의 일 실시예에 따른 문 잠금 액세서리에 대한 예시적인 서비스 정의들을 도시한다.

도 22는 본 발명의 일 실시예에 따른 제어기를 액세서리와 페어링하는 프로세스의 예를 도시한다.

도 23은 본 발명의 일 실시예에 따른 문를 잠금해제하기 위한 프로세스의 예를 도시한다.

도 24는 본 발명의 일 실시예에 따른 IP 카메라 액세서리를 위한 동작 환경을 도시한다.

도 25a 및 도 25b는 본 발명의 일 실시예에 따른 IP 카메라 액세서리에 대한 예시적인 서비스 정의들을 도시한다.

도 26a 내지 도 26e는 본 발명의 일 실시예에 따른 도 25에 도시된 서비스들의 특성들에 대한 예시적인 정의들을 도시한다.

도 27a 내지 도 27d는 본 발명의 일 실시예에 따른 IP 카메라 액세서리에 대한 액세서리 정의 레코드의 예를 도시한다.

도 28은 본 발명의 일 실시예에 따른 IP 카메라 액세서리를 제어하기 위한 프로세스의 예를 도시한다.

도 29는 본 발명의 일 실시예에 따른 시작 미디어 세션 요청의 예를 도시한다.

도 30은 본 발명의 일 실시예에 따른 시작 미디어 세션 응답의 예를 도시한다.

도 31은 본 발명의 일 실시예에 따른 종료 미디어 세션 요청의 예를 도시한다.

도 32는 본 발명의 일 실시예에 따른 종료 미디어 세션 응답의 예를 도시한다.

도 33은 본 발명의 일 실시예에 따른 IP 스트리밍 서비스에 대한 예시적인 서비스 정의를 도시한다.

도 34는 본 발명의 일 실시예에 따른 도 33의 IP 스트리밍 서비스에 대한 특성 정의들의 예들을 도시한다.

도 35는 본 발명의 일 실시예에 따른 IP 스트리밍을 위한 프로세스의 예를 도시한다.

도 36은 본 발명의 일 실시예에 따른 제어기의 단순화된 블록도이다.

도 37은 본 발명의 일 실시예에 따른 액세서리의 단순화된 블록도이다.

도 38은 본 발명의 일 실시예에 따른 제어기 아키텍처의 단순화된 블록도이다.

도 39는 본 발명의 일 실시예에 따른 액세서리 아키텍처의 단순화된 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0014] 본 발명의 소정 실시예들은 제어기 디바이스(또는 "제어기")와, 제어되어야 할 임의의 수의 다른 전자 디바이스들(본 명세서에서 "액세서리 디바이스들" 또는 단순히 "액세서리들"로 지칭됨) 사이의 통신을 위한 "균일한" 프로토콜에 관한 것이다. 제어기는 예를 들어, 데스크톱 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터, 휴대 전화, 다른 핸드헬드 또는 착용가능한 컴퓨팅 디바이스와 같은 범용 컴퓨팅 디바이스 상에서, 적절한 실행가능한 프로그램 코드를 범용 컴퓨팅 디바이스에 제공함으로써, 구현될 수 있으며; 대안적으로, 제어기는 특수목적 컴퓨팅 디바이스일 수 있다. 액세서리는 제어기에 의해 제어가능한 임의의 디바이스를 포함할 수 있다. 액세서리들의 예들은 조명 기구, 온도조절기, 문 잠금장치, 자동 문 열림장치(예컨대, 차고문 열림장치), 스틸 또는 비디오 카메라 등을 포함한다. 액세서리들 및 제어기들은 Wi-Fi, 블루투스, 블루투스 LE 등과 같은 표준 전송 프로토콜들을 사용하여 유선 또는 무선 채널들을 통해 서로 통신할 수 있다.

[0015] 예시적인 환경

[0016] 도 1은 본 발명의 일 실시예에 따른 홈 환경(100)을 도시한다. 홈 환경(100)은 환경(100)에 위치된 다양한 액세서리 디바이스들(액세서리들로도 지칭됨)과 통신할 수 있는 제어기(102)를 포함한다. 제어기(102)는, 예를 들어, 데스크톱 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터, 스마트 폰, 착용가능한 컴퓨팅 디바이스, 개인 휴대 정보 단말기, 또는 본 명세서에 기술된 바와 같이 커맨드-및-제어 메시지들을 액세서리들에 전달하고 사용자 인터페이스를 제시하여 사용자가 액세서리들 상에 원하는 동작들을 나타내도록 할 수 있는, 임의의 다른 컴퓨팅 디바이스 또는 디바이스들의 세트를 포함할 수 있다. 일부 실시예들에서, 제어기(102)는 다수의 개별 디바이스들을 사용하여 구현될 수 있다. 예를 들어, 환경(100) 내의 고정된 위치에 설치될 수 있는 액세서리들과 통신하는 기지국, 및 액세서리들에 대한 제어를 달성하기 위해 사용자 인터페이스를 제공하고 기지국과 통신하는 하나 이상의 모바일 원격-제어국(예컨대, 휴대 전화, 태블릿 컴퓨터, 스마트 시계, 안경 등과 같은 핸드헬드 또는 착용가능한 디바이스)이 있을 수 있다.

[0017] 임의의 유형의 액세서리 디바이스가 제어될 수 있다. 액세서리 디바이스들의 예들은 문 잠금장치(104), 차고문 시스템(106), 조명 기구(108), 보안 카메라(110), 및 온도조절기(112)를 포함한다. 일부 경우들에서, 제어기(102)는 액세서리와 직접 통신할 수 있으며; 예를 들어, 제어기(102)는 문 잠금장치(104) 및 차고문 시스템(106)과 직접 통신하는 것으로 도시된다. 다른 경우들에서, 제어기(102)는 중개자를 통해 통신할 수 있다. 예를 들어, 제어기(102)는 무선 네트워크 액세스 포인트(114)를 통해, 액세스 포인트(114)에 의해 제공된 무선 네트워크 상에 있는 액세서리들(108, 110, 112)과 통신하는 것으로 도시된다. 전술한 바와 같이, 일부 실시예들서, 제어기(102)는 기지국을 포함할 수 있고, 기지국 기능성은 액세스 포인트(114)에, 또는 제어되어야 할 액세서리들 중 하나(예컨대, 온도조절기(112))에 통합될 수 있다.

[0018] 다양한 통신 전송들 및 전송들의 조합들이 사용될 수 있으며, 상이한 전송들이 상이한 디바이스들에서 사용될 수 있다. 통신 전송의 일례는, 블루투스 SIG 사(Bluetooth SIG, Inc.)(http://www.bluetooth.com)에 의해 정의 및 공표된 블루투스® 통신 표준들 및 프로토콜들에 부합하는 전송일 수 있으며; 본 명세서에서 사용되는 "블루투스"라는 용어는 일반적으로 블루투스® 통신 표준들 및 프로토콜들을 지칭하고, 본 명세서에서 사용되는 "블루투스 LE"라는 용어는 블루투스® 스마트 통신 표준들 및 프로토콜들을 지칭한다. 블루투스 프로토콜들은 제한된 범위 내에서 디바이스들 사이에 직접적인 지점 대 지점 간 통신을 지원할 수 있다. 통신 전송의 다른 예는, Wi-Fi 얼라이언스(Alliance)®(http://www.wi-fi.org)에 의해 정의 및 공표된 Wi-Fi® 통신 표준들 및 프로토콜들에 부합하는 전송일 수 있으며; 본 명세서에서 사용되는 바와 같이 "Wi-Fi"는 일반적으로 Wi-Fi® 표준들 및 프로토콜들을 지칭한다. Wi-Fi 프로토콜들은 네트워크 상에서 상이한 디바이스들 간의 통신들을 라우

팅하는 중앙 액세스 포인트를 갖는 무선 네트워크를 정의할 수 있다. 네트워크는 예를 들어, TCP 및 HTTP를 포함하는 표준 인터넷 프로토콜 스위트(suite)(IP)를 지원할 수 있다. 블루투스 및 Wi-Fi가 통신 전송들 및 프로토콜들의 예들로서 사용되고; 다른 전송들 및 프로토콜들이 또한 사용될 수 있다는 것이 이해될 것이다. 또한, 무선 통신 전송들이 도시되어 있지만, 유선 전송들이 또한 액세서리들의 일부 또는 전부에 제공될 수 있다. 예를 들면, 전구(108)는 유선 접속에 의해 액세스 포인트(114)에 접속될 수 있고, 제어기(102)는 액세스 포인트(114)에 무선으로 메시지들을 송신함으로써 전구(108)와 통신할 수 있으며, 액세스 포인트(114)는 브리지로서 기능하여, 유선 접속을 통해 메시지들을 전구(108)에 전달할 수 있다. 유선 및 무선 통신의 다른 조합들이 또한 가능하다.

[0019] 또한, 하나의 제어기(102)가 도시되어 있지만, 홈 환경(100)은 다수의 제어기 디바이스들을 가질 수 있다. 예를 들어, 집에 사는 각각의 사람은 액세서리들(104 내지 112)의 일부 또는 전부에 대한 제어기들로서 기능할 수 있는 하나 이상의 개인용 디바이스(예컨대, 휴대 전화, 태블릿, 랩톱, 착용가능한 디바이스)를 가질 수 있다. 상이한 제어기 디바이스들이 액세서리들의 상이한 서브셋들과 통신하도록 구성될 수 있으며; 예를 들어, 아이의 제어기가 온도조절기(112) 상의 설정들을 수정하는 것이 차단되는 반면, 부모의 제어기 디바이스는 설정들을 수정하도록 허용될 수 있다. 이러한 허가들은, 예를 들어, 아래에서 기술되는 페어링 기술들을 사용해 구성되고 제어될 수 있다.

[0020] 본 발명의 소정 실시예들은, 제어기(102)와 같은 제어기들에 의한, 액세서리들(104 내지 112) 중 임의의 것 또는 전부와 같은 하나 이상의 액세서리와의 통신을 용이하게 하는 균일한 액세서리 프로토콜에 관한 것이다. 프로토콜은 액세서리를 서비스들의 집합으로서 모델링하는 단순하고 확장가능한 프레임워크를 제공할 수 있으며, 이때 각 서비스는 특성들의 세트로서 정의되고, 특성들 각각은 임의의 주어진 시간에서 정의된 값을 갖는다. 특성들은 액세서리의 상태의 다양한 미세한 양상들을 표현할 수 있다. 예를 들어, 온도조절기(112)의 경우, 특성들은 전원(온도조절기가 온인지 오프인지 여부), 온도조절기(112)에 의해 측정된 현재 온도, 및 온도조절기(112)가 설정되는 목표 온도를 포함할 수 있다. 서비스들 및 특성들을 사용하는 액세서리 모델들의 예들이 아래에 기술된다.

[0021] 프로토콜은 또한, 제어기들(예컨대, 제어기(102))에 의해 액세서리들(예컨대, 온도조절기(112))에 커맨드-및-제어 메시지들(요청들)을 송신하는 데 사용가능한 메시지 포맷들, 및 액세서리들(예컨대, 온도조절기(112))에 의해 제어기들(예컨대, 제어기(102))에 응답 메시지들을 송신하는 데 사용가능한 메시지 포맷들을 정의할 수 있다. 커맨드-및-제어 메시지들은, 제어기가 액세서리 특성들의 현재 상태를 질의(예컨대, 판독)하도록 하고, 일부 경우들에서 액세서리 특성들을 수정(예컨대, 그에 기록)하도록 할 수 있다. 예를 들어, 온도조절기(112)의 전원 특성을 수정하는 것은 온도조절기(112)를 끄거나 켤 수 있다. 이와 같이, 임의의 유형의 액세서리는, 기능 또는 제조사에 상관없이, 적절한 메시지들을 송신함으로써 제어될 수 있다. 메시지 포맷들은 제어기들 및 액세서리들에 걸쳐 균일할 수 있으며; 예들이 아래에 기술된다. 일부 실시예들에서, 액세서리는 액세서리 정의 레코드를 제어기에 제공할 수 있다. 액세서리 정의 레코드는 액세서리의 모든 액세스가능한 특성들에 관한 완전한 정보를 포함할 수 있다. 제어기는 액세서리와 상호작용하는 방법을 결정하는 데 액세서리 정의 레코드를 사용할 수 있다. 예를 들어, 제어기는 액세서리 정의 레코드로부터의 정보를 사용하여, 액세서리를 동작시키기 위한 사용자 인터페이스를 구성할 뿐만 아니라 액세서리로의 요청 메시지들을 구성할 수 있다.

[0022] 프로토콜은 또한, 상태 변화의 경우에 액세서리(112)(또는 다른 액세서리들)가 선택적으로 제어기(102)에 통지할 수 있게 하는, 통지 메커니즘들을 정의할 수 있다. 예들은, 임의의 특성들이 변화했는지 여부를 알아내기 위해 제어기(102)가 액세서리(예컨대, 액세서리(112))에 질의할 수 있는, 수동적 통지 메커니즘들뿐만 아니라; 특정한 특성이 변화할 때 액세서리(112)(또는 다른 액세서리들)가 선택적으로 하나 이상의 제어기로의 메시지들을 생성하고/하거나 광고를 브로드캐스팅할 수 있는, 능동적인, 광고형 또는 이벤트-기반 통지 메커니즘들을 포함한다. 다수의 통지 메커니즘이 동시에 지원될 수 있고, 제어기는 특정 액세서리, 서비스, 또는 특성에 사용될 통지 메커니즘을 선택할 수 있다. 아래에 예들이 기술된다.

[0023] 일부 실시예들에서, 주어진 액세서리와의 통신은 인가된 제어기들로 제한될 수 있다. 프로토콜은, 제어기(102)로 하여금 액세서리(104)를 제어할 수 있게 하도록 사용자가 의도하는 고도의 확실성을 제공하는 상황들 하에서, 제어기(102)와 주어진 액세서리(예컨대, 문 잠금 액세서리(104)) 사이에 "페어링"을 확립하기 위한 하나 이상의 메커니즘을 명시할 수 있고, 특정 액세서리와 페어링을 확립한 제어기는 그 액세서리에 대해 인가된 것으로 간주될 수 있다. 페어링은, 예를 들어, 단기간(short-term) 키들 및 대역외 공유 비밀(shared secret)을 사용하여 안전한 암호화 프레임워크를 확립함으로써, 확립될 수 있다. 액세서리 및 제어기를 위한 장기간 공개 키들은 이 프레임워크 내에서 교환될 수 있고, 액세서리 및 제어기는 교환된 키들을 지속적으로 저장함으로써,

페어링을 확립할 수 있다. 페어링이 확립된 이후에, 액세서리(104)는 수신된 통신들이 페어링된 제어기(102) 또는 다른 디바이스로부터의 것인지 여부를 검증할 수 있고, 액세서리(104)는 페어링된 제어기(102)로부터의 것이 아닌 임의의 통신들을 거부할 수 있다(그 반대도 마찬가지다). 예를 들어, 이전에 페어링을 확립한 액세서리와 제어기가 재접속할 때, 이들은 (예컨대, 각각이 다른 것의 장기간 공개 키를 소유한다는 것을 증명함으로써) 이전의 페어링을 검증하고 페어-검증된(pair-verified) 세션 내에서의 통신에 사용하기 위한 세션-특정 암호화 키들을 생성할 수 있다. 일부 실시예들에서, 다수의 제어기가 동일한 액세서리와 페어링을 확립할 수 있고, 액세서리는 자신의 페어링된 제어기들 중 임의의 것으로부터의 통신들을 수락하고 그에 응답하면서, 페어링되지 않은 제어기들로부터의 통신들은 거부 또는 무시할 수 있다. 페어링 프로세스들의 예들이 아래에 기술된다.

[0024] 홈 환경(100)은 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 본 발명의 실시예들은, 사용자가 제어기 디바이스를 사용하여 하나 이상의 액세서리 디바이스를 제어하고자 하는 임의의 환경에서 구현될 수 있으며, 이는 집, 자동차 또는 다른 차량, 사무실 건물, 다수의 건물을 갖는 캠퍼스(예컨대, 대학 또는 기업 캠퍼스) 등을 포함하지만 이로 제한되지 않는다. 제어기는 하나 이상의 다른 디바이스(액세서리)를 제어하는 데 사용되는 임의의 디바이스일 수 있고, 액세서리는 그것의 동작들의 일부 또는 전부가 제어기에 의해 제어되도록 하는 임의의 디바이스일 수 있다. 제어기(102)는 제어기에서 구현되거나 그에 포함되는 것으로서 본 명세서에서 기술된 특징(feature)들 중 임의의 것 또는 전부를 구현하거나 포함할 수 있으며, 액세서리들(104 내지 112)과 같은 액세서리들은 액세서리에서 구현되거나 그에 포함되는 것으로서 본 명세서에서 기술된 특징들 중 임의의 것 또는 전부를 구현하거나 포함할 수 있다.

[0025] 일부 실시예들에서, 제어기(102)는 원격 위치에서(예컨대, 전세계 어디에서나) 액세서리(예컨대, 액세서리(108))와 통신할 수 있다. 예를 들어, 원격 환경에 위치해 있는 동안, 제어기(102)는, (예컨대, 액세서리(108)와 로컬로 통신할 수 있는, 환경(100)에 위치해 있는 액세스 포인트(114)와 통신함으로써) 메시지들을 액세서리(108)에 중계할 수 있는 능력을 갖는 서버와, 광역 네트워크(예컨대, 인터넷)를 통해 통신할 수 있다. 제어기(102)와 액세서리(108) 사이의 통신의 내용은 서버에 불투명할 수 있으며; 예를 들어, 제어기(102) 및 액세서리(108)는 메시지들이 암호화되는 보안 통신 세션(예를 들어, 본 명세서에서 기술된 바와 같은 페어-검증된 세션)을 확립할 수 있고, 서버는 그 내용에 관해 비인식을 유지하면서(remaining agnostic) 암호화된 데이터를 단순히 전달할 수 있다. 따라서, 액세서리들은 (예컨대, 액세서리에 대한 직접적인 통신 경로를 확립할 수 있는 제어기에 의해) 로컬로 동작될 수 있거나, 또는 (예컨대, 중계 서버 등을 통해 간접적으로 통신하는 제어기에 의해) 원격으로 동작될 수 있다.

[0026] 예시적인 액세서리 모델

[0027] 일부 실시예들에서, 균일한 액세서리 프로토콜은 임의의 액세서리를 "서비스들"의 집합으로서 모델링하기 위한 균일한 프레임워크 및 신택스를 제공할 수 있다. 본 명세서에서 사용되는 "서비스"는 액세서리 디바이스의 특징, 기능, 또는 동작을 달성하기 위한 데이터 및 연관된 거동(behavior)들의 집합(또는 그것의 일부분)을 지칭할 수 있다. 각각의 서비스는 "특성들"의 집합으로서 모델링될 수 있으며, 특성들 각각은 액세서리의 미세한 데이터 요소 또는 거동(액세서리 상태의 요소로도 지칭됨)을 표현한다. 후술되는 바와 같이, 액세서리는, 액세서리의 서비스들 및 특성들을 정의하는 구조화된 데이터 객체일 수 있는 액세서리 정의 레코드를 제어기에 제공함으로써, 자신을 제어기에 설명할 수 있다. 구조화된 데이터 객체는 예를 들어, JSON(JavaScript Object Notation), 블루투스 LE GATT(Generic Attribute Profile), 또는 구조화된 데이터를 표현하고 전달하기 위한 다른 기술들 및 포맷들을 사용하여, 다양한 특정 포맷으로 표현될 수 있다. 후술되는 바와 같이, 제어기는 액세서리를 제어하는 방법을 결정하기 위해 액세서리 정의 레코드를 사용할 수 있다.

[0028] 예를 들어, 도 1의 온도조절기 액세서리(112)는 온도를 어떤 목표에 가깝게 유지하기 위해 일정 구역(예컨대, 방이나 건물 또는 건물의 부분)의 온도를 조절하는 기능을 수행하는 컴포넌트를 포함할 수 있다. 액세서리(112)를 모델링하기 위한 목적으로, 온도 조절은 "온도조절기(thermostat)" 서비스로서 식별될 수 있다. 온도조절기 서비스의 특성들은, 온도조절기가 온인지 오프인지의 여부; 현재 설정된 목표 온도; 현재 측정된 온도; 및 온도조절기에 의해 조절되는 시스템이 현재 (측정된 온도를 더 높은 목표 온도를 향해 구동하기 위해 난방장치를 켤 수 있는) 난방 모드에 있는지 (측정된 온도를 더 낮은 목표 온도를 향해 구동하기 위해 냉방장치를 켤 수 있는) 냉방 모드에 있는지의 여부를 포함할 수 있다. 보다 복잡한 예에서, 온도조절기는, 예를 들어, 하루 중 시간 및/또는 요일에 따라 상이한 목표 온도를 선택하도록 프로그램될 수 있고, 그리고 그러한 경우들에서 특성들은, 시간 범위들 및 각각의 시간 범위에 대한 연관된 목표 온도의 세트와 같은, 프로그래밍 특징들을 포

함할 수 있다.

[0029] 일부 경우들에서, 액세서리는 다수의 서비스를 제공할 수 있다. 예를 들어, 차고문 액세서리(106)는, 문을 열고 닫을 수 있고 또한 예를 들어, 차고 내부를 조명하도록 제어될 수 있는 조명을 갖는, 자동 차고문 열림장치를 사용하여 구현될 수 있다. 따라서, 차고문 액세서리(106)의 정의는 차고문을 열고 닫는 기능을 달성하는 "문-열림장치(door-opener)" 서비스, 및 조명을 켜거나 끄는 (상이한) 기능을 달성하는 "전구(light-bulb)" 서비스를 포함할 수 있다. 문-열림장치 서비스의 특성들은 문의 현재 상태(예를 들어, 열림, 닫힘, 여는 중, 닫는 중), 문이 잠겨 있는지 여부(예를 들어, 열림장치가 문을 여는 것을 방지하기 위함), 및 문이 가로막혀 있는지의 여부(예를 들어, 문-열림장치 시스템의 장애물 센서가 문이 닫히는 것을 방지하는 장애물을 검출했는지의 여부)를 포함할 수 있다. 전구 서비스의 특성들은, 조명이 온인지 오프인지의 여부 및 현재 밝기 레벨(조명이 가변-밝기 제어부를 갖는 경우)을 포함할 수 있다.

[0030] 임의의 액세서리가 서비스들의 집합으로서 모델링될 수 있고, 동일한 환경 내의 상이한 액세서리들은 동일하거나 유사한 서비스들의 일부 또는 전부를 포함할 수 있다는 것이 이해되어야 한다. 예를 들어, 홈 환경(100)과 같은 환경은 상이한 위치들에 있는 다수의 전구들(예를 들어, 각각의 방에 있는 조명들), 다수의 문 잠금장치들(예를 들어, 각각의 외부 문의 잠금장치, 내부 문들의 잠금장치들) 등을 가질 수 있다. 일부 실시예들에서, 균일한 액세서리 프로토콜은, 각각의 액세서리 및 서비스가 고유하게 식별될 수 있도록 이름 공간(namespace)을 정의하여, 상이한 액세서리들 상의 동일한 서비스의 인스턴스들, 또는 동일한 액세서리 상의 유사한 서비스들의 다수의 인스턴스들이 쉽게 구별될 수 있게 함으로써, 그러한 중첩을 명시적으로 허용한다.

[0031] 일부 실시예들에서, 균일한 액세서리 프로토콜은, 자주 사용될 것으로 예상되는 그리고/또는 상이한 액세서리 유형들의 범위에 걸쳐 존재할 것으로 예상되는 특성들을 포함할 수 있는, "핵심(core)" 특성들의 세트를 정의할 수 있다. 특성들의 세트는 확장가능하게 만들어질 수 있으며; 예를 들어, 액세서리 제조사들은 제조사-특정 특성들(본 명세서에서 "확장(extension)" 특성들로도 지칭됨)을 정의하도록 허용될 수 있다. 따라서, 액세서리들은 핵심 특성들로 제한되지 않는다. 그러나, 적용가능한 경우 핵심 특성들을 사용하는 것은, 시스템 설계 및 동작을 용이하게 할 수 있다. 예를 들어, 제어기의 시스템 소프트웨어는 핵심 특성들의 속성들을 정의하는 코드를 포함할 수 있고, 핵심 특성들만을 사용하는 액세서리는 그것의 특성들 및 이들의 현재 값들을 식별함으로써 자신을 제어기에 설명할 수 있다.

[0032] 도 2a 내지 도 2d는 본 발명의 일 실시예에 따라 정의될 수 있는 표준 특성들(201 내지 229)의 일부 예들을 도시한다. 각각의 특성(201 내지 229)은, 특성의 주어진 인스턴스가 속하는 서비스를 제공하는 액세서리의 상태의 일부 양상(또는 액세서리의 부분)을 표현하는 값(도 2a 내지 도 2d에 도시되지 않음)을 가질 수 있다. 이 예에서, 각각의 특성(201 내지 229)은 타입, 허가들(permissions), 및 포맷을 갖는 것으로서 설명된다. 포맷에 따라, 추가적인 정보(예컨대, 최소값, 최대값, 단계 크기(step size), 단위, 또는 유효한 값들의 열거된 목록(enumerated list))도 포함된다.

[0033] 특성의 "타입(type)"은 그 특성에 할당된 고유 이름 또는 식별자일 수 있다(예를 들어, 문자열(character string)). 이러한 예에서, 역 도메인 이름 규약(reverse domain name convention)은 타입들을 할당하는 데 사용되며, 이는 액세서리 제조사들에 의한 확장 특성들의 정의를 용이하게 할 수 있다. 예를 들어, 도 2a 내지 도 2d에 도시된 특성 정의들을 포함하는 균일한 액세서리 프로토콜의 공표자가 "proto.com"으로 불리는 인터넷 도메인의 소유자인 경우, 모든 핵심 특성 타입들은 스트링 "com.proto.ch."로 시작한 다음에, "on", "brightness" 등과 같은 특성-특정 이름이 이어질 수 있다. 다른 명명 규약들이 사용될 수 있다.

[0034] 일부 실시예들에서, 명명된 타입에 추가하여 또는 그 대신에, 각각의 특성은 고유 숫자 식별자(도 2a 내지 도 2d에 도시되지 않음)가 할당될 수 있다. 예를 들어, 고유 숫자 식별자는 IETF(Internet Engineering Task Force, 국제 인터넷 표준화 기구) RFC 4122에 부합하는 36개의 16진수로 구성된 UUID일 수 있다(본 명세서에서 참조된 모든 "IETF RFC" 문헌들은 http://ietf.org/rfc.html를 통해 액세스될 수 있음). 균일한 액세서리 프로토콜은 모든 핵심 특성들에 대해 공통적인 "기본(base)" UUID(예를 들어, 마지막 28개의 16진수)를 정의하고, 고유한 "짧은(short)" UUID(예를 들어, 처음 8개의 16진수)를 각각의 특성에 할당할 수 있으며; 임의의 번호 체계가 사용될 수 있다. UUID들의 사용은, 특히 UUID들을 절단하기(truncate) 위한 규약과 조합되어, 제어기 또는 액세서리로 하여금, 문자열을 사용하는 것에 비하여 더 적은 양의 전송된 데이터를 사용하여 관심있는 특성을 명시하도록 허용할 수 있다. 예를 들어, 일부 실시예들에서, 절단 규약은 36-자리의 UUID를 단지 2개 또는 3개의 16진수들로 줄일 수 있다.

[0035] 특성의 "허가들"은, 제어기가 특성과 상호작용하도록 허용되는 방식을 나타낼 수 있다(예를 들어, 특성을 질의

하거나 수정함). 일부 실시예들에서, 허가들은 스트링들의 어레이로서 표현될 수 있으며, 여기서 각각의 스트링은 특정 상호작용 방식에 대응한다. 스트링이 존재하는 경우, 대응하는 상호작용 방식이 허용되고; 스트링이 존재하지 않는 경우, 상호작용은 허용되지 않는다. 정의될 수 있는 허가 스트링들의 예들이 표 1에 도시되어 있다.

[0036] [표 1]

허가 스트링	설명
"Paired Read" ("PR")	페어링된 제어기가 (예를 들어, 후술되는 바와 같이 HTTP GET 요청을 사용해) 특성을 판독할 수 있음
"Paired Write" ("PW")	페어링된 제어기가 (예를 들어, 후술되는 바와 같이 HTTP PUT 요청을 사용해) 특성에 기록할 수 있음
Unpaired Read ("UR")	페어링되지 않은 제어기가 (예를 들어, 후술되는 바와 같이 HTTP GET 요청을 사용해) 특성을 판독할 수 있음
Unpaired Write ("UW")	페어링되지 않은 제어기가 (예를 들어, 후술되는 바와 같이 HTTP PUT 요청을 사용해) 특성에 기록할 수 있음
Admin Read ("AR")	관리자 허가를 갖는 페어링된 제어기(아래 참조)가 특성을 판독할 수 있음
Admin Write ("AW")	관리자 허가를 갖는 페어링된 제어기(아래 참조)가 특성에 기록할 수 있음

[0037]

[0038] 일부 실시예들에서, 특성에 대한 판독 허가는, 제어기들이 액세서리 상태의 대응하는 양상을 알아내는 것이 바람직한 경우에 승인되고, 기록 허가는, 제어기들이 액세서리 상태의 대응하는 양상에 변화를 일으킬 수 있는 것이 바람직한 경우에 승인된다. 따라서, 예를 들어, 특성들은 액세서리가 직접 제어할 수 있는 조건에 관련된 특성들(예를 들어, 목표 온도 특성(210))은 판독 및 기록 허가를 가질 수 있지만, 액세서리가 직접 제어할 수 없는 조건에 관련된 특성들(예를 들어, 현재 온도 특성(209))은 판독 허가만 가질 수 있다.

[0039] 각각의 특성(201 내지 229)은 액세서리 상태의 대응하는 속성 또는 양상을 반영하는 값을 가질 수 있다. 도 2a 내지 도 2d는 각각의 특성(201 내지 229)의 값에 대한 포맷을 명시한다. 본 명세서에서 사용되는 바와 같이, <boolean> 포맷은 특성이 참(true) 및 거짓(false)의 값들을 취하는 것을 나타내고; <int> 포맷은 특성이 부호 붙임 정수(signed integer) 값을 취하는 것을 나타내고; <float>은 특성이 부호 붙임 부동 소수점 값들을 취하는 것을 나타내고; <string>은 특성이 예컨대, EUC-KR 인코딩을 사용해 문자열 값을 취하는 것을 나타내고; <date>는 특성이 날짜 값(예컨대, ISO 8601 포맷의 EUC-KR 날짜 스트링)을 취하는 것을 나타내고; <data>는 특성이 데이터 블랍(data blob)(즉, 프로토콜이 그것의 콘텐츠에 관해 비인식할 수 있는, 데이터의 블록)을 저장하는 데 사용될 수 있음을 나타낸다. <enum> 포맷은 특성이 특정 조건들을 표현하는 값들의 정의된 세트 중 하나를 취하는 것을 나타내고, 가능한 값들은 그 특성에 대한 "유효한 값들(valid values)"로서 열거된다. 일부 실시예들에서, 열거형 값(enumerated value) 포맷은 각각의 유효한 값에 상이한 정수를 할당함으로써 구현될 수 있다. <tlv> 포맷은 패킷 타입-길이-값(packed type-length-value) 데이터 타입을 나타내며, 여기서 첫 번째 바이트는 타입을 나타내고, 두 번째 바이트는 길이(N 바이트)를 나타내며, 나머지 N 바이트는 값을 포함한다. 일부 실시예들에서, 다른 포맷들이 사용될 수 있으며; 예들은, 본 명세서에서 <object>로서 나타낸 데이터 객체 포맷(데이터 객체는 또한 키-값(key-value) 포맷을 사용해 정의될 수 있음) 및 배열 포맷(이것은 다른 포맷들 중 임의의 것으로 된 값들의 고정된-길이 또는 가변-길이의 배열일 수 있음)을 포함한다. 일부 실시예들에서, 균일한 액세서리 프로토콜은 특성들에 대한 허용된 포맷들의 폐쇄형 유니버스(closed universe)를 명시하여, 모든 특성들(확장 특성들 포함)의 값들이 허용된 포맷들 중 하나로 표현되도록 할 수 있다.

[0040] 특성들의 값은 액세서리 상태의 속성 또는 양상을 나타낼 수 있다. 값은 표현되고 있는 정보에 적합한 포맷으로 명시될 수 있다. 일부 경우들에서, 특성 정의는 값들의 범위를 명시할 수 있다. 예를 들어, "최대(Max)"(또는 "최대값(maxValue)")는 상한을 나타낼 수 있고, "최소(Min)"(또는 "최소값(minValue)")은 하한을 나타낼 수 있고, "단계(Step)"(또는 "단계크기(stepSize)")은 이산 값을 취하는 특성들에 대한 최소 증분을 나타낼 수 있다. "단위(Units)"는 값이 측정되는 특정 단위들을 나타내도록 정의될 수 있으며; 이 정보는 제어기에 의한

값의 해석을 용이하게 할 수 있다.

[0041] 예를 들어, "온(on)" 특성(201)은, 액세서리의 전원이 온인 경우 "true"이고, 액세서리의 전원이 오프인 경우 "false"의 부울 값(Boolean value)을 가질 수 있다. 이 특성은 예를 들어, 전구, 조명 스위치, 또는 온 및 오프 상태들을 갖는(그리고 오프 상태에 있는 동안 제어기와 통신할 수 있는) 다른 액세서리와 관련하여 사용될 수 있다. 다른 예로서, "사용중인 콘센트(outlet in use)" 특성(202)은 전원 콘센트를 갖는 액세서리에 사용될 수 있으며, 부울 값은 전원 플러그가 전원 콘센트에 연결되어 있는지 여부를 나타낼 수 있다. 이 경우, 액세서리는 물리적으로 전원 플러그를 삽입 또는 제거할 수 없다고 가정되며, 따라서, 특성은 판독-전용 허가를 갖는다.

[0042] 액세서리 모델을 사용하는 액세서리 제어의 간단한 예로서, 전원 콘센트 액세서리가 콘센트로의 전력의 흐름을 시작 또는 중지할 수 있는 제어 스위치를 갖는다고 가정한다. 전원 콘센트 액세서리는 "온" 특성(201) 및 "사용중인 콘센트" 특성(202)을 갖는 서비스로서 모델링될 수 있다. 제어기는 전원 플러그가 콘센트에 연결되어 있는지 여부를 결정하기 위해 "사용중인 콘센트" 특성(202)을 판독하고, 이어서 전원 플러그가 연결되어 있는지 여부에 따라 콘센트로의 전력을 인에이블 또는 디스에이블하기 위해 "온" 특성(201)에 기록할 수 있다.

[0043] 밝기(brightness) 특성(203), 색상(hue) 특성(204), 및 채도(saturation) 특성(205)이, 예를 들어, 광원을 제공하는 액세서리와 관련하여 사용될 수 있다. 밝기 특성(203)는 액세서리에 의해 지원되는 최대 밝기의 백분율로 밝기 레벨을 나타내기 위해 (도 2a의 "최소, 최대, 단계" 필드에 나타낸 바와 같이) 0 내지 100의 범위의 정수 값을 가질 수 있다. 백분율 단위로 명시된 모든 특성들과 마찬가지로, 액세서리들은 디바이스 상태의 대응하는 양상(이 경우에서는, 밝기)에 대해 정확히 100개의 별개의 설정들을 지원하도록 요구되지 않는다는 것이 이해되어야 하며; 그 대신에, 액세서리는 이용가능한 설정에 주어진 백분율 값을 매핑할 수 있다. 색상 특성(204)은 0 내지 360도(degree) 범위에서 색상을 명시하는 부동 소수점 값을 가질 수 있으며; 도 단위로 된 값의 물리적 색상으로의 변환은 표준색 모델링 관행들을 따를 수 있다. 채도 특성(205)은 원하는 색상 채도를 최대의 백분율로서 정의하기 위해 0 내지 100 범위에서 부동 소수점 값을 가질 수 있다. 제어기는 현재 설정들을 결정하기 위해 이들 특성을 판독하고, 설정들을 변경하기 위해 특성들에 기록할 수 있다. 다른 색상 모델들이 또한 지원될 수 있다(예를 들면, CIE 1931 색 공간, 색 온도 등).

[0044] 오디오 피드백(audio feedback) 특성(206)이 사용될 수 있으며, 여기서 액세서리는 사용자 입력 또는 액세서리에서 검출된 다른 이벤트들에 응답하여 선택적 오디오 피드백(예를 들면, 삐 소리(beeping))을 갖는다. 오디오 피드백은 이 특성에 부울 값을 기록함으로써 인에이블 또는 디스에이블될 수 있다. 출력 음량(output volume) 특성(207)은 소리를 생성하는 액세서리 상에서 출력 음량을 판독 또는 설정하는 데 사용될 수 있으며; 값은 액세서리가 생성할 수 있는 최대 음량의 백분율을 나타낼 수 있다.

[0045] 로그(logs) 특성(208)은 액세서리가 활동의 타임스탬프된 로그를 유지하는 경우에 사용될 수 있다. 로그의 구조는 액세서리 제조사에 의해 정의되거나, 균일한 액세서리 프로토콜의 공표자에 의해 특정 유형의 액세서리에 대해 명시될 수 있다. 이 예에서, 제어기는 특성(208)을 판독함으로써 액세서리의 로그를 획득할 수 있지만 특성(208)에 기록하지는 않는데, 이는 로그를 업데이트하는 것이 제어기의 역할이 아니기 때문이다. 액세서리는 그것의 루틴 동작의 일부로서 로그에 제어기 상호작용들의 레코드들을 추가할 수 있고, 제어기는 나머지 로그와 함께 임의의 그러한 레코드들을 수신할 수 있다는 것이 이해될 것이다.

[0046] 도 2b는 온도조절기 액세서리에 관련된 특성들의 예들을 도시하며, 이는 환경(예를 들면, 집 또는 방) 내의 현재 온도를 모니터링할 수 있는 임의의 액세서리를 포함하고, 온도를 목표 온도 쪽으로 조정하기 위해 난방 및/또는 냉방 시스템을 제어할 수 있다. 현재 온도(current temperature) 특성(209)은 액세서리에 의해 측정된 현재 온도를 결정하기 위해 제어기에 의해 판독될 수 있다. 목표 온도(target temperature) 특성(210)은 온도조절기 액세서리의 목표 온도 설정을 결정하기 위해 제어기에 의해 판독될 수 있고, 또한 목표 온도 설정을 변경하기 위해 제어기에 의해 기록될 수 있다. 이 예에서, 온도는 섭씨로 명시되며; 다른 스케일들(예를 들어, 화씨, 켈빈)이 대체될 수 있다. 어떤 스케일이 프로토콜에 의해 명시되는지에 관계없이, 제어기 또는 액세서리는 항상, 내부 사용 및/또는 사용자에게 표시하기 위한 상이한 스케일로 변환할 수 있다. 예를 들어, 온도 단위(temperature units) 특성(211)은 사용자에게 온도를 표시할 때 액세서리가 사용해야 하는 단위를 나타내기 위해 제어기에 의해 판독 또는 기록될 수 있다. 또한, 온도 특성들(209, 210)은 허용된 값들의 범위를 명시하지만, 다른 범위들이 사용될 수 있거나, 또는 범위는 명시되지 않은 채 남아 있을 수 있다.

[0047] 일부 온도조절기들은 난방 및 냉방 둘 다를 제어하도록 동작가능할 수 있다. 따라서, 현재 난방/냉방 상태(current heat/cool status) 특성(212)은 온도조절기가 현재 난방 중인지(목표 온도 쪽으로 능동적으로 가온),

냉방 중인지(목표 온도 쪽으로 능동적으로 냉방), 또는 오프인지를 결정하기 위해 판독될 수 있다. 제어기는 언제 난방하거나 냉방할지를 결정하지 않기 때문에, 기록 허가는 제공되지 않는다. 온도조절기의 동작 모드는 목표 난방/냉방 모드(target heat/cool mode) 특성(213)에 기록함으로써 제어될 수 있다. 이 예에서, 모드 특성(213)에 대한 유효한 값들은 난방 모드(온도조절기가 목표 온도 쪽으로 환경을 가온함), 냉방 모드(온도조절기가 목표 온도 쪽으로 환경을 냉방함), 자동 모드(환경 조건들에 따라, 온도조절기가 난방 모드 또는 냉방 모드를 동적으로 선택할 수 있음), 및 오프를 포함한다. 자동 모드가 선택되면, 온도조절기를 난방 또는 냉방 모드로 전환시키기 위한 온도 임계치들을 명시하는 것이 바람직할 수 있다. 예를 들어, 냉방 임계치 온도(cooling threshold temperature) 특성(214)은 냉방 임계치 온도를 설정하기 위해 제어기에 의해 기록되어서, 현재 온도가 냉방 임계치 온도를 초과하는 경우 온도조절기가 냉방 모드를 인에이블하도록 할 수 있고, 난방 임계치 온도(heating threshold temperature) 특성(215)은 난방 임계치 온도를 설정하기 위해 제어기에 의해 기록되어서, 현재 온도가 난방 임계치 온도 미만으로 떨어지는 경우 온도조절기가 난방 모드를 인에이블하도록 할 수 있다. 도 2b에 도시된 바와 같이, 냉방 온도 임계치 특성(214) 및 난방 온도 임계치 특성(215)은, 서로, 그리고 목표 온도 특성(210)과 상이한 상한 및 하한을 가질 수 있지만 그럴 필요는 없다.

[0048] 도 2c는 문 열림장치들(예컨대, 차고문 열림장치, 또는 문을 열고/열거나 닫는 다른 자동화 메커니즘) 및 문 잠금장치들에 관련된 특성들의 예들을 도시한다. 현재 문 상태(current door state) 특성(216)은 문이 현재 열려 있는지, 닫혀 있는지, 열거나 닫는 중인지, 또는 중지되어 있는지(예를 들어, 완전히 열려 있거나 완견히 닫혀 있지 않지만 움직이지는 않음)를 결정하기 위해 제어기에 의해 판독될 수 있다. 목표 문 상태(target door state) 특성(217)은 문이 열리거나 닫혀야 하는지를 나타내기 위해 제어기에 의해 기록될 수 있다. 제어기가 현재 문 상태 특성(216)과 매칭되지 않는 목표 문 상태 특성(217)에 값을 기록하는 경우, 액세서리는 현재 상태를 목표와 매칭되도록 변경하기 위해 문-열림장치 메커니즘을 구동함으로써 응답할 수 있다. 예를 들어, 문이 열려 있고(현재 문 상태 특성(216)이 "open" 값을 가짐) 제어기가 목표 문 상태 특성(217)에 "closed"의 값을 기록하는 경우, 액세서리는 문을 닫기 위해 문-열림장치를 구동할 수 있다. 문-열림장치의 구동 시, 액세서리는 현재 문 상태 특성(216)을 "closing"으로 업데이트할 수 있고, 문이 완전히 닫히면, 액세서리는 추가로 현재 문 상태 특성(216)을, 목표 상태와 매칭되는 "closed"로 업데이트할 수 있다. 제어기는 목표 문-상태 특성(217)을 판독함으로써 언제든지 문의 현재 상태를 알 수 있다.

[0049] 일부 실시예들에서, 문-열림장치 액세서리는 제어기에 추가적인 정보를 제공할 수 있다. 예를 들어, 움직임 검출(motion detected) 특성(218)은 액세서리가 문 주변에서 움직임(예를 들어, 문에서 노크함)을 검출했는지 여부를 나타내는 데 사용될 수 있다. 장애물 검출(obstruction-detected) 특성(219)은 액세서리가 문의 이동을 방지할 수 있는 장애물을 검출했는지 여부를 나타내는 데 사용될 수 있다. 제어기는 이들 특성을 판독함으로써 이 정보를 획득할 수 있다. 일부 실시예들에서, 액세서리는 이들 특성의 변화가 검출되는 경우(예를 들면, 문이 막혔을 경우 또는 움직임이 검출되는 경우) 제어기에 통지를 송신할 수 있다.

[0050] 문를 잠그는 것은 문을 열거나 닫는 것과 별도로 처리될 수 있다. 예를 들어, "잠금 메커니즘(lock mechanism)" 액세서리는 데드볼트, 자기 잠금장치(magnetic lock), 또는 문이 열리는 것을 방지하기 위해 체결될 수 있는 임의의 다른 물리적 메커니즘을 제어하는 임의의 디바이스에 대해 구현될 수 있다. 잠금 메커니즘 현재 상태(lock mechanism current state) 특성(220)은 잠금 메커니즘이 현재 고정되어 있지 않거나(잠금해제), 고정되어 있거나(잠금), 끼이거나(jammed)(잠금 또는 잠금해제될 수 없는), 또는 알려지지 않은 상태인지를 결정하기 위해 제어기에 의해 판독될 수 있다. 잠금 메커니즘 목표 상태(lock mechanism target state) 특성(221)은 문의 잠금 또는 잠금해제를 요청하기 위해 제어기에 의해 기록될 수 있다. 잠금 메커니즘 마지막 행동(lock mechanism last action) 특성(222)은 잠금장치 상에서 수행된 마지막 알려진 행동을 결정하기 위해 제어기에 의해 판독될 수 있다. 다양한 레벨의 세부 사항이 지원될 수 있다. 예를 들어, 일 실시예에서, 유효 값들은, (1) 물리적 이동을 이용해 내부로부터 고정됨(예를 들어, 사용자는 물리적으로 데드볼트 레버를 이동); (2) 물리적 이동을 이용해 외부로부터 고정됨; (3) 키패드를 이용하여 고정됨; (4) 원격으로 고정됨(예를 들어, 제어기로부터의 요청에 기초하여); (5) 타임아웃 조건에 기초하여 고정됨; (6) 물리적 이동을 이용해 내부로부터 고정해제됨(unsecured); (7) 물리적 이동을 이용해 외부로부터 고정해제됨; (8) 키패드를 이용해 고정해제됨; 및 (9) 원격으로 고정해제됨을 포함할 수 있다. 유효 값들의 다른 조합들이 또한 원하는 정보의 세분성에 따라 정의될 수 있다.

[0051] 추가적인 특성들이 잠금 메커니즘을 관리하는 것과 연관될 수 있다. 예를 들어, 잠금 관리 자동 타임아웃(lock management auto timeout) 특성(223)은, 액세서리가 그 후에 자동으로 잠금장치를 재잠금하는 타임아웃 기간을 설정하는 데 사용될 수 있다. 타임아웃 기간은 예를 들어, 잠금장치가 잠금해제될 때 시작될 수 있다. 일부

실시예들에서, 타임아웃 기간의 지속시간은 특성의 숫자 값(예컨대, 초 단위)일 수 있으며, 0의 값은 타임아웃 기간이 없다는 것을 나타내기 위해 사용될 수 있다(즉, 잠금장치는 그것을 잠그기 위한 특정 행동이 취해질 때까지 잠금해제 상태를 유지할 수 있다). 잠금 관리 제어 포인트(lock management control point) 특성(224)은 예를 들어, 기능을 식별하는 값을 특성(224)에 기록함으로써, 잠금장치에 관련된 특정 기능들을 호출하는 데 사용될 수 있다. 호출될 수 있는 기능들의 예들로는, 잠금 행위의 로그를 판독하는 것(이는 액세서리가 로그 특성(208)에 대한 값을 반환하는 결과를 발생시킬 수 있음), 로그를 소거하는 것, 잠금장치에서 시간을 설정하는 것 등이 포함된다. 다른 예로서, 벤더는, 하루 중 특정 시간들 사이에서만 잠금장치의 원격 열림을 허용하는 것과 같은, 잠금장치의 사용에 관한 다양한 정책들을 사용자들이 설정할 수 있게 하는 것을 원할 수 있다. 일부 실시예들에서, 제어기는 잠금 관리 제어 포인트 특성(224)에 기록함으로써 잠금장치 액세서리에 이들 정책을 제공할 수 있다. 일부 실시예들에서, 균일한 액세서리 프로토콜은 잠금 관리 제어 포인트 특성(224)에 기록되는 데이터의 콘텐츠를 명시하지는 않지만, 데이터가 TLV 포맷 또는 다른 포맷으로 제공되어야 함을 명시하여, 데이터가 균일한 액세서리 프로토콜을 사용해 제어기로부터 액세서리에 쉽게 전달되고, 데이터가 벤더-특정 방식으로 액세서리에 의해 해석되게 할 수 있다.

[0052] 도 2d는 핵심 특성들의 추가적인 예들을 도시한다. 회전 방향(rotation direction) 특성(225)은 팬(또는 회전 요소를 갖는 임의의 다른 액세서리)을 시계 방향 또는 반시계 방향 중 어느 한 방향으로 회전시키도록 제어하기 위해 사용될 수 있다. 회전 속도(rotation speed) 특성(226)은 팬(또는 다른 회전 요소)의 회전 속도를 제어하는 데 사용될 수 있으며, 이때 속도는 최대 속도의 백분율로 나타낸다.

[0053] 이름(name) 특성(227)은 액세서리 또는 서비스에 인간-판독가능한 이름을 할당하기 위해 사용될 수 있다. 이름은 문자열로(예컨대, EUC-KR 포맷으로) 표현될 수 있다.

[0054] 관리자-전용(administrator-only) 액세스 특성(228)은 액세서리 또는 서비스에 대한 액세스를, 액세서리에 대한 관리자로서(즉, 관리자 허가를 갖는) 확립된 제어기로 제한하기 위해 사용될 수 있다. 제어기를 관리자로서 확립하기 위한 기술들의 예들이 아래에서 기술된다. 일부 실시예들에서, 액세서리의 관리자로서 확립된 제어기만이 특성(228)에 기록할 수 있다.

[0055] 버전(version) 특성(229)은 액세서리 또는 서비스에 관한 버전 정보를 제공하는 데 사용될 수 있다.

[0056] 도 2a 내지 도 2d에 도시된 특성들은 예들로서 제공된다는 것이 이해되어야 한다. 임의의 수의 핵심 특성들이 정의될 수 있다. 예를 들어, 실내 환경의 다른 제어가능한 양상들에 관련된 특성들(예를 들어, 상대 습도)이 정의될 수 있다. 균일한 액세서리 프로토콜의 일부로서 정의된 특성들의 특정 조합이 원하는 대로 변경될 수 있다. 일부 실시예들에서, 핵심 특성으로서 정의되지 않은 특성들은 여기에 설명된 기술들을 이용하여 제3자(예를 들어, 액세서리 제조사)에 의해 확장 특성들로서 정의될 수 있다.

[0057] 일부 실시예들에서, 특성들의 세트는 확장가능하다. 액세서리는 특성에 대해 속성들(또는 디스크립터들)의 세트를 제공함으로써 새로운 특성("확장" 특성으로도 지칭됨)을 정의할 수 있다. 도 2e는 본 발명의 일 실시예에 따른 특성의 정의가능한 속성들의 세트를 도시한다. 핵심 특성들에 대해, 이들 속성은 (예를 들어, 도 2a 내지 도 2d에 도시된 바와 같은) 프로토콜에 의해 정의될 수 있다. 일부 실시예들에서, 액세서리는 핵심 특성의 속성들 중 일부를 재정의하고 그렇게 함으로써 그 특성에 대해 정의된 디폴트 속성들을 무시할(override) 수 있다. 일부 실시예들에서, 확장 특성은 액세서리 정의 레코드 내에서 이들 속성을 제공함으로써 정의될 수 있다. 아래에 예들이 기술된다.

[0058] "타입(type)"(230)은 전술한 바와 같이 예를 들어, 역 도메인 이름 규약을 사용해 특성의 타입을 식별하는 스트링일 수 있다. 일부 실시예들에서, 숫자 식별자(예컨대, 전술한 바와 같은 UUID)는, 스트링에 추가하여 또는 그 대신에 타입 식별자로서 사용될 수 있다.

[0059] "허가(permissions)"(231)는 제어기들에 허용되는 액세스의 타입을 식별하는 스트링들의 배열(예를 들어, 위의 표 1에서의 허가들 중 일부 또는 전부)일 수 있다.

[0060] "통지모드(notificationMode)"(232)는, 특성에 대한 변화들을 제어기가 어떻게 통지받을지를 나타내는 데 사용되는, 스트링들의 배열일 수 있다. 일부 실시예들에서, 제어기는 특정 통지 모드에 가입하기 위해 이 속성에 기록할 수 있다. 표 2는 지원될 수 있는 통지 모드들의 일부 예들을 열거한다. 이들 각각의 통지 모드의 동작이 아래에 기술된다.

[0061] [표 2]

통지 모드	설명
수동적	액세서리는 그것의 내부 상태 카운터를 업데이트함으로써 특성의 변화를 나타낸다(예를 들어, 후술되는 바와 같은 "수동적(passive)" 통지)
광고형	액세서리는 업데이트된 정보를 디바이스 검색 서비스를 통해 광고함으로써 이 특성의 변화를 나타낸다(예를 들어, 후술하는 바와 같은 "광고형(advertised)" 통지).
이벤트	액세서리는, 특성이 변화할 때, 접속되고 가입된 제어기에 통지를 송신할 것이다(예를 들어, 후술되는 바와 같이 비요청(unsolicited) HTTP EVENT 응답을 사용함)
능동적 접속	액세서리는 가입된 제어기에 대한 접속을 개시함으로써 이 특성의 변화를 나타낸다(예를 들어, 후술되는 바와 같은 "능동적(active)" 통지)

[0062]

[0063] 일부 실시예들에서, 수동적 통지는, 제어기에 의한 임의의 가입에 관계없이, 모든 액세서리들에 의해 모든 특성들에 대해 지원되며; 다른 지원되는 통지 모드들은 제어기들에 의한 가입 요청들에 기초하여 선택적으로 인에이블될 수 있다. 일부 실시예들에서, "Paired Read" 허가를 갖는 모든 핵심 특성들은 또한 적어도 "이벤트" 통지 모드를 지원한다.

[0064] "포맷(format)"(233)은 특성의 값에 대한 포맷을 식별하는 스트링, 예를 들어, 부울, 스트링, 정수, 부동 소수점, TLV 등일 수 있다. 일부 실시예들에서, 프로토콜은 인식되는 포맷들의 세트를 정의할 수 있고, 포맷(233)은 정의된 세트로부터 선택될 수 있다.

[0065] "값(value)"(234)은 특성의 현재 값일 수 있다.

[0066] "최소값(minValue)"(235) 및 "최대값(maxValue)"(236)는, 한계들이 필요한 경우, 특성에 대한 하한 및 상한을 설정하는 데 사용될 수 있다. 유사하게, "단계크기(stepSize)"(237)는, 최소 증분이 필요한 경우, 특성의 값을 변경하기 위한 최소 증분을 명시하는 데 사용될 수 있다. 최소값, 최대값, 및 단계크기가 명시되는 경우, 액세서리는 범위 내의 임의의 유효 값을 인식하고 이에 응답할 것으로 예상된다. 그러나, 전술한 바와 같이, 이는 액세서리 상에서 이용가능한 설정들의 수가 유효 값들의 수와 동일해야 한다는 것을 의미하지 않는다. 예를 들어, 전구 액세서리가 밝기 특성(203)을 갖는 경우, 액세서리는 (예를 들어, 전구에 공급되는 전류를 조절함으로써) 밝기를 제어하여, 밝기 특성이 100으로 설정되는 경우에 밝기가 최대이고 밝기 특성이 0으로 설정되는 경우에 밝기가 0이 되도록 할 수 있다. 밝기 특성이 정의된 전구가 0과 최대 밝기 사이의 적어도 하나의 중간 계조를 가질 것으로 예상되며(엄격하게 요구되지는 않지만); 액세서리 지원들이 100개보다 많거나 적은 중간 계조들인 경우, 액세서리는 그 밝기 계조들로 밝기 특성 값들을 매핑하는 것을 정의할 수 있다.

[0067] 일부 실시예들에서, 열거형 값 포맷이 지원될 수 있고, "유효값들(validValues)" 속성(238)은 포맷(233)이 "enumerated"로 명시되는 경우에 유효 값들을 열거하는 데 사용될 수 있다.

[0068] "단위(unit)" 속성(239)은 특성에 사용할 단위들을 나타낼 수 있다(예를 들어, 백분율, 특정 온도 스케일 등). 일부 실시예들에서, 프로토콜은 선호되는 단위계를 명시할 수 있고, 주어진 특성에 대한 단위 속성(239)은 이 단위계 내에서 선택될 수 있다.

[0069] "사용자디스크립터(userDescriptor)" 속성(240)은 인간-판독가능한 방식으로 특성 및 그것의 기능을 설명하는 스트링을 제공할 수 있다. 예를 들어, 현재 난방/냉방 상태 특성(212)에 대한 사용자 디스크립터는 "난방/냉방 시스템이 현재 난방중인지, 냉방중인지, 또는 오프인지를 나타낸다"고 말할 수 있다.

[0070] "소유자(owner)" 속성(241)은 특성을 정의 또는 재정의한 조직 단위(organizational unit)를 식별할 수 있다. 일부 실시예들에서, 이것은 사용자 또는 개발자가 다양한 특성들의 정의들의 소스를 이해하는 데 도움을 줄 수 있다.

[0071] 도 2e의 속성들 중 임의의 것 또는 전부(및 선택적으로 다른 속성들)가 특성에 대해 정의될 수 있다. 전술한 바와 같이, 핵심 특성들에 대해, 프로토콜은 디폴트 정의들을 명시할 수 있고, 핵심 특성이 사용되는 경우에, (예를 들어, 후술되는 바와 같이) 액세서리 정의 레코드 내의 특성의 속성들 모두를 포함할 필요는 없다. 일부

실시예들에서, 액세서리는 재정의되어야 하는 그러한 속성들을 액세서리 정의 레코드에 포함시킴으로써 핵심 특성을 재정의할 수 있다. 확장 특성들은 그것들의 정의 속성들 모두를 액세서리 정의 레코드에 포함시킴으로써 정의될 수 있다.

[0072] 일부 실시예들에서, 특성의 속성들은 제어기에 의해 판독될 수 있고, 특성을 제어하고/하거나 현재 값을 제시하기 위한 그래픽 사용자 인터페이스를 렌더링하는 방법을 결정하는데 사용될 수 있다. 예를 들어, 밝기 특성(203)(도 2a) 또는 백분율로 표현된 임의의 다른 특성의 경우, 제어기는 0 내지 100%의 슬라이더로서 제어부를 렌더링하고 1%의 단계로 슬라이더를 이동시킬 수 있다. 특성(201)(도 2a) 또는 부울로 표현된 임의의 다른 특성의 경우, 제어기는 온/오프 스위치를 렌더링할 수 있다. 온도-관련 특성들의 경우, 제어기는 "섭씨"의 단위(또는 다른 온도 단위)를 갖는 특성에 기초하여 온도 게이지 등을 렌더링할 수 있거나, 또는 그것은 단순히 수치 값을 표시할 수 있다. 사용자 디스크립터(240)가 특성에 대해 정의되는 경우, 제어기는 특성에 대한 사용자 인터페이스 제어 요소와 관련하여 텍스트를 렌더링할 수 있고, 이것은 제어 요소에 대한 사용자의 이해를 용이하게 할 수 있다. 따라서, 액세서리는 자기-설명(self-describing)할 수 있어서, 액세서리 정의 레코드가 주어지는 경우, 제어기는 그 액세서리에 대해 특별히 프로그래밍되어 있지 않으면서 임의의 액세서리를 제어하기 위한 인터페이스를 동적으로 생성할 수 있게 된다.

[0073] 도 2a 내지 도 2d에 도시된 특성들 및 도 2e에 도시된 속성들(또는 디스크립터들)은 예시적이며 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 균일한 액세서리 프로토콜은 임의의 수 및 조합의 특성들을 정의할 수 있고, 특성들은 도시된 것들과 상이한 속성들 및 값들의 세트를 사용하여 정의될 수 있다. 예를 들어, 최대 스트링 길이 속성은 스트링의 길이에 대한 상한을 명시하는 데 사용될 수 있고, 최대 데이터 길이 속성은 데이터-포함(data-containing) 특성들(예를 들어, 데이터 객체들 또는 데이터 블랍들)에 대한 상한을 명시하는 데 사용될 수 있다.

[0074] 또한, 언급한 바와 같이, 프로토콜은 액세서리 제조사들이 그들의 액세서리들에 대해 맞춤화된, 또는 제조사-특정의, 확장 특성들을 정의하도록 할 수 있다. 예로서, (인터넷 도메인 "discoball.com"을 소유한) 제조사가, 상이한 방향들 및 상이한 속도들로 회전하도록 제어될 수 있는 미러 공(mirrored ball), 및 공의 표면을 향해 지향될 수 있는 광원을 포함하는 "디스코 공(disco ball)" 시스템을 제조한다고 가정해보자. 광원은 도 2a 내지 도 2d의 핵심 특성들(예를 들어, "온" 특성(201) 및 밝기 특성(203))을 사용하여 모델링되고 제어될 수 있다. 제조사는 광원 및/또는 미러 공의 추가적인 제어를 위한 확장 특성들을 정의하고자 할 수 있다. 도 2f는 이 시나리오를 위한 확장 특성들의 예들을 도시한다.

[0075] 스트로브(strobe) 특성(242)은 스트로브 효과를 제어하기 위한 예시적인 특성이다. true이면, 광은 스트로브 모드에서 동작되고; false이면, 광이 정상 모드에서 동작된다.

[0076] 방향 특성(243)은 미러 공의 회전 방향을 제어하기 위한 예시적인 특성이다. 공은 회전중이지 않거나(정지됨), 시계방향으로 회전중이거나, 반시계방향으로 회전중일 수 있다. 핵심 특성들이 회전 방향 특성(225)을 포함하는 실시예들에서, 제조사는 핵심 특성을 사용할지 또는 확장 특성을 정의할지를 선택할 수 있다.

[0077] 속도 특성(244)은 미러 공의 회전 속도를 제어하기 위한 예시적인 특성이다. 이 예에서 속도는 2개의 설정(느린 회전의 0, 빠른 회전의 1)을 갖는다. 핵심 특성들이 회전 속도 특성(226)을 포함하는 실시예들에서, 제조사는 핵심 특성을 사용할지 또는 확장 특성을 정의할지를 선택할 수 있다.

[0078] 전술한 바와 같이, 균일한 액세서리 프로토콜은 하나 이상의 서비스로서 액세서리를 모델링할 수 있으며, 이때 각각의 서비스는 특성들의 집합으로서 모델링된다. 따라서, 서비스는, 핵심 특성들 및/또는 확장 특성들의 임의의 조합을 포함할 수 있는 그 구성 특성들을 식별함으로써 정의될 수 있다. 일부 실시예들에서, 균일한 액세서리 프로토콜은, 자주 사용될 것으로 예상되는 그리고/또는 다양한 액세서리 유형들에 걸쳐 유용할 것으로 예상되는 서비스들을 포함할 수 있는, 핵심 서비스들의 세트를 정의할 수 있다. 서비스들의 세트는 확장가능하게 만들어질 수 있으며; 예를 들어, 액세서리 제조사들은 제조사-특정 특성들을 핵심 서비스에 추가하거나 또는 추가적인 "확장" 서비스들을 정의하도록 허용될 수 있다. 그러나, 가능한 경우 핵심 서비스들을 사용하는 것은, 시스템 설계자들이 미리정의된 서비스들 및 특성들을 활용하도록 허용함으로써 시스템 설계 및 동작을 용이하게 할 수 있다.

[0079] 도 2g 및 도 2h는 본 발명의 일 실시예에 따라 정의될 수 있는 핵심 서비스들(251 내지 258)의 일부 예들을 도시한다. 각각의 서비스(251 내지 258)는 서비스를 식별하는 고유 이름일 수 있는 "타입"을 할당받을 수 있다. 이러한 예에서, 특성 타입들과 유사하게, 역 도메인 이름 규약은, 액세서리 제조사들에 의한 새로운 서비스들의

정의들을 용이하게 하기 위해 타입들을 정의하는 데 사용된다. 따라서, 도 2g 및 도 2h의 핵심 서비스 타입들은 "com.proto.svc"로 시작한다(이때 "svs"는 타입이 특성보다는 서비스에 대한 것임을 나타내기 위해 사용될 수 있다).

[0080] 일부 실시예들에서, 타입에 추가하여 또는 그 대신에, 각각의 서비스는 고유 숫자 서비스 식별자(도 2g 및 도 2h에 도시되지 않음)가 할당될 수 있다. 예를 들어, IETF RFC 4122에 의해 정의된 UUID(36개의 16진수로 구성된 식별자)가 각각의 서비스에 할당될 수 있다. 균일한 액세서리 프로토콜은 모든 서비스들에 대해 공통적인 "기본" UUID(예를 들어, 마지막 28개의 16진수)(이는, 원하는 바에 따라, 모든 특성들에 할당된 기본 UUID와 동일하거나 그와 상이할 수 있음)를 정의하고, 고유한 "짧은" UUID(예를 들어, 처음 8개의 16진수)를 각각의 서비스에 할당할 수 있으며; 임의의 번호 체계가 사용될 수 있다. UUID들의 사용은, 특히 "짧은" UUID에 기초하여 UUID들을 절단하기 위한 규약과 조합되어, 제어기 또는 액세서리로 하여금, 문자열을 사용하는 것에 비하여 더 적은 양의 전송된 데이터를 사용하여 관심있는 서비스를 명시하도록 허용할 수 있다.

[0081] 각각의 서비스(251 내지 258)는, 액세서리가 구현할 수 있고 필수 특성들의 세트(도 2g 및 도 2h의 "필수 특성(Required Ch.)" 속성) 및 선택적 특성들의 세트(도 2g 및 도 2h의 "선택적 특성(Optional Ch.)" 속성)를 참조하여 정의되는 기능을 표현한다. 본 명세서의 예들에서, 각각의 서비스는 적어도 하나의 선택적 특성(이름)을 갖고; 다른 실시예들에서, 선택적 특성들의 세트는 적어도 일부 서비스들에 대해 비어있을 수 있다.

[0082] 이 예에서, 특성은, 주어진 핵심 서비스를 제공한다고 주장하는 임의의 호환가능한 액세서리가 "필수" 특성을 인식하고 이를 사용해 액세서리를 제어할 것으로 예상되는 경우들에서, 그 핵심 서비스에 대해 "필수인" 것으로 정의된다. 예를 들면, 전구(light bulb) 서비스(251)는 필수 특성 "com.proto.ch.on"(도 2a의 특성(201))을 포함하며; 이것은, 만약 액세서리가 "com.proto.svc.lightbulb" 서비스를 제공한다고 주장하면, 액세서리는 조명을 온(true) 또는 오프(false)시킴으로써 특성 "com.proto.ch.on"에 대한 기록 요청들에 응답하고, 조명이 온(true) 또는 오프(false)인지 여부를 나타내는 부울 값을 반환함으로써 특성 "com.proto.ch.on"에 대한 판독 요청들에 응답할 것으로 예상된다는 것을 의미한다. 일부 실시예들에서, 액세서리는 인가된(또는 페어링된) 제어기들로부터의 요청들에만 응답할 수 있고; 인가는 아래에서 기술된다.

[0083] 이 예에서, 특성은, 액세서리가 그것의 서비스 정의에 그 특성을 포함하도록 요구되지는 않지만 그렇게 할 수 있는 경우들에서, 주어진 핵심 서비스에 대해 "선택적인" 것으로서 정의된다. 예를 들어, 전구 서비스(251)는 선택적 특성 "com.proto.ch.brightness"(도 2a의 특성(203))을 포함한다. 온 또는 오프 설정들만을 갖는 전구 액세서리(예컨대, 많은 종래의 형광 조명 기구들)는 밝기 제어에 대한 어떠한 사용도 갖지 않을 것이고, 그러한 액세서리는 "com.proto.ch.brightness" 특성을 지원할 필요가 없다. 밝기 제어를 갖는 전구 액세서리(예를 들면, 조광가능한(dimmable) 램프)는, 제어기가 조광조작기(dimmer)를 동작시킬 수 있도록 하는 "com.proto.ch.brightness" 특성을 포함할 수 있다. 유사하게, 색상 및 채도는 전구 서비스(251)에 대해 선택적 특성들일 수 있다.

[0084] 다른 서비스들(252 내지 258)은 유사하게 정의되어, 필수의 그리고 선택적인 특성들의 조합을 명시할 수 있다. 특성들은 도 2g 및 도 2h에서 타입에 의해 식별되며, 전술한 도 2a 내지 도 2d에서 도시된 바와 같이 정의될 수 있다. 특성은 하나의 핵심 서비스에서는 요구되고 다른 것에서는 선택적일 수 있다는 것에 유의해야 한다. 예를 들어, 잠금-메커니즘 현재 상태 및 목표 상태는 차고문 열림장치 서비스(253)에 대해서는 선택적 특성들이지만, 잠금 메커니즘 서비스(254)에 대해서는 필수 특성들이다.

[0085] 도 2g 및 도 2h의 핵심 서비스 예들은 예시의 목적으로 제공된다고 이해되어야 한다. 임의의 수의 핵심 서비스들이 정의될 수 있고, 주어진 핵심 서비스와 연관된 특정한 필수 특성 및/또는 선택적 특성은 원하는 바에 따라 변경될 수 있다. 또한, 확장 특성들이 지원되는 실시예들에서, 제조사는 확장 특성들을 추가함으로써 핵심 서비스를 증강시키도록 허용될 수 있다. 추가적으로 또는 그 대신에, 일부 실시예들에서, 제조사는 핵심 특성들 및/또는 확장 특성들의 임의의 조합을 사용하여 확장 서비스를 정의할 수 있다.

[0086] 액세서리 자체는, 프로토콜에 의해 명시된 핵심 서비스일 수 있는 "액세서리 정보" 서비스를 사용하여 설명될 수 있다. 일부 실시예들에서, 프로토콜은, 모든 프로토콜-호환 액세서리들이 액세서리 정보 서비스의 인스턴스를 그것들의 액세서리 정의 레코드들에 포함시키는 것을 명시할 수 있다. 도 2i는 본 발명의 일 실시예에 따른 액세서리 정보 서비스(261)에 대한 정의를 (도 2g 및 도 2h와 동일한 포맷으로) 도시하고, 도 2j는 액세서리 정보 서비스(261)에 대한 추가적인 특성들(271 내지 276)의 정의들을 (도 2a 내지 도 2d와 동일한 포맷으로) 도시한다.

[0087] 식별(identify) 특성(271)은 액세서리의 자기-식별 루틴을 호출하기 위해 제어기에 의해 기록될 수 있다. 이 자기-식별 루틴은, 액세서리가 사용자-관측가능한 행동을 개시하는 것을 포함할 수 있다. 예를 들어, 액세서리는 조명을 깜박이거나, 소리를 방출하거나, 진동하거나, 이동가능한 컴포넌트를 이동시키거나(예를 들어, 문을 열고 닫음), 특정 메시지(예를 들어, "나 여기 있어(Here I am)")를 표시하거나, 또는 사용자가 관측할 수 있는 일부 다른 물리적 행동을 수행할 수 있다. 예를 들어, 사용자가 제어하고자 하는 액세서리와 제어기가 통신하고 있다는 것을 확인하는 것과 관련하여, 액세서리의 자기-식별 루틴을 제어기로부터 호출하는 것은 유용할 수 있다. 일부 실시예들에서, 제어기는 식별 특성(271)에 "true"를 기록함으로써 액세서리의 자기-식별 루틴을 호출할 수 있다.

[0088] 제조사 이름(manufacturer name) 특성(272), 모델 이름(model name) 특성(273), 및 일련 번호(serial number) 특성(274)은 액세서리에 관한 식별 정보를 획득하기 위해 제어기에 의해 판독될 수 있다. 일부 실시예들에서, 값들은 인간-판독가능한 문자열들일 수 있다.

[0089] 펌웨어 개정(firmware revision) 특성(275), 하드웨어 개정(hardware revision) 특성(276), 및/또는 소프트웨어 개정(software revision) 특성(277)은, 액세서리와 상호작용하는 방법을 결정하기 위해 제어기에 의해 사용될 수 있는, 액세서리에 대한 세대 정보(generational information)를 획득하기 위해 제어기에 의해 판독될 수 있다. 일부 실시예들에서, 개정 정보는 표준 포맷, 예를 들어, <x>.<y>.<z>;<w>로 표현될 수 있으며, 여기서 <x>는 메이저 버전 번호이고, <y>는 마이너 버전 번호이고, <z>는 개정 버전 번호이며, <w>는 추가 정보를 포함할 수 있다.

[0090] 특성들 및 서비스들의 전술한 예들은 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 균일한 액세서리 프로토콜은 임의의 수 및 조합의 핵심 특성들 및 핵심 서비스들을 정의할 수 있고, 주어진 서비스는 도시된 것들과 상이한 특성들의 세트로 정의될 수 있다. 언급한 바와 같이, 핵심 서비스들은 (예를 들어, 액세서리 제조사들에 의해 정의된 바와 같이) 확장 특성들 및/또는 확장 서비스들로 증강되어, 균일한 통신 및 제어 프레임워크 내에서 다양한 요구들에 상당한 정도의 융통성 및 적응성을 제공할 수 있다.

[0091] 일부 실시예들에서, 핵심 서비스 정의들의 상이한 버전들이 공존할 수 있다. 상이한 세대들의 제품들 사이의 호환성을 용이하게 하기 위해, 핵심 서비스 정의의 이후 버전들은 새로운 선택적 특성들을 추가하는 것으로 제한될 수 있으며; 필수 특성들의 일관적인 세트를 유지하는 것은 상호운용성을 용이하게 할 수 있다.

[0092] 전술한 바와 같이, 일부 실시예들에서, 액세서리 제조사는 서비스에 확장 특성들을 추가할 수 있다. 예를 들어, 액세서리가 스트로브 옵션을 갖는 조명인 경우, 제조사는 핵심 전구 서비스(251)에 스트로브 특성(예를 들어, 도 2f의 스트로브 특성(242))을 추가할 수 있다. 제조사는 또한 확장 서비스들을 정의할 수 있다. 예를 들어, 전술한 디스코 공의 경우, 제조사는 미러 공 및 조명을 제어하기 위해 확장 서비스 "com.discoball.svc.discoball"을 정의할 수 있다. 이 서비스는 다음의 특성들을 포함할 수 있다:

[0093] ● com.proto.ch.on (도 2a의 특성(201));

[0094] ● com.proto.ch.brightness (도 2a의 특성(203));

[0095] ● com.proto.ch.hue (도 2a의 특성(204));

[0096] ● com.discoball.ch.strobe-on (도 2f의 특성(242));

[0097] ● com.discoball.ch.rotate-direction (도 2f의 특성(243)); 및

[0098] ● com.discoball.ch.rotate-speed (도 2f의 특성(244)).

[0099] 특성들을 갖는 서비스들의 집합으로서 액세서리를 표현하는 액세서리 모델은, 액세서리 객체로서 제어기에 전달될 수 있다. 액세서리 객체는 JSON 또는 구조화된 데이터 객체들을 표현하기 위한 다른 표기법들을 사용하여 (예컨대, 중첩된 키-값 쌍(nested key-value pair)들을 사용하여) 전달될 수 있다. 도 3a 내지 도 3c는 본 발명의 일 실시예에 따른 액세서리 객체(300)의 예를 도시한다. 액세서리 객체(300)는 JSON을 사용해 표현되고; 다른 표현들이 대체될 수 있다. 부착된 조명을 갖는 차고문 열림장치가 예시의 목적으로 사용되지만, 액세서리 모델들은 임의의 특정 액세서리로 제한되지 않는다.

[0100] 액세서리 객체(300)는, 서비스 인스턴스들(310, 320, 330)의 배열로서 표현될 수 있으며, 그 각각은 특성 인스

턴스들의 배열로서 표현될 수 있다. 따라서, 서비스 인스턴스(310)는 특성 인스턴스들(311 내지 315)을 포함할 수 있고; 서비스 인스턴스(320)는 특성 인스턴스들(321 내지 325)을 포함할 수 있으며; 서비스 인스턴스(330)는 특성 인스턴스들(331 및 332)을 포함할 수 있다. 이 예에서, 서비스 인스턴스(310)는 도 2i의 액세서리 정보 서비스(261)의 인스턴스이고, 서비스 인스턴스(320)는 도 2g의 차고문 열림장치 서비스(253)의 인스턴스이며, 서비스 인스턴스(330)는 도 2g의 전구 서비스(251)의 인스턴스이다. (용어들 "서비스" 및 "서비스 인스턴스"는 본 명세서에서 상호교환적으로 사용될 수 있으며, 용어들 "특성" 및 "특성 인스턴스"도 그럴 수 있다.)

[0101] 각각의 서비스 인스턴스(310, 320, 330) 및 각각의 특성 인스턴스(311 내지 315, 321 내지 325, 331 및 332)는 서비스 또는 특성 타입을 포함할 수 있으며, 이는 그것이 인스턴스로 되어 있는 서비스 또는 특성을 식별한다. 이 예에서, 타입 스트링들이 사용된다. 일부 실시예들에서, UUID 또는 절단된 UUID가 사용되어, 서비스 및 특성 타입들이 스트링 대신에 수치적으로 식별되게 할 수 있다. 이것은 액세서리 객체(300)의 크기를 줄일 수 있다. 각각의 서비스 인스턴스 및 각각의 특성 인스턴스는 또한 인스턴스 식별자를 할당받는다. 이 예에서, 액세서리의 각각의 서비스 인스턴스 및 특성 인스턴스는, 순차적으로 또는 임의의 다른 바람직한 방식으로 할당될 수 있는 고유한 인스턴스 식별자를 갖는다. 이것은 후술하는 바와 같이, 액세서리 내의 임의의 서비스 인스턴스 또는 특성 인스턴스가 그것의 인스턴스 식별자를 참조하여 어드레싱되도록 할 수 있다. 일부 실시예들에서, 상이한 유일성 규칙(uniqueness rule)들이 구현될 수 있다. 예를 들어, 각각의 서비스 인스턴스는 고유한 서비스 인스턴스 식별자를 가질 수 있고, 각각의 특성 인스턴스는 서비스 인스턴스 내의 특성 인스턴스들 사이에서 고유한 특성 인스턴스 식별자를 가질 수 있다. 이것은, 서비스 인스턴스가 그것의 인스턴스 식별자를 참조하여 어드레싱되도록 하고, 특성 인스턴스가 그것이 속하는 서비스 인스턴스의 인스턴스 식별자와 함께 특성의 인스턴스 식별자에 의해 어드레싱되도록 할 수 있다. 다른 방식들이 사용될 수 있다.

[0102] 도 3a에서, 서비스 인스턴스(310)는 도 2i의 액세서리 정보 서비스(261)의 인스턴스일 수 있다. 일부 실시예들에서, 프로토콜은, 각각의 액세서리가 액세서리 정보 서비스의 단일 인스턴스를 갖고 액세서리 정보 서비스 인스턴스가 1인 인스턴스 ID를 갖는 것을 명시할 수 있으며; 상이한 규칙들이 원하는 대로 명시될 수 있다. 특성 인스턴스들(311 내지 315)은 서비스(261)를 위한 필수 특성들의 인스턴스들일 수 있다. 도 3b에서, 서비스 인스턴스(320)는 도 2g의 차고문 열림장치 서비스(253)의 인스턴스일 수 있다. 특성 인스턴스들(321 내지 323)은 서비스(253)를 위한 필수 특성들의 인스턴스들일 수 있다. 이 예에서, 서비스 인스턴스(320)는 또한 차고문 열림장치의 잠금 메커니즘을 제어하기 위해 선택적 특성 인스턴스들(324, 325)을 포함한다. 도 3c에서, 서비스 인스턴스(320)는 도 2g의 전구 서비스(251)의 인스턴스일 수 있다. 특성 인스턴스(331)는 서비스(251)를 위한 필수 특성의 인스턴스일 수 있고, 특성 인스턴스(332)는 선택적 밝기 특성의 인스턴스일 수 있다.

[0103] 액세서리 정보 서비스 이외의 서비스들에 대해, 동일한 서비스의 다수의 인스턴스들이 액세서리 객체 내에서 공존할 수 있다. 예를 들어, 다수의 독립적으로-제어가능한 전구들을 동작시키는 액세서리는 각각의 전구에 대해 전구 서비스(251)의 상이한 인스턴스를 가져서, 각각의 전구의 상태가 독립적으로 제어되게 할 수 있다.

[0104] 액세서리 객체(300)는 또한 판독 허가를 갖는 각각의 특성 인스턴스에 대한 현재 값을 제공할 수 있다. 예를 들어, 차고문이 현재 닫혀 있고(현재 문 상태 특성 인스턴스(321)가 2의 값을 갖고, 이는 "closed"로 매핑됨), 조명은 오프이다(온 특성 인스턴스(331)는 false의 값을 가짐). 식별 특성 인스턴스(315)는 null 값을 갖는데, 이는 이 특성에 대한 액세스가 기록-전용이기 때문이다.

[0105] 각각의 특성 인스턴스에 대한 허가들은 "perms" 스트링들의 배열로서 나타낼 수 있다. 이 예에서, 배열은, 액세서리가 이 특성에 관한 이벤트 통지를 지원하는 것을 나타내기 위해서 "Events" 스트링을 포함할 수 있다. 예를 들어, 후술되는 바와 같이, 제어기는, 그 허가들이 "Events" 스트링을 포함하는 임의의 특성 인스턴스에 대한 이벤트 통지들에 가입할 수 있다. 이벤트 통지 메커니즘들의 예들이 아래에 기술된다.

[0106] 도 3c는 핵심 특성을 부분적으로 재정의하는 액세서리의 예를 도시한다. 밝기 특성 인스턴스(332)는, 새로운 최소, 최대, 및 단계 값들을 명시함으로써 밝기 증분들의 수를 줄이기 위해, (핵심 특성(203)에 대하여) 재정의된다. 재정의는 인스턴스(332)에만 적용되고, 액세서리 객체(300) 내에서 정의된 임의의 다른 서비스 인스턴스의 밝기 특성, 또는 주어진 제어기가 상호운용되는 임의의 다른 액세서리의 밝기 특성에는 영향을 미치지 않을 것이다.

[0107] 도 3a 내지 도 3c의 액세서리 객체는 예시적이며 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 액세서리는 임의의 수 및 조합의 서비스 인스턴스들을 포함할 수 있으며, 액세서리 내의 서비스 인스턴스는 임의의 수 및 조합의 특성 인스턴스들을 포함할 수 있다. 주어진 특성 인스턴스의 더 많거나 적은 속성들이 액세서리 객체 내에서 명시될 수 있고; 예를 들어, 도 2e에 도시된 다양한 속성들 중 임의의 것 또는 전부가 명시될 수 있

다. 액세서리가 다수의 서비스 인스턴스를 포함하는 경우, 서비스 인스턴스들 중 하나(예를 들어, 액세서리 정보 서비스 인스턴스 이외의 액세서리 객체에서 열거된 제1 서비스 인스턴스)는 1차(primary) 서비스로 지정될 수 있다.

[0108] 또한, 도 3a 내지 도 3c는 JSON을 사용하는 구현예를 도시하지만, 임의의 특정 표기법 또는 신택스의 사용은 요구되지 않는다. 예를 들어, 본 개시내용에 대한 액세스를 갖는 통상의 기술자는, 액세서리 서비스들 및 특성들이 블루투스 LE GATT(Generic Attribute Profile)를 활용함으로써 표현되어, 그에 의해 블루투스 LE를 사용하는 제어기들과 액세서리들 사이의 통신을 용이하게 할 수 있다는 것을 이해할 것이다. 예를 들어, 서비스들 및 특성들은, 각각의 서비스 및 각각의 특성이 고유 UUID를 갖도록 UUID들에 의해 식별될 수 있다.

[0109] 또한, 일부 실시예들에서, 제어기는 하나 이상의 액세서리와 상호작용하기 위해 단일 종점(endpoint)(액세서리 서버로도 지칭됨)과 통신할 수 있다. 이를 지원하기 위해, 액세서리 정의 레코드는 하나 이상의 액세서리 객체의 배열을 포함할 수 있으며, 여기서 각각의 액세서리 객체는 도 3a 내지 도 3c에 도시된 방식으로 표현될 수 있다. 각각의 액세서리 객체는, 액세서리 객체들의 배열 내에서 고유한, 액세서리 인스턴스 ID를 할당받을 수 있다. 주어진 액세서리 서버가 하나 이상의 액세서리 객체를 제공할 수 있고, 액세서리 객체들 모두는 그 액세서리 서버에 대한 단일 액세서리 정의 레코드 내에 표현된다는 것이 이해되어야 한다. 제어기는 임의의 수의 액세서리 서버들과 통신할 수 있다. 일부 실시예들에서, 다수의 액세서리를 제어하는(또는 메시지들을 다른 액세서리에 중계하는) 단일 종점은 "브리지"로 지칭될 수 있고, 그러한 액세스 포인트에 대한 액세서리 정의 레코드는, 브리지에 대한 액세서리 객체뿐만 아니라 브리지에 의해 제어되는 각각의 액세서리에 대한 액세서리 객체를 포함할 수 있다. 브리지에 대한 액세서리 객체는 단지 액세서리 정보 서비스의 단일 인스턴스를 포함할 수 있고, 브리지에 대한 액세서리 객체의 존재는 브리지가 존재한다는 것을 제어기에 나타낼 수 있다.

[0110] 동작 시, 각각의 액세서리(또는 액세서리 서버)는 그것의 액세서리 정의 레코드를 지속적 저장소에 저장할 수 있다. 액세서리는 요청 시 액세서리 정의 레코드의 전부 또는 일부를 제어기에 제공할 수 있다. 후술되는 바와 같이, 이것은 디바이스 검색 프로세스의 일부로서, 또는 다른 시간들에(예를 들어, 페어링된 제어기로부터의 요청 시) 일어날 수 있다. 일부 실시예들에서, 제어기는 액세서리와 페어링하거나 또는 달리 그에 접속할지 여부를 결정하기 위해 액세서리 정의 레코드로부터의 정보를 사용할 수 있다. 페어링 또는 접속이 확립되는 경우, 제어기는 액세서리를 제어하는 방법을 결정하기 위해 액세서리 정의 레코드를 사용할 수 있다.

[0111] 액세서리 정의 레코드는 자기-내장형(self-contained)일 수 있으며, 이는 제어기가 액세서리와 상호작용하기 위해 액세서리에 대한 어떠한 다른 정보도 필요로 하지 않는다는 것을 의미한다. 예를 들어, 액세서리 정의 레코드는 제조사-특정 특성(예를 들어, "com.discoball.ch.rotate-direction") 및/또는 제조사-특정 서비스(예를 들어, "com.discoball.svc.discoball")의 완전한 정의를 포함할 수 있으며, 정의는 특성들 및 서비스들의 인간-판독가능한 디스크립터들을 포함할 수 있다. 제어기는 다양한 특성들에 대한 인간-판독가능한 디스크립터 및 (예를 들어, 특성의 "단위" 속성에 기초하여 선택된) 사용자-동작가능한 제어 요소를 제시하는 사용자 인터페이스를 생성하도록 프로그램될 수 있고, 사용자는 원하는 대로 액세서리를 제어하기 위해 제어 요소를 동작시킬 수 있다. 제어기는 사용자 입력에 기초하여 제어 메시지들(요청들)을 송신함으로써(예를 들어, 새로운 값들을 특성들에 기록함), 제어기가 액세서리-특정 소프트웨어 또는 다른 액세서리-특정 맞춤화를 필요로 하지 않으면서도, 액세서리의 제어를 허용할 수 있다.

[0112] 예시적인 액세서리 검색 프로세스

[0113] 액세서리를 제어하기 전에, 제어기는 먼저 제어될 액세서리와 통신을 확립한다. 본 명세서에서 사용된 "액세서리 검색"은 일반적으로, 제어기가 그것이 통신해야 할 액세서리의 위치를 찾을 수 있는, 임의의 프로세스를 지칭한다. 일부 경우들에서 검색은 제어기와 액세서리 사이의 통신이 발생해야 한다는 사용자 검증을 포함할 수 있다. 일부 실시예들에서, 액세서리 검색은, 무선 또는 다른 네트워크 상에서 디바이스들 및/또는 서비스들의 위치를 찾는 것을 용이하게 하는 기존의 서비스 검색 프로토콜들, 예컨대 SSDP(Simple Service Discovery Protocol), UPnP 포럼(http://www.upnp.org)에 의해 개발된 프로토콜, 애플 사(Apple Inc.)에 의해 개발된 Bonjour® 네트워킹 기술(IETF RFC 6762 및 IETF RFC 6763으로서 출판되고, 본 명세서에서 "Bonjour"로 지칭됨)을 활용할 수 있다. 디바이스 검색 서비스에서, 하나의 디바이스(예를 들어, 액세서리)는 그것의 존재, 주소, 및 선택적으로 그것의 능력들에 대한 추가적인 정보를 나타내는 정보를 광고할 수 있다. 다른 디바이스들(예를 들어, 제어기들)은 광고들을 브라우징하고, 브로드캐스트 정보에 기초하여 관심있는 디바이스들을 식별할 수 있다. 주소를 사용하여, 브라우징 디바이스는 광고자와의 통신을 개시할 수 있다.

[0114] 네트워크 및 검색 서비스에 따라, 광고는, (예를 들어, 멀티캐스트 또는 비콘 신호를 통한) 정보의 실시간 브로

드캐스팅, 및/또는 다른 디바이스들이 정보를 검색할 수 있는 중앙 저장소(예를 들어, 네트워크 액세스 포인트에 있음)로 광고 정보를 제공하는 것을 포함할 수 있지만, 그럴 필요는 없다. 광고들의 브라우징은 브로드캐스트 광고들을 검출하는 것 및/또는 중앙 저장소로부터 광고 정보를 검색하는 것을 포함할 수 있다.

[0115] 도 4는 본 발명의 일 실시예에 따른 제어기(404)에 의해 액세서리(402)를 검색하기 위한 프로세스(400)의 흐름도이다. 액세서리(402)는, 예를 들어, 도 1의 액세서리들 중 임의의 것일 수 있고, 제어기(404)는, 예를 들어, 도 1의 제어기(102)일 수 있다.

[0116] 블록(410)에서, 액세서리(402)는 그것이 현재 페어링되어 있지 않다는 것(또는 페어링할 제어기를 찾아보는 것)을 나타내도록 상태 비트를 설정한다. 이것은 예를 들어, 후술되는 상태 플래그 표시자 "sf#" 내의 비트일 수 있다.

[0117] 블록(412)에서, 액세서리(402)는 디바이스 검색 서비스 상에서 균일한 액세서리 프로토콜("UAP")을 지원하는 액세서리로서 자신의 존재를 광고할 수 있다. 예를 들어, Bonjour를 사용하여, 액세서리는 이름 및 서비스 타입으로 자신을 광고할 수 있다. 이름은 액세서리에 대한 사용자-판독가능한 이름(예를 들어, "Thermostat")일 수 있고, 일부 경우들에서 광고된 이름은 액세서리 정의 레코드의 액세서리 정보 서비스 인스턴스 내에 명시된 이름일 수 있다. 서비스 타입은 균일한 액세서리 프로토콜에 대해 정의될 수 있다(예를 들면, 서비스 타입 "_uap._tcp"). 광고는 또한 추가 정보를 포함할 수 있다. 예를 들어, 액세서리는 표 3에 도시된 키들을 갖는 Bonjour TXT 레코드를 제공할 수 있다.

[0118] [표 2]

키	설명
“c#”	현재 구성 번호. 액세서리 정의 레코드가 수정될 때(예를 들어, 액세서리, 서비스, 또는 특성이 추가 또는 제거될 때) 업데이트하고; 재부팅에 걸쳐, 전원 사이클을 지속하는 것 등. 액세서리 구성 변화들을 검출하기 위해 페어링된 제어기에 의해 사용될 수 있음.
“ff”	특징 플래그(feature flag)들 선택된 특징들의 존재 또는 부재를 나타내기 위해 사용가능한 비트마스크일 수 있음.
“id”	액세서리의 전역 고유 ID(예를 들어, 1 차 MAC 주소)
“md”	액세서리 모델 이름
“ps”	액세서리의 1 차 서비스 타입(예를 들어, "com.proto.svc.lightbulb") 존재한다면, (이름 이외의) 액세서리 기능들에 대한 추가 정보를 제어기에 제공할 수 있음.
“pv”	프로토콜 버전 스트링(예를 들어, <major>.<minor>의 포맷으로 됨) 호환성을 확인하는 데 사용될 수 있음. 일부 실시예들에서, 액세서리가 나타내는 메이저 버전이 제어기의 소프트웨어의 메이저 버전보다 큰 경우, 제어기는 이용가능한 세트로부터 액세서리를 배제할 수 있다.
“s#”	현재 상태 번호. 특성의 값이 변화할 때마다 증분하고; 재부팅에 걸쳐, 전원 사이클을 지속하는 것 등.
“sf#”	검색 상태 플래그들. 현재 상태의 양상들을 나타내는 비트마스크인 수 있음(예를 들어, 페어링되지 않음, 액세서리 상에서 문제가 검출됨, 액세서리에 액세스하기 위해 인가가 요구됨, 액세서리가 무선 네트워크에 참여하도록 구성되지 않음).

[0119]

[0120] 통상의 기술자는, 유사한 정보가 다른 서비스 검색 프로토콜들 및 기술들을 사용해 분산될 수 있다는 것을 이해할 것이다. 예를 들어, SSDP를 사용하여, 액세서리는 멀티캐스트 HTTP NOTIFY 메시지를 사용해 이름 및 서비스

타입 URI를 광고할 수 있고, URI는 액세서리로의 유니캐스트 요청을 통해 추가 정보를 검색하기 위해 제어기에 의해 사용될 수 있다.

[0121] 블록(414)에서, 제어기(404)는 구성되지 않은(unconfigured) 액세서리들을 브라우징할 수 있다. 특별한 타이밍이 요구되지는 않지만, 일반적으로 제어기는, 제어기가 브라우징할 때에 액세서리의 광고가 검출가능한 경우에만, 액세서리를 검색할 것이다.

[0122] 블록(416)에서, 제어기(404)는, 예를 들어, 블록(412)으로부터의 광고를 검출함으로써, 액세서리(402)을 발견할 수 있다. 블록(418)에서, 제어기(404)는, 광고에 기초하여, 액세서리(402)가 "관심대상"인지, 또는 상호운용을 위한 잠재적 후보인지를 결정할 수 있다. 예를 들어, 제어기(404)는, 액세서리가 이미 구성되어 있거나 또는 제어기와 페어링되어 있는지 여부를 결정하기 위해 표 2의 검색 상태 플래그들 "sf#"을 확인할 수 있다. 다른 예로서, 제어기(404)는, 액세서리의 프로토콜 버전이 제어기의 것과 호환가능한지 여부를 결정하기 위해, 표 2의 프로토콜 버전 "pv"를 확인할 수 있다. 또한, 일부 경우들에서, 제어기는, 특정 콘텍스트에서(예를 들어, 특정 애플리케이션을 실행함) 액세서리들을 브라우징하고 있을 수 있고, 광고된 이름, 1차 서비스 식별자, 액세서리 모델 이름, 특징 플래그들, 또는 액세서리의 광고로부터 이용가능한 임의의 다른 정보에 기초하여, 관심대상인 액세서리들을 제한할 수 있다. 제어기(404)는 액세서리가 관심대상이 아니라고 결정하는 경우, 제어기(404)는 블록(414)으로 복귀하고 계속해서 브라우징할 수 있다. (일부 실시예들에서, 브라우징 동작은 관심대상인 액세서리가 발견되지 않으면 타임아웃할 수 있다.)

[0123] 블록(422)에서, 제어기(404)는 사용자에게 액세서리에 관한 정보를 제시할 수 있고, 블록(424)에서, 사용자는, 제어기가 액세서리와 페어링을 확립해야 하는지 여부를 나타내는 입력을 제공할 수 있다. 예를 들어, 제어기(404)는 액세서리의 광고로부터 획득된 정보 중 임의의 것 또는 전부를 사용자에게 제시하고, 제어기(404)가 액세서리(402)에 접속해야 하는지 여부를 나타내도록 사용자에게 프롬프트할 수 있다. 요구되는 것은 아니지만, 사용자 확인을 요청하는 것은, 제어기와 액세서리 사이의 스퓨리어스(spurious) 페어링 또는 원하지 않는 페어링을 방지하는 데 도움이 될 수 있다.

[0124] 블록(426)에서, 제어기(404)는 블록(424)에서 수신된 사용자 입력을 해석하고, 그것이 액세서리(402)와 페어링해야 하는지 여부를 결정할 수 있다. 만약 그렇지 않으면, 제어기(404)는 다른 액세서리들을 찾기 위해 블록(414)로 복귀할 수 있다. 제어기(404)와 액세서리(402)가 페어링되어야 하는 경우, 블록들(428, 430)에서, 제어기(404) 및 액세서리(402)는 페어 셋업 프로세스를 실행할 수 있다. 일부 실시예들에서, 페어 셋업 프로세스는 제어기(404)와 액세서리(402) 사이의 안전한 통신을 용이하게 하기 위해 암호화 키들을 설정하는 데 사용될 수 있고; 블록들(428, 430)에서 구현될 수 있는 페어 셋업 프로세스의 예들이 아래에 기술된다. 일부 실시예들에서, 사용자 확인은 페어 셋업 프로세스에 통합될 수 있고, 페어 셋업을 개시하기 이전에 별도의 사용자 확인은 요구되지 않는다.

[0125] 페어 셋업 프로세스가 성공적으로 완료한다고 가정하면, 블록(431)에서, 액세서리(402)는, 예를 들어, 전술한 상태 플래그 표시자 "sf#"를 업데이트함으로써, 액세서리와 통신하기 위해 지금 인가가 요구된다는 것 및/또는 액세서리가 적어도 하나의 제어기와 지금 페어링되어 있다는 것을 나타내기 위해, 그것의 상태를 업데이트할 수 있다.

[0126] 블록(432)에서, 제어기(404)는 액세서리(402)로부터 액세서리 정의 레코드를 획득하고 캐싱(cach)할 수 있으며, 액세서리(402)는 블록(434)에서 요청 시 레코드를 제공할 수 있다. 제어기(404)가 액세서리의 정의 레코드를 캐싱하는 경우, 정보는 액세서리(402) 내의 상태 변화들을 검출하는 것을 용이하게 하기 위해 사용될 수 있다. 일부 실시예들에서, 제어기(404)는 또한 액세서리의 광고로부터의 정보를 캐싱할 수 있고(예를 들어, 상기 표 2로부터의 정보 중 임의의 것 또는 전부), 이 정보는 또한 예를 들어, 후술되는 바와 같은 상태 카운터 "s#"를 사용하여, 액세서리 내의 상태 변화들을 검출하는 데 사용될 수 있다.

[0127] 블록들(436, 438)에서, 제어기(402) 및 액세서리(404)는 커맨드 및 제어 메시지들을 교환하기 시작하여, 제어기(404)가 액세서리(404)를 제어하도록 허용할 수 있다. 일부 실시예들에서, 이들 메시지는, 페어 셋업 프로세스에서 또는 후술되는 바와 같은 후속적인 페어 검증 프로세스에서 확립되는 키들을 사용해 암호화될 수 있다.

[0128] 본 명세서에서 기술되는 검색 프로세스는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 또한, Bonjour 서비스가 디바이스 검색 서비스의 예로서 사용되지만 유사한 개념들이 다른 디바이스 검색 서비스들의 콘텍스트에서 적용될 수 있다.

[0129] 일부 실시예들에서, 특정 액세서리와 페어링할지 여부를 결정하기 이전에, 제어기(404)는 액세서리(402)로부터 액세서리 정의 레코드(또는 그 일부분)를 요청할 수 있다. 예를 들어, 제어기(404)는 액세서리 정의 레코드를 요청하기 위해 액세서리에 메시지(예를 들어, HTTP GET 요청)를 송신할 수 있다. HTTP GET 요청에 사용하는 URL은 균일한 액세서리 프로토콜의 규약에 의해, 또는 액세서리의 광고 내에서(예를 들어, TXT 레코드 내) 명시될 수 있다. 구성에 따라, 액세서리(402)는 페어링되지 않은 제어기로부터의 요청에 응답하여 그것의 액세서리 정의 레코드의 전부 또는 일부를 제공하거나 그 중 아무 것도 제공하지 않을 수 있다. 일부 실시예들에서, 액세서리(402)는 부분적 액세서리 정의 레코드를 제공할 수 있다. 예를 들어, 전술한 바와 같이, 일부 특성들은, 그것들이 단지 페어링된 제어기들에 의해 판독 및/또는 기록될 수 있음을 명시하는 속성들을 가질 수 있고, 그러한 특성들에 관한 정보를 페어링되지 않은 제어기에 제공하는 것은 바람직하지 않을 수 있다. 따라서, 액세서리(402)는 "공개(public)" 특성들을, 즉, 페어링되지 않은 제어기들이 판독 및/또는 기록하도록 허용되는 특성들을, 식별할 수 있으며, 페어링되지 않은 제어기에 제공된 부분적 액세서리 정의 레코드는 적어도 하나의 공개 특성을 갖는 서비스 인스턴스들만을 포함할 수 있고, 공개 및 비-공개 특성들 둘 다를 갖는 서비스 인스턴스의 공개 특성 인스턴스들만을 포함할 수 있다. 일부 실시예들에서, 액세서리 정의 레코드들은 페어링이 확립되기 전에는 전혀 액세스가능하게 되어 있지 않고; 그러한 경우, 페어링할지 여부의 결정은 액세서리의 광고, 제어기의 콘텍스트, 및/또는 사용자 입력에 기초할 수 있다.

[0130] 일부 실시예들에서, 검색 프로세스(400) 또는 유사한 프로세스가 상태 변화들을 검출하는데 사용될 수 있다. 예를 들어, 표 2에 언급된 바와 같이, 상태 번호 "s#"는 상태가 변할 때 증분될 수 있다. 액세서리는 예를 들어, 업데이트된 TXT 레코드를 브로드캐스팅함으로써 상태 변화를 광고할 수 있고, (예를 들어, 프로세스(400)의 블록(432)에서) 이전에 TXT 레코드를 캐싱한 페어링된 제어기는 "s#"의 브로드캐스트 값을 그것의 캐싱된 값과 비교함으로써 변화를 검출할 수 있다.

[0131] 도 4의 프로세스는 제어기 및 액세서리가 TCP/IP를 이용하여 통신하는 경우들에서 사용될 수 있다. 일부 실시예들에서, 균일한 액세서리 프로토콜은 TCP/IP에 추가하여 또는 그 대신에, 블루투스 LE와 같은 다른 전송들을 지원할 수 있다. 이런 경우, 액세서리들은 블루투스 LE를 사용하여 그들의 서비스들을 광고할 수 있다. 예를 들어, 액세서리는 하나 이상의 블루투스 LE 광고 채널들 상에서 광고할 수 있다. 광고 데이터는, 예를 들어, 액세서리에 대한 로컬 이름; 고유 액세서리 식별자; 액세서리가 검색가능하다는 것을 나타내는 플래그들; (후술하는 바와 같은 페어링 서비스를 포함하는) 서비스들 중 적어도 일부에 대한 UUID들; 액세서리 상태의 표시자(표 2의 현재 상태 번호와 유사함); 및 액세서리가 적어도 하나의 제어기와 페어 셋업을 수행했는지 여부에 관한 표시 중 임의의 것 또는 전부를 포함할 수 있다. 이 정보는 전술한 프로세스(400)와 유사한 검색 프로세스에서 사용될 수 있다.

[0132] 예시적인 통신

[0133] 제어기가 액세서리와 페어링을 확립한 후, 제어기는 액세서리에 커맨드-및-제어 메시지들(요청들)을 송신할 수 있다. 이들 요청은 액세서리의 상태에 관한 정보를 획득하고/하거나 액세서리의 상태의 양상들을 변경하기 위해 사용될 수 있다. 일부 실시예들에서, 커맨드-및-제어 메시지들은, 액세서리 정의 레코드 내에 정의된 바와 같은, 특정 서비스 또는 특성를 위한 경로를 반영하는 URL로 어드레싱된 HTTP 요청들일 수 있다. 이러한 실시예들에서, 액세서리는 HTTP 서버로서 기능할 수 있는 한편(리소스들에 대한 요청들을 수신하고 그에 응답함), 제어기는 HTTP 클라이언트로서 기능할 수 있다(서버/액세서리로의 요청들을 생성하고 응답들을 수신함). 일부 액세서리들은 예를 들어, 상이한 인터페이스들 또는 도메인들 상에서 통신을 관리하기 위한, 다수의 HTTP 서버들을 구현할 수 있다는 것에 유의해야 한다. 일부 실시예들에서, 액세서리들 및 제어기는 (예를 들어, 응답을 수신하기 이전에 다수의 HTTP 요청들을 송신하여) 단일 TCP 접속 및/또는 HTTP 파이프라이닝을 통해 다수의 HTTP 요청들을 지원할 수 있다.

[0134] 예를 들어, 도 3a 내지 도 3c에 도시된 액세서리 객체(300)가 주어지면, 제어기는 리소스를 표현하기 위해, 이 경우에는 도 3b에 정의된 문-열림장치 서비스 인스턴스(320)의 문-상태 특성 인스턴스(321)를 표현하기 위해 도 5a의 URL(500)을 구성할 수 있다. URL(500)은 프로토콜-식별 접두부(502)("http://"), 액세서리에 대한 호스트명(504)(이는 전술한 도 4의 검색 프로세스를 통해 획득될 수 있음), 및 로컬 경로(506)를 포함한다. 로컬 경로(506)는 균일한-액세서리 프로토콜을 통해 노출되는 서비스에 대한 참조를 나타내기 위한 프로토콜 키워드(508)("proto/"), 특정 서비스 인스턴스를 식별하기 위한 서비스 인스턴스 식별자(510)("service/<serviceID>"), 및 서비스 인스턴스 내에서 특성 인스턴스를 식별하기 위한 특성 인스턴스 식별자(512)("characteristic/<characteristicID>")를 포함할 수 있다. 도 5a에서, <serviceID> 및 <characteristicID>는 액세서리 객체 내에 정의된 인스턴스 식별자들로 대체된다. 따라서, 도 3a 내지 도 3c를

참조하면, instanceID가 7인 서비스 인스턴스는 차고문 열림장치 서비스 인스턴스(320)이고, instanceID가 8인 특성 인스턴스는 현재-상태 특성 인스턴스(321)일 것이다. 따라서, URL(500)은 차고문 열림장치의 현재 문 상태를 참조하는 것으로서 이해될 수 있다. 유사한 방식으로, 다른 URL들이 서비스들 및 특성들에 대한 instanceID들을 사용해 액세서리 정의 레코드들로부터 생성될 수 있다. 임의의 서비스 인스턴스의 임의의 특성 인스턴스는 이러한 방식으로 식별될 수 있다.

[0135] 제어기는, 특성을 판독(기록)하기 위해 이러한 방식으로 구성된 URL로의 HTTP GET(PUT) 요청을 생성할 수 있다. 또한, URL(500)의 계층 구조는, 제어기가 단일 트랜잭션에서 다수의 특성들 또는 다수의 서비스들로부터 판독하거나 그에 기록할 수 있게 하기 위해, 이용될 수 있다. 예를 들어, 특정 서비스의 모든 특성들을 판독하기 위해 GET 요청을 송신하는 것은 특성 식별자를 생략함으로써 달성될 수 있다. 응답은 도 3a 내지 도 3c와 유사한 포맷의, 특성들을 포함하는 JSON 객체일 수 있다.

[0136] 액세서리 리소스의 현재 상태를 결정하기 위해, 제어기는 도 5a의 URL(500)에 기초한 HTTP GET 요청을 송신할 수 있다. 도 5b는 도 5a의 URL(500)로부터 구성된 GET 요청(520)의 예를 도시한다. GET 요청(520)은 URL(500)의 호스트(504)로 어드레싱되고, 검색될 리소스로서 로컬 경로(526)(URL(500)의 로컬 경로(506)와 유사함)를 명시한다. 이 예에서, 로컬 경로(526)는 단일 특성을 명시하지 않고; 이것은 명시된 서비스 인스턴스의 모든 특성들을 판독하려는 요청으로 해석될 수 있다.

[0137] HTTP GET 요청(520)에 응답하여, 액세서리는 요청된 리소스(상태 정보)를 제공하는 HTTP 응답을 송신할 수 있다. 도 5c는 도 5b의 요청에 응답하여 제공될 수 있는 예시적인 HTTP 응답(530)의 일부분을 도시한다. 응답은, 응답 및 상태("200 OK")를 식별하고, 콘텐츠(534)가 균일한 액세서리 프로토콜에 의해 정의된 바와 같은 JSON 객체로서 포맷되어 있음을 나타내는, HTTP 응답 헤더(532)를 포함한다. 콘텐츠(534)는 요청된 리소스에 대한 상태 정보를 포함하며, 이는 이 예에서 도 3b의 문-열림장치 서비스 인스턴스(320)이다. 서비스 인스턴스(320)는 5개의 특성 인스턴스들을 포함하며, 요청이 모든 특성들을 판독하는 것이기 때문에 5개 특성들 모두의 현재 상태가 보고된다. (단지 2개의 특성이 도 5c에 도시되어 있으며; 나머지가 유사한 방식으로 제시될 수 있음이 이해될 것이다.) 다른 레벨들의 세분성에서의 판독 요청들 및 응답들은 원하는 레벨의 세분성을 반영하도록 경로(526)를 수정함으로써 구성될 수 있다(예를 들어, 단일 특성 인스턴스 또는 모든 서비스 인스턴스들).

[0138] 유사하게, 제어기는 적절한 URL로의 HTTP PUT 요청을 송신함으로써 액세서리에 대한 상태 정보를 변경할 수 있다. 도 5d는 도 5a의 URL(500)로부터 구성된 PUT 요청(540)의 예를 도시한다. PUT 요청(540)는 URL(500)의 호스트(504)로 어드레싱되고, 콘텐츠(542)가 기록될 리소스로서 로컬 경로(546)(URL(500)의 경로(506)와 유사함)를 명시한다. 이 예에서, PUT 요청(540)은 하나의 특성 인스턴스에 기록해서, 경로(546)가 원하는 특성 인스턴스(이는 도 3b의 문-열림장치 목표 상태 특성 인스턴스(322)에 대응함)의 instanceID를 포함하게 된다. 콘텐츠(542)는 기록될 값을 명시한다. 도 3b의 예에서, 특성 인스턴스(322)는 타입이 "com.proto.ch.door-state.target"이고, 전술한 바와 같이, 이 특성 타입에 대한 값 1은 "open" 상태로 매핑될 수 있다. 따라서, PUT 요청(540)은 차고문을 열기 위한 명령어로서 액세서리에 의해 해석될 수 있다.

[0139] 액세서리는 빈(empty) HTTP "204 콘텐츠 없음(No Content)" 응답으로 응답할 수 있고, 예를 들어, 문을 엶으로써, 상태 업데이트를 구현할 수 있다. (문이 능동적으로 열리는 동안에, 액세서리는 문이 열리고 있음을 나타내는 값으로 현재 문 상태 특성 인스턴스(321)를 업데이트하고, 이어서 문이 완전히 열리면 "open"을 나타내는 값으로 업데이트할 수 있다.) 일부 실시예들에서, 액세서리 응답은 예를 들어, 도 5f를 참조하여 후술하는 바와 같이, 콘텐츠를 포함할 수 있다. 예를 들어, 오류가 발생하는 경우, 액세서리는 HTTP 오류 응답으로 응답할 수 있다. 예를 들어, HTTP 오류 코드 400은 잘못된 요청(예를 들어, 신택스 오류 또는 무효한 특성)을 나타낼 수 있고, 오류 코드 403은 금지된 행동(예를 들어, 제어기가 기록 허가를 갖고 있지 않은 리소스에 기록하려는 시도)을 나타낼 수 있고, 오류 코드 500은 액세서리에서의 내부 오류를 나타낼 수 있다. 일부 경우들에서, 오류 응답의 콘텐츠는 오류에 관한 정보를 갖는 "응답" 객체를 포함할 수 있다. 응답 객체의 예가 아래에 기술된다.

[0140] 일부 실시예들에서, 다수의 특성이 단일 PUT 요청을 사용하여 기록될 수 있다. 도 5e는 도 5a의 URL(500)로부터 구성된 PUT 요청(550)의 예를 도시한다. PUT 요청(550)은 PUT 요청(540)과 유사할 수 있는데, 다만, 로컬 경로(556)는, 다수의 특성을 기록하는 것을 허용하는, 서비스 레벨의 세분성 상태에 있다. 도 3의 액세서리 정의(300) 및 도 2a의 특성 정의들을 참조하면, PUT 요청(550)은 차고문을 잠금해제하고 열려는 요청으로서 해석될 수 있다. 이 예가 도시하는 바와 같이, PUT 요청을 사용하여 서비스의 특성들의 서브셋에 기록하는 것이 가능하다. 요청은 단지 기록될 특성들만을 포함할 수 있고; 각 특성 데이터 객체 내에서 타입 및 instanceID 키들의 존재는 모호성을 방지할 수 있다. 또한, 기록될 특성들은 임의의 순서로 열거될 수 있다.

[0141] 액세서리는 각각의 특성의 현재 상태를 포함하는 메시지로 응답할 수 있다. 도 5f는 본 발명의 일 실시예에 따른 예시적인 응답(560)을 도시한다. 응답(560)은, 그 콘텐츠(562)가 각각의 업데이트된 특성(563, 564)의 현재 값을 포함하는, 비어있지 않은(not-empty) HTTP 응답이다. 각각의 특성에 대해, "응답" 객체(565, 566)가 포함되어 있다. 도시된 바와 같이, 응답 객체는 "developerMessage"를 포함할 수 있다. 오류가 발생한 경우, developerMessage는 오류의 평이한-언어 설명(plain-language explanation)을 제공하고, 선택적으로 문제를 정정하기 위한 제안들을 제공할 수 있다. "errorCode"는 (예를 들어, 그에 응답하여 취해질 제어기 행동들을 선택하는 데 사용될 수 있는) 숫자 코드일 수 있다. "moreInfo" 스트링은, 오류에 대해 돕기 위해 참조되어야 하는 프로토콜 규격 또는 기타 문서의 섹션을 식별하는 것과 같은, 추가 정보를 제공할 수 있다. 도 5f에 도시된 예에서, 오류는 발생하지 않았고, 각각의 특성의 상태는 도 5e에서 요청된 상태이다. 오류가 발생하는 경우, 상태는 요청된 상태와 매칭되지 않을 수 있다.

[0142] 유사한 방식으로, 도 5e의 요청(550)과 유사한 PUT 요청이 서비스 또는 다수의 서비스에 기록하는 데 사용될 수 있고, 도 5f의 요청(506)과 유사한 응답이 생성될 수 있다.

[0143] 기술된 바와 같이, 제어기는 PUT 요청을 송신함으로써 액세서리의 상태 변화를 요청할 수 있다. 일부 경우들에서, 상태 변화는, 완료하는 데 시간이 요구될 수 있는 액세서리 기능들(예를 들어, 커맨드들 또는 동작들의 실행)을 호출하는 것을 수반할 수 있다. 응답하는 데 있어서 액세서리 유연성을 허용하기 위해, 일부 실시예들은 일부 특성들 또는 서비스들에 대하여 지연된-응답 거동(deferred-response behavior)을 지원한다.

[0144] 제어기가 HTTP PUT을 사용해 특성에 기록할 때, 액세서리는 요청을 평가하고 (예를 들어, 기록된 특정한 특성에 기초하여) 응답 옵션을 선택할 수 있다. 일부 실시예들에서, 2개의 응답 옵션이 제공된다: "인라인 결과" 및 "질의 결과". 인라인 결과의 경우, 액세서리는 요청된 동작을 수행하고, 이어서 (예를 들어, 도 5e 및 도 5f를 참조하여 전술한 바와 같은) HTTP 응답으로 결과를 반환한다. 응답은 동작을 완료하는 데 필요한 만큼 지연될 수 있다. 질의 결과의 경우, 액세서리는 동작을 수행하기 전에 HTTP 응답을 반환한다. 이 경우, 동작-전(pre-operation) HTTP 응답은, 제어기가 나중에 결과를 얻기 위해 HTTP GET 요청을 송신하는 데 사용할 수 있는, (액세서리에 의해 할당된) "transaction ID"를 포함할 수 있다. 동작-전 HTTP 응답은 또한, 제어기가 GET 요청을 송신하기 전에 대기해야 하는 최소 시간을 나타내는, (또한 액세서리에 의해 할당된) "transaction duration"을 포함할 수 있다. 액세서리는 요청당 기준(per-request basis)으로 응답 옵션을 선택할 수 있다.

[0145] 예시로서, 구성 포트를 여는 기능성을 구현하는 액세서리를 고려한다. 포트는 다양한 특성을 갖는 서비스로서 모델링될 수 있다. 도 5g는 제어기가 포트를 여는 것을 요청하기 위해 부울 true를 기록할 수 있는 특성 인스턴스(571)를 도시한다. 도 5h는 본 발명의 일 실시예에 따른 포트를 여는 것을 요청하기 위해 제어기에 의해 사용될 수 있는 PUT 요청(572)을 도시한다. (특성 인스턴스(571)는 instanceID 22를 갖는 서비스 인스턴스 내에서 정의된다는 것이 이해되어야 한다.) 액세서리는 예를 들어, 제어기 허가들 또는 액세서리의 현재 상태에 따라, 요청을 이행하거나 거부할 수 있다. 액세서리는, 먼저 요청에 따라 작동한 다음에 응답할지(인라인 결과), 또는 응답한 다음에 작동할지(질의 결과)를 선택할 수 있다.

[0146] 액세서리가 먼저 요청에 따라 작동할 것을 선택하는 경우, 이어서 액세서리는 결과를 나타내는 인라인 응답을 송신할 수 있다. 예를 들어, 요청을 이행하는 것은 소켓을 생성하고 포트를 청취하기 시작하는 것을 포함할 수 있다. 액세서리는 구성된 포트에 관한 정보를 그것의 HTTP 응답 내에 전달할 수 있다. 도 5i는, 도 5h의 요청(572)에 대한 인라인 HTTP 응답(573)의 예를 도시한다. 콘텐츠(574)는 요청에 대한 포트 식별자 및 상태 표시자를 포함할 수 있다.

[0147] 액세서리가 먼저 응답할 것을 선택하는 경우, 액세서리는, 제어기가 나중에 결과에 대해 질의하는 데 사용할 수 있는 정보를 포함하는 트랜잭션 응답을 송신할 수 있다. 도 5j는 본 발명의 일 실시예에 따른 트랜잭션 응답(575)의 예를 도시한다. 트랜잭션 응답(575)은, 트랜잭션을 식별하기 위해 액세서리에 의해 생성된 UUID 또는 다른 식별자일 수 있는 트랜잭션 ID(576), 및 트랜잭션 지속기간(577)(예를 들어, 초 단위)을 포함한다. 응답(575)은, 제어기가 5초 기다린 다음에, 응답을 검색하기 위해 액세서리에 질의해야 한다는 것을 나타낸다. 그 시간 동안, 액세서리는 행동을 수행하고 트랜잭션 ID와 함께 결과를 저장할 수 있다.

[0148] 트랜잭션 지속시간이 경과한 후, 제어기는 예를 들어, 첫 번째 요청과 동일한 URL을 사용하여 HTTP GET 요청을 송신함으로써, 액세서리에 질의할 수 있다. 도 5k는 본 발명의 일 실시예에 따른 트랜잭션 상태 질의에 사용될 수 있는 HTTP GET 요청(578)의 예를 도시한다. GET 요청(578)은 트랜잭션 응답(575)으로부터의 트랜잭션 ID를 포함할 수 있다. 이것은, 질의가 액세서리가 저장했을 수 있는 특정 트랜잭션 결과에 대한 것임을, 액세서리에 나타낸다. 액세서리는 트랜잭션의 결과를 포함하는 HTTP 응답으로 응답할 수 있으며; 이 응답은 도 5i의 인라

인 응답(573)과 유사하거나 동일할 수 있다.

[0149] 리소스들에 액세스하기 위한 다른 기술들이 구현될 수 있다. 하나의 대안적인 구현은 비-계층적 URL을 사용하고 액세서리의 각 서비스 인스턴스 및 특성 인스턴스의 고유 인스턴스 ID를 이용할 수 있으며, 이는 HTTP 메시지들의 길이를 줄일 수 있다.

[0150] 예를 들어, 도 6a는 액세서리의 특성들 및 서비스들에 대한 액세스를 허용하도록 구성될 수 있는 단순화된 URL(600)을 도시한다. URL(600)은 프로토콜-식별 접두부(602)("http://"), 호스트명(604)(이는 전술한 도 4의 검색 프로세스를 통해 제공될 수 있음), 및 이 예에서는 "/characteristics"인 로컬 URL(606)을 포함한다. 로컬 URL(606)은 액세서리에 의해 지원된 URL들의 세트로부터 선택될 수 있다. 표 4는, 균일한 액세서리 프로토콜이 본 발명의 일 실시예에 따라 정의할 수 있는 URL들을 열거한다.

[0151] [표 4]

로컬 URL	지원되는 요청들
/accessories	GET: 액세서리 정의 레코드를 획득함
/characteristics	GET: 특성(들)을 질의함 PUT: 특성(들)을 수정함
/identify	POST: 액세서리 자기-식별 루틴을 호출함(적어도 하나의 확립된 페어링을 갖는 액세서리에 의해 무시되거나 거부될 수 있음)
/pair-setup	POST: 페어-셋업 요청
/pair-verify	POST: 페어-검증 요청
/pairings	POST: 페어링들을 추가, 제거, 또는 열거하도록 요청

[0152]

[0153] 이 예에서, 제어기는 도 6a에 도시된 방식으로 구성된 URL로의 요청들을 송신함으로써 액세서리 기능들을 호출할 수 있으며, 이때 로컬 URL은 호출될 기능에 기초하여 표 4로부터 선택된다. /pair-setup, /pair-verify, 및 /pairings URL들은, 제어기와 액세서리 사이에서 페어링들을 확립 및 검증하는 것과 관련하여 사용될 수 있으며; 예들이 아래에 기술된다. 페어링된 제어기는 액세서리 정의 레코드를 획득하기 위해 액세서리의 /accessories URL로의 GET 요청을 송신할 수 있다. /characteristics URL은 액세서리 특성들을 판독 또는 기록하는 것을 수반하는 모든 상호작용들에 사용될 수 있다.

[0154] /identify URL은 예를 들어, 액세서리가 임의의 제어기와 페어링을 확립하기 이전에, 페어링되지 않은 제어기가 액세서리의 자기-식별 루틴을 호출하도록 허용할 수 있다. 액세서리가 임의의 제어기와 페어링되지 않은 경우, 액세서리는 HTTP 204 "콘텐츠 없음" 응답으로 응답하고, 자기-식별 루틴을 호출하는 것으로 진행할 수 있다. 액세서리가 제어기와 페어링을 확립한 경우, 액세서리는 (예를 들어, HTTP 400 "잘못된 요청(Bad Request)" 응답으로) 요청을 거절하여 URL이 유효하지 않다는 것을 나타낼 수 있다. 일부 실시예들에서, 페어링된 제어기는, 도 2i 및 도 3a에 도시된 바와 같이 액세서리 식별 서비스에 포함될 수 있는, 식별 특성(271)에 기록함으로써 액세서리의 자기-식별 루틴을 호출할 수 있다.

[0155] 도 6b는 도 3a 내지 도 3c의 액세서리 객체(300)에 의해 정의된 액세서리의 특성들을 판독하기 위해 URL(600)을 사용해 송신될 수 있는 HTTP GET 요청(610)의 예를 도시한다. GET 요청(610)는 URL(600)의 호스트(604)로 어드레싱되고 로컬 URL(606)을 명시한다. 스트링(612)은 판독될 특성 인스턴스(들)를 명시하는 URL 파라미터이다. 사용되는 포맷은 <accessoryIID>.<characteristicIID(들)>이며, 여기서 <accessoryIID>는 액세서리의 인스턴스 식별자이고, <characteristicIID(들)>은 판독될 특성들에 대한 인스턴스 식별자들의 콤마-분리된 목록이다. 도 3b의 인스턴스 ID들을 참조하면, GET 요청(610)은 현재 문 상태 및 목표 문 상태를 판독하려는 요청으로서 이해될 수 있다. 특성 인스턴스 식별자들의 범위(예를 들어, URL 파라미터 "?1.8-12")와 같은 다른 포맷들이 사용될 수 있다. 이 예에서, 액세서리의 각각의 특성은 고유 인스턴스 식별자를 할당받기 때문에, 서비스 인스턴스 식별자를 명시할 필요가 없다. 다른 실시예들에서, 서비스 인스턴스 식별자는 원하는 경우에 포함될 수 있다.

[0156] 도 6c는 GET 요청(610)에 응답하여 송신될 수 있는 HTTP 응답(620)의 예를 도시한다. 응답(620)은, 응답 및 상태("200 OK")를 식별하고, 콘텐츠(624)가 표준 액세서리 프로토콜에 대한 JSON 객체로서 포맷되어 있음을 나타내는, HTTP 응답 헤더(622)를 포함할 수 있다. 콘텐츠(624)는 각각의 요청된 특성 인스턴스에 대한 상태 정보

(값)를 포함할 수 있다. 이 예에서, 각각의 특성에 대해 전송된 데이터는 단지 값 및 인스턴스 ID를 포함한다. 액세서리의 각각의 특성 인스턴스가 고유 인스턴스 ID를 갖는 한, 이 정보는 요청에 응답하는 데 충분하다.

[0157] 액세서리의 임의의 수의 특성들(상이한 서비스 인스턴스들의 특성들을 포함함)은 단일 GET 요청 및 응답을 사용하여 이러한 방식으로 판독될 수 있다. 또한, 제어기가 다수의 액세서리 객체들을 제공하는 액세서리 서버와 통신하고 있는 경우, 제어기는 예를 들어, <accessoryIID>.<characteristicIID(들)>의 콤마-분리된 목록으로서 특성들을 명시함으로써, 서버 상에서 다수의 액세서리들의 특성들을 판독하려는 단일 GET 요청을 송신할 수 있다. 예를 들어, URL 파라미터("?1.8.9,2.6,7")는 인스턴스 ID1을 갖는 액세서리로부터 특성 인스턴스들 8 및 9를 판독하고, 인스턴스 ID2를 갖는 액세서리로부터 특성 인스턴스들 6 및 7을 판독하려는 요청으로서 액세서리 서버에 의해 이해될 수 있다.

[0158] 특성들은 액세서리의 /characteristics URL로의 HTTP PUT 요청들을 사용해 기록될 수 있다. 도 6d는 액세서리의 임의의 수의 특성들에 기록하는 데 사용될 수 있는 PUT 요청(630)의 예를 도시한다. 기록될 각각의 특성에 대해, 콘텐츠(634)는 액세서리 인스턴스 ID, 특성 인스턴스 ID, 및 새로운 값을 명시할 수 있다. 이 예에서, 2개의 특성이 기록되어 있지만, 배열은 동일한 액세서리 서버 상의 상이한 액세서리들의 특성들을 포함하는, 임의의 수의(하나 이상의) 특성들을 포함할 수 있다.

[0159] 오류가 발생하지 않는 경우, 액세서리는 빈 HTTP "204 콘텐츠 없음" 응답으로 응답할 수 있고, 적절한 행동들을 개시함으로써(예를 들어, 문을 움직이기 위해 차고문 열림장치의 모터를 트리거링함) 상태 업데이트를 구현할 수 있다. 오류가 발생하는 경우, 액세서리는 오류 메시지로 응답할 수 있다. 도 6e는 PUT 요청(630)에 응답하여 송신될 수 있는 오류 메시지(640)의 예를 도시한다. 이 예에서, 콘텐츠(644)는, (다시 액세서리 인스턴스 ID 및 특성 인스턴스 ID를 사용하여) 기록이 시도되었음이 식별된, 각각의 특성 인스턴스를 식별한다. 새로운 값은 표시되지 않는다. 상태 코드들(646, 648)은 어떤 기록들이 실패했는지를 전달하기 위해 사용된다. 이 예에서, 상태 코드(646)(값이 0)는, instance ID 8을 갖는 특성에 관해서 오류가 발생하지 않았음을 나타낸다. 상태 코드(648)(값이 -12345)는, instance ID 9를 갖는 특성에 관해서 오류가 발생했음을 나타낸다. 상태 코드(648)의 값은 오류의 타입을 나타낼 수 있다. 예를 들면, 상태 코드들은 원하는 대로, 불충분한 권한(예를 들어, 제어기가 특성에 기록하도록 인가되지 않음), 리소스에 액세스할 수 없음(예를 들어, 액세서리 전원이 오프됨), 리소스가 분주함(busy), 요청이 지원되지 않음(예를 들어, 판독-전용 특성에 기록하는 것이 시도됨), 값이 무효함(예를 들어, 최소 및 최대 값들을 갖는 특성의 범위 밖임), 및 다른 오류 조건들을 나타내도록 정의될 수 있다.

[0160] 도 5a 내지 도 5k 및 도 6a 내지 도 6e의 이들 커맨드 및 제어 메시지 포맷들 및 시퀀스들은 예시적이고 변형들 및 수정들이 가능하다는 것이 이해될 것이다. HTTP는 임의의 리소스가 언제든지 요청되도록 허용하며; 이에 따라, 제어기는 임의의 순서로 HTTP 요청들을 송신할 수 있고, 액세서리는 수신된 순서로 HTTP 요청들에 응답할 수 있다. 액세서리는 (오류 메시지들을 포함하는) 표준 HTTP 응답들로 응답할 수 있고, 제어기는 확인응답(acknowledgement)(또는 오류의 경우에서의 부정적 확인응답) 메시지들로서 HTTP 응답들을 처리할 수 있다. 일부 실시예들에서, HTTP 리소스들로 액세서리 서비스들 및 특성들을 매핑하는 것은, 제어기들이 정보를 검색하는 방법에 있어서 상당한 유연성을 허용할 수 있다. 일부 실시예들에서, 더 짧은 URL을 갖는 인스턴스 식별자들의 사용은 더 짧은 요청들 및 응답들을 생성할 수 있으며, 이는 대역폭 요건들을 감소시킬 수 있다.

[0161] 또한, 실시예들이 HTTP를 사용하는 것으로 도시되지만, 본 발명은 임의의 특정 프레이밍 프로토콜로 제한되지 않으며; 다른 프로토콜들이 대체될 수 있다. 예를 들어, 제어기와 액세서리가 블루투스 LE를 사용하여 통신하는 실시예들에서, 제어기는 블루투스 LE GATT 계층, 및 각각의 서비스 및 특성에 할당된 UUID를 활용함으로써 특성들로부터 판독하고 그에 기록할 수 있다.

[0162] 위의 예들에서, 제어기는 예를 들어, HTTP 요청을 사용하여, 액세서리에 커맨드-및-제어 메시지(요청)를 송신함으로써 액세서리에의 상태 변경을 개시할 수 있다. 다른 경우들에서, 액세서리 상태 변경은 제어기 이외의 소스에 의해 개시될 수 있으며, 제어기는 이것이 발생할 때 액세서리에 접속되어 있을 수 있거나 접속되지 않을 수도 있다. 예를 들어, 상이한 제어기들이 상이한 시간들에 동일한 액세서리와 상호작용하고 있을 수 있거나, 또는 사용자는 (예를 들어, 차고문 열림장치를 활성화시키기 위해 차고의 열림/닫힘 버튼을 눌러서) 수동으로 액세서리을 동작시킬 수 있다. 따라서, 액세서리 상태는 제어기가 인식하지 않고서 변화할 수 있다. 따라서, 균일한 액세서리 프로토콜의 일부 구현예들은 상태 변화를 제어기에 통지하기 위한 메커니즘들을 제공할 수 있다. 다수의 통지 메커니즘이 동시에 지원될 수 있고, 제어기는 예를 들어, 특성당 또는 서비스당 기준으로 그것이 어떤 메커니즘을 선호하는지를 결정할 수 있다.

[0163] 통지 메커니즘들의 예들은 위의 표 2에 열거되어 있다. 일부 경우들에서, 액세서리는 액세서리의 상태가 변경될 때마다 증분될 수 있는 내부 상태 카운터를 유지할 수 있다. 단일 상태 카운터가 액세서리 레벨로 유지될 수 있거나, 또는 필요에 따라 상이한 상태 카운터들이 상이한 서비스들 또는 특성들에 대해 유지될 수 있다. 다양한 실시예는 후술되는 통지 메커니즘들 중 임의의 것 또는 전부를 지원할 수 있다.

[0164] 도 7은 본 발명의 일 실시예에 따른 "수동적" 통지 프로세스(700)의 예를 도시한다. 프로세스(700)는, 상태 변화가 발생했는지 여부를 결정하기 위해 제어기가 액세서리의 내부 상태 카운터를 판독하는 것을 수반한다. 따라서, 제어기가 액세서리로부터 접속해제되어 있는 동안에 변화가 발생한 사실은, 그것이 이후에 재접속할 때 제어기에 의해 검출될 수 있다. 일부 실시예들에서, 수동적 통지가 디폴트로(예를 들어, 모든 액세서리가 내부 상태 카운터를 유지함) 인에이블되고, 그것을 인에이블하는 데 특별한 동작(예를 들어, 가입)은 필요하지 없다.

[0165] 블록들(706, 708)에서, 제어기(702) 및 액세서리(704)는 접속을 확립할 수 있다. 다양한 실시예에서, 접속을 확립하는 것은 프로세스(400) 및/또는 후술되는 다른 프로세스들(예를 들어, 페어 검증)을 수행하는 것을 포함할 수 있다.

[0166] 블록(712)에서, 액세서리(704)는 그것의 현재 내부 상태 카운터 값을 제어기(702)에 제공할 수 있으며, 제어기(702)는 블록(714)에서 값을 획득하고 저장할 수 있다. 일부 실시예들에서, 제어기(702)는 적절한 리소스(예를 들어, 액세서리 정보 서비스 내에 정의될 수 있는 판독가능한 상태 카운터 특성)로의 HTTP GET 요청을 송신할 수 있다. 블록(712)에서의 액세서리(704)에 의한 응답은 카운터 값을 포함할 수 있다. 제어기(702)는 이 값을 저장할 수 있다. 그 후 어떤 시점에, 제어기(702)는 액세서리(704)로부터 접속해제할 수 있다(블록(716)).

[0167] 블록(718)에서, 제어기(702)가 접속해제한 후, 상태 변화가 액세서리(704)에서 발생할 수 있다. 이에 응답하여, 액세서리(704)는 블록(720)에서 그것의 상태 카운터를 업데이트할 수 있다.

[0168] 그 후, 제어기(702)는 액세서리(704)에 재접속할 수 있다(블록(722)). 블록(724)에서, 액세서리(704)는 현재 내부 상태 카운터 값을 제어기(702)에 제공하고, 제어기(702)는 블록(726)에서 값을 획득한다. 일부 실시예들에서, 이는 적절한 리소스에 대한 제어기(702)에 의한 HTTP GET 요청을 통해 수행될 수 있고; 액세서리(704)에 의한 응답은 카운터 값을 포함할 수 있다. 블록(728)에서, 제어기(702)는 예를 들어, 블록(726)에서 획득된 카운터 값을 블록(714)에서 이전에 획득되고 저장된 카운터 값과 비교함으로써, 상태 변화를 검출할 수 있다. 일부 실시예들에서, 제어기(702)는 오래된 저장된 카운터 값을 새로운 값으로 덮어쓸 수 있다. 그 후, 블록(730)에서, 액세서리(704)는 업데이트된 상태 정보를 제어기(702)에 제공할 수 있으며, 제어기(702)는 블록(732)에서 정보를 획득한다. 일부 실시예들에서, 이것은 제어기(702)에 의한 HTTP GET 요청 및 액세서리(704)에 의한 응답을 사용해 수행될 수 있다. 제어기는 액세서리 정의 레코드를 요청하거나, 또는 단지 상태 변화들이 제어기의 관심대상이 될 특정 특성들을 요청할 것을 선택할 수 있다.

[0169] 도 8은 본 발명의 일 실시예에 따른 "광고형" 통지 프로세스(800)의 예를 도시한다. 프로세스(800)에서, 액세서리는 상태 변화가 발생했음을 광고하기 위해 (예를 들어, 전술한 바와 같은) 디바이스 검색 서비스를 사용할 수 있다. 광고를 검출하는 제어기는 액세서리에 접속하고 업데이트된 상태 정보를 획득할 수 있다.

[0170] 블록들(806, 808)에서, 제어기(802) 및 액세서리(804)는 접속을 확립할 수 있다. 다양한 실시예에서, 접속을 확립하는 것은 프로세스(400) 및/또는 후술되는 페어링 프로세스들을 수행하는 것을 포함할 수 있다. 블록(810)에서, 제어기(802)는 광고형 통지들에 가입하고자 하는 요구를 나타낼 수 있다. 예를 들어, 도 2e를 참조하여 전술한 바와 같이, 각각의 특성은 notificationMode 속성을 가질 수 있으며, 제어기는 notificationMode 속성에 기록함으로써 통지 모드를 명시할 수 있다. 일부 실시예들에서, 액세서리(804)는, 적어도 하나의 제어기가 광고형 통지들에 가입하지 않은 한, 상태 변화들을 광고하지 않으며; 이것은 네트워크 트래픽을 줄일 수 있다.

[0171] 블록(812)에서, 액세서리(804)는 현재 내부 상태 카운터 값을 제어기(802)에 제공할 수 있으며, 제어기(802)는 블록(814)에서 값을 획득하고 저장한다. 일부 실시예들에서, 이는 적절한 리소스에 대한 제어기(802)에 의한 HTTP GET 요청을 통해 수행될 수 있고; 액세서리(804)에 의한 응답은 카운터 값을 포함할 수 있다. 제어기(802)는 이 값을 저장할 수 있다. 그 후 어떤 시점에, 제어기(802)는 액세서리(804)로부터 접속해제할 수 있다(블록(816)).

[0172] 블록(818)에서, 제어기(802)가 접속해제한 후, 상태 변화가 액세서리(804)에서 발생할 수 있다. 이에 응답하여, 액세서리(804)는 블록(820)에서 그것의 내부 상태 카운터를 업데이트할 수 있다. 블록(822)에서, 액세서리(804)는 또한 업데이트된 상태 카운터를 광고할 수 있다. 예를 들어, 위의 표 3에 도시된 바와 같이, 액

세서리는 예를 들어, Bonjour TXT 레코드 등 내에 상태 번호 "s#"를 포함하는 정보를 광고할 수 있다. 액세서리는 필드를 업데이트하고 새로운 TXT 레코드를 브로드캐스트팅함으로써 상태 변화를 광고할 수 있다.

[0173] 블록(824)에서, 제어기(802)가 브로드캐스트들을 청취하고 있다고 가정하면, 제어기(802)는 변화를 검출할 수 있다. 예를 들어, 제어기(802)는 브로드캐스트로부터 현재 상태 카운터를 추출하고, 브로드캐스트 상태 카운터를 저장된 상태 카운터와 비교하고, 액세서리 상태가 변화했음을 나타내는 불일치를 검출할 수 있다. 블록(826)에서, 제어기(802)는 액세서리(804)에 재접속할 수 있다. 블록(828)에서, 액세서리(804)는 업데이트된 상태 정보를 제어기(802)에 제공할 수 있으며, 제어기(802)는 블록(830)에서 정보를 획득한다. 일부 실시예들에서, 이것은 제어기(802)에 의한 HTTP GET 요청 및 액세서리(804)에 의한 응답을 통해 수행될 수 있다.

[0174] 도 9는 본 발명의 일 실시예에 따른 "능동적" 통지 프로세스(900)의 예를 도시한다. 프로세스(900)에서, 액세서리는 상태 정보가 업데이트될 때 제어기에 알리기 위해 제어기에 대한 접속을 개시할 수 있다. 예를 들어, 제어기는 상태 업데이트들을 수신하는 것에 전용되는 디바이스 검색 서비스(예를 들어, Bonjour 서비스)에 서비스 레코드를 등록할 수 있고, 액세서리는 제어기에 대한 접속을 개시하기 위해 제어기의 서비스 레코드를 사용할 수 있다.

[0175] 블록들(906, 908)에서, 제어기(902) 및 액세서리(904)는 접속을 확립할 수 있다. 다양한 실시예들에서, 접속을 확립하는 것은 프로세스(400) 및/또는 후술되는 페어링 프로세스들을 수행하는 것을 포함할 수 있다. 블록(910)에서, 제어기(902)는 능동적 통지들에 가입하고자 하는 요구를 나타낼 수 있다. 예를 들어, 전술한 바와 같이, 각각의 특성은 notificationMode 속성을 가질 수 있고, 제어기는 notificationMode 속성에 기록함으로써 통지 모드를 명시할 수 있다. 일부 실시예들에서, 액세서리(904)는 적어도 하나의 제어기가 능동적 통지들에 가입한 경우에만 능동적 통지에 관련된 동작들을 수행하며; 이것은 네트워크 트래픽을 줄일 수 있다.

[0176] 블록(912)에서, 제어기(902)는 능동적 통지들을 청취할 포트를 셋업할 수 있다. 블록(914)에서, 제어기(902)는 디바이스 검색 서비스에 서비스 레코드를 등록할 수 있다. 예를 들어, Bonjour 서비스가 사용되는 경우, 제어기(902)는 고유 DNS SRV 레코드를 등록할 수 있다. SRV 레코드가 고유한 경우, 제어기는 프로빙(probing), 광고, 또는 TXT 레코드의 제공과 같은 동작들을 회피함으로써, 네트워크 트래픽을 줄일 수 있다. 일 실시예에서, DNS SRV 레코드는 다음의 포맷을 가질 수 있으며:

[0177] <ID>._guid._tcp.local. <TTL> IN SRV <priority> <weight> <port> <target>,

[0178] 여기서 <ID>는 제어기에 대한 고유 식별자(예를 들어, 소문자로 되어 있고 대시(dash)들이 제거된 GUID, UUID, 또는 다른 식별자)이고; "._guid._tcp"는 DNS 서비스 타입이고; "local"은 도메인 이름이고; <TTL>은 SRV 레코드에 대한 존속시간(이는, 예를 들어, 120초일 수 있음); <priority>는 목표 호스트의 우선순위(이는, 예를 들어, 서비스에서 인식되는 가장 높은 우선순위일 수 있음)이고, <weight>는 동일한 우선순위를 갖는 레코드들에 대한 상대적 가중치(이는 예를 들어, 0일 수 있음)이고; <port>는 제어기(902) 상에서 실행되는 균일한 액세서리 프로토콜 서버의 TCP 포트 번호(예를 들어, 블록(912)에서 셋업된 포트)이고; <target>은 균일한 액세서리 프로토콜 서버에 접속하기 위한 IP 주소를 획득하기 위해 사용될 수 있는 DNS 이름이다.

[0179] 블록(916)에서, 제어기(902)는 접속해제할 수 있다.

[0180] 블록(918)에서, 제어기(902)가 접속해제한 후, 상태 변화가 액세서리(904)에서 발생할 수 있다. 이에 응답하여, 블록(920)에서, 액세서리(904)는 예를 들어, 서비스 타입 "._guid._tcp"에 대한 멀티캐스트 DNS SRV의 질의를 송신함으로써, 등록된 서비스의 위치를 찾기 위해 디바이스 검색 서비스에 질의할 수 있다. (일부 실시예들에서, 액세서리(904)는 블록(920)에서 질의를 수행하고 적어도 하나의 제어기가 능동적 통지들에 가입한 경우에만 다음의 동작들을 수행한다.) 블록(922)에서, 제어기(904)는 블록들(912, 914)에서 확립된 DNS 이름 및 포트 식별자로 질의에 응답할 수 있다. 블록(924)에서, 액세서리(904)는 DNS 이름을 주소분해(resolve)하기 위해 포트에 유니캐스트 질의를 송신할 수 있다. 일부 실시예들에서, IPv4 및 IPv6 어드레싱 버전들 둘 모두를 지원하는 액세서리는, IPv4 및 IPv6 어드레싱 둘 모두를 사용하여 질의들을 송신할 수 있고(예를 들어, IPv4에 대해 DNS A 질의, 및 IPv6에 대해 DNS AAAA 질의); 액세서리가 하나의 IP 주소 버전만을 지원하는 경우, 그것은 하나의 질의를 송신할 수 있다.

[0181] 블록(926)에서, 제어기(902)는 그것의 주소분해된 주소들을 갖는 유니캐스트 응답을 송신할 수 있다. 액세서리가 2개의 질의를 송신한 경우, 제어기는 그것이 지원하는 IP 주소 버전(들)에 따라, 둘 중 어느 하나 또는 둘 다에 응답할 수 있다.

[0182] 블록(928)에서, 액세서리(904)는 주소분해된 주소를 사용하여 제어기(902)에 대한 새로운 접속을 개시할 수 있

다. 일부 실시예들에서, 제어기가 IPv4 및 IPv6 주소들 둘 다를 지원하는 경우, IPv6이 바람직할 수 있다. 블록(930)에서, 제어기(902)는 접속을 수락할 수 있다. 접속 시도가 실패할 경우, 액세서리는 재시도할 수 있고; 일부 실시예들에서, 재시도 빈도는 예를 들어, 60초마다 한 번으로 제한될 수 있다.

[0183] 블록(932)에서, 액세서리(904)는 업데이트된 상태 정보를 송신할 수 있고, 블록(934)에서, 제어기(902)는 업데이트된 상태 정보를 수신할 수 있다. 일부 실시예들에서, 이전에 확립된 페어링의 검증(예를 들어, 후술되는 페어 검증 프로세스를 사용함)이, 업데이트된 상태 정보를 송신하기 전에 요구될 수 있다. 이것은, 액세서리가 통지들에 가입한 동일한 제어기에 상태 정보를 보고하고 있다는, 확신을 제공할 수 있다.

[0184] 도 10은 본 발명의 일 실시예에 따른 "이벤트 통지" 프로세스(1000)의 예를 도시한다. 프로세스(1000)에서, 액세서리는 비요청 HTTP 응답(본 명세서에서 EVENT 메시지로도 지칭됨)을, 변화한 특성에 대한 이벤트 통지들을 수신하도록 현재 가입되어 있는 제어기에 송신할 수 있다. 이 예에서, 제어기는 그것이 액세서리에 접속되어 있는 동안 언제든지 통지들에 가입할 수 있고, 액세서리로부터 접속해제할 때 자동으로 가입해제된다.

[0185] 블록들(1006, 1008)에서, 제어기(1002) 및 액세서리(1004)는 접속을 확립할 수 있다. 다양한 실시예에서, 접속을 확립하는 것은 프로세스(400) 및/또는 다른 후술되는 다른 페어링 프로세스들을 수행하는 것을 포함할 수 있다. 블록(1010)에서, 제어기(1002)는 이벤트 통지들에 가입하고자 하는 요구를 나타낼 수 있다. 예를 들어, 전술한 바와 같이, 각각의 특성은 통지-모드 속성을 가질 수 있고, 제어기는 이 속성에 기록함으로써 통지 모드를 명시할 수 있다.

[0186] 도 11a는 특성에 대한 이벤트 통지들에 가입하려는 HTTP PUT 요청(1100)의 예를 도시한다. 콘텐츠(1102)는 (액세서리 인스턴스 ID 및 특성 인스턴스 ID에 의해) 특성을 식별하고 notificationMode 속성에 대한 값 "ev"를 포함한다. 다수의 특성이 단일 가입 요청 내에 열거되고, 일부 실시예들에서, 가입 요청은 하나 이상의 특성에 값들을 기록하려는 요청과 조합될 수 있다. 액세서리는 명시된 특성에 대한 이벤트 통지들에 가입하려는 요청으로서 요청(1100)을 이해할 수 있고, 확인(예를 들어, HTTP 204 "콘텐츠 없음") 또는 오류 응답(예를 들어, 이벤트 통지들이 특성에 대해 지원되지 않는 경우)으로 응답할 수 있다. 유사한 메시지가, 원하는 통지 방법(들)을 명시함으로써 다른 타입들의 통지들(예를 들어, 전술한 바와 같은 능동적 또는 광고형 통지들)에 가입하기 위해 제어기에 의해 사용될 수 있다.

[0187] 가입한 후, 블록(1014)에서, 제어기(1002)는 이벤트-기반 통지로부터 가입해제할 수 있다. 일부 실시예들에서, 제어기(1002)는 그것이 액세서리(1004)로부터 접속해제하는 경우 자동으로 가입해제된다. 그에 추가하여 또는 그 대신에, 제어기(1002)는 액세서리(1004)로부터 접속해제하지 않으면서 새로운 값을 통지-모드 속성에 기록함으로써 명시적으로 가입해제할 수 있다. 제어기(1002)가 (자동으로 또는 명시적으로) 가입해제하는 경우, 임의의 후속 이벤트 통지들은 더 이상 제어기(1002)로 송신되지 않을 것이고, 프로세스(1000)는 블록(1016)에서 종료할 수 있다. (가입해제된 제어기(1002)는, 물론, 다시 가입하기 위해 프로세스(1000)를 실행할 수 있다.) 제어기(1002)를 자동으로 또는 명시적으로 가입해제하는 것은 액세서리(1004)에 의한 전력 소비를 줄이는 데 도움이 될 수 있으며, 이는 액세서리(1004)가 관심대상이 아닌(또는 접속되어 있지 않은) 제어기로의 이벤트 메시지들을 생성하거나, 송신하려고 시도하는 것을 회피할 수 있다.

[0188] 블록(1018)에서, 상태 변화가 액세서리(1004)에서 발생할 수 있으며, 블록(1020)에서, 액세서리(1004)는 그것의 내부 상태 카운터를 업데이트할 수 있고, 내부 상태 카운터를 업데이트하는 것은 예를 들어, 전술한 바와 같이, 수동적 및/또는 광고형 통지들과 관련하여 유용할 수 있다. 또한, 제어기(1002)가 이벤트-기반 통지들에 가입되는 경우, 액세서리(1004)는 제어기(1002)로의 통지를 생성할 수 있다.

[0189] 예를 들어, 블록(1024)에서, 액세서리(1004)는, 임의의 제어기들(예를 들어, 제어기(1002))이 현재, 영향을 받는 특성에 대한 상태 변화들에 대한 이벤트-기반 통지들에 가입되어 있는지 여부를 결정할 수 있다.

[0190] 제어기들이 현재 가입되어 있지 않는 경우, 추가 동작이 취해지지 않고, 프로세스(1000)는 블록(1026)에서 종료할 수 있다. 적어도 하나의 제어기(예를 들어, 제어기(1002))가 현재 가입되어 있는 경우, 이어서 블록(1028)에서, 액세서리(1004)는 업데이트된 상태 정보를 포함하는 이벤트 메시지를 생성할 수 있다.

[0191] 도 11b는 본 발명의 일 실시예에 따른 이벤트 메시지(1120)의 예를 도시한다. 이벤트 메시지(1120)는 HTTP 응답과 구조가 유사할 수 있지만; HTTP 응답과 달리, 이벤트 메시지(1120)는 특정 HTTP 요청에 응답하여 송신되지 않는다. 헤더 섹션(1122)은 메시지(1120)를 이벤트 메시지로서 식별할 수 있고 버전 정보 및 상태 코드를 포함할 수 있다. 바디(1122)는 변화한 특성의 업데이트된 값을 포함할 수 있으며; 본 명세서에서의 다른 예들에서와 같이, 특성은 액세서리 식별자 및 인스턴스 식별자에 의해 식별될 수 있다. 다수의 특성이 변화한 경우, 액

세서리는 단일 이벤트 메시지(1120)에 다수의 특성을 포함할 수 있다는 것이 이해되어야 하며; 즉, 이벤트 메시지들은 합병될 수 있다. 다른 포맷들이 또한 사용될 수 있다.

[0192] 다시 도 10을 참조하면, 블록(1030)에서, 액세서리(1004)는 제어기(1002)에 이벤트 메시지를 송신할 수 있고, 블록(1032)에서, 제어기(1002)는 이벤트 메시지를 수신할 수 있다. 예를 들어, 제어기(1002) 내의 HTTP 스택은 (예를 들어, 헤더(1122)에 기초하여) 이벤트 메시지(1100)를 인식하도록 수정될 수 있다. 제어기(1002)는 수신된 이벤트 메시지로부터, 업데이트된 상태 정보를 추출할 수 있다.

[0193] 전술한 다양한 통지 프로세스는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 또한, 동일한 액세서리는 통지 프로세스들 중 임의의 것 또는 전부를 동시에 지원할 수 있으며, 동일한 상태 변화는, 다양한 제어기가 상태 변화의 시기에 어떤 통지 옵션들에 가입했는지에 따라, 내부 상태 카운터를 업데이트하는 것, 업데이트된 상태 카운터를 광고하는 것, 하나 이상의 가입된 제어기와 접속을 개시하는 것, 및/또는 가입된 제어기에 이벤트 메시지를 송신하는 것 중 임의의 것 또는 전부를 생성할 수 있다. 다른 통지 메커니즘들 및 프로세스들이 도 7 내지 도 10에 도시된 것들에 추가하여 또는 그 대신에 지원될 수 있다.

[0194] 전술한 바와 같이, 제어기들은 예를 들어, 특성의 notificationMode 속성에 기록함으로써, 특성당 기준으로 상태-변화 통지들에 가입할 수 있다. 액세서리는, 임의의 제어기들이 논의되는(in question) 특성에 관련하여 각각의 통지 메커니즘에 가입되어 있는지 여부에 기초하여, 어떤 통지 메커니즘(들)을 수행할지를 결정할 수 있다. 일부 실시예들에서, 수동적 통지는 디폴트 메커니즘이고, 내부 상태 카운터는 임의의 제어기가 특별히 요청한 것에 관계없이 항상 업데이트된다. 광고형, 이벤트, 및/또는 능동적 통지들이 네트워크 트래픽을 발생시킬 수 있기 때문에, 이들 메커니즘의 사용은, 수동적 통지에 의존하는 것이 사용자 경험에 부정적인 영향을 끼칠 경우들로 제한될 수 있다.

[0195] 일부 실시예들에서, 다양한 정책들이, (예를 들어, 광고형, 이벤트, 및/또는 능동적 통지들을 통한) 상태 변화들의 알림에 의해 생성된 네트워크 트래픽을 감소시키기 위해 도입될 수 있다. 예를 들어, 상태 변화 광고는 적어도 하나의 제어기가 광고 통지들에 가입되는 경우들로 제한될 수 있고, 접속들을 개시하기 위해 서비스 레코드들에 대해 질의하는 것은 적어도 하나의 제어기가 능동적 통지들에 가입되는 경우들로 제한될 수 있다. 또한, 액세서리는, 추가로 네트워크 트래픽을 감소시키기 위해, 최소한의 지연 기간(예를 들면, 1 초)을 갖고 모든 광고, 능동적, 또는 이벤트 통지들을 병합하도록 요구될 수 있다. 다른 예로서, 광고들, 이벤트 통지들, 및/또는 액세서리-개시된 접속들은 빈도에 있어서(예를 들어, 30초마다 한 번으로) 제한될 수 있다. 네트워크 트래픽을 최소화하는 수동적 통지가, 디폴트로 사용될 수 있다. 이러한 제한들은 설계 선택의 문제이며, 필요에 따라 변경되거나 제거될 수 있다. 제한들이 부과되고 액세서리가 제한들을 위반하는 경우, 위반을 검출하는 제어기는 사용자에게 잘못된 거동(mishavior)을 알리고/알리거나 액세서리와의 그것의 페어링을 종료할 수 있다.

[0196] 예시적인 페어링 프로세스

[0197] 일부 실시예들에서, 액세서리와 제어기 사이의 통신들을 안전하게 하는 것이 유용할 수 있다. 예를 들어, 도 1을 참조하면, 제어기(102)는 예를 들어, 전술한 바와 같이 요청 메시지들을 송신함으로써, 문(104)을 잠금해제하거나 차고(106)를 여는 데 사용될 수 있다. 통상의 기술자는 이러한 동작들이 특정한 인가된 제어기들로 제한되어야 한다는 것을 이해할 것이다.

[0198] 따라서, 본 발명의 일부 실시예들은 인증된 페어링 및 단대단 암호화와 같은 보안 조치들을 제공한다. 인증된 페어링은 액세서리와 제어기 사이의 페어링을 확립하는 것(페어 셋업으로도 지칭됨)의 일부로서 발생할 수 있으며, 그 동안에 액세서리 및 제어기는 (예를 들어, 암호화 기술들을 사용하여) 보안 프레임워크를 확립할 수 있고 그 프레임워크 내에서 장기간 공개 키들을 교환할 수 있다. 일부 실시예들에서, 페어 셋업은 정보의 대역외 교환(예를 들어, 셋업 코드)을 통합할 수 있고, 대역외 교환은, 액세서리에게 그것이 특정 제어기와 페어링해야 한다는 것을 검증하고/하거나 제어기에게 그것이 특정 액세서리와 페어링해야 한다는 것을 검증하기 위한, 사용자 입력(또는 사용자 행동)을 통합할 수 있다. 이러한 방식으로 공개 키들을 교환한 후, 제어기 및 액세서리는 키들을 저장하고, 페어링이 확립되었음을 검증하기 위해 이들을 나중에 사용할 수 있다. 단대단 암호화는, (예를 들어, 페어링을 검증한 이후에) 액세서리 및 제어기 둘 모두 내에서 세션 키들을 생성하는 것 및 각각의 메시지가 송신 디바이스를 떠나기 전에 이를 암호화하기 위해 세션 키들을 사용하는 것을 포함할 수 있어서, 메시지가 인터셉트되는 경우, 인터셉터가 그것을 사용할 수 없게 되도록 할 것이다. 따라서, 링크 계층 또는 전송 계층에서 통신 네트워크를 안전하게 할 필요가 없다. 예를 들면, 새로운 세션 키들이, (예를 들어, 후술되는

바와 같이 페어 검증 프로세스를 사용하여) 액세서리와 제어기가 재접속할 때마다 생성될 수 있다. 또한, (예를 들어, 저장된 장기간 공개 키를 제거함으로써) 확립된 페어링들을 제거하기 위한 메커니즘들이 제공되어, 특정 액세서리를 제어하도록 한번 인가되었던 제어기가 인가되지 않은 것으로 되게 할 수 있다.

[0199] 일부 실시예들에서, (후술되는 키들 중 임의의 것 또는 전부를 포함하는) 암호화 키들은, 디바이스를 위한 데이터를 안전하게 저장할 수 있는 전용 집적회로 칩("보안 저장 요소"로도 지칭됨)과 같은 "보안 요소" 내에 배타적으로 저장될 수 있다. 보안 요소는, 수신된 장기간 공개 키들, 및/또는 페어링 관계가 확립된 다른 디바이스들을 식별하는 다른 정보의 지속적이고 안전한 저장을 제공하는 데 사용될 수 있다. 이것은 공격자가, 적절한 셋업 프로세스를 거치지 않으면서 페어링을 추가하거나(이는 제어기의 불법 사용을 생성할 수 있음), 또는 인가 없이 페어링을 제거하는 것(이는 제어기의 합법적인 사용을 막을 수 있음)을 방지하는 데 도움을 줄 수 있다. 또한, 일부 실시예들에서, 보안 요소는 또한 로직 회로를 포함하여, 그것이 액세서리 또는 제어기의 메인 프로세서에 대한 코프로세서(또는 "보안 컴퓨팅 요소")로서 작용하도록 할 수 있다. 보안 요소는, 입력들 상에서 암호화 동작들을 수행하기 위해 다양한 입력들 및 명령어들을 수신할 수 있다. 보안 요소는 동작들을 수행하고 출력들을 제공할 수 있다. 본 명세서에서 기술된 암호화 동작들(예컨대, 키들을 생성하는 것, 비밀 키로 서명하는 것, 및 키들과 관련된 다른 동작들) 중 임의의 것 또는 전부는 보안 컴퓨팅 요소에 위임될 수 있다. 본 명세서에서 사용된 바와 같이, "보안 요소"는 보안 저장 및/또는 보안 컴퓨팅의 임의의 조합을 제공할 수 있다.

[0200] 일부 실시예들에서, 페어링을 지원하는 액세서리는 그것의 액세서리 모델의 일부로서 페어링 프로파일을 가질 수 있다. 본 명세서에서 기술된 다른 서비스들과 유사하게, 페어링 프로파일은 특성들의 집합으로서 정의될 수 있다. 일부 실시예들에서, 균일한 액세서리 프로토콜은 모든 액세서리들이 페어링 프로파일을 갖는 것을 명시할 수 있다. 다른 실시예들에서, 페어링 프로파일은 선택적 특징일 수 있지만, 프로토콜은, 액세서리가 페어링 프로파일을 갖는 경우, 제어기가 임의의 커맨드-및-제어 메시지들을 교환하기 전에 액세서리와 페어링하도록 요구될 수 있다는 것을, 명시할 수 있다. 페어링 요건들의 추가 조정이 또한 가능하다. 예를 들어, 액세서리의 페어링 프로파일은 페어링을 요구하는 특정 서비스 인스턴스들을 식별하여, 페어링 없이 액세서리의 서비스들 중 일부에 대한 액세스를 허용할 수 있다.

[0201] 도 12는 본 발명의 일 실시예에 따른 페어링 프로파일을 위한 예시적인 특성들(1201 내지 1205)을 도시한다. 포맷은 도 2a와 대체로 유사하다. 본 명세서에서 기술된 다른 특성들과 마찬가지로, 특성들(1201 내지 1205)은 예를 들어, HTTP GET 요청들 등을 이용하여 제어기에 의해 판독될 수 있고; 기록(허용되는 경우)은 예를 들어, HTTP PUT 또는 HTTP POST 요청들 등을 이용하여 수행될 수 있다.

[0202] 페어링 상태 요청(pairing state request) 특성(1201)은 페어링 프로세스의 상태의 변화를 요청하기 위해 (예를 들어, 후술되는 페어 셋업, 페어 검증, 페어 추가, 및/또는 페어 제거 프로세스들 동안에 다양한 요청을 송신하기 위해) 제어기에 의해 기록될 수 있다. 일부 실시예들에서, 제어기는 특성(1201)에 기록하기보다는 (예를 들어, 위의 표 4에 도시된 바와 같이) 페어링 URL로의 HTTP POST 요청을 송신함으로써 페어링 프로세스의 상태의 변화를 요청할 수 있다. 페어링 프로세스 상태들 및 요청들의 예들이 특정 페어링 프로세스들과 관련하여 후술된다.

[0203] 특징 플래그(feature flags) 특성(1202)은 액세서리에 의해 지원되는 페어링 특징들을 정의하는 비트마스크일 수 있다. 예를 들어, 후술되는 바와 같이, 다양한 액세서리는 셋업-코드-기반 페어링(이는 페어링이 발생해야 한다는 것을 확인하기 위해, PIN과 같은 셋업 코드를 입력하도록 사용자에게 요구할 수 있음), 인증서-기반 페어링(이는 후술되는 바와 같이, 디바이스들 중 어느 하나 또는 둘 다 내의 인증 칩들을 사용하여 제공될 수 있는 인증서 인프라구조를 사용할 수 있음), 및/또는 위임된 페어링(이는 이미-페어링된 제어기가, 다른 제어기가 페어링되어야 한다는 것을 검증할 수 있게 함)을 지원할 수 있다. 일부 실시예들에서, 특징 플래그 특성(1202)은 또한, 액세서리가 현재, 새로운 페어링이 페어 셋업을 사용하여 확립될 수 있는 모드에 있는지 여부를 나타내는 비트를 포함할 수 있다. 특성(1202)은 제어기에 의해 판독되지만 기록될 수 없으며, 제어기는 페어 셋업을 수행하는 방법 및 수행할지 여부를 결정하는 데 정보를 사용할 수 있다.

[0204] 페어링 현재 상태(pairing current state) 특성(1203)은 페어링 프로세스의 현재 상태, 예를 들어, 오류가 발생했는지 여부 또는 후술되는 다양한 다른 상태들을 나타낼 수 있다. 그것은 제어기에 의해 판독될 수 있지만 기록될 수는 없다.

[0205] 페어링 목록(pairing list) 특성(1204)은 액세서리와 확립된 모든 페어링들의 목록을 저장할 수 있다. 예를 들어, 각각의 페어링에 대해, 페어링된 제어기의 (셋업-코드-기반 페어링을 위한) 공개 키 및/또는 (인증서-기반 페어링을 위한) 인증서뿐만 아니라 각각의 제어기가 어떤 허가들을 갖는지를 나타내는, TLV 항목이 생성될 수

있다. 이들 TLV 항목은 서브-TLV들 단일 최상위 TLV와 같이 (분리자들과) 함께 패킹될 수 있다. 일부 실시예들에서, 액세서리는 요청 제어기로 최상위 TLV를 송신하기 전에 그것을 암호화한다.

[0206] 페어링 ID 특성(1205)은 액세서리에 대한 전역 고유 식별자, 예컨대, MAC 주소, 액세서리의 공개 키의 일부분, (예를 들어, 후술되는 바와 같은) 액세서리 "사용자명" 등일 수 있다.

[0207] 일부 실시예들에서, 특성들(1201 내지 1205)은, 적어도 액세서리가 적어도 하나의 제어기와 페어링을 확립할 때까지는 페어링되지 않은 제어기들에 보여질 수 있는, 액세서리 페어링 서비스 인스턴스를 정의함으로써 제어기들에 노출될 수 있다. 예를 들어, 액세서리가 통신 전송로서 블루투스 LE를 사용하는 경우들에서, 액세서리는 블루투스 LE를 통해 그것의 페어링 서비스를 광고할 수 있다.

[0208] 페어링 서비스에 추가적으로 또는 그 대신에, 액세서리는 제어기들이 페어링 기능성에 액세스하기 위해 참조할 수 있는 하나 이상의 URL을 정의할 수 있다. 예를 들어, 위의 표 4를 참조하면, 제어기는 페어 셋업 프로세스 동안에 요청들을 행하고 연관된 정보를 제공하기 위해 URL /pair-setup로의 HTTP POST 요청을 송신할 수 있다. 제어기는 페어 검증 프로세스 동안에 요청들을 행하고 연관된 정보를 제공하기 위해 URL /pair-verify로의 HTTP POST 요청을 송신할 수 있다. 제어기는 페어링들을 관리하기 위해, 예를 들어, 페어 추가 및 페어 제거 프로세스들을 개시하기 위해 또는 액세서리에 대한 확립된 페어링들의 목록을 검색하기 위해, URL /pairings로의 HTTP POST 요청을 송신할 수 있고; POST 요청은 특정 요청을 나타내는 데이터 객체를 포함할 수 있다.

[0209] 액세서리와 제어기 사이에 페어링을 확립하기 위한 프로세스들("페어 셋업"으로 지칭됨)의 예들이 이제 기술될 것이다. 이들 또는 다른 프로세스들 중 임의의 것이, 예를 들어, 도 4의 프로세스(400)의 블록들(428, 430)에서 구현될 수 있다. 이에 따라, 이하에서, 액세서리 및 제어기가 이미 검색을 수행하였고, 안전하지 않을 수 있는 통신 채널을 열었다고 가정한다. 또한, 사용자가 제어기에, 그것이 특정 액세서리와 페어링해야 한다는 것을, 이미 확인했다고 가정한다.

[0210] 일부 실시예들에서, 페어 셋업은 액세서리가 페어링 모드에 있는 경우에만 허용된다. 페어링 모드에 액세서리를 두는 것은, 액세서리와 사용자의 물리적 접촉을 수반할 수 있다. 예를 들어, 사용자는, 액세서리 상의 리셉터클 내에 물리적 물체(예컨대, 키)를 삽입하는 것, 액세서리 상의 스위치를 "페어링" 위치로 이동시키는 것 등을 수행하도록 요구될 수 있다. 일부 실시예들에서, 페어 셋업은 액세서리가 어떠한 제어기와도 확립된 페어링을 갖지 않는 경우에만 허용된다. 액세서리가 하나의 확립된 페어링을 가지면, 추가적인 페어링들은 후술되는 바와 같이 페어 추가 프로세스(또는 다른 위임된 페어링 프로세스)를 사용하여 확립될 수 있다.

[0211] 도 13a 내지 도 13c는 본 발명의 일 실시예에 따른 제어기(1302) 및 액세서리(1304)를 위한 셋업-코드-기반 페어 셋업 프로세스(1300)를 도시한다. 셋업-코드-기반 페어 셋업에서, 사용자는 양방향 인증을 제공하기 위해 제어기에서 액세서리의 셋업 코드(이는 예를 들어, 그 액세서리에 대해 특정적이고 쉽게 추측되지 않는 8자리 식별 번호일 수 있음)를 입력하도록 요구될 수 있다. 보안을 위해, 액세서리의 셋업 코드는 액세서리에 의해 디스플레이 상에 보여지거나 사용자에 의해 액세서리에 수동으로 제공될 수 있거나(예를 들어, 액세서리 상에서 물리적 다이얼들을 설정하거나, 키패드에 숫자들을 타이핑함으로써), 또는 셋업 코드는 액세서리 하우징 또는 패키징 상의 또는 그 내부의 어딘가에(예를 들어, 눈에 띄지 않는 표면의 라벨 상에) 인쇄되어 있을 수 있다. 액세서리 제조사들은, 공개적으로-액세스가능한 정보(예를 들어, 일련 번호, MAC 주소, 제조사 이름, 제조 일자 등)로부터 셋업 코드를 도출하지 않음으로써, 보안을 강화할 수 있다. 본 명세서의 예들에서, 페어 셋업은 보안 원격 패스워드(Secure Remote Password, "SRP") 프로토콜(http://srp.standford.edu에 기록되어 있음)를 활용하지만; 다른 프로토콜들이 사용될 수 있다.

[0212] 도 13a를 먼저 참조하면, 블록(1306)에서, 제어기(1302)는 연관된 사용자명("userC")을 찾아볼 수 있다. 사용자명은 하나의 제어기를 다른 것과 구별하는 데 도움을 주도록 액세서리에 의해 사용될 수 있는 제어기(1302)의 임의의 식별자 및/또는 제어기(1302)의 인가된 사용자를 포함할 수 있다. 블록(1308)에서, 제어기(1302)는 예를 들어, 적절한 URL로의 HTTP POST 요청으로서, 페어 셋업 시작 요청을 송신할 수 있다. 페어 셋업 시작 요청은, 페어 셋업 시작 요청이 송신되었음을 나타내는 상태 표시자(일부 실시예들에서, 이 상태 표시자 및 후속적인 상태 표시자는 제어기에 의해 도 12의 페어링 상태 요청 특성(1201)에 기록될 수 있음), 셋업-코드-기반 페어링 방법이 사용 중임을 나타내는 방법 표시자, 및 제어기 사용자명 userC를 포함할 수 있다.

[0213] 블록(1310)에서, 액세서리(1304)는 페어 셋업 시작 요청을 스로틀링(throttle)할지 여부를 결정할 수 있다. 예를 들어, 임의의 추측(random guessing)에 기초한 공격들을 저지하기 위해, 액세서리(1304)는 지수적인 스로틀링(exponential throttling)을 구현할 수 있으며, 이때 다음 시작 요청을 기다리는 시간은 각각의 성공적이지

못한 페어 셋업 시도 후에 두 배로 된다(예를 들어, 지난 n회의 시도가 성공적이지 못한 경우, (n+1)번째 시도는 적어도 2^{n-1} 초 기다려야 한다). 스로틀링은 세션당 또는 접속당 대신에 전역적으로 적용될 수 있다. 스로틀링 시간은 성공적인 페어 셋업 시도 이후 리셋될 수 있다. 따라서, 블록(1310)에서, 스로틀링이 유효하고 스로틀링 시간이 마지막 시도 이래로 경과하지 않은 경우, 액세서리(1304)는 블록(1312)에서 스로틀링 메시지를 송신할 수 있다. 스로틀링 메시지는 제어기가 재시도하기 전에 기다려야 하는 시간을 나타낼 수 있다. 블록(1314)에서, 스로틀링 메시지가 수신되는 경우, 제어기(1302)는 (적절한 시간을 기다린 후) 재시도할지 여부를 결정할 수 있다. 제어기(1302)가 재시도하기로 결정하는 경우, 프로세스(1300)는 적절한 대기 시간 후에 새로운 페어 셋업 시작 요청을 송신하기 위해 블록(1308)로 복귀할 수 있다.

[0214] 블록(1310)에서, 액세서리(1304)가 요청을 스로틀링하지 않는 경우, 프로세스(1300)는 블록(1316)으로 진행하고, 여기서 액세서리(1304)는 예를 들어, SRP_new(SRP6a_server_method())와 같은 적절한 SRP 프로토콜 함수들을 호출함으로써 SRP 세션을 생성하고; (SRP_set_username 또는 SRP_set_user_raw를 이용해) 제어기의 사용자명을 수신된 값 userC로 설정하고; SRP_set_params를 이용해 설정될 수 있는 랜덤 솔트(random salt)(예를 들어, 적어도 16 바이트)를 생성하고; (예를 들어, SRP_set_auth_password 또는 SRP_set_auth_password_raw를 이용해) 그 자신의 셋업 코드를 선택 및 설정하고; (예를 들어, SRP_gen_pub을 이용해) 공개 키("pkA")를 생성할 수 있다. 공개 키 pkA는 프로세스(1300)의 지속시간 동안 사용된 다음에 폐기되는 단기간 키 쌍(비밀 키 "skA"와 함께)의 일부일 수 있다.

[0215] 액세서리는 다양한 기술을 이용하여 셋업 코드를 선택할 수 있다. 예를 들어, 셋업 코드는 EEPROM에 미리 프로그램될 수 있다(그리고, 액세서리 또는 라벨 등 사용자가 찾을 수 있는 곳 상에 인쇄될 수 있다). 대안적으로, 프로세스(1300)에 앞서(또는 동안에), 사용자-선택된 셋업 코드는 액세서리 상에 또는 액세서리 내에 제공된 기계적 또는 전자 입력 디바이스들을 사용하여 사용자에 의해 액세서리 내에 입력될 수 있다. 또 다른 옵션으로, 액세서리가 (예를 들어, 셋업 코드를 디스플레이하거나, 또는 다른 식으로 사용자에게 그것을 시그널링함으로써) 사용자에게 셋업 코드를 알리는 능력을 갖는다면, 액세서리는 페어 셋업 프로세스(1300)를 실행하는 각 경우 동안에 랜덤 셋업 코드를 생성할 수 있다. 다른 기술들이 또한 사용될 수 있다.

[0216] 블록(1318)에서, 액세서리(1304)는 랜덤 솔트 및 공개 키 pkA를 제어기(1302)에 송신할 수 있고, 제어기(1302)는 블록(1320)에서 그것들을 수신할 수 있다. 블록(1318)에서 랜덤 솔트 및 공개 키 pkA를 송신하면, 액세서리(1304)는 액세서리의 공개 키가 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0217] 블록(1322)에서, 제어기(1302)는 사용자로부터 액세서리의 셋업 코드를 획득할 수 있다. 예를 들어, 제어기(1302)는 액세서리의 셋업 코드를 입력하도록 사용자에게 프롬프트하는 사용자 인터페이스 스크린을 제시할 수 있다. 셋업 코드가 대역 외로, 즉, 페어 셋업 프로세스를 수행하는 데 사용되고 있는 통신 채널에 독립적인 통신 채널을 통해 제어기(1302)로 전달되는 한, 다른 기술들이 사용될 수 있다. 예를 들어, 일부 실시예들에서, 사용자는 제어기(1302)의 온보드 카메라를 동작시켜, 셋업 코드가 액세서리 상에 나타날 때 셋업 코드의 이미지를 캡처할 수 있고(이미지는 인간-판독가능한 표현 또는 기계-판독가능한 표현, 예컨대 바코드 또는 QR(quick-response) 코드 등을 포함할 수 있음), 제어기(1302)는 셋업 코드를 추출하기 위해 이미지 처리를 수행할 수 있다. 액세서리(1304)로부터 제어기(1302)로 셋업 코드를 제공하는 데 사용가능한 기술들의 다른 예들에는: 제어기(1302)가 액세서리(1304)에 물리적으로 근접하게 배치되거나 유지되는 동안의, 액세서리(1304)로부터 제어기(1302)로의 근거리 무선 통신(NFC) 전송; 제어기(1302)에 의해 검출가능한 액세서리(1304)에 의한 음파 또는 초음파 전송; 고속 광 시그널링(예를 들어, 제어기(1302)의 카메라 또는 광 검출기의 시야 내에서 액세서리(1304)에 의해 생성된 광 펄스들의 시퀀스); 또는 액세서리(1304) 또는 셋업 코드를 저장하는 연관된 디바이스를 제어기(1302)의 커넥터 인터페이스에 물리적으로 연결하는 것이 포함될 수 있다.

[0218] 블록(1324)에서, 제어기(1302)는 예를 들어, SRP_new(SRP6a_server_method())와 같은 적절한 SRP 프로토콜 함수들을 호출함으로써 SRP 세션을 생성하고; (SRP_set_username 또는 SRP_set_user_raw를 이용해) 제어기의 사용자명을 설정하고; (SRP_set_params를 이용해) 액세서리로부터 수신된 솔트를 설정하고; (예를 들어, SRP_gen_pub을 이용해) 자신의 공개 키("pkC")를 생성하고; 사용자-입력된 셋업 코드를 이용해(예를 들어, SRP_set_auth_password 또는 SRP_set_auth_password_raw를 이용해) SRP 패스워드를 설정하고; (예를 들어, SRP_compute_key를 이용해) 공유 비밀("inputKey")을 계산할 수 있다. 액세서리 공개 키 pkA와 마찬가지로, 공개 키 pkC는 프로세스(1300)의 지속시간 동안 사용된 다음에 폐기되는 단기간 키 쌍(비밀 키 "skC"와 함께)의 일부일 수 있다.

[0219] 블록(1326)에서, 제어기(1302)는 (예를 들어, SRP_respond를 이용하여) 자신의 아이덴티티를 증명하기 위해 제

어기 증명("proofC")을 생성할 수 있다. 블록(1328)에서, 제어기(1302)는 예를 들어, 적절한 URL에 대한 HTTP POST 요청으로서, 공개 키 pkC 및 증명 proofC를 포함하는 검증 요청을 액세서리에 송신할 수 있다. 블록(1328)에서 요청을 송신하면, 제어기(1302)는 제어기의 증명이 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0220] 블록(1330)에서, 액세서리(1304)는 제어기 증명 proofC를 검증할 수 있다. 예를 들어, 액세서리(1304)는 (예를 들어, SRP_compute_key를 사용하여) 공유 비밀("inputKey")을 계산하고, 공유 비밀을 사용하여 (예를 들어, SRP_verify를 사용하여) proofC를 검증할 수 있다.

[0221] 도 13b에 도시된 바와 같이, 블록(1332)에서, 블록(1330)에서의 검증이 실패하는 경우, 이어서 블록(1334)에서, 액세서리(1304)는 제어기(1302)에 오류 메시지를 송신할 수 있다. 블록(1336)에서, 제어기(1302)가 오류 메시지를 수신하는 경우, 프로세스(1300)는 블록(1338)에서 종료할 수 있다. 일부 실시예들에서, 제어기는 사용자에게 오류를 보고할 수 있다.

[0222] 블록(1332)에서, 검증이 성공하는 경우, 이어서 블록(1340)에서, 액세서리(1304)는 (예를 들어, SRP_respond를 사용하여) 자신의 아이덴티티를 증명하기 위해 액세서리 증명("proofA")을 생성할 수 있다. 블록(1342)에서, 액세서리(1304)는 액세서리 증명 proofA를 제어기(1302)에 송신할 수 있고, 제어기(1302)는 블록(1344)에서 proofA를 수신할 수 있다. 블록(1342)에서 proofA를 송신하면, 액세서리(1304)는 액세서리의 증명이 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0223] 블록(1346)에서, 제어기(1302)는 (예를 들어, SRP_verify를 사용하여), proofA를 검증할 수 있다. 블록(1348)에서, 검증이 실패하는 경우, 프로세스(1300)는 종료할 수 있다(블록(1350)). 검증이 블록(1348)에서 성공하는 경우, 액세서리 및 제어기는 이제 각각 인증된 공유 비밀을 소유하게 된다.

[0224] 따라서, 블록(1352)에서, 제어기(1302)는 공유 비밀 inputKey로부터 새로운 암호화 키("Key")를 생성할 수 있다. 예를 들어, 제어기(1302)는 inputKey, 랜덤 솔트, 및 추가 정보 항목(이는 제어기(1302) 내에 미리 프로그램될 수 있음)을 입력들로서 사용하는 512-비트 해시들에 대한 SHA(Secure Hash Algorithm) 버전 2(IETF RFC 2104 내에 정의된, HMAC-SHA-512)를 구현하는 HMAC-기반 키 도출 함수를 사용할 수 있다. 블록(1354)에서, 제어기(1302)는 Key를 사용하여 자신의 장기간 공개 키(LTPKC)를 암호화할 수 있다. 장기간 공개 키는, 제어기 상에서 (예를 들어, 전술한 바와 같이 보안 요소 내에) 지속적으로 저장되고, 프로세스(1300)에서 이전에 생성된 단기간 공개 키 pkC와는 관련 없는, 키일 수 있다. 암호화는, 키로서의 Key, 메세지로서의 LTPKC, 및 임의 값(nonce)를 갖고 ChaCha20-Poly1305(IETF Internet Draft draft-agl-tls-chacha20poly1305-03에서 설명되며, https://tools.ietf.org/html/draft-agl-tls-chacha20poly1305-03에서 이용가능함)와 같은 암호화-및-인증 알고리즘을 사용하여, 암호화된 데이터 edataC 및 인증 태그 authTagC를 생성할 수 있다.

[0225] 블록(1356)에서, 제어기(1304)는 edataC 및 authTagC를 액세서리(1304)에 송신할 수 있고; 제어기(1302)는 또한 LTPKC가 액세서리에 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0226] 블록(1358)에서, edataC 및 authTagC를 수신하면, 액세서리(1302)는 제어기(1302)가 블록(1352)에서 사용한 것과 동일한 방법을 사용하여 암호화 키("Key")를 생성할 수 있다. 모든 것이 이 지점까지 정확하게 갔다면, 그것은 동일한 Key이어야 한다. 블록(1360)에서, 액세서리(1302)는 수신된 authTagC를 검증할 수 있다.

[0227] 도 13c에 도시된 바와 같이, 블록(1362)에서, 블록(1360)에서의 검증이 실패하는 경우, 이어서 블록(1364)에서, 액세서리(1304)는 제어기(1302)에 오류 메시지를 송신할 수 있다. 블록(1366)에서, 제어기(1302)가 오류 메시지를 수신하는 경우, 프로세스(1300)는 블록(1368)에서 종료할 수 있다. 일부 실시예들에서, 제어기는 사용자에게 오류를 보고할 수 있다.

[0228] 블록(1362)의 결과가 성공인 경우, 이어서 블록(1370)에서, 액세서리(1304)는 LTPKC를 복호화할 수 있고, 블록(1372)에서, 액세서리(1304)는 LTPKC 및 제어기의 사용자명 userC를 페어링된 제어기 레코드로서 지속적으로 저장할 수 있다. 이러한 저장소는 전술한 바와 같이 보안 요소 내에 있을 수 있다.

[0229] 블록(1374)에서, 액세서리(1304)는 액세서리의 장기간 공개 키(LTPKA) 및 액세서리와 연관된 사용자명을 포함하는 데이터 객체를 구축할 수 있다. 액세서리의 장기간 공개 키는, (예를 들어, 액세서리(1304)의 보안 요소 내에) 지속적으로 저장되고, 프로세스(1300)에서 이전에 생성된 단기간 공개 키 pkA와는 관련 없는, 키일 수 있다. 제어기 사용자명 userC과 같이, 액세서리 사용자명 userA는 하나의 액세서리를 다른 것과 구별하는 데 도움을 주도록 제어기에 의해 사용될 수 있는 액세서리(1304)의 임의의 식별자 또는 액세서리(1304)의 인가된 사용자를 포함할 수 있다. 블록(1376)에서, 액세서리(1304)는 edataA 및 authTagA를 생성하기 위해 블록

(1374)에서 생성된 데이터 객체를 암호화할 수 있다. 블록(1354)에서 제어기(1302)에 의해 사용된 것과 동일한 암호화 알고리즘이 사용될 수 있다. 블록(1378)에서, 액세서리(1304)는 제어기(1302)에 edataA 및 authTagA를 송신할 수 있고; 액세서리(1304)는 또한 LTPKA가 제어기에 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0230] 블록(1380)에서, edataA 및 authTagA를 수신하면, 제어기(1302)는 수신된 authTagA을 검증할 수 있다.

[0231] 블록(1382)에서, 블록(1380)에서의 검증이 실패하는 경우, 이어서 블록(1384)에서, 프로세스(1300)는 종료할 수 있고, 제어기(1304)는 사용자에게 오류를 보고할 수 있다. 검증이 성공하는 경우, 이어서 블록(1386)에서, 제어기(1304)는 LTPKA를 복호화하고, LTPKA 및 액세서리의 사용자명 userA를 페어링된 액세서리 레코드로서 지속적으로 저장할 수 있다. 이러한 저장소는 전술한 보안 요소 내에 있을 수 있다. 블록(1388)에서, 페어 셋업이 완료되고, 페어링 상태는 그러한 것을 나타내기 위해 업데이트될 수 있다.

[0232] 프로세스(1300)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 예를 들어, 제어기가 액세서리에 메시지를 송신하거나 또는 그 반대일 때마다(예를 들어, 페어링 프로세스 상태가 변화할 때), 오류들이 검출될 수 있다. 일부 오류 조건들이 나타나지만, 오류가 임의의 지점에서 검출되는 경우, 프로세스(1300)는 종료할 수 있고, 제어기는 오류를 사용자에게 통지할 수 있다는 것이 이해되어야 한다. 또한, SRP 및 특정 암호화 및/또는 인증 알고리즘들에 대한 참조는 예시의 목적을 위한 것이며, 안전하지 않은 통신 채널을 통한 데이터의 안전한 교환을 위한 다른 프로토콜들 및 알고리즘들이 대체될 수 있다.

[0233] 일부 실시예들에서, 페어링 프로세스는 인증 인프라구조를 활용할 수 있다. 예를 들어, 인증 칩(집적회로 디바이스, 또는 IC)이 액세서리 및/또는 제어기 디바이스들 내에 통합될 수 있다. 인증 칩은 디바이스를 위한 암호화 키들, 디바이스를 위한 보안 인증서, 및 다른 디바이스들에 의해 제시될 수 있는 유효하거나 무효한 보안 인증서들에 관한 정보를 안전하게 저장할 수 있다. 일부 실시예들에서, 인증 칩은 전술한 보안 요소(또는 그 일부)를 구현할 수 있다. 주어진 액세서리가 인증 칩을 포함하는 경우들에서, 인증 칩은 페어 셋업 프로세스에서 사용될 수 있다.

[0234] 도 14a 내지 도 14c는 본 발명의 일 실시예에 따른 인증 칩들을 사용하는 제어기(1402) 및 액세서리(1404)를 위한 페어 셋업 프로세스(1400)의 예를 도시한다. 제어기(1402)는, 프로세스(1400)를 개시하기 전에, 액세서리(1404)가 인증 칩을 갖고 있음을 확인했다고 가정된다. 예를 들어, 전술한 특징 플래그 특성(1202)은 액세서리가 인증서-기반 페어링을 지원하는지 여부를 나타내는 플래그를 포함할 수 있다. 이것은 인증 칩을 갖고 있는 액세서리들에 대해 설정될 수 있다. 다른 예로서, 액세서리가 인증서-기반 페어링을 지원하는지 여부를 나타내는 플래그는, 액세서리에 대한 서비스 검색 레코드의 특징 플래그 필드(예를 들어, 표 3의 Bonjour TXT 레코드의 특징 플래그 "ff" 필드)에 포함될 수 있으며, 이는 플래그가 페어링되지 않은 제어기에 더 쉽게 액세스가능하게 할 수 있다. (그럼에도 불구하고 인증 칩이 없는 액세서리가 이 플래그를 설정하는 경우, 제어기는 프로세스(1400)를 실행하는 정상적인 과정에서 인증 칩의 부재를 검출하고 오류를 보고할 수 있다.)

[0235] 도 14a를 참조하면, 블록(1406)에서, 제어기(1402)는 타원곡선 Diffie-Hellman 키 합의 프로토콜(elliptic curve Diffie-Hellman key agreement protocol)에 기반한 Curve25519 알고리즘(http://cr.yp.to/ecdh.html에 기록되어 있음)을 이용하여 공개/비밀 키 쌍(pkC, skC)을 생성할 수 있다. 이러한 키 쌍은 프로세스(1400)의 지속시간 동안 사용된 다음에 폐기되는 단기간 키 쌍일 수 있다. 블록(1408)에서, 제어기(1402)는 예를 들어, 적절한 URL로의 HTTP POST 요청으로서, 액세서리(1404)에 페어 셋업 시작 요청을 송신할 수 있으며; 요청은 공개 키 pkC를 포함할 수 있다. 페어 셋업 시작 요청은 페어 셋업 시작 요청이 송신되었음을 나타내는 상태 표시자를 포함할 수 있다(일부 실시예들에서, 이 상태 표시자 및 후속 상태 표시자는 도 12의 페어링 제어 포인트 특성(1201)에 기록될 수 있음).

[0236] 블록(1410)에서, 페어 셋업 시작 요청에 응답하여, 액세서리(1404)는 예를 들어, Curve25519 알고리즘을 사용하여, 공개/비밀 키 쌍(pkA, skA)을 생성할 수 있다. (pkC, skC)와 마찬가지로, 이 키 쌍은 프로세스(1400)의 지속시간 동안 사용된 다음에 폐기되는 단기간 키 쌍일 수 있다. 도시되지는 않았지만, 프로세스(1300)를 참조하여 전술한 스로틀링 거동이 통합될 수 있고, 액세서리(1404)는, 성공하지 못한 시도 후에 페어 셋업 시작 요청이 너무 빨리 수신되는 경우 그것을 거부할 수 있다.

[0237] 블록(1412)에서, 액세서리(1404)는 skA 및 pkC를 사용하여 공유 비밀("inputKey")을 생성할 수 있다. 블록

(1414)에서, 액세서리(1404)는 공개 키들 pkA 및 pkC를 연결함(concatenate)으로써 메시지를 구성할 수 있고, 블록(1416)에서, 액세서리(1404)는 자신의 인증 칩을 사용하여 메시지를 서명하여, 메시지 "smsg"를 생성할 수 있다. 인증 칩은 자신의 지속적 키 쌍(pkA 및 skA에 독립적임)을 가질 수 있고, 원하는 임의의 알고리즘, 예컨대 SHA-1 또는 SHA-2(미국 국가안전보장국(National Security Agency)에 의해 설계되고, 연방 정보 처리 표준(Federal Information Processing Standards) 공보 180-4에 문서로 기록된, 암호화 해시 함수들)를 구현할 수 있다.

[0238] 블록(1418)에서, 액세서리(1404)는 대칭키("Key")를 생성할 수 있다. 예를 들어, 액세서리(1404)는 inputKey, 솔트(예를 들어, 미리정의된 스트링), 및 추가 정보를 입력들로서 사용하는 HMAC-SHA-512를 사용할 수 있다.

[0239] 블록(1420)에서, 액세서리(1404)는, 블록(1416)으로부터의 서명된 메시지 smsg를 블록(1418)에서 생성된 키 Key를 사용하여 암호화하여, 증명 "proofA"를 생성할 수 있다. 임의의 대칭 암호화 알고리즘이 사용될 수 있다.

[0240] 블록(1422)에서, 액세서리(1404)는 제어기(1402)에 응답을 송신할 수 있다. 응답은 공개 키 pkA, 액세서리 증명 proofA, 및 인증 칩에 의해 제공된 액세서리 인증서를 포함할 수 있다. 블록(1422)에서 응답을 송신하면, 액세서리(1404)는 액세서리의 증명이 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0241] 블록(1424)에서, 제어기(1402)는 액세서리 인증서를 검증할 수 있다. 예를 들어, 제어기(1402)는 자신의 인증 칩, 또는 유효한 액세서리 인증서들을 식별하는 정보를 저장하는 다른 보안 데이터 저장소를 가질 수 있고; 그 정보는 제어기(1402)로 제공되고 일부 경우들에서 신뢰 인증서 기관에 의해 업데이트될 수 있다. 제어기(1402)는 수신된 액세서리 인증서가 유효함을 확인하기 위해 이 정보를 사용할 수 있다. 일부 실시예들에서, 특정 인증서들은 특정 클래스들의 액세서리들에 대해서만 유효할 수 있고, 제어기(1402)는 액세서리로부터 이전에 수신된 정보(예를 들어, 액세서리 정의 레코드 또는 전술한 바와 같이 디바이스 검색 동안에 제공된 다른 정보)를 사용하여, 액세서리의 클래스 및 수신된 액세서리 인증서가 액세서리의 클래스에 대해 유효한지 여부를 결정할 수 있다.

[0242] 도 14b를 참조하면, 블록(1426)에서, 액세서리 인증서가 유효하지 않은 경우, 프로세스(1400)는 종료할 수 있고(블록(1428)), 제어기(1402)는 사용자에게 오류를 알릴 수 있다. 액세서리 인증서가 유효한 경우, 이어서 블록(1430)에서, 제어기(1402)는 skC 및 pkA를 사용하여 공유 비밀("inputKey")을 생성할 수 있다. 블록(1432)에서, 제어기(1402)는 대칭키("Key")를 생성할 수 있다. 예를 들어, 액세서리(1404)는 inputKey, 솔트(예를 들어, 미리정의된 스트링), 및 추가 정보를 입력들로서 사용하는 HMAC-SHA-512를 사용할 수 있다. 블록(1434)에서, 제어기(1404)는, proofA를 암호화하기 위해 액세서리(1404)에 의해 사용된 동일한 알고리즘을 사용하여, 액세서리로부터 수신된 proofA를 복호화하기 위해 대칭키 Key를 사용할 수 있다. 따라서 제어기(1402)는 서명된 메시지 smsg를 획득한다. 블록(1436)에서, 제어기(1402)는 액세서리 인증서를 사용하여 smsg 상의 서명을 검증할 수 있다. 블록(1438)에서, 서명이 유효하지 않은 경우, 프로세스(1400)는 종료할 수 있고(블록(1440)), 제어기(1402)는 사용자에게 오류를 알릴 수 있다.

[0243] 도 14c를 참조하면, 서명이 유효한 경우, 이어서 블록(1442)에서, 제어기(1402)는 제어기 사용자명 "userC" 및 제어기의 장기간 공개 키 LTPKC를 갖는 데이터 객체를 구축할 수 있다(이는 전술한 프로세스(1300)에서와 동일할 수 있음). 블록(1444)에서, 제어기(1402)는 edataC 및 authTagC를 생성하기 위해 데이터 객체를 암호화할 수 있다. 블록(1442)에서의 암호화는, 전술한 프로세스(1300)와 유사하게, 키로서 Key, 메시지로서 LTPKC, 및 임의값을 갖고 ChaCha20-Poly1305와 같은 암호화-및-인증 알고리즘을 사용할 수 있다. 블록(1446)에서, 제어기(1402)는 edataC 및 authTagC를 포함하는 교환 요청 메시지를 액세서리(1404)에 송신할 수 있고; 제어기(1402)는 또한 LTPKC가 액세서리에 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0244] 블록(1448)에서, 액세서리(1404)는 자신의 대칭키 Key를 사용하여 수신된 authTagC를 검증할 수 있다. 블록(1450)에서, 블록(1448)에서의 검증이 실패하는 경우, 이어서 블록(1452)에서, 액세서리(1404)는 제어기(1402)에 오류 메시지를 송신할 수 있다. 블록(1454)에서, 제어기(1402)가 오류 메시지를 수신하는 경우, 프로세스(1400)는 블록(1456)에서 종료할 수 있다. 일부 실시예들에서, 제어기는 사용자에게 오류를 보고할 수 있다.

[0245] 블록(1450)의 결과가 성공인 경우, 이어서 블록(1458)에서, 액세서리(1404)는 LTPKC를 복호화할 수 있고, 블록(1460)에서, 액세서리(1304)는 LTPKC 및 제어기의 사용자명 userC를 페어링된 제어기 레코드로서 지속적으로 저장할 수 있다. 이러한 저장소는 전술한 보안 요소 내에 있을 수 있다.

[0246] 블록(1462)에서, 액세서리(1404)는 액세서리의 장기간 공개 키(LTPKA) 및 액세서리와 연관된 사용자명(userA)을 포함하는 데이터 객체를 구축할 수 있으며, 이들은 둘 모두 전술한 프로세스(1300)에서와 동일할 수 있다. 액

세서리의 장기간 공개/비밀 키 쌍은 액세서리의 인증 칩 내의 키 쌍과 상이할 수 있다. 블록(1464)에서, 액세서리(1404)는 edataA 및 authTagA를 생성하기 위해 블록(1462)에서 생성된 데이터 객체를 암호화할 수 있다. 블록(1444)에서 제어기(1402)에 의해 사용된 것과 동일한 암호화 알고리즘이 사용될 수 있다. 블록(1466)에서, 액세서리(1304)는 제어기(1402)에 edataA 및 authTagA를 송신할 수 있고; 액세서리(1404)는 또한 LTPKA가 제어기에 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0247] 블록(1468)에서, edataA 및 authTagA를 수신하면, 제어기(1402)는 이전에 생성된 키 Key를 사용하여 수신된 authTagA을 검증할 수 있다.

[0248] 블록(1470)에서, 블록(1468)에서의 검증이 실패하는 경우, 이어서 블록(1472)에서, 프로세스(1400)는 종료할 수 있고, 제어기(1404)는 사용자에게 오류를 보고할 수 있다. 검증이 성공하는 경우, 이어서 블록(1474)에서, 제어기(1404)는 LTPKA를 복호화하고, 페어링된 액세서리 레코드로서 LTPKA 및 액세서리의 사용자명 userA를 지속적으로 저장할 수 있다(그러한 저장소는 전술한 보안 요소 내에 있을 수 있음). 블록(1476)에서, 페어 셋업이 완료되고, 페어링 상태는 그러한 것을 나타내기 위해 업데이트될 수 있다.

[0249] 프로세스(1400)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 예를 들어, 제어기가 액세서리에 메시지를 송신하거나 또는 그 반대일 때마다(예를 들어, 페어링 프로세스 상태가 변화할 때), 오류들이 검출될 수 있다. 일부 오류 조건들이 표시되는 동안, 오류가 임의의 시점에 검출되는 경우, 프로세스(1400)는 종료할 수 있고, 제어기는 오류를 사용자에게 통지할 수 있다는 것이 이해되어야 한다. 또한, 특정 암호화 및/또는 인증 알고리즘들에 대한 참조는 예시의 목적을 위한 것이며, 안전하지 않은 통신 채널을 통한 데이터의 안전한 교환을 위한 다른 프로토콜들 및 알고리즘들이 대체될 수 있다.

[0250] 기술된 바와 같은 프로세스(1400)는, 액세서리가 검증을 위해 제어기에 인증서를 송신하지만 제어기는 액세서리에 대응하는 인증서를 송신하지 않는다는 점에서, 비대칭이다. 일부 실시예들에서, 양방향 인증서 검증이 구현될 수 있다. 예를 들어, 인증서를 갖는 제어기는 블록들(1412 내지 1416)과 유사한 처리를 구현하여 제어기 증명("proofC")을 생성할 수 있으며, 이는 제어기 인증서와 함께 액세서리에 송신될 수 있다. 액세서리는 제어기의 증명을 검증하기 위해 블록들(1424 내지 1436)과 유사한 처리를 구현할 수 있다.

[0251] 일부 실시예들에서, 인증 칩은 특별한 디바이스에 대해 특정적일 수 있으며, 각각의 디바이스는 고유한 인증서를 가질 수 있다. 다른 실시예들에서, 동일한 클래스 내의 액세서리들(또는 제어기들)은 동일한 인증 칩들을 갖고, 따라서 동일한 인증서들을 가질 수 있다. 이런 경우, 페어 셋업 동안 중간자 공격들에 대한 보호가 감소될 수 있지만; 장기간 공개 키들이 교환되면, 이들 키는 후속 페어 검증 동안에 양방향 인증에 대해 신뢰성 있게 사용될 수 있다.

[0252] 다른 기술들이 중간자 공격들 또는 다른 취약점 공격(exploit)들의 위험을 추가로 감소시키기 위해 사용될 수 있다. 예를 들어, 페어 셋업 프로세스에서 두 디바이스 중 하나(또는 둘 모두)가 다른 디바이스가 얼마나 가까이 있는지를 (예를 들어, 블루투스 LE 또는 NFC를 사용하여) 검출할 수 있는 근접 센서를 갖고 있는 경우, 센서를 가진 디바이스는 다른 디바이스가 가깝게 근접해(예를 들어, 수 인치 이내) 있지 않거나 그렇게 유지되지 않는 경우에 페어 셋업 프로세스를 중단시킬 수 있다. 이것은 중간자 공격들의 가능성을 감소시킬 수 있는데, 이는 공격하는 디바이스가 프로세스에서 의도된 참가자들에 물리적으로 가까울 필요가 있을 것이며, 이 상황은 사용자가 알아차릴 가능성이 있을 것이기 때문이다.

[0253] 일부 실시예들은 보안 인증서들 및 셋업-코드-기반 페어링 둘 모두를 페어 셋업 프로세스 내에 통합시킬 수 있다. 도 15a 내지 도 15f는 본 발명의 일 실시예에 따른, 보안 인증서 및 셋업 코드를 사용한 제어기(1502) 및 액세서리(1504)를 위한 페어 셋업 프로세스(1500)의 예를 도시한다. 프로세스(1500)는 전술한 프로세스들(1300, 1400)의 특징들을 통합할 수 있다.

[0254] 도 15a를 먼저 참조하면, 블록(1506)에서, 제어기(1502)는 페어 셋업 프로세스를 시작하기 위하여 액세서리(1504)에 시동(startup) 요청을 송신할 수 있다. 예를 들어, 제어기(1502)는 액세서리(1504)의 /pair-setup URL로의 HTTP POST 요청을 송신할 수 있다. POST 요청은 원하는 페어링 상태(예를 들어, "시작 페어 셋업")를 나타내는 TLV 데이터 객체, 및 셋업 코드 및/또는 인증서가 사용될 것인지 여부를 나타내는 방법 식별자를 포함할 수 있다. 블록(1508)에서, 액세서리(1504)는 시동 요청을 수신할 수 있고 이에 따라 현재 페어링 상태를 설정할 수 있다. 블록(1510)에서, 액세서리(1504)는 존재할 수 있는 임의의 다른 제어기들이 페어 셋업을 개시하는 것을 막을 수 있다. 예를 들어, 페어 셋업을 시작하려는 요청이 블록(1510) 이후에 수신되는 경우, 액세서

리는 오류 응답(예를 들어, HTTP 429 "너무 많은 요청(Too Many Requests)" 메시지)를 반환할 수 있다. 액세서리(1504)는 프로세스(1500)가 완료될(또는 오류로 인해 종료될) 때까지 이러한 방식으로 페어-셋업 요청들을 계속해서 막을 수 있다.

[0255] 일부 실시예들에서, 액세서리(1504)는, 전역으로 또는 제어기당 단위로, 성공하지 못한 페어-셋업 시도들의 수를 시도들의 최대 수로 제한할 수 있다. 예를 들어, 액세서리(1504)는 한계(예를 들어, 5회 시도, 10회 시도, 또는 임의의 다른 한계)을 정의하고 성공하지 못한 시도들의 카운터를 유지할 수 있다. 카운터가 한계에 도달하거나 이를 초과하면, 블록(1512)에서, 액세서리는 요청을 거부할 수 있고, 프로세스(1500)는 블록(1514)에서 종료될 수 있다. 일부 실시예들에서, 액세서리(1504)는 오류의 원인(예를 들어, 너무 많은 시도)을 나타내는 오류 응답을 제어기(1502)에 송신할 수 있다. 대안적으로, (예를 들어, 도 13a를 참조하여 전술한 바와 같은) 스로틀링 기법또는 다른 기법들이, 인가되지 않은 제어기에 의해 페어 셋업을 수행하려는 무차별(brute-force) 시도들을 방지하기 위해 사용될 수 있다.

[0256] 시도의 최대 수가 초과되지 않았다고 가정하면, 블록(1516)에서, 액세서리(1504)는 페어 셋업을 시작할 수 있다. 예를 들어, 액세서리(1504)는, 예를 들어, SRP_new(SRP6a_server_method())와 같은 적절한 SRP 프로토콜 함수들을 호출함으로써, SRP 세션을 생성하고; (예를 들어, 사용자명으로서, 일반적일 수 있는, 고정 스트링을 사용한, SRP_set_username로) 세션에 대한 SRP 사용자명을 설정하고; SRP_set_params로 설정될 수 있는, 랜덤 솔트(예를 들어, 적어도 16 바이트)를 생성하고; (예를 들어, SRP_set_auth_password 또는 SRP_set_auth_password_raw를 사용해) SRP 패스워드로서 설정될 셋업 코드를 선택(예를 들어, 검색)할 수 있다.

[0257] 프로세스(1300)와 유사하게, 액세서리(1504)는, EEPROM으로부터 코드를 판독하는 것, 사용자로부터 코드를 수신하는 것, (예를 들어, 프로세스(1500)의 실행 동안에) 랜덤 코드를 생성하는 것 등을 포함하는, 다양한 기법들을 사용하여 셋업 코드를 선택할 수 있다. 블록(1518)에서, 액세서리(1504)는 사용자에게 셋업 코드를 제시할 수 있다. 예를 들어, 액세서리 및 셋업 코드에 따라, 코드는 액세서리(1504) 또는 그것의 패키징에 부착된 라벨 상에 인쇄되거나, 액세서리(1504)의 디스플레이 상에 제시되는 등일 수 있다. 일부 경우들에서, 사용자에게 셋업 코드를 제시하는 대신에, 액세서리(1504)는, NFC 채널 또는 단거리(예를 들어, 50 cm 미만)를 갖는 다른 시그널링 채널과 같은, 페어 셋업 프로세스(1500)에 사용되고 있는 채널에 독립적인 통신 채널을 사용해 제어기(1502)에 코드를 전달할 수 있다.

[0258] 블록(1520)에서, 액세서리(1504)는 (예를 들어, SRP_gen_pub을 사용하여) 공개 키("pkA")를 생성할 수 있다. 공개 키 pkA는 프로세스(1500)의 지속시간 동안 사용된 다음에 폐기되는 단기간 키 쌍(비밀 키 "skA"와 함께)의 일부일 수 있다.

[0259] 블록(1522)에서, 액세서리(1504)는 예를 들어, HTTP 응답 메시지를 사용해, 시동 요청에 대한 응답을 송신할 수 있다. 응답은 공개 키 pkA 및 랜덤 솔트를 포함할 수 있다. 블록(1522)에서 응답을 송신하면, 액세서리(104)는 액세서리의 공개 키가 송신되었음을 나타내기 위해 현재 페어링 상태를 업데이트할 수 있다.

[0260] 블록(1524)에서, 제어기(1502)는 시동 요청에 대한 응답을 수신할 수 있다. 블록(1526)에서, 제어기(1502)는 사용자로부터 셋업 코드를 획득할 수 있다. 예를 들어, 제어기(1502)는 액세서리의 셋업 코드를 입력하도록 사용자에게 프롬프트하는 사용자 인터페이스 스크린을 제시할 수 있다. 전술한 프로세스(1300)의 블록(1322)과 마찬가지로, 셋업 코드가 대역 외로, 즉, 페어 셋업 프로세스를 수행하는 데 사용되고 있는 통신 채널에 독립적인 통신 채널을 통해 제어기(1502)로 전달되는 한, 다른 기술들이 사용될 수 있다. 사용되는 특정 기술에 따라, 제어기에 의해 셋업 코드를 획득하는 것은, 일부 형태의 사용자 행위(예를 들어, 코드를 입력하는 것, 액세서리에 근접하게 제어기를 들고 있는 것, 제어기의 카메라를 동작시키는 것 등)을 포함할 수 있으며, 그러한 행위는, 디바이스들의 아이덴티티의 대역외 확인을 제공하는 것에 더하여, 사용자가 제어기(1502)와 액세서리(1504) 사이에 페어링을 확립하려고 한다는 것을 확인하는 것으로서 기능할 수 있다.

[0261] 도 15b를 참조하면, 블록(1528)에서, 제어기(1502)는 예를 들어, SRP_new(SRP6a_server_method())와 같은 적합한 SRP 프로토콜 함수들을 호출함으로써, SRP 세션을 생성하고; (예를 들어, 사용자명으로서, 일반적일 수 있는, 고정 스트링을 사용하여, SRP_set_username으로) SRP 사용자명을 설정하고; 액세서리로부터 수신된 솔트로 (SRP_set_params으로) 솔트를 설정하고; (예를 들어, SRP_gen_pub를 사용하여) 자신의 공개 키("pkC")를 생성하고; (예를 들어, SRP_set_auth_password 또는 SRP_set_auth_password_raw를 사용하여) 블록(1526)에서 획득된 셋업 코드를 사용해 SRP 패스워드를 설정할 수 있다. 액세서리 공개 키 pkA와 마찬가지로, 공개 키 pkC는 프로세스(1500)의 지속시간 동안 사용된 다음에 폐기되는 단기간 키 쌍(비밀 키 "skC"와 함께)의 일부일 수 있다. 블록(1530)에서, 제어기(1502)는 (예를 들어, SRP_compute_key를 사용하여) 공유 비밀("inputKey")을 계

산할 수 있다. 블록(1532)에서, 제어기(1502)는 (예를 들어, SRP_Respond를 사용하여) 그것이 공유 비밀 inputKey를 갖는다는 증명("proofC")을 생성할 수 있다.

[0262] 블록(1534)에서, 제어기(1502)는 액세서리(1504)에 검증 요청을 송신할 수 있다. 예를 들어, 제어기(1502)는 액세서리(1504)의 /pair-setup URL로의 HTTP POST 요청을 송신할 수 있다. POST 요청은 원하는 페어링 상태(예를 들어, "페어 셋업 검증")를 나타내는 TLV 데이터 객체, 제어기의 공개 키 pkC, 및 제어기 증명 proofC를 포함할 수 있다.

[0263] 블록(1536)에서, 액세서리(1504)는 검증 요청을 수신할 수 있다. 블록(1538)에서, 액세서리(1504)는 (예를 들어, SRP_compute_key를 사용하여) 공유 비밀("inputKey")을 계산할 수 있고; 이것은 블록(1530)에서 제어기(1502)에 의해 계산된 공유 비밀과 매칭되어야 한다.

[0264] 블록(1540)에서, 액세서리(1504)는 (예를 들어, SRP_verify를 사용하여) 제어기 증명 proofC를 검증하기 위해 블록(1538)에서 계산된 공유 비밀을 사용할 수 있다. 도 15b에 도시되지 않았지만, 검증이 실패하는 경우, 액세서리(1504)는 프로세스(1500)를 종료하고 제어기(1502)에 오류 메시지를 송신할 수 있다. (본 명세서에 기술된 다른 페어링 프로세스들에서와 같이, 제어기(1502)는 사용자에게 오류를 보고할 수 있다.)

[0265] proofC가 검증된다고 가정하면, 블록(1542)에서, 액세서리(1504)는 (예를 들어, SRP_respond를 사용하여) 그것이 공유 비밀을 소유한다는 것을 증명하는 액세서리 증명("proofA")을 생성할 수 있다. 블록(1544)에서, 액세서리(1504)는 공유 비밀로부터 세션 암호화 키(eKey)를 도출할 수 있다. 예를 들어, 액세서리(1504)는 inputKey, 솔트, 및 추가 정보를 입력들로서 사용하는 512-비트 해시들에 대한 SHA(Secure Hash Algorithm) 버전 2(IETF RFC 6234에 기록되어 있는, HKDF-SHA-512)를 구현하는 HKDF-기반 키 도출 함수를 사용할 수 있다.

[0266] 도 15c를 참조하면, 이 예에서, 액세서리(1504)는 신뢰 인증서 기관에 의해 발행된 보안 인증서를 갖고 있다고 가정된다. 일부 실시예들에서, 인증서는 도 14a 내지 도 14c를 참조하여 전술한 바와 같이 인증 칩에 통합될 수 있다. 이런 경우, 액세서리(1504)는 자신의 인증서를 사용하여, 자신의 아이덴티티를 제어기(1502)에 추가로 인증할 수 있다. 일부 실시예들에서, 블록(1506)에서 제어기(1502)에 의해 송신된 시동 요청, 또는 블록(1534)에서 제어기(1502)에 의해 송신된 검증 요청은 액세서리가 인증서로 인증해야 하는지 여부의 표시를 포함할 수 있다.

[0267] 액세서리(1504)가 인증서로 인증되어야 경우, 블록(1546)에서, 액세서리(1504)는 블록(1538)에서 계산된 공유 비밀(inputKey)로부터 서명될 챌린지(challenge)를 생성할 수 있다. 예를 들어, 챌린지는 inputKey, 솔트, 및 추가 정보 항목을 입력들로서 사용하는 HKDF-SHA-512와 같은 키 도출 함수를 적용함으로써 생성될 수 있다. 솔트 및 추가 정보 항목은 그것의 운영 체제 또는 펌웨어의 일부로서 액세서리(1504) 내로 프로그램될 수 있는 미리정의된 값들을 가질 수 있다. 블록(1548)에서, 액세서리(1504)는 자신의 인증서를 사용해 챌린지를 서명할 수 있다. 예를 들어, 액세서리(1504)가 인증 칩을 포함하는 경우, 인증 칩은 챌린지를 서명할 수 있다. 전술한 바와 같이, 인증 칩은 (pkA 및 skA 또는 LTPKA 및 LTSKA에 독립적인) 자신의 지속적인 키 쌍을 가질 수 있고, SHA-1과 같은, 원하는 임의의 서명 알고리즘을 구현할 수 있다.

[0268] 블록(1550)에서, 액세서리(1504)는 인증 칩으로부터 검색될 수 있는, 액세서리 인증서 및 서명된 챌린지를 포함하는 데이터 구조를 구축할 수 있다. 블록(1552)에서, 액세서리(1504)는 블록(1544)에서 생성된 암호화 키(eKey)를 사용하여, 블록(1550)에서 구축된 데이터 구조를 암호화할 수 있다. ChaCha20-Poly1305 AEAD 알고리즘과 같은, 임의의 대칭 암호화 알고리즘이 사용될 수 있다. 암호화 알고리즘은 암호화된 데이터 구조 및 태그(authTagA)를 생성할 수 있다.

[0269] 블록(1554)에서, 액세서리(1504)는 제어기(1502)에 검증 응답을 송신할 수 있다. 검증 응답은 블록(1542)에서 생성된 액세서리 증명 proofA 뿐만 아니라, 블록(1550)에서 생성된 암호화된 데이터 구조 및 authTagA를 포함할 수 있다. 전술한 바와 같이, 일부 실시예들서, 인증서-기반 인증은 (예를 들어, 제어기(1502)가 인증서-기반 인증을 요청했는지 여부에 따라) 선택적으로 수행되거나 수행되지 않을 수 있다. 인증서-기반 인증이 수행되고 있지 않는 경우들에서, 검증 응답은 암호화된 데이터 구조 및 authTagA를 생략할 수 있다.

[0270] 블록(1556)에서, 제어기(1502)는 액세서리(1504)로부터 검증 응답을 수신할 수 있다. 블록(1558)에서, 제어기(1502)는 (예를 들어, SRP_verify를 사용하여) 액세서리 증명 proofA를 검증할 수 있다. 검증이 실패하는 경우, 프로세스(1500)는 오류로 종료될 수 있다. 검증이 성공한다고 가정하면, 블록(1560)에서, 제어기(1502)는 블록(1530)에서 계산된 공유 비밀(inputKey)로부터 암호화 키(eKey)를 도출할 수 있다. 제어기(1502)는 액세서리(1504)가 블록(1544)에서 사용한 것과 동일한 키 도출 알고리즘 및 입력들을 사용하여, 블록(1560)에서

도출된 eKey가 블록(1544)에서 액세서리에 의해 생성된 eKey와 매칭될 것으로 예상되도록 할 수 있다.

[0271] 블록(1562)에서, 제어기(1502)는 수신된 authTagA를 검증할 수 있고, 블록(1564)에서, 제어기(1502)는 수신된 데이터 구조를 복호화할 수 있다.

[0272] 이제 도 15d를 참조하면, 블록(1566)에서, 제어기(1502)는 복호화된 데이터 구조로부터 추출된 액세서리 인증서를 검증할 수 있다. 예를 들어, 전술한 바와 같이, 제어기(1502)는 자신의 인증 칩, 또는 유효한 액세서리 인증서들을 식별하는 정보를 저장하는 다른 보안 데이터 저장소를 가질 수 있고; 그 정보는 신뢰 인증서 기관에 의해 제공되고 일부 경우들에서 업데이트될 수 있다. 대안적으로, 제어기(1502)는 수신된 인증서를 검증하기 위해 신뢰 인증 기관과 실시간으로 통신할 수 있다. 제어기(1502)는 액세서리 인증서가 유효함을 확인하기 위해 (로컬로 캐싱되거나 실시간으로 획득된) 인증 기관으로부터의 정보를 사용할 수 있다. 일부 실시예들에서, 특정 인증서들은 특정 클래스들의 액세서리들에 대해서만 유효할 수 있고, 제어기(1502)는 액세서리로부터 이전에 수신된 정보(예를 들어, 전술한 바와 같이 디바이스 검색 동안에 제공된 정보)를 사용하여, 액세서리의 클래스 및 액세서리 인증서가 액세서리의 클래스에 대해 유효한지 여부를 결정할 수 있다. 인증서가 유효하지 않는 경우, 프로세스(1500)는 오류로 종료할 수 있다.

[0273] 인증서가 유효하다고 가정하면, 블록(1568)에서, 제어기(1502)는 공유 비밀 inputKey로부터 챌린지를 생성할 수 있다. 제어기(1502)는 액세서리(1504)가 블록(1546)에서 사용한 것과 동일한 알고리즘 및 입력들(예를 들어, 미리정의된 솔트 및 추가 정보 항목을 갖는 inputKey)을 사용할 수 있으며, 이때 그 결과로 제어기(1502) 및 액세서리(1504) 둘 모두가 동일한 챌린지를 생성해야 한다. 이 기술을 이용하면, 제어기(1502)가 액세서리(1504)에 보통문으로(in the clear) 챌린지를 송신할 필요가 없다. 또한, 챌린지가 공유 비밀 inputKey를 통합하는 경우, 사칭자가 챌린지를 추측하는 것은 어려울 수 있다. 블록(1570)에서, 제어기(1502)는 액세서리 인증서로부터의 공개 키를 사용하여 서명된 챌린지를 검증할 수 있다. 검증이 실패하는 경우, 프로세스(1500)는 오류로 종료할 수 있다.

[0274] 검증이 성공한다고 가정하면, 제어기는 이제 액세서리와 장기간 공개 키들을 교환할 준비가 된다. 블록(1572)에서, 제어기(1502)는 공유 비밀의 표현(예를 들어, 그것의 운영 체제 또는 펌웨어의 일부로서 제어기(1502) 내에 프로그램될 수 있는 미리정의된 값들을 가질 수 있는, 솔트 및 추가 정보 항목을 갖는 공유 비밀의 HDKF-SHA-512), 제어기의 장기간 공개 키(LTPKC), 및 제어기의 식별자를 연결하는 LTPKC 메시지를 생성할 수 있다. 일부 실시예들에서, 제어기는 블록(1572)에서 사용될 수 있는 미리정의된 (LTPKC, LTSKC) 키 쌍을 갖고; 다른 실시예들에서는, (LTPKC, LTSKC) 키 쌍이 블록(1572)에서 생성될 수 있다. 블록(1574)에서, 제어기(1502)는 예를 들어, Ed25519와 같은 서명 알고리즘(http://ed25519.cr.yp.to에 기록되어 있음)을 LTPKC 메시지에 적용함으로써, 그것의 장기간 비밀 키(LTSKC)를 사용하여 LTPKC 메시지를 서명할 수 있다. 블록(1576)에서, 제어기(1502)는 LTPKC 메시지로부터의 서명, LTPKC, 및 제어기의 ID를 포함하는 데이터 구조를 구축할 수 있다. (LTPKC 메시지 자체는 이 데이터 구조로부터 생략될 수 있는데, 이는 액세서리(1504)가 그것을 재구성할 수 있을 것이기 때문이다.) 블록(1578)에서, 제어기(1502)는 블록(1560)에서 도출된 암호화 키 eKey를 사용하여 데이터 구조를 암호화하고 인증 태그(authTagC)를 생성할 수 있다.

[0275] 블록(1580)에서, 제어기(1502)는 액세서리에 키 교환 요청을 송신할 수 있다. 예를 들어, 제어기(1502)는 액세서리(1504)의 /pair-setup URL로의 HTTP POST 요청을 송신할 수 있다. 키 교환 요청은 블록(1578)에서 생성된 암호화된 데이터 구조 및 인증 태그를 포함할 수 있다. 블록(1582)에서, 액세서리(1504)는 제어기(1502)로부터 키 교환 요청을 수신할 수 있다.

[0276] 이제 도 15e를 참조하면, 블록(1584)에서, 액세서리(1504)는 수신된 인증 태그를 검증할 수 있다. 블록(1586)에서, 액세서리(1504)는 데이터 구조를 복호화할 수 있고, 블록(1588)에서, 액세서리(1504)는 데이터 구조에 포함된 서명을 검증할 수 있다. 검증들 중 임의의 것이 실패하는 경우, 프로세스(1500)는 오류로 종료할 수 있다.

[0277] 인증 태그 및 서명이 검증된다고 가정하면, 블록(1590)에서, 액세서리(1504)는 데이터 구조로부터 추출된 LTPKC 및 제어기 ID를 페어링된 제어기 레코드로서 지속적으로 저장할 수 있다. 이러한 저장소는 전술한 보안 요소 내에 있을 수 있다.

[0278] 액세서리(1504)는 유사한 방식으로 제어기(1502)에 자신의 장기간 공개 키(LTPKA)를 송신할 수 있다. 예를 들어, 블록(1592)에서, 액세서리(1504)는 공유 비밀의 표현(예를 들어, 액세서리의 시스템 소프트웨어 또는 펌웨어 내에 프로그램될 수 있는 미리정의된 값들을 가질 수 있는, 솔트 및 추가 정보 항목을 갖는 공유 비밀의

HDKF-SHA-512), 액세서리의 장기간 공개 키(LTPKA), 및 액세서리의 식별자를 연결하는 LTPKA 메시지를 생성할 수 있다. 일부 실시예들에서, 액세서리는, 블록(1592)에서 사용될 수 있는 미리정의된 (LTPKA, LTSKA) 키 쌍을 갖고; 다른 실시예들에서는, (LTPKA, LTSKA) 키 쌍이 블록(1592)에서 생성될 수 있다. 블록(1594)에서, 액세서리(1504)는 예를 들어, Ed25519와 같은 서명 알고리즘을 LTPKA 메시지에 적용함으로써, 그것의 장기간 비밀 키(LTSKA)를 사용하여 LTPKA 메시지를 서명할 수 있다. 블록(1596)에서, 액세서리(1504)는 LTPKA 메시지로부터의 서명, LTPKA, 및 액세서리의 ID를 포함하는 데이터 구조를 구축할 수 있다. (LTPKA 메시지 자체는 이 데이터 구조로부터 생략될 수 있는데, 이는 제어기(1502)가 그것을 재구성할 수 있을 것이기 때문이다.) 블록(1598)에서, 액세서리(1504)는 블록(1544)에서 도출된 암호화 키 eKey를 사용하여 데이터 구조를 암호화하고 인증 태그(authTagB)를 생성할 수 있다.

[0279] 블록(1501)에서, 액세서리(1504)는 제어기(1502)에 키 교환 응답을 송신할 수 있다. 키 응답은 블록(1598)에서 생성된 암호화된 데이터 구조 및 인증 태그를 포함할 수 있다. 블록(1503)에서, 제어기(1502)는 키 교환 응답을 수신할 수 있다.

[0280] 이제 도 15f를 참조하면, 블록(1505)에서, 제어기(1502)는 수신된 인증 태그를 검증할 수 있다. 블록(1507)에서, 제어기(1502)는 데이터 구조를 복호화할 수 있고, 블록(1509)에서, 제어기(1502)는 데이터 구조에 포함된 서명을 검증할 수 있다. 검증들 중 임의의 것이 실패하는 경우, 프로세스(1500)는 오류로 종료할 수 있다.

[0281] 인증 태그 및 서명이 검증된다고 가정하면, 블록(1511)에서, 제어기(1502)는 데이터 구조로부터 추출된 LTPKA 및 액세서리 ID를 페어링된 액세서리 레코드로서 지속적으로 저장할 수 있다. 이러한 저장소는 전술한 바와 같이 보안 요소 내에 있을 수 있다.

[0282] 블록(1513)에서, 페어 셋업이 완료되고, 페어링 상태는 그러한 것을 나타내기 위해 업데이트될 수 있다.

[0283] 프로세스(1500)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 예를 들어, 제어기가 액세서리에 메시지를 송신하거나 또는 그 반대일 때마다(예를 들어, 페어링 프로세스 상태가 변화할 때), 오류들이 검출될 수 있다. 일부 오류 조건들이 표시되는 동안, 오류가 임의의 시점에 검출되는 경우, 프로세스(1500)는 종료할 수 있고, 제어기는 오류를 사용자에게 통지할 수 있다는 것이 이해되어야 한다. 또한, 특정 암호화 및/또는 인증 알고리즘들에 대한 참조는 예시의 목적을 위한 것이며, 안전하지 않은 통신 채널을 통한 데이터의 안전한 교환을 위한 다른 프로토콜들 및 알고리즘들이 대체될 수 있다.

[0284] 기술된 바와 같은 프로세스(1500)는, 액세서리가 검증을 위해 제어기에 인증서를 송신하지만 제어기는 액세서리에 대응하는 인증서를 송신하지 않는다는 점에서, 비대칭이다. 일부 실시예들에서, 양방향 인증서 검증이 구현될 수 있다. 예를 들어, 인증서를 갖는 제어기는 블록들(1546 내지 1552)과 유사한 처리를 구현하여, 챌린지를 생성하고 제어기 인증서를 사용해 그것을 서명할 수 있으며, 서명된 챌린지는 제어기 인증서와 함께 액세서리에 송신될 수 있다. 액세서리는 제어기의 증명을 검증하기 위해 블록들(1566 내지 1570)과 유사한 처리를 구현할 수 있다.

[0285] 일부 실시예들에서, 주어진 액세서리 또는 제어기는 페어 셋업 프로세스들(1300, 1400, 1500)(및/또는 구체적으로 기술되지 않은 다른 프로세스들) 중 임의의 것 또는 전부를 지원할 수 있고, 사용될 페어 셋업 프로세스는 페어링당 기준으로 선택될 수 있다. 균일성을 위해, 제어기는 (예를 들어, 인증서들을 갖는 그리고 갖지 않는) 다수의 페어 셋업 프로세스들을 지원할 수 있고, 주어진 액세서리가 어떤 프로세스(들)를 지원하는지에 기초하여 그 액세서리에 대한 프로세스를 선택할 수 있다. 프로세스들은 (예를 들어, 제공된 상대적 보안에 기초하여) 우선 순위를 할당받을 수 있고, 제어기는 주어진 액세서리가 지원하는 가장 선호되는 프로세스를 선택할 수 있다. 제어기는 예를 들어, 프로세스 식별자를 시동 요청 메시지에 포함시킴으로써, 사용될 프로세스를 명시할 수 있다.

[0286] 페어 셋업의 보다 일반적인 도면은 도 16을 참조하여 얻어질 수 있으며, 이는 본 발명의 일 실시예에 따른 제어기(1602)와 액세서리(1604) 사이의 일반화된 페어 셋업 프로세스(1600)를 도시한다. 페어 셋업 프로세스(1600)는 블록(1606)에서 제어기(1602)에 의해 개시될 수 있다. 예를 들어, 전술한 바와 같이, 제어기(1602)는 페어 셋업이 수행되어야 하는 액세서리(예를 들어, 액세서리(1604))를 식별하기 위해 (블록(426)을 통해) 프로세스(400)를 실행할 수 있다. 제어기(1602)는 액세서리(1604)에 메시지(예를 들어, 적절한 URL로의 POST 요청)를 송신하여, 액세서리(1604)가 또한 블록(1608)에서 페어 셋업을 시작하게 할 수 있다.

[0287] 블록들(1610, 1612)에서, 액세서리(1604) 및 제어기(1602)는 공유 비밀(예를 들어, 전술한 프로세스들(1300,

1400, 1500)에서의 inputKey)을 확립할 수 있다. 공유 비밀을 확립하는 것은 양방향 정보 교환을 포함할 수 있다. 예를 들어, 프로세스(1300)에서, 액세서리는 솔트 및 단기간 공개 키를 제공하고, 제어기는 자신의 단기간 공개 키를 제공한다. 프로세스들(1400, 1500)에서, 액세서리는 자신의 단기간 공개 키 및 인증서를 제공하고, 제어기는 자신의 단기간 공개 키를 제공한다. 일부 실시예들에서, 공유 비밀은 또한 (예를 들어, 보안 요소 내 또는 다른 곳에서) 두 디바이스 모두 내에 프로그램되는 다른 정보를 포함할 수 있다. 공유 비밀은 또한 대역외 정보를 포함할 수 있으며, 이는 제어기가 액세서리와 상호동작하도록 (예를 들어, 사용자에 의해) 인가되어 있다는 증거를 제공할 수 있다. 예를 들어, 프로세스(1300) 또는 프로세스(1500)에서, 액세서리의 셋업 코드는 제어기 및 액세서리에 의해 공유 비밀을 구성하기 위해 사용된다. 전술한 바와 같이, 액세서리의 셋업 코드는 대역외로, 즉, 증명들을 송신 및 수신하기 위해 사용되고 있는 채널 이외의 통신 채널을 사용하여, 제어기에 제공될 수 있다. 대역외 채널들의 예들은 사용자 입력(예를 들어, 사용자는 제어기의 사용자 인터페이스에서 액세서리의 셋업 코드를 입력하고, 제어기의 카메라를 사용하여 셋업 코드의 사진을 촬영할 수 있음), 근거리 통신 채널, 광학 시그널링 채널, 유선 전자 시그널링 채널, 오디오(예를 들어, 초음파) 채널 등을 포함할 수 있다. 일부 경우들에서, 대역외 채널은 사용자 개입(예를 들어, 셋업 코드를 입력하는 것, 액세서리에 근거리 근접하게 제어기를 유지하는 것, 사진을 촬영하는 것, 커넥터를 플러그 인하는 것)을 포함할 수 있고, 대역외 채널을 통해 셋업 코드를 전달하는 사실은, 사용자가 페어링을 확립하는 것을 승인하는 표시로서 기능할 수 있다.

[0288] 블록(1614)에서, 제어기(1602)는 증명을 생성하고 액세서리(1604)에 송신할 수 있다. 본 명세서에서 사용되는 바와 같이, "증명", 또는 "암호 증명"은, 공유 비밀을 소유한 수신 디바이스(이 경우, 액세서리(1604))가, 송신 디바이스(이 경우, 제어기(1602)) 또한 공유 비밀을 소유하고 있다는 것을 검증하기 위해 사용할 수 있는, 정보의 임의의 조각을 포함할 수 있다. 제어기의 증명의 예들은 프로세스들(1300, 1400, 1500)에서 "proofC"라고 라벨링된 메시지를 포함한다.

[0289] 블록(1616)에서, 액세서리(1604)는 제어기의 증명을 수신할 수 있고, 블록(1618)에서 액세서리(1604)는, 공유 비밀의 그것의 로컬로 생성된 사본에 기초하여, 증명을 검증할 수 있다. 공유 비밀이 매칭되지 않는 경우, 제어기의 증명은 검증되지 않을 것이고, 프로세스(1600)는 오류로 종료할 수 있다. 셋업 코드가 공유 비밀을 생성하는 데 사용되는 경우, 블록(1618)에서의 검증은 또한 제어기의 액세서리에 대한 인증의 역할을 할 수 있음에 유의해야 한다.

[0290] 증명이 검증된다고 가정하면, 블록(1620)에서, 액세서리(1604)는 자신의 증명을 생성하고 제어기(1602)에 송신하여, 액세서리(1604) 또한 공유 비밀을 소유하고 있음을 입증할 수 있다. 액세서리 증명은, 예를 들어, (프로세스들(1300, 1500)에서와 같이) 공유 비밀을 또한 포함하는 제어기의 증명과는 상이한 암호화된 메시지일 수 있다. 일부 실시예들에서, 액세서리의 아이덴티티의 다른 증명들이 포함될 수 있으며; 예를 들어, 프로세스들(1400, 1500)에서, 액세서리는 액세서리의 아이덴티티의 적어도 일부 양상들을 확인하는 인증서를 사용하여 메시지를 서명할 수 있다.

[0291] 블록(1624)에서, 제어기는 수신된 증명을 검증할 수 있다. 블록(1618)과 유사하게, 공유 비밀이 매칭되지 않는 경우, 액세서리의 증명은 검증되지 않을 것이고, 프로세스(1600)는 오류로 종료할 수 있다. 셋업 코드가 공유 비밀을 생성하는 데 사용되는 경우, 블록(1624)에서의 검증은 또한 액세서리의 제어기에 대한 인증의 역할을 할 수 있음에 유의해야 한다. 예를 들어, 액세서리 증명이 (프로세스들(1400, 1500)에서와 같이) 인증서를 사용하여 서명된 메시지를 포함하는 경우에, 추가 인증이 제공될 수 있다.

[0292] 블록(1624)에서 증명이 검증된다고 가정하면, 두 디바이스 모두가 서로에 대해 인증되는 것으로 간주될 수 있고, 공유 비밀은 전술한 장기간 공개 키들과 같은 추가 정보를 교환하는 데 사용되어, (지속적인) 페어링을 확립할 수 있다. 예를 들어, 블록(1626)에서, 제어기(1602)는 자신의 장기간 공개 키(LTPKC)를 송신할 수 있다. LTPKC는 예를 들어, (프로세스들(1300, 1400, 1500)에서와 같이) 공유 비밀로부터 도출된 키를 사용하여, 암호화된 형태로 송신될 수 있다. 블록(1628)에서, 액세서리(1604)는 LTPKC를 수신하고 (예를 들어, 전술한 바와 같이 보안 요소에) 지속적으로 저장할 수 있다. 블록(1630)에서, 액세서리(1604)는 자신의 장기간 공개 키(LTPKA)를 송신할 수 있다. LTPKA는 또한, 예를 들어, (프로세스들(1300, 1400, 1500)에서와 같이) 공유 비밀로부터 도출된 키를 사용하여, 암호화된 형태로 송신될 수 있다. 블록(1632)에서, 제어기(1602)는 LTPKA를 수신하고 (예를 들어, 전술한 바와 같이 보안 요소에) 지속적으로 저장할 수 있다. 그 후, 페어 셋업이 완료되는데(블록(1634)), 이는 각각의 디바이스가 이제 다른 것과의 페어링을 확립하는, 인증되고 지속적으로 저장된 레코드를 갖기 때문이다.

[0293] 프로세스(1600)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 예를 들어, 증명들은 어느 순서로나(액세서리-우선 또는 제어기-우선) 교환될 수 있으며, 장기간 공개 키들 또한 어느 순서로나 교환될 수 있다.

[0294] 장기간 공개 키들은, 개인(또는 비밀) 키와는 달리, 그것들이 다른 디바이스들에 제공될 수 있다는 것을 나타내기 위해 본 명세서에서 "공개"로 지칭된다는 것에 유의해야 한다. 그러나, 프로세스들(1300, 1400, 1500, 1600)에 도시된 바와 같이, 페어 셋업은, 공유 비밀이 다른 정보(예를 들어, 셋업 코드)를 사용해 확립된 이후에, 디바이스들로 하여금 장기간 공개 키들을 암호화된 형태로 교환할 수 있게 한다. 일부 실시예들에서, 장기간 공개 키의 복호화 및 저장은 수신 디바이스 내의 보안 컴퓨팅 요소 내에서 발생할 수 있으며, 이는 또한 장기간 공개 키들이 인가되지 않은 디바이스들에 노출되는 것으로부터 보호할 수 있다. 이것은, 인가되지 않은 디바이스가 페어 셋업 레코드를 위조하는 것을 상당히 더 어렵게 만들 수 있다. 이러한 보안 수준은, 후술되는 바와 같이, 페어 셋업 레코드들이, 디바이스들이 재접속할 때 더 단순한 페어 검증 프로세스에 사용될 수 있게 한다.

[0295] 전술한 페어 셋업 프로세스들 중 임의의 것을 통해 확립된 페어링은, 액세서리 및 제어기가 그들이 수신하는 장기간 공개 키들을 지속적 저장소(예를 들어, 비휘발성 메모리, 자기 디스크 등)에 저장할 수 있다는 점에서, 지속적 상태일 수 있으며; 지속적 저장소는 보안 요소 내에 있을 수 있다. 일부 실시예들에서, 액세서리가 하나의 제어기와 페어 셋업을 수행했으면, 액세서리는 예를 들어, 페어 셋업을 개시하려는 임의의 수신된 요청에 오류 메시지로 응답함으로써, 임의의 다른 제어기가 페어 셋업을 수행하는 것을 방지할 수 있다. 액세서리와 페어링 셋업을 수행한 제어기는 그 액세서리의 "관리자"로서 지정될 수 있고, 예를 들어, 후술되는 바와 같은 페어 추가 프로세스를 사용하여, 다른 제어기들과 페어링들을 확립하도록 액세서리에 지시하는 것이 허용될 수 있다.

[0296] 일부 실시예들에서, 액세서리는 동시에 다수의 제어기들과 페어링을 확립했을 수 있으며; 각각의 페어링은 페어 셋업, 페어 추가(후술됨) 등을 사용해 확립될 수 있다. 예를 들어, 액세서리는, 페어링을 확립한 각 제어기의 식별자 및 연관된 LTPKC를 포함하는 룩업테이블 또는 다른 데이터 구조를 지속적으로 저장할 수 있다. 데이터 구조는 또한, 제어기에 관한 다른 정보를 포함할 수 있다. 예를 들면, 상이한 제어기들은 액세서리에 대해 상이한 정도들의 제어(허가들로 지칭됨)를 승인받을 수 있고, 액세서리에 의해 유지되는 데이터 구조는 각각의 제어기가 어떤 허가들을 갖는지 명시하는 표시자들(예를 들어, 플래그들)을 포함할 수 있다. 다양한 허가들이 정의될 수 있다. 정의될 수 있는 하나의 허가는 "관리자" 허가이고, 일부 실시예들서, 관리자 허가를 갖는 제어기만이 (예를 들어, 후술되는 바와 같은 페어 추가 프로세스를 사용하여) 다른 제어기들에 대한 페어링 레코드들을 액세서리에 추가할 수 있다. 일부 실시예들에서, 페어 셋업을 성공적으로 수행하는 제어기는 자동으로 관리자 허가를 승인받을 수 있다. 다른 제어기들은 선택적으로 (예를 들어, 후술되는 바와 같이) 페어 추가 동안에 관리자 허가를 승인받을 수 있다.

[0297] 유사하게, 제어기는 동시에 다수의 액세서리와 페어링을 확립했을 수 있다. 예를 들어, 제어기는 다수의 액세서리와 페어 셋업을 수행할 수 있거나, 또는 일부 경우들에서, 제어기는 후술하는 바와 같이 페어 추가 프로세스를 통해 액세서리와의 페어링을 추가할 수 있다. 제어기는, 제어기가 페어링을 확립한 각 액세서리의 식별자 및 연관된 LTPKA를 포함하는 룩업테이블 또는 다른 데이터 구조를 지속적으로 저장할 수 있다. 데이터 구조는 또한, 제어기가 그 액세서리에 대해 어떤 허가들을 갖는지와 같은, 정보를 포함할 수 있다.

[0298] 페어링을 확립한 제어기 및 액세서리는 그 후 서로 일정한 통신 상태를 유지하지 않을 수도 있음을 고려한다. 예를 들어, 액세서리 또는 제어기는 전원이 오프일 수 있거나, 또는 하나의 디바이스가 서로의 범위를 벗어나 이동될 수 있다. 서로 페어링을 확립한 제어기 및 액세서리가 통신 상태로 복귀하면, 페어 셋업을 다시 수행하는 대신에, 디바이스들은, 이전에 확립된 페어링의 존재를 검증하기 위하여 상이한 프로세스(본 명세서에서 페어-검증으로 지칭됨)를 사용할 수 있다. 페어 검증은 (예를 들어, 페어 셋업 또는 페어 추가 동안에) 이전에 생성되고 저장된 장기간 공개 키 레코드들을 사용할 수 있다. 페어 검증 프로세스는 또한, 디바이스들이 페어-검증된 상태를 유지하는 동안에 후속 메시지들을 암호화하는 데 사용될 수 있는 새로운 공유 비밀 및/또는 세션 키를 생성할 수 있다. 페어-검증된 상태(본 명세서에서 페어-검증된 세션으로 지칭됨)는, 어느 디바이스에 의해서든 예를 들어, 자신의 세션 키의 복사본을 삭제함으로써 종료될 수 있으며, 이는 그것이 다른 디바이스로부터의 미래의 메시지들을 복호화하는 데 사용될 수 없게 만들 것이다. 일부 실시예들에서, 제어기가 자신과 확립된 페어링을 갖고 있는 액세서리와의 균일한 액세서리 프로토콜 통신을 위한 채널을 열려고 시도할 때마다,

페어 검증이 수행될 수 있다.

[0299] 도 17a 내지 도 17c는 본 발명의 일 실시예에 따른 제어기(1702)와 액세서리(1704) 사이의 페어 검증 프로세스(1700)의 예를 도시한다. 프로세스(1700)는, 제어기(1702)가 페어 검증이 적절한 동작이라고 결정할 때, 예를 들어, 제어기(1702)가 액세서리(1704) 상의 일부 동작을 제어하는 사용자 명령을 수신하고 페어-검증된 세션이 현재 제어기(1702)와 액세서리(1704) 사이에 존재하지 않을 때, 또는 제어기(1702)가 액세서리(1704)와 재접속할 때에, 시작할 수 있다.

[0300] 도 17a를 참조하면, 블록(1706)에서, 제어기(1704)는 예를 들어, 전술한 바와 같이 Curve25519 알고리즘을 사용하여, 새로운 단기간 키 쌍(pkC, skC)을 생성할 수 있다. 블록(1708)에서, 제어기(1704)는 액세서리(1704)에 페어 검증 시작 요청을 송신할 수 있고; 요청은 단기간 공개 키 pkC(및 선택적으로, 페어링이 확립되었을 때 액세서리(1704)에 제공된, 제어기 ID 또는 제어기 사용자명 userC와 같은 추가 정보)를 포함할 수 있다. 일부 실시예들에서, 페어 검증 시작 요청은 액세서리의 /pair-verify URL로의 HTTP POST 요청으로서 송신될 수 있다. 일부 실시예들에서, 상태 표시자는 제어기(1704)가 페어 검증 시작 상태를 요청했음을 나타내기 위해 도 12의 페어링 상태 요청 특성(1201)에 기록될 수 있다.

[0301] 블록(1710)에서, 액세서리(1704)는 페어 검증 시작 요청을 수신할 수 있고, 자신의 페어링된 제어기들의 목록에서 장기간 공개 키(LTPKC)를 찾아볼 수 있다. 일부 실시예들에서, 룩업은 보안 요소 내에서 수행될 수 있고, 액세서리(1704)의 다른 로직 컴포넌트들은 룩업이 성공했는지 여부를 간단히 알 수 있다. 전술한 바와 같이, 제어기 ID(또는 사용자명 userC)를 LTPKC와 연관시키는 페어링 레코드는 페어링이 확립될 때 지속적으로 저장될 수 있어서, 블록(1710)은, 액세서리(1704)로 하여금 제어기(1702)와의 페어링이 이미 확립되어있는지 여부를 결정하게 할 수 있다.

[0302] 블록(1712)에서, 제어기(1702)와의 페어링이 이미 확립되어 있지 않는 경우, 액세서리(1704)는 블록(1714)에서 오류 메시지를 송신할 수 있다. 블록(1716)에서, 제어기(1702)가 오류 메시지를 수신하는 경우, 프로세스(1700)는 블록(1718)에서 종료할 수 있고, 제어기(1702)는 사용자에게 오류를 보고할 수 있다.

[0303] 액세서리(1704)가 그것이 제어기(1702)와 확립된 페어링을 갖고 있다고 결정하는 경우, 이어서 블록(1720)에서, 액세서리(1704)는 예를 들어, Curve25519 알고리즘을 사용하여, 단기간 공개/비밀 키 쌍(pkA, skA)을 생성할 수 있다. 블록(1722)에서, 액세서리(1704)는 skA 및 pkC를 사용하여 공유 비밀("inputKey")을 생성할 수 있다. 블록(1724)에서, 액세서리(1704)는 대칭키("Key")를 도출할 수 있다. 예를 들어, 액세서리(1704)는 inputKey, 솔트(예를 들어, 페어 셋업에서 사용된 솔트들과 상이할 수 있는, 미리정의된 스트링), 및 추가 정보를 입력들로서 사용하는 HDKF-SHA-512를 사용할 수 있다.

[0304] 블록(1726)에서, 액세서리(1704)는 액세서리 정보 메시지를 생성 및 서명할 수 있다. 예를 들어, 액세서리(1704)는 pkA와 pkC를 연결하고, 액세서리의 장기간 비밀 키(LTSKA)로 그 연결을 서명한다. 액세서리 ID와 같은 추가 정보가 또한 연결될 수 있다. 블록(1728)에서, 액세서리(1704)는 서명된 메시지를 대칭키 Key를 사용해 암호화하여, 액세서리 증명(proofA) 및 인증 태그(authTagA)를 생성할 수 있다. 블록(1730)에서, 액세서리(1704)는 제어기(1702)에 페어 검증 시작 응답을 송신할 수 있다. 응답은 proofA, 인증 태그, 및 단기간 공개 키 pkA를 포함할 수 있다. 액세서리 식별자 또는 사용자명("userA", 이는 페어링이 확립되었을 때 제어기에 주어진 액세서리 이름일 수 있음)과 같은 다른 정보가, 또한 포함될 수 있다. 블록(1730)에서 응답을 송신하면, 액세서리(1704)는 액세서리의 증명이 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0305] 블록(1732)에서, 응답을 수신한 후, 제어기(1702)는 예를 들어, 액세서리 ID 또는 액세서리 사용자명에 기초하여, 자신의 페어링된 액세서리들의 목록에서 장기간 공개 키(LTPKA)를 찾아볼 수 있다. 일부 실시예들에서, 룩업은 보안 요소 내에서 수행될 수 있고, 제어기(1702)의 다른 로직 컴포넌트들은 룩업이 성공했는지 여부를 간단히 알 수 있다. 전술한 바와 같이, 액세서리 ID(또는 사용자명 userA)를 LTPKA와 연관시키는 페어링 레코드는 페어링이 확립될 때 지속적으로 저장될 수 있어서, 블록(1732)은, 제어기(1702)로 하여금 액세서리(1704)와의 이미 페어링이 확립되어있는지 여부를 결정하게 할 수 있다.

[0306] 도 17b를 참조하면, 블록(1734)에서, 액세서리(1704)와의 페어링이 이미 확립되어 있지 않은 경우, 프로세스(1700)는 블록(1736)에서 종료할 수 있고, 제어기(1702)는 사용자에게 오류를 보고할 수 있다. 일부 실시예들에서, 제어기(1702)는, 프로세스(1700)를 시작하기 전에, 액세서리(1704)와의 확립된 페어링을 나타내는 저장된 레코드를 제어기(1702)가 갖고 있는지 여부를 결정할 수 있고, 블록(1734)에서의 추가 확인은 생략될 수 있다.

[0307] 제어기(1702)가 그것이 액세서리(1704)와 확립된 페어링을 갖고 있다고 결정하는 경우, 이어서 블록(1738)에서,

제어기(1702)는 skC 및 pkA를 사용하여 공유 비밀("inputKey")을 생성할 수 있다. 블록(1740)에서, 제어기(1702)는 대칭키("Key")를 도출할 수 있다. 예를 들어, 제어기(1702)는 inputKey, 솔트 및 추가 정보 항목을 입력들로서 사용하는 HDKF-SHA-512를 사용할 수 있다. 일부 실시예들에서, 솔트 및 추가 정보 항목은 페어 셋업 동안에 사용되는 미리정의된 값들과는 상이한 미리정의된 값들을 가질 수 있다. 오류가 발생하지 않은 경우, 블록(1740)에서 도출된 Key는 블록(1724)에서 액세서리(1704)에 의해 도출된 Key와 매칭될 수 있다.

[0308] 블록(1742)에서, 제어기(1702)는 대칭키 Key를 사용해 수신된 proofA를 복호화할 수 있으며, 또한 authTagA를 검증할 수 있다. 블록(1744)에서, 제어기(1702)는 proofA로부터 추출된 서명된 액세서리-정보 메시지에 대하여 액세서리의 서명을 검증할 수 있다. 이 검증은 액세서리(1704)와의 확립된 페어링으로부터의 저장된 LTPKA를 사용할 수 있다. 성공하는 경우, 이 검증은, (다른 액세서리가 동일한 장기간 키 쌍을 갖지 않을 것이라는 가정 하에서) 액세서리(1704)가 이전에 LTPKA를 제공했던 액세서리와 동일한 것임을 확인한다. 블록(1746)에서, authTagA 또는 서명이 검증되지 않는 경우(또는 복호화가 실패하는 경우), 프로세스(1700)는 블록(1748)에서 종료할 수 있으며, 제어기(1702)는 사용자에게 오류를 보고할 수 있다.

[0309] 도 17c를 참조하면, 블록(1746)에서 검증이 성공하는 경우, 이어서 블록(1750)에서, 제어기(1702)는 제어기 정보 메시지를 생성 및 서명할 수 있다. 예를 들어, 제어기(1702)는 pkC와 pkA를 연결하고(이 예에서의 순서는 블록(1724)에서의 연결로부터 반전됨), 제어기의 장기간 비밀 키로 그 연결을 서명할 수 있다. 제어기 ID와 같은 추가 정보가 또한 연결될 수 있다. 블록(1752)에서, 제어기(1702)는 서명된 메시지를 대칭키 Key를 사용해 암호화하여, 제어기 증명(proofC) 및 인증 태그(authTagC)를 생성할 수 있다. 블록(1754)에서, 제어기(1702)는 액세서리(1704)에 검증 완료 요청을 송신할 수 있다. 요청은 proofC 및 authTagC를 포함할 수 있다. 블록(1754)에서 요청을 송신하면, 제어기(1702)는 제어기의 증명이 송신되었음을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0310] 블록(1756)에서, 액세서리(1704)는 검증 완료 요청을 수신할 수 있고, 수신된 proofC를 대칭키 Key를 사용해 복호화하고 authTagC를 검증할 수 있다. 블록(1758)에서, 액세서리(1704)는 proofC로부터 추출된 서명된 제어기-정보 메시지에 대하여 제어기의 서명을 검증할 수 있다. 이 검증은 제어기(1702)와의 확립된 페어링으로부터의 저장된 LTPKC를 사용할 수 있다. 성공하는 경우, 이 검증은, (다른 제어기가 동일한 장기간 키 쌍을 갖지 않을 것이라는 가정 하에서) 제어기(1702)가 이전에 LTPKC를 제공했던 제어기와 동일한 것임을 확인한다. 블록(1760)에서, authTagC 또는 서명이 검증되지 않는 경우(또는 복호화가 실패하는 경우), 액세서리(1704)는 블록(1762)에서 제어기(1702)에 오류 응답을 송신할 수 있다. 검증이 성공하는 경우, 액세서리(1704)는 블록(1764)에서 제어기(1702)에 성공 응답을 송신할 수 있다. 어느 경우이든, 응답을 송신하면, 액세서리(1704)는 적절한 응답을 나타내기 위해 페어링 상태를 업데이트할 수 있다.

[0311] 블록(1766)에서, 제어기(1702)는 어떤 응답이 수신되었는지를 결정할 수 있다. 오류 메시지(1762)가 수신된 경우, 프로세스(1700)는 블록(1768)에서 종료할 수 있고, 제어기(1702)는 사용자에게 오류를 보고할 수 있다. 성공 메시지(1764)가 수신된 경우, 페어 검증은 블록(1770)에서 완료되고, 페어링 상태는 그러한 것을 나타내기 위해 업데이트될 수 있다.

[0312] 프로세스(1700)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 예를 들어, 제어기가 액세서리에 메시지를 송신하거나 또는 그 반대일 때마다(예를 들어, 페어링 프로세스 상태가 변화할 때), 오류들이 검출될 수 있다. 일부 오류 조건들이 나타나지만, 오류가 임의의 지점에서 검출되는 경우에, 프로세스(1700)는 종료할 수 있고, 제어기는 오류를 사용자에게 통지할 수 있다는 것이 이해되어야 한다. 또한, 특정 암호화 및/또는 인증 알고리즘들에 대한 참조는 예시의 목적을 위한 것이며, 데이터의 안전한 교환을 위한 다른 프로토콜들 및 알고리즘들이 대체될 수 있다.

[0313] 또한, 프로세스(1700)는, 페어링이 확립되었을 때 장기간 공개 키들이 교환된 (예를 들어, 그 페어 셋업 또는 페어 추가 프로세스가 사용된) 방식에 독립적이라는 것에 유의해야 한다. 액세서리 및 제어기는 각각의 디바이스가 다른 것의 장기간 공개 키를 갖는 한 페어-검증할 수 있으며, 장기간 공개 키는 항상 (예를 들어, 각 디바이스의 보안 요소 내에) 저장된 상태를 유지할 수 있다.

[0314] 또한, 액세서리 및 제어기와 연관된 사용자명들("userA" 및 "userC")은, 페어 검증 동안에 다른 디바이스가 장기간 공개 키를 찾아볼 수 있도록 허용하는 임의의 정보를 포함할 수 있다. 이것은 실제 사용자의 이름 또는 다른 식별자를 포함할 수 있지만, 반드시 그럴 필요는 없다. 일부 실시예들에서, 사용자명은 특정한 장기간 공개 키가 속하는 디바이스의 디바이스 식별자를 포함할(또는 디바이스 식별자일) 수 있다.

[0315] 전술한 다양한 페어 셋업 프로세스들과 마찬가지로, 페어 검증 프로세스(1700)는 새로운 암호화 키(Key)를 생성하는 것을 포함할 수 있다. 일부 실시예들에서, 이 키는 페어 검증 프로세스(1700)의 완료 이후에 송신되는 메시지들 중 임의의 것 또는 전부(예를 들어, 전술한 바와 같은 요청들 및 응답들)를 암호화하는 세션 키로서 사용될 수 있다. 세션 키는 디바이스 중 어느 하나가 자신의 세션 키의 사본을 삭제 또는 무효화할 때까지 지속될 수 있다. 따라서, 어느 디바이스든 자신의 세션 키의 사본을 삭제 또는 무효화함으로써 언제든지 다른 디바이스와의 통신을 일방적으로 차단할 수 있다. 예를 들어, 액세서리는, 제어기가 근접 임계치 밖으로 이동하거나 액세서리와의 접속성을 잃는 경우, 또는 통신이 타임아웃 기간 이내에 발생하지 않는 경우, 또는 세션 지속 시간에 대한 고정 상한(이는 액세서리 제조사 또는 프로그래머가 선택한 만큼 짧거나 길 수 있음) 이후에, 세션 키를 삭제할 수 있다. 이것은 액세서리들이 원하는 대로 제어기의 액세스를 제한할 수 있게 한다.

[0316] 일부 실시예들에서, 페어 검증 프로세스(1700) 동안 도출된 암호화 키는 프로세스(1700) 동안에만 사용된다. 페어-검증 세션 내에서의 후속 통신에 대해, 제어기(1702) 및 액세서리(1704)는 각각 하나 이상의 새로운 세션 키를 계산할 수 있다. 예를 들어, 액세서리-제어기 세션 키(accessory-to-controller session key, "AC session key")는, 제어 솔트 및 추가 정보 항목(이는 미리정의된 상수 값들을 가질 수 있음)과 함께 페어 검증 동안에(예를 들어, 프로세스(1700)의 블록들(1722, 1738)에서) 생성된 공유 비밀(inputKey)에 HKDF-SHA-512(또는 유사한 알고리즘)를 적용함으로써 도출될 수 있다. 제어기-액세서리 세션 키(controller-to-accessory session key, "CA session key")는, 다른 제어 솔트 및 추가 정보 항목(이것 또한 미리정의된 상수 값들을 가질 수 있음)과 함께, 페어 검증 동안에 생성된 공유 비밀(inputKey)에 HKDF-SHA-512(또는 유사한 알고리즘)를 적용함으로써 도출될 수 있다. 일부 실시예들에서, 상이한 제어 솔트들 및/또는 상이한 추가 정보 항목들이 AC 세션 키 및 CA 세션 키를 생성하는 데 사용될 수 있어서, 두 세션 키는 동일해야 할 필요가 없다. 제어기 및 액세서리는 각각 AC 및 CA 세션 키들을 생성할 수 있다. 페어-검증된 세션 내의 후속 통신 동안에, AC 세션 키는, 액세서리에 의해 그것이 제어기에 송신하는 메시지들을 암호화하는 데 사용되고 제어기에 의해 그것이 액세서리로부터 수신하는 메시지들을 복호화하는 데 사용될 수 있는 반면, CA 세션 키는, 제어기에 의해 그것이 액세서리에 송신하는 메시지들을 암호화하는 데 사용되고 액세서리에 의해 그것이 제어기로부터 수신하는 메시지들을 복호화하는 데 사용될 수 있다. 어느 디바이스이든 그것의 세션 키들을 무효화함으로써(예를 들어, 세션 키들을 삭제하거나, 또는 세션 키들을 사용해 암호화되는 임의의 수신된 메시지들에 대해 오류로 응답함으로써) 세션을 종료할 수 있다.

[0317] 또한, 일부 실시예들에서, 단일 제어기는 다수의 장기간 키 쌍(LTPKC, LTSKC)을 정의할 수 있다. 예를 들어, 다수의 인가된 사용자를 갖는 제어기(예를 들어, 공유된 컴퓨터)는 각각의 인가된 사용자에 대해 상이한 장기간 키 쌍을 가질 수 있어서, 상이한 사용자들이 액세서리들의 상이한 서브셋들과 상호작용하게 할 수 있다. 제어기가 각각의 키 쌍에 대해 별개의 사용자명을 갖는 한, 액세서리는 제어기가 하나 초과의 키 쌍을 갖고 있다는 것을 알 필요가 없다. 다른 예로서, 제어기는 상이한 액세서리들과 페어링들을 확립하기 위해 상이한 장기간 키 쌍들을 사용할 수 있다. 다수의 장기간 키 쌍을 사용하는 제어기는, 페어링이 확립되는 각 액세서리에 대해 어느 (LTPKC, LTSKC) 쌍이 사용되었는지를 추적할 수 있다. 유사하게, 액세서리는 다수의 장기간 키 쌍(LTPKA, LTSKA)을 가질 수 있고, 페어링이 확립되는 각 제어기에 대해 어느 쌍이 사용되었는지를 추적할 수 있다. 일부 실시예들에서, 디바이스는 그것이 특별한 장기간 공개 키를 제공하는 다른 디바이스들의 개수를 제한할 수 있고, 다수의 장기간 공개 키를 갖는 것은 디바이스가 시간에 따라 상이한 키로 전환되게 할 수 있다.

[0318] 일부 실시예들에서, 장기간 공개 키들(또는 일부 경우들에서의, 인증서들)은, "페어 추가", 또는 페어링을 추가하는 것으로 지칭될 수 있는 프로세스에서, 페어 셋업 또는 페어 검증 이후에 언제든지 디바이스들 사이에서 교환될 수 있다. 예를 들어, 전술한 바와 같이, 액세서리는 자신을 하나의 제어기(이는 액세서리에 대한 "관리자"("administrator" 또는 "admin")로 지칭될 수 있음)와 페어 셋업을 수행하는 것으로 제한할 수 있고, 첫 번째 성공적인 페어 셋업 이후의(예를 들어, 후술되는 바와 같이, 적어도 페어링이 제거될 때까지) 모든 후속 페어 셋업 요청들을 거절할 수 있다. 다른 제어기들로 하여금 액세서리와 상호작용하도록 하기 위해, 관리자 제어기는, 상이한 제어기와 액세서리 사이에(또는 일부 경우들에서는, 동일한 제어기의 상이한 장기간 공개 키와 액세서리 사이에) 페어링을 확립하기 위해 페어 추가 프로세스를 수행할 수 있다.

[0319] 도 18a 및 도 18b는 본 발명의 일 실시예에 따른 제어기(1802)(예를 들어, 관리자 허가를 갖는 제어기)와 액세서리(1804) 사이의 페어 추가 프로세스(1800)의 예를 도시한다. 이 프로세스에서, 제어기(1802)는 이전에 액세서리(1804)와 페어 셋업을 수행했거나(그리고 그렇게 함으로써 관리자 허가를 획득했음), 또는 페어 추가 프로세스(1800)를 실행하는 이전의 인스턴스를 통해 관리자로서 추가되었음이 가정된다.

[0320] 도 18a를 먼저 참조하면, 블록들(1806, 1808)에서, 제어기(1802) 및 액세서리(1804)는 공유 비밀 및 세션

키(들)(예를 들어, 전술한 바와 같은 AC 키 및 CA 키)를 확립하기 위해 페어 검증 프로세스(예를 들어, 프로세스(1700))를 완료할 수 있다.

[0321] 블록(1810)에서, 제어기(1802)는 액세서리(1804)와 교환할 장기간 공개 키(LTPKN)를 식별할 수 있다. 이것은, 예를 들어, 페어링이 확립될 제어기(본 명세서에서 "새로운" 제어기로도 지칭됨)에 속하는 장기간 공개 키일 수 있으며; 이것은 제어기(1802) 이외의 제어기일 수 있다. 일부 경우들에서, 보안 인증서(이는 장기간 공개 키를 포함할 수 있음)가 새로운 제어기에 대하여 획득될 수 있다. 블록(1812)에서, 제어기(1802)는, 데이터 블록이 페어링을 추가하려는 요청에 관련된다는 표시, 장기간 공개 키 LTPKN, 새로운 제어기의 제어기 식별자, 및 새로운 제어기에 승인될 허가들을 나타내는 허가 정보(예를 들어, 플래그들)를 포함하는 데이터 블록을 생성할 수 있다. 예를 들어, 전술한 바와 같이, 액세서리와 페어 셋업을 수행하는 첫 번째 제어기는 그 액세서리에 대한 관리자로서 자동으로 지정될 수 있다. 페어 추가 프로세스(1800)를 통해 추가되는 각각의 새로운 제어기에 대해, 허가 정보는 새로운 제어기가 또한 관리자로서 지정되어야 하는지 여부를 명시할 수 있다. 일부 실시예들에서, 액세서리에 대한 관리자들은 그 액세서리에 대해 페어 추가를 수행하는 것이 허용되지만, 관리자들이 아닌 제어기들은 페어 추가를 수행하도록 허용되지 않는다.

[0322] 블록(1814)에서, 제어기(1802)는 액세서리(1804)에 페어 추가 시작 요청을 송신할 수 있고; 요청은 블록(1812)에서 생성된 데이터 블록을 포함할 수 있다. 페어-검증된 세션 내의 모든 통신들과 마찬가지로, 요청은 적절한 세션 키를 사용해 암호화될 수 있다. 일부 실시예들에서, 페어 추가 시작 요청은 페어 추가 시작 요청을 식별하는 상태 표시자를 포함할 수 있다(이 상태 표시자 및 후속 상태 표시자들은 도 12의 페어링 상태 요청 특성(1201)에 기록될 수 있음). 일부 실시예들에서, 페어 추가 시작 요청은 액세서리(1804)의 /pairings URL(또는 다른 적절한 URL)로의 HTTP POST 요청으로서 송신될 수 있다.

[0323] 블록(1816)에서, 액세서리(1804)는 페어 추가 시작 요청을 수신할 수 있다. 페어-검증된 세션 내의 제어기로부터의 임의의 수신된 요청과 마찬가지로, 액세서리(1804)는 적절한 세션 키를 사용해 요청을 복호화함으로써 시작할 수 있고; 복호화가 실패하는 경우, 액세서리(1804)는 오류 응답을 반환할 수 있다(또는 전혀 응답하지 않음). 블록(1818)에서, 액세서리(1804)는 제어기(1802)가 페어 추가를 수행하도록 허용되는지 여부를 결정할 수 있다. 예를 들어, 전술한 바와 같이, 제어기들은 선택적으로 관리자 허가를 승인받을 수 있고, 페어 추가는 관리자 허가를 갖는 제어기들로 제한될 수 있다. 다른 예로서, 사용자 인터페이스를 갖는 액세서리는 페어 추가 요청을 허용할지 여부를 나타내도록 사용자에게 프롬프트할 수 있다. 또 다른 예로서, 전술한 바와 같이, 일부 경우들에서 사용자는 기계적 동작을 통해 액세서리가 페어링 모드로 진입하게 할 수 있고, 일부 액세서리들은 페어링 모드 동안에만 페어 추가 요청을 허용하도록 구성될 수 있다. 또 다른 예로서, 일부 실시예들에서, 액세서리(1804)는 동시에 저장될 수 있는 페어링들의 수에 대한 상한(예를 들어, 16 페어링, 32 페어링, 또는 어떤 다른 한계)을 가질 수 있고, 액세서리(1804)는 그것이 이 한계를 초과하는 결과를 생성할 경우 허용되지 않는(unpermitted) 것으로서 페어 추가 요청을 처리할 수 있다. 액세서리(1804)가 특정한 페어 추가 요청을 허용해야 하는지 여부를 결정하기 위해 다른 기술들이 또한 사용될 수 있다. 요청이 허용되지 않는 경우, 이어서 액세서리(1804)는 블록(1820)에서 오류 메시지를 송신할 수 있다.

[0324] 도 18b를 참조하면, 페어 추가 요청이 허용되는 경우, 이어서 블록(1832)에서, 액세서리(1804)는 수신된 메시지에 기초하여 새로운 페어링을 (예를 들어, 그것의 보안 요소 내에) 지속적으로 저장할 수 있다. 예를 들어, 액세서리(1804)는 수신된 장기간 공개 키 LTPKN을 새로운 제어기에 대한 수신된 제어기 식별자 및 허가들과 연관하여 저장할 수 있다.

[0325] 일부 실시예들에서, 액세서리(1804)는 프로세스(1800) 동안 제어기(1802)에 장기간 공개 키(LTPKA)를 제공할 필요가 없는데, 이는 제어기(1802)가 프로세스(1800)을 개시하기 이전에 그것을 수신했을 것이기 때문이다. 그러나, 다른 실시예들에서는, 액세서리(1804)가 상이한 제어기들과 상이한 장기간 공개 키들을 사용하는 것이 바람직할 수 있다. 이런 경우, 액세서리(1804)는 새로운 제어기에 의해 사용되어야 하는 장기간 공개 키를 포함하는 데이터 블록을 준비할 수 있다. 예를 들어, 블록(1834)에서, 액세서리(1804)는 새로운 제어기에 의해 사용될 장기간 공개 키를 식별할 수 있으며; 이것은 이전에 제어기(1802)에 제공된 장기간 공개 키(LTPKA)와 동일하거나 상이할 수 있다. 블록(1836)에서, 액세서리(1804)는, 블록(1834)에서 식별된 장기간 공개 키, 및 새로운 제어기에 의해 사용될 액세서리 식별자(이는 이전에 제어기(1802)에 제공된 액세서리 식별자와 동일하거나 상이할 수 있음)를 포함하는 데이터 블록을 생성할 수 있다.

[0326] 블록(1838)에서, 액세서리(1804)는 제어기(1802)에 페어 추가 응답을 송신할 수 있다. 데이터 블록이 블록(1836)에서 생성된 경우, 데이터 블록은 페어 추가 응답에 포함될 수 있다. 페어-검증된 세션 내의 모든 통신

들과 마찬가지로, 응답은 적절한 세션 키를 사용해 암호화될 수 있다. 페어 추가 응답은 페어 추가 응답이 송신되었음을 나타내기 위해 상태 표시자를 업데이트하는 것을 포함할 수 있다.

[0327] 블록(1840)에서, 제어기(1802)는 페어 추가 응답을 수신할 수 있다. 페어-검증된 세션 내의 액세서리로부터의 임의의 수신된 응답과 마찬가지로, 제어기(1802)는 적절한 세션 키를 사용해 응답을 복호화함으로써 시작할 수 있으며; 복호화가 실패하는 경우, 응답은 무시될 수 있고 프로세스(1800)는 오류로 종료할 수 있다.

[0328] 블록(1844)에서, 제어기(1802)는 응답이 성공을 나타내는지 여부를 결정할 수 있다. 그렇지 않다면, 프로세스(1800)는 블록(1846)에서 종료할 수 있고, 제어기(1802)는 사용자에게 오류를 통지할 수 있다. 응답이 성공을 나타내는 경우, 이어서 블록(1848)에서, 제어기(1802)는 새로운 제어기에 페어링을 통지할 수 있다. 예를 들어, 제어기(1802)는 액세서리(1804)에 대한 액세서리 식별자 및 장기간 공개 키 LTPKA를 새로운 제어기로 전달할 수 있다. 일부 실시예들에서, 제어기(1802)는 액세서리(1804)에 대한 이전에 저장된 LTPKA 및 액세서리 식별자를 새로운 제어기에 제공할 수 있고; 다른 실시예들에서, 제어기(1802)는 페어 추가 응답에서 액세서리(1804)에 의해 제공된 정보를 새로운 제어기에 제공할 수 있다. 새로운 제어기는 수신된 LTPKA 및 액세서리 식별자를 페어링 레코드로서 지속적으로 저장할 수 있다. 그 후, 새로운 제어기는 액세서리(1804)를 이용해(제어기(1802)의 추가 개입 없이) 페어 검증 프로세스(예를 들어, 프로세스(1700))를 수행할 수 있고, 액세서리(1804)와 상호작용할 수 있다.

[0329] 프로세스(1800)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 예를 들어, 제어기가 액세서리에 메시지를 송신하거나 또는 그 반대일 때마다(예를 들어, 페어링 프로세스 상태가 변화할 때), 오류들이 검출될 수 있다. 일부 오류 조건들이 나타나지만, 오류가 임의의 지점에서 검출되는 경우, 프로세스(1800)는 종료할 수 있고, 제어기는 오류를 사용자에게 통지할 수 있다는 것이 이해되어야 한다. 또한, 특정 암호화 및/또는 인증 알고리즘들에 대한 참조는 예시의 목적을 위한 것이며, 데이터의 안전한 교환을 위한 다른 프로토콜들 및 알고리즘들이 대체될 수 있다. 일부 실시예들에서, 페어 추가 프로세스(1800)는 후술되는 바와 같이 위임된 페어링의 모드로서 사용될 수 있다.

[0330] 전술한 바와 같이, 페어 셋업 및 페어 추가는, 수신 디바이스들에 의해 지속적으로 그리고 안전하게 저장될 수 있는 장기간 공개 키들을 교환함으로써 액세서리들과 제어기들 사이에 페어링이 확립되도록 할 수 있다. 일부 경우들에서, 예를 들어, 지속적 저장소로부터 장기간 공개 키를 제거함으로써 확립된 페어링을 제거하는 것이, 바람직할 수 있다. 따라서, 본 발명의 특정 실시예들은 페어 제거 프로세스들을 제공한다.

[0331] 도 19a 및 도 19b는 본 발명의 일 실시예에 따른 제어기(1902)(예를 들어, 관기자 허가를 갖는 제어기)와 액세서리(1904) 사이의 페어 제거 프로세스(1900)의 예를 도시한다. 프로세스(1900)는 프로세스(1800)와 대체로 유사할 수 있으며, 다만 프로세스(1900)는 새로운 페어링을 확립하는 대신에 기존의 페어링을 제거하는 결과를 발생시킨다. 이 프로세스에서, 제어기(1902)는 이전에 액세서리(1904)와 페어 셋업을 수행했거나(그리고 그렇게 함으로써 관리자 허가를 획득했음), 또는 예를 들어, 페어 추가 프로세스(1800)의 실행을 통해 관리자로서 추가되었음이 가정된다.

[0332] 도 19a를 참조하면, 블록들(1906, 1908)에서, 제어기(1902) 및 액세서리(1904)는 공유 비밀 및 세션 키(들)(예를 들어, 전술한 바와 같은 AC 키 및 CA 키)를 확립하기 위해 페어 검증 프로세스(예를 들어, 프로세스(1700))를 완료할 수 있다.

[0333] 블록(1910)에서, 제어기(1902)는 제어기의 제거될 식별자를 획득할 수 있다. 이것은 액세서리(1804)에 의해 페어링 레코드에 저장되는 식별자일 수 있다. 일부 경우들에서, 이것은 제어기(1902) 자신의 식별자일 수 있고(제어기는 자신의 페어링을 제거할 수 있음); 다른 경우들에서, 그것은 다른 제어기의 식별자일 수 있다. 일부 경우들에서, 블록(1910)은, 제어기의 장기간 공개 키가 식별자에 더하여 제거되게 하는 것을 포함할 수 있다. 블록(1912)에서, 제어기(1902)는, 데이터 블록이 페어링을 제거하려는 요청에 관련된다는 것을 나타내는 표시, 및 페어링이 제거되어야 할 제어기의 식별자를 포함하는, 데이터 블록을 생성할 수 있다. 일부 실시예들에서, 데이터 블록은, 제거되고 있는 제어기의 장기간 공개 키와 같은, 다른 정보를 포함할 수 있다.

[0334] 블록(1914)에서, 제어기(1902)는 액세서리(1904)에 페어 제거 시작 요청을 송신할 수 있고; 요청은 블록(1912)에서 생성된 데이터 블록을 포함할 수 있다. 페어-검증된 세션 내의 모든 통신들과 마찬가지로, 요청은 적절한 세션 키를 사용해 암호화될 수 있다. 일부 실시예들에서, 페어 제거 시작 요청은 페어 제거 시작 요청이 송신되었음을 나타내는 상태 표시자를 포함할 수 있다(이 상태 표시자 및 후속 상태 표시자는 도 12의 페어링 상태

요청 특성(1201)에 기록될 수 있음). 일부 실시예들에서, 페어 제거 시작 요청은 액세서리(1904)의 /pairings URL(또는 다른 적절한 URL)로의 HTTP POST 요청으로서 송신될 수 있다.

[0335] 블록(1916)에서, 액세서리(1904)는 페어 제거 시작 요청을 수신할 수 있다. 페어-검증된 세션 내의 제어기로부터의 임의의 수신된 요청과 마찬가지로, 액세서리(1904)는 적절한 세션 키를 사용해 요청을 복호화함으로써 시작할 수 있으며; 복호화가 실패하는 경우, 액세서리(1904)는 오류 응답을 반환할 수 있다(또는 전혀 응답하지 않음). 블록(1918)에서, 액세서리(1904)는 제어기(1902)가 페어 제거를 수행하도록 허용되는지 여부를 결정할 수 있다. 예를 들어, 전술한 바와 같이, 제어기들은 선택적으로 관리자 허가를 승인받을 수 있고, 페어 제거는 관리자 허가를 갖는 제어기들로 제한될 수 있다. 다른 예로서, 사용자 인터페이스를 갖는 액세서리는 페어 제거 요청을 허용할지 여부를 나타내도록 사용자에게 프롬프트할 수 있다. 또 다른 예로서, 전술한 바와 같이, 일부 경우들에서 사용자는 기계적 동작을 통해 액세서리가 페어링 모드로 진입하게 할 수 있으며, 일부 액세서리들은 페어링 모드 동안에만 페어 제거 요청을 허용하도록 구성될 수 있다. 액세서리(1904)가 특정한 페어 제거 요청을 허용해야 하는지 여부를 결정하기 위해 다른 기술들이 또한 사용될 수 있다. 요청이 허용되지 않는 경우, 이어서 액세서리(1904)는 블록(1920)에서 오류 메시지를 송신할 수 있다.

[0336] 도 19b를 참조하면, 페어 제거 요청이 허용되는 경우, 이어서 블록(1932)에서, 액세서리(1904)는, (예를 들어, 페어링된 레코드를 보안 저장 요소로부터 삭제함으로써) 수신된 데이터 블록 내에 명시된 제어기와의 페어링을, 자신의 확립된 페어링들의 목록으로부터 제거할 수 있다.

[0337] 일부 실시예들에서, 액세서리(1904)는 프로세스(1900) 동안에 장기간 공개 키(LTPKA)를 제거하기 위해 역수(reciprocal) 명령어를 제공할 필요가 없다. 액세서리(1904)가 페어링 레코드를 제거한 이후에, 제거된 제어기는 액세서리(1904)와 페어 검증을 수행할 수 없을 것이고, 이것은, 제거된 제어기가 또한 자신의 페어링 레코드를 제거하는지 여부에 상관없이, 제거된 제어기가 액세서리(1904)와 상호작용하는 것을 방지할 수 있다. 그러나, 다른 실시예들에서는, 액세서리(1904)가 제거될 페어링을 명시하는 것이 바람직할 수 있다. 이런 경우, 액세서리(1904)는, 새롭게 제거되는 제어기에 의해 제거되어야 하는 장기간 공개 키 및 액세서리 식별자를 포함하는 데이터 블록을 준비할 수 있다. 예를 들어, 블록(1934)에서, 액세서리(1904)는 새롭게 제거되는 제어기로부터 제거되어야 하는 장기간 공개 키를 식별할 수 있으며; 이것은, 예를 들어, 제어기가 추가되었을 때 프로세스(1800)의 블록(1834)에서 식별된 키일 수 있다. 블록(1936)에서, 액세서리(1904)는, 블록(1934)에서 식별된 장기간 공개 키 및 이 장기간 공개 키와 연관된 액세서리 식별자를 포함하는, 데이터 블록을 생성할 수 있다.

[0338] 블록(1938)에서, 액세서리(1904)는 제어기(1902)에 페어 제거 응답을 송신할 수 있다. 데이터 블록이 블록(1936)에서 생성된 경우, 데이터 블록은 페어 제거 응답에 포함될 수 있다. 페어-검증된 세션 내의 모든 다른 통신들과 마찬가지로, 응답은 적절한 세션 키를 사용해 암호화될 수 있다. 페어 제거 응답은 페어 제거 응답이 송신되었음을 나타내기 위해 상태 표시자를 업데이트하는 것을 포함할 수 있다.

[0339] 블록(1940)에서, 제어기(1902)는 페어 제거 응답을 수신할 수 있다. 페어-검증된 세션 내의 액세서리로부터의 임의의 수신된 응답과 마찬가지로, 제어기(1902)는 적절한 세션 키를 사용해 응답을 복호화함으로써 시작할 수 있으며; 복호화가 실패하는 경우, 응답은 무시될 수 있고 프로세스(1900)는 오류로 종료할 수 있다.

[0340] 블록(1944)에서, 제어기(1902)는 응답이 성공을 나타내는지 여부를 결정할 수 있다. 그렇지 않다면, 프로세스(1900)는 블록(1946)에서 종료할 수 있고, 제어기(1902)는 사용자에게 오류를 통지할 수 있다. 응답이 성공을 나타내는 경우, 이어서 블록(1948)에서, 제어기(1902)는 제거된 제어기에 그것의 페어링이 제거되었음을 통지할 수 있다. 일부 실시예들에서, 제어기(1902)는 또한 액세서리(1904)에 대한 액세서리 식별자 및/또는 장기간 공개 키 LTPKA를 제거된 제어기로 전달할 수 있다. 그 후, 제거된 제어기는 더 이상 액세서리(1904)와 페어 검증 프로세스를 수행할 수 없으며, 이는 제거된 제어기가 액세서리(1904)와 상호작용할 수 없게 되는 결과를 발생시킬 수 있다. 프로세스(1900)를 통해 제거된 제어기는, 예를 들어, 페어 추가 프로세스(1800)를 통해 나중에 다시 추가될 수 있다. 일부 실시예들은 제거된 제어기를 "블랙리스트"에 올리는 옵션을 액세서리(1904)에 제공할 수 있으며, 이는 제거된 제어기가 액세서리(1904)와 페어링을 재확립하는 것을 방지할 수 있다. 예를 들어, 제어기(1902)로부터의 페어 제거 요청은 제거된 제어기가 블랙리스트에 올려져야 하는지 여부에 관한 표시를 포함할 수 있으며, 액세서리(1904)는 블랙리스트에 올려진 제어기들의 목록을 지속적으로 저장할 수 있다. 페어 셋업 또는 페어 추가 동안에, 액세서리(1904)는 블랙리스트를 확인하고 제어기가 블랙리스트 상에 있는 경우 오류를 반환할 수 있다.

[0341] 프로세스(1900)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될

수 있다. 예를 들어, 제어기가 액세서리에 메시지를 송신하거나 또는 그 반대일 때마다(예를 들어, 페어링 프로세스 상태가 변화할 때), 오류들이 검출될 수 있다. 일부 오류 조건들이 나타나지만, 오류가 임의의 지점에서 검출되는 경우, 프로세스(1900)는 종료할 수 있고, 제어기는 오류를 사용자에게 통지할 수 있다는 것이 이해되어야 한다. 또한, 특정 암호화 및/또는 인증 알고리즘들에 대한 참조는 예시의 목적을 위한 것이며, 데이터의 안전한 교환을 위한 다른 패스워드 프로토콜들 및 알고리즘들이 대체될 수 있다. 일부 실시예들에서, 페어 제거 프로세스(1900)는 후술되는 바와 같이 위임된 페어링과 관련하여 사용될 수 있다.

[0342] 전술한 실시예들은 액세서리들 및 제어기들로 하여금 페어링을 생성(셋업), 검증, 추가 및 제거할 수 있게 하며, 여기서 페어링은, 페어링 내의 각각의 파트너 디바이스에 의한, 다른 파트너 디바이스의 장기간 공개 키 및/또는 인증서의 지속적 저장을 포함할 수 있다.

[0343] 일부 실시예들에서, 사용자는, 주어진 액세서리와 페어링된 제어기들의 목록, 또는 주어진 제어기와 페어링된 액세서리들의 목록을 획득하는 것이 유용할 수 있다. 후자의 경우(제어기와 페어링된 액세서리들), 제어기 상에서 실행되는 프로그램(예를 들어, 운영 체제 또는 애플리케이션 프로그램)은 제어기의 저장된 페어링 정보를 판독함으로써 목록을 생성할 수 있고, 제어기의 자신의 사용자 인터페이스 상에서 사용자에게 목록을 제시할 수 있다. 주어진 액세서리와 페어링된 제어기들의 목록을 제공하기 위해, 특정 실시예들은 제어기로 하여금, 액세서리로부터 모든 페어링된 제어기들의 목록을 검색할 수 있게 하고; 제어기는 이어서 사용자에게 목록을 제시할 수 있다.

[0344] 예를 들어, 도 12에 도시된 바와 같이, 액세서리는 그것의 페어링 프로파일의 일부로서 페어링 목록 특성(1204)을 가질 수 있다. 특성(1204)은 액세서리와 확립된 페어링을 갖는 각 제어기에 관한 정보(예를 들어, 제어기 ID 등)를 포함할 수 있지만, 임의의 제어기의 장기간 공개 키를 포함할 필요는 없다. 일부 실시예들에서, 제어기는 예를 들어, HTTP GET 요청 등을 사용하여, 임의의 다른 특성에서와 같이 페어링 목록 특성(1204)을 판독할 수 있다. 일부 실시예들에서, 제어기는 액세서리의 /pairings URL(또는 다른 적절한 URL)로의 HTTP POST 요청을 송신함으로써 액세서리의 페어링 목록을 획득할 수 있고; POST 요청은 페어링 목록이 요청되고 있음을 나타내는 항목을 포함할 수 있다.

[0345] 일부 실시예들에서, 제어기는 페어-검증된 세션 내에서만 페어링 목록(1204)을 판독하도록 허용되어서, 액세서리 응답이 암호화되게 할 수 있다. 또한, 일부 실시예들에서, 페어링 목록(1204)의 판독은 관리자 허가를 갖는 제어기들로만 제한될 수 있다.

[0346] 일부 실시예들에서, 액세서리는 한 번에 하나의 제어기와만 페어링을 확립할 수 있고, 페어링을 확립하는 제어기만이 나중에 페어링을 제거하도록 허용된다. 이것은 보안을 향상시킬 수 있다. 그러나, 일부 애플리케이션들의 경우, 그것은 불편하게 제한적일 수 있다. 예를 들어, 홈 환경(100)(도 1)의 경우, 다수의 사람들이 홈에 사는 경우, 각각의 사람이 문 잠금장치(104)를 잠금해제할 수 있는 제어기(102)(예를 들어, 휴대 전화)를 갖는 것이 바람직할 수 있다.

[0347] 따라서, 본 발명의 특정 실시예들은 액세서리로 하여금 다수의 제어기와 동시에 페어링을 유지하도록 허용할 수 있다. 예를 들어, 각각의 제어기는 전술한 페어 셋업 프로세스들을 사용하여 개별적으로 페어링을 확립할 수 있다. 그러나, 제어기들이 독립적으로 페어링을 확립하도록 하는 것은, 일부 애플리케이션들에 대해 불충분하게 안전할 수 있다. 예를 들어, 집의 현관 문에 있는 잠금장치의 경우, 주택 소유자는 다른 사람이 주택 소유자의 명시적 허가 없이 페어링을 확립하는 것을 방지하기를 원할 수 있다.

[0348] 관리형(managed) 페어링을 허용하기 위해, 특정 실시예들은 "위임된(delegaged)" 페어링 프로세스들을 제공할 수 있다. 위임된 페어링 프로세스에서, 제1("관리자" 또는 "마스터") 제어기는 예를 들어, 전술한 바와 같은 페어 셋업을 사용해 액세서리와 페어링을 확립할 수 있고, 그로 인해 관리자 허가를 획득할 수 있다. 그 후, 관리자 제어기는 임의의 후속("위임된") 제어기들에 대한 페어 셋업 프로세스들에 참여하는 것이 요구될 수 있다.

[0349] 위임된 페어링의 하나의 유형은 "직접" 위임일 수 있다. 직접 위임에서, 마스터 제어기는 도 18a 및 도 18b의 페어 추가 프로세스(1800) 또는 유사한 프로세스를 사용하여, 위임된 제어기에 대한 제어기 공개 키(및 제어기 사용자 이름)를 액세서리에 전달할 수 있다. 직접 위임을 위해, 마스터 제어기는 위임된 제어기의 장기간 공개 키를 획득하기 위하여 위임된 제어기와 통신할 수 있다.

[0350] 위임된 페어링의 다른 유형은 "외부", 또는 "포워드된" 위임일 수 있다. 외부 위임에서, 액세서리는 페어 셋업 및 페어 검증을 "인가자(authorizer)" 디바이스에 위임하도록 구성될 수 있으며, 인가자 디바이스는 마스터 제

어기 또는 마스터 제어기와 통신할 수 있는 일부 다른 디바이스에 통합될 수 있다. 인가자 디바이스는 전술한 바와 같이 액세서리와 페어 셋업을 수행할 수 있고, 페어 셋업 이후에, 인가자는 액세서리로의 보안 채널을 유지할(또는 페어 검증을 사용해 재확립할) 수 있다. 액세서리가 페어 셋업 또는 페어 검증 요청을 수신하는 경우, 액세서리는 보안 채널을 통해 인가자에게 요청을 포워드할 수 있다. 인가자는 동작을 수행하고/하거나 동작이 허용되어야 하는지 여부를 액세서리에 나타낼 수 있다.

[0351] 위임된 페어링의 제3 유형은 "인증서-기반" 위임일 수 있다. 인증서-기반 위임에서, 마스터 제어기는 신뢰 인증서 체인(trust certificate chain)으로 액세서리를 구성할 수 있다. 예를 들어, 마스터 제어기는 신뢰 인증서 체인을 액세서리에 안전하게 전달하기 위해, 도 18a 및 도 18b의 페어 추가 프로세스(1800) 또는 유사한 프로세스를 사용할 수 있다. 예를 들어, "인증서 체인 설정 요청(set certificate chain request)" 메시지는, 제어기가 액세서리에 송신하는 것으로 정의될 수 있다(이것은 예를 들어, 인증서 체인들을 수신하도록 정의된 특성 또는 URL로의 HTTP PUT 요청일 수 있다). 이 요청 메시지의 콘텐츠는, 예를 들어, 루트로부터 가장 깊은 서브-인가(sub-authority)의 순서로, 인증서 체인을 포함하는, TLV 항목 또는 TLV 항목들의 시리즈를 포함할 수 있다. TLV 항목(들)은 페어 검증 동안에 확립된 대칭키로 암호화되고, 마스터 제어기의 장기간 비밀 키 LTSKC를 사용해 디지털로 서명될 수 있다. 액세서리는 저장된 LTPKC를 사용해 서명을 검증하고, 인증서 체인을 추출하기 위해 대칭키를 사용하여 암호화된 TLV 항목(들)을 복호화할 수 있다. 일부 실시예들에서, 인증서 체인은 액세서리의 보안 저장 요소에 저장될 수 있다. 액세서리는 성공 또는 실패를 나타내는 응답 메시지를 송신할 수 있다.

[0352] 다른 제어기가 액세서리와 페어 셋업을 수행하려고 시도할 때, 그 제어기는 그것의 장기간 공개 키에 추가하여 또는 그 대신에 인증서를 제공하도록 요구될 수 있다. 액세서리는 인증서가 이전에-수신된 신뢰 인증서 체인에 의해 서명되어 있는지 여부를 결정할 수 있고, 인증서가 그렇게 서명되어 있거나 서명되어 있지 않은지에 기초하여 페어 셋업을 수락 또는 거부할 수 있다. 위임된 제어기들은, (예를 들어, 마스터 제어기, 신뢰 인증서 기관 등으로부터) 일부 다른 채널을 통해 적절히 서명된 인증서를 획득하고 그것을 액세서리에 제시함으로써, 액세서리에 대해 인가될 수 있다.

[0353] 통상의 기술자는, 본 명세서에서 기술된 페어링 프로세스들 및 아키텍처들이 예시적인 것이며, 변형들 및 수정들이 가능하다는 것을 이해할 것이다. 상이한 암호화 알고리즘들, 키들 및 프로토콜들이 대체될 수 있고, 다수의 상이한 알고리즘들, 키들, 및 프로토콜들이 주어진 동작 환경에서 동시에 지원될 수 있다.

[0354] 예시적인 액세서리: 문 잠금장치

[0355] 본 발명의 실시예들에 존재할 수 있는 다양한 양태 및 특징을 추가로 예시하기 위해, 특정한 액세서리 예들이 이제 기술될 것이다.

[0356] 첫 번째 예는, 문에 설치될 수 있는 문 잠금장치에 결합된 전자식 잠금 유닛을 포함할 수 있는, 문 잠금장치 액세서리이다. 일부 실시예들에서, 문 잠금장치는 자체가 기계식 잠금장치(예를 들면, 데드볼트 유형)일 수 있고, 전자식 잠금 유닛은 기계식 잠금장치를 잠금 위치와 잠금해제 위치 사이에서 이동시키는 전자기계식 액추에이터들을 동작시킬 수 있다. 위치 센서들 등이 또한 전자식 잠금 유닛으로 하여금, 기계식 잠금장치가 현재 잠금 위치 또는 잠금해제 위치에 있는지 여부를 결정하게 하도록 제공될 수 있다. 다른 실시예들에서, 다른 유형들의 문 잠금장치들이, 예를 들어, 자기 잠금장치, 전자식 잠금장치, 및 전기 제어 신호를 공급하고/하거나 기계적 힘을 가함으로써 잠금 또는 잠금해제될 수 있는 임의의 다른 유형의 잠금장치가 사용될 수 있다. 전자식 잠금 유닛은, 전술한 바와 같은 균일한 액세서리 프로토콜을 구현하는 로직 회로 및 하나 이상의 제어기와 통신하는 통신 회로를 내장하거나 그에 접속될 수 있다. 전자 잠금 유닛은 예를 들어, 전자기계식 액추에이터들 등을 통해 문 잠금장치를 동작시키기 위한 전기 신호들(예를 들어, 하나 이상의 와이어 상의 전압 또는 전류 레벨들)을 생성할 수 있다. 일부 실시예들에서, 전자 잠금 유닛 및 문 잠금장치는, 문에, 또는 문 또는 문 프레임 내부에 부착되는 모듈과 같은, 일반적인 하우징 내에 물리적으로 위치될 수 있다. 다른 실시예들에서, 전자식 잠금 유닛 및 문 잠금장치는 별개의 하우징들 내에 있을 수 있다. 예를 들어, 문 잠금장치가 문 내부에 있을 수 있는 반면, 전자식 잠금 유닛은 근처의 벽에 장착될 수 있다. 문 잠금장치 및 전자식 잠금 유닛 내의 액추에이터들 사이에 유선 접속들이 제공될 수 있다. 무선 접속들이 또한 본 발명의 범위로부터 벗어나지 않고 사용될 수 있지만, 통상의 기술자는 이러한 콘텍스트에서의 무선 접속이 추가적인 보안 문제를 제기할 수 있다는 것을 이해할 것이다.

[0357] 도 20을 참조하면, 본 발명의 일 실시예에 따른 문 잠금 액세서리(2004)를 위한 동작 환경(2000)이 도시된다. 동작 환경(2000)은 예를 들어, 사무실 건물, 집, 아파트 또는 호텔 건물, 또는 적어도 하나의 내부 또는 외부

문(2006)을 갖는 임의의 다른 구조물에서 구현될 수 있다. 예시의 목적을 위해, 문(2006)을 잠금 및 잠금해제하는 능력을 다수의 개인에게 제공하는 것이 바람직하다고 가정된다. 또한 문(2006)이 "소유자"(즉, 누가 문(2006)을 통과하는 것이 허용되어야 할지 허용되지 않아야 할지를 결정하도록 법적으로 또는 계약에 의해 권리가 주어진, 어떤 사람 또는 엔티티)를 갖는다고 가정된다. 예를 들어, 환경(2000)이 사무실 건물에서 구현되는 경우, 건물의 소유자(또는 소유자의 권한에 따라 행동하는 세입자)가 문(2006)의 소유자일 수 있다. 환경(2000)이 집에서 구현되는 경우, 소유자는 주택 소유자, 세대주, 또는 거주자들에 의해 지정된 다른 개인일 수 있다.

[0358] 이 예에서, 문 잠금 액세서리(2004)는 하나 이상의 사용자-동작(user-operated) 제어기(2012)(3개로 도시되어 있지만, 임의의 개수가 사용될 수 있음)와 무선으로 통신하도록 구성될 수 있다. 예를 들어, 문 잠금 액세서리(2004)는, 제어기들(2012) 중 하나에 대한 물리적 근접에 의해 트리거되어 통신을 개시할 수 있는, 센서(2020)를 제공할 수 있다. 문 잠금 액세서리(2004)는 또한 마스터 제어기(관리자로도 지칭됨)(2018)와 통신할 수 있다. 이 예에서, 마스터 제어기(2018)와의 통신은 무선 접속에 의한 것이지만, 유선 접속이 또한 사용될 수 있다.

[0359] 마스터 제어기(2018)는, 데스크톱 컴퓨터, 랩톱 컴퓨터, 휴대 전화, 태블릿 등을 포함하는, 문(2006)의 소유자에 의해 소유되거나 동작되는 컴퓨팅 디바이스일 수 있다. 사무실 건물의 경우, 마스터 제어기(2018)를 동작시키는 사람은 지정된 보안 요원일 수 있다. 마스터 제어기(2018)는 물리적으로 문(2006)에 대해 어디에나 위치될 수 있다. 예를 들어, 마스터 제어기(2018)는 문(2006)이 액세스를 제공하는 방 안, 건물 내의 다른 곳에 위치된 경비실 내, 또는 다른 건물 내 모두에 있을 수 있다. 일부 실시예들에서, 마스터 제어기(2018)는 또한 사용자 디바이스(예를 들어, 휴대 전화)로서 기능을 할 수 있다.

[0360] 문 잠금 악세서리(2004)의 설치에 이어서, 마스터 제어기(2018)는 마스터 제어기로서 자신을 확립하기 위해 전술한 바와 같은 페어 셋업 프로세스를 수행할 수 있으며; 예를 들어, 마스터 제어기(2018)는 페어 셋업을 수행한 결과로서 (예를 들어, 전술한 바와 같은) 관리자 허가를 획득할 수 있다. 그 후, 위임된 페어링 기술들(예를 들어, 전술한 바와 같은 페어 추가)이 문 잠금 액세서리(2004)와 각각의 사용자 디바이스들(2012) 사이에 페어링을 확립하는 데 사용될 수 있다. 대안적으로, 문 잠금 액세서리(2004)는 특정한 물리적 조건들(예를 들어, 액세서리(2004) 내에 물리적으로 삽입된 키) 하에서만 페어 셋업이 수행될 수 있도록 구성될 수 있고, 문 잠금 액세서리(2004)는 페어링을 위한 물리적 조건들이 획득되는 시간에 사용자 디바이스들(2012) 각각과 페어 셋업을 수행할 수 있다.

[0361] 일부 실시예들에서, 문 잠금 액세서리(2004)를 위한 액세서리 모델은, 문을 잠금 및 잠금해제하는 능력을 제공하는 잠금 메커니즘 서비스(예를 들어, 도 2g의 잠금 메커니즘 서비스(254)의 인스턴스)를 포함할 수 있다. 문 잠금 액세서리를 위한 액세서리 모델(2004)은 또한 잠금 관리 서비스(예를 들어, 도 2h의 잠금 관리 서비스(255)의 인스턴스)를 포함할 수 있고, 잠금 관리 서비스는 누가 문을 잠그거나 잠금해제했는지 그리고 언제인지를 나타내는 엔트리들을 갖는 잠금 로그를 유지하는 것과 같은, 다른 잠금-관련 기능들을 지원할 수 있다. 잠금 로그는 예를 들어, 마스터 제어기(2018)에 의해 액세스될 수 있다.

[0362] 도 21은 본 발명의 일 실시예에 따른 문 잠금 액세서리(2004)에 대한 액세서리 객체(2100)의 예를 도시한다. 객체(2100)는 표 형식으로 도시되지만, 도 3a 내지 도 3c와 유사한 JSON 객체로도 표현될 수 있다. 도시된 바와 같이, 문 잠금 액세서리(2004)는 액세서리 식별 서비스 인스턴스(2102), 잠금 메커니즘 서비스 인스턴스(2104), 및 잠금 관리 서비스 인스턴스(2106)를 포함할 수 있다. 각각의 서비스 인스턴스는 도시된 바와 같은 특성들을 포함할 수 있으며; 특성들은 도 2a 내지 도 2d 및 도 2j에 따라 정의될 수 있다. 예를 들어, 현재 상태 특성(2110)은 문이 현재 잠금 또는 잠금해제되어 있는지(또는 걸려 있는지(jammed) 등) 여부를 결정하기 위해 제어기에 의해 판독될 수 있고, 목표 상태 특성(2112)은 문의 잠금 또는 잠금해제를 요청하기 위해 제어기에 의해 기록될 수 있다.

[0363] 잠금 로그 특성(2116)은, 각각이 데이터 객체일 수 있는, 잠금-로그 이벤트 레코드들의 배열을 포함할 수 있다. 이 예에서, 각각의 잠금 로그 이벤트 레코드는 문 잠금장치(2004)에 액세스한 엔티티(사람 또는 디바이스)의 식별자, 액세스가 발생한 시간, 및 수행되는 동작(예를 들어, 잠금 또는 잠금해제, 잠금 로그를 판독, 잠금 로그를 소거)을 포함할 수 있다. 선택적인 스트링 요소가 액세스에 관한 추가적인 벤더-특정 정보를 제공하기 위해 제공될 수 있다. 일부 실시예들에서, 제어기는 잠금 로그에 액세스하기 위해 특성(2116)을 판독할 수 있다.

[0364] 잠금 관리 제어 포인트 특성(2114)은, 예를 들어, 잠금 로그를 판독 또는 소거하려는 요청들을 송신하는 데 사용될 수 있다. 예를 들어, 제어기는 특성(2114)에 데이터 객체를 기록하려는 요청을 송신할 수 있고, 데이터

객체는 특정 요청으로서 해석될 수 있다. 이 예에서, 지원되는 요청들은, 잠금 로그를 판독하는 것(데이터 객체 내에 명시된 시작 시간에 시작함), 잠금 로그를 소거하는 것(이는 데이터 객체 내에 명시된 바와 같이, 잠금 로그로부터 모든 엔트리들 또는 엔트리들의 명시된 범위를 삭제할 수 있음), 및 문 잠금 액세서리(2004)가 미래의 잠금 로그 엔트리들을 레코딩하기 위한 기준으로 사용할 수 있는 현재 시간을 설정하는 것(시간을 설정하는 것은, 예를 들어, 일광 절약 시간 등을 고려하는 데 유용할 수 있음)을 포함할 수 있다. 도 5g 내지 도 5k를 참조하여 전술한 바와 같이, 일부 실시예들에서, 문 잠금 액세서리(2004)는 인라인으로 또는 질의 결과 메커니즘을 통해 제어 포인트 특성(2114)에 대한 기록 요청들에 응답할지 여부를 선택할 수 있다. 결정은 요청당 기준일 수 있다.

[0365] 도 21에 도시된 다른 특성들은 도 2a 내지 도 2d 및 도 2j를 참조하여 전술한 바와 같이 동작할 수 있다.

[0366] 도 21에 도시된 서비스들 및 특성들은 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 서비스들 및 특성들의 다른 세트들 및 서브셋들은, 잠금장치의 기능성에 의존하여, 잠금 액세서리에 대해 정의될 수 있다.

[0367] 추가로 예시하기 위해, 특정 구현 시나리오가 이제 기술될 것이다. 이 시나리오에서, 문(2006)의 소유자는 문 잠금 액세서리(2004)를 구입하고 문(2006)에 설치할 수 있다. (문 잠금 액세서리(2004)는 문(2006)의 일부로서 또는 부품시장 업그레이드로서 판매될 수 있다.) 문(2006)의 소유자는 문 잠금 액세서리(2004)와 통신하도록 마스터 제어기(2018)를 구성할 수 있다. 예를 들어, 마스터 제어기(2018)는, 전술한 균일한 액세서리 프로토콜에 부합하는 메시지들을 송신 및 수신하고, 문(2006)의 소유자가 문 잠금 액세서리(2004)와 상호작용할 수 있게 하는 그래픽 사용자 인터페이스(또는 다른 유형의 사용자 인터페이스)를 생성하도록, 프로그램(예를 들어, 운영체제 프로그램 또는 사용자-설치 애플리케이션 프로그램)을 실행할 수 있는 데스크톱 또는 휴대용(예를 들어, 랩톱, 핸드헬드, 모바일, 착용가능) 컴퓨터 시스템일 수 있다.

[0368] 이 설치를 수행했으면, 문(2006)의 소유자는 마스터 제어기(2018)를 문 잠금 액세서리(2004)와 페어링할 수 있다. 도 22는 본 발명의 일 실시예에 따른, 문(2006)의 소유자와 같은 사용자가 마스터 제어기(2018)를 문 잠금 액세서리(2004)와(또는 임의의 다른 제어기를 임의의 다른 액세서리와) 페어링할 수 있는 프로세스의 흐름도이다.

[0369] 블록(2202)에서, 문 잠금 액세서리(2004)(또는 임의의 다른 액세서리)는 페어링 모드로 진입할 수 있다. 일부 실시예들에서, 사용자는 액세서리가 페어링 모드로 진입하게 할 수 있다. 일부 실시예들에서, 각각의 액세서리 제조사는 그것의 액세서리들이 페어링 모드로 진입하게 하는 특정 사용자 행동을 정의할 수 있다. 예를 들어, 문 잠금 액세서리(2004)의 경우에, 액세서리 제조사는, 액세서리 제조사에 의해 제공된 키를 사용자가 삽입하는, 액세서리 하우징 내의 어딘가에 물리적 키홀 또는 키 슬롯을 제공할 수 있다. 또 다른 예로서, 문 잠금 액세서리(2004)는 페어링 모드를 인에이블 또는 디스에이블하기 위해 그것의 하우징 상에 또는 그 내부에 기계적 스위치를 가질 수 있으며; 이 스위치는 문(2006)이 닫힐 때 그것이 액세스가능하지 않도록 배치될 수 있다. 일반적으로, 액세서리를 페어링 모드로 진입하게 하는 것은, 인가된 사용자가 존재하고 제어기를 액세서리와 페어링하려 하고 있다는 것을 액세서리에 나타내기 위한 다양한 행동들을 수반할 수 있지만; 액세서리가 페어링 모드로 진입하게 하기 위한 임의의 특정 기술의 사용이 요구되는 것은 아니다. 또한, 일부 실시예들에서, 문 잠금 액세서리(2004)는, 그것이 먼저 설치되고 전원이 투입될 때, 또는 그것이 임의의 제어기와 확립된 페어링이 없을 때마다, 자동으로 페어링 모드로 진입할 수 있다.

[0370] 블록(2204)에서, 제어기는 액세서리를 발견할 수 있다. 예를 들어, 문 잠금 액세서리(2004) 및 마스터 제어기(2018)의 경우에, 액세서리(2004)는, 그것이 페어링 모드에 배치될 때, 예를 들어, 전술한 바와 같은 디바이스 검색 서비스를 사용하여 페어링에 대한 자신의 가용성을 광고하기 시작할 수 있다. 마스터 제어기(2018)는 예를 들어, 도 4의 액세서리 검색 프로세스(400)를 블록(422)까지 수행함으로써 문 잠금 액세서리(2004)의 위치를 찾을 수 있는, 균일한 제어기 프로그램(또는 다른 프로그램 코드)을 실행할 수 있다. 다른 예로서, 일부 제어기들은 (규칙적인 간격으로, 또는 영역에 들어가는 것과 같은 이벤트들에 응답하여) 자동으로 액세서리들을 검색하도록, 또는 제어기에 의해 제공된 "FIND" 제어부를 작동시키는 것과 같은 사용자 입력에 응답하여 액세서리들을 검색하도록 구성될 수 있다.

[0371] 블록(2206)에서, 사용자는 블록(2204)에서 발견된 액세서리와 페어링하도록 제어기에 지시할 수 있다. 예를 들어, 전술한 바와 같이, 액세서리 검색 프로세스(400)는, 제어기가 액세서리에 관한 정보를 사용자에게 제시하고 페어링이 발생해야 하는지 여부를 나타내도록 사용자에게 프롬프트하는 것을 포함할 수 있다. 다른 예로서, 액세서리 검색을 수행하는 제어기는 발견된 모든 액세서리들의 목록을 사용자에게 제시할 수 있고, 사용자는 페어

링될 액세서리를 목록으로부터 선택할 수 있다. 사용자 명령은 다양한 형태를 취할 수 있다. 일부 경우들에서, 사용자 명령은 문 잠금 액세서리(2004)의 셋업 코드를 입력하는 것을 포함할 수 있고; 다른 경우들에서, 셋업 코드의 입력은 별개의 사용자 행동일 수 있다.

[0372] 블록(2208)에서, 사용자 명령에 응답하여, 제어기는 블록(2204)에서 발견된 액세서리와 페어 셋업 프로세스를 개시할 수 있다. 예를 들어, 전술한 페어 셋업 프로세스들(프로세스들(1300, 1400, 1500, 1600)) 중 임의의 것이 블록(2208)에서 개시될 수 있다.

[0373] 블록(2210)에서, 페어 셋업 프로세스 동안의 어떤 시점에, 사용자는, 페어 셋업을 개시한 특정 제어기와의 페어링이 발생해야 한다는 검증을 액세서리에 제공할 수 있다. 검증이 요구되는지 여부 및 어떤 검증이 요구되는지는, 액세서리 제조사에 의해 결정될 수 있다. 일부 실시예들에서, 액세서리는 그것이 페어링 모드에 있고 제어기가 페어링하려고 시도하고 있다는 사실 외에는 어떠한 추가 검증도 요구하지 않는다. 또한, 일부 실시예들에서, 블록(2206)에서의 사용자 입력(이는 액세서리 셋업 코드를 포함할 수 있음)은 또한 블록(2210)에서의 검증으로서 기능할 수 있다. 액세서리가 검증을 요구하는 경우, 이러한 검증은, 예를 들어, 사용자가 액세서리에 의해 검출가능한 어떤 행동을 수행해야 한다는 요건을 페어 셋업 프로세스에 포함시킴으로써 구현될 수 있으며, 이때 그 행동이 정확히 수행되지 않으면 액세서리는 제어기에 대해 오류 응답을 생성한다.

[0374] 예를 들어, 페어링 프로세스들(1300, 1500)은, 제어기가 액세서리의 셋업 코드를 갖고 있다는 제어기의 증명 형태로 된, 액세서리에 대한 검증을 포함한다. 전술한 바와 같이, 제어기는 사용자로부터 액세서리의 셋업 코드를 획득하고 공유 비밀을 생성하는 데 셋업 코드를 사용할 수 있으며; 제어기가 공유 비밀을 정확히 생성했다는 사실은, 제어기가 셋업 코드를 갖고 있다는 액세서리에 대한 검증일 수 있다. 다른 예로서, 제어기와 액세서리 둘 다가 근거리 통신(Near-Field Communication)(NFC) 기술(예를 들어, NFC 포럼(http://nfc-forum.org)에 의해 공표된 표준들에 부합하는 통신 기술)을 구비하는 경우, 액세서리는, 사용자가 NFC 통신 범위 내로 제어기를 가져올 것을 요구할 수 있고, NFC 채널 상의 제어기가 다른 채널 상에서 페어 셋업을 수행하고 있는 제어기와 동일한 것임을 확인하기 위해 정보를 제어기와 교환할 수 있다. (예를 들어, 제어기는 NFC 채널 및 페어-셋업 채널 둘 다를 통해 자신의 공유 비밀의 증명(proofC)을 제공하도록 요구될 수 있다.) 다른 검증 동작들이 대체될 수 있고, 단일 사용자 행동이 블록(2206)의 페어링하라는 명령 및 블록(2210)의 검증 둘 다를 제공할 수도 있다.

[0375] 블록(2212)에서, 페어 셋업 프로세스는 완료될 수 있고, (오류가 발생하지 않았다고 가정하면) 사용자는 액세서리와의 페어링이 확립되었음을 (예를 들어, 제어기에 의해) 통지받을 수 있다. 그 후, 블록(2214)에서, 액세서리는 페어링 모드를 나갈 수 있다. 일부 실시예들에서, 페어링 모드를 나가는 것은, 페어링 모드로부터 액세서리를 제거하는 사용자 행동을 포함할 수 있다. 이러한 사용자 행동은 물리적 키를 제거하는 것, 페어링 스위치를 그것의 디스에이블된 위치로 플리핑(flipping)하는 것 등과 같은, 액세서리를 페어링 모드로 진입하게 하기 위해 블록(2202)에서 취해진 사용자 행동의 반전일 수 있다. 일부 실시예들에서, 액세서리는 페어 셋업이 완료되면 자동으로 페어링 모드를 나갈 수 있다.

[0376] 도 20을 참조하면, 환경(2000)에서, 프로세스(2200)는, 문(2006)의 소유자가 문 잠금 액세서리(2004)와 페어링하고자 하는 각각의 제어기(2012)에 대해 반복될 수 있다. 그러나, 일부 환경들에서, 각각의 제어기(2012)를 개별적으로 페어링하는 것은 불편할 수 있다. 예를 들어, 페어링하는 것이 가까운 물리적 근접을 요구하는 경우, 각각의 제어기(2012)는 그러한 근접 내에 들어가야 하고 결과적으로 페어링된다. 많은 수의 제어기들(2012)이 페어링되어야 하는 경우(예를 들어, 문(2006)이 많은 승객들이 있는 큰 건물의 현관인 경우), 이것은 매우 시간 소모적인 것이 될 수 있다. 또한, 일부 실시예들에서, 문 잠금 액세서리(2004)는, 제1 제어기가 성공적으로 페어링을 확립한 이후에 제2 제어기가 페어 셋업을 수행하도록 허용하지 않을 수 있다.

[0377] 따라서, 일부 실시예들에서, 프로세스(2200)는 문 잠금장치(2004)와 하나의 제어기(예를 들어, 마스터 제어기(2018)) 사이에서 페어링을 확립하는 데 사용될 수 있다. 그 후, 마스터 제어기(2018)는, 문 잠금장치(2004)와 추가 제어기들(2012)의 페어링을 확립하기 위해, 위임된 페어링 프로세스들(예를 들어, 전술한 페어 추가 프로세스(1800) 또는 다른 위임된 페어링 프로세스들)을 사용할 수 있다. 예를 들어, 마스터 제어기(2018)가 특정 제어기(2012)에 대한 장기간 공개 키(또는 보안 인증서)를 갖는 경우, 마스터 제어기(2018)는 페어 추가 프로세스(1800)를 사용해 키(또는 인증서)를 문 잠금 액세서리(2004)에 제공할 수 있다. 다른 예로서, 마스터 제어기(2018)는 (전술한 바와 같이) 신뢰 인증서 체인을 문 잠금 액세서리(2004)에 제공할 수 있고, 각각의 제어기(2012)는 신뢰 인증서 체인에 의해 서명된 인증서를 가질 수 있으며, 이는 주어진 제어기(2012)가 페어링을 확립하고 문 잠금 액세서리(2004)에 액세스하도록 하는 데 사용될 수 있다. 전술한 바와 같이, 일부 실시예들에

서, 마스터 제어기(2018)는 선택적으로 임의의 추가된 제어기(2012)에 관리자 허가를 승인할 수 있고, 관리자 허가를 갖는 제어기(2012)는 액세서리(2004)와 추가 제어기들(2012) 사이에 페어링을 확립하기 위해 페어 추가를 수행할 수 있다.

[0378] 또한, 문(2006)에 액세스하도록 인가된 사용자들의 세트는 시간이 지남에 따라 변할 수 있다. 예를 들어, 사무실 건물에서, 직원이 직장을 그만둘 수 있고, 그 때문에 건물에 대한 그녀의 액세스는 종료되어야 한다. 집에서, 룸메이트가 이사를 나갈 수 있다. 마스터 제어기(2018)(또는 관리자 허가를 갖는 다른 제어기)는 예를 들어, 전술한 페어 제거 프로세스(1900)를 사용하여, 더 이상 액세스가 승인되지 않아야 하는 임의의 제어기(2012)와 문 잠금 액세서리(2004)의 페어링을 제거함으로써, 그러한 업데이트들을 수행할 수 있다.

[0379] 일부 실시예들에서, 마스터 제어기(2018)(또는 문 잠금 액세서리(2004)에 대한 관리자 허가를 갖는 다른 제어기(2012)) 상에서 실행되는 균일한 제어기 프로그램은, 문(2006)에 대한 액세스를 관리하는 것을 용이하게 하기 위해 다양한 사용자 인터페이스를 제공할 수 있다. 예를 들어, 소유자 또는 보안 요원은, 예를 들어, 전술한 바와 같은 페어링 목록 요청을 사용해 인가된 제어기들의 목록을 볼 수 있고(일부 실시예들에서, 목록은 또한 인가된 제어기들의 사용자들을 식별하는 것을 포함할 수 있음); 인가된 목록에 추가될 새로운 제어기를 식별하고; 그리고/또는 제거되어야 할 인가된 제어기를 인가된 목록으로부터 선택할 수 있다. 따라서, 조직 또는 다중-사용자 환경에서의 보안 동작들은 간소화될 수 있다.

[0380] 특정 제어기(2012)가 (예를 들어, 직접 또는 위임된 페어링을 포함하는, 전술한 기술들 중 임의의 것을 사용하여) 문 잠금 액세서리(2004)와 페어링을 확립했으면, 그 제어기(2012)는 문(2006)에 액세스하는 데 사용될 수 있다. 예를 들어, 제어기(2012)는, 전술한 바와 같은 균일한 액세서리 프로토콜에 부합하는 메시지들을 제어기(2012)가 송신 및 수신할 수 있게 하는, 프로그램 코드(예를 들어, 운영 체제 또는 애플리케이션 코드)를 제공받을 수 있다. 프로그램은 또한, 제어기(2012)의 사용자로 하여금 액세서리와 상호작용할 수 있게 하는, 그래픽 사용자 인터페이스(또는 다른 유형의 사용자 인터페이스)를 정의할 수 있다.

[0381] 도 23은 본 발명의 일 실시예에 따른 문을 잠금해제하기 위한 프로세스(2300)의 흐름도이다. 프로세스(2300)는 예를 들어, 도 20의 임의의 제어기(2012)(또는 제어기(2018))와 문 잠금 액세서리(2004) 사이에서 수행될 수 있다.

[0382] 프로세스(2300)는, 제어기(2012)가 문(2006)이 잠금해제되어야 한다고 결정할 때, 블록(2302)에서 시작할 수 있다. 예를 들어, 휴대용 제어기(2012)를 들고 있는 사용자가 문 잠금 액세서리(2004)의 범위 내에 올 수 있다. "범위 내"는 무선 통신 범위 내 또는 원하는 대로 더 좁게 정의될 수 있다. 예를 들어, 문들의 의도하지 않은 잠금해제를 피하기 위해, 제어기(2012) 및 문 잠금 액세서리(2004)는 근접 감지 능력을 갖도록(예를 들어, 블루투스 LE, NFC 등을 사용해) 구성될 수 있고, 제어기(2012)는 문을 잠금해제하려고 시도하기 위한 최대 범위(예를 들어, 수 인치, 2 피트, 6 피트, 또는 어떤 다른 범위 이내)를 정의하도록 구성될 수 있다. 일부 실시예들에서, 제어기(2012)가 문 잠금 액세서리(2004)가 범위 내에 있음을 검출할 때, 제어기(2012)는 사용자가 문(2006)을 잠금해제하고자 하는지 확인하기 위해 사용자와 상호작용할 수 있다. 예를 들어, 사용자는 범위 내에 올 때 문을 잠금해제 하기 위해 애플리케이션 또는 시스템-레벨 프로그램(또는 그 일부)을 시작하거나 달리 활성화할 수 있다. 다른 실시예들에서, 제어기(2012)는 그것이 확립된 페어링을 갖고 있는 문 잠금 액세서리들에 대해 스캔하는 백그라운드 프로세스로서 프로세스(2300)의 부분들을 실행할 수 있고; 그러한 액세서리를 범위 내에서 검출할 시, 제어기(2012)는 사용자에게 프롬프트를 생성할 수 있다. 또한, 제어기(2012)가 문이 잠금해제되어야 함을 확인하기 위해 사용자에게 프롬프트할 때, 제어기(2012)는, 제어기(2012)가 인가된 사용자에 의해 휴대되고 있음을 검증하기 위해, 사용자에게 인증 크리덴셜(예를 들어, 패스워드 또는 지문과 같은 생체측정 크리덴셜)을 공급할 것을 요구할 수 있다. 또 다른 실시예들에서, 제어기(2012)는, 제어기(2012)가 문 잠금 액세서리(2004)의 범위 내에 올 때마다 사용자 입력 없이 문(2006)을 잠금해제하려고 자동으로 시도하도록 구성될 수 있다. 다른 구현예들이 또한 가능하다. 특정 구현예는 특정 문 잠금 액세서리(2004)에 대해 요구되는 보안 수준에 의존할 수 있어서, 사용자가 어떤 문에 접근했는지에 따라 동일한 제어기가 상이하게 동작할 수 있게 된다.

[0383] 또한, 문이 잠금해제되어야 함을 결정하기 위해 다른 기술들이 사용될 수 있으며, 제어기(2012)와 문 잠금 액세서리(2004) 사이의 물리적 근접은 필수적인 것은 아니다. 예를 들어, 사용자는, 문을 선택하고 그것이 잠금해제되어야 함을 나타내기 위해 제어기(2012) 상의 사용자 인터페이스를 동작시킴으로써, 원격으로(예를 들어, 다른 방 또는 완전히 다른 위치에서) 문(2006)을 잠금해제할 수 있다. 일부 실시예들에서, 원격 잠금해제를 수행하기 이전에, 제어기(2012)는, 제어기(2012)가 인가된 사용자에 의해 동작되고 있음을 검증하기 위해, 사용자에

게 인증 크리덴셜(예를 들어, 패스워드 또는 지문과 같은 생체측정 크리덴셜)을 공급할 것을 요구할 수 있다.

[0384] 블록들(2306, 2308)에서, 제어기(2012) 및 문 잠금 액세서리(2004)는 페어 검증 동작(예를 들어, 전술한 프로세스(1700))을 수행할 수 있다. 이 동작은 두 디바이스 사이에서 페어링이 이전에 확립되었다는 것을 검증할 수 있다. 블록(2310)에서, 페어 검증 프로세스가 성공하지 않은 경우, 프로세스(2300)은 블록(2312)에서 종료할 수 있고, 제어기(2012)는 사용자에게 오류를 경고할 수 있다. 일부 실시예들에서, 사용자는 재시도하도록 프롬프트될 수 있다. 페어 검증은 제어기(2012)가 문 잠금 액세서리(2004)를 잠금해제하려고 시도할 때마다 요구될 수 있다.

[0385] 페어 검증 프로세스가 성공한 경우, 이어서 블록(2314)에서, 제어기(2012)는 문을 열려는 암호화된 요청을 문 잠금 액세서리(2004)에 송신할 수 있다. 예를 들어, 도 5a 내지 도 5d를 참조하여 전술한 바와 같이, 제어기(2012)는 문 잠금 액세서리(2004)의 잠금 상태 특성(2102)에 부울 "true"의 값(open)을 기록하기 위한 HTTP PUT 요청을 송신할 수 있다. 페어-검증된 세션 내의 모든 통신들과 마찬가지로, 요청은 페어 검증 동안에(또는 그 후에) 확립된 세션 키를 사용해 암호화될 수 있다. 일부 실시예들에서, 요청은 제어기(2012)의 장기간 비밀 키로 서명될 수 있지만, 이것은 요구되는 것은 아니다.

[0386] 블록(2316)에서, 문 잠금 액세서리(2004)는 암호화된 요청을 수신할 수 있다. 블록(2318)에서, 문 잠금 액세서리(2004)는 요청을 검증 및 복호화할 수 있다. 페어-검증된 세션 내의 모든 통신들과 마찬가지로, 액세서리는 요청이 정확한 세션 키로 암호화되지 않은 경우 요청을 무시할 수 있다. 또한, 요청이 제어기의 장기간 비밀 키로 서명된 경우, 액세서리는 서명을, 그리고 이에 따라 제어기의 아이덴티티를 검증하기 위해, 제어기의 장기간 공개 키의 사본(이것은 확립된 페어링을 위해 지속적으로 저장될 수 있음)을 사용할 수 있다. 액세서리(2004)는 또한, 제어기(2012)가 문을 잠금해제하는 것이 허용된다는 것을 검증할 수 있다. 예를 들어, 일부 실시예들에서, 특정 시간들에서의 액세스는 관리자 허가를 갖는 제어기들로 제한될 수 있다. 이 제한은, 도 2d의 특성(228)의 인스턴스일 수 있는, 관리자-전용 액세스 특성(2122)(도 21)에 기록하는 관리자 특권을 갖는 제어기에 의해 설정 또는 제거될 수 있다. 다른 규칙들 또는 결정 기준들이 또한 적용될 수 있다.

[0387] 블록(2320)에서, 요청이 유효한 경우(예를 들어, 그것이 정확히 복호화되었고 허가된 제어기로부터의 것인 경우), 문 잠금 액세서리(2004)는 블록(2322)에서 문을 잠금해제하는 것으로 진행할 수 있다. 일부 실시예들에서, 문 잠금 액세서리(2004)는 (예를 들어, HTTP 200 OK 응답을 송신함으로써) 제어기(2012)에 잠금해제를 보고할 수 있다. 원하는 경우, 제어기(2012) 및/또는 문 잠금 액세서리(2004)는 또한 성공을 나타내는 사용자-감지가능한 출력을 제공할 수 있다. 예를 들어, 디바이스들 중 어느 하나 또는 둘 모두는 삐 소리를 내거나 클릭 소리를 내거나, 녹색 광을 깜박이는 것 등을 할 수 있다. 일부 실시예들에서, 액세서리의 피드백 거동은 오디오 피드백 특성(206)(도 2a) 또는 원하는 바에 따른 다른 특성들과 같은 적절한 특성들을, 잠금 서비스 인스턴스(2106)의 정의 내에 포함함로써 맞춤화될 수 있다.

[0388] 블록(2302)에서, 요청이 유효하지 않은 경우, 이어서 블록(2324)에서, 문 잠금 액세서리(2004)는 (예를 들어, HTTP 오류 응답을 송신함으로써) 제어기(2012)에 오류 메시지를 송신할 수 있다. 일부 실시예들에서, 제어기(2012)는 사용자에게 오류를 알리고/알리거나, 사용자에게 다시 시도하도록 프롬프트할 수 있다. 원하는 경우, 제어기(2012) 및/또는 문 잠금 액세서리(2004)는 또한 잠금해제 시도가 실패했음을 나타내는 사용자-감지가능한 출력을 제공할 수 있다. 예를 들어, 디바이스들 중 어느 하나 또는 둘 모두는 오류 사운드를 내거나(예를 들어, 성공 사운드와 상이한 삐 소리), 적색 광을 깜박거리는 것 등을 수행할 수 있다.

[0389] 프로세스(2300)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 프로세스(2300)는 이전에 확립된 페어링의 존재에 의존하지만, 페어링이 확립된 방식(예를 들어, 인증서-기반 또는 셋업-코드-기반, 직접 또는 위임)에는 독립적일 수 있다는 것에 유의해야 한다. 따라서, 이 예에서의 액세서리 및 제어기는, 그것들이 확립된 페어링을 갖는다는 것은 알아야 하지만 페어링이 어떻게 또는 언제 확립되었는지는 알 필요가 없을 것이다.

[0390] 또한, 프로세스(2300)는 문을 잠그는 특정 콘텍스트에서의 제어기와 액세서리 사이의 상호작용을 예시하지만, 본 개시내용에 대한 액세스를 갖는 통상의 기술자는 유사한 프로세스들이 임의의 액세서리의 임의의 동작을 제어하도록 구현될 수 있다는 것을 인식할 것이다.

[0391] 일부 실시예들에서, 문 잠금 액세서리(2004)는 (예를 들어, 프로세스(2300)와 유사한 프로세스를 통해) 문을 잠그기 위한 익스프레스 명령어(express instruction)가 수신될 때까지 문(2006)을 잠금해제된 상태로 둘 수

있다. 다른 실시예들에서, 문 잠금 액세서리(2004)는 자동 재잠금 구간을 적용할 수 있다. 예를 들어, 블록(2322)에서 문(2006)을 잠금해제한 이후에, 액세서리(2004)는 고정된 시간 기간(예를 들어, 10초)을 기다린 다음에 재잠금할 수 있다. 블록(2322)에서 잠금해제한 이후에 사용자가 문(2006)을 신속하게(예를 들어, 5초 이내에) 열지 않는 경우 자동으로 문(2006)을 재잠금하는 것과 같은, 다른 보안 조치들이 또한 구현될 수 있다. 자동 재잠금 거동은 예를 들어, 자동-타임아웃 특성 인스턴스(2124)(도 21)에 0이 아닌 값을 기록함으로써, 구현될 수 있다.

[0392] 일부 실시예들에서, 문 잠금 액세서리(2004)는 문을 잠금해제하기 전에 추가 조건들 또는 사용자 행동들을 요구할 수 있다. 예를 들어, 위에서 언급한 바와 같이, 문 잠금 액세서리(2004)는 NFC 송수신기를 포함할 수 있고, 문 잠금 액세서리(2004)는 제어기(2012)가 잠금해제 이전에 NFC 범위 내로 올 것을 요구할 수 있다. 또는, 문 잠금 액세서리(2004)는 사용자가 문를 터치하고 있을 때를 검출하는 센서들을 가질 수 있고, 액세서리가 잠금해제 요청이 유효하다고 결정한 이후에 문이 신속하게(예를 들어, 2초 또는 5초 이내에) 터치되지 않는 경우 잠금해제할 것을 거부할 수 있다.

[0393] 다른 예로서, 일부 실시예들에서, 문 잠금장치들(또는 임의의 다른 액세서리)의 제조사 또는 벤더는 제어기와 액세서리 사이에서 교환되는 "데이터 블랍"의 사용에 의해 제조사-특정 거동을 포함할 것을 선택할 수 있다. 본 명세서에서 사용되는 바와 같이, "데이터 블랍"은, 제어기에 의해 저장되고(예를 들어, 도 20의 제어기들(2012) 및 마스터 디바이스(2018) 각각은 데이터 블랍을 저장할 수 있음), 액세서리로의 임의의 요청의 일부로서 그 액세서리로 송신되거나 선택적으로 특정 요청들(예를 들어, 특정 특성들에 기록하려는 요청들)과 함께 송신될 수 있는, 데이터의 블록을 지칭할 수 있다. 제어기의 관점에서, 데이터 블랍은 불투명할 수 있고; 데이터 블랍을 수신하는 액세서리는 액세서리-특정 방식으로 그것의 콘텐츠들을 해석할 수 있다.

[0394] 예시로서, 문 잠금 액세서리(2004)의 경우, 제조사는 문 잠금 액세서리(2004)의 사용자들이 잠금장치의 인가된 사용자들로서 등록할 것을 요구할 수 있다. 예를 들어, 제조사는, 사용자가 (예를 들어, 서버에서 계정을 생성함으로써) 잠금장치에 대한 인가 코드(authorization code)를 획득할 수 있는 웹사이트를 지원하는 서버를 제공할 수 있다. (인가 코드 및 액세서리 제조사에 의해 요구되는 임의의 추가 정보를 포함하는 데이터 객체일 수 있는) 인가 블록(authorization block)은 서버에 의해 생성되고 사용자의 제어기(2012)에 전달될 수 있다. 제어기(2012)는 액세서리(2004)와 연관된 데이터 블랍으로서 인가 블록을 저장할 수 있다. 제어기(2012)가 이후에 액세서리(2004)에 요청들을 송신할 때, 제어기(2012)는 저장된 데이터 블랍을 요청 내에 포함시킬 수 있다.

[0395] 일부 실시예들에서, 데이터 블랍을 송신하는 것은 요청의 성질에 의존할 수 있다. 예를 들어, 액세서리 정의 레코드는, 제어기들이 그 특성에 관하여 갖는 허가들을 포함하는, (예를 들어, 도 3a 내지 도 3c에 도시된 바와 같은) 특성들을 정의할 수 있다. 일부 실시예들에서, 허가 스트링(예를 들어, "Blob Required")은 데이터 블랍이 특성에 액세스하기 위해 요구된다는 것을 나타내도록 정의될 수 있고, 액세서리 제조사는, 제조사가 데이터 블랍들을 수신하고자 하는 임의의 특성에 대한 허가들의 정의 내에, 이 스트링을 포함시킬 수 있다. 특정 특성에 대한 요청들을 생성할 때, 제어기(2012)는 데이터 블랍이 요구되는지 여부를 결정하기 위해 (그것이 캐싱할 수 있는) 액세서리 정의 레코드로부터의 허가들을 참고할 수 있고; 만약 그렇다면, 제어기(2012)는 데이터 블랍을 요청에 추가할 수 있다. 다른 실시예들에서, 데이터 블랍을 송신할지 여부의 결정은, 액세서리당 레벨에서 수행될 수 있다(예를 들어, 제어기가 특정 액세서리에 대해 저장된 데이터 블랍을 갖는 경우, 제어기는 그것이 그 액세서리로 송신하는 모든 요청에 데이터 블랍을 추가할 수 있다).

[0396] 예시적인 액세서리: IP 카메라

[0397] 본 발명의 다양한 실시예에 따라 제어될 수 있는 액세서리의 두 번째 예는 IP 카메라 액세서리이다. IP 카메라 액세서리는 (오디오를 갖거나 그것 없이) 비디오 이미지들을 캡처하고, 캡처된 미디어(오디오 및/또는 비디오)를 다른 디바이스들에 스트리밍할 수 있는 카메라를 포함할 수 있다. 일부 경우들에서, IP 카메라는 또한, 레코딩 및/또는 이전에 레코딩된 내용의 재생과 같은, 다른 기능성을 제공할 수 있다. 본 명세서에서 기술된 바와 같은 균일한 액세서리 프로토콜을 사용하는 임의의 다른 액세서리와 마찬가지로, 이들 기능성은 서비스들로서 모델링될 수 있고, 다양한 서비스의 특성 인스턴스들을 판독하고 그에 기록함으로써 제어될 수 있다.

[0398] 전술한 균일한 액세서리 프로토콜의 실시예들에서, 제어기와 액세서리 사이의 통신은 HTTP 요청들 및 응답들을 사용해 일어날 수 있다. IP 카메라의 경우, HTTP 요청들 및 응답들은 카메라를 셋업하고 그것의 거동(예를 들어, 카메라를 조준하는 것, 레코딩을 시작 및 중지하는 것 등)을 제어하기 위해 사용될 수 있지만; HTTP는 디바이스들 사이의 미디어 콘텐츠의 실시간 스트리밍에는 적합하지 않을 수 있다. 따라서, 본 명세서에서 기술된 IP 카메라 실시예들에서, IP 카메라는, 미디어 보안에 사용되는 SRTP 프로토콜(예를 들어, IETF RFC 3711에서

정의된 바와 같음)과 함께 RTP(예를 들어, IETF RFC 3550에서 정의된 바와 같음)와 같은 상이한 프로토콜을 사용할 수 있다. 다른 미디어 스트리밍 프로토콜들이 대체될 수 있다는 것이, 명백할 것이다. 균일한 액세서리 프로토콜은 스트리밍 프로토콜을 지원하는 스트리밍 세션(이는 균일한 액세서리 프로토콜에 따라 정의된 페어-검증된 세션과 구별될 수 있음)을 확립하는 데 사용가능한 특성들을 정의할 수 있고, 스트리밍된 콘텐츠의 전달은 스트리밍 세션을 통해 일어날 수 있다.

[0399] 도 24를 참조하면, 본 발명의 일 실시예에 따른 IP 카메라 액세서리(2404)를 위한 동작 환경(2400)이 도시된다. 이 예에서, IP 카메라 액세서리(2404)는 예를 들어, 무선 액세스 포인트(2406)를 통해, 제어기(2402)와 통신할 수 있다. 직접 통신도 지원될 수 있다. IP 카메라 액세서리(2404)는 영역 내에 영구적으로 또는 제거가능하게 설치되는 고정기구(예를 들어, 건물의 방, 복도, 또는 입구에 설치된 보안 또는 감시 카메라)일 수 있거나, 또는 그것은 다양한 상이한 설정들로 비디오를 캡처하는 데 사용될 수 있는 휴대용 카메라일 수 있다. 제어기(2402)는 IP 카메라(2404)를 제어할 임의의 디바이스일 수 있다. 예를 들어, IP 카메라(2404)가 건물의 보안 카메라로서 사용되는 경우, 제어기(2402)는 건물의 경비실에서 구현될 수 있다. 또 다른 예로서, IP 카메라(2404)는 아이의 방에서, 예를 들어, 아기 모니터로서 설치될 수 있고, 제어기(2402)는 부모의 휴대 전화, 태블릿, 또는 부모가 대체로 휴대할 다른 디바이스일 수 있다. 또 다른 예로서, 사용자는 공원 또는 야생동물 보호구역에 IP 카메라(2404)를 가지고 가서 눈에 띄지 않는 장소에 그것을 설치하고, 이어서 원격 위치로 물러나 제어기(2402)를 사용해 IP 카메라(2404)를 제어함으로써, 야생동물 등의 고품질 비디오들을 획득할 가능성을 증가시킬 수 있다. 따라서, IP 카메라(2404)의 사용은 어떠한 특정 환경 또는 어떠한 특정 제어기로도 제한되지 않는다.

[0400] 본 명세서에서 기술된 다른 액세서리들처럼, IP 카메라 액세서리(2402)는 서비스들의 집합으로서 모델링될 수 있다. 도 25a 및 도 25b는 본 발명의 일 실시예에 따른 IP 카메라 액세서리(2402)에 대한 액세서리 모델에 포함될 수 있는 서비스들(2501 내지 2505)에 대한 예시적인 정의들을 도시한다. 위의 도 2g 및 도 2h에서 도시된 서비스 정의 예들과 유사하게, 서비스 정의들(2501 내지 2505)은 필수 및/또는 선택적 특성들을 명시할 수 있다. 도 26a 내지 도 26e는 도 25a 및 도 25b의 서비스들의 특성들에 대한 예시적인 정의들을 도시한다. 도 26a 내지 도 26e에서 제공되지 않는 도 25a 및 도 25b에 열거된 특성들의 예시적인 정의들은, 도 2a 내지 도 2d에서 발견될 수 있다. 이들 서비스 및/또는 특성 중 임의의 것은 핵심 서비스들 및 특성들로서 정의될 수 있거나, 또는 그렇게 정의되지 않는다면, 확장 서비스들 및 특성들로서 정의될 수 있다.

[0401] 도 25a를 참조하면, IP 카메라 스트리밍 서비스(2501)는 미디어 세션들을 관리하는 데 사용되는 실시간 통신 서비스를 설명할 수 있다. 예를 들어, IP 카메라 스트리밍 서비스(2501)는 예를 들어, 후술되는 바와 같이, 제어기(2402)와 IP 카메라 액세서리(2404) 사이에서 미디어 스트림의 셋업 및 제어를 용이하게 하는 특성들을 포함할 수 있다.

[0402] 레코딩 서비스(2502)는 레코딩 디바이스를 제어하는 데, 예를 들어, 레코딩을 시작, 중지, 또는 예약하는 데 사용되는 서비스를 설명할 수 있다. 유사하게, 재생 서비스(2503)는 저장된 미디어의 액세서리에 의한 재생을 제어하는 데, 예를 들어, 재생을 시작 및 일시 정지하는 데 사용되는 서비스를 설명할 수 있다. 이 예에서 도시되지 않았지만, 통상의 기술자는, 추가 특성들이 재생할 저장된 미디어 항목을 선택하기 위해, 재생 서비스(2503)의 일부로서 또는 별개의 콘텐츠 선택 서비스로서, 정의될 수 있다는 것을 인식할 것이다.

[0403] 도 25b를 참조하면, 카메라 서비스(2504)는 카메라를 온 또는 오프시키는 것과 같은 카메라 설정들의 제어를 제공할 수 있다. 일부 실시예들에서, 카메라 설정들에 관련된 다른 특성들은, 카메라 배향 특성들(예를 들어, 팬, 틸트, 줌, 회전)과 같이, 선택적 특성들로서 이 서비스에 포함될 수 있다. 다른 선택적 특성들은 야간 시야(온 또는 오프) 및 미러 모드(캡처되는 이미지의 미러링을 인에이블 또는 디스에이블함)와 같은 이미지-처리 특성들을 포함할 수 있다. 특정 카메라 액세서리가 제공하는 임의의 사용자-선택가능한 설정은 카메라 서비스(2504) 내에 정의되고 포함된 대응하는 특성을 가질 수 있다.

[0404] 마이크로폰 서비스(2505)는 소리를 레코딩하도록 동작가능한 마이크로폰의 제어를 제공할 수 있다. 서비스(2505)는 마이크로폰 입력을 갖는 임의의 카메라 액세서리에 대한 정의 내에 포함되고, 그렇지 않은 임의의 카메라 액세서리에 대해서는 생략될 수 있다.

[0405] 스피커 서비스(2506)는 소리를 출력하도록 동작가능한 스피커에 대한 제어를 제공할 수 있다. 서비스(2506)는 스피커 출력을 갖는 임의의 카메라 액세서리에 대한 정의 내에 포함되고, 그렇지 않은 임의의 카메라 액세서리에 대해서는 생략될 수 있다.

[0406] 도 26a 및 도 26b를 참조하면, IP 카메라 관리에 유용한 특성들은 세션 시작(session start) 특성(2601), 세션 종료(session end) 특성(2602), 및 추가 특성들(2603 내지 2615)을 포함할 수 있다. 세션 시작 특성(2601) 및 세션 종료 특성(2602)(도 26a에 도시됨)은 스트리밍 세션을 시작 및 종료하기 위해 제어기에 의해 기록될 수 있다. 추가 특성들(2603 내지 2615)은 IP 카메라의 비디오 및/또는 오디오 스트리밍의 속성들에 관한 정보를 획득하기 위해 제어기에 의해 판독될 수 있다.

[0407] 세션 시작 특성(2601)은, IP 카메라 액세서리에 의해 스트리밍 세션을 시작하는 데 사용가능한 정보를 제공하기 위해 제어기가 기록할 수 있는 (예를 들어, TLV 포맷 또는 다른 키-값 포맷의) 데이터 객체를, 값으로 가질 수 있다. 도 26c는 세션-시작 특성(2601)에 포함될 수 있는 데이터 요소들의 예들을 도시한다. 세션 ID(2631)는 시작되어야 할 스트리밍 세션에 대한 UUID일 수 있다. 제어기 IP 주소(2632) 및 제어기 포트(2633)는 (예를 들어, IPv4 또는 IPv6을 사용하여) 스트리밍된 데이터가 송신되어야 하는 목적지를 식별할 수 있다. 제어기 SRTP 마스터 키(2634) 및 제어기 SRTP 마스터 솔트(2635)는 세션 내에서 스트리밍되는 미디어를 암호화하는 데 사용될 솔트 및 마스터 암호화 키를 제공할 수 있다. 비디오 최대 대역폭(2636) 및 오디오 최대 대역폭(2637)은 (예를 들어, 초당 킬로비트(kbps) 단위로) 스트리밍 데이터 대역폭에 대한 상한들을 나타낼 수 있다.

[0408] 도 26a를 다시 참조하면, 세션 종료 특성(2602)은, 종료될 세션의 세션 식별자를 제공하는 (예를 들어, TLV 포맷 또는 다른 키-값 포맷의) 데이터 객체를 값으로 가질 수 있다. 일부 실시예들에서, 다른 정보는 세션을 종료하는 데 필요하지 않다.

[0409] 비디오 코덱 이름(2603)은 IP 카메라 서비스 인스턴스의 비디오 코덱에 의해 제공되는 미디어 타입을 표현하는 스트링을 제공할 수 있다. 일부 실시예들에서, 스트링에 대한 유효 값들의 세트(예를 들어, http://www.iana.org를 통해 액세스가능한, IANA(Internet Assigned Numbers Authority, 인터넷 할당 번호 관리 기관)에 의해 정의된 코덱 이름들의 세트))가 정의될 수 있다. 일부 실시예들에서, IP 카메라 서비스(2501)의 주어진 인스턴스는 하나의 코덱(예를 들어, IETF RFC 6184에 정의된 바와 같은 H.264 코덱)을 지원하고, 다수의 코덱을 지원하는 IP 카메라 액세서리는 IP 카메라 서비스(2501)의 다수의 인스턴스를 정의할 수 있다. 제어기는 IP 카메라 서비스(2501)의 대응하는 인스턴스의 세션 시작 특성에 기록함으로써 세션에 대한 원하는 코덱을 선택할 수 있다.

[0410] 비디오 코덱 파라미터(2604)는 비디오 코덱에 대한 추가 파라미터들을 제공할 수 있다. 포함되는 특정 파라미터들은 비디오 코덱 이름(2603)에 의존할 수 있고, 키-값 포맷으로 표현될 수 있다. 예를 들어, H.264 코덱의 경우, 비디오 코덱 파라미터들(2604)은 H.264 코덱의 서브-프로파일 및 레벨을 명시하는 프로파일-레벨 ID, 및 코덱이 단일 NAL 단위 모드를 지원하는지 인터리빙되지 않은(non-interleaved) 모드를 지원하는지 여부를 명시하는 패킷화 모드를 포함할 수 있다. 다른 파라미터들은, 서비스 인스턴스에 의해 지원되는 특정 비디오 코덱에 따라 정의될 수 있다.

[0411] 비디오 속성 특성(2605)은 서비스-레벨 속성들(예를 들어, SDP 속성들)을 제공할 수 있다. 예들은, 이미지 속성들 및 방향 속성들(예를 들어, 송신 전용, 수신 전용, 또는 양방향(송신 및 수신))을 포함한다. 일부 실시예들에서, 방향이 명시되지 않는 경우, "송신 전용"은 디폴트로 추정될 수 있다.

[0412] RTP 비디오 페이로드 타입 특성(2606)은 예를 들어, RTP에 대해 명시되는 바와 같은, 7-비트 정수 페이로드 타입을 제공할 수 있다.

[0413] RTP 프로토콜 특성(2607)은 사용 중인 특정 RTP-기반 프로파일을 식별하는 스트링을 제공할 수 있다. 예를 들어, "RTP/SAVP"는 IETF RFC 3550 내에 정의된 프로파일을 지칭할 수 있는 한편, "RTP/SAVPF"는 IETF RFC 5104 내에 정의된 프로파일을 지칭할 수 있다. 다른 프로파일들 및 스트링들이 또한 정의될 수 있다.

[0414] RTP 확장(extensions) 특성(2608)은 IP 카메라 서비스(2501)의 이 인스턴스에 의해 지원되는 RTP 확장들을 열거하는 스트링들의 배열을 제공할 수 있다. RTP 확장들의 예들은, 픽처 손실(picture loss) 표시, 시간-공간 트레이드오프 요청들, 일시적 최대 미디어 스트리밍 비트 레이트(temporary maximum media streaming bit rate) 등을 포함할 수 있다.

[0415] SRTP 암호 스위트(SRTP crypto suite) 특성(2609)은 안전한 RTP 스트리밍에 사용될 암호 스위트를 식별하는 스트링을 제공할 수 있다. 일부 실시예들에서, 스트링은 암호 스위트의 IANA-등록 이름에 부합할 수 있다. 일부 실시예들에서, SRTP 암호 스위트 특성(2609)은, "none"의 값이 SRTP가 사용되지 않음(예를 들어, 스트리밍된 비디오 데이터가 암호화되지 않음)을 나타내게 할 수 있다.

[0416] 도 26b를 참조하면, 오디오 코덱 이름 특성(2610)은 오디오 코덱에 의해 제공된 미디어 타입을 표현하는 스트링을 제공할 수 있다. 일부 실시예들에서, 스트링에 대한 유효 값들의 세트(예를 들어, IANA(인터넷 할당 번호 관리 기관)에 의해 정의된 코덱 이름들의 세트))가 정의될 수 있다. 일부 실시예들에서, IP 카메라 서비스(2501)의 주어진 인스턴스는 오디오 및 비디오 코덱의 하나의 조합을 지원할 수 있고, IP 카메라 액세서리는 IP 카메라 서비스(2501)의 다수의 인스턴스를 정의함으로써 코덱들의 다수의 조합을 지원할 수 있다. 제어기는 IP 카메라 서비스(2501)의 대응하는 인스턴스의 세션 시작 특성에 기록함으로써 세션에 대한 원하는 오디오 및 비디오 코덱을 선택할 수 있다.

[0417] 오디오 코덱 파라미터들(2611)은 오디오 코덱에 대한 추가 파라미터들을 제공할 수 있다. 포함되는 특정 파라미터들은 오디오 코덱 이름(2610)에 의존할 수 있고, 키-값 포맷으로 표현될 수 있다. 예를 들면, Opus 코덱의 경우에, 오디오 코덱 파라미터들(2604)은, 일정한 비트 레이트가 인에이블되어 있는지 또는 가변 비트 레이트가 인에이블되어 있는지에 관한 표시자를 포함할 수 있다. 다른 파라미터들이, 특정 오디오 코덱에 따라 정의될 수 있다.

[0418] 오디오 속성(audio attributes) 파라미터(2613)는 미디어-레벨 속성들(예를 들어, SDP 속성들)을 제공할 수 있다. 예들은, 방향 속성들(예를 들어, 송신 전용, 수신 전용, 또는 양방향(송신 및 수신))을 포함한다. 일부 실시예들에서, 방향이 명시되지 않는 경우, "송신 전용"은 디폴트로 추정될 수 있다.

[0419] RTP 오디오 클록 속도(audio clock rate) 특성(2614)은 예를 들어, RTP에 대해 명시된 바와 같은, 오디오에 대한 RTP 클록 속도를 제공할 수 있다.

[0420] RTP 오디오 페이로드 타입(audio payload type) 특성(2615)은 예를 들어, RTP에 대해 명시된 바와 같은, 7-비트 정수 페이로드 시간을 제공할 수 있다.

[0421] 다양한 실시예에서, 다른 특성들 및 서비스들이 또한 스트리밍 미디어에 대해 정의될 수 있다. 일부 실시예들에서, 모든 특성들이 모든 서비스 인스턴스에서 사용되는 것은 아니다. 예를 들어, 오디오를 수신하거나 스트리밍하지 않는 IP 카메라 액세서리에 대한 액세서리 모델은 특성들(2610 내지 2615)을 생략할 수 있다.

[0422] 도 26d는 본 발명의 일 실시예에 따른 카메라 서비스(2503)의 다양한 특성들(2651 내지 2656)에 대한 예시적인 정의들을 도시한다. 이들 특성(선택적일 수 있음)은 카메라 서비스가 대응하는 거동을 지원하는 경우 정의될 수 있다. 제어기는 카메라 배향 및 이미징 거동을 제어하기 위해 이들 특성에 기록할 수 있다. 예를 들어, 야간 시야(night vision) 특성(2651)은 야간-시야 이미징 모드를 인에이블 또는 디스에이블하기 위해 사용가능한 부울 값을 가질 수 있다.

[0423] 팬(pan) 특성(2652)은 카메라의 패닝(panning)을 제어하기 위해 사용될 수 있다. 예를 들어, 카메라는 수평면에서 회전가능할 수 있다. 패닝의 양은 예를 들어, 중심 위치에 대한 카메라의 최대 수평 회전의 백분율로서 명시될 수 있다. 중심 위치는 0의 팬 값을 갖는 것으로 정의될 수 있으며, 이때 팬 특성(2652)의 양의 값들은 우측으로의 패닝을 나타내고, 팬 특성(2652)의 음의 값들은 좌측으로의 패닝을 나타낸다. 도(degree)와 같은 다른 단위들이 사용될 수 있다.

[0424] 유사하게, 틸트(tilt) 특성(2653)은 카메라의 틸트 각도(예를 들어, 광축의 수평에 대한 각도)를 제어하는 데 사용될 수 있다. 틸트의 양은 예를 들어, 중심 위치에 대한 카메라의 최대 틸트의 백분율로서 명시될 수 있다. 중심 위치는 0의 틸트 값을 갖는 것으로 정의될 수 있으며, 이때 틸트 특성(2653)의 양의 값들은 상향 틸트를 나타내고, 틸트 특성(2652)의 음의 값들은 음의 틸트를 나타낸다. 도와 같은 다른 단위들이 사용될 수 있다.

[0425] 회전(rotation) 특성(2654)은 광축을 중심으로 한 카메라의 회전 각도를 제어하는 데 사용될 수 있다. 회전의 양은 예를 들어, 도 단위로 명시될 수 있다. 일부 실시예들에서, 열거형 값이 (예를 들어, 우측 90도, 좌측 90도, 180도, 및 회전 없음의 회전 설정들을 지원하기 위해) 사용될 수 있다.

[0426] 줌(zoom) 특성(2655)은 카메라에 대한 줌(또는 확대) 인자를 명시하는 데 사용될 수 있다.

[0427] 미러(mirror) 특성(2656)은, 그것을 표시, 스트리밍, 또는 저장하기 이전에, 이미지에 미러링 변환(예를 들어, 수직축을 중심으로 한 미러링)이 적용되어야 하는지 여부를 나타내기 위해 사용되는 부울 값일 수 있다.

[0428] 도 26e는 도 25a의 레코딩 서비스(2502) 및 재생 서비스(2503)에 포함될 수 있는 추가 특성들의 정의들을 도시한다. 레코딩 제어(recording control) 특성(2661)은 레코딩을 시작 또는 중지하기 위해 제어기에 의해 기록될 수 있다. 값은, 수행할 동작(예를 들어, 레코딩 시작, 레코딩 중지, 레코딩 예약)의 식별자, 및 동작에 적절한 추가 파라미터들(예를 들어, 레코딩의 지속시간, 예약된 레코딩에 대한 시작 및/또는 중지 시간 등)을 포함할

수 있는, (예를 들어, TLV 포맷 또는 다른 키-값 포맷의) 데이터 객체일 수 있다. 레코딩 상태(recording status) 특성(2662)은 레코딩 서비스가 레코딩하고 있는지 여부를 결정하기 위해 제어기에 의해 판독될 수 있다. 값은, 카메라가 현재 레코딩하고 있는지 여부의 표시를 포함할 수 있는 (예를 들어, TLV 포맷 또는 다른 키-값 포맷의) 데이터 객체일 수 있고; 다른 정보(예를 들어, 레코딩에 대한 예약된 종료 시간, 예약된 미래의 레코딩에 관한 정보, 레코딩들을 위한 이용가능한 그리고/또는 사용된 저장 공간의 양 등)가 또한 포함될 수 있다.

[0429] 재생 제어(playback control) 특성(2663)은 저장된 미디어 콘텐츠(예를 들어, 이전에 레코딩된 콘텐츠)의 재생을 제어하기 위해 제어기에 의해 기록될 수 있다. 값은, 수행할 동작(예를 들어, 재생 시작, 재생 일시 정지, 재생 종료 등)의 식별자를 포함할 수 있는 (예를 들어, TLV 포맷 또는 다른 키-값 포맷의) 데이터 객체일 수 있다. 재생될 콘텐츠 항목의 식별자와 같은 추가 파라미터들이 또한 포함될 수 있다. 재생 상태(playback status) 특성(2664)은 현재 재생 상태를 결정하기 위해 제어기에 의해 판독될 수 있다. 값은, 재생이 진행 중인지 여부의 표시를 포함할 수 있는 (예를 들어, TLV 포맷 또는 다른 키-값 포맷의) 데이터 객체일 수 있고; 다른 정보(예를 들어, 재생되고 있는 콘텐츠 항목의 식별자, 콘텐츠 항목의 지속시간, 재생 위치 등)가 또한 포함될 수 있다. 재생 속도(playback speed) 특성(2665)은 재생 속도를 제어하는 데 사용될 수 있고, 이때 값은 1.0의 표준 재생 속도에 대한 속력증가(speedup) 인수를 나타낸다. 일부 실시예들에서, 재생 속도 특성(2665)에 대한 유효 값들은, 재생 서비스(2502)의 특정 인스턴스가 지원하는 속도들로 제한될 수 있다.

[0430] 본 명세서에서 기술된 서비스들 및 특성들은 예시의 목적이다. 다른 서비스들 및 특성들이 또한 정의될 수 있다. 예를 들어, 재생될 특정 콘텐츠 항목을 식별하는 것을 용이하게 하기 위해, 제어기가 데이터베이스 또는 이용가능한 콘텐츠 항목들의 다른 목록을 내비게이팅(예를 들어, 브라우징 또는 검색)할 수 있게 하는 것이 바람직할 수 있다. 추가 특성들이 제어기에 의한 데이터베이스의 내비게이션을 용이하게 하도록 정의될 수 있다. 이들 특성은 재생 서비스(2502) 또는 원하는 바에 따른 상이한 서비스(예를 들어, 콘텐츠 브라우징 서비스)에 포함될 수 있다.

[0431] 도 27a 내지 도 27d는 함께, 본 발명의 일 실시예에 따른 IP 카메라 액세서리(2402)에 대한 액세서리 객체(2700)를 도시한다. 이 예에서, 액세서리 객체(2700)는 JSON으로 표현되고, 도 3a 내지 도 3c의 액세서리 객체(300)와 구조가 유사하다. 다른 포맷들 및 언어들이 또한 사용될 수 있다.

[0432] 도 27a에 도시된 액세서리 정보 서비스 인스턴스(2702)는 위에서 정의된 액세서리 정보 서비스(261)와 유사하거나 동일할 수 있다. 이 예에서, 액세서리 객체(2700)는 다음 3개의 다른 서비스들의 인스턴스들을 포함한다: 도 27b 및 도 27c에 도시된 IP 카메라 스트리밍 서비스(2704)(도 25a의 서비스 정의(2501)에 부합함), 도 27d에 도시된 카메라 서비스(2706)(도 25b의 서비스 정의(2503)에 부합함), 및 도 27d에 도시된 마이크로폰 서비스(2708)(도 25b의 서비스 정의(2504)에 부합함). 이 예에서, IP 카메라 액세서리(2402)는 스피커를 갖고 있지 않고, 로컬 레코딩 또는 재생 능력도 갖고 있지 않으며, 결과적으로 이들 서비스의 인스턴스들이 정의되지 않는다. (본 개시내용에 액세스할 수 있는 통상의 기술자는 그러한 서비스들의 인스턴스들에 대한 적절한 액세서리 객체들을 구성할 수 있을 것이다.) 이 예에서, 각각의 특성 인스턴스의 현재 값은 액세서리 객체(2700) 내에 명시되고; null 값이 기록-전용 특성들에 대해 명시된다.

[0433] 도 24의 IP 카메라 액세서리(2404) 및 제어기(2402)와의 사용자 상호작용은 전술한 상호작용들과 유사할 수 있다. 예를 들어, 사용자는 원하는 거동을 달성하기 위해 제어기(2402)를 동작시킬 수 있고, 제어기(2402)는 원하는 거동을 달성하도록 액세서리(2404)에 요청들(예를 들어, HTTP 요청들)을 송신할 수 있다. 도 28은 본 발명의 일 실시예에 따른 상호작용 시나리오를 보여주는 프로세스(2800)의 흐름도이다.

[0434] 블록(2802)에서, 사용자는 IP 카메라 액세서리(2404)를 셋업할 수 있다. 예를 들어, 사용자는 원하는 동작 위치 및 배향에 카메라를 배치하거나 장착하고, 전원 케이블들을 연결하고, (액세서리(2404)가 자동으로 페어링 모드로 진입하지 않는 경우) 액세서리(2404)를 페어링 모드로 진입하게 하는 것 등을 수행할 수 있다. 블록(2804)에서, 제어기(2402)는 IP 카메라 액세서리(2404)를 검색할 수 있다. 예를 들어, 전술한 바와 같이 제어기(2402)는 도 4의 프로세스(400)의 제어기-실행가능한 부분들을 구현할 수 있는 애플리케이션 프로그램을 실행할 수 있다. 애플리케이션 프로그램은 익스프레스 사용자 명령에 응답하여 또는 백그라운드 프로세스로서 실행될 수 있고, 사용자는, 원하는 액세서리가 발견되었음을 확인하거나 페어링에 이용가능한 액세서리들의 목록으로부터 액세서리(2404)를 선택하도록, 프롬프트될 수 있다. 블록(2806)에서, 제어기(2402) 및 IP 카메라 액세서리(2404)는 페어-검증된 세션을 확립할 수 있다. 예를 들어, 제어기(2402) 및 IP 카메라 액세서리(2404)는 전술한 페어 셋업 프로세스들 중 임의의 것을 수행한 다음에, 전술한 바와 같은 페어 검증 프로세스를 수행할

수 있다. 페어 셋업 동안의 검증 요건들이 원하는 대로 적용될 수 있고; 예를 들면, 문 잠금 액세서리(2004)를 참조하여 전술한 옵션들 중 임의의 것이 사용될 수 있다. 특정 검증 요건들은 특정 IP 카메라 액세서리의 성질 또는 의도된 사용에 의존할 수 있다. 예를 들어, 건물 보안 카메라는 다수의 형태의 검증을 요구할 수 있는 한편, 소비자 "웹캠" 타입의 카메라는, 카메라가 페어링 모드에 있어야 하고 사용자가 카메라 상의 라벨에 인쇄된 셋업 코드를 입력하도록 하는 것만을 요구할 수 있다.

[0435] 블록(2808)에서, 사용자는 예를 들어, IP 카메라 스트리밍 서비스(2704)의 "on" 특성(인스턴스 ID 8)에 값 "true"를 기록하려는 요청을 송신함으로써, 페어링된 제어기(2402)를 사용하여 IP 카메라 액세서리(2404)에 전원을 투입할 수 있다. 이 예에서, IP 카메라 액세서리(2404)는, 그것의 송수신기가 페어링된 제어기로부터 신호들을 수신할 수 있지만 다른 서비스들(예를 들어, 액세서리 객체(2700)에 열거된 서비스들)은 전원이 차단되는, 저전력 모드를 가질 수 있다.

[0436] 도 25a 및 도 25b에 도시된 바와 같이, 서비스들(2501 내지 2505) 중 상이한 것들은 각각 별개의 "on" 특성을 가질 수 있다. 이것은 제어기로 하여금 서비스 인스턴스들이 사용되어야 할 때 선택적으로 서비스 인스턴스들에 전원을 투입함으로써 전력 소모를 관리하게 할 수 있다. 일부 실시예들에서, 하나의 서비스의 "on" 특성을 변경하는 것은 다른 서비스들에 영향을 줄 수 있다. 예를 들어, IP 카메라 스트리밍 서비스(2501)의 전원을 오프하는 것은 모든 비디오 스트림들의 전송을 중지하는 결과를 발생시킬 수 있지만, 마이크로폰 서비스(2505) 또는 스피커 서비스(2506)의 동작을 중지할 필요는 없다. IP 카메라 스트리밍 서비스(2501)는 다른 서비스들이 오프를 유지하는 동안 전원이 온일 수 있지만, 미디어 스트림이 활성인 경우, 그 스트림과 연관된 임의의 서비스 인스턴스들은 전원이 온인 상태를 유지해야 하거나 또는 스트리밍이 중지될 수 있다. 또한, 일부 실시예들에서, IP 카메라 스트리밍 서비스(2501)는, 새로운 세션에 대한 스트림들 중 하나와 연관된 서비스 인스턴스가 전원이 오프인 경우, 새로운 세션이 시작되는 것을 허용하지 않는다. IP 카메라 서비스(2501)의 전원을 오프하는 것은, 모든 미디어 세션들을 종료하고, 다른 관련된 서비스 인스턴스들(예를 들어, 마이크로폰 서비스(2506), 스피커 서비스(2506))의 전원을 자동으로 오프할 수 있다. IP 카메라 서비스(2501)의 전원을 온하는 것은 모든 관련된 서비스 인스턴스들을 전원 온하는 결과를 발생시키지만(이들은 사용되고 있지 않는 경우 후속적으로 전원 오프될 수 있음), 임의의 이전 미디어 세션들은 복원될 필요가 없다. 다른 전원 관리 규칙들이 또한 구현될 수 있다.

[0437] 블록(2810)에서, 사용자는 예를 들어, 제어기(2402)의 사용자 인터페이스와 상호작용함으로써, IP 카메라 액세서리(2404)로부터 비디오를 스트리밍하는 것을 시작하도록 제어기(2402)에 명령할 수 있다. 예를 들어, 사용자는 카메라가 비디오를 스트리밍해야 함을 제어기(2402)에 지시할 수 있다(일부 실시예들에서, 사용자는 다른 목적지 디바이스를 명시할 수 있음). 이에 응답하여, 블록(2812)에서, 제어기(2402)는 미디어 세션을 시작하려는 요청을 생성하고 IP 카메라 액세서리(2404)에 송신할 수 있다.

[0438] 일부 실시예들에서, 시작 미디어 세션 요청은 IP 카메라 액세서리(2404)의 /characteristics URL에 대한 HTTP PUT 요청으로서 송신될 수 있다. 도 27a 내지 도 27d의 액세서리 객체(2700)를 사용하는 실시예에 대한 하나의 예가, 도 29에 도시된다. 요청 메시지(2900)는 메시지 바디(2902)를 가질 수 있다. 메시지 바디(2902)는 기록될 인스턴스의 인스턴스 식별자(2904)를 포함할 수 있다(인스턴스 ID 9는 IP 카메라 서비스 인스턴스(2704)(도 27b)에 대한 세션 시작 특성에 대응한다). 값(2906)은, IP 카메라 액세서리(2404)에 의해 제어기(2402)와 스트리밍 세션을 확립하는 데 사용가능한 정보, 예컨대 세션 ID, 제어기(2402)(또는 다른 목적지 디바이스)를 위한 IP 주소 및 포트, 및 데이터를 SRTP 스트리밍하는 데 사용될 암호화 키 및 솔트를 포함할 수 있다.

[0439] 블록(2814)에서, IP 카메라 액세서리(2404)는 시작 미디어 세션 요청에 대한 응답을 송신할 수 있다. 일부 실시예들에서, 응답은 HTTP 응답으로서 송신될 수 있다. 도 29에 도시된 시작 미디어 세션 요청의 실시예에 대한 하나의 예가, 도 30에 도시된다. 응답 메시지(3000)는 메시지 바디(3002)를 가질 수 있다. 메시지 바디(3002)는 값(3004)를 포함할 수 있다. 값(3004)은 오류 코드(3002)를 포함할 수 있으며, 이는 성공적인 완료(예를 들어, 오류 코드 0) 또는 발생한 특정 오류(예를 들어, 무효 파라미터들, 리소스가 분주함, 액세스가 거부됨 등)를 나타내는 값을 가질 수 있다. 오류가 없다고 가정하면, 값(3004)은 IP 카메라 액세서리(2404)와의 스트리밍 세션에 접속하기 위해 제어기(2402)에 의해 사용가능한 추가 정보를, 예컨대 액세서리의 IP 주소 및 포트, 및 액세서리의 SRTP 암호화 파라미터들을 제공할 수 있다.

[0440] 이 예에서, SRTP 암호화 파라미터들은 시작 세션 요청 및 응답에 포함될 수 있다. 이들 파라미터는 보통문으로 송신될 필요가 없다는 것이 이해되어야 한다. 요청(2900) 및 응답(3000)이 제어기(2402)와 액세서리(2404) 사이의 페어-검증된 세션 내에서 교환되는 실시예들에서, 파라미터들은 암호화된 형태로(페어-검증된 세션의 세션

키를 사용하여) 송신되고 따라서 침입자들로부터 보호된다.

[0441] 요청 및 응답의 대안적인 실시예들이 또한 구현될 수 있다. 전술한 예시적인 서비스 정의에서, IP 카메라 스트리밍 서비스(2501)의 주어진 인스턴스는 연관된 코덱, 속성들, 페이로드 등을 가지며; 이들은 서비스 인스턴스의 고정된 특성들일 수 있고, 상이한 세트의 특성들이 서비스의 상이한 인스턴스로서 정의될 수 있다. 다른 실시예들에서, 다른 구현이 사용될 수 있다. 예를 들면, 스트리밍 세션 시작 요청은, 원하는 보안에 따라, RTP 또는 SRTP를 사용해 전달될 수 있는 미디어 스트림들을 구성하기 위해 SDP를 활용할 수 있다. 본 명세서에서 사용되는 바와 같이, "미디어 스트림"은 하나 이상의 "미디어 흐름"으로 구성될 수 있으며, 여기서 각각의 미디어 흐름은 한 방향으로의 오디오 또는 비디오 데이터의 송신이다. 예를 들어, "단방향 비디오" 스트림은 IP 카메라 액세서리(2404)로부터 제어기(2402)로의 하나의 비디오 흐름을 포함할 수 있다. "단방향 오디오" 스트림은 IP 카메라 액세서리(2404)로부터 제어기(2402)로의 하나의 오디오 흐름을 포함할 수 있다. "단방향 오디오 및 비디오" 스트림은 IP 카메라 액세서리(2404)로부터 제어기(2402)로의 하나의 오디오 흐름 및 하나의 비디오 흐름을 포함할 수 있으며; 두 흐름은 동기화될 수 있다. "단방향 비디오 및 양방향 오디오" 스트림은 제어기(2402)로부터 IP 카메라 액세서리(2404)로의 하나의 오디오 흐름, 및 IP 카메라 액세서리(2404)로부터 제어기(2402)로의 2개의 미디어 흐름(하나의 비디오, 하나의 오디오)을 포함할 수 있으며; 후자의 두 흐름은 동기화될 수 있다. SDP는 미디어 능력들 및 전송 파라미터들을 포함하는, 미디어 세션을 설명하기 위한 명명법 및 신택스를 제공할 수 있다. 예를 들어, SDP는 다음의 정보를 기술하는 것을 지원한다: 미디어 타입들(예를 들어, 오디오, 비디오); 미디어 코덱들(예를 들어, LPCM 오디오, H.264 비디오); 네트워크 전송(예를 들어, 수신 IP 주소 및 포트); 스트림 속성들(예를 들어, 수신-전용, 송신-전용, 송신-및-수신); 및 미디어 보안(예를 들어, SRTP를 사용할지 여부). SDP는 또한 제어기와 카메라 액세서리 사이에서 미디어 세션을 협상하기 위한 편리한 모델을 제공할 수 있으며, 여기서 디바이스들은 미디어 타입들, 미디어 코덱들 등과 같은 상호 수용가능한 미디어 설정들에 수렴한다(converge).

[0442] 따라서, IP 카메라 스트리밍 서비스의 일부 대안적인 구현예들에서, 제어기(예컨대, 제어기(2402))는 시작 미디어 세션 요청 내에(예를 들어, 요청(2900)에 포함된 데이터 객체 내에) SDP 제안(offer)을 포함할 수 있다. 요청을 수신하는 액세서리(예를 들어, IP 카메라 액세서리(2404))는 시작 미디어 세션 요청에 대한 그것의 응답 내에(예를 들어, 응답(3000)에 포함된 데이터 객체 내에), 미디어 세션 요청에 대한 응답의 일부로서 SDP 응답을 포함할 수 있다.

[0443] 도 28을 다시 참조하면, 스트리밍 세션이 확립되었으면, 블록(2816)에서, IP 카메라 액세서리(2404)는 제어기(2404)(또는 시작 미디어 세션 요청에 명시된 바와 같은 다른 목적지 디바이스)로 미디어를 스트리밍하는 것을 시작할 수 있다. 특정한 스트리밍된 미디어 및 스트리밍 포맷은 액세서리의 IP 스트리밍 서비스의 특성들에 의존할 수 있으며; 예를 들어, 스트리밍은 비디오 및/또는 오디오를 포함할 수 있다. 또한, 일부 경우들에서, 스트리밍 서비스는 반대 방향의 흐름을 지원할 수 있고, 그 흐름은 동시에 시작할 수 있다. 제어기(2402)(또는 다른 목적지 디바이스)는 수신된 미디어를 사용자에게 제시하고/하거나 수신된 미디어를 이후의 제시를 위해 저장할 수 있다. 미디어 콘텐츠의 스트리밍은 무한히 계속될 수 있다.

[0444] 어떤 시점에서, 사용자는 예를 들어, 사용자 인터페이스의 "정지" 제어부를 동작시킴으로써, 블록(2818)에서 미디어 스트리밍을 정지하도록 제어기(2402)에 지시할 수 있다. 이에 응답하여, 블록(2820)에서, 제어기(2402)는 종료 미디어 세션 요청을 생성하고 IP 카메라 액세서리(2404)에 송신할 수 있다. 도 31은 요청(2900) 및 응답(3000)이 교환된 실시예에 대한 종료 미디어 세션 요청 메시지(3100)의 예를 도시한다. 시작 미디어 세션 요청(2900)과 마찬가지로, 종료 미디어 세션 요청(3000)은 HTTP PUT 요청일 수 있다. 요청 메시지(3100)는 메시지 바디(3102)를 가질 수 있다. 메시지 바디(3102)는 기록될 인스턴스의 인스턴스 식별자(3104)를 포함할 수 있다(인스턴스 ID 10은 IP 카메라 서비스 인스턴스(2704)(도 27b)에 대한 세션 종료 특성에 대응한다). 값(3106)은 제어기(2402)와의 스트리밍 세션을 종료하기 위해 IP 카메라 액세서리(2404)에 의해 사용가능한 정보를 포함할 수 있고; 이 예에서, 세션 식별자는 충분하다.

[0445] 도 28을 다시 참조하면, 블록(2822)에서, 액세서리(2404)는 요청에 대한 응답을 송신할 수 있다. 도 32는, 도 31의 요청(3100)에 응답하여 액세서리(2404)가 생성할 수 있는 종료 미디어 세션 응답(3200)의 예를 도시한다. 응답 메시지(3200)는 메시지 바디(3202)를 가질 수 있다. 메시지 바디(3202)는 값(3204)을 포함할 수 있다. 값(3204)은 오류가 발생했는지 여부를 나타내는 오류 코드를 포함할 수 있다. 오류 코드는 미디어 세션 시작 요청에 응답하기 위한 오류 코드들과 유사하게 정의될 수 있다.

[0446] 블록(2824)에서, IP 카메라 액세서리(2404)는 제어기(2402)로 미디어를 스트리밍하는 것을 중지할 수 있다.

(정의된 스트림이 반대 방향의 흐름을 포함하는 경우, 그 흐름은 동시에 종료할 수 있다.) 블록(2824) 이후에, 제어기(2402) 및 액세서리(2404)는 페어-검증된 세션 내에 남아있을 수 있고, 제어기(2402)는 스트리밍을 다시 개시하고/하거나 페어-검증된 세션 내에서 제어기(2402)의 다른 기능들을 호출할 수 있다.

[0447] 프로세스(2800)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 상이한 콘텍스트들에서, 사용자는 프로세스(2800)의 적절한 부분들을 사용하여 IP 카메라 액세서리(2404)와 상호작용할 수 있다. 일부 실시예들에서, 페어 검증은 미디어 세션을 시작하거나 종료하는 것과 같은 특정 행동들을 수행하기 전에 요구될 수 있거나, 또는 페어 검증은 IP 카메라 스트리밍 서비스와의 임의의 상호작용의 전제 조건일 수 있다. 페어-검증된 세션 내에서, 모든 요청들 및 응답들은 암호화될 수 있다. 또한, SDP, RTP 및 SRTP와 같은 특정 미디어-관련 프로토콜들이 예시의 목적으로 사용되지만, 다른 프로토콜이 대체될 수 있다.

[0448] 프로세스(2800)는, 사용자가 페어링된 액세서리를 제어하기 위해 제어기를 동작시키는 것에 의해 수행될 수 있는 다양한 상호작용 제어 동작들의 예로서 이해될 수 있으며, 이는 미디어 캡처 동작들을 포함하지만 이로 제한되지 않는다. 액세서리가 수행할 수 있는 임의의 유형의 기능 또는 동작은 본 명세서에서 기술된 것들과 유사한 프로세스들을 사용하여 제어될 수 있다.

[0449] 예: 스트리밍 인터페이스

[0450] 일부 액세서리들은 자신들과 제어기 사이의 IP 스트리밍을 지원할 수 있다. 위의 IP 카메라 예에서와 같이, 실시간 미디어 스트리밍이 하나의 옵션이지만, 다른 유형들의 데이터 스트리밍이 또한 지원될 수 있다. 예를 들어, TCP 또는 UDP 스트리밍이 지원될 수 있다. 일부 실시예들에서, 액세서리들은 암호화된 형태로 모든 데이터를 스트리밍하도록 요구될 수 있다. 예를 들어, TCP 스트림들에 대해, TLS-PSK (ChaCha20/Poly1305)가 사용될 수 있지만, UDP 스트림들에 대해서는 DTLS-PSK (ChaCha20/Poly1305)가 사용될 수 있다. 일부 실시예들에서, IP 스트리밍의 구현은 전술한 IP 카메라 서비스 예와 유사할 수 있다. 대안적인 구현예들이 또한 가능하다. 이제 일례가 기술될 것이다.

[0451] 도 33은 본 발명의 일 실시예에 따른 IP 스트리밍 서비스(3301)에 대한 예시적인 서비스 정의를 도시한다. 특성들은 스트리밍 능력(streaming capabilities) 특성(3401), 스트리밍 제어 입력(streaming control input) 특성(3402), 및 스트리밍 제어 결과(streaming control result)(3403)로서 도 34에 정의되어 있다. 이들 특성 각각은 데이터 객체일 수 있으며, 그 키들 및 값들은 나타난 바와 같을 수 있다. 선택적 "protocol-info" 키(그 값은 객체임)가 각각의 특성(3401 내지 3403)에 포함되며, 이것은 추가적인 프로토콜-특정 메타데이터를 제공하는 데 사용될 수 있다.

[0452] 이 예에서, 제어기는 스트림을 열거나 닫기 위해 스트리밍 제어 입력 특성(3402)에 기록할 수 있고, 스트리밍 제어 결과 특성(3403)을 판독함으로써 스트리밍의 상태를 획득할 수 있다. 예를 들어, 전술한 바와 같이, 페어링된 제어기는 스트리밍 제어 결과 특성(3403)에 대한 이벤트 통지들에 가입할 수 있고, 액세서리는 스트리밍 제어 결과 특성(3403)의 값이 변화할 때마다 제어기에 비요청 이벤트 응답을 송신할 수 있다. 또한, 페어링된 제어기는 또한 (예를 들어, 전술한 바와 같은 HTTP GET 요청을 사용해) 스트리밍 제어 결과 특성(3403)을 판독할 수 있다.

[0453] 일부 실시예들에서, 액세서리는 인라인 결과로 응답할지 질의 결과로 응답할지를, 요청당 기준으로 결정하는 옵션을 가질 수 있다. 인라인 및 질의 결과들의 예들이 도 5g 내지 도 5k를 참조하여 전술되었으며 이 콘텍스트에서 적용될 수 있다. 이 예에서, "on"은 IP 스트리밍 서비스(3301)의 특성으로서 도시되지 않으며; 원하는 경우에, 그것은 정의에 포함될 수 있다.

[0454] IP 스트리밍을 지원하는 액세서리는 그것의 액세서리 모델에 IP 스트리밍 서비스를 포함할 수 있다. 본 개시내용에 액세스할 수 있는 통상의 기술자는 JSON 또는 다른 설명 언어들 또는 표기법들로 적절한 표현을 구성할 수 있을 것이다.

[0455] 도 35는 본 발명의 일 실시예에 따른 제어기에 의해 실행될 수 있는 IP 스트리밍을 위한 프로세스(3500)의 흐름도이다. 블록(3502)에서, 제어기는 도 33 및 도 34에 정의된 바와 같은 IP 스트리밍 서비스를 갖는 액세서리와 페어링할 수 있다. 예를 들어, 전술한 페어 셋업 및 페어 검증 프로세스들이 사용될 수 있고, 제어기는 액세서리가 IP 스트리밍 서비스를 갖고 있음을 결정하기 위해 액세서리 정의 레코드를 판독할 수 있다.

[0456] 블록(3504)에서, 제어기는 예를 들어, 스트리밍 능력 특성(3401)에 관한 HTTP GET 요청을 송신함으로써, 액세서

리로부터 스트리밍 능력 특성(3401)을 판독할 수 있다. 이 요청은 전술한 예들에 따라 구성될 수 있다. 블록(3506)에서, 제어기는 스트림 식별자를 생성할 수 있고; 식별자들(예를 들어, UUID)을 생성하기 위한 종래의 기술들이 사용될 수 있다. 블록(3508)에서, 제어기는 예를 들어, 블록(3506)에서 생성된 스트림 식별자로 설정된 stream-id 및 1인 request-type을 갖는 스트리밍 제어 입력 특성(3402)에 대한 HTTP PUT 요청을 송신함으로써, 액세서리에 "open stream" 요청을 송신할 수 있다. 스트리밍 세션에 사용하기 위한 IP 주소 및 포트와 같은 다른 정보가, 예를 들어, protocol-info 객체에 포함될 수 있다. 요청은 전술한 예들에 따라 구성될 수 있다.

[0457] 블록(3510)에서, 제어기는 액세서리로부터 응답을 수신할 수 있다. 일부 실시예들에서, 응답은 인라인 결과 응답(예를 들어, 도 5i의 응답(573)과 유사함)일 수 있거나 또는, transaction-id를 확립하고 이어서 제어기가 스트리밍 제어 결과 특성(3403)을 판독하고 확립된 transaction-id를 참조하는, 질의 결과 응답(예를 들어, 도 5j의 응답들(574) 및 도 5k의 후속 요청(578)과 유사함)일 수 있다. 어느 경우이든, 응답은 포트 식별자, 및 스트리밍 세션에 접속하기 위해 제어기에 의해 사용가능한 임의의 다른 정보를 포함할 수 있다.

[0458] 블록(3512)에서, 제어기는 응답 내에 제공된 정보를 사용하여 스트리밍 세션에 접속할 수 있다. 블록(3514)에서, 제어기는 데이터 스트림에 대한 암호화를 셋업할 수 있다. 예를 들어, TLS 또는 DTLS 암호화(예를 들어, IETF RFC 4279 또는 IETF Internet Draft draft-keoh-lwig-dtls-iot-01에 기록된 바와 같으며, https://tools.ietf.org/html/draft-keoho-lwig-dtls-iot-01에서 이용가능한, 전송 계층 보안 또는 데이터그램 전송 계층 보안)를 위한 키들은, 페어 셋업 또는 페어 검증 동안에(블록(3502)에서) 생성된 공유 비밀 및 스트림 ID(블록(3506)에서 생성됨)로부터 도출될 수 있다. 액세서리는 동일한 정보로부터 키들을 도출할 수 있다. 블록(3516)에서, 디바이스들은 암호화된 데이터를 스트리밍할 수 있다. 데이터는 원하는 대로 (제어기로부터 액세서리로 그리고/또는 액세서리로부터 제어기로) 어느 하나의 방향 또는 두 방향으로 흐를 수 있다.

[0459] 제어기가 스트림을 중단하기로 결정하면, 블록(3518)에서, 제어기는 예를 들어, 블록(3506)에서 생성된 스트림 식별자로 설정된 stream-id 및 2인 request-type을 갖는 스트리밍 입력 포인트 특성(3402)으로의 HTTP PUT 요청을 송신함으로써, 액세서리에 "close stream" 요청을 송신할 수 있다. 이 요청은 또한 전술한 예들에 따라 구성될 수 있다. 블록(3520)에서, 제어기는 액세서리가 포트를 닫았다는 것을 확인하는 응답을 수신할 수 있다. 이는 블록(3510)에서의 응답과 동일한 방식으로 제공될 수 있다.

[0460] 프로세스(3500)는 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 순차적인 것으로서 설명되는 단계들은 병렬로 실행될 수 있고, 단계들의 순서가 변할 수 있으며, 단계들이 수정, 조합, 추가 또는 생략될 수 있다. 스트리밍 서비스는 액세서리와 제어기 사이에서 어느 방향으로든 임의의 종류의 데이터를 안전하게 스트리밍하기 위한 일반적인 IP 스트리밍 인터페이스를 제공할 수 있다. 도시된 예에서, TCP 및 UDP는 실시간 프로토콜들이 아니다. 다른 실시예들이 예를 들어, 전술한 IP 카메라 스트리밍 서비스 또는 유사한 서비스들을 사용해 실시간 스트리밍을 제공할 수 있다.

[0461] 일부 실시예들에서, 제어기는, 그것의 상태가 변하는 경우(예를 들어, 스트리밍 동안에 오류가 발생하는 경우)에 액세서리로 하여금 제어기에 경고할 수 있게 하는, 통지들에 가입할 수 있다. 예를 들어, 도 34를 참조하면, 제어기는 스트리밍 제어 결과 특성(3403)에 대한 이벤트 통지들(또는 지원될 수 있는 다른 유형들의 통지들)에 가입할 수 있다. 스트리밍 상태가 액세서리 말단에서 변하는 경우, 액세서리는 상태 코드를 업데이트하고 제어기로의 통지(예를 들어, 전술한 바와 같은 비요청 이벤트 메시지)를 생성할 수 있다.

[0462] 예시적인 디바이스

[0463] 본 명세서에서 기술된 실시예들은, 일반적으로 종래의 설계를 가지며 제어기(제1 전자 디바이스)가 액세서리(제2 전자 디바이스)의 동작을 제어할 수 있게 하는 커맨드-및-제어 동작들을 지원하는 균일한 액세서리 프로토콜에 부합하도록 구성될 수 있는, 전자 디바이스들에서 구현될 수 있다.

[0464] 도 36은 본 발명의 일 실시예에 따른 제어기(3600)의 단순화된 블록도이다. 제어기(3600)는 본 명세서에서 기술된 제어기 기능들, 거동들, 및 능력들 뿐만 아니라, 명시적으로 기술되지 않은 다른 기능들, 거동들, 및 능력들 중 임의의 것 또는 전부를 구현할 수 있다. 제어기(3600)는 처리 서브시스템(3610), 저장 디바이스(3612), 사용자 인터페이스(3614), 통신 인터페이스(3616), 보안 요소(3618), 및 암호화 로직 모듈(3620)을 포함할 수 있다. 제어기(3600)는 또한 다른 컴포넌트들(명시적으로 도시되지 않음), 예컨대 배터리, 전력 제어기들, 및 다양한 향상된 능력들을 제공하도록 동작가능한 다른 컴포넌트들을 포함할 수 있다. 다양한 실시예에서, 제어기(3600)는 데스크톱 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터, 스마트 폰, 착용가능 컴퓨팅 디바이스, 또는 임의의 원하는 폼 팩터를 갖는 다른 시스템들에서 구현될 수 있다. 또한, 전술한 바와 같이, 제어기(3600)는 부분적으

로 기지국에서, 그리고 부분적으로는 기지국과 통신하고 사용자 인터페이스를 제공하는 모바일 유닛에서 구현될 수 있다.

[0465] 저장 디바이스(3612)는 예를 들어, 디스크, 플래시 메모리, 또는 임의의 다른 비일시적 저장 매체, 또는 매체들의 조합을 이용하여 구현될 수 있으며, 휘발성 및/또는 비휘발성 매체들을 포함할 수 있다. 일부 실시예들에서, 저장 디바이스(3612)는 처리 서브시스템(3610)에 의해 실행될 하나 이상의 애플리케이션 및/또는 운용 체제 프로그램들을 저장할 수 있으며, 이는 제어기에 의해 수행되는 것으로 본 명세서에서 기술된 임의의 또는 모든 동작들을 구현하는 프로그램들을 포함한다. 예를 들어, 저장 디바이스(3612)는, 액세서리 정의 레코드를 판독할 수 있고 그 안의 정보에 기초하여 액세서리를 제어하기 위한 그래픽 사용자 인터페이스를 생성할 수 있는, 균일한 제어기 애플리케이션을 저장할 수 있다. 일부 실시예들에서, 본 명세서에서 기술된 제어기 기능성의 부분들(또는 전부)은 애플리케이션들 대신에 운영 체제 프로그램들에서 구현될 수 있다. 일부 실시예들에서, 저장 디바이스(3612)는 또한 특정 액세서리 또는 특정 카테고리들의 액세서리들을 위해 설계된 앱들(예를 들어, IP 카메라 액세서리를 관리하기 위한 IP 카메라 앱 또는 문 잠금 액세서리들과 상호작용하기 위한 보안 앱)을 저장할 수 있다.

[0466] 사용자 인터페이스(3614)는 터치 패드, 터치 스크린, 스크롤 휠, 클릭 휠, 다이얼, 버튼, 스위치, 키패드, 마이크로폰 등과 같은 입력 디바이스들 뿐만 아니라, 비디오 스크린, 지시등, 스피커, 헤드폰 잭 등과 같은 출력 디바이스들과 함께, 지원 전자기기(예를 들어, 디지털-아날로그 또는 아날로그-디지털 변환기, 신호 프로세서 등)를 포함할 수 있다. 사용자는 제어기(3600)의 기능성을 호출하기 위해 사용자 인터페이스(3614)의 입력 디바이스들을 동작시킬 수 있고, 사용자 인터페이스(3614)의 출력 디바이스들을 통해 제어기(3600)로부터의 출력을 보고/보거나 들을 수 있다.

[0467] 처리 서브시스템(3610)은 하나 이상의 집적회로, 예컨대, 하나 이상의 단일-코어 또는 멀티-코어 마이크로프로세서 또는 마이크로제어기로서 구현될 있으며, 그 예들은 종래 기술에서 알려져 있다. 동작 시, 처리 시스템(3610)은 제어기(3600)의 동작을 제어할 수 있다. 다양한 실시예에서, 처리 서브시스템(3610)는 프로그램 코드에 응답하여 각종 프로그램들을 실행할 수 있고, 다수의 동시에 실행 중인 프로그램들 또는 프로세스들을 유지할 수 있다. 임의의 주어진 시간에, 실행될 프로그램 코드의 일부 또는 전부는 처리 서브시스템(3610)에 그리고/또는 저장 디바이스(3612)와 같은 저장 매체들에 존재할 수 있다.

[0468] 적합한 프로그래밍을 통해, 처리 서브시스템(3610)은 제어기(3600)를 위한 다양한 기능성을 제공할 수 있다. 예를 들어, 일부 실시예들에서, 처리 서브시스템(3610)은 제어기에 의해 구현되는 것으로 전술한 다양한 프로세스들(또는 그 부분들)을 구현할 수 있다. 처리 서브시스템(3610)은 또한 저장 디바이스(3612)에 저장될 수 있는 프로그램들을 포함하는, 제어기(3600)의 다른 기능들을 제어하기 위한 다른 프로그램들을 실행할 수 있다. 일부 실시예들에서, 이들 프로그램은 예를 들어, 액세서리로 송신될 메시지들을 생성하고/생성하거나 액세서리로부터 메시지들을 수신함으로써, 액세서리와 상호작용할 수 있다. 그러한 메시지들은 전술한 바와 같이 균일한 액세서리 프로토콜에 부합할 수 있다.

[0469] 통신 인터페이스(3616)는 제어기(3600)를 위한 음성 및/또는 데이터 통신 능력을 제공할 수 있다. 일부 실시예들에서, 통신 인터페이스(3616)는 무선 음성 및/또는 데이터 네트워크들에 액세스하기 위한 무선 주파수(RF) 송수신기 컴포넌트들(예를 들어, 셀룰러 전화 기술, 3G, 4G/LTE, Wi-Fi(IEEE 802.11 계열 표준들)와 같은 데이터 네트워크 기술, 또는 다른 이동 통신 기술들, 또는 이들의 임의의 조합을 사용함), 단거리 무선 통신을 위한 컴포넌트들(예를 들어, 블루투스 및/또는 블루투스 LE 표준들, NFC 등을 사용함), 및/또는 다른 컴포넌트들을 포함할 수 있다. 일부 실시예들에서, 통신 인터페이스(3616)는 무선 인터페이스에 더하여 또는 그 대신에 유선 네트워크 접속성(예컨대, 이더넷)을 제공할 수 있다. 통신 인터페이스(3616)는 하드웨어(예컨대, 드라이버 회로, 안테나, 변조기/복조기, 인코더/디코더, 및 다른 아날로그 및/또는 디지털 신호 처리 회로) 및 소프트웨어 컴포넌트들의 조합을 사용하여 구현될 수 있다. 일부 실시예들에서, 통신 인터페이스(3616)는 동일한 전송 또는 상이한 전송들을 사용하여, 동시에 다수의 통신 채널들을 지원할 수 있다.

[0470] 보안 저장 모듈(3618)은 제어기(3600)를 위한 암호 정보를 안전하게 저장할 수 있는 집적회로 등일 수 있다. 보안 저장 모듈(3618) 내에 저장될 수 있는 정보의 예들은 제어기의 장기간 공개 키 및 비밀 키(3622)(전술한 바와 같은 LTPKC, LTSKC), 및 페어링된 액세서리들의 목록(3627)(예를 들어, 전술한 바와 같이 페어 셋업 또는 페어 추가 프로세스를 완료한 액세서리들에 대하여 액세서리 ID를 액세서리 장기간 공개 키 LTPKA로 매핑하는 룩업 테이블)을 포함한다.

[0471] 일부 실시예들에서, 암호화 동작들은 보안 저장 모듈(3618)과 통신하는 암호화 로직 모듈(3620)에서 구현될 수

있다. 물리적으로, 암호화 로직 모듈(3620)은, 원하는 바에 따라 보안 저장 모듈(3618)을 갖는 동일한 집적회로 또는 상이한 집적회로(예를 들어, 처리 서브시스템(3610) 내의 프로세서)에서 구현될 수 있다. 암호화 로직 모듈(3620)은, 전술한 임의의 또는 모든 암호화 동작들을 포함하는 제어기(3600)의 암호화 동작들을 구현하거나 지원하는 다양한 로직 회로들(원하는 바에 따라 고정형 또는 프로그래밍가능형)을 포함할 수 있다. 보안 저장 모듈(3618) 및/또는 암호화 로직 모듈(3620)은 제어기(3600)의 나머지 부분에 대해 "블랙 박스"처럼 보일 수 있다. 따라서, 예를 들면, 통신 인터페이스(3616)는 그것이 복호화할 수 없는 암호화된 형태로 메시지를 수신할 수 있고, 단지 메시지를 처리 서브시스템(3610)에 전달할 수 있다. 처리 서브시스템(3610) 또한 메시지를 복호화할 수 없을 수 있지만, 그것은 메시지를 암호화된 것으로 인식하고 그것을 암호화 로직 모듈(3620)에 전달할 수 있다. 암호화 로직 모듈(3620)은 (예를 들어, 보안 저장 모듈(3618)로부터 추출된 정보를 사용해) 메시지를 복호화하고, 어떤 정보가 처리 서브시스템(3610)으로 반환되어야 할지를 결정할 수 있다. 그 결과, 특정 정보는 보안 저장 모듈(3618) 및 암호화 로직 모듈(3620) 내에서만 이용가능할 수 있다. 보안 저장 모듈(3618) 및 암호화 로직 모듈(3620)이 단지 내부 보안 저장소로부터의 코드를 실행하는 단일 집적회로 상에 구현되는 경우, 이것은 정보의 추출을 매우 어렵게 할 수 있고, 이는 높은 수준의 보안을 제공할 수 있다. 다른 구현예들이 또한 가능하다.

[0472] 도 37은 본 발명의 일 실시예에 따른 액세서리(3700)의 단순화된 블록도이다. 액세서리(3700)는 본 명세서에서 기술된 액세서리 기능들, 거동들, 및 능력들 뿐만 아니라, 명시적으로 기술되지 않은 다른 기능들, 거동들, 및 능력들 중 임의의 것 또는 전부를 구현할 수 있다. 액세서리(3700)는 저장 디바이스(3728), 처리 서브시스템(3730), 사용자 인터페이스(3732), 액세서리-특정 하드웨어(3734), 통신 인터페이스(3736), 보안 요소(3738), 및 암호화 로직 모듈(3740)을 포함할 수 있다. 액세서리(3700)는 또한 다른 컴포넌트들(명시적으로 도시되지 않음), 예컨대 배터리, 전력 제어기들, 및 다양한 향상된 능력들을 제공하도록 동작가능한 다른 컴포넌트들을 포함할 수 있다.

[0473] 액세서리(3700)는 제어기(3600)와 같은 제어기에 의해 동작될 수 있는 폭넓은 클래스의 액세서리들을 대표하고, 이러한 액세서리들은 능력, 복잡성, 및 폼 팩터에 있어서 광범위하게 변할 수 있다. 다양한 액세서리들은 도 37에 명시적으로 도시되지 않은 컴포넌트들을 포함할 수 있으며, 이는 고정식 또는 이동식 저장 매체를 가진 저장 디바이스(디스크, 플래시 메모리 등); 비디오 스크린, 스피커, 또는 외부 오디오/비디오 디바이스들에 접속하기 위한 포트; 렌즈, 이미지 센서, 및 동일한 것을 위한 제어부(예를 들어, 조리개, 줌, 노출 시간, 프레임 레이트 등)와 같은 카메라 컴포넌트; 오디오를 레코딩하기 위한 마이크로폰(단독으로 또는 비디오 레코딩과 관련하여)을 포함하지만, 이들로 제한되지 않는다.

[0474] 저장 디바이스(3728)는 예를 들어, 디스크, 플래시 메모리, 또는 임의의 다른 비일시적 저장 매체, 또는 매체들의 조합을 이용하여 구현될 수 있으며, 휘발성 및/또는 비휘발성 매체들을 포함할 수 있다. 일부 실시예들에서, 저장 디바이스(3728)는 처리 서브시스템(3730)에 의해 실행될 하나 이상의 프로그램을 저장할 수 있으며, 이는 액세서리에 의해 수행되는 것으로서 전술한 다양한 동작들 뿐만 아니라 특정 액세서리 거동들에 관련된 동작들을 구현하는 프로그램들을 포함한다. 저장 디바이스(3728)는 또한 예를 들어, 전술한 바와 같이, 제어기 디바이스들에 제공될 수 있는 (예를 들어, 전술한 바와 같은) 액세서리 객체 또는 액세서리 정의 레코드를 저장할 수 있다. 저장 디바이스(3728)는 또한 액세서리 상태 정보 및 액세서리(3700)의 동작 동안에 사용될 수 있는 임의의 다른 데이터를 저장할 수 있다.

[0475] 처리 서브시스템(3730)은 예를 들어, 액세서리(3700)와 연관된 다양한 기능들을 수행하기 위한 프로그램 코드를 실행하는 하나 이상의 단일-코어 또는 멀티-코어 마이크로프로세서 및/또는 마이크로제어기를 포함할 수 있다. 예를 들어, 처리 서브시스템(3730)은 예를 들어, 저장 디바이스(3728)에 저장된 프로그램 코드를 실행함으로써, 액세서리에 의해 구현되는 것으로 본 명세서에서 기술된 임의의 또는 모든 동작들을 구현할 수 있다. 처리 서브시스템(3730)은 또한 액세서리(3730)의 다른 기능들을 제어하는 다른 프로그램들을 실행할 수 있다. 일부 경우들에서 처리 서브시스템(3730)에 의해 실행되는 프로그램들은 예를 들어, 제어기로 송신될 메시지들을 생성하고/하거나 제어기로부터 메시지들을 수신함으로써, 제어기(예를 들어, 제어기(3600))와 상호작용할 수 있다. 그러한 메시지들은 전술한 바와 같이 균일한 액세서리 프로토콜에 부합할 수 있다.

[0476] 사용자 인터페이스(3732)는 터치 패드, 터치 스크린, 스크롤 휠, 클릭 휠, 다이얼, 버튼, 스위치, 키패드, 마이크로폰 등과 같은 사용자-동작가능 입력 디바이스들 뿐만 아니라, 비디오 스크린, 지시등, 스피커, 헤드폰 잭 등과 같은 출력 디바이스들과 함께, 지원 전자기기(예를 들어, 디지털-아날로그 또는 아날로그-디지털 변환기, 신호 프로세서 등)를 포함할 수 있다. 특정 액세서리(3700)의 구현예에 따라, 사용자는 액세서리(3700)의 기능성을 호출하기 위해 사용자 인터페이스(3732)의 입력 디바이스들을 동작시킬 수 있고, 사용자 인터페이스(373

4)의 출력 디바이스들을 통해 액세서리(3700)로부터의 출력을 보고/보거나 들을 수 있다. 일부 액세서리들은 사용자 인터페이스를 최소로 제공하거나 제공하지 않을 수 있다.

[0477] 액세서리-특정 하드웨어(3734)는 그것의 기능성을 인에이블하거나 지원하는 액세서리(3700)에 존재할 수 있는 임의의 다른 컴포넌트들을 포함할 수 있다. 예를 들어, 다양한 실시예에서, 액세서리-특정 하드웨어(3734)는 고정식 또는 이동식 저장 매체들을 사용하는 하나 이상의 저장 디바이스; GPS 수신기; 전원 공급장치 및/또는 전력 관리 회로; 카메라; 마이크로폰; 하나 이상의 액추에이터; 환경 센서(예를 들면, 온도 센서, 압력 센서, 가속도계, 화학 센서 등) 등을 포함할 수 있다. 임의의 유형의 액세서리 기능성이 적절한 액세서리-특정 하드웨어(3734)를 제공함으로써 지원될 수 있다는 것이 이해될 것이다.

[0478] 통신 인터페이스(3736)는 액세서리(3700)를 위한 음성 및/또는 데이터 통신 능력을 제공할 수 있다. 일부 실시예들에서, 통신 인터페이스(3736)는 무선 음성 및/또는 데이터 네트워크들에 액세스하기 위한 무선 주파수(RF) 송수신기 컴포넌트들(예를 들어, 셀룰러 전화 기술, 3G, 4G/LTE, Wi-Fi(IEEE 802.11 계열 표준들)와 같은 데이터 네트워크 기술, 또는 다른 이동 통신 기술들, 또는 이들의 임의의 조합을 사용함), 단거리 무선 통신을 위한 컴포넌트들(예를 들어, 블루투스 및/또는 블루투스 LE 표준들, NFC 등을 사용함), 및/또는 다른 컴포넌트들을 포함할 수 있다. 일부 실시예들에서, 통신 인터페이스(3736)는 무선 인터페이스에 더하여 또는 그 대신에 유선 네트워크 접속성(예컨대, 이더넷)을 제공할 수 있다. 통신 인터페이스(3736)는 하드웨어(예컨대, 드라이버 회로, 안테나, 변조기/복조기, 인코더/디코더, 및 다른 아날로그 및/또는 디지털 신호 처리 회로) 및 소프트웨어 컴포넌트들의 조합을 사용하여 구현될 수 있다. 일부 실시예들에서, 통신 인터페이스(3736)는 동일한 전송 또는 상이한 전송들을 사용하여, 동시에 다수의 통신 채널들을 지원할 수 있다.

[0479] 보안 저장 모듈(3738)은 액세서리(3700)를 위한 암호 정보를 안전하게 저장할 수 있는 집적회로 등일 수 있다. 보안 저장 모듈(3738) 내에 저장될 수 있는 정보의 예들은 액세서리의 장기간 공개 키 및 비밀 키(3742)(전술한 바와 같은 LTPKA, LTSKA), 및 페어링된 제어기들의 목록(3744)(예를 들어, 전술한 바와 같이 페어 셋업 또는 페어 추가 프로세스를 완료한 제어기들에 대하여 제어기 ID를 제어기 장기간 공개 키 LTPKC로 매핑하는 룩업 테이블)을 포함한다.

[0480] 일부 실시예들에서, 암호화 동작들은 보안 저장 모듈(3738)과 통신하는 암호화 로직 모듈(3740)에서 구현될 수 있다. 물리적으로, 암호화 로직 모듈(3740)은, 원하는 바에 따라 보안 저장 모듈(3738)을 갖는 동일한 집적회로 또는 상이한 집적회로(예를 들어, 처리 서브시스템(3730) 내의 프로세서)에서 구현될 수 있다. 암호화 로직 모듈(3740)은, 전술한 임의의 또는 모든 암호화 동작들을 포함하는 액세서리(3700)의 암호화 동작들을 구현하거나 지원하는 다양한 로직 회로들(원하는 바에 따라 고정형 또는 프로그래밍가능형)을 포함할 수 있다. 보안 저장 모듈(3738) 및/또는 암호화 로직 모듈(3740)은 액세서리(3700)의 나머지 부분에 대해 "블랙 박스"처럼 보일 수 있다. 따라서, 예를 들면, 통신 인터페이스(3736)는 그것이 복호화할 수 없는 암호화된 형태로 메시지를 수신할 수 있고, 단지 메시지를 처리 서브시스템(3730)에 전달할 수 있다. 처리 서브시스템(3730) 또한 메시지를 복호화할 수 없을 수 있지만, 그것은 메시지를 암호화된 것으로 인식하고 그것을 암호화 로직 모듈(3740)에 전달할 수 있다. 암호화 로직 모듈(3740)은 (예를 들어, 보안 저장 모듈(3738)로부터 추출된 정보를 사용해) 메시지를 복호화하고, 어떤 정보가 처리 서브시스템(3730)으로 반환되어야 할지를 결정할 수 있다. 그 결과, 특정 정보는 보안 저장 모듈(3738) 및 암호화 로직 모듈(3740) 내에서만 이용가능할 수 있다. 보안 저장 모듈(3738) 및 암호화 로직 모듈(3740)이 단지 내부 보안 저장소로부터의 코드를 실행하는 단일 집적회로 상에 구현되는 경우, 이것은 정보의 추출을 매우 어렵게 할 수 있고, 이는 높은 수준의 보안을 제공할 수 있다. 다른 구현예들이 또한 가능하다.

[0481] 액세서리(3700)는 제어기(3600)와 같은 제어기와 상호작용하는 임의의 전자 장치일 수 있다. 일부 실시예들에서, 제어기(3600)는 전술한 바와 같이 액세서리(3700)의 동작들에 대한 원격 제어를 제공할 수 있다. 예를 들어, 제어기(3600)는 입력 및 출력 제어부들(예를 들어, 액세서리(3700)로부터 획득된 현재 상태 정보를 표시하기 위한 디스플레이 스크린, 및 상태 정보에 대한 변경을 허용하기 위한 터치스크린 오버레이와 같은 입력 제어부) 둘 다를 포함할 수 있는 액세서리(3700)를 위한 원격 사용자 인터페이스를 제공할 수 있다. 다양한 실시예에서 제어기(3600)는 액세서리(3700)의 임의의 기능을 제어할 수 있으며, 또한 액세서리(3700)로부터 데이터를 수신할 수 있다.

[0482] 도 38은 본 발명의 일 실시예에 따른 제어기(3800)를 위한 제어기 아키텍처의 예를 도시한다. 제어기 아키텍처는 상호작용하는 서브시스템들의 세트로서 도시되며, 여기서 각각의 서브시스템은 하나 이상의 모듈을 포함한다. 모듈들 각각은 하나 이상의 프로그램가능 프로세서 상에서 그리고/또는 하나 이상의 고정-기능

(fixed-function) 프로세서에서 실행되는 프로그램 코드를 사용하여 구현될 수 있고, 프로세서(들)는 다른 하드웨어 디바이스들(예컨대, 액추에이터, 디스플레이 등)을 제어하기 위한 출력 시그널링 및/또는 다른 하드웨어 디바이스들(예컨대, 키보드; 터치스크린; 액추에이터, 모터, 또는 센서로부터의 피드백 또는 상태 신호 등)로부터 신호들을 수신하기 위한 입력 시그널링을 포함할 수 있다는 것이 이해되어야 한다. 서브시스템들 중 일부는 임의의 유형의 비휘발성 저장 디바이스(예를 들어, 반도체 플래시 메모리, EEPROM, 자기 또는 광학 디스크 등)를 사용해 구현될 수 있는 지속적 데이터 저장소를 포함할 수 있다. 도시되지 않았지만, 서브시스템들의 일부 또는 전부는, 디스플레이, 키보드, 터치스크린, 마이크로폰, 스피커, 센서 등과 같은 추가적인 하드웨어 요소들을 포함할 수 있다

[0483] 보안 서브시스템(3802)은 보안 저장 요소(3804), 페어 셋업 모듈(3806), 페어 검증 모듈(3808), 페어 추가 모듈(3810), 페어 제거 모듈(3812), 및 암호화 로직 모듈(3814)을 포함할 수 있다. 보안 저장 요소(3804)는 보안 저장 요소(3618) 또는 전술된 다른 보안 저장 요소들과 유사하거나 동일할 수 있다. 일부 실시예들에서, 보안 저장 요소(3804)는 제어기(3800)에 대한 장기간 공개/비밀 키 쌍(예를 들어, 전술한 바와 같은 LTPKC, LTSKC) 뿐만 아니라, 제어기(3800)와 확립된 페어링을 갖는 각각의 액세서리에 대한 페어링 레코드들을 안전하게 저장하는 데 사용될 수 있다. 전술한 바와 같이, 각각의 페어링 레코드는 페어링된 액세서리의 식별자, 페어링된 액세서리의 장기간 공개 키, 및 선택적으로 페어링된 액세서리와 제어기(3800)의 상호작용들에 대한 허가 설정들과 같은 다른 정보(예를 들어, 제어기(3800)가 관리자 허가를 갖는지 여부)를 포함할 수 있다. 제어기(3800)가 상이한 액세서리들과 관련하여 상이한 장기간 공개 키들을 사용하는 실시예들에서, 각각의 페어링 레코드는 또한 페어링된 액세서리와 함께 사용될 장기간 공개 키의 표시자를 포함할 수 있다. 원하는 경우 다른 정보가 포함될 수 있다.

[0484] 페어 셋업 모듈(3806)은 페어 셋업 프로세스의 제어기 부분들을 구현할 수 있다. 페어 셋업 프로세스는, 제어기(3800) 및 액세서리가 각각의 디바이스가 이후에 다른 디바이스의 아이덴티티를 검증하는 데 사용할 수 있는 장기간 공개 키들을 안전하게 교환하는, 임의의 프로세스일 수 있다. 일부 실시예들에서, 페어 셋업 프로세스는, 액세서리의 아이덴티티를 검증하기 위한 제어기(3800)와 액세서리 사이의 정보 항목(예를 들어, 셋업 코드, 액세서리의 보안 인증서의 유효성)의 대역외 교환을 포함할 수 있다. 전술한 페어 셋업 프로세스들(예를 들어, 프로세스들(1300, 1400, 1500, 및/또는 1600)) 중 임의의 것, 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 셋업 모듈(3806)은 페어 셋업 동안에 액세서리와 통신을 달성하기 위해 (후술되는) 액세서리 상호작용 서브시스템(3850)과 상호작용할 수 있다. 일부 실시예들에서, 페어 셋업 모듈(3806)은 페어 셋업 프로세스와 관련하여 암호화 동작들을 수행하기 위해 암호화 로직 모듈(3814)의 기능들을 호출할 수 있다.

[0485] 페어 검증 모듈(3808)은 페어 검증 프로세스의 제어기 부분들을 구현할 수 있다. 페어 검증 프로세스는, 제어기(3800) 및 액세서리가 다른 디바이스의 아이덴티티를 검증하기 위해 이전에 저장된 장기간 공개 키들을 사용하는, 임의의 프로세스일 수 있다. 전술한 페어 검증 프로세스들 중 임의의 것(예를 들어, 프로세스(1700)) 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 검증 모듈(3808)은 페어 검증 동안에 액세서리와 통신을 달성하기 위해 (후술되는) 액세서리 상호작용 서브시스템(3850)과 상호작용할 수 있다. 일부 실시예들에서, 페어 검증 모듈(3808)은 페어 검증 프로세스와 관련하여 암호화 동작들을 수행하기 위해 암호화 로직 모듈(3814)의 기능들을 호출할 수 있다.

[0486] 페어 추가 모듈(3810)은 페어 추가 프로세스의 제어기 부분들을 구현할 수 있다. 페어 추가 프로세스는, 제어기(3800)가 액세서리와 페어링을 확립한 이후에, 액세서리가 페어링을 확립할 "새로운" 제어기에 대한 장기간 공개 키를 액세서리에 제공하는, 임의의 프로세스일 수 있으며; 새로운 제어기는 제어기(3800)와 상이한 디바이스일 수 있다. 전술한 페어 추가 프로세스들 중 임의의 것(예를 들어, 프로세스(1800)) 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 추가 모듈(3810)은 페어 추가 동안에 액세서리와 통신을 달성하기 위해 (후술되는) 액세서리 상호작용 서브시스템(3850)과 상호작용할 수 있다. 일부 실시예들에서, 페어 추가 모듈(3810)은 또한 추가될 새로운 제어기에 대한 장기간 공개 키(또는 인증서)를 획득하기 위해, 다른 제어기 또는 키 정보의 다른 외부 소스와 통신할 수 있다. 일부 실시예들에서, 페어 추가 모듈(3810)은 페어 검증 프로세스와 관련하여 암호화 동작들을 수행하기 위해 암호화 로직 모듈(3814)의 기능들을 호출할 수 있다.

[0487] 페어 추가 모듈(3812)은 페어 제거 프로세스의 제어기 부분들을 구현할 수 있다. 페어 제거 프로세스는, 제어기(3800)가 액세서리와 페어링을 확립한 이후에, 액세서리에 의해 페어링이 제거되어야 할 제어기의 식별자를 액세서리에 제공하는, 임의의 프로세스일 수 있으며; 제거되는 제어기는 제어기(3800)와 상이한 디바이스일 수 있다. 전술한 페어 제거 프로세스들 중 임의의 것(예를 들어, 프로세스(1900)) 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 제거 모듈(3812)은 페어 제거 동안에 액세서리와 통신을 달성하기 위해 (후

술되는) 액세서리 상호작용 서브시스템(3850)과 상호작용할 수 있다. 일부 실시예들에서, 페어 제거 모듈(3812)은 또한, 제거될 제어기에 대한 식별 정보를 획득하기 위해 다른 제어기 또는 정보의 다른 외부 소스와 통신할 수 있다. 일부 실시예들에서, 페어 제거 모듈(3808)은 페어 제거 프로세스와 관련하여 암호화 동작들을 수행하기 위해 암호화 로직 모듈(3812)의 기능들을 호출할 수 있다.

[0488] 암호화 로직 모듈(3814)은 제어기(3800)에 의해 사용가능한 암호화 알고리즘들을 구현할 수 있다. 예들로는, 키 생성 알고리즘; SRP에서 사용되는 알고리즘 및 함수; 해시 알고리즘; ChaCha20-Poly1305, Curve25519, Ed25519와 같은 키-기반 암호화/복호화 알고리즘, 및/또는 다른 알고리즘이 포함된다. 일부 실시예들에서, 암호화 로직 모듈(3814)은, 암호화 알고리즘들 및 관련 서비스들을 호출하기 위해 제어기(3800)의 다른 모듈들에 의해 사용가능한 API(애플리케이션 프로그램 인터페이스)를 제공할 수 있다. 임의의 수 및 조합의 암호화 알고리즘들 및 관련 서비스들이 지원될 수 있다.

[0489] 사용자 상호작용 서브시스템(3830)은 제어기(3800)의 사용자와의 상호작용들을 관리할 수 있다. 예를 들어, 사용자 인터페이스 생성 모듈(3832)은 예를 들어, 디스플레이 디바이스 상에서, 사용자에게 제시될 사용자 인터페이스를 생성할 수 있다. 사용자 인터페이스는 액세서리와 상호작용하기 위해 사용자에 의해 동작가능한 제어 요소들을 포함할 수 있다. 예를 들어, 전술한 바와 같이, 제어기(3800)는 액세서리 객체에서 제공된 정보에 기초하여 그래픽 사용자 인터페이스를 렌더링할 수 있다. 사용자 입력 수신기 모듈(3834)은 사용자 인터페이스로부터 입력을 수신하고, 입력에 응답하여 취해질 동작(예를 들어, 액세서리에 송신될 메시지들을 생성함)을 결정하도록 입력을 처리할 수 있다. 일부 실시예들에서, 사용자 입력 수신기 모듈(3834)은 사용자 입력에 응답하여 제어기(3800)의 다른 모듈들의 기능들을 호출할 수 있다.

[0490] 액세서리 상호작용 서브시스템(3850)은 제어기(3800)와 액세서리 사이의 상호작용들을 지원할 수 있다. 액세서리 객체들 저장 요소(3852)는, 휘발성 또는 비휘발성 저장 매체들(예컨대, 반도체 플래시 메모리, EEPROM, DRAM, SRAM, 자기 또는 광학 디스크 등)을 사용하여 구현될 수 있다. 일부 실시예들에서, 액세서리 객체들 저장 요소(3852)는, 제어기(3800)가 그에 대한 정보를 갖고 있는 각각의 액세서리의 표현을 저장하는 데 사용될 수 있다. 예를 들어, 전술한 바와 같이, 액세서리와 페어링을 확립한 이후에, 제어기(3800)와 같은 제어기는 하나 이상의 액세서리 객체를 포함할 수 있는 액세서리 정의 레코드를 액세서리로부터 획득할 수 있다. 제어기(3800)는 이렇게 획득된 액세서리 객체들을 액세서리 객체들 저장 요소(3852)에 저장할 수 있다. 저장된 액세서리 객체들은, (예를 들어, 사용자 인터페이스 생성 모듈(3832)에 의해) 사용자 인터페이스들을 생성하는 것, (예를 들어, 사용자 입력 수신기 모듈(3834)에 의해) 사용자 입력을 해석하는 것, 액세서리로의 요청들을 생성하는 것, 및/또는 액세서리로부터 응답들 또는 통지들을 수신하는 것을 포함하는 다수의 방법으로 사용될 수 있다.

[0491] 액세서리 검색 모듈(3854)은 액세서리를 검색하는 것과 관련된 동작들, 예를 들어, 브로드캐스트들을 청취하는 것, 검색된 액세서리와 페어링할지 여부를 결정하는 것 등을 수행할 수 있다. 예를 들어, 액세서리 검색 모듈(3854)은 도 4와 관련하여 전술한 제어기 동작들을 구현할 수 있다.

[0492] 요청 생성 모듈(3856)은 요청들을 생성하고 액세서리들에 송신할 수 있다. 예를 들어, 사용자 입력 수신기 모듈(3834)로부터의 명령어(예를 들어, 문을 잠금해제하는 것)에 응답하여, 요청 생성 모듈(3856)은 액세서리로의 적절한 요청 메시지(예를 들어, 전술한 바와 같은 잠금-상태 특성에 기록함)를 생성할 수 있다. 요청 메시지들의 예들이 위에서 기술되었다. 일부 실시예들에서, 메시지를 생성하는 것은 메시지를 암호화하는 것을 포함할 수 있고, 요청 생성 모듈(3856)은 요청을 생성하는 것과 관련하여 암호화 로직 모듈(3814)에 의해 지원되는 기능들을 호출할 수 있다. 일부 실시예들에서, 요청 생성 모듈(3856)은 페어 셋업, 페어 검증, 페어 추가, 또는 페어 제거 동작 동안에 요청들(예를 들어, 도 13 내지 도 19와 관련하여 전술한 요청들 중 임의의 것)을 생성하고 액세서리에 송신하기 위해 보안 서브시스템(3802)과 상호작용할 수 있다.

[0493] 응답 처리 모듈(3858)은 액세서리들로부터 수신될 수 있는 요청 메시지들에 대한 임의의 응답들을 수신 및 처리할 수 있다. 예를 들어, 요청 생성 모듈(3856)이 액세서리에 (예를 들어, 전술한 바와 같은 잠금-상태 특성에 기록하려는) 요청 메시지를 송신한 이후에, 응답 처리 모듈(3858)은 액세서리로부터 응답 메시지를 수신할 수 있고 메시지를 해석할 수 있다. 일부 실시예들에서, 응답 메시지는 암호화된 형태로 수신될 수 있고, 요청 처리 모듈(3858)은 응답을 해석하는 것과 관련하여 암호화 로직 모듈(3814)에 의해 지원되는 기능들을 호출할 수 있다. 응답 처리 모듈(3858)은 또한 응답(예를 들어, 상태 코드들, 오류가 발생했는지 여부 등)에 기초하여 사용자 인터페이스 서브시스템(3830)에 정보를 제공할 수 있고, 사용자 인터페이스 서브시스템(3830)은 이 정보에 기초하여 사용자에게 피드백을 생성할 수 있다. 일부 실시예들에서, 응답 처리 모듈(3858)은 또한 응답 메시지

에 포함된 정보에 기초하여 액세서리 객체들 저장 요소(3852)를 업데이트할 수 있다. 일부 실시예들에서, 응답 처리 모듈(3858)은 페어 셋업, 페어 검증, 페어 추가, 또는 페어 제거 동작 동안에 액세서리로부터의 수신된 응답들(예를 들어, 도 13 내지 도 19와 관련하여 전술한 응답들 중 임의의 것)을 수신 및 처리하기 위해 보안 서브시스템(3802)과 상호작용할 수 있다.

[0494] 통지 처리 모듈(3860)은 액세서리들로부터 수신될 수 있는 통지 메시지들을 수신 및 처리할 수 있다. 전술한 바와 같이, 다양한 통지 메커니즘이 지원될 수 있고, 통지 처리 모듈(3860)은 이들 통지 메커니즘 중 임의의 것 또는 전부(예를 들어, 전술한 프로세스들(700, 800, 900, 1000) 중 임의의 것 또는 전부)를 지원할 수 있다. 예를 들어, 수동적 통지의 경우, 통지 처리 모듈(3860)은 액세서리에 의해 보고된 상태 카운터 값을, (예를 들어, 액세서리 객체들 저장 요소(3852) 내의) 저장된 상태 카운터 값과 비교할 수 있고, 불일치를 검출할 수 있다. 일부 실시예들에서, 불일치를 검출하면, 통지 처리 모듈(3860)은 추가적인 상태 정보(예를 들어, 업데이트된 액세서리 정의 레코드 또는 그 부분들)를 획득하기 위한 요청을 생성하고 액세서리에 송신하도록 요청 생성 모듈(3856)에 지시할 수 있다. 광고형 통지의 경우에, 통지 처리 모듈(3860)은 액세서리 검색 모듈(3854)을 통해 수신된 광고들을 처리하여, (예를 들어, 액세서리 저장 요소(3852)에 저장된 액세서리 객체들의 상태 카운터들에 기초하여) 상태 변화를 갖는 알려진 액세서리를 검출할 수 있다. 이벤트 통지의 경우에, 비요청 응답 메시지는 응답 처리 모듈(3858)에 의해 수신될 수 있으며, 이는 메시지를 비요청 응답(예를 들어, 전술한 바와 같은 EVENT 메시지)으로서 인식할 수 있고 추가 처리를 위해 메시지를 통지 모듈(3860)에 제공할 수 있다. 특정 통지 메커니즘에 상관없이, 통지 모듈(3860)은 변경된 상태 정보의 성질을 결정하고, 사용자 상호작용 서브시스템(3830)에 적절한 정보를 제공할 수 있다. 일부 실시예들에서, 통지 모듈(3860)은 또한 액세서리 객체들 저장 요소(3852) 내의 저장된 액세서리 객체들을 업데이트할 수 있다.

[0495] 통신 인터페이스 모듈(3870)은, 액세서리들을 포함한 다른 디바이스들과의 통신을 지원하는 서비스들을 제공할 수 있다. 일부 실시예들에서, 통신 인터페이스 모듈(3870)은 블루투스 LE 프로토콜 스택(3872) 및/또는 HTTP/IP 프로토콜 스택(3874)을 구현할 수 있다. 블루투스 LE 프로토콜 스택(3872)은 블루투스 LE 전송 프로토콜들에 따른 발신 메시지들의 포맷팅 및 수신된 메시지들의 해석을 제공할 수 있다. HTTP/IP 프로토콜 스택(3874)은 HTTP 및 IP 전송 프로토콜들에 따른 발신 메시지들의 포맷팅 및 수신된 메시지들의 해석을 제공할 수 있다. 블루투스 LE 및 HTTP/IP가 예로서 사용되지만, 전송 프로토콜들의 임의의 조합이 통신 인터페이스 모듈(3870) 내에서 지원될 수 있고 제어기(3800)의 주어진 인스턴스가 하나 이상의 전송 프로토콜을 지원할 수 있다는 것이 이해되어야 한다. 전술한 바와 같이, 제어기(3800)는 디바이스 상호작용의 클라이언트/서버 모델에서 클라이언트 디바이스의 역할을 할 수 있고, 블루투스 LE 프로토콜 스택(3872) 및/또는 HTTP/IP 프로토콜 스택(3874)은 클라이언트 거동을 지원하도록 구성될 수 있다.

[0496] 일부 실시예들에서, 통신 인터페이스 모듈(3870) 내의 프로토콜 스택은 특정한 비표준 메시지들을 인식하도록 수정될 수 있다. 예를 들어, 전술한 바와 같이, HTTP/IP 프로토콜 스택(3874)은 액세서리로부터의 비요청 "이벤트" 메시지(예를 들어, 전술한 도 11b의 이벤트 메시지(1120))를 인식하도록 구성될 수 있다.

[0497] 일부 실시예들에서, 통신 인터페이스 모듈(3870)은 외부 디바이스들로의 메시지들을 송신 및/또는 수신하기 위해 다른 모듈들에 의해 사용가능한 API를 제공할 수 있다. API는 전송-비인식(transport-agnostic)이도록 설계될 수 있고, 특정 메시지에 대한 전송의 선택은 통신 인터페이스 모듈(3870) 내에서, 제어기(3800) 내의 다른 모듈들에 투명하게 이루어질 수 있다. 제어기(3800)의 통신 포트(도시되지 않음)에서 수신된 메시지들은 포트 구성로 기초하여 블루투스 LE 스택(3872) 또는 HTTP/IP 스택(3874)에 송신될 수 있으며, 블루투스 LE 스택(3872) 및 HTTP/IP 스택(3874) 각각은 적절하게 구성된 통신 포트로 발신 메시지들을 송신할 수 있다.

[0498] 도 39는 본 발명의 일 실시예에 따른 액세서리(3900)를 위한 액세서리 아키텍처의 예를 도시한다. 액세서리 아키텍처는 상호작용하는 서브시스템들의 세트로서 도시되며, 여기서 각각의 서브시스템은 하나 이상의 모듈을 포함한다. 모듈들 각각은 하나 이상의 프로그램가능 프로세서 상에서 그리고/또는 하나 이상의 고정-기능 프로세서에서 실행되는 프로그램 코드를 사용하여 구현될 수 있고, 프로세서(들)는 다른 하드웨어 디바이스들(예컨대, 액추에이터, 디스플레이 등)을 제어하기 위한 출력 시그널링 및/또는 다른 하드웨어 디바이스들(예컨대, 키보드; 터치스크린; 액추에이터, 모터, 또는 센서로부터의 피드백 또는 상태 신호 등)로부터 신호들을 수신하기 위한 입력 시그널링을 포함할 수 있다는 것이 이해되어야 한다. 서브시스템들 중 일부는 임의의 유형의 비휘발성 저장 디바이스(예를 들어, 반도체 플래시 메모리, EEPROM, 자기 또는 광학 디스크 등)를 사용해 구현될 수 있는 지속적 데이터 저장소를 포함할 수 있다. 도시되지 않았지만, 서브시스템들의 일부 또는 전부는, 디스플레이, 키보드, 터치스크린, 마이크로폰, 스피커, 모터, 액추에이터, 센서 등과 같은 추가적인 하드웨어 요소

들을 포함할 수 있다

[0499] 보안 서브시스템(3902)은 보안 저장 요소(3904), 페어 셋업 모듈(3906), 페어 검증 모듈(3908), 페어 추가 모듈(3910), 페어 제거 모듈(3912), 및 암호화 로직 모듈(3914)을 포함할 수 있다. 보안 저장 요소(3904)는 보안 저장 요소(3738) 또는 전술한 다른 보안 저장 요소들과 유사하거나 동일할 수 있다. 일부 실시예들에서, 보안 저장 요소(3904)는 액세서리(3900)에 대한 장기간 공개/비밀 키 쌍(예를 들어, 전술한 바와 같은 LTPKA, LTSKA) 뿐만 아니라, 액세서리(3900)가 확립된 페어링을 갖는 각각의 제어기에 대한 페어링 레코드들을 안전하게 저장하는 데 사용될 수 있다. 전술한 바와 같이, 각각의 페어링 레코드는 페어링된 제어기의 식별자, 페어링된 제어기의 장기간 공개 키, 및 선택적으로 페어링된 제어기와 액세서리(3900)의 상호작용들에 대한 허가 설정들(예를 들어, 특정한 페어링된 제어기가 관리자 허가를 갖는지 여부) 및/또는 페어링된 제어기에 대한 가입 설정들(예를 들어, 제어기가 수동적 모드 이외의 특정 통지 모드에 가입했는지 여부의 표시자)과 같은 다른 정보를 포함할 수 있다. 액세서리(3900)가 상이한 제어기들과 관련하여 상이한 장기간 공개 키들을 사용하는 실시예들에서, 각각의 페어링 레코드는 또한 페어링된 제어기와 함께 사용될 장기간 공개 키의 표시자를 포함할 수 있다. 원하는 경우 다른 정보가 포함될 수 있다.

[0500] 페어 셋업 모듈(3906)은 페어 셋업 프로세스의 액세서리 부분들을 구현할 수 있다. 페어 셋업 프로세스는, 액세서리(3900) 및 제어기가 각각의 디바이스가 이후에 다른 디바이스의 아이덴티티를 검증하는 데 사용할 수 있는 장기간 공개 키들을 안전하게 교환하는, 임의의 프로세스일 수 있다. 일부 실시예들에서, 페어 셋업 프로세스는, 액세서리(3900)의 아이덴티티를 검증하기 위한 액세서리(3900)와 제어기 사이의 정보 항목(예를 들어, 셋업 코드, 액세서리의 보안 인증서의 유효성)의 대역외 교환을 포함할 수 있다. 전술한 페어 셋업 프로세스들(예를 들어, 프로세스들(1300, 1400, 1500, 및/또는 1600)) 중 임의의 것, 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 셋업 모듈(3806)은 페어 셋업 동안에 제어기와 통신을 달성하기 위해 (후술되는) 제어기 상호작용 서브시스템(3950)과 상호작용할 수 있다. 일부 실시예들에서, 페어 셋업 모듈(3906)은 페어 셋업 프로세스와 관련하여 암호화 동작들을 수행하기 위해 암호화 로직 모듈(3914)의 기능들을 호출할 수 있다.

[0501] 페어 검증 모듈(3908)은 페어 검증 프로세스의 액세서리 부분들을 구현할 수 있다. 페어 검증 프로세스는, 액세서리(3900) 및 제어기가 다른 디바이스의 아이덴티티를 검증하기 위해 이전에 저장된 장기간 공개 키들을 사용하는, 임의의 프로세스일 수 있다. 전술한 페어 검증 프로세스들 중 임의의 것(예를 들어, 프로세스(1700)) 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 검증 모듈(3908)은 페어 검증 동안에 액세서리와 통신을 달성하기 위해 (후술되는) 제어기 상호작용 서브시스템(3950)과 상호작용할 수 있다. 일부 실시예들에서, 페어 검증 모듈(3908)은 페어 검증 프로세스와 관련하여 암호화 동작들을 수행하는 암호화 로직 모듈(3914)의 기능들을 호출할 수 있다.

[0502] 페어 추가 모듈(3910)은 페어 추가 프로세스의 액세서리 부분들을 구현할 수 있다. 페어 추가 프로세스는, 액세서리(3900)와 확립된 페어링을 갖는 제어기가 액세서리(3900)가 페어링을 확립할 "새로운" 제어기에 대한 장기간 공개 키를 액세서리(3900)에 제공하는, 임의의 프로세스일 수 있다. 전술한 페어 추가 프로세스들 중 임의의 것(예를 들어, 프로세스(1800)) 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 추가 모듈(3910)은 페어 추가 동안에 이전에 페어링된 제어기와 통신을 달성하기 위해 (후술되는) 제어기 상호작용 서브시스템(3950)과 상호작용할 수 있다. 일부 실시예들에서, 페어 추가 모듈(3910)은 또한 추가될 새로운 제어기에 대한 장기간 공개 키(또는 인증서)를 획득하기 위해 키 정보의 외부 소스와 통신할 수 있다. 일부 실시예들에서, 페어 추가 모듈(3910)은 페어 검증 프로세스와 관련하여 암호화 동작들을 수행하기 위해 암호화 로직 모듈(3914)의 기능들을 호출할 수 있다.

[0503] 페어 제거 모듈(3912)은 페어 제거 프로세스의 액세서리 부분들을 구현할 수 있다. 페어 제거 프로세스는, 액세서리(3900)와 확립된 페어링을 갖는 제어기가 액세서리(3900)에 의해 페어링이 제거되어야 할 제어기의 식별자를 액세서리(3900)에 제공하는, 임의의 프로세스일 수 있으며; 제거되는 제어기는 페어 제거 프로세스를 호출하는 제어기와 상이한 디바이스일 수 있다. 전술한 페어 제거 프로세스들 중 임의의 것(예를 들어, 프로세스(1900)) 또는 다른 프로세스들이 사용될 수 있다. 일부 실시예들에서, 페어 제거 모듈(3912)은 페어 제거 동안에 액세서리와 통신을 달성하기 위해 (후술되는) 액세서리 상호작용 서브시스템(3950)과 상호작용할 수 있다. 일부 실시예들에서, 페어 제거 모듈(3912)은 또한 제거될 제어기에 대한 식별 정보를 획득하기 위해, 다른 제어기 또는 정보의 다른 외부 소스와 통신할 수 있다. 일부 실시예들에서, 페어 제거 모듈(3908)은 페어 제거 프로세스와 관련하여 암호화 동작들을 수행하는 암호화 로직 모듈(3912)의 기능들을 호출할 수 있다.

[0504] 암호화 로직 모듈(3914)은 액세서리(3900)에 의해 사용가능한 암호화 알고리즘들을 구현할 수 있다. 예들로는, 키 생성 알고리즘; SRP에서 사용되는 알고리즘 및 함수; 해시 알고리즘; ChaCha20-Poly1305, Curve25519, Ed25519와 같은 키-기반 암호화/복호화 알고리즘, 및/또는 다른 알고리즘들이 포함된다. 일부 실시예들에서, 암호화 로직 모듈(3914)은, 암호화 알고리즘들 및 관련 서비스들을 호출하기 위해 액세서리(3900)의 다른 모듈들에 의해 사용가능한 API(애플리케이션 프로그램 인터페이스)를 제공할 수 있다. 임의의 수 및 조합의 암호화 알고리즘들 및 관련 서비스들이 지원될 수 있다.

[0505] 액세서리 행동 서브시스템(3930)은 예를 들어, 제어기 상호작용 서브시스템(3950)을 통해 제어기로부터 수신된 요청들에 응답하여, 액세서리(3900)의 하드웨어 및/또는 소프트웨어 컴포넌트들의 다양한 동작들을 관리할 수 있다. 예를 들어, 액세서리(3900)는 특정 행동들(예를 들어, 문을 열거나 닫는 것, 카메라를 동작시키는 것 등)을 취할 수 있는 다양한 동작 컴포넌트들(3932)을 통합할(또는 그와 통신할) 수 있다. 동작 컴포넌트들(3932)은 하드웨어 및/또는 소프트웨어 컴포넌트들을 포함할 수 있고, 주어진 동작 컴포넌트(3932)는 이펙터 모듈(3934)로부터의 수신된 제어 신호들(예를 들어, 디지털 또는 아날로그 형태의 전기 신호들)에 응답하고/하거나 피드백 모듈(3936)로의 피드백 신호들(예를 들어, 디지털 또는 아날로그 형태의 전기 신호들)을 생성할 수 있다.

[0506] 이펙터 모듈(3934)은 예를 들어, 사용자에 의해 요청된 동작을 달성하거나 구현하기 위해 동작 컴포넌트들(3932)로의 제어 신호들을 생성할 수 있다. 특정 신호들은 어드레싱되고 있는 특정 동작 컴포넌트(3932)에 의존할 수 있다. 예시로서, 동작 컴포넌트들(3932)은 전원을 온 또는 오프로 전환시킬 수 있는 스위칭 회로를 포함할 수 있고, 이펙터 모듈(3932)은 전원을 온 또는 오프시키기 위한 스위칭 회로로의 신호를 생성할 수 있다. 다른 예로서, 동작 컴포넌트들(3932)은 전기 제어 신호에 응답하여 물리적 물체의 움직임(예를 들어, 데드볼트를 래치결합(latching)하거나 래치결합해제하는 것, 문을 열거나 닫는 것)을 생성할 수 있는 전자기계식 액추에이터를 포함할 수 있고, 이펙터 모듈(3932)은 액추에이터로의 신호를 생성할 수 있다. 또 다른 예로서, 동작 컴포넌트들(3932)은 디지털 카메라를 제어하기 위한 API를 포함할 수 있고(구현예에 따라, 카메라 자체가 동작 컴포넌트일 수 있거나 아닐 수 있음), 이펙터 모듈(3932)은 디지털 카메라를 제어하기 위해 API 호출(call)들을 호출할 수 있다. 다양한 실시예에서, 이펙터 모듈(3934)은 제어기 인터페이스 서브시스템(3950)을 통해 수신된 제어기로부터의 요청들 및/또는 액세서리(3900)의 사용자 인터페이스에서 수신된 입력들에 응답하여 동작할 수 있다.

[0507] 피드백 모듈(3936)은 동작 컴포넌트들(3932)로부터 피드백 신호들을 수신할 수 있다. 특정 신호들은 특정 동작 컴포넌트(3932)에 의존할 수 있다. 예를 들면, 스위치 회로는 스위치의 현재 상태를 나타내는 피드백 신호를 제공할 수 있다. 전기기계식 액추에이터는 현재 상태(예를 들어, 물리적 물체의 위치 및/또는 움직임)를 나타내는 피드백 신호들을 제공할 수 있다. API는 (API 호출로부터의 복귀 시) 오류 또는 상태 코드들을 제공할 수 있다. 또 다른 예로서, 동작 컴포넌트들(3932)은 다양한 환경 조건을 위한 하나 이상의 센서(예를 들어, 움직임 센서, 위치 센서, 온도 센서, 장애물 센서 등)를 포함할 수 있으며, 피드백 모듈(3936)은 센서들로부터 센서 데이터 신호들을 수신할 수 있다. 일부 실시예들에서, 피드백 모듈(3936)은 수신된 피드백 신호들에 기초한 피드백 정보를 제어기 상호작용 서브시스템(3950)에 제공할 수 있다.

[0508] 제어기 상호작용 서브시스템(3950)은 액세서리(3900)와 제어기 사이의 상호작용들을 지원할 수 있다. 액세서리 객체(들) 저장 요소(3952)는 휘발성 또는 비휘발성 저장 매체들(예컨대, 반도체 플래시 메모리, EEPROM, DRAM, SRAM, 자기 또는 광학 디스크 등)을 사용하여 구현될 수 있다. 일부 실시예들에서, 액세서리 객체들 저장 요소(3852)는 액세서리(3900)와 상호작용하기 위해 제어기에 의해 사용될 수 있는 하나 이상의 액세서리 객체의 표현을 저장하는 데 사용될 수 있다. 저장된 액세서리 객체(들)는 요청 시(예를 들어, 제어기와 페어 검증 프로세스를 수행한 이후에) 제어기들에 제공될 수 있고, 저장된 액세서리 객체(들)는 액세서리의 상태가 변화함에 따라 업데이트될 수 있다. 예를 들어, 피드백 모듈(3936)은 동작 컴포넌트들(3932)로부터 수신된 피드백 신호들에 기초하여 저장된 액세서리 객체(들)를 업데이트할 수 있다.

[0509] 검색 모듈(3954)은 액세서리(3900)를 제어기에 대해 검색가능하게 만드는 것과 관련된 동작들, 예컨대 광고를 브로드캐스팅하는 것, 확립된 페어링을 갖지 않는 제어기로부터 페어 셋업을 수행하려는 요청을 수신하는 것 등을 수행할 수 있다. 예를 들어, 검색 모듈(3954)은 도 4와 관련하여 전술한 액세서리 동작들을 구현할 수 있다.

[0510] 요청 처리 모듈(3956)은 제어기들로부터 요청 메시지들을 수신하고 처리할 수 있다. 예를 들어, 수신된 요청 메시지(예를 들어, 전술한 바와 같이 잠금-상태 특성에 기록하는 것)에 응답하여, 요청 처리 모듈(3956)은 요청

이 허용되는지 여부(예를 들어, 제어기와의 페어-검증된 상태가 존재하는지 여부, 메시지가 유효 세션 키를 사용해 암호화되어 있는지 여부, 및 제어기가 요청된 동작을 수행할 허가를 갖는지 여부)를 결정할 수 있다. 요청이 유효하다고 가정하면, 요청 처리 모듈(3956)은 (예를 들어, 잠금 메커니즘을 작동시키기 위해) 이펙터 모듈(3934)로의 명령어들을 생성할 수 있다. 일부 실시예들에서, 요청이 허용되는지 여부를 결정하는 것은 메시지를 복호화하는 것을 포함할 수 있고, 요청 처리 모듈(3956)은 요청을 처리하는 것과 관련하여 암호화 로직 모듈(3914)에 의해 지원되는 기능들을 호출할 수 있다. 일부 실시예들에서, 요청 처리 모듈(3956)은 페어 셋업, 페어 검증, 페어 추가, 또는 페어 제거 동작 동안에 제어기로부터의 수신된 요청들(예를 들어, 도 13 내지 도 19와 관련하여 전술한 요청들 중 임의의 것)을 수신 및 처리하기 위해 보안 서브시스템(3902)과 상호작용할 수 있다.

[0511] 응답 생성 모듈(3958)은 요청 메시지들에 대한 응답들을 생성 및 송신하고 응답 메시지들을 제어기들에 송신할 수 있다. 예를 들어, 요청 처리 모듈(3956)이 요청을 수신하고 그것이 허용되지 않는다고 결정하는 경우, 요청 처리 모듈(3956)은 응답 생성 모듈(3958)에 그러한 것을 알리고, 응답 생성 모듈(3958)은 오류 응답을 생성할 수 있다. 반면에, 요청 처리 모듈(3956)이 요청을 수신하고 그것이 허용된다고 결정하는 경우, 요청 처리 모듈(3956)은 허용된 요청이 수신되었고 이펙터 모듈(3934)에 의해 처리되고 있음을 응답 생성 모듈(3958)에 알릴 수 있다. 일부 실시예들에서, 응답 모듈(3958)은 피드백 모듈(3936)로부터 피드백 정보를 수신하기를 기다릴 수 있고, 이어서 피드백 정보를 통합하는 응답 메시지를 생성할 수 있다. 예를 들어, 응답 생성 모듈(3958)이 센서를 판독하거나 잠금장치를 열려는 요청을 수신하는 경우, 응답 생성 모듈(3958)은 센서 판독치 또는 피드백 모듈(3936)로부터의 잠금장치 열림의 확인을 수신할 것을 기다릴 수 있고, 이어서 적절한 응답 메시지를 생성할 수 있다. 일부 실시예들에서, 응답 메시지는 송신 이전에 암호화될 수 있고, 응답 생성 모듈(3958)은 메시지들을 암호화하는 것과 관련하여 암호화 로직 모듈(3914)에 의해 지원되는 기능들을 호출할 수 있다. 일부 실시예들에서, 응답 생성 모듈(3958)은 페어 셋업, 페어 검증, 페어 추가, 또는 페어 제거 동작 동안에 응답들(예를 들어, 도 13 내지 도 19와 관련하여 전술한 응답들 중 임의의 것)을 생성하고 제어기에 송신하기 위해 보안 서브시스템(3902)과 상호작용할 수 있다.

[0512] 통지 생성 모듈(3960)은 (예를 들어, 액세서리 객체(들) 저장 요소(3952)에 저장된 액세서리 객체가 업데이트될 때마다) 피드백 모듈(3936)로부터 정보를 수신할 수 있고, 정보에 기초하여 제어기들로의 통지 메시지들을 생성할 수 있다. 전술한 바와 같이, 다양한 통지 메커니즘이 지원될 수 있고, 통지 생성 모듈(3960)은 이들 통지 메커니즘 중 임의의 것 또는 전부(예를 들어, 전술한 프로세스들(700, 800, 900, 1000) 중 임의의 것 또는 전부)를 지원할 수 있다. 예를 들어, 수동적 통지의 경우에, 통지 처리 모듈(3960)은 단지 액세서리 객체(들) 저장 요소(3952)에서 유지되는 내부 상태 카운터를 업데이트할 수 있다. 광고형 통지의 경우, 통지 생성 모듈(3960)은 상태 카운터를 업데이트하고, 업데이트된 상태 카운터 값을 포함하는 광고를 생성하도록 검색 모듈(3954)에 지시할 수 있다. 이벤트 통지의 경우, 통지 모듈(3960)은, 전술한 바와 같이 가입된 제어기로 송신될 비요청 응답(예를 들어, 전술한 바와 같은 EVENT 메시지)을 생성하도록 응답 생성 모듈(3958)에 지시할 수 있다. 일부 실시예들에서, 통지 모듈(3960)은 다양한 통지 메커니즘 및/또는 다양한 특성에 대한 가입된 제어기들의 목록을 유지할 수 있고, 임의의 제어기들이 가입되어 있는지 여부에 따라 하나 이상의 메커니즘을 실시하게 할 수 있다. 일부 실시예들에서, 가입 정보는 액세서리 객체(들) 저장 요소(3952) 내에 유지될 수 있다.

[0513] 통신 인터페이스 모듈(3970)은, 제어기들을 포함한 다른 디바이스들과의 통신을 지원하는 서비스들을 제공할 수 있다. 일부 실시예들에서, 통신 인터페이스 모듈(3970)은 블루투스 LE 프로토콜 스택(3972) 및/또는 HTTP/IP 프로토콜 스택(3974)을 구현할 수 있다. 블루투스 LE 프로토콜 스택(3972)은 블루투스 LE 전송 프로토콜들에 따른 발신 메시지들의 포맷팅 및 수신된 메시지들의 해석을 제공할 수 있다. HTTP/IP 프로토콜 스택(3974)은 HTTP 및 IP 전송 프로토콜들에 따른 발신 메시지들의 포맷팅 및 수신된 메시지들의 해석을 제공할 수 있다. 블루투스 LE 및 HTTP/IP가 예로서 사용되지만, 전송 프로토콜들의 임의의 조합이 통신 인터페이스 모듈(3970) 내에서 지원될 수 있고 제어기(3900)의 주어진 인스턴스가 하나 이상의 전송 프로토콜을 지원할 수 있다는 것이 이해되어야 한다. 전술한 바와 같이, 액세서리(3900)는 디바이스 상호작용의 클라이언트/서버 모델에서 서버 디바이스의 역할을 할 수 있고, 블루투스 LE 프로토콜 스택(3872) 및/또는 HTTP/IP 프로토콜 스택(3874)은 서버 거동을 지원하도록 구성될 수 있다.

[0514] 일부 실시예들에서, 통신 인터페이스 모듈(3970) 내의 프로토콜 스택은 특정한 비표준 메시지들을 생성하도록 수정될 수 있다. 예를 들어, 전술한 바와 같이, HTTP/IP 프로토콜 스택(3974)은 액세서리로부터 비요청 "이벤트" 메시지(예를 들어, 전술한 도 11b의 이벤트 메시지(1120))를 생성하도록 구성될 수 있다.

[0515] 일부 실시예들에서, 통신 인터페이스 모듈(3970)은 외부 디바이스들에 메시지들을 송신 및/또는 수신하기 위해

다른 모듈들에 의해 사용가능한 API를 제공할 수 있다. API는 전송-비인식이도록 설계될 수 있고, 특정 메시지에 대한 전송의 선택은 통신 인터페이스 모듈(3970) 내에서, 액세서리(3900) 내의 다른 모듈들에 투명하게 이루어질 수 있다. 액세서리(3900)의 통신 포트(도시되지 않음)에서 수신된 메시지들은 포트 구성에 기초하여 블루투스 LE 스택(3972) 또는 HTTP/IP 스택(3974)에로송신될 수 있으며, 블루투스 LE 스택(3972) 및 HTTP/IP 스택(3974) 각각은 적절하게 구성된 통신 포트로 발신 메시지들을 송신할 수 있다.

[0516] 본 명세서에서 기술되는 시스템 구성들 및 컴포넌트들은 예시적이며, 변형들 및 수정들이 가능하다는 것이 이해될 것이다. 제어기(3600)(또는 제어기(3800))의 구현예는 제어기에 의해 수행되는 것으로서 전술한 동작들 중 임의의 것 또는 전부를 수행할 수 있고, 액세서리(3700)(또는 액세서리(3800))의 구현예는 액세서리에 의해 수행되는 것으로서 전술한 동작들 중 임의의 것 또는 전부를 수행할 수 있으며; 상이한 도면들과 관련한 상이한 참조 부호들의 사용은 다른 식으로 의미하도록 의도되지 않는다는 것이 이해되어야 한다. 제어기 및/또는 액세서리는 본 명세서에서 구체적으로 기술되지 않은 다른 능력들(예컨대, 휴대 전화, 위성 위치확인 시스템(GPS), 광대역 데이터 통신, 인터넷 접속 등)을 가질 수 있다. 구현예에 따라, 디바이스들은, 디바이스들 중 어느 하나(또는 둘 다)에 의해 지원되는 임의의 기능성을 제공하거나 또는 각각의 디바이스에서 부분적으로 구현되는 기능성을 제공하기 위해 상호동작할 수 있다. 일부 실시예들에서, 특정 액세서리는, 특정 제어기를 통해서는 액세스가능하거나 호출가능하지 않지만 다른 제어기를 통해서 또는 액세서리와 직접 상호작용함으로써 액세스가능한, 일부 기능성을 가질 수 있다.

[0517] 또한, 제어기 및 액세서리가 특정 블록들을 참조하여 본 명세서에서 기술되지만, 이들 블록들이 설명의 편리함을 위해 정의되며 컴포넌트 부분들의 특정 물리적 배열을 의미하도록 의도되지 않는다는 것이 이해될 것이다. 또한, 블록들은 물리적으로 별개인 컴포넌트들에 대응할 필요는 없다. 블록들은 예컨대 프로세서를 프로그래밍하거나 적절한 제어 회로를 제공함으로써 다양한 동작들을 수행하도록 구성될 수 있으며, 다양한 블록들은 초기 구성이 어떻게 획득되는지에 재구성 가능하거나 재구성 가능하지 않을 수 있다. 본 발명의 실시예들은 회로 및 소프트웨어의 임의의 조합을 사용하여 구현된 전자 디바이스들을 포함한, 다양한 장치에서 실현될 수 있다.

[0518] 다수의 동작 및 상호작용이 지원될 수 있다. 예를 들어, 일부 실시예들에서, 액세서리는, 그것이 제어기와 페어링하는 데 이용가능하다는 것을 나타내는 광고를 디바이스 검색 서비스 상에서 브로드캐스팅할 수 있다. 제어기는 예를 들어, 광고를 검출함으로써 액세서리를 발견할 수 있고, 페어 셋업 프로세스(예를 들어, 도 13 내지 도 16를 참조하여 전술한 프로세스들 중 임의의 것)를 개시하여, 예를 들어, 장기간 키들, 및 셋업 코드 또는 보안 인증서와 같은 대역외 정보 항목을 안전하게 교환함으로써, 페어링을 확립할 수 있다. 페어 셋업 프로세스가 완료되면, 액세서리는 자신이 (예를 들어, 임의의 페어 셋업 시작 요청들에 응답하여 오류를 무시하거나 반환함으로써) 임의의 추가 제어기들과 페어 셋업을 수행하는 데 이용가능하지 않게 할 수 있고, 페어 셋업을 수행한 제어기는 관리자 허가를 승인받을 수 있다. 관리자 허가를 갖는 제어기는, (예를 들어, 도 18과 관련하여 전술한 바와 같은) 페어 추가 프로세스를 수행하거나 또는 전술한 다른 위임된 페어링 프로세스들(예를 들어, 신뢰 인증서 체인을 액세서리에 제공하는 것)을 사용함으로써, 하나 이상의 추가 제어기와 페어링을 확립하도록 액세서리에 지시할 수 있다. 관리자 허가를 갖는 제어기는 또한, (예를 들어, 도 19a 및 도 19b를 참조하여 전술한 바와 같은) 페어 제거 프로세스를 수행함으로써 하나 이상의 다른 제어기와의(또는 관리자 허가를 갖는 제어기와의) 확립된 페어링을 제거하도록 액세서리에 지시할 수 있다.

[0519] 이러한 방식으로, 제어기 및 액세서리의 사용자는, 액세서리와 페어링하는 다른 제어기들에 대한 제어를 유지할 수 있는데, 이는 사용자의 제어기가 임의의 추가 페어링을 확립하는 것에 참여하도록 요구될 수 있기 때문이다. 일부 실시예들에서, 사용자는 예를 들어, 하나 이상의 추가 제어기에 관리자 허가를 승인하도록 액세서리에 지시함으로써, 다른 것들과 그 제어를 공유할 수 있다. 관리자 허가를 갖는 제어기의 사용자는 또한, 더 이상 원하지 않는 임의의 페어링들을 포함한, 확립된 페어링을 제거하도록 액세서리에 지시할 수 있다.

[0520] 액세서리와 페어링을 확립한 제어기("페어링된 제어기"로도 지칭됨)는, 액세서리에 대한 계속적인 접속을 반드시 유지하지는 않으면서도 액세서리에 대한 제어를 수행할 수 있다. 예를 들어, 페어링된 제어기가 액세서리에 재접속하는 경우, 제어기는, 액세서리와 제어기가 그들 사이에 확립된 페어링이 존재함을 검증할 수 있게 하는 (예를 들어, 도 17을 참조하여 전술한 바와 같은) 페어 검증 프로세스를 개시할 수 있다. 페어 검증 프로세스는 또한, 액세서리와 제어기가 접속을 유지하는 동안 송신될 수 있는 임의의 후속 메시지들을 안전하게 하는 데 사용가능한 하나 이상의 세션 키를 제공할 수 있으며; 이 조건은 "페어-검증된 세션"으로 지칭될 수 있다.

[0521] 접속 동안, 제어기는 커맨드-및-제어 메시지들(또는 요청 메시지들)을 액세서리에 송신할 수 있다. 적절한 요청 메시지들을 사용함으로써, 제어기는 액세서리의 현재 상태를 결정할 수 있고, 일부 경우들에서는 그것의 현

재 상태의 양상을 변경하도록 액세서리에 지시할 수 있다. 예를 들어, 액세서리는 특성들 및 서비스들의 집합으로서 그것의 상태를 설명하는 액세서리 모델을 (예를 들어, 액세서리 객체로서) 유지할 수 있다. 제어기는 특성들 중 하나 이상(이는, 특성들 전부 또는 그것의 임의의 서브셋을 포함할 수 있음)을 판독하거나 달리 질의함으로써 액세서리의 현재 상태의 양상들을 결정할 수 있고, 특성들 중 하나 이상에 기록하거나 달리 수정함으로써 그것의 현재 상태의 양상을 변경하도록 액세서리에 지시할 수 있다. 액세서리가 모든 그러한 요청들이 페어-검증된 세션 내에서 암호화된 메시지들로서 송신될 것을 요구하는 경우, 이어서 액세서리의 동작은 인가된 제어기들로 제한될 수 있다.

[0522] 액세서리는 자기-정의(self-defining)할 수 있다. 예를 들어, 페어를 확립한 이후에, 페어링된 제어기는 액세서리로부터 액세서리의 정의 레코드를 요청할 수 있다. 액세서리 정의 레코드는, 제어기와 페어링되는 액세서리를 통해 제어될 수 있는 각각의 액세서리에 대한 액세서리 객체를 정의할 수 있다. 일부 실시예들에서, 액세서리 정의 레코드의 전부 또는 일부는 액세서리가 임의의 페어링을 확립하기 전에 제어기에 이용가능하게 되어, 액세서리 정의 레코드로부터의 정보가 페어링을 확립할지 여부를 결정하는 데 사용되게 할 수 있다. 다른 실시예들에서, 제어기는 액세서리의 광고에 포함된 정보(예를 들어, 전술한 바와 같은 TXT 레코드)에 기초하여 페어링을 확립할지 여부를 결정할 수 있고, 액세서리 정의 레코드는 페어링이 확립된 이후에만 제어기에 이용가능해질 수 있다. 페어링된 제어기는 특성들을 질의 및 또는 수정하려는 요청들을 생성하기 위해 액세서리 정의 레코드를 사용함으로써, 액세서리의 제어를 가능하게 할 수 있다. 제어기는, 원하는 바에 따라, 사용자 입력에 응답하여 또는 (예를 들어, 제어기가 다양한 조건들을 검출하는 것에 기초하여) 자동으로, 그러한 요청들을 생성할 수 있다. 일부 실시예들에서, 제어기는 액세서리를 제어하도록 동작가능한 사용자 인터페이스를 동적으로 생성할 수 있으며, 이때 사용자 인터페이스는 액세서리 정의 레코드에 제공된 정보에 기초한다.

[0523] 일부 실시예들에서, 액세서리는 그것의 상태의 변화들에 대해 임의의 페어링된 제어기들에 통지할 수 있다. 예를 들어, (예컨대, 도 7에 도시된 바와 같은) 수동적 통지 프로세스들, (예컨대, 도 8에 도시된 바와 같은) 광고형 통지 프로세스들, (예컨대, 도 9에 도시된 바와 같은) 능동적 통지 프로세스들, 및/또는 (예컨대, 도 10에 도시된 바와 같은) 이벤트-메시지 통지 프로세스들의 임의의 조합이 지원될 수 있다. 일부 실시예들에서, 페어링된 제어기는, 특정 특성에 관하여 특정 통지 방법(예를 들어, 광고형, 능동적, 및/또는 이벤트-메시지)에 가입하려는 요청을 액세서리에 송신할 수 있다. 액세서리는 다양한 제어기에 대한 가입 상태 정보를 유지할 수 있고, 현재 가입 상태에 기초하여 특정 유형의 통지들을 생성할 수 있다.

[0524] 추가 실시예들

[0525] 본 발명은 특정 실시예들에 대하여 설명되었지만, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 다수의 수정들이 가능하다는 것을 인식할 것이다. 단일 제어기는 본 명세서에서 기술된 프로세스들을 사용하여, 임의의 수의 액세서리들과 페어링을 확립하고 선택적으로 상이한 시간들에 상이한 액세서리들과 통신할 수 있다. 유사하게, 단일 액세서리는 그것이 (예를 들어, 전술한 바와 같이 페어 셋업 및 페어 추가를 사용해) 페어링을 확립한 다수의 제어기들에 의해 제어될 수 있다. 액세서리의 임의의 기능은, 하나 이상의 특성을 갖는 서비스로서 기능을 모델링하고 제어기가 서비스 및/또는 그것의 특성들과 상호작용하도록(예를 들어, 판독, 수정, 업데이트들을 수신) 허용함으로써, 제어될 수 있다. 따라서, 본 명세서에서 기술된 바와 같은 프로토콜들 및 통신 프로세스들은 "균일"할 수 있으며, 이는 액세서리 기능 또는 제어기 폼 팩터 또는 특정 인터페이스들에 상관없이 하나 이상의 제어기 및 하나 이상의 액세서리와의 임의의 콘텍스트에서 그것들이 적용될 수 있음을 의미한다.

[0526] 또한, 상기 일부 예들은, 표준 인터넷-프로토콜(IP 전송 스택(예컨대, TCP/IP))을 지원하는 로컬-영역 및 광역-영역 네트워크들을 통해 사용될 수 있는 프로토콜인, HTTP를 특별히 참조한다. 그러나, 다른 전송 프로토콜들이 또한 사용될 수 있다. 예를 들어, 블루투스 LE 프로토콜 스택은 하나의 디바이스가 다른 디바이스의 속성들을 질의 및 수정할 수 있도록 허용하는 일반적인 속성(GATT) 계층을 포함한다. 일부 실시예들에서, 액세서리 특성들의 인스턴스들은 GATT 모델에 기초한 속성들로서 제어기들에 노출될 수 있다. 따라서, 제어기는 또한 블루투스 LE를 사용해 액세서리 특성들을 질의(예를 들어, 판독) 및 수정(예를 들어, 기록)할 수 있다. 일부 실시예들에서, 특정 액세서리는 IP 및/또는 블루투스 LE 전송 프로토콜들 중 어느 하나 또는 둘 다를 지원할 수 있으며, 제어기는, 액세서리의 능력들에 따라 그리고 제어기에 의해 확립된 선호도에 따라, IP를 사용해 일부 액세서리들과 상호작용하고 그리고 블루투스 LE를 사용해 다른 액세서리들과 상호작용할 수 있다.

[0527] 본 명세서에서 기술된 다양한 특징들, 예를 들어, 방법들, 장치들, 컴퓨터 판독가능 매체들 등은, 전용 컴포넌트들 및/또는 프로그램가능한 프로세서들 및/또는 다른 프로그램가능한 디바이스들의 임의의 조합을 이용하여

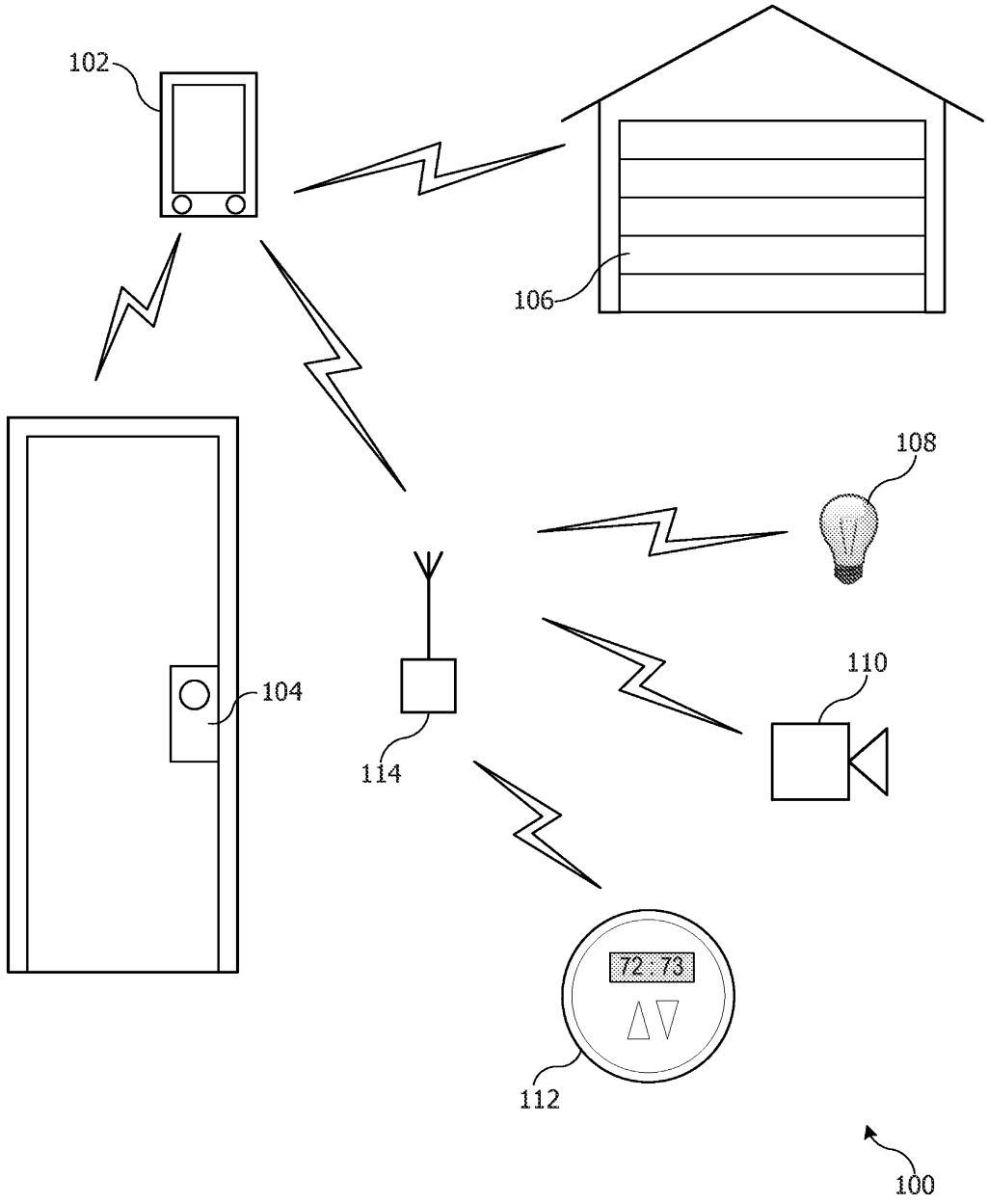
실현될 수 있다. 본 명세서에 기술된 다양한 프로세스들은 동일한 프로세서 또는 임의의 조합의 상이한 프로세서들 상에서 구현될 수 있다. 컴포넌트들이 소정 동작들을 수행하도록 구성되는 것으로 기술되는 경우에, 그러한 구성은 예컨대 동작을 수행하도록 전자 회로들을 설계함으로써, 동작을 수행하도록 프로그램가능한 전자 회로들(예컨대 마이크로프로세서들)을 프로그래밍함으로써, 또는 이들의 임의의 조합에 의해 달성될 수 있다. 또한, 전술한 실시예들은 특정 하드웨어 및 소프트웨어 컴포넌트들을 참조할 수 있지만, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 하드웨어 및/또는 소프트웨어 컴포넌트들의 상이한 조합들이 또한 사용될 수 있으며, 하드웨어로 구현되는 것으로서 기술된 특정 동작들이 또한 소프트웨어로 구현될 수 있거나 그 역도 마찬가지임을 이해할 것이다.

[0528] 본 명세서에서 기술된 다양한 특징들을 통합한 컴퓨터 프로그램들이 다양한 컴퓨터 판독가능 저장 매체들 상에 인코딩 및 저장될 수 있으며; 적합한 매체들은 자기 디스크 또는 테이프, 콤팩트 디스크(CD) 또는 DVD(digital versatile disk)와 같은 광 저장 매체, 플래시 메모리, 및 다른 비일시적 매체를 포함한다. 프로그램 코드로 인코딩된 컴퓨터 판독가능 매체는 호환 가능한 전자 디바이스와 패키징될 수 있거나, 또는 프로그램 코드는 (예컨대, 인터넷 다운로드를 통해 또는 별도로 패키징된 컴퓨터 판독가능 저장 매체로서) 전자 디바이스들과는 별개로 제공될 수 있다.

[0529] 따라서, 본 발명이 특정 실시예들에 대하여 기술되었지만, 본 발명은 하기의 청구범위의 범주 내의 모든 수정들 및 등가물들을 커버하도록 의도된다는 것이 이해될 것이다.

도면

도면1



도면2a

특성	속성	값
온 201	타입	com.proto.ch.on
	허가들	Paired Read, Paired Write
	포맷	<boolean>
사용중인 콘센트 202	타입	com.proto.ch.outlet-in-use
	허가들	Paired Read
	포맷	<boolean>
밝기 203	타입	com.proto.ch.brightness
	허가들	Paired Read, Paired Write
	포맷	<integer>
	최소, 최대, 단계	0, 100, 1
	단위	percentage
색상 204	타입	com.proto.ch.hue
	허가들	Paired Read, Paired Write
	포맷	<float>
	최소, 최대, 단계	0, 360, 1
	단위	arcdegrees
채도 205	타입	com.proto.ch.saturation
	허가들	Paired Read, Paired Write
	포맷	<float>
	최소, 최대, 단계	0, 100, 1
	단위	percentage
오디오 피드백 206	타입	com.proto.ch.audio-feedback
	허가들	Paired Read, Paired Write
	포맷	<boolean>
출력 음량 207	타입	com.proto.ch.volume.output
	허가들	Paired Read, Paired Write
	포맷	<float>
	최소, 최대, 단계	0, 100, 1
	단위	percentage
로그들 208	타입	com.proto.ch.logs
	허가들	Paired Read
	포맷	<tlv>

도면2b

특성	속성	값
현재 온도 209	타입	com.proto.ch.temp.current
	허가들	Paired Read
	포맷	<float>
	최소, 최대, 단계	0, 100, 0.1
	단위	celsius
목표 온도 210	타입	com.proto.ch.temp.target
	허가들	Paired Read, Paired Write
	포맷	<float>
	최소, 최대, 단계	10, 38, 0.1
	단위	celsius
온도 단위 211	타입	com.proto.ch.temp.units
	허가들	Paired Read, Paired Write
	포맷	<enum>
	유효 값들	“Celsius”, “Fahrenheit”
현재 난방/냉방 상태 212	타입	com.proto.ch.heat-cool.current
	허가들	Paired Read
	포맷	<enum>
	유효 값들	“Heat”,”Cool”, “Off”
목표 난방/냉방 모드 213	타입	com.proto.ch.heat-cool.target
	허가들	Paired Read, Paired Write
	포맷	<enum>
	유효 값들	“Heat”,”Cool”, “Auto”, “Off”
냉방 임계치 온도 214	타입	com.proto.ch.temp.cool-threshold
	허가들	Paired Read, Paired Write
	포맷	<float>
	최소, 최대, 단계	10, 35, 0.1
	단위	celsius
난방 임계치 온도 215	타입	com.proto.ch.temp.heat-threshold
	허가들	Paired Read, Paired Write
	포맷	<float>
	최소, 최대, 단계	0, 25, 0.1
	단위	celsius

도면2c

특성	속성	값
현재 문 상태 216	타입	com.proto.ch.door-state.current
	허가들	Paired Read
	포맷	<enum>
	유효 값들	Open, Opening, Closed, Closing, Stopped
목표 문 상태 217	타입	com.proto.ch.door-state.target
	허가들	Paired Read, Paired Write
	포맷	<enum>
	유효 값들	Open, Closed
움직임 검출 218	타입	com.proto.ch.motion-detected
	허가들	Paired Read
	포맷	<boolean>
장애물 검출 219	타입	com.proto.ch.obstruction-detected
	허가들	Paired Read
	포맷	<boolean>
잠금 메커니즘 현재 상태 220	타입	com.proto.ch.lock-mech.current-state
	허가들	Paired Read
	포맷	<enum>
	유효 값들	Unsecured, Secured, Jammed, Unknown
잠금 메커니즘 목표 상태 221	타입	com.proto.ch.lock-mech.target-state
	허가들	Paired Read, Paired Write
	포맷	<enum>
	유효 값들	Unsecured, Secured
잠금 메커니즘 마지막 행동 222	타입	com.proto.ch.lock-mech.last-action
	허가들	Paired Read, Paired Write
	포맷	<enum>
	유효 값들	Secured/Unsecured, Physical/Remote, Interior/ Exterior, etc.
잠금 관리 자동 타임아웃 223	타입	com.proto.ch.lock-mgt.auto-timeout
	허가들	Paired Read, Paired Write
	포맷	<int>
	최소, 최대, 단계	0, intmax, 1
잠금 관리 제어 포인트 224	타입	com.proto.ch.lock-mgt.control-point
	허가들	Paired Write
	포맷	<tlv>

도면2d

특성	속성	값
회전 방향 225	타입	com.proto.ch.rotation-direction
	허가들	Paired Read, Paired Write
	포맷	<enum>
	유효 값들	Clockwise, Counterclockwise
회전 속도 226	타입	com.proto.ch.brightness
	허가들	Paired Read, Paired Write
	포맷	<integer>
	최소, 최대, 단계	0, 100, 1
	단위	percentage
이름 227	타입	com.proto.ch.name
	허가들	Paired Read
	포맷	<string>
관리자-전용 액세스 228	타입	com.proto.ch.admin-only-access
	허가들	Paired Read, Admin Write
	포맷	<boolean>
버전 229	타입	com.proto.ch.version
	허가들	Paired Read
	포맷	<string>

도면2e

	속성	포맷
230	타입	<string>
231	허가들	Array of <string>
232	통지모드	Array of <string>
233	포맷	<string>
234	값	[포맷 속성에 의해 명시된 바와 같음]
235	최소값(최소)	<number>
236	최대값(최대)	<number>
237	크기단계(단계)	<number>
238	유효값들	Array of <string>
239	단위	<string>
240	사용자디스크립터	<string>
241	소유자	<string>

도면2f

속성	스트로브 (242)	방향 (243)	속도 (244)
타입	com.discoball.ch.strobe	com.discoball.ch.rotate.direction	com.discoball.ch.rotate.speed
허가들	PR, PW	PR, PW	PR, PW
통지모드	Events	Events	Events
포맷	boolean	enum	<int>
값	false	not rotating	1
최소값	--	--	0
최대값	--	--	1
크기단계	--	--	1
유효값들	--	Not rotating, Clockwise, Counterclockwise	--
단위	--	--	--
사용자디스크립터	Strobe on or off	Rotation direction for ball	Rotation speed for ball
소유자	Discoball Inc.	Discoball Inc.	Discoball Inc.

도면2g

서비스	속성	값
전구 251	타입	com.proto.svc.lightbulb
	필수 특성	com.proto.ch.on
	선택적 특성	com.proto.ch.brightness
		com.proto.ch.hue
		com.proto.ch.saturation
		com.proto.ch.name
온도조절기 252	타입	com.proto.svc.thermostat
	필수 특성	com.proto.ch.heat-cool.current
		com.proto.ch.heat-cool.target
		com.proto.ch.temp.current
		com.proto.ch.temp.target
		com.proto.ch.temp.units
	선택적 특성	com.proto.ch.temp.cool-threshold
		com.proto.ch.temp.heat-threshold
		com.proto.ch.name
차고문 열림장치 253	타입	com.proto.svc.garage-door-opener
	필수 특성	com.proto.ch.door-state.current
		com.proto.ch.door-state.target
		com.proto.ch.obstruction-detected
	선택적 특성	com.proto.ch.lock-mech.current-state
		com.proto.ch.lock-mech.target-state
		com.proto.ch.name
잠금 메커니즘 254	타입	com.proto.svc.lock-mech
	필수 특성	com.proto.ch.lock-mech.current-state
		com.proto.ch.lock-mech.target-state
	선택적 특성	com.proto.ch.name

도면2h

서비스	속성	값
잠금 관리 255	타입	com.proto.svc.lock-mgt
	필수 특성	com.proto.ch.lock-mgt.control-point
		com.proto.ch.version
	선택적 특성	com.proto.ch.logs
		com.proto.ch.audio-feedback
		com.proto.ch.admin-only-access
		com.proto.ch.lock-mgt.auto-timeout
		com.proto.ch.lock-mech.last-action
		com.proto.ch.door-state.current
		com.proto.ch.motion-detected
		com.proto.ch.name
콘센트 256	타입	com.proto.svc.outlet
	필수 특성	com.proto.ch.on
		com.proto.ch.outlet-in-use
	선택적 특성	com.proto.ch.name
스위치 257	타입	com.proto.svc.switch
	필수 특성	com.proto.ch.on
	선택적 특성	com.proto.ch.name
팬 258	타입	com.proto.svc.fan
	필수 특성	com.proto.ch.on
	선택적 특성	com.proto.ch.rotation.direction
		com.proto.ch.rotation.speed
		com.proto.ch.name

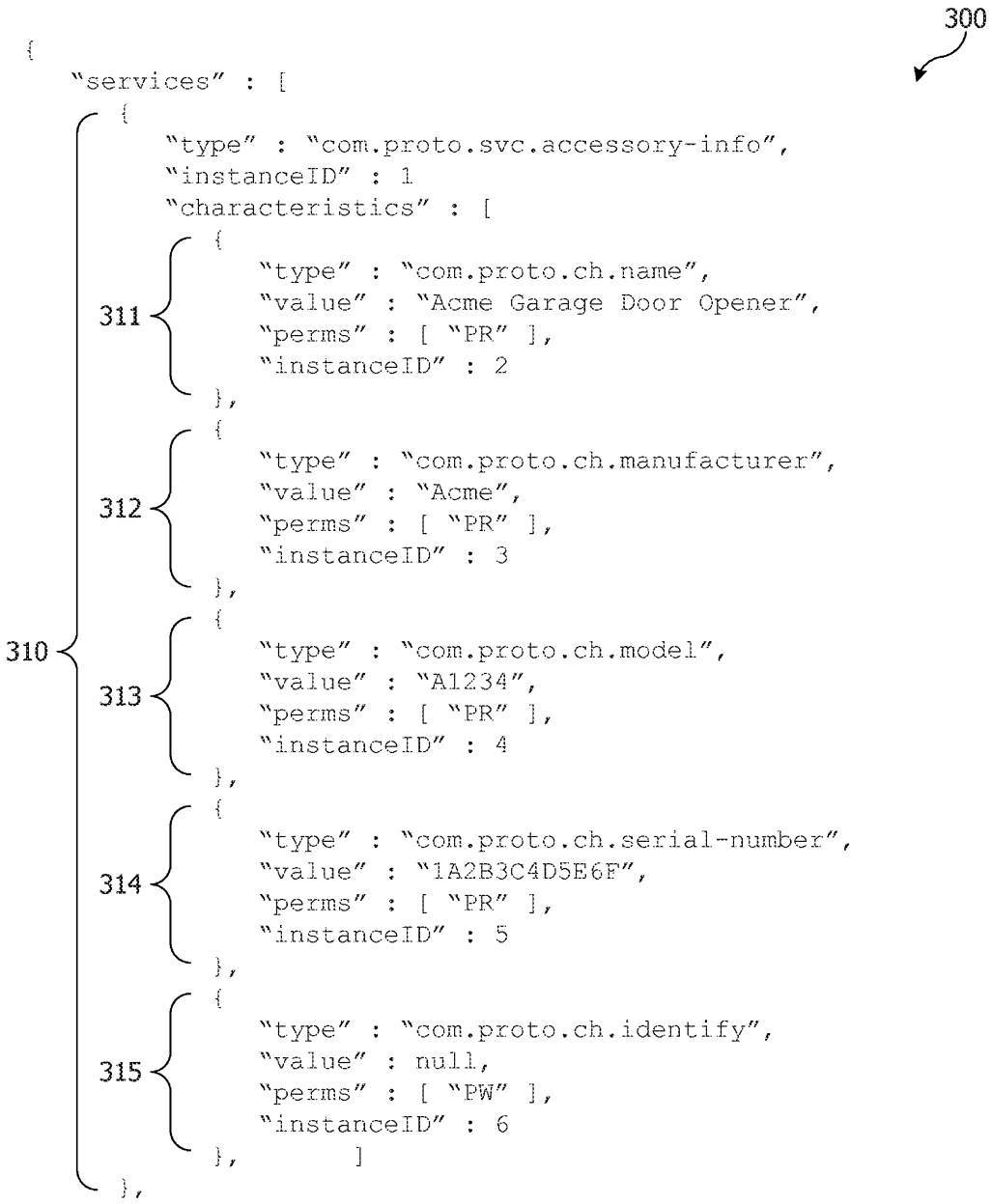
도면2i

서비스	속성	값
액세서리 정보 261	타입	com.proto.svc.accessory-info
	필수 특성	com.proto.ch.identify
		com.proto.ch.manufacturer
		com.proto.ch.model
		com.proto.ch.name
		com.proto.ch.serial-number
	선택적 특성	com.proto.ch.firmware-revision
		com.proto.ch.hardware-revision
		com.proto.ch.software-revision

도면2j

특성	속성	값
식별 271	타입	com.proto.ch.identify
	허가들	Paired Write
	포맷	<boolean>
제조사 이름 272	타입	com.proto.ch.manufacturer
	허가들	Paired Read
	포맷	<string>
모델 이름 273	타입	com.proto.ch.model
	허가들	Paired Read
	포맷	<string>
일련번호 274	타입	com.proto.ch.serial-number
	허가들	Paired Read
	포맷	<string>
펌웨어 개정 275	타입	com.proto.ch.firmware-rev
	허가들	Paired Read
	포맷	<string>
하드웨어 개정 276	타입	com.proto.ch.hardware-rev
	허가들	Paired Read
	포맷	<string>
소프트웨어 개정 277	타입	com.proto.ch.software-rev
	허가들	Paired Read
	포맷	<string>

도면3a

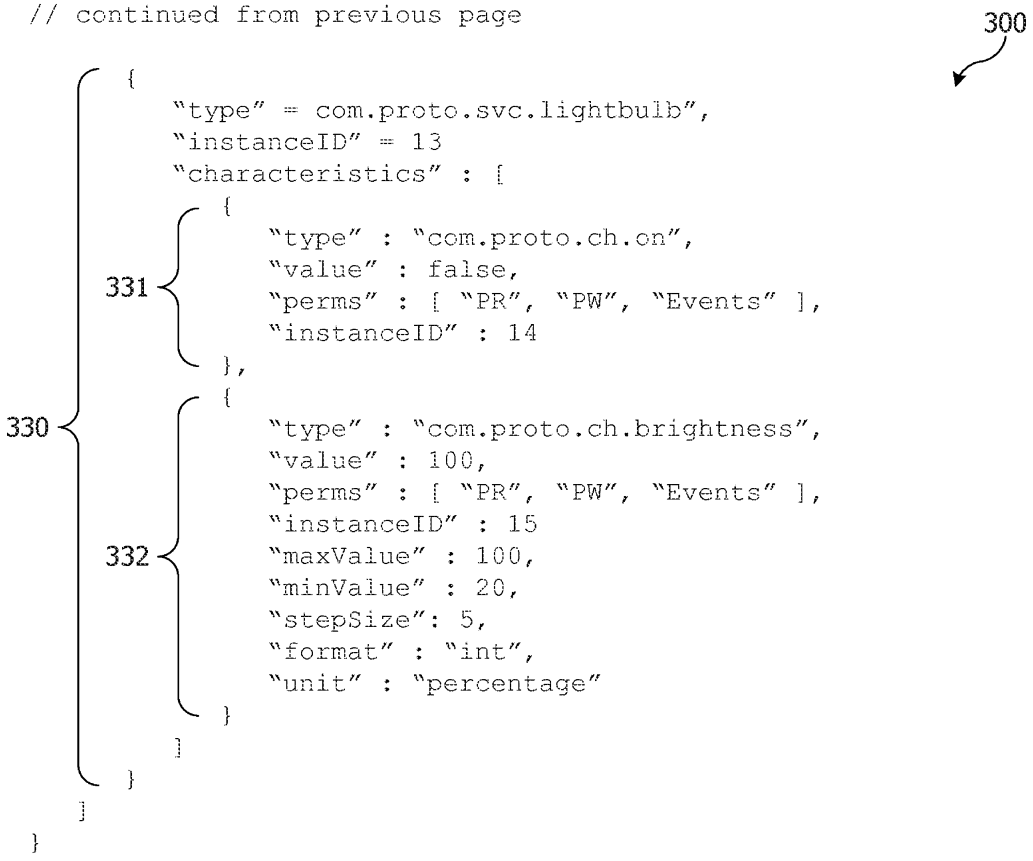


// continued next page

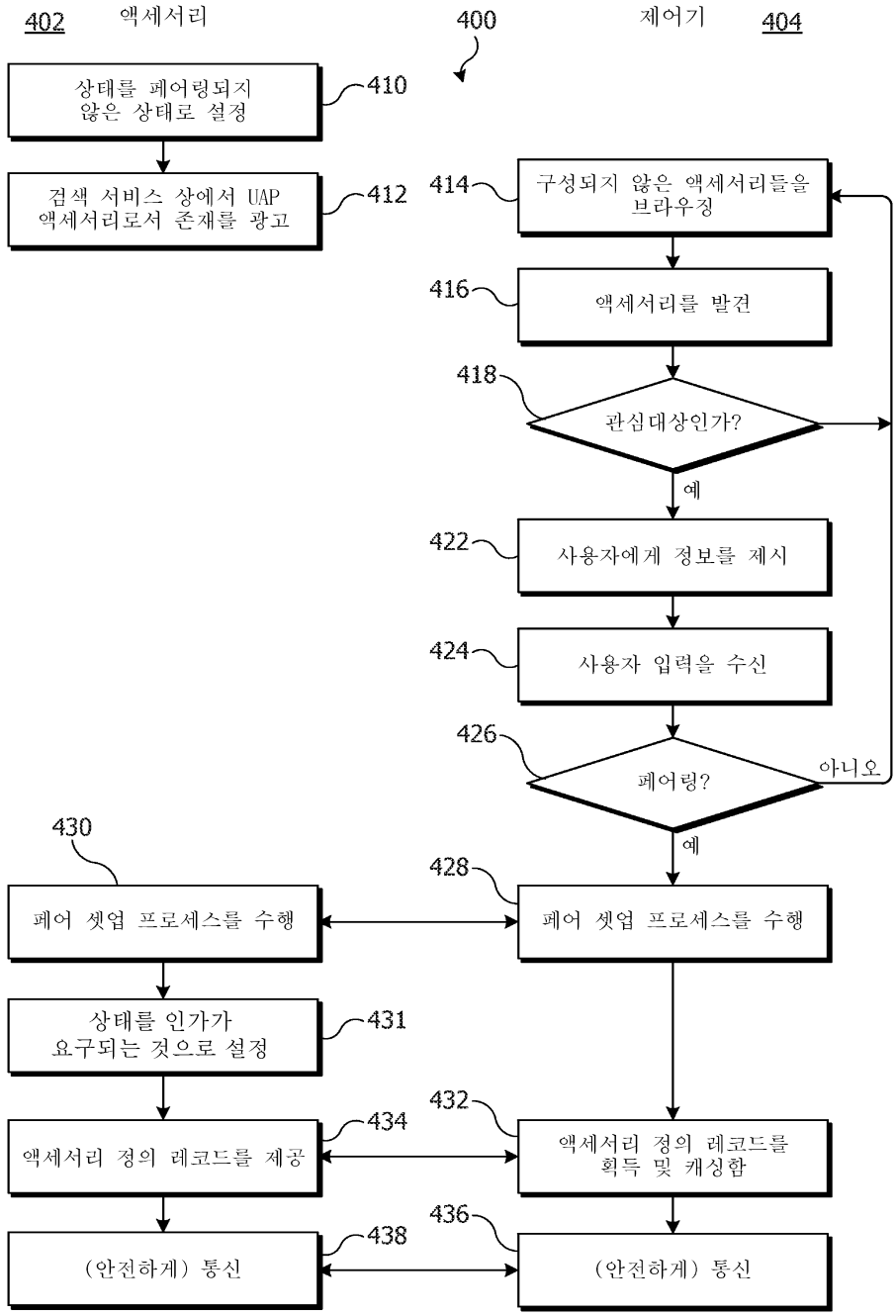
도면3b


// continued from previous page
300
{
"type" : "com.proto.svc.garage-door-opener",
"instanceID" : 7,
"characteristics" : [
{
"type" : "com.proto.ch.door-state.current",
"value" : 2,
"perms" : ["PR", "Events"],
"instanceID" : 8
},
321
{
"type" : "com.proto.ch.door-state.target",
"value" : 0,
"perms" : ["PR", "PW", "Events"],
"instanceID" : 9
},
322
{
"type" : "com.proto.ch.obstruction-detected",
"value" : true,
"perms" : ["PR", "Events"],
"instanceID" : 10
},
323
{
"type" : "com.proto.ch.lock-mech.current",
"value" : true,
"perms" : ["PR", "Events"],
"instanceID" : 11
},
324
{
"type" : "com.proto.ch.lock-mech.target",
"value" : true,
"perms" : ["PR", "PW", "Events"],
"instanceID" : 12
}
325
]
},
320
// continued next page

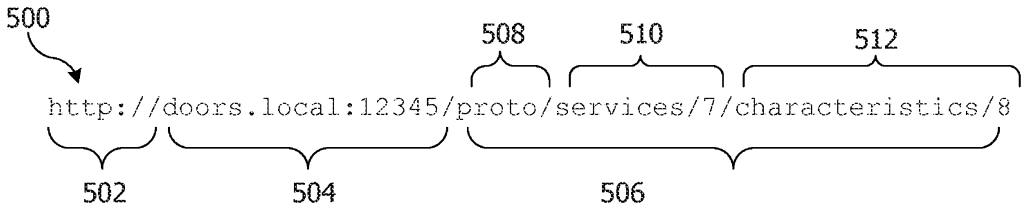
도면3c


// continued from previous page
300
{
"type" = com.proto.svc.lightbulb",
"instanceID" = 13
"characteristics" : [
{
"type" : "com.proto.ch.on",
"value" : false,
"perms" : ["PR", "PW", "Events"],
"instanceID" : 14
},
{
"type" : "com.proto.ch.brightness",
"value" : 100,
"perms" : ["PR", "PW", "Events"],
"instanceID" : 15
"maxValue" : 100,
"minValue" : 20,
"stepSize": 5,
"format" : "int",
"unit" : "percentage"
}
]
}
]
}
330
331
332

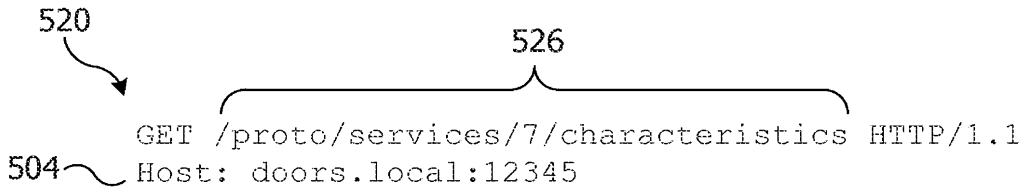
도면4


402
액세서리
400
제어기
404
상태를 페어링되지 않은 상태로 설정
410
검색 서비스 상에서 UAP 액세서리로서 존재를 광고
412
414
구성되지 않은 액세서리들을 브라우징
416
액세서리를 발견
418
관심대상인가?
예
아니오
422
사용자에게 정보를 제시
424
사용자 입력을 수신
426
페어링?
예
아니오
428
페어 셋업 프로세스를 수행
430
페어 셋업 프로세스를 수행
상태를 인가가 요구되는 것으로 설정
431
432
액세서리 정의 레코드를 획득 및 캐싱함
434
액세서리 정의 레코드를 제공
436
(안전하게) 통신
438
(안전하게) 통신

도면5a

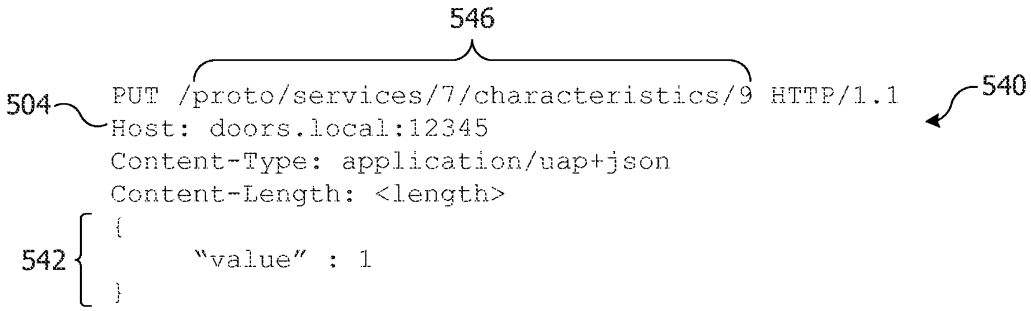

500
508
510
512
http://doors.local:12345/proto/services/7/characteristics/8
502
504
506

도면5b

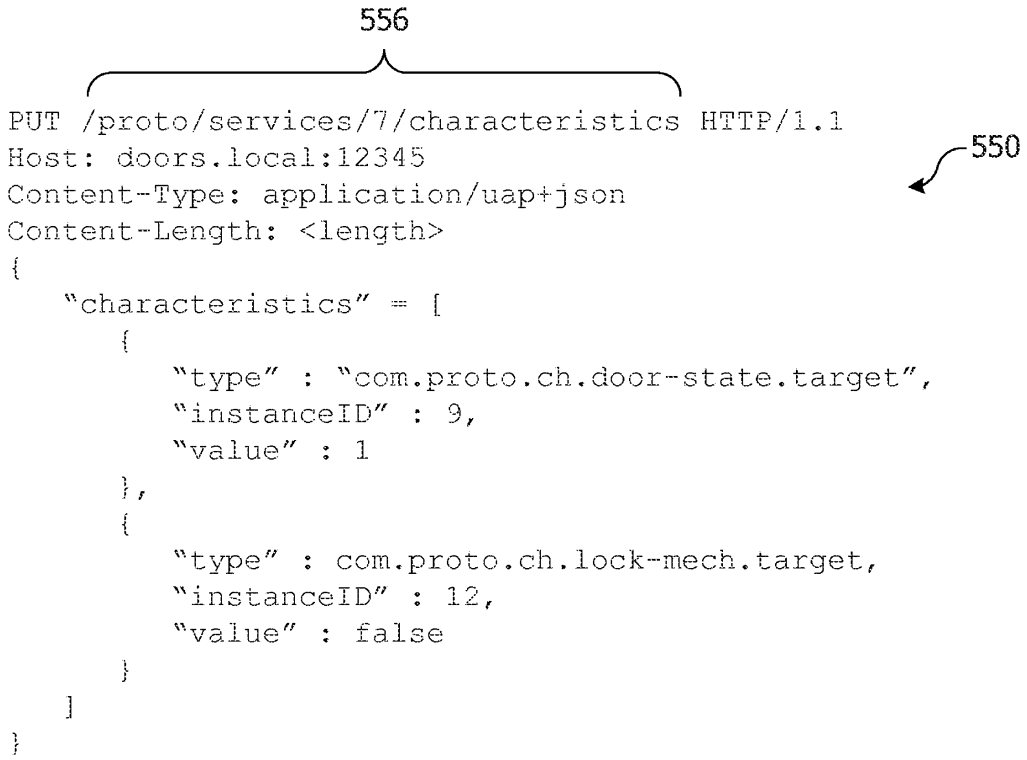


도면5c

도면5d



도면5e


556
PUT /proto/services/7/characteristics HTTP/1.1
Host: doors.local:12345
Content-Type: application/uap+json
Content-Length: <length>
550
{
"characteristics" = [
{
"type" : "com.proto.ch.door-state.target",
"instanceID" : 9,
"value" : 1
},
{
"type" : com.proto.ch.lock-mech.target,
"instanceID" : 12,
"value" : false
}
]
}

도면5f


HTTP/1.1 200 OK
Content-Type: application/uap+json
Content-Length: <length>
560
{
"characteristics" = [
{
"type" : com.proto.ch.door-state.target,
"value" : 0
"instanceID" : 0,
"response: : {
"developerMessage" : "No error occurred.",
"errorCode" : 0,
"moreInfo" : null
}
},
{
"type" : com.proto.ch.lock-mech.target,
"value" : true
"instanceID" : 1,
"response: : {
"developerMessage" : "No error occurred.",
"errorCode" : 0,
"moreInfo" : null
}
}
]
}
562
563
564
565
566

도면5g

```
{
   "type" : com.acme.ch.port.open,
   "perms" : [ "PW" ]
   "value" : null,
   "instanceID" : 33
}
```

571

도면5h

572

```
PUT /proto/services/22/characteristics/33 HTTP/1.1
Host: ipcamera.local:12345
Content-Type: application/uap+json
Content-Length: <length>
{
   "value" : true
}
```

도면5i

573

```
   HTTP/1.1 200 OK
   Content-Type: application/uap+json
   Content-Length: <length>
   {
      "value" :
         {
            "port" : 45678,
            "status-code" : 0
         }
   }
```

574

도면5j

```
HTTP/1.1 200 OK
Content-Type: application/uap+json
Content-Length: <length>
{
   "value" :
      {
         "transactionID" : nnnnnnnnnnnn,
         "transactionDuration" : 5
      }
}
```

575

576

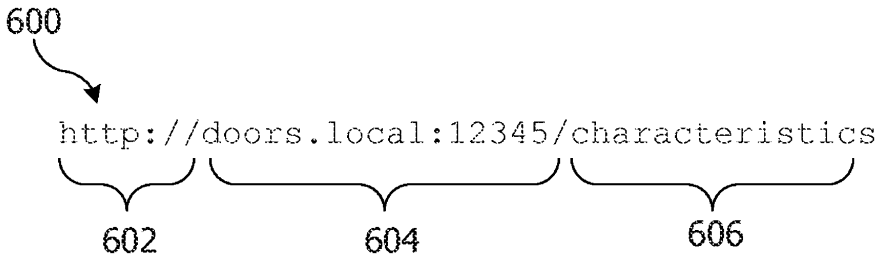
577

도면5k

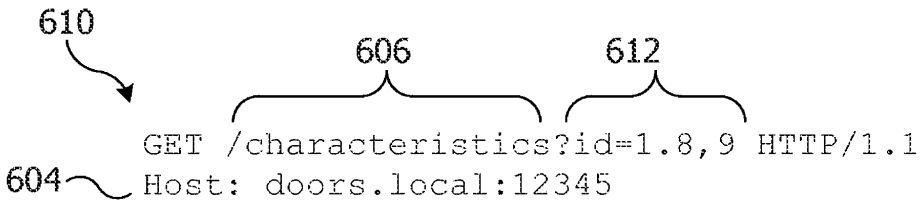
```
GET /proto/services/22/characteristics/33 HTTP/1.1
Host: ipcamera.local:12345
Content-Type: application/json
Content-Length: <length>
{
   "value" :
      {
         "transactionID" : nnnnnnnnnnnn
      }
}
```

578

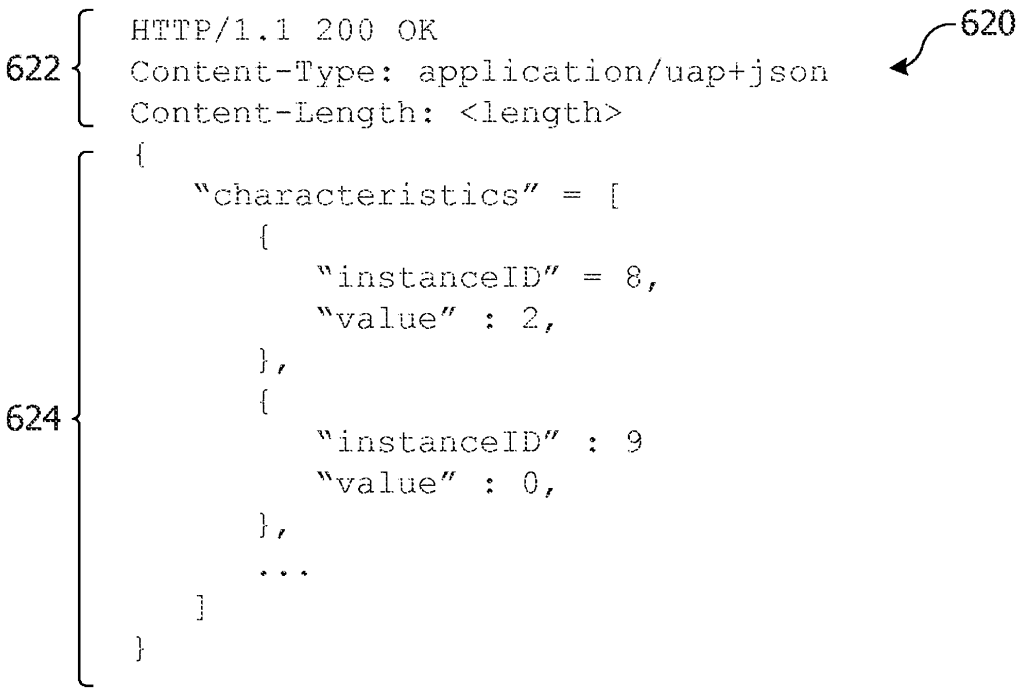
도면6a



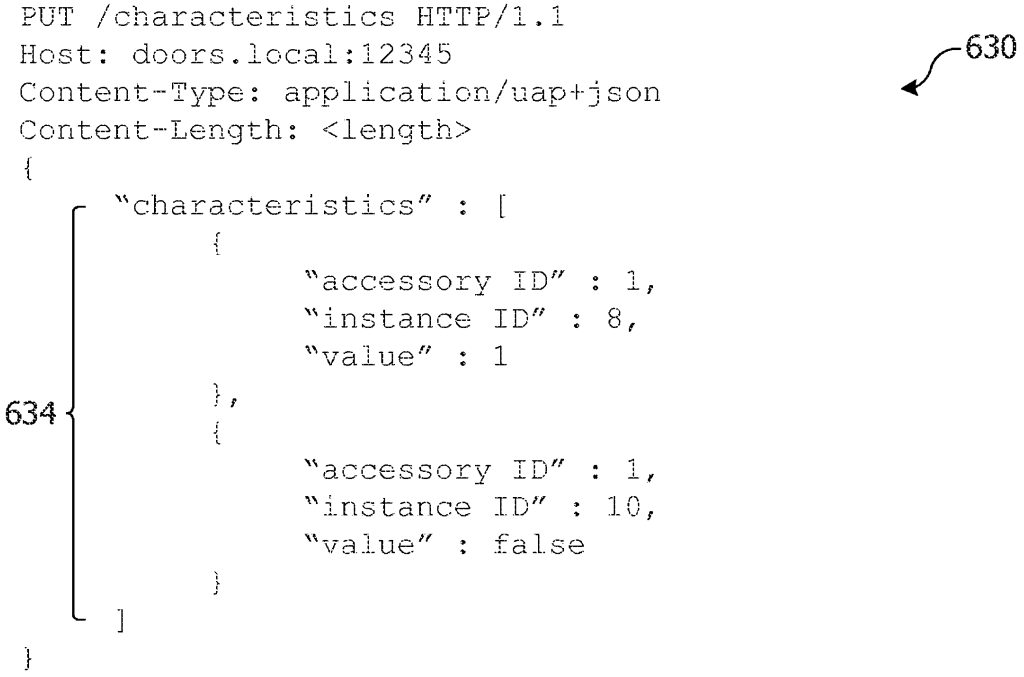
도면6b



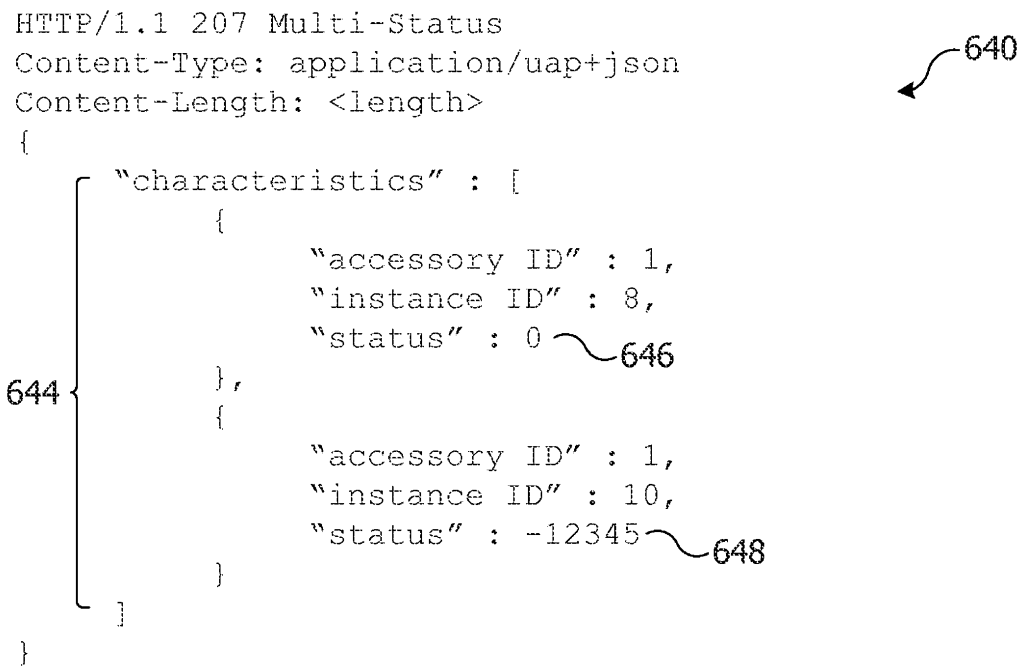
도면6c


620
622
HTTP/1.1 200 OK
Content-Type: application/uap+json
Content-Length: <length>
624
{
"characteristics" = [
{
"instanceID" = 8,
"value" : 2,
},
{
"instanceID" : 9
"value" : 0,
},
...
]
}

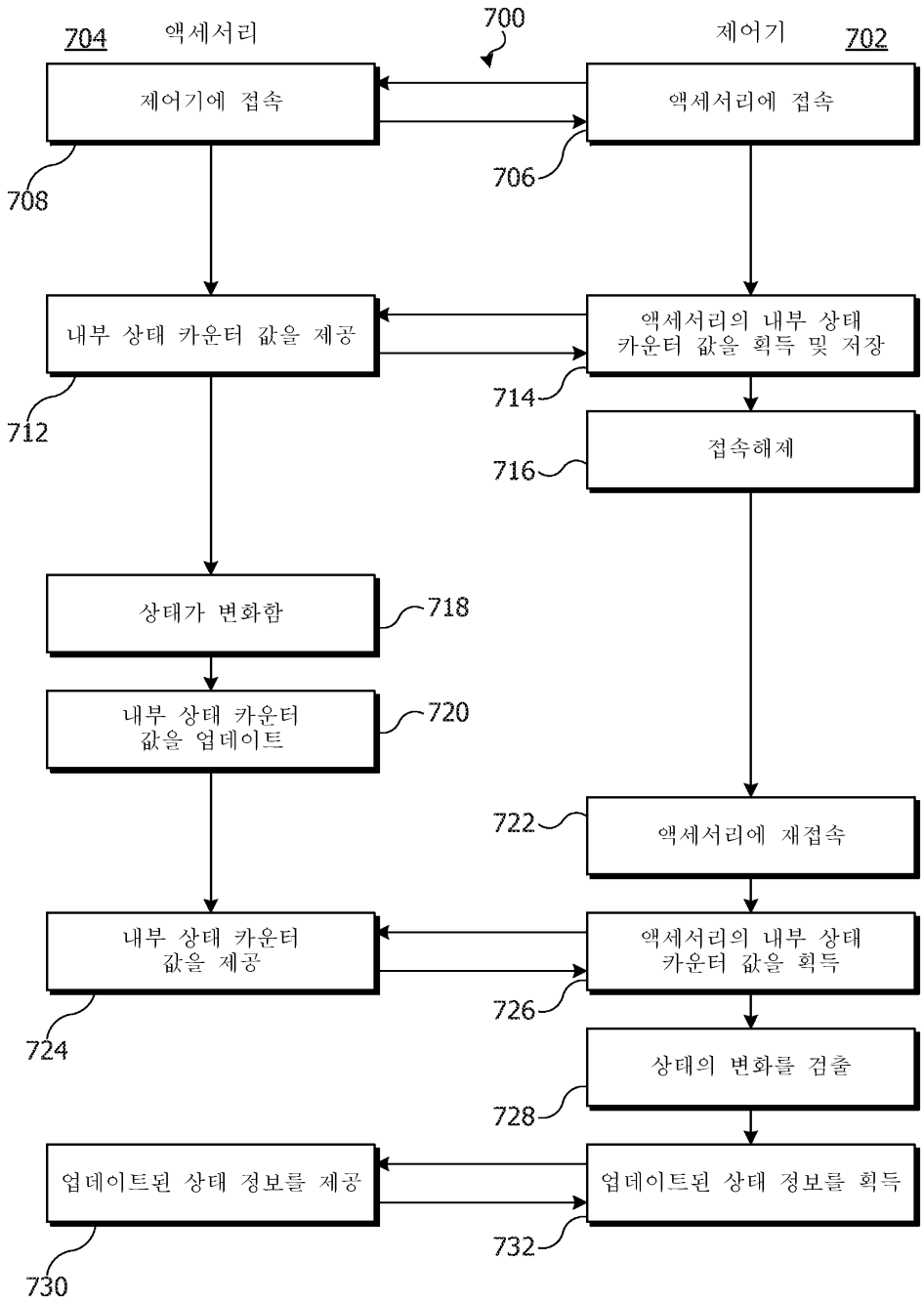
도면6d


630
PUT /characteristics HTTP/1.1
Host: doors.local:12345
Content-Type: application/uap+json
Content-Length: <length>
{
"characteristics" : [
634
{
"accessory ID" : 1,
"instance ID" : 8,
"value" : 1
},
{
"accessory ID" : 1,
"instance ID" : 10,
"value" : false
}
]
}

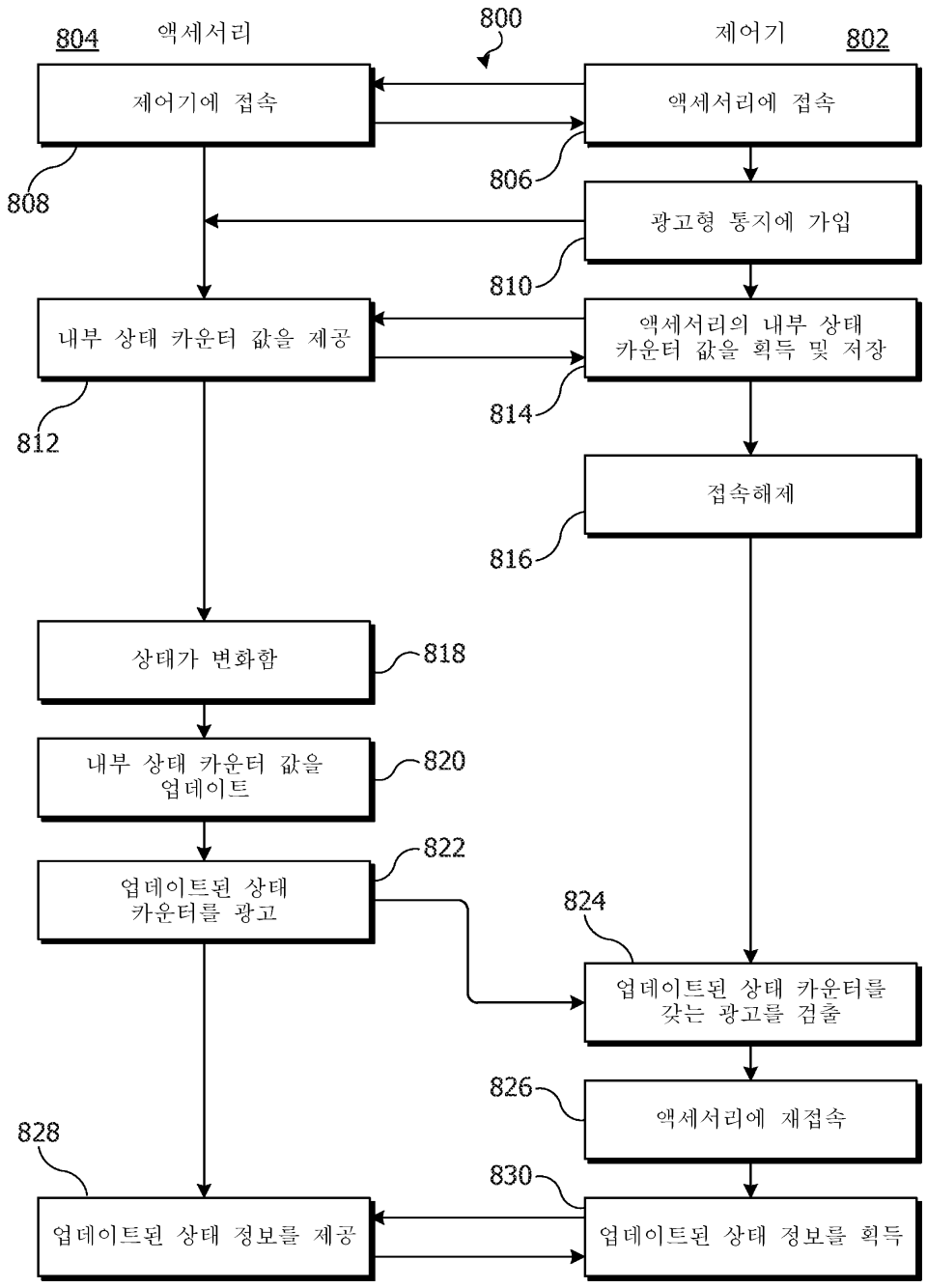
도면6e


HTTP/1.1 207 Multi-Status
Content-Type: application/uap+json
Content-Length: <length>
{
"characteristics" : [
{
"accessory ID" : 1,
"instance ID" : 8,
"status" : 0
},
{
"accessory ID" : 1,
"instance ID" : 10,
"status" : -12345
}
]
}
640
644
646
648

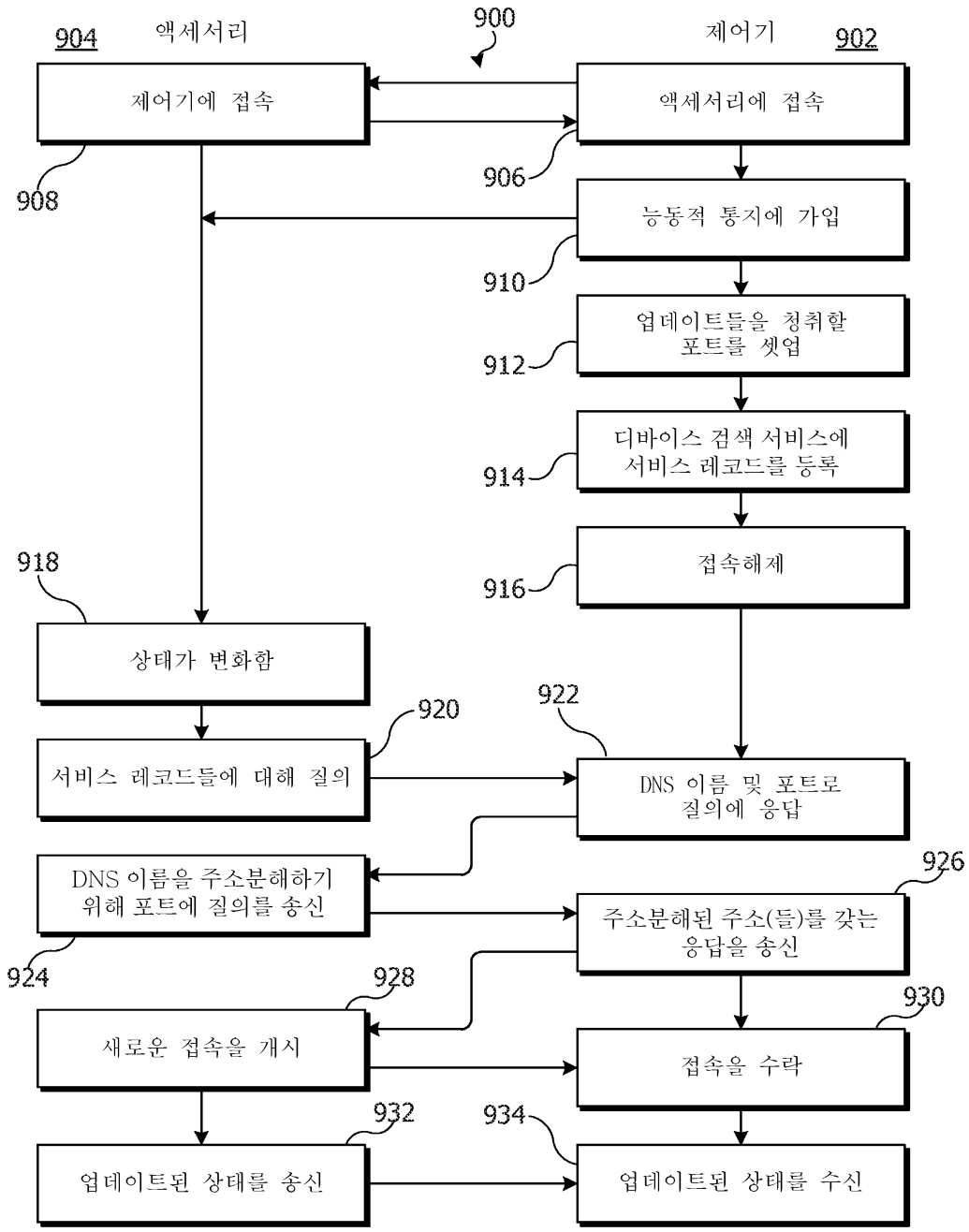
도면7


704
액세서리
700
제어기
702
제어기에 접속
액세서리에 접속
708
706
내부 상태 카운터 값을 제공
액세서리의 내부 상태 카운터 값을 획득 및 저장
712
714
접속해제
716
상태가 변화함
718
내부 상태 카운터 값을 업데이트
720
722
액세서리에 재접속
내부 상태 카운터 값을 제공
액세서리의 내부 상태 카운터 값을 획득
724
726
상태의 변화를 검출
728
업데이트된 상태 정보를 제공
업데이트된 상태 정보를 획득
730
732

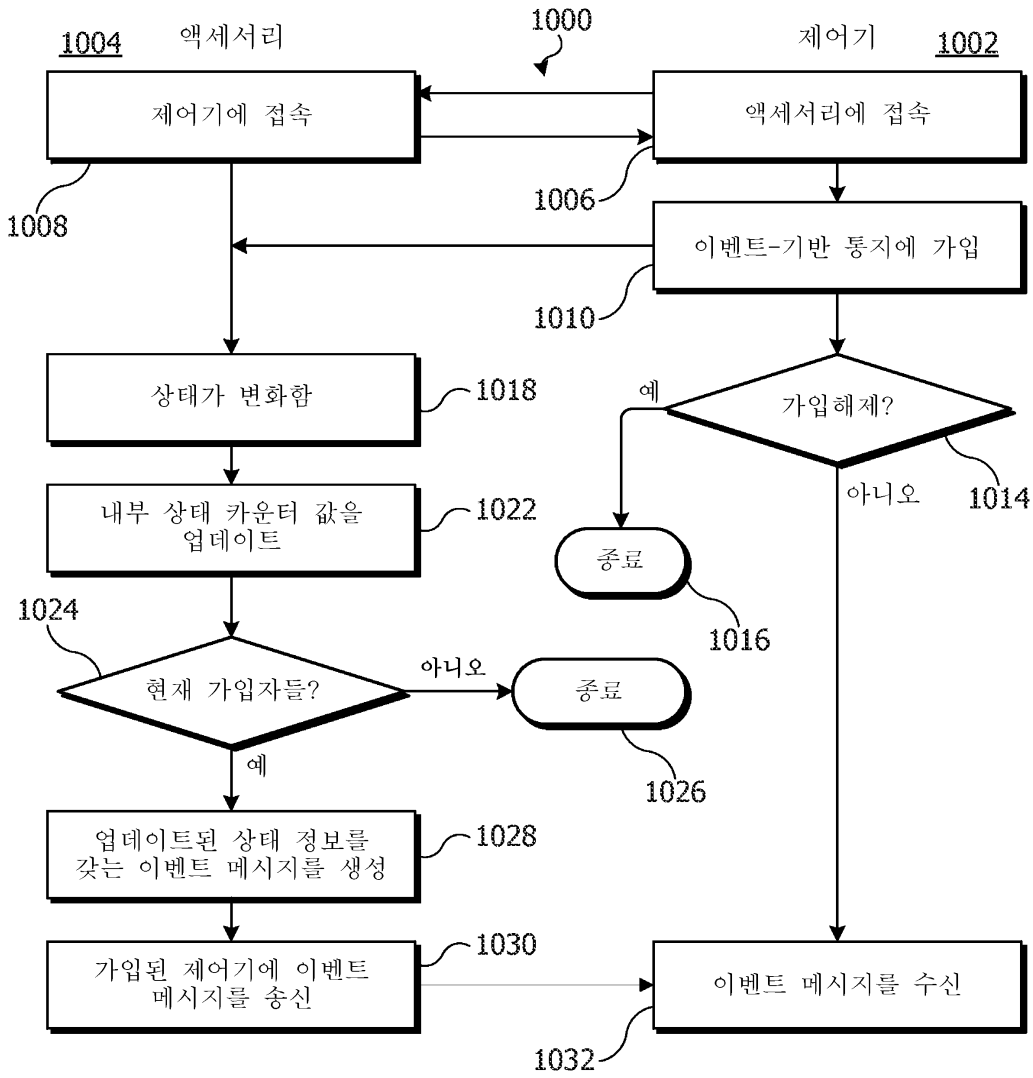
도면8


800
804
액세서리
제어기
802
제어기에 접속
액세서리에 접속
806
808
광고형 통지에 가입
810
내부 상태 카운터 값을 제공
액세서리의 내부 상태 카운터 값을 획득 및 저장
812
814
접속해제
816
상태가 변화함
818
내부 상태 카운터 값을 업데이트
820
822
업데이트된 상태 카운터를 광고
824
업데이트된 상태 카운터를 갖는 광고를 검출
826
액세서리에 재접속
828
830
업데이트된 상태 정보를 제공
업데이트된 상태 정보를 획득

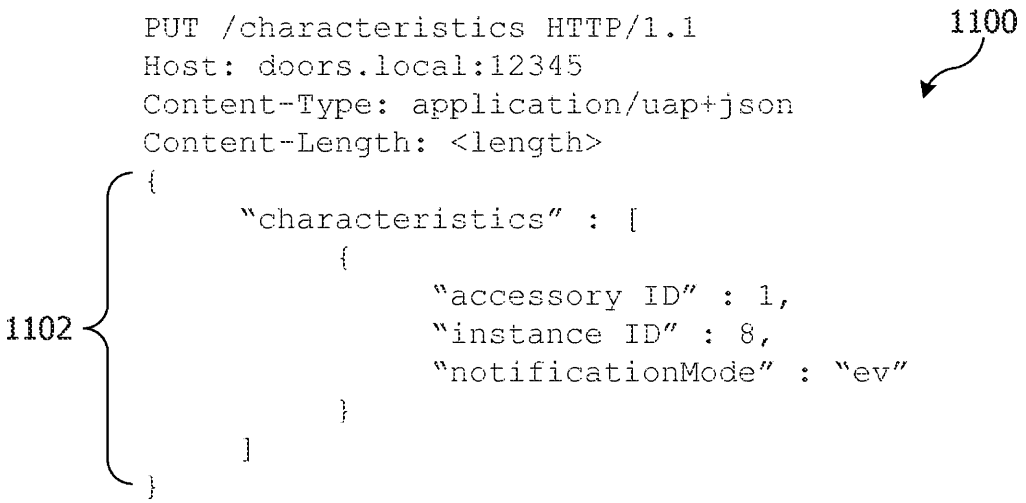
도면9


900
904
액세서리
제어기
902
제어기에 접속
액세서리에 접속
906
908
능동적 통지에 가입
910
업데이트들을 청취할 포트를 셋업
912
디바이스 검색 서비스에 서비스 레코드를 등록
914
접속해제
916
918
상태가 변화함
920
922
서비스 레코드들에 대해 질의
DNS 이름 및 포트로 질의에 응답
DNS 이름을 주소분해하기 위해 포트에 질의를 송신
926
주소분해된 주소(들)를 갖는 응답을 송신
924
928
930
새로운 접속을 개시
접속을 수락
932
934
업데이트된 상태를 송신
업데이트된 상태를 수신

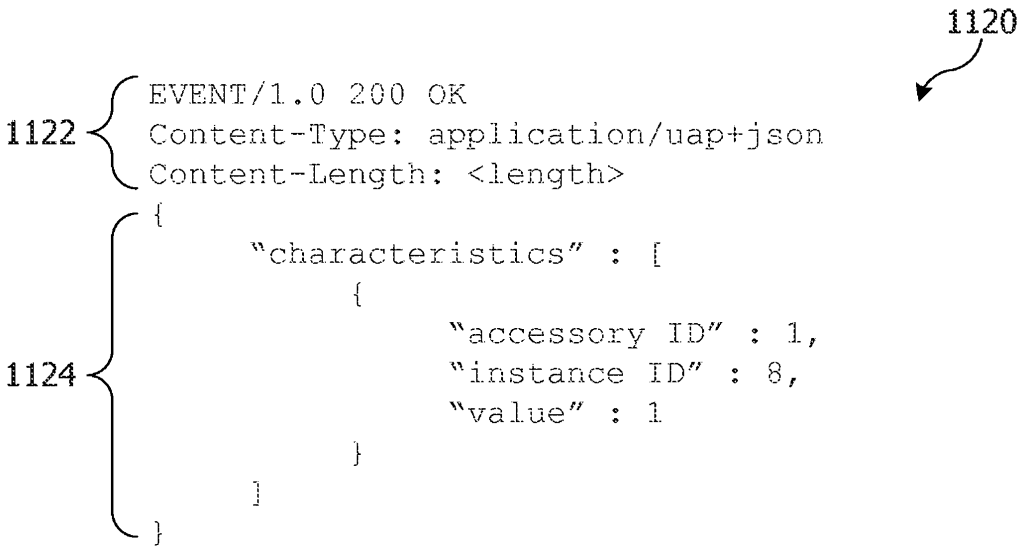
도면10


1000
1004
액세서리
제어기
1002
제어기에 접속
액세서리에 접속
1008
1006
이벤트-기반 통지에 가입
1010
상태가 변화함
1018
예
가입해제?
아니오
1014
내부 상태 카운터 값을 업데이트
1022
종료
1016
1024
현재 가입자들?
아니오
종료
예
1026
업데이트된 상태 정보를 갖는 이벤트 메시지를 생성
1028
가입된 제어기에 이벤트 메시지를 송신
1030
이벤트 메시지를 수신
1032

도면11a


1100
PUT /characteristics HTTP/1.1
Host: doors.local:12345
Content-Type: application/uap+json
Content-Length: <length>
1102
{
"characteristics" : [
{
"accessory ID" : 1,
"instance ID" : 8,
"notificationMode" : "ev"
}
]
}

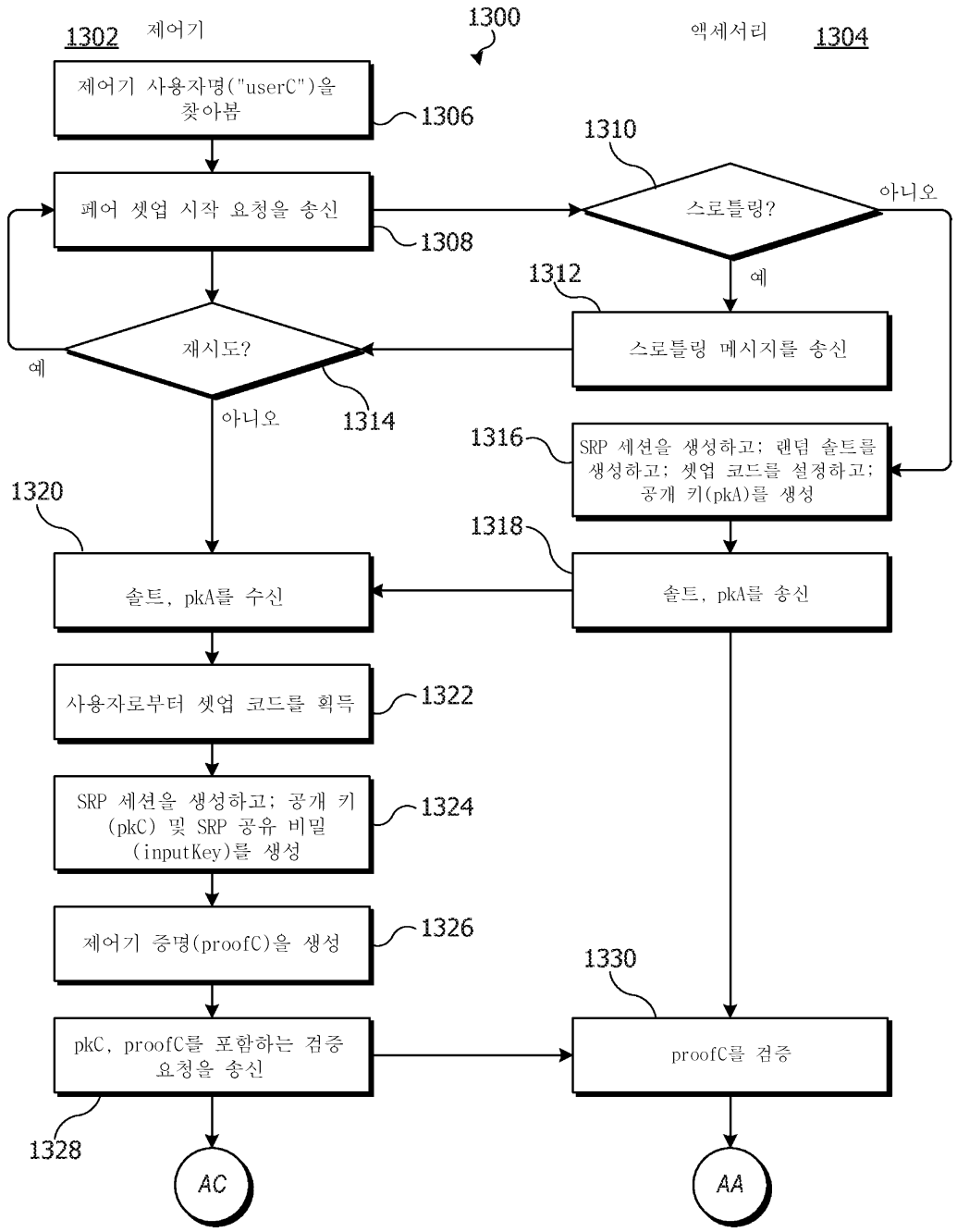
도면11b



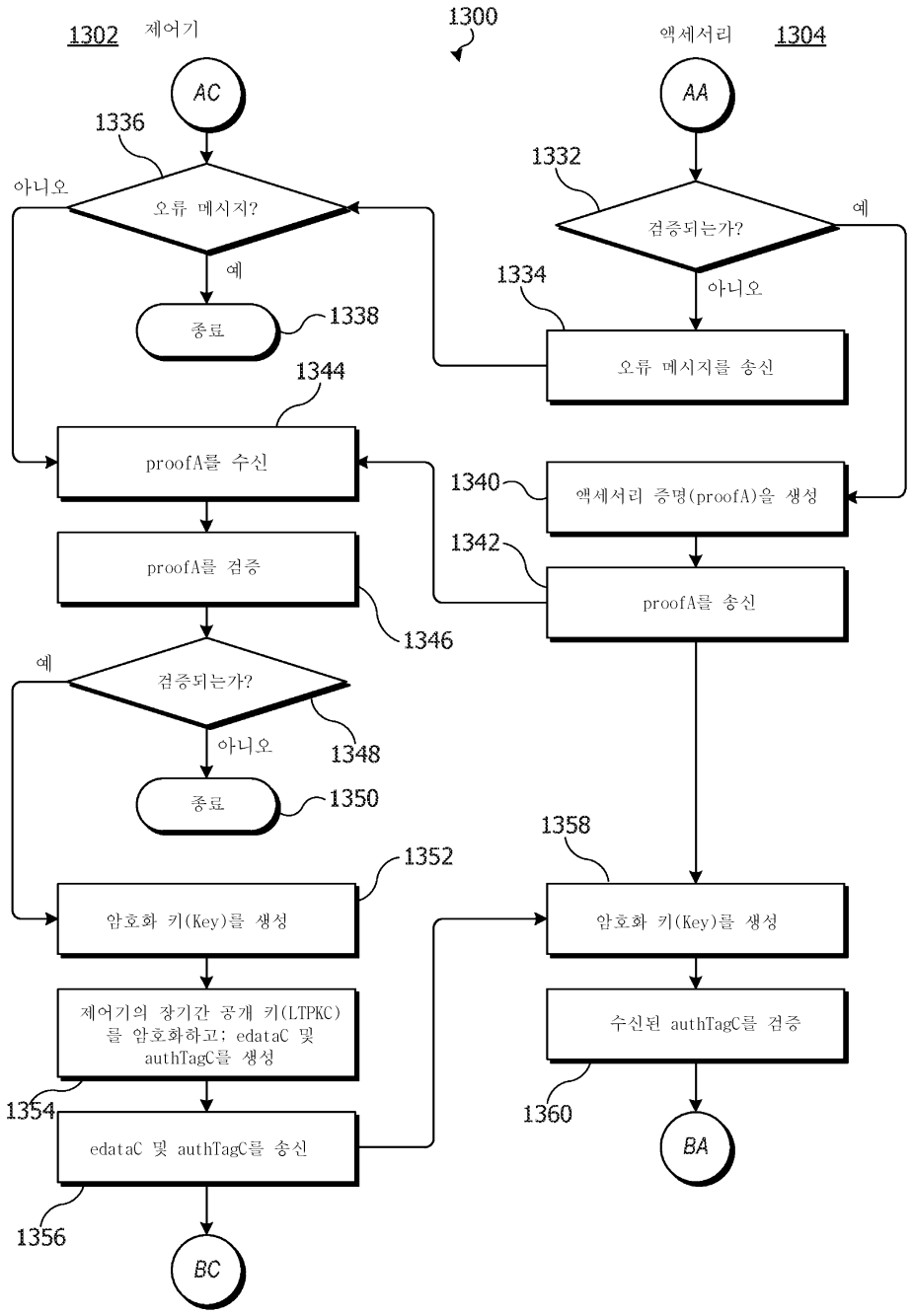
도면12

특성	속성	값
페어링 상태 요청 1201	포맷	<tlv>
	타입	com.proto.pairing.state-request
	허가들	UR, UW, PR, PW
특징 플래그들 1202	포맷	<int>
	타입	com.proto.pairing-features
	허가들	UR, PR
페어링 현재 상태 1203	포맷	<tlv>
	타입	com.proto.pairing.state-current
	허가들	UR, PR
페어링 목록 1204	포맷	<tlv>
	타입	com.proto.pairing-list
	허가들	PR
페어링 ID 1205	포맷	<string>
	타입	com.proto.pairing-identifier
	허가들	UR, PR

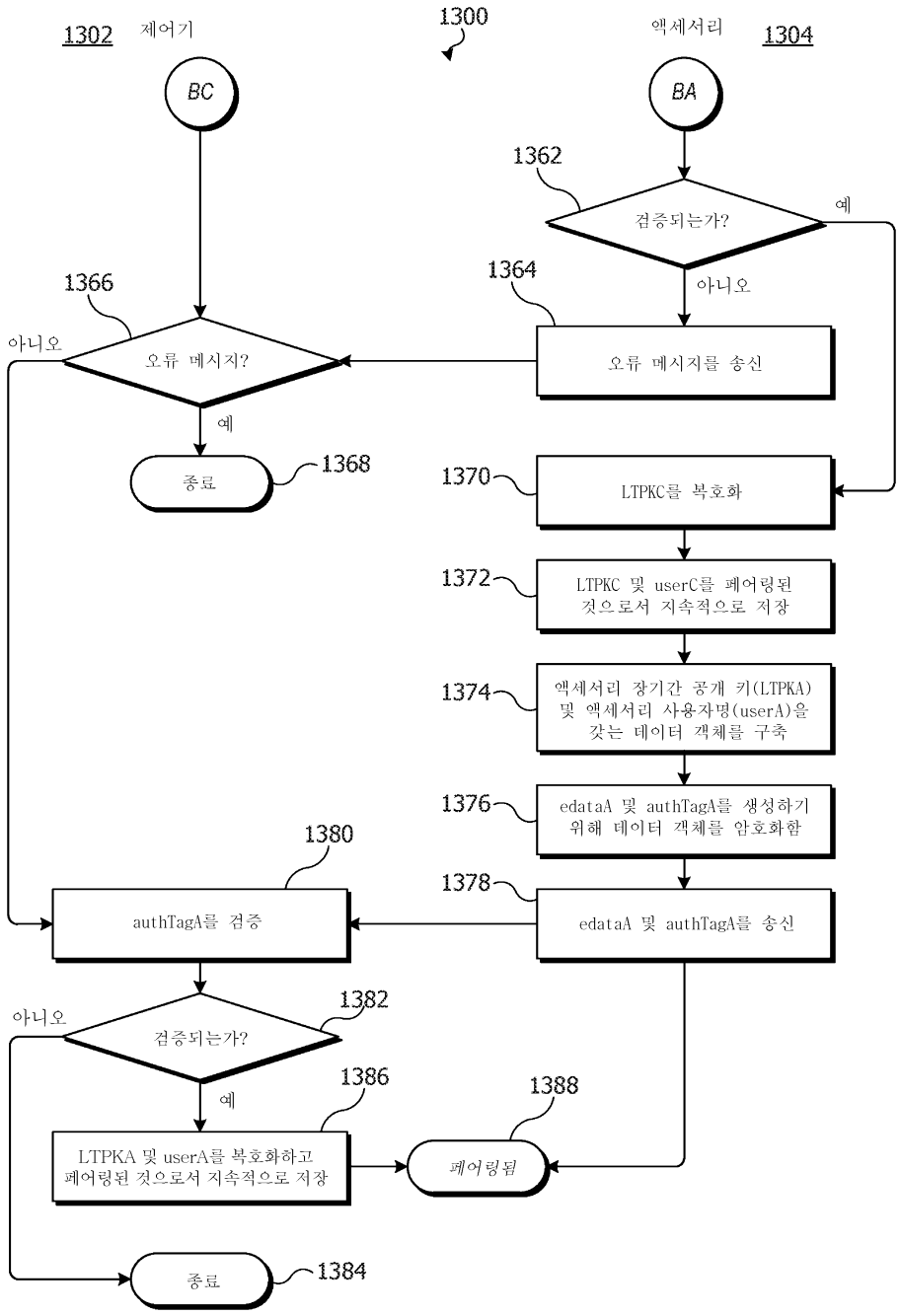
도면13a


1300
1302
제어기
액세서리
1304
제어기 사용자명("userC")을 찾아봄
1306
페어 셋업 시작 요청을 송신
1308
1310
스로틀링?
아니오
예
1312
스로틀링 메시지를 송신
재시도?
예
1314
아니오
1316
SRP 세션을 생성하고; 랜덤 솔트를 생성하고; 셋업 코드를 설정하고; 공개 키(pkA)를 생성
1318
솔트, pkA를 송신
1320
솔트, pkA를 수신
사용자로부터 셋업 코드를 획득
1322
SRP 세션을 생성하고; 공개 키(pkC) 및 SRP 공유 비밀(inputKey)를 생성
1324
제어기 증명(proofC)을 생성
1326
1330
pkC, proofC를 포함하는 검증 요청을 송신
proofC를 검증
1328
AC
AA

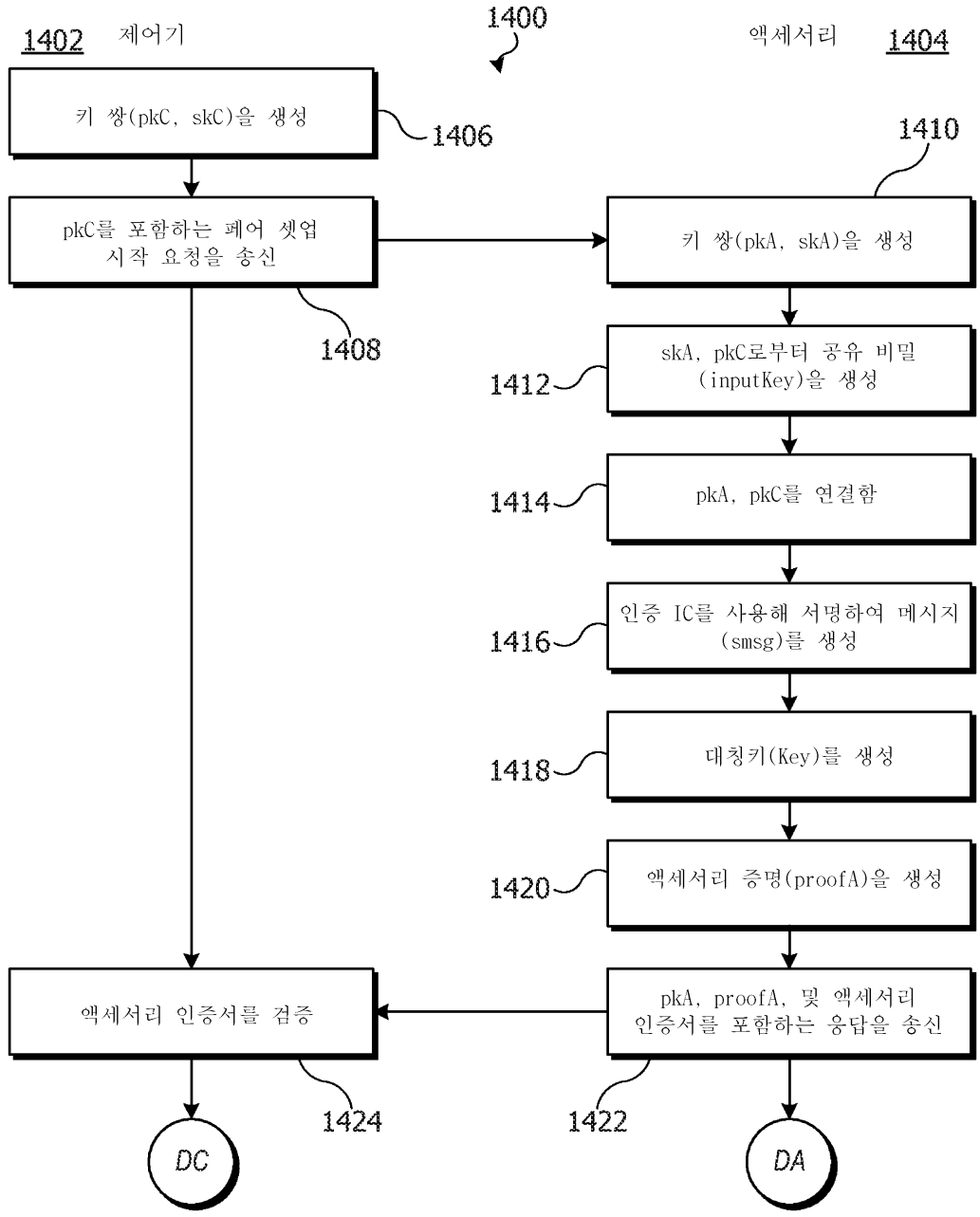
도면13b


1300
1302 제어기
액세서리 1304
AC
AA
1336
1332
오류 메시지?
검증되는가?
아니오
예
예
아니오
1338
1334
종료
오류 메시지를 송신
1344
proofA를 수신
1340
액세서리 증명(proofA)을 생성
proofA를 검증
1342
proofA를 송신
1346
예
검증되는가?
1348
아니오
종료
1350
1358
1352
암호화 키(Key)를 생성
암호화 키(Key)를 생성
제어기의 장기간 공개 키(LTPKC)를 암호화하고; edataC 및 authTagC를 생성
수신된 authTagC를 검증
1354
1360
edataC 및 authTagC를 송신
BA
1356
BC

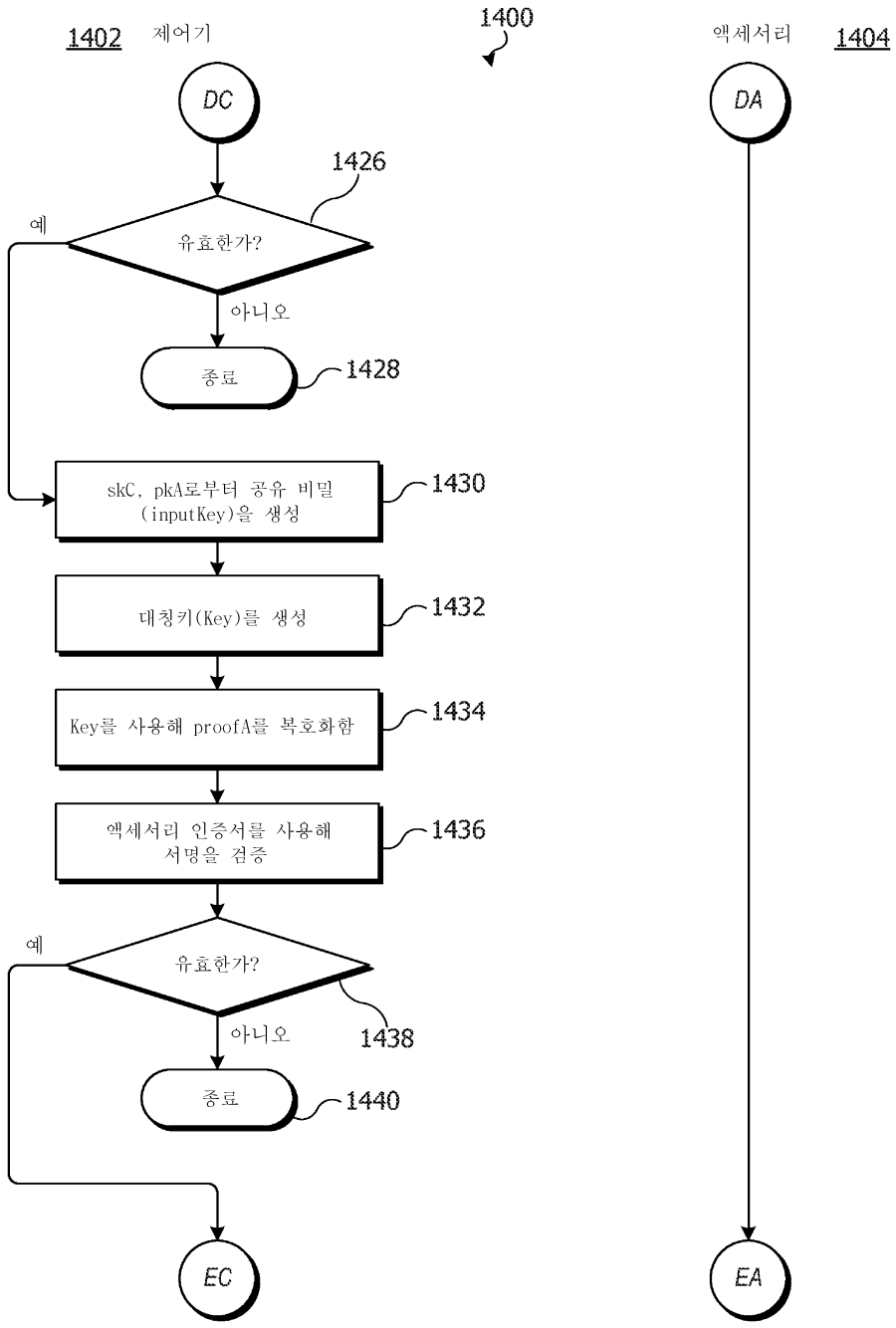
도면13c


1300
1302
제어기
액세서리
1304
BC
BA
1362
검증되는가?
예
아니오
1364
오류 메시지를 송신
1366
아니오
오류 메시지?
예
종료
1368
1370
LTPKC를 복호화
1372
LTPKC 및 userC를 페어링된 것으로서 지속적으로 저장
1374
액세서리 장기간 공개 키(LTPKA) 및 액세서리 사용자명(userA)을 갖는 데이터 객체를 구축
1376
edataA 및 authTagA를 생성하기 위해 데이터 객체를 암호화함
1378
edataA 및 authTagA를 송신
1380
authTagA를 검증
아니오
검증되는가?
1382
예
1386
LTPKA 및 userA를 복호화하고 페어링된 것으로서 지속적으로 저장
1388
페어링됨
종료
1384

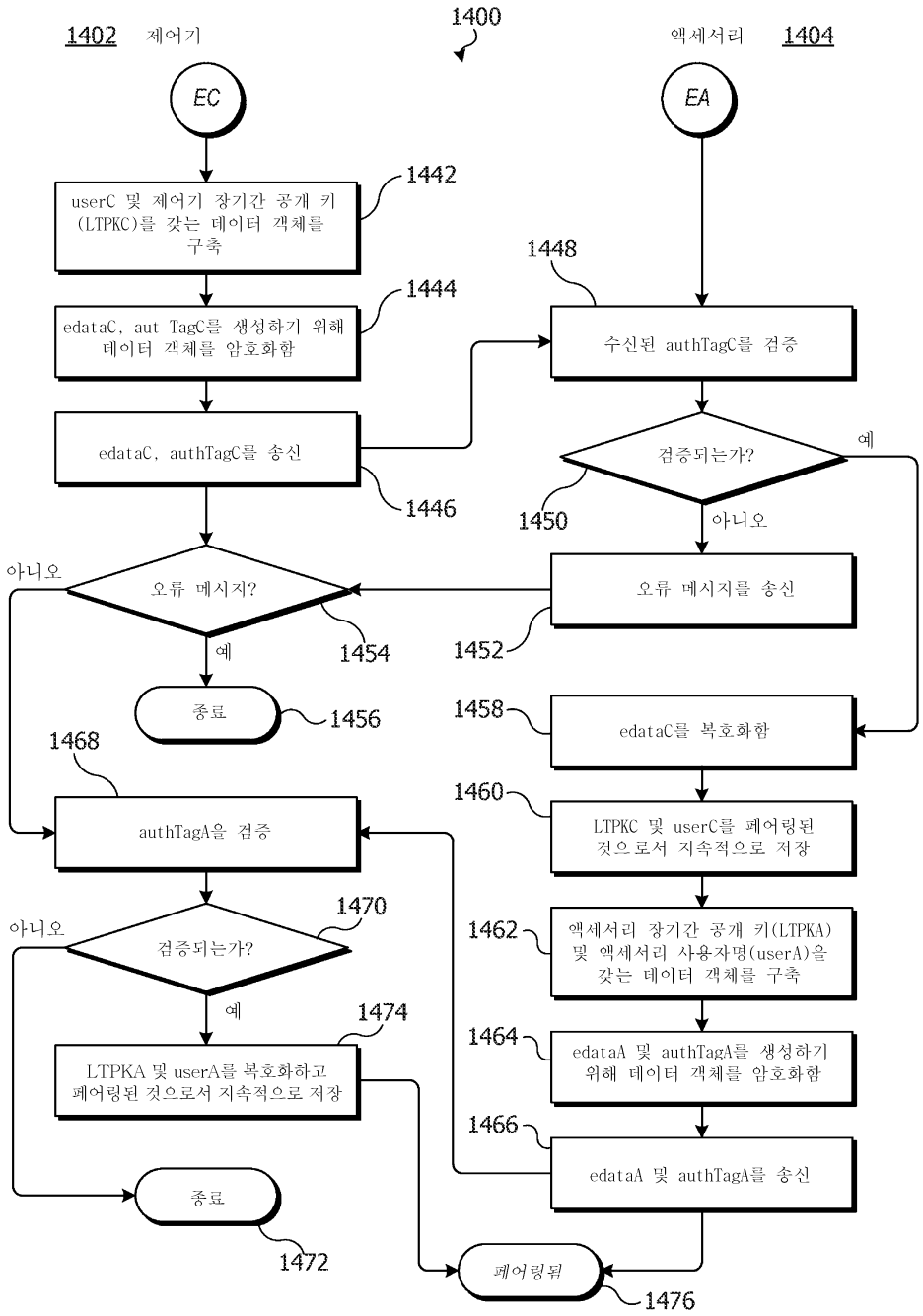
도면14a


1400
1402
제어기
액세서리
1404
키 쌍(pkC, skC)을 생성
1406
1410
pkC를 포함하는 페어 셋업 시작 요청을 송신
키 쌍(pkA, skA)을 생성
1408
skA, pkC로부터 공유 비밀(inputKey)을 생성
1412
pkA, pkC를 연결함
1414
인증 IC를 사용해 서명하여 메시지(smsg)를 생성
1416
대칭키(Key)를 생성
1418
액세서리 증명(proofA)을 생성
1420
액세서리 인증서를 검증
pkA, proofA, 및 액세서리 인증서를 포함하는 응답을 송신
1424
1422
DC
DA

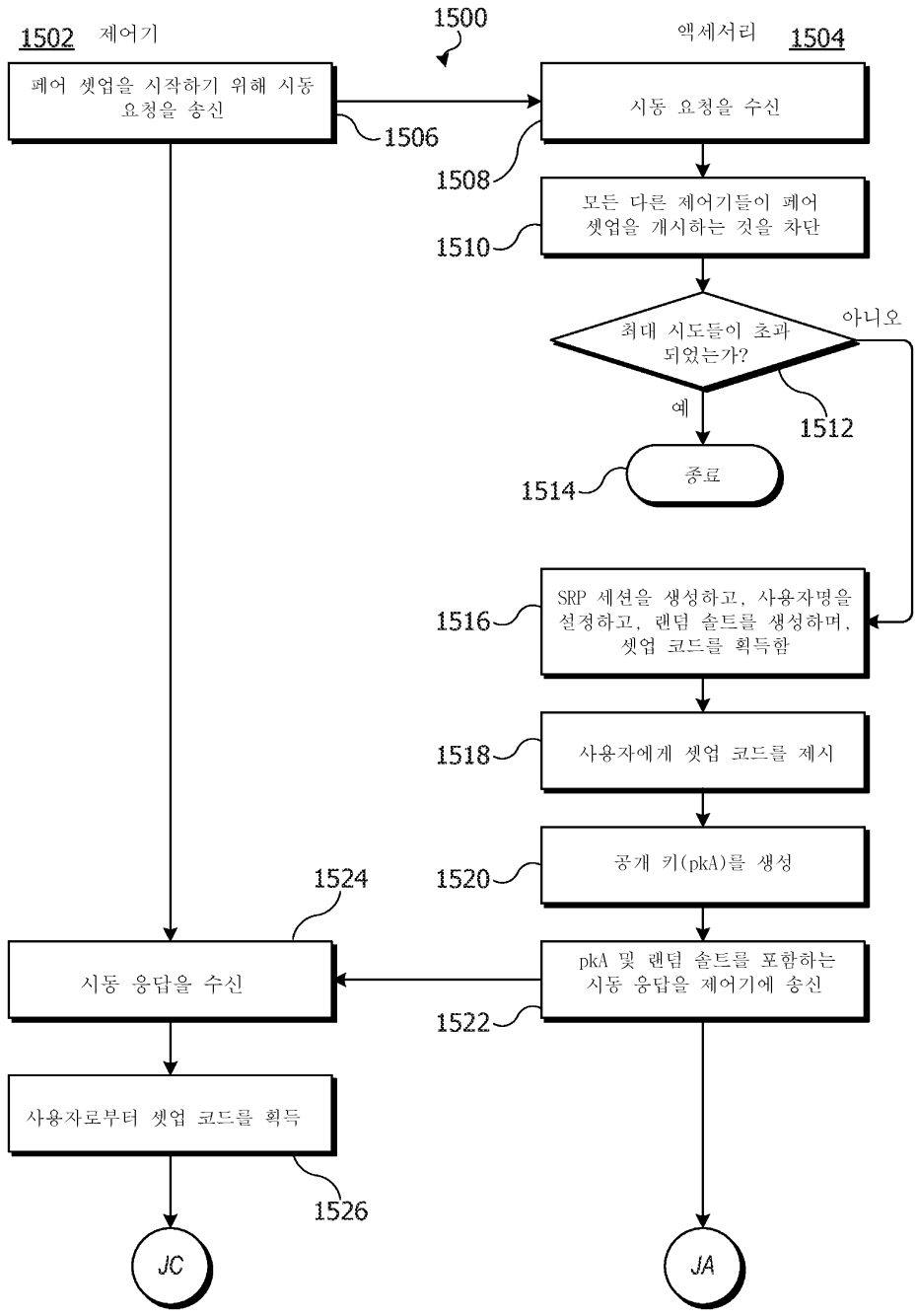
도면14b


1400
1402 제어기
액세서리 1404
DC
DA
1426
예
유효한가?
아니오
종료
1428
skC, pkA로부터 공유 비밀 (inputKey)을 생성
1430
대칭키(Key)를 생성
1432
Key를 사용해 proofA를 복호화함
1434
액세서리 인증서를 사용해 서명을 검증
1436
예
유효한가?
아니오
1438
종료
1440
EC
EA

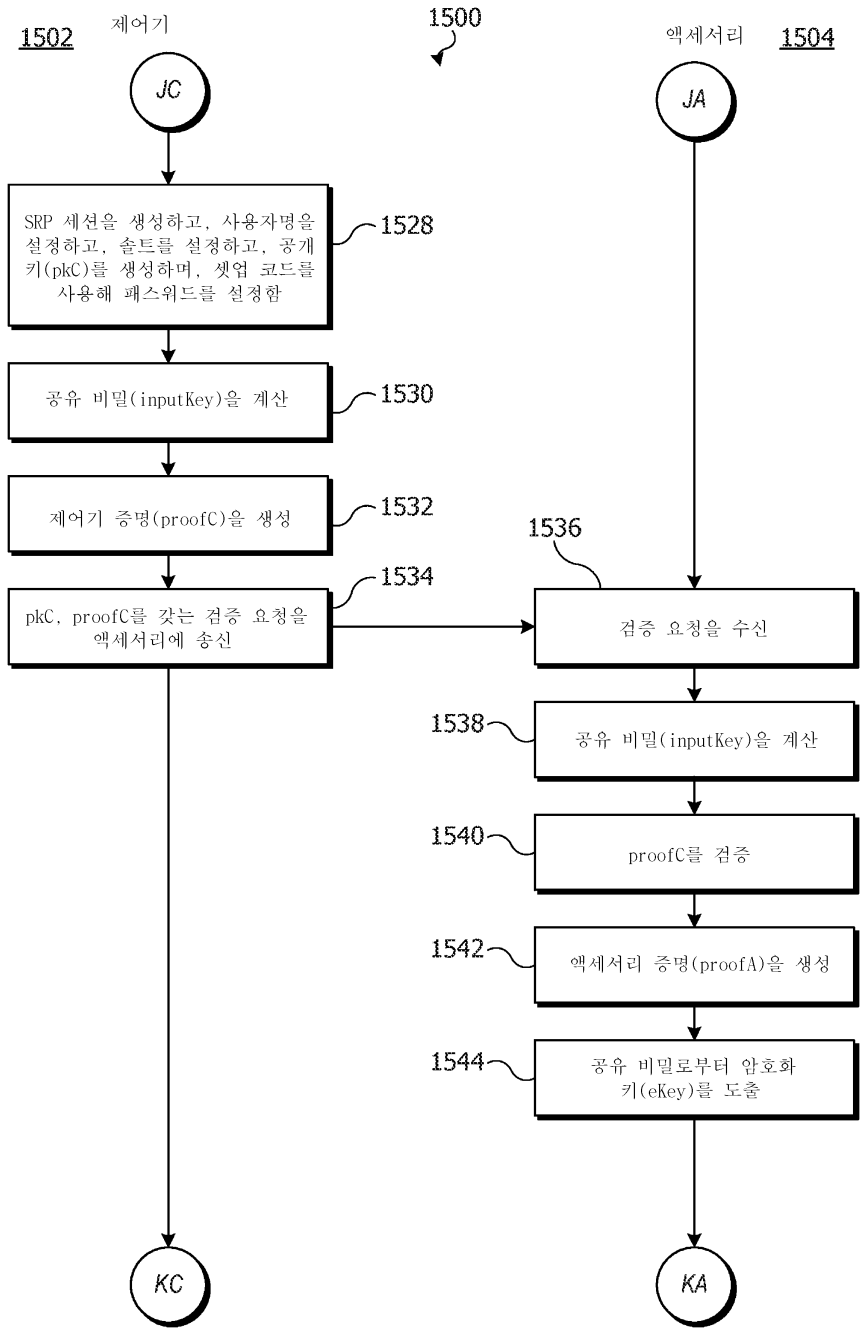
도면14c


1400
1402 제어기
액세서리 1404
EC
EA
userC 및 제어기 장기간 공개 키(LTPKC)를 갖는 데이터 객체를 구축
1442
edataC, aut TagC를 생성하기 위해 데이터 객체를 암호화함
1444
edataC, authTagC를 송신
1446
1448
수신된 authTagC를 검증
검증되는가?
예
1450
아니오
오류 메시지를 송신
1452
아니오
오류 메시지?
1454
예
종료
1456
1458
edataC를 복호화함
1460
LTPKC 및 userC를 페어링된 것으로서 지속적으로 저장
1462
액세서리 장기간 공개 키(LTPKA) 및 액세서리 사용자명(userA)을 갖는 데이터 객체를 구축
1464
edataA 및 authTagA를 생성하기 위해 데이터 객체를 암호화함
1466
edataA 및 authTagA를 송신
1468
authTagA을 검증
아니오
검증되는가?
1470
예
1474
LTPKA 및 userA를 복호화하고 페어링된 것으로서 지속적으로 저장
종료
1472
페어링됨
1476

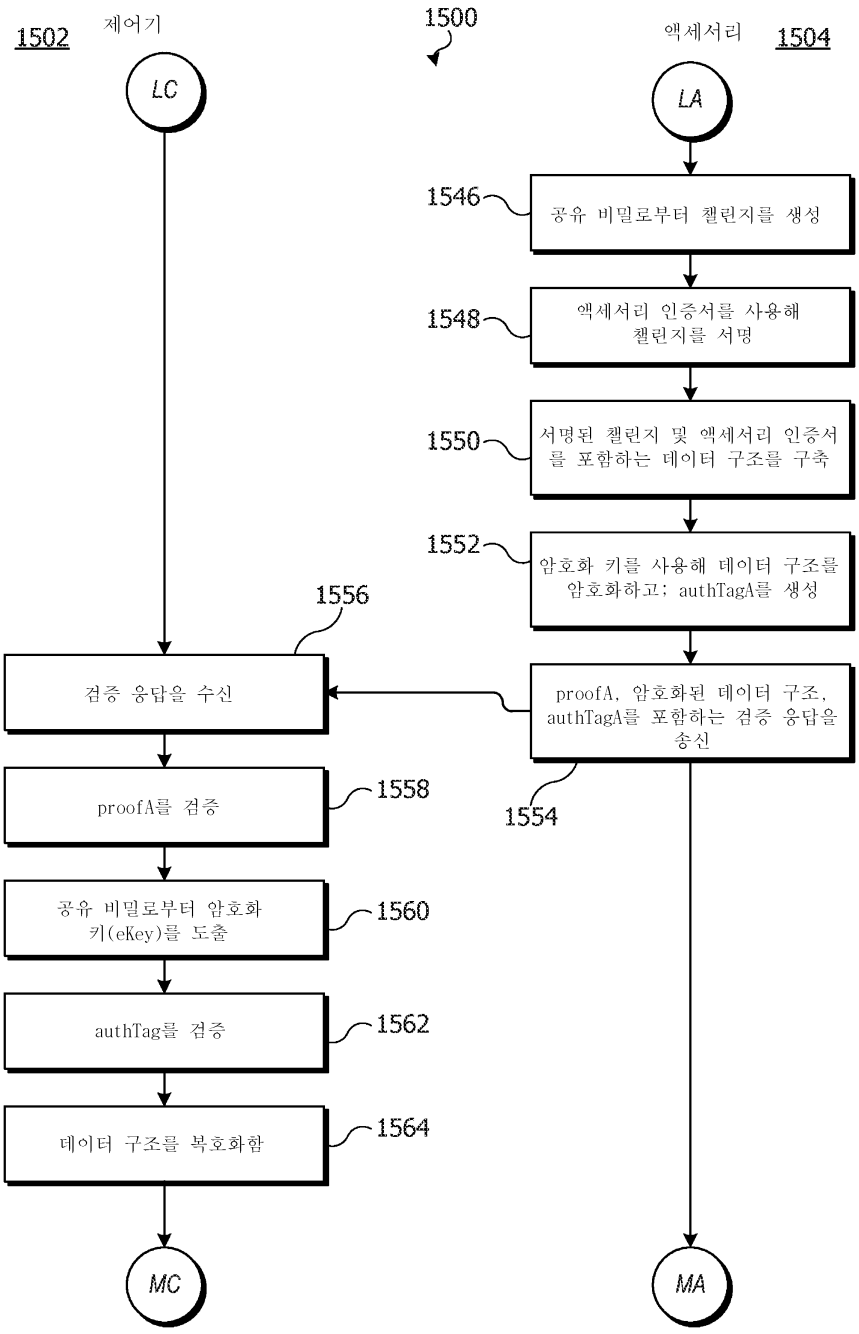
도면15a


1500
1502 제어기
액세서리 1504
페어 셋업을 시작하기 위해 시동 요청을 송신
1506
시동 요청을 수신
1508
모든 다른 제어기들이 페어 셋업을 개시하는 것을 차단
1510
최대 시도들이 초과 되었는가?
아니오
예
1512
종료
1514
SRP 세션을 생성하고, 사용자명을 설정하고, 랜덤 솔트를 생성하며, 셋업 코드를 획득함
1516
사용자에게 셋업 코드를 제시
1518
공개 키(pkA)를 생성
1520
1524
시동 응답을 수신
pkA 및 랜덤 솔트를 포함하는 시동 응답을 제어기에 송신
1522
사용자로부터 셋업 코드를 획득
1526
JC
JA

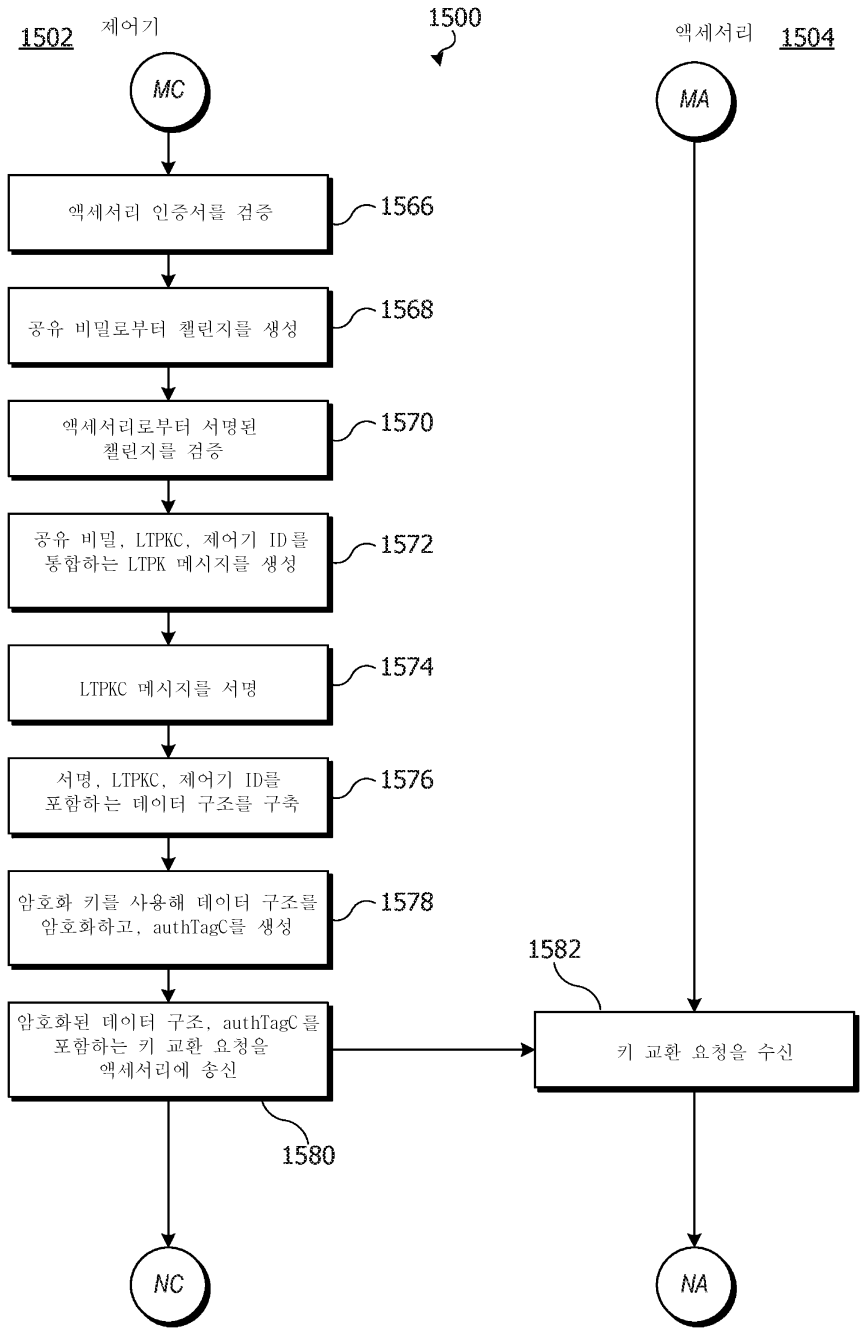
도면15b


1502
제어기
1500
액세서리
1504
JC
JA
SRP 세션을 생성하고, 사용자명을 설정하고, 솔트를 설정하고, 공개키(pkC)를 생성하며, 셋업 코드를 사용해 패스워드를 설정함
1528
공유 비밀(inputKey)을 계산
1530
제어기 증명(proofC)을 생성
1532
pkC, proofC를 갖는 검증 요청을 액세서리에 송신
1534
1536
검증 요청을 수신
1538
공유 비밀(inputKey)을 계산
1540
proofC를 검증
1542
액세서리 증명(proofA)을 생성
1544
공유 비밀로부터 암호화 키(eKey)를 도출
KC
KA

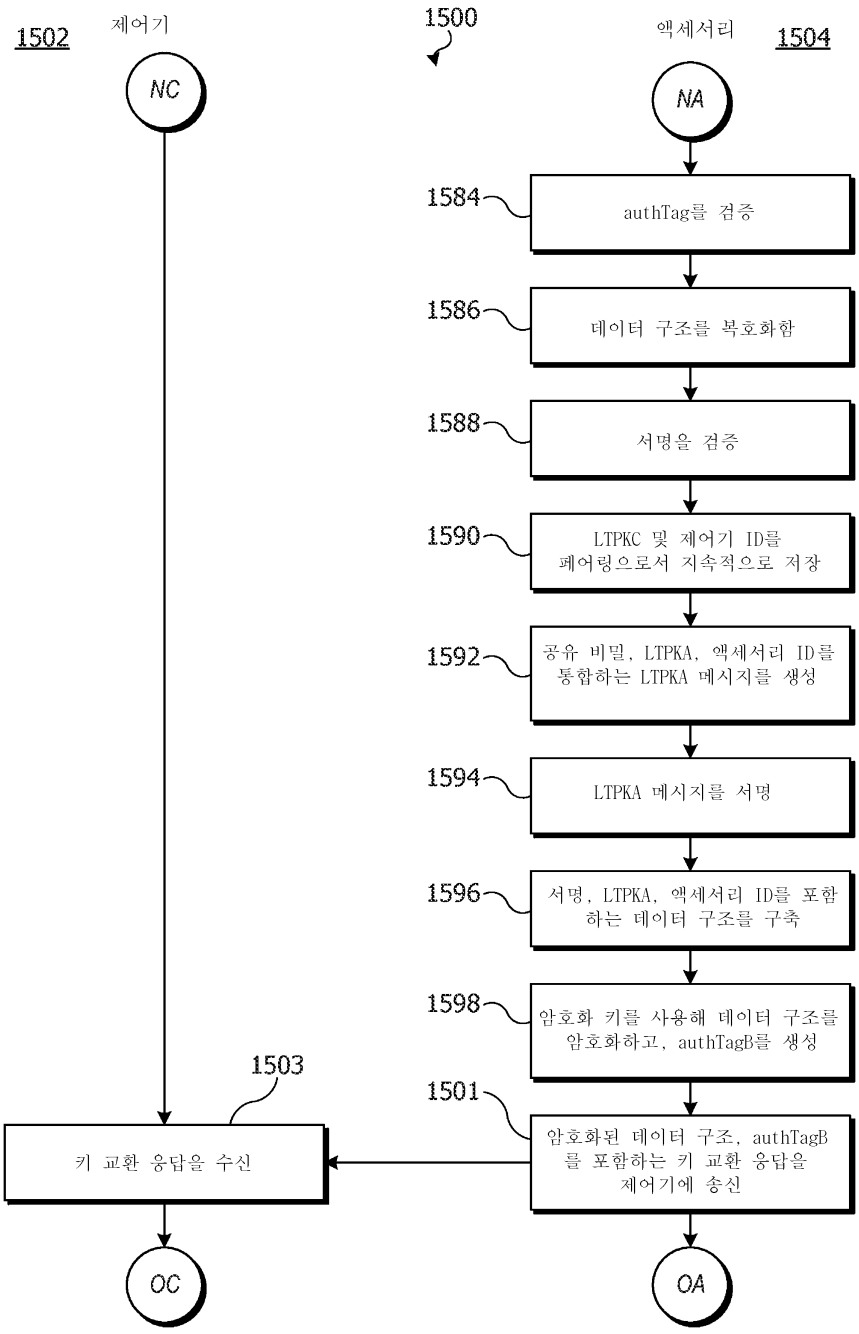
도면15c


1502
제어기
1500
액세서리
1504
LC
LA
1546
공유 비밀로부터 챌린지를 생성
1548
액세서리 인증서를 사용해 챌린지를 서명
1550
서명된 챌린지 및 액세서리 인증서를 포함하는 데이터 구조를 구축
1552
암호화 키를 사용해 데이터 구조를 암호화하고; authTagA를 생성
1556
검증 응답을 수신
proofA, 암호화된 데이터 구조, authTagA를 포함하는 검증 응답을 송신
1554
proofA를 검증
1558
공유 비밀로부터 암호화 키(eKey)를 도출
1560
authTag를 검증
1562
데이터 구조를 복호화함
1564
MC
MA

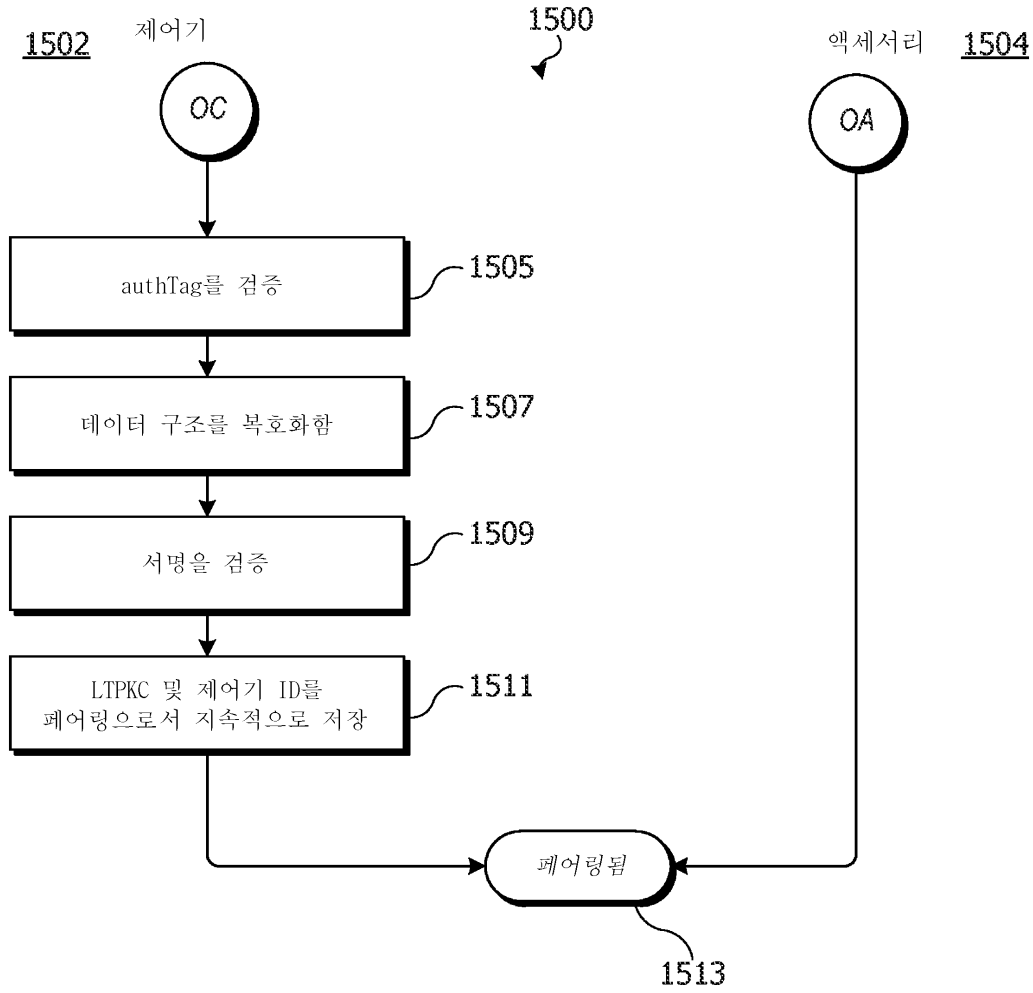
도면15d


1502
제어기
1500
액세서리
1504
MC
MA
액세서리 인증서를 검증
1566
공유 비밀로부터 챌린지를 생성
1568
액세서리로부터 서명된 챌린지를 검증
1570
공유 비밀, LTPKC, 제어기 ID를 통합하는 LTPK 메시지를 생성
1572
LTPKC 메시지를 서명
1574
서명, LTPKC, 제어기 ID를 포함하는 데이터 구조를 구축
1576
암호화 키를 사용해 데이터 구조를 암호화하고, authTagC를 생성
1578
1582
암호화된 데이터 구조, authTagC를 포함하는 키 교환 요청을 액세서리에 송신
키 교환 요청을 수신
1580
NC
NA

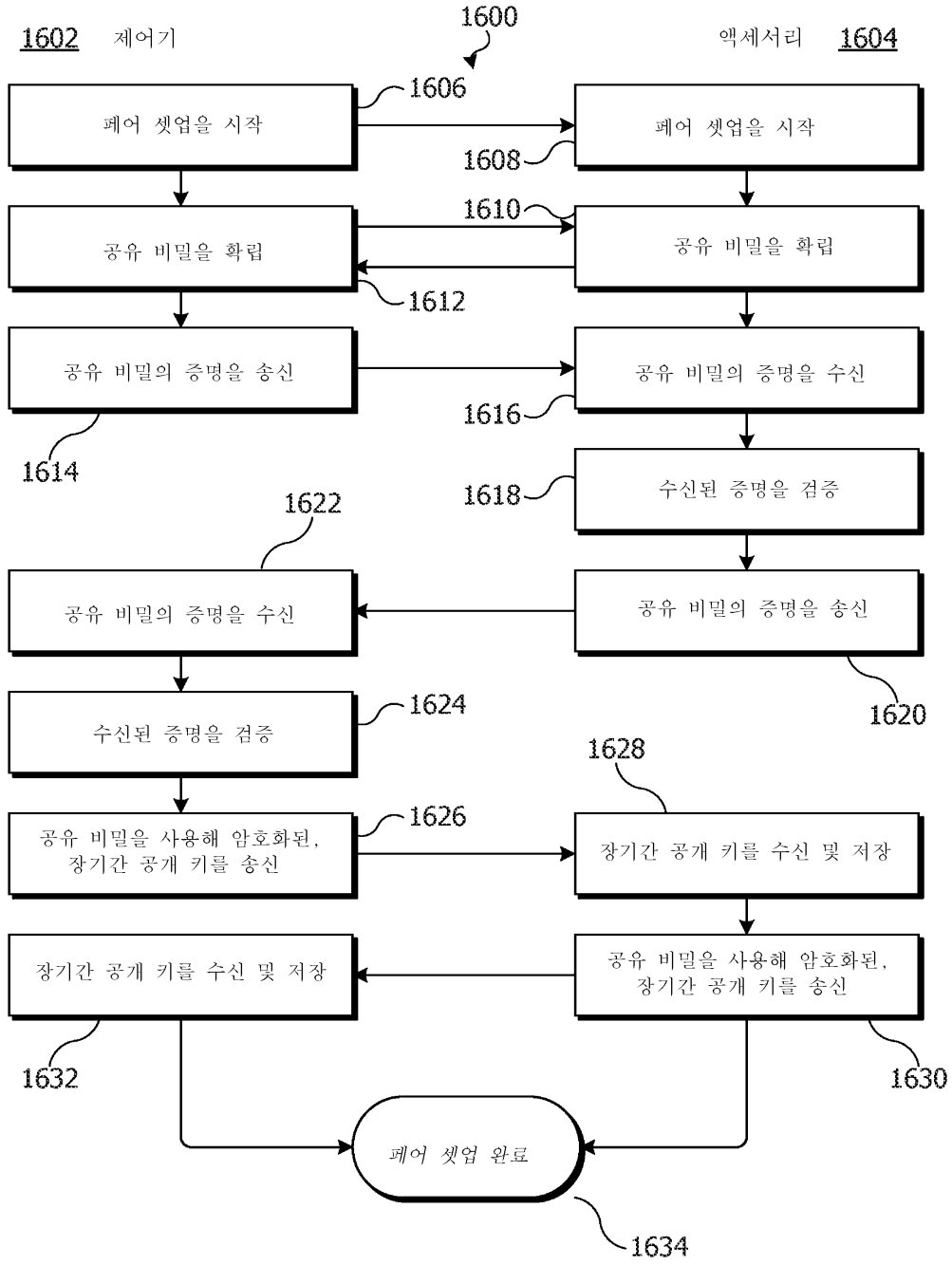
도면15e


1502
제어기
NC
1500
액세서리
1504
NA
1584
authTag를 검증
1586
데이터 구조를 복호화함
1588
서명을 검증
1590
LTPKC 및 제어기 ID를 페어링으로서 지속적으로 저장
1592
공유 비밀, LTPKA, 액세서리 ID를 통합하는 LTPKA 메시지를 생성
1594
LTPKA 메시지를 서명
1596
서명, LTPKA, 액세서리 ID를 포함하는 데이터 구조를 구축
1598
암호화 키를 사용해 데이터 구조를 암호화하고, authTagB를 생성
1501
암호화된 데이터 구조, authTagB를 포함하는 키 교환 응답을 제어기에 송신
1503
키 교환 응답을 수신
OC
OA

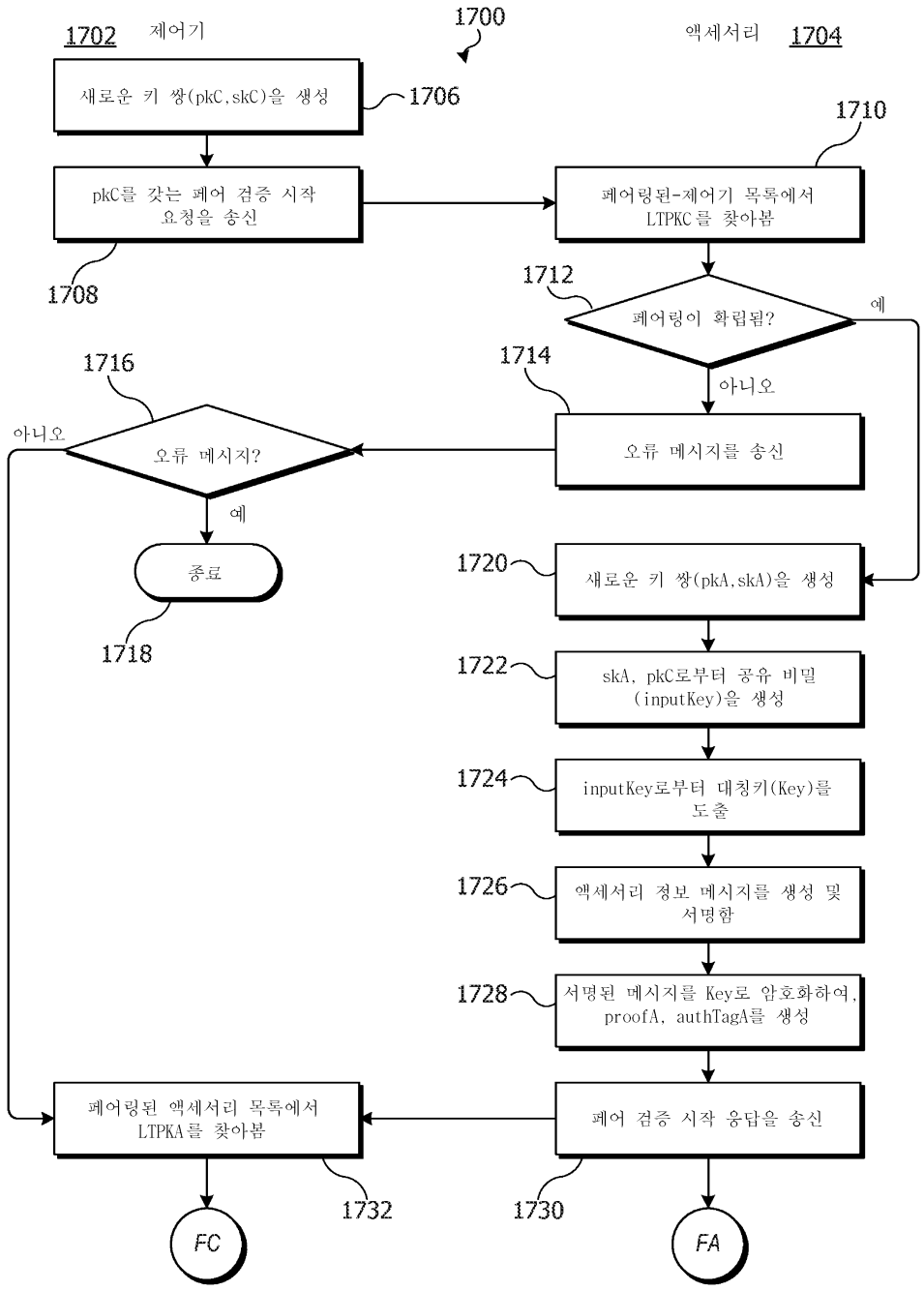
도면15f


1502
제어기
1500
액세서리
1504
OC
OA
authTag를 검증
1505
데이터 구조를 복호화함
1507
서명을 검증
1509
LTPKC 및 제어기 ID를
페어링으로서 지속적으로 저장
1511
페어링됨
1513

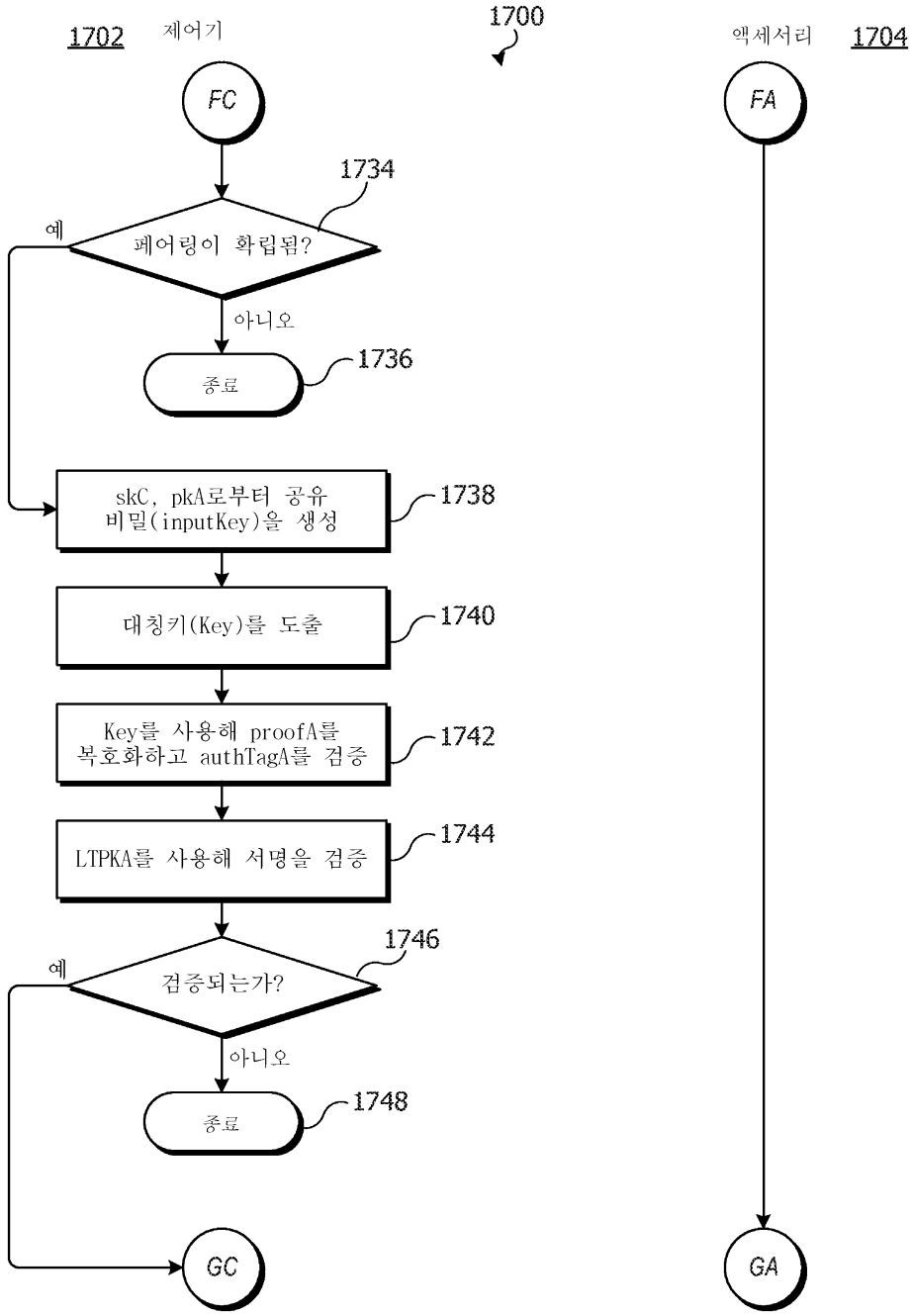
도면16


1600
1602 제어기
액세서리 1604
1606
페어 셋업을 시작
페어 셋업을 시작
1608
1610
공유 비밀을 확립
공유 비밀을 확립
1612
공유 비밀의 증명을 송신
공유 비밀의 증명을 수신
1614
1616
수신된 증명을 검증
1618
1622
공유 비밀의 증명을 수신
공유 비밀의 증명을 송신
1620
수신된 증명을 검증
1624
1628
공유 비밀을 사용해 암호화된, 장기간 공개 키를 송신
1626
장기간 공개 키를 수신 및 저장
장기간 공개 키를 수신 및 저장
공유 비밀을 사용해 암호화된, 장기간 공개 키를 송신
1632
1630
페어 셋업 완료
1634

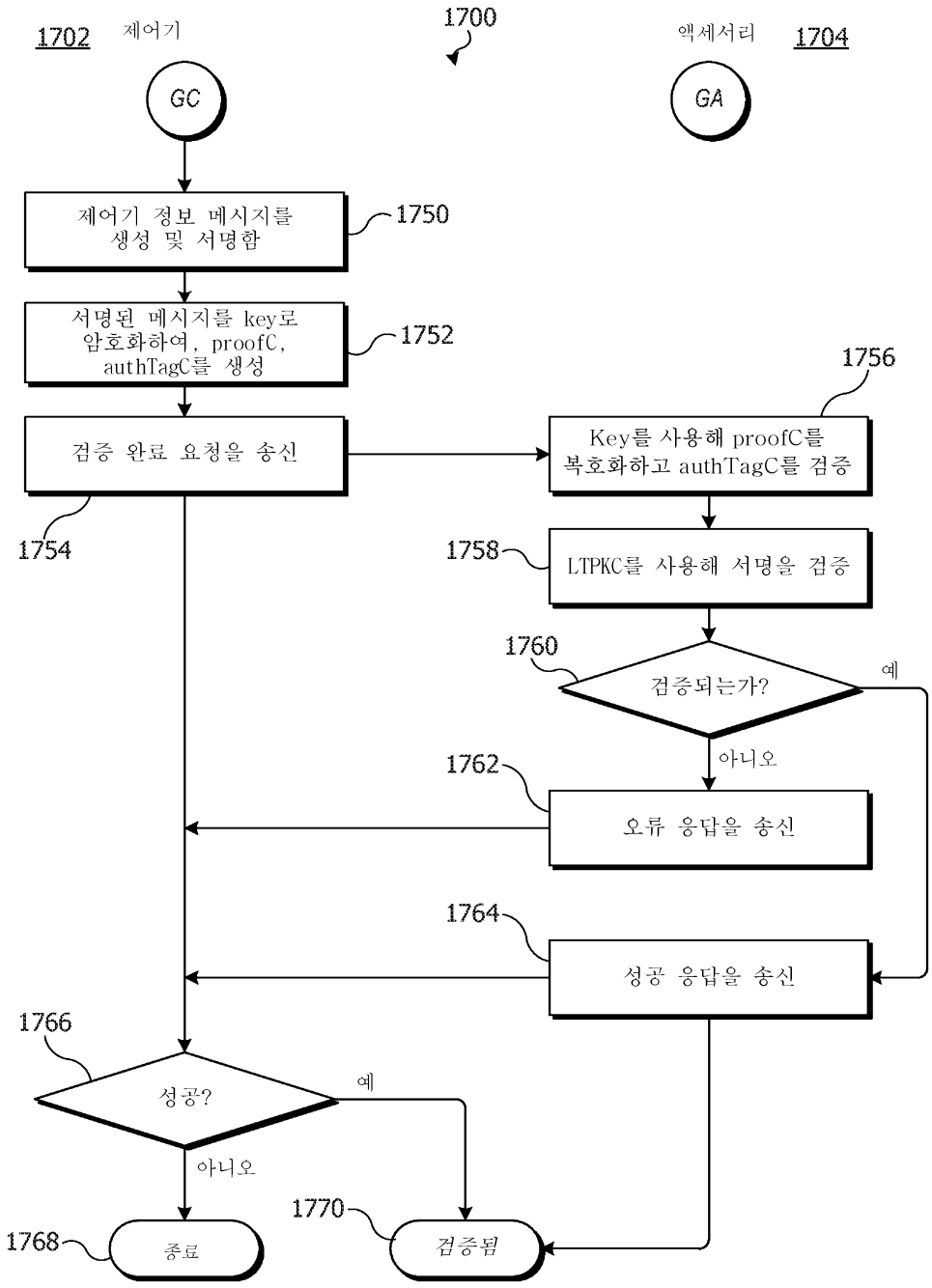
도면17a


1700
1702 제어기
액세서리 1704
새로운 키 쌍(pkC,skC)을 생성
1706
pkC를 갖는 페어 검증 시작 요청을 송신
1708
페어링된-제어기 목록에서 LTPKC를 찾아봄
1710
1712
페어링이 확립됨?
예
아니오
1714
오류 메시지를 송신
1716
오류 메시지?
아니오
예
종료
1718
1720
새로운 키 쌍(pkA,skA)을 생성
1722
skA, pkC로부터 공유 비밀(inputKey)을 생성
1724
inputKey로부터 대칭키(Key)를 도출
1726
액세서리 정보 메시지를 생성 및 서명함
1728
서명된 메시지를 Key로 암호화하여, proofA, authTagA를 생성
페어 검증 시작 응답을 송신
1730
페어링된 액세서리 목록에서 LTPKA를 찾아봄
1732
FC
FA

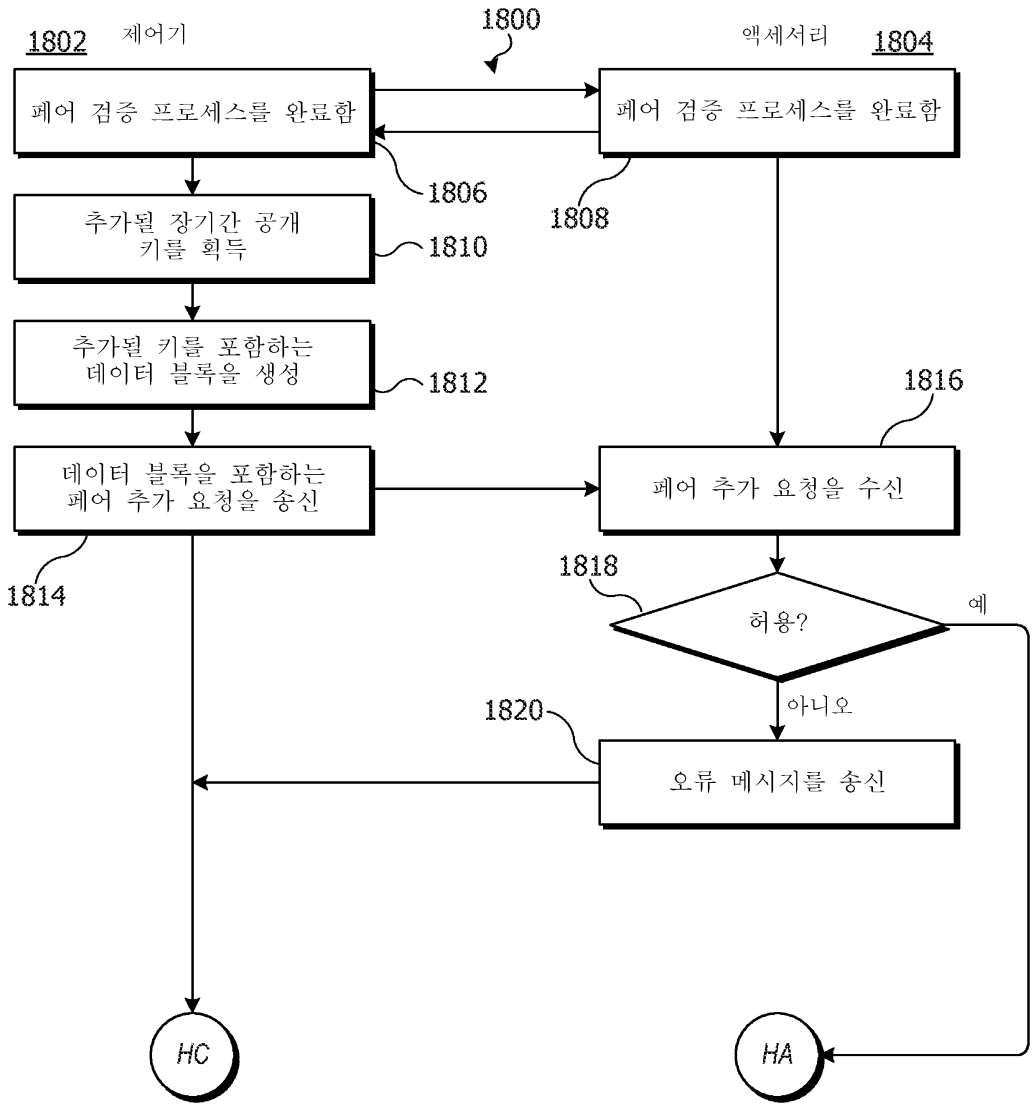
도면17b


1700
1702
제어기
액세서리
1704
FC
FA
1734
예
페어링이 확립됨?
아니오
1736
종료
skC, pkA로부터 공유 비밀(inputKey)을 생성
1738
대칭키(Key)를 도출
1740
Key를 사용해 proofA를 복호화하고 authTagA를 검증
1742
LTPKA를 사용해 서명을 검증
1744
1746
예
검증되는가?
아니오
1748
종료
GC
GA

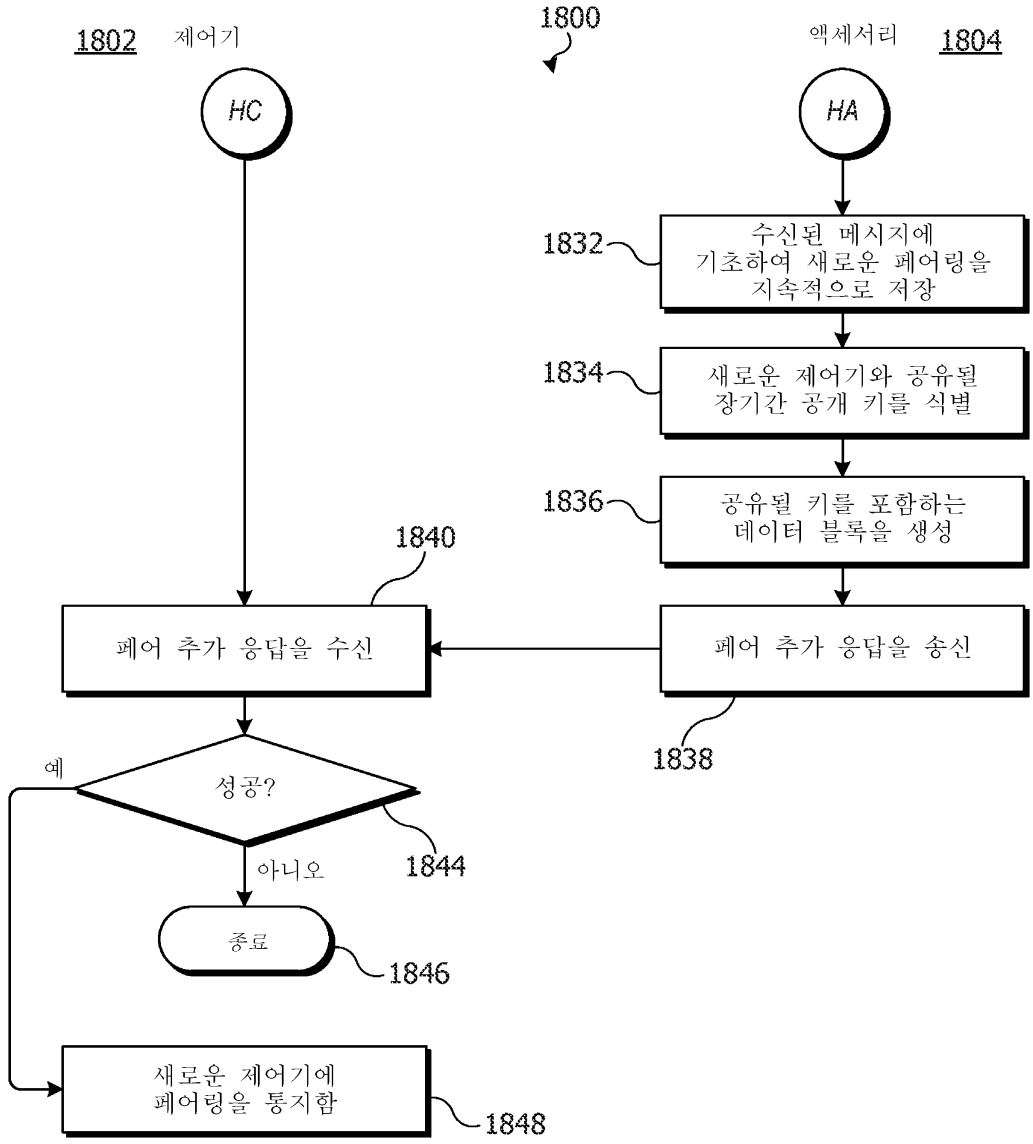
도면17c


1700
1702
제어기
액세서리
1704
GC
GA
제어기 정보 메시지를 생성 및 서명함
1750
서명된 메시지를 key로 암호화하여, proofC, authTagC를 생성
1752
1756
검증 완료 요청을 송신
Key를 사용해 proofC를 복호화하고 authTagC를 검증
1754
1758
LTPKC를 사용해 서명을 검증
1760
검증되는가?
예
아니오
1762
오류 응답을 송신
1764
성공 응답을 송신
1766
성공?
예
아니오
1770
1768
종료
검증됨

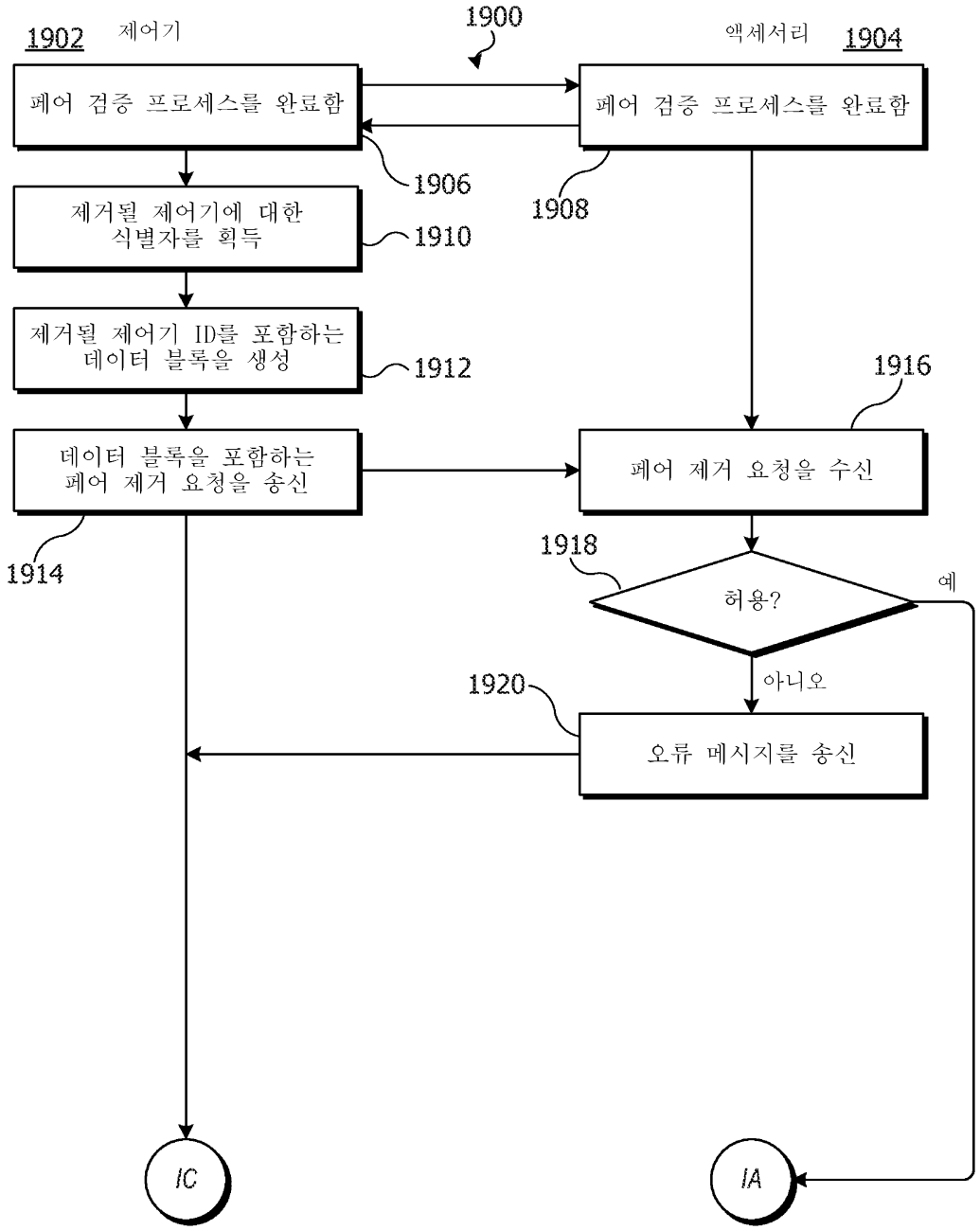
도면18a


1800
1802
제어기
액세서리
1804
페어 검증 프로세스를 완료함
페어 검증 프로세스를 완료함
1806
1808
추가될 장기간 공개 키를 획득
1810
추가될 키를 포함하는 데이터 블록을 생성
1812
1816
데이터 블록을 포함하는 페어 추가 요청을 송신
페어 추가 요청을 수신
1814
1818
허용?
예
아니오
1820
오류 메시지를 송신
HC
HA

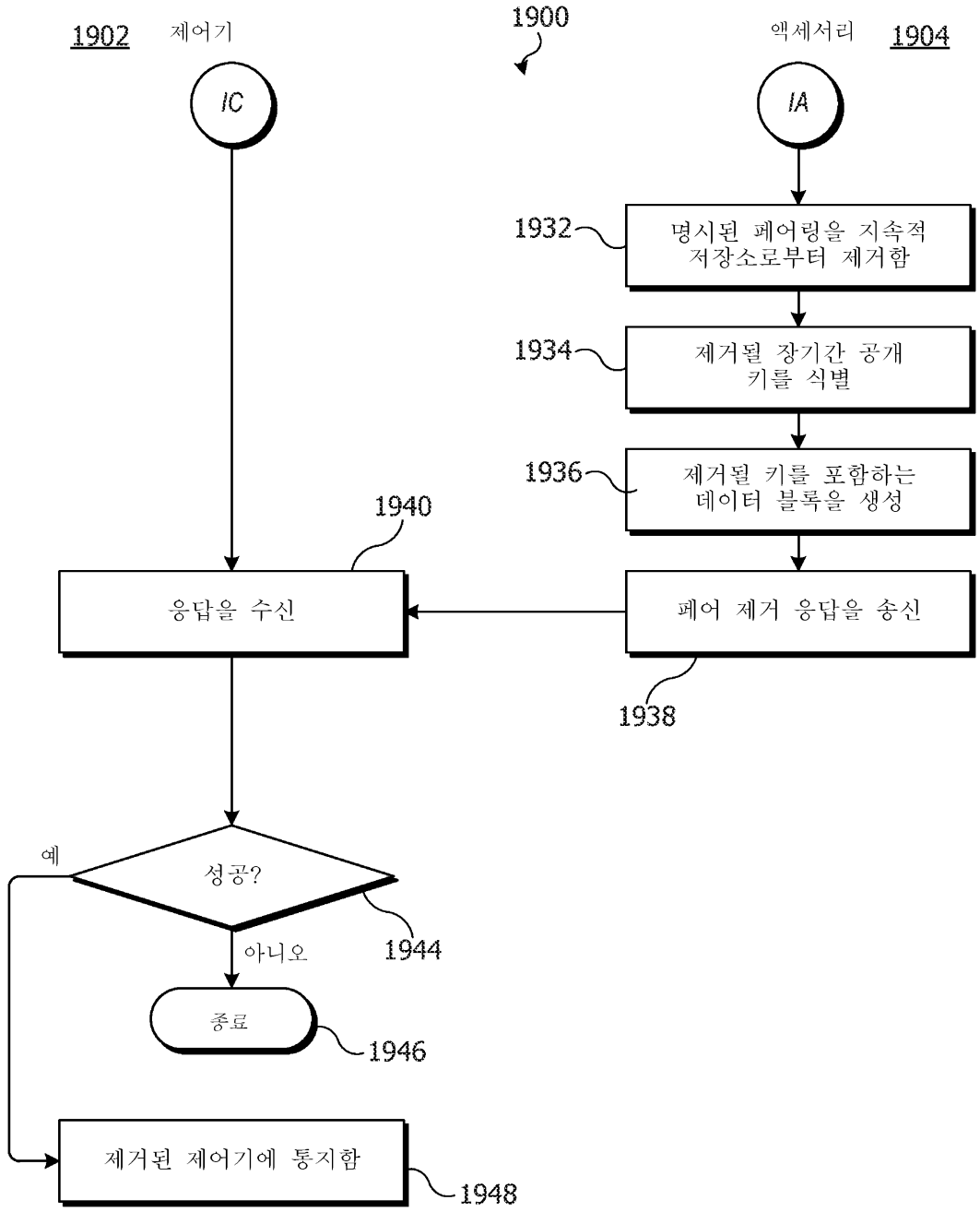
도면18b


1802
제어기
1800
액세서리
1804
HC
HA
1832
수신된 메시지에 기초하여 새로운 페어링을 지속적으로 저장
1834
새로운 제어기와 공유될 장기간 공개 키를 식별
1836
공유될 키를 포함하는 데이터 블록을 생성
1840
페어 추가 응답을 수신
페어 추가 응답을 송신
1838
예
성공?
1844
아니오
종료
1846
새로운 제어기에 페어링을 통지함
1848

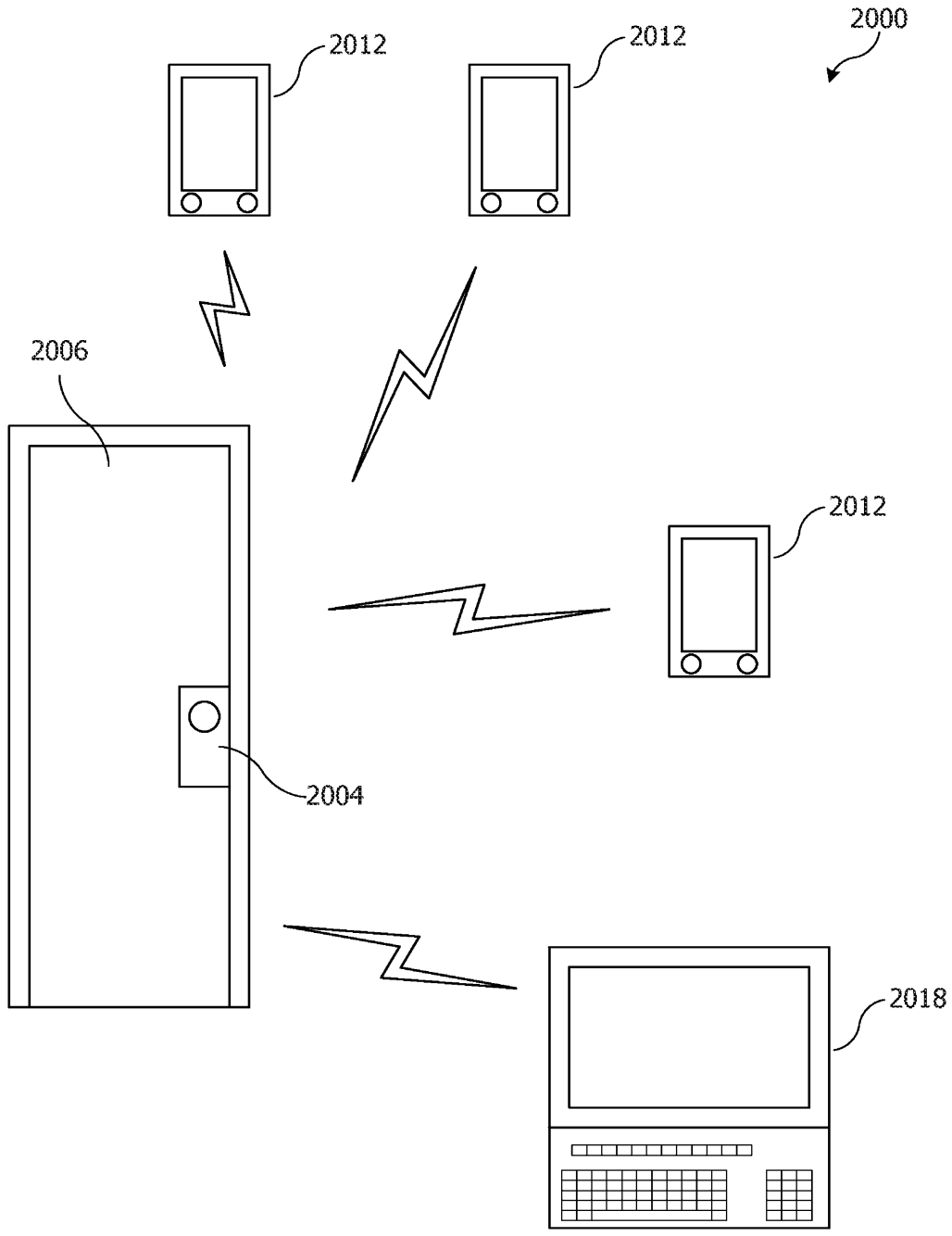
도면19a


1900
1902
제어기
액세서리
1904
페어 검증 프로세스를 완료함
페어 검증 프로세스를 완료함
1906
1908
제거될 제어기에 대한 식별자를 획득
1910
제거될 제어기 ID를 포함하는 데이터 블록을 생성
1912
1916
데이터 블록을 포함하는 페어 제거 요청을 송신
페어 제거 요청을 수신
1914
1918
허용?
예
아니오
1920
오류 메시지를 송신
IC
IA

도면19b


1902
제어기
1900
액세서리
1904
IC
IA
1932
명시된 페어링을 지속적 저장소로부터 제거함
1934
제거될 장기간 공개 키를 식별
1936
제거될 키를 포함하는 데이터 블록을 생성
1940
응답을 수신
페어 제거 응답을 송신
1938
예
성공?
1944
아니오
종료
1946
제거된 제어기에 통지함
1948

도면20

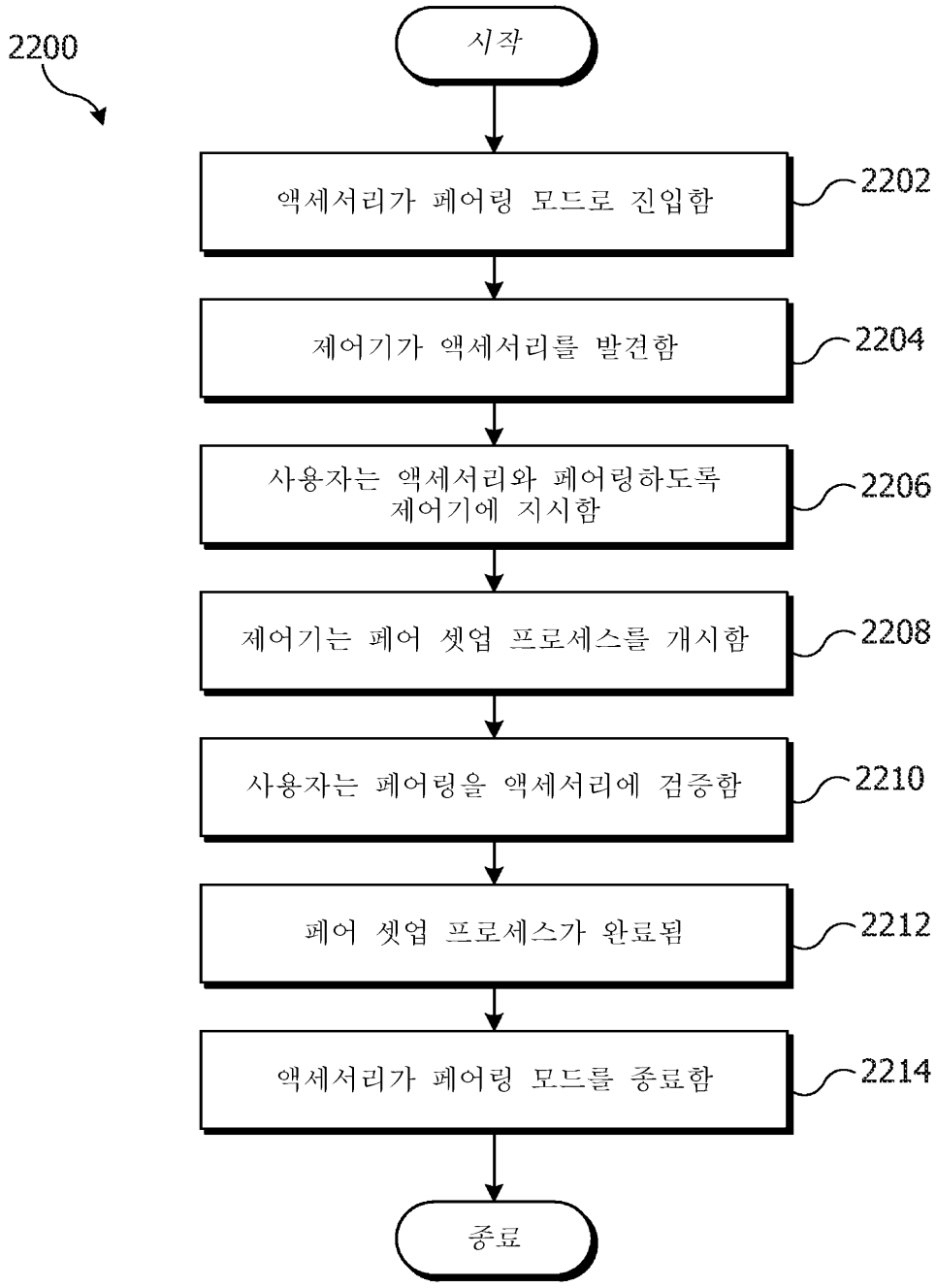


도면21

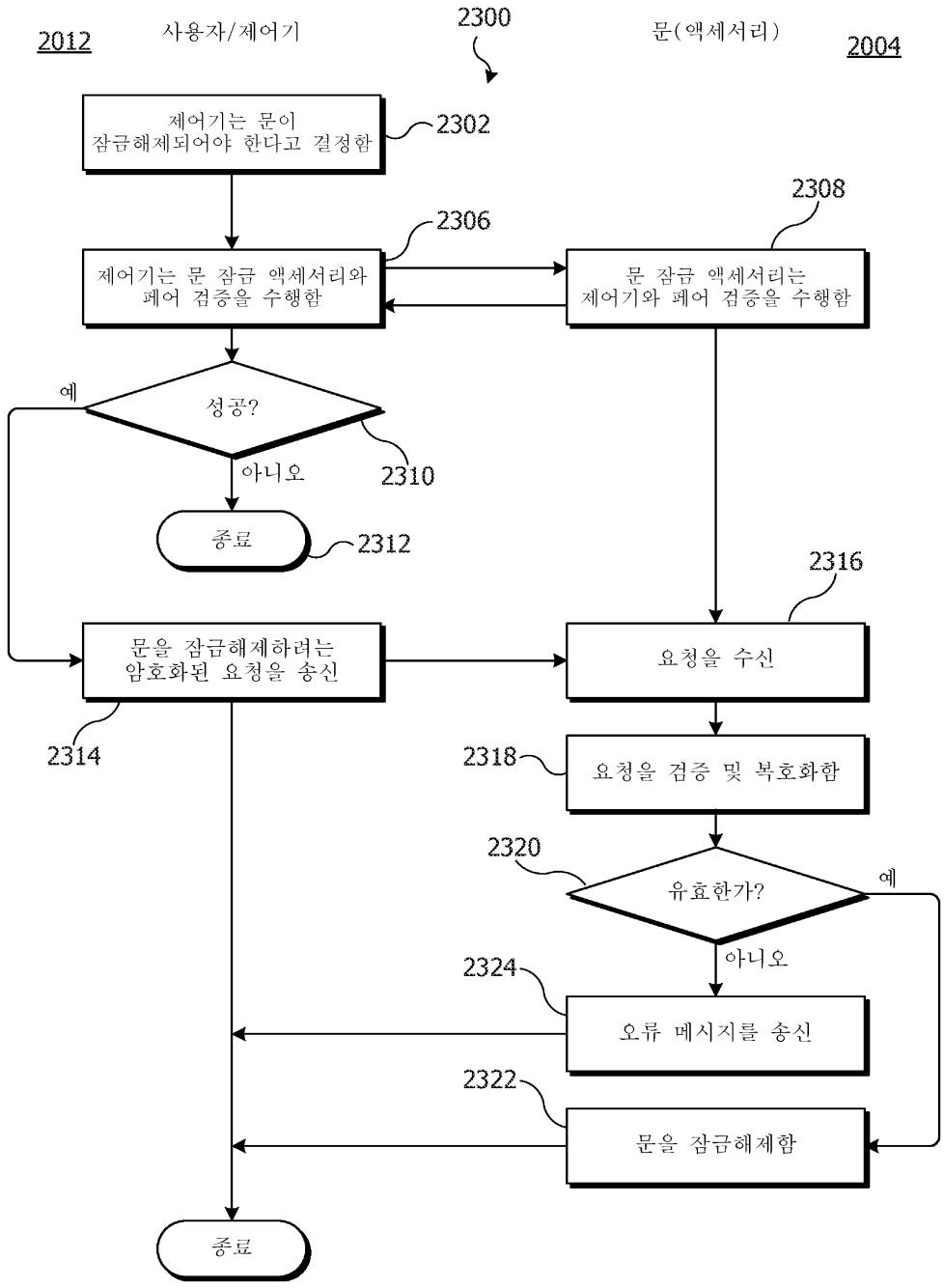
2100

서비스	특성들	IID	값
액세서리 정보 (instance ID=1) 2102	com.proto.ch.identify	2	null
	com.proto.ch.manufacturer	3	"Acme"
	com.proto.ch.model	4	"A-Lock"
	com.proto.ch.name	5	"A Name"
	com.proto.ch.serial-number	6	"7353"
잠금 메커니즘 (instance ID=7) 2104	com.proto.ch.lock-mech.current-state 2110	8	unlocked
	com.proto.ch.lock-mech.target-state 2112	9	unlocked
	com.proto.ch.name	10	"A-Lock-Mech"
잠금 관리 (instance ID=11) 2106	com.proto.ch.lock-mgt.control-point 2114	12	null
	com.proto.ch.version	13	1.2.3
	com.proto.ch.logs	14	<tlv>
	com.proto.ch.admin-only-access	15	false
	com.proto.ch.lock-mgt.auto-timeout	16	5
	com.proto.ch.lock-mech.last-action	17	6
	com.proto.ch.door-state.current	18	closed
	com.proto.ch.name	19	"A-Lock-Mgr"

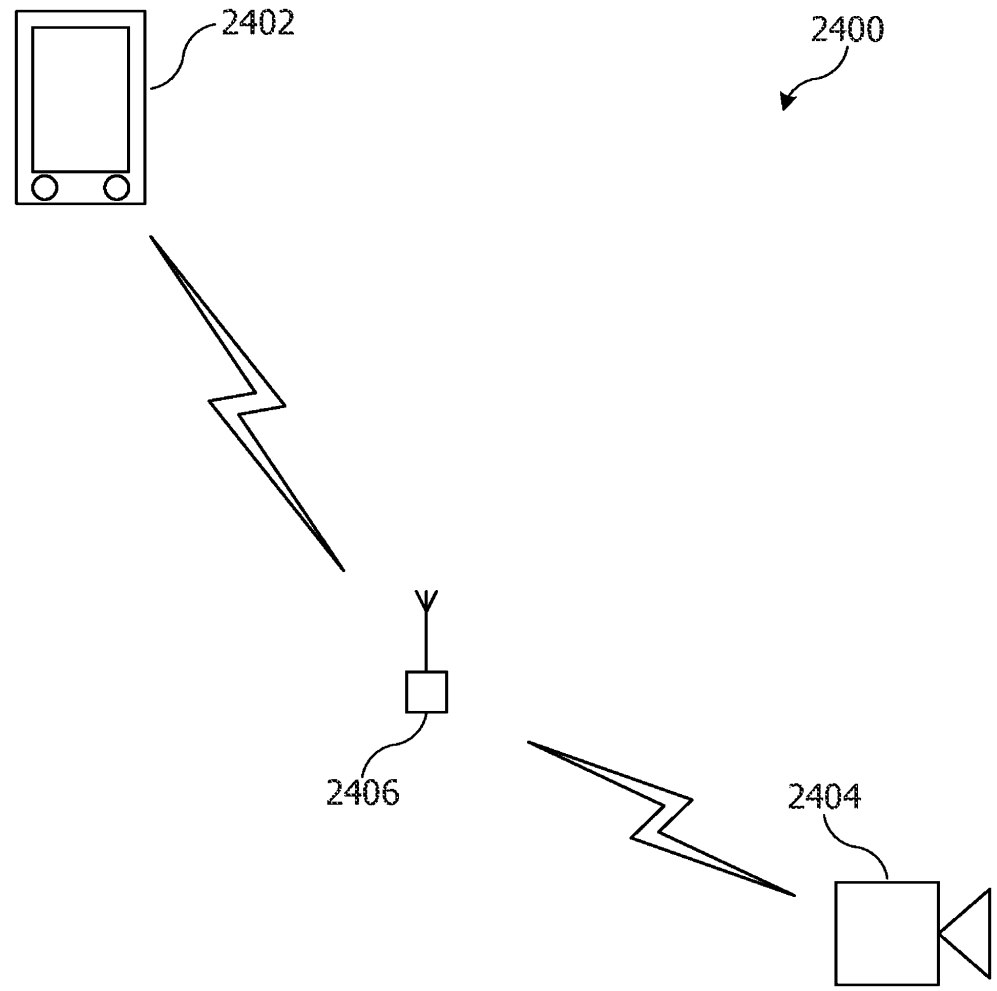
도면22


2200
시작
액세서리가 페어링 모드로 진입함
2202
제어기가 액세서리를 발견함
2204
사용자는 액세서리와 페어링하도록 제어기에 지시함
2206
제어기는 페어 셋업 프로세스를 개시함
2208
사용자는 페어링을 액세서리에 검증함
2210
페어 셋업 프로세스가 완료됨
2212
액세서리가 페어링 모드를 종료함
2214
종료

도면23


2012
사용자/제어기
2300
문(액세서리)
2004
제어기는 문이 잠금해제되어야 한다고 결정함
2302
2306
제어기는 문 잠금 액세서리와 페어 검증을 수행함
2308
문 잠금 액세서리는 제어기와 페어 검증을 수행함
예
성공?
아니오
2310
종료
2312
2316
문을 잠금해제하려는 암호화된 요청을 송신
요청을 수신
2314
2318
요청을 검증 및 복호화함
2320
유효한가?
예
아니오
2324
오류 메시지를 송신
2322
문을 잠금해제함
종료

도면24



도면25a

서비스	속성	값
IP 카메라 스트리밍 2501	타입	com.proto.svc.ip-camera-streaming
	필수 특성	com.proto.ch.on
		com.proto.ch.ip-cam-mgt.session.start
		com.proto.ch.ip-cam-mgt.session.end
		com.proto.ch.video.codec.name
		com.proto.ch.video.codec.parameters
		com.proto.ch.video.attributes
		com.proto.ch.rtp.video-payload-type
		com.proto.ch.rtp.protocol
		com.proto.ch.rtcp.extensions
		com.proto.ch.srtp.crytpo-suite
	선택적 특성	com.proto.ch.audio.codec.name
		com.proto.ch.audio.codec.parameters
		com.proto.ch.audio.codec.channels
		com.proto.ch.audio.attributes
		com.proto.ch.rtp.audio.clock-rate
		com.proto.ch.rtp.audio.payload-type
레코딩 2502	타입	com.proto.svc.recording
	필수 특성	com.proto.ch.recording.control
		com.proto.ch.recording.status
재생 2503	타입	com.proto.svc.playback
	필수 특성	com.proto.ch.playback.control
		com.proto.ch.playback.status
	선택적 특성	com.proto.ch.playback.speed

도면25b

서비스	속성	값
카메라	타입	com.proto.svc.camera
	필수 특성	com.ch.proto.on
2504	선택적 특성	com.ch.proto.pan
		com.ch.proto.tilt
		com.ch.proto.zoom
		com.ch.proto.rotate
		com.ch.proto.night-vision
		com.ch.proto.mirror
마이크로폰	타입	com.proto.svc.microphone
2505	필수 특성	com.proto.ch.on
		com.proto.ch.volume
스피커	타입	com.proto.svc.speaker
2506	필수 특성	com.proto.ch.on
		com.proto.ch.volume

도면26a

특성	속성	값
세션 시작 2601	타입	com.proto.ch.ip-cam-mgt.session.start
	허가들	Paired Write
	포맷	<object>[도 22b 참조]
세션 종료 2602	타입	com.proto.ch.ip-cam-mgt.session.end
	허가들	Paired Write
	포맷	<object>
비디오 코덱 이름 2603	타입	com.proto.ch.video.codec.name
	허가들	Paired Read
	포맷	<string>
비디오 코덱 파라미터들 2604	타입	com.proto.ch.video.codec.parameters
	허가들	Paired Read
	포맷	<object>
비디오 속성들 2605	타입	com.proto.ch.video.attributes
	허가들	Paired Read
	포맷	<object>
RTP 비디오 페이로드 타입 2606	타입	com.proto.ch.rtp.video.payload-type
	허가들	Paired Read
	포맷	<int>
RTP 프로토콜 2607	타입	com.proto.ch.rtp.protocol
	허가들	Paired Read
	포맷	<string>
RTCP 확장들 2608	타입	com.proto.ch.rtp.extensions
	허가들	Paired Read
	포맷	Array of <string>
SRTP 암호 스위트 2609	타입	com.proto.ch.srtp.crypto-suite
	허가들	Paired Read
	포맷	<string>

도면26b

특성	속성	값
오디오 코덱 이름 2610	타입	com.proto.ch.audio.codec.name
	허가들	Paired Read
	포맷	<string>
오디오 코덱 파라미터들 2611	타입	com.proto.ch.audio.codec.parameters
	허가들	Paired Read
	포맷	<object>
오디오 코덱 채널들 2612	타입	com.proto.ch.audio.codec.channels
	허가들	Paired Read
	포맷	<int>
오디오 속성들 2613	타입	com.proto.ch.audio.attributes
	허가들	Paired Read
	포맷	<object>
RTP 오디오 클록 속도 2614	타입	com.proto.ch.rtp.audio.clock-rate
	허가들	Paired Read
	포맷	<int>
RTP 오디오 페이로드 타입 2615	타입	com.proto.ch.rtp.audio.payload-type
	허가들	Paired Read
	포맷	<int>

도면26c

키 이름	값 타입	설명
"session-id" 2631	UUID	스트리밍 세션에 대한 UUID
"controller-IP" 2632	string	비디오 스트림을 수신하는 제어기의 IP 주소
"controller-port" 2633	int	비디오 스트림을 수신하는 제어기의 포트
"controller-srtp-master-key" 2634	string	제어기로부터 카메라로 송신되는 RTCP에 대한 SRTP 마스터 키
"controller-srtp-master-salt" 2635	string	제어기로부터 카메라로 송신되는 RTCP에 대한 SRTP 마스터 솔트
"video-max-bandwidth" 2636	int	비디오에 대한 최대 대역폭(kbps)
"audio-max-bandwidth" 2637	int	오디오에 대한 최대 대역폭(kbps)

도면26d

특성	속성	값
야간 시야 2651	타입	com.proto.ch.ip-camera-night-vision
	포맷	<int>
	허가들	Paired Read, Paired Write
팬 2652	타입	com.proto.ch.pan
	포맷	<int>
	허가들	Paired Read, Paired Write
틸트 2653	타입	com.proto.ch.tilt
	포맷	<int>
	허가들	Paired Read, Paired Write
회전 2654	타입	com.proto.ch.rotation
	포맷	<int>
	허가들	Paired Read, Paired Write
줌 2655	타입	com.proto.ch.zoom
	포맷	<float>
	허가들	read/write/update
미러 2656	타입	com.proto.ch.ip-camera-mirror
	포맷	<boolean>
	허가들	read/write/update

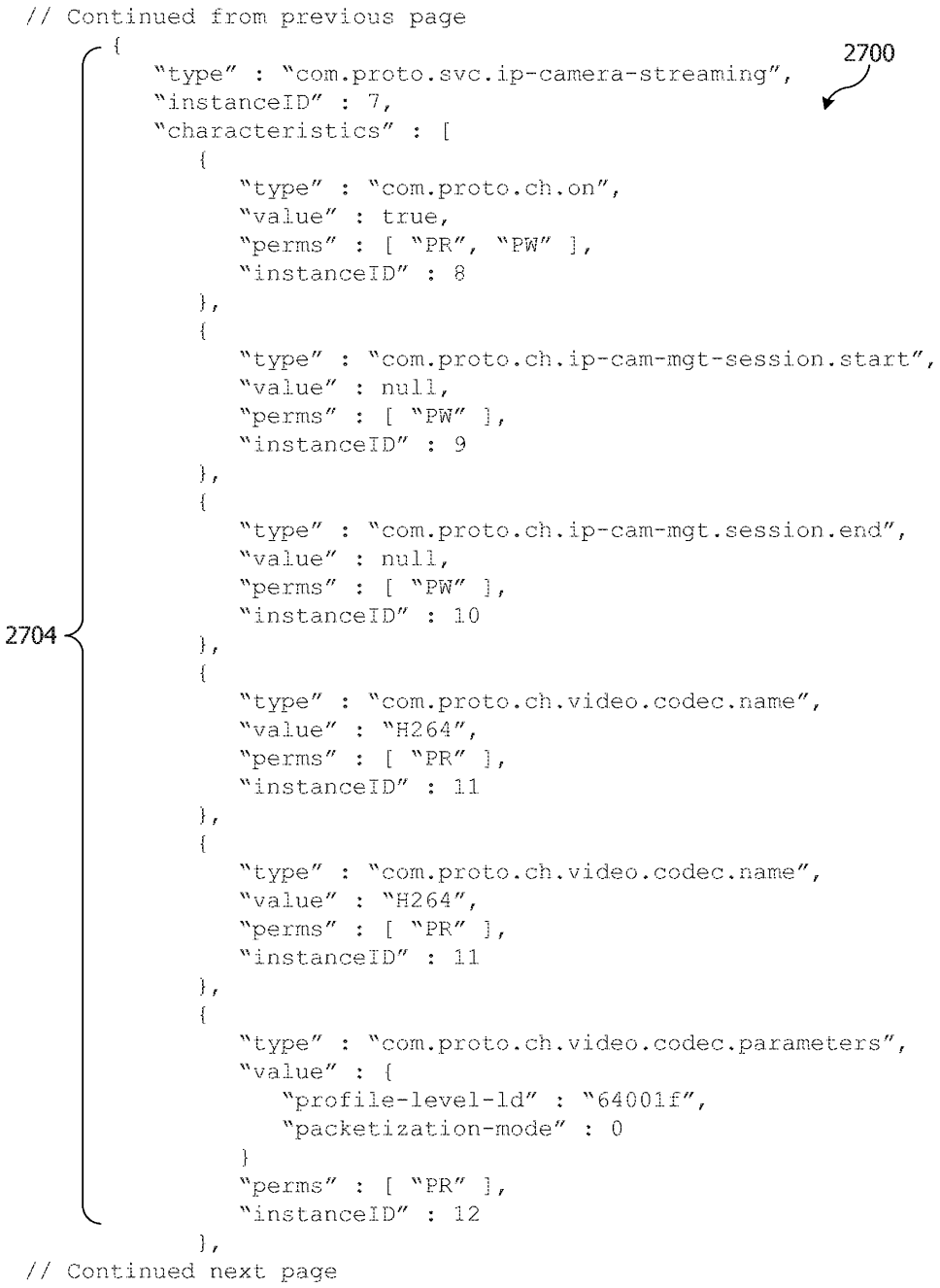
도면26e

특성	속성	값
레코딩 제어 2661	타입	com.proto.ch.recording.control
	포맷	<object>
	허가들	Paired Write
레코딩 상태 2662	타입	com.proto.ch.recording.status
	포맷	<object>
	허가들	Paired Read
재생 제어 2663	타입	com.proto.ch.playback.control
	포맷	<object>
	허가들	Paired Write
재생 상태 2664	타입	com.proto.ch.playback.status
	포맷	<object>
	허가들	Paired Read
재생 속도 2665	타입	com.proto.ch.playback.speed
	포맷	<float>
	허가들	Paired Read, Paired Write

도면27a


2700
2702
{
"services" : [
{
"type" : "com.proto.svc.accessory-info",
"instanceID" : 1,
"characteristics" : [
{
"type" : "com.proto.ch.id",
"value" : "55:44:33:22:11:00",
"perms" : ["PR"],
"instanceID" : 2
},
{
"type" : "com.proto.ch.name",
"value" : "Acme IP Camera",
"perms" : ["PR"],
"instanceID" : 3
},
{
"type" : "com.proto.ch.manufacturer",
"value" : "Acme",
"perms" : ["PR"],
"instanceID" : 4
},
{
"type" : "com.proto.ch.model",
"value" : "Q5678",
"perms" : ["PR"],
"instanceID" : 5
},
{
"type" : "com.proto.ch.serial-number",
"value" : "9Z8Y7X6W5V4U",
"perms" : ["PR"],
"instanceID" : 6
}
]
},
// Continued next page

도면27b


// Continued from previous page
{
"type" : "com.proto.svc.ip-camera-streaming",
"instanceID" : 7,
"characteristics" : [
{
"type" : "com.proto.ch.on",
"value" : true,
"perms" : ["PR", "PW"],
"instanceID" : 8
},
{
"type" : "com.proto.ch.ip-cam-mgt-session.start",
"value" : null,
"perms" : ["PW"],
"instanceID" : 9
},
{
"type" : "com.proto.ch.ip-cam-mgt.session.end",
"value" : null,
"perms" : ["PW"],
"instanceID" : 10
},
{
"type" : "com.proto.ch.video.codec.name",
"value" : "H264",
"perms" : ["PR"],
"instanceID" : 11
},
{
"type" : "com.proto.ch.video.codec.name",
"value" : "H264",
"perms" : ["PR"],
"instanceID" : 11
},
{
"type" : "com.proto.ch.video.codec.parameters",
"value" : {
"profile-level-ld" : "64001f",
"packetization-mode" : 0
}
"perms" : ["PR"],
"instanceID" : 12
},
// Continued next page
2700
2704

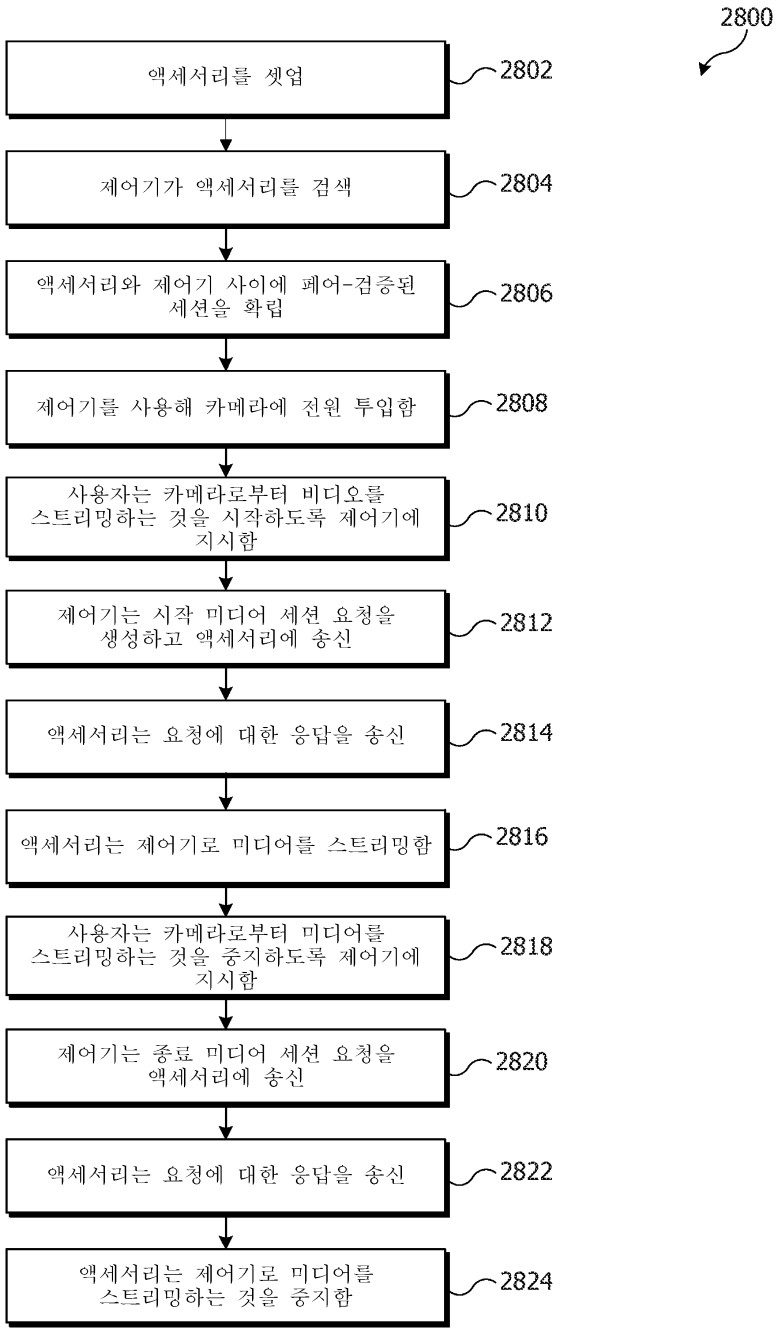
도면27c


2700
2704
// Continued from previous page
{
"type" : "com.proto.ch.video.attributes",
"value" : {
"imgattr" : "send [x=1280,y=720]"
},
"perms" : ["PR"],
"instanceID" : 13
},
{
"type" : "com.proto.ch.video.payload-type",
"value" : 99,
"perms" : ["PR"],
"instanceID" : 14
},
{
"type" : "com.proto.ch.rtp.protocol",
"value" : "RTP/SAVPF",
"perms" : ["PR"],
"instanceID" : 15
},
{
"type" : "com.proto.ch.rtp.extensions",
"value" : ["tstr", "tmmbr"],
"perms" : ["PR"],
"instanceID" : 16
},
{
"type" : "com.proto.ch.srtp.crypto-suite",
"value" : "AES_256_CM_HMAC_SHA1_80",
"perms" : ["PR"],
"instanceID" : 17
}
]
},
// Continued next page

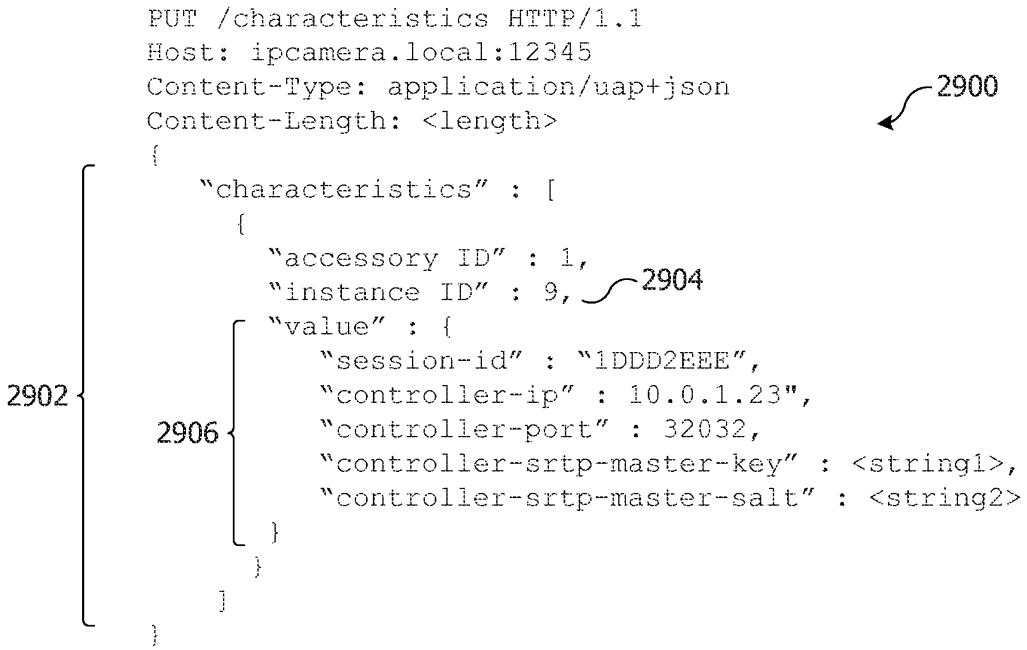
도면27d


2700
2706
2708
// Continued from previous page
{
"type" : "com.proto.svc.camera",
"instanceID" = 18,
"characteristics" : [
{
"type" : "com.proto.ch.on",
"perms" : ["PR", "PW"],
"value" : true,
"instanceID" : 19
},
{
"type" : "com.proto.ch.camera.night-vision",
"perms" : ["PR", "PW"],
"value" : false,
"instanceID" : 20
},
{
"type" : "com.proto.ch.camera.zoom",
"perms" : ["PR", "PW"],
"value" : 1.0,
"instanceID" : 21
},]
}
{
"type" = com.proto.svc.microphone",
"instanceID" = 22,
"characteristics" : [
{
"type" : "com.proto.ch.on",
"perms" : ["PR", "PW"],
"value" : true,
"instanceID" : 23
}
]
}
}
}
// Continued next page

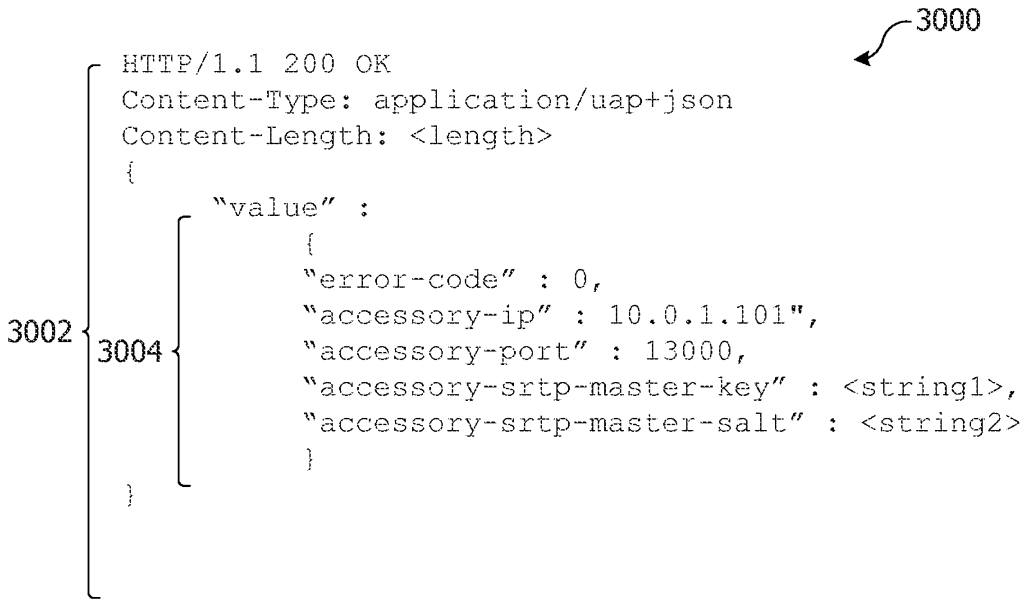
도면28


2800
액세서리를 셋업
2802
제어기가 액세서리를 검색
2804
액세서리와 제어기 사이에 페어-검증된 세션을 확립
2806
제어기를 사용해 카메라에 전원 투입함
2808
사용자는 카메라로부터 비디오를 스트리밍하는 것을 시작하도록 제어기에 지시함
2810
제어기는 시작 미디어 세션 요청을 생성하고 액세서리에 송신
2812
액세서리는 요청에 대한 응답을 송신
2814
액세서리는 제어기로 미디어를 스트리밍함
2816
사용자는 카메라로부터 미디어를 스트리밍하는 것을 중지하도록 제어기에 지시함
2818
제어기는 종료 미디어 세션 요청을 액세서리에 송신
2820
액세서리는 요청에 대한 응답을 송신
2822
액세서리는 제어기로 미디어를 스트리밍하는 것을 중지함
2824

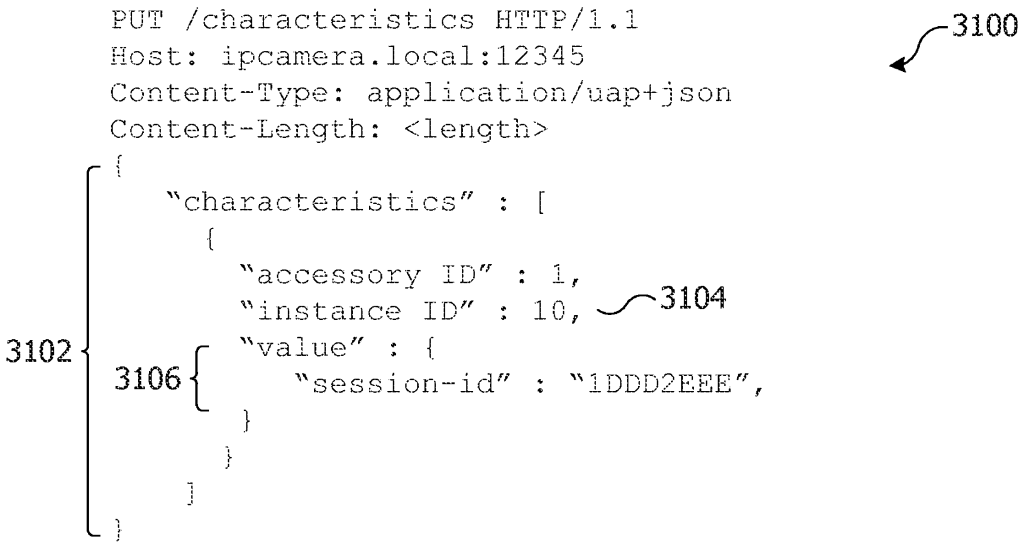
도면29


PUT /characteristics HTTP/1.1
Host: ipcamera.local:12345
Content-Type: application/uap+json
Content-Length: <length>
{
"characteristics" : [
{
"accessory ID" : 1,
"instance ID" : 9,
"value" : {
"session-id" : "1DDD2EEE",
"controller-ip" : 10.0.1.23",
"controller-port" : 32032,
"controller-srtp-master-key" : <string1>,
"controller-srtp-master-salt" : <string2>
}
}
]
}
2900
2902
2904
2906

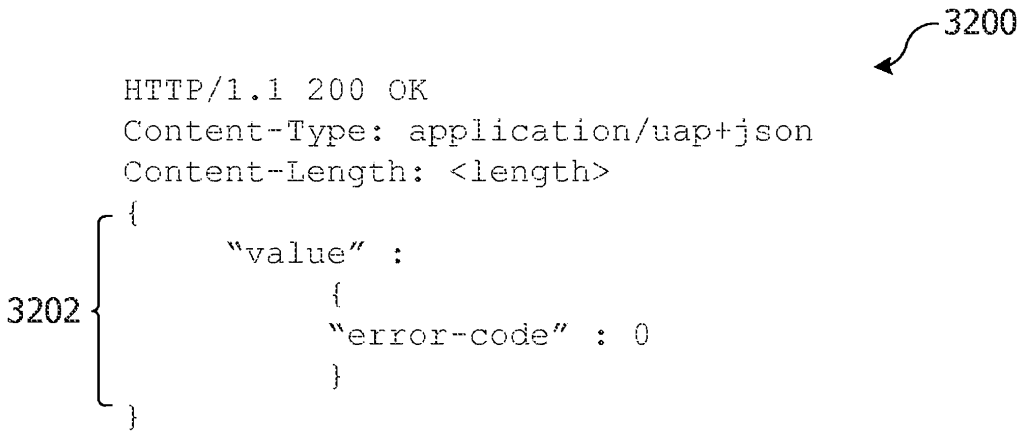
도면30


HTTP/1.1 200 OK
Content-Type: application/uap+json
Content-Length: <length>
{
"value" :
{
"error-code" : 0,
"accessory-ip" : 10.0.1.101",
"accessory-port" : 13000,
"accessory-srtp-master-key" : <string1>,
"accessory-srtp-master-salt" : <string2>
}
}
3000
3002
3004

도면31



도면32



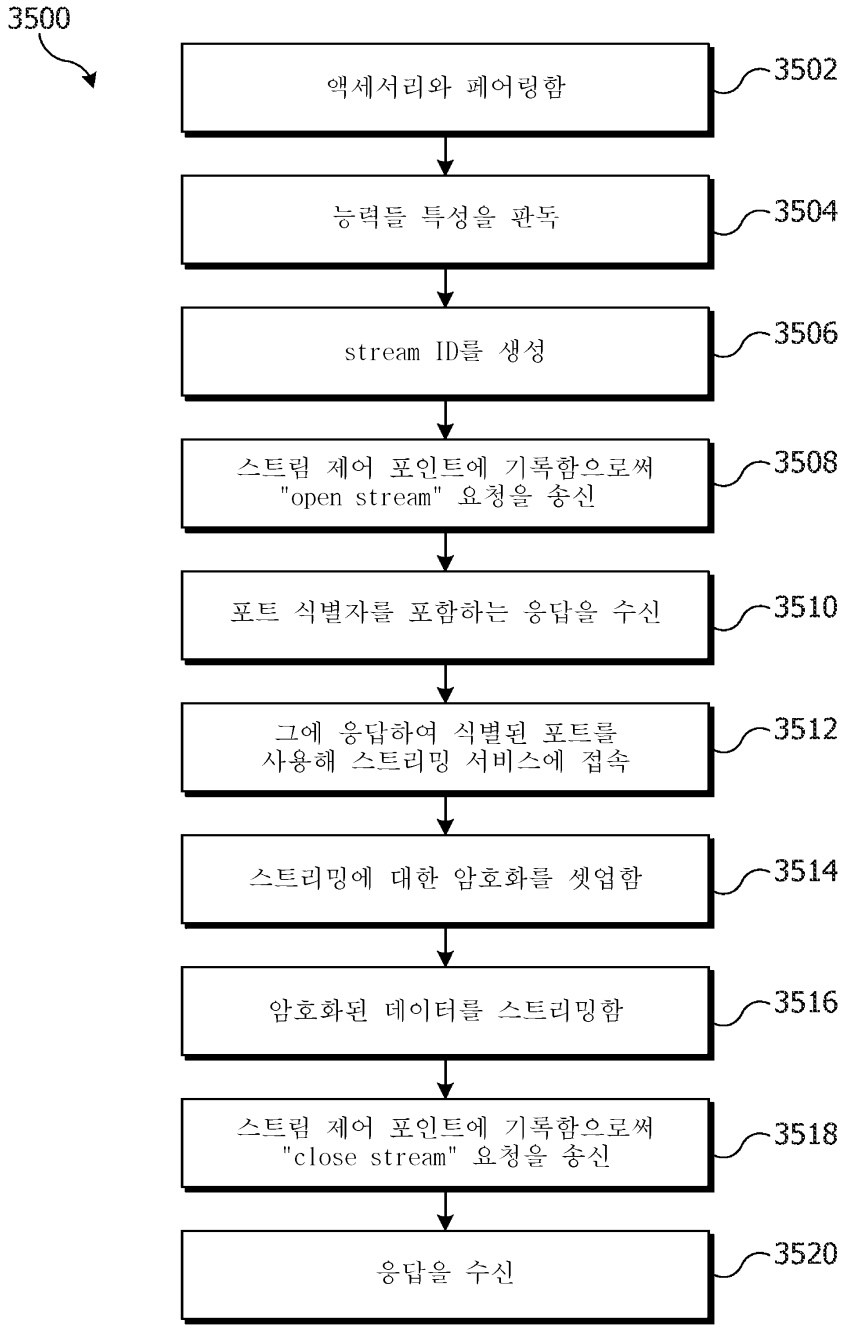
도면33

서비스	속성	값
IP 스트리밍	타입	com.proto.svc.stream
	특성	com.proto.ch.stream.capability
		com.proto.ch.stream.control.input
3301		com.proto.ch.stream.control.result

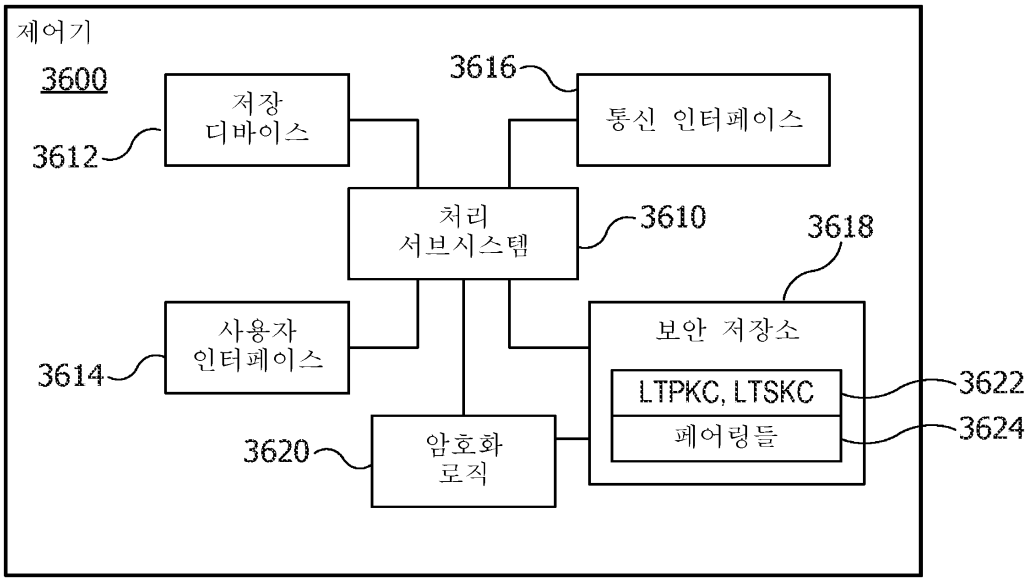
도면34

<table>
<tr><th>특성</th><th>속성</th><th>값</th></tr>
<tr><td rowspan="3">스트리밍 능력들
3401</td><td>타입</td><td colspan="2">com.proto.ch.stream.capabilities</td></tr>
<tr><td>허가들</td><td colspan="2">Paired Read</td></tr>
<tr><td>데이터 객체</td><td>protocol</td><td>1=TCP, 2=UDP</td></tr>
<tr><td></td><td></td><td>protocol-info</td><td><object></td></tr>
<tr><td rowspan="4">스트리밍 제어 입력
3402</td><td>타입</td><td colspan="2">com.proto.ch.stream.control.input</td></tr>
<tr><td>허가들</td><td colspan="2">Paired Write</td></tr>
<tr><td rowspan="3">데이터 객체</td><td>request-type</td><td>1=open, 2=close</td></tr>
<tr><td>stream-id</td><td><string></td></tr>
<tr><td>protocol-info</td><td><object></td></tr>
<tr><td rowspan="7">스트리밍 제어 결과
3403</td><td>타입</td><td colspan="2">com.proto.ch.stream.control.result</td></tr>
<tr><td>허가들</td><td colspan="2">Paired Read, Events</td></tr>
<tr><td rowspan="5">데이터 객체</td><td>stream-id</td><td><string></td></tr>
<tr><td>port</td><td><integer></td></tr>
<tr><td>transaction-id</td><td><string></td></tr>
<tr><td>status-code</td><td><integer></td></tr>
<tr><td>protocol-info</td><td><object></td></tr>
</table>

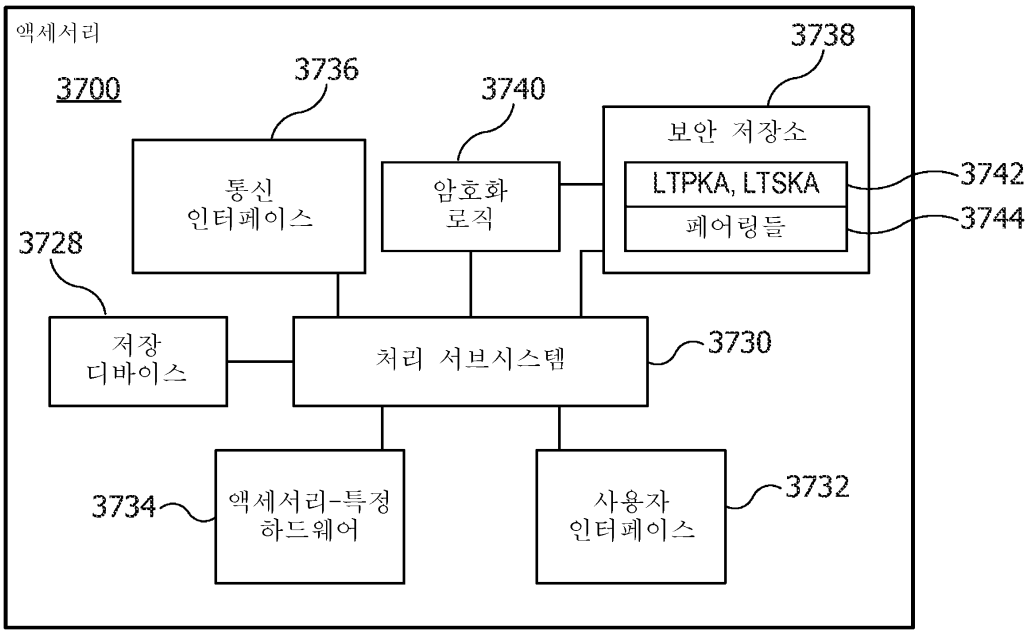
도면35


3500
액세서리와 페어링함
3502
능력들 특성을 판독
3504
stream ID를 생성
3506
스트림 제어 포인트에 기록함으로써 "open stream" 요청을 송신
3508
포트 식별자를 포함하는 응답을 수신
3510
그에 응답하여 식별된 포트를 사용해 스트리밍 서비스에 접속
3512
스트리밍에 대한 암호화를 셋업함
3514
암호화된 데이터를 스트리밍함
3516
스트림 제어 포인트에 기록함으로써 "close stream" 요청을 송신
3518
응답을 수신
3520

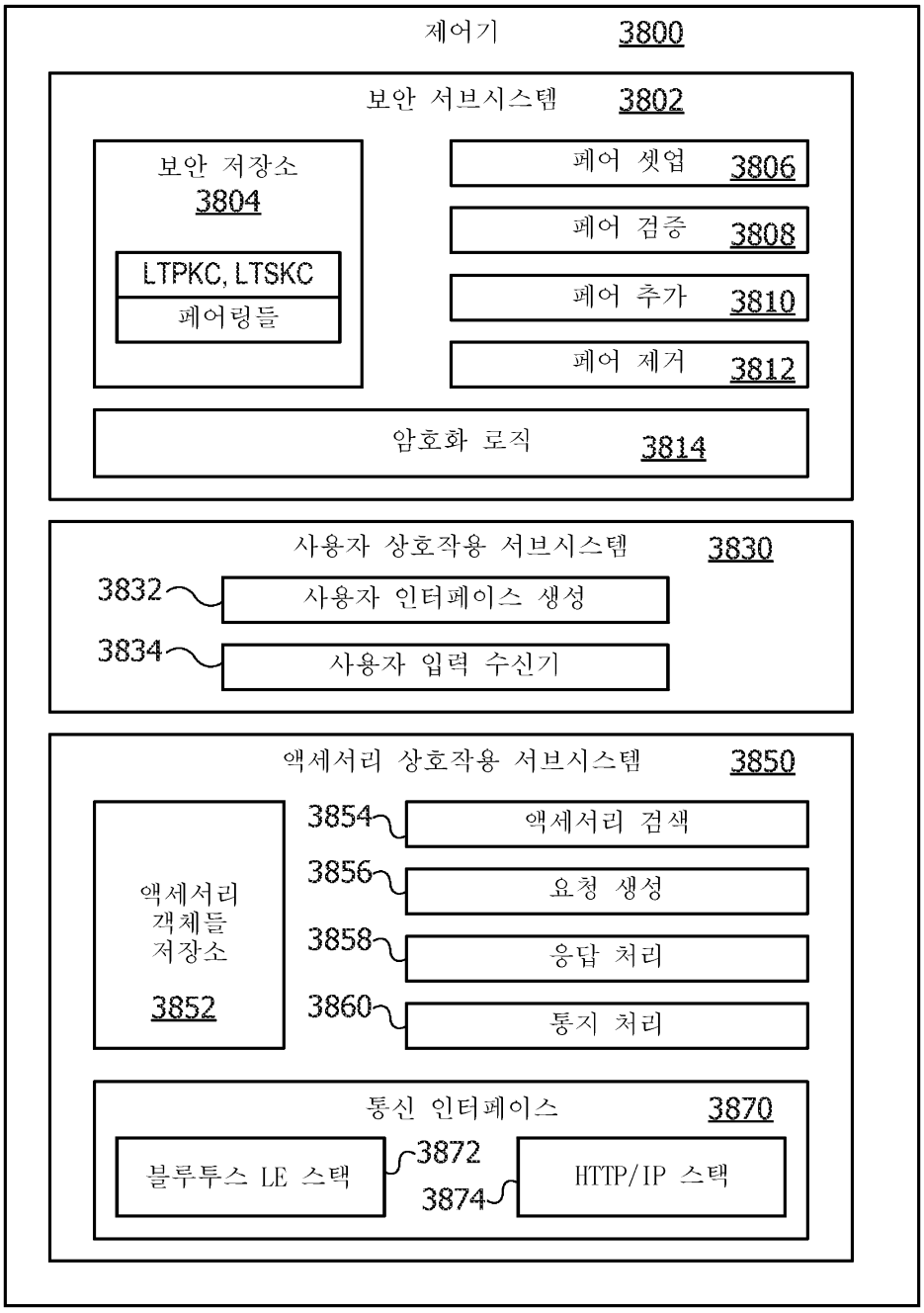
도면36


제어기
3600
저장 디바이스
3612
3616
통신 인터페이스
처리 서브시스템
3610
3618
보안 저장소
사용자 인터페이스
3614
LTPKC, LTSKC
3622
페어링들
3624
암호화 로직
3620

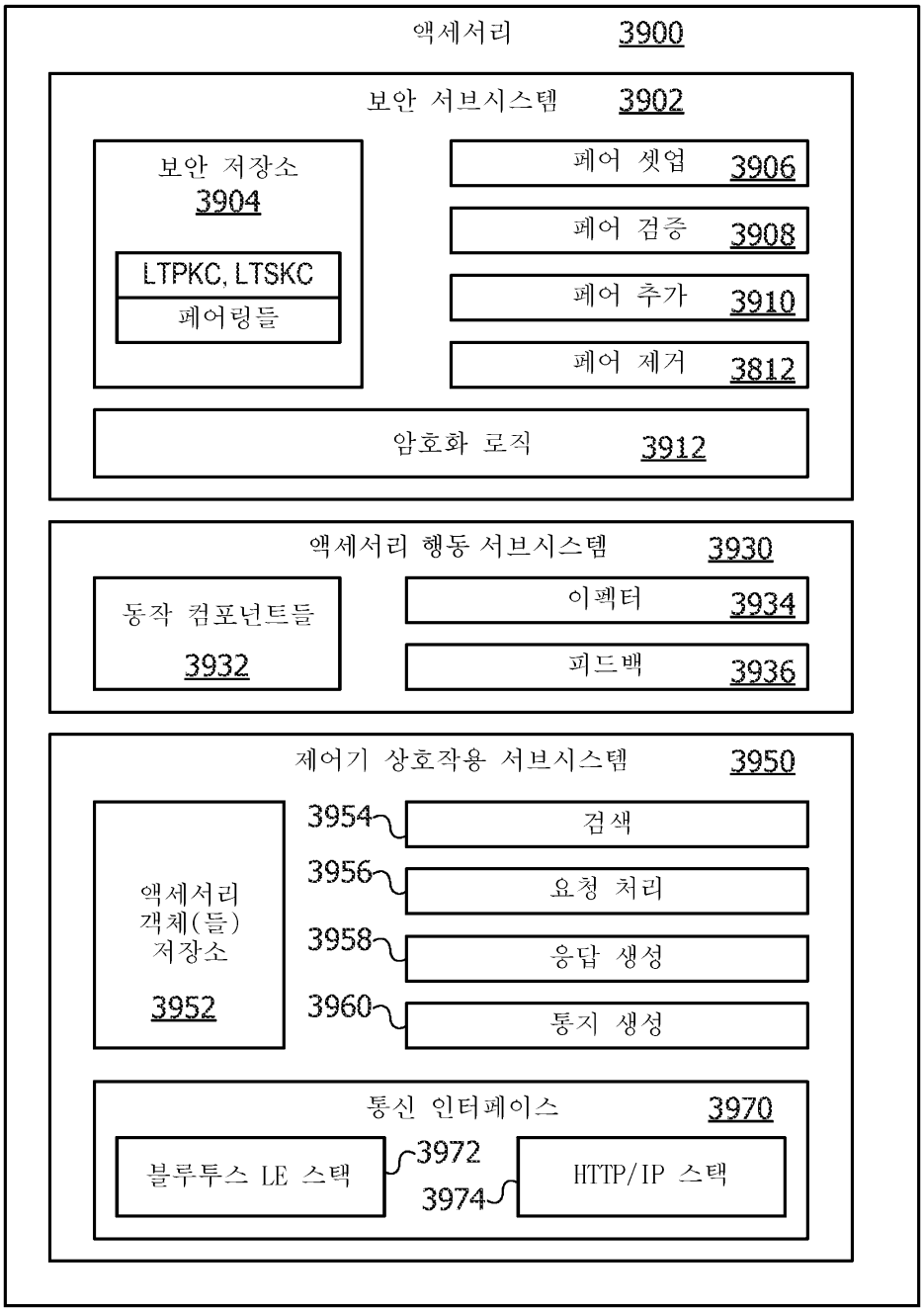
도면37


액세서리
3700
3736
통신 인터페이스
3740
암호화 로직
3738
보안 저장소
LTPKA, LTSKA
3742
페어링들
3744
3728
저장 디바이스
처리 서브시스템
3730
3734
액세서리-특정 하드웨어
사용자 인터페이스
3732

도면38


제어기 3800
보안 서브시스템 3802
보안 저장소 3804
LTPKC, LTSKC
페어링들
페어 셋업 3806
페어 검증 3808
페어 추가 3810
페어 제거 3812
암호화 로직 3814
사용자 상호작용 서브시스템 3830
3832
사용자 인터페이스 생성
3834
사용자 입력 수신기
액세서리 상호작용 서브시스템 3850
액세서리 객체들 저장소 3852
3854
액세서리 검색
3856
요청 생성
3858
응답 처리
3860
통지 처리
통신 인터페이스 3870
블루투스 LE 스택
3872
3874
HTTP/IP 스택

도면39


액세서리 3900
보안 서브시스템 3902
보안 저장소 3904
LTPKC, LTSKC
페어링들
페어 셋업 3906
페어 검증 3908
페어 추가 3910
페어 제거 3812
암호화 로직 3912
액세서리 행동 서브시스템 3930
동작 컴포넌트들 3932
이펙터 3934
피드백 3936
제어기 상호작용 서브시스템 3950
액세서리 객체(들) 저장소 3952
3954
검색
3956
요청 처리
3958
응답 생성
3960
통지 생성
통신 인터페이스 3970
블루투스 LE 스택
3972
3974
HTTP/IP 스택