(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0040887
(43) 공개일자 2025년03월25일

(51) 국제특허분류(Int. Cl.)
*H04W 12/04* (2021.01) *G06N 20/00* (2019.01)
*H04W 12/40* (2021.01) *H04W 12/50* (2021.01)
*H04W 4/70* (2018.01) *H04W 4/80* (2018.01)
*H04W 8/00* (2009.01)
(52) CPC특허분류
*H04W 12/04* (2021.01)
*G06N 20/00* (2021.08)
(21) 출원번호 10-2024-7039429
(22) 출원일자(국제) 2023년05월18일
심사청구일자 없음
(85) 번역문제출일자 2024년11월27일
(86) 국제출원번호 PCT/US2023/022788
(87) 국제공개번호 WO 2023/225233
국제공개일자 2023년11월23일
(30) 우선권주장
17/748,603 2022년05월19일 미국(US)

(71) 출원인
**어페로, 인크.**
미국 캘리포니아 로스앨토스 엘 카미노 리얼 4410 스위트 200 (우: 94022)
(72) 발명자
**로버츠, 데이브**
미국 94022 캘리포니아 로스 앨토스 엘 카미노 리얼 4410 스위트 200
**퀸, 케리**
미국 94022 캘리포니아 로스 앨토스 엘 카미노 리얼 4410 스위트 200
(74) 대리인
**특허법인(유)남아이피그룹**

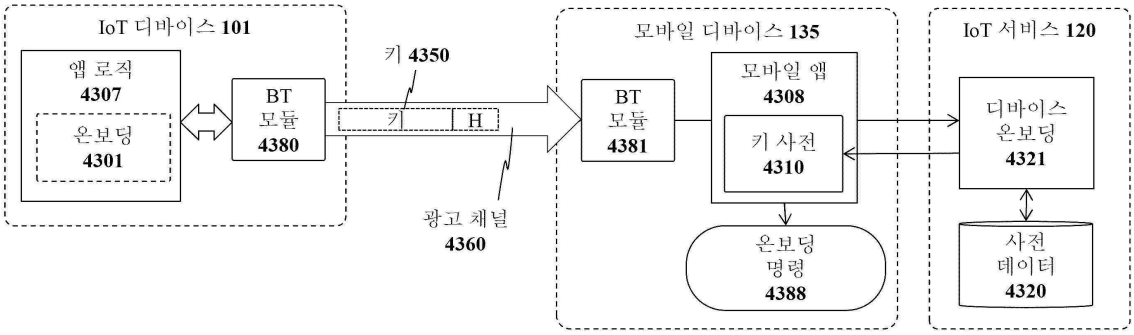
전체 청구항 수 : 총 24 항

(54) 발명의 명칭 **블루투스 광고 채널을 이용한 IoT 디바이스 식별 및 초기화 시스템 및 방법**

## (57) 요 약

광고 채널에서 키 브로드캐스트로부터 IoT 디바이스 모델을 식별하기 위한 시스템 및 방법이 기술된다. 예를 들어, 시스템의 일 실시예는 블루투스(BT) 광고 채널을 통해 키를 전송하는 타겟 사물 인터넷(IoT) 디바이스 - 키는 타겟 IoT 디바이스의 모델과 연관됨 -; 모바일 디바이스에 설치된 때 모바일 디바이스로 하여금 BT 광고 채널을 청취하여 키를 추출하게 하는 모바일 애플리케이션 프로그램 코드; 및 모바일 디바이스에 설치되며 복수의 키 각각을 IoT 디바이스 모델 및 연관된 데이터와 매핑하는 키 사전을 포함하고, 모바일 애플리케이션 프로그램 코드는, BT 광고 채널로부터 추출된 키를 사용하여 룩업을 수행하여 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 식별하고, 타겟 IoT 디바이스를 온보딩하기 위해 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 사용하는 것이다.

*대 표 도* - 도43



(52) CPC특허분류
*H04W 12/40* (2021.01)
*H04W 12/50* (2021.01)
*H04W 4/70* (2018.02)
*H04W 4/80* (2018.02)
*H04W 8/005* (2013.01)

# 명 세 서

## 청구범위

### 청구항 1

시스템으로서,

키를 블루투스(BT) 광고 채널을 통해 전송하는 타겟 사물 인터넷(IoT) 디바이스 - 상기 키는 상기 타겟 IoT 디바이스의 모델과 연관됨 -;

모바일 디바이스에 설치된 때 상기 모바일 디바이스로 하여금 상기 BT 광고 채널을 청취하여 상기 키를 추출하게 하는 모바일 애플리케이션 프로그램 코드; 및

상기 모바일 디바이스에 설치될 키 사전을 포함하고, 상기 키 사전은 복수의 키 각각을 IoT 디바이스 모델 및 연관된 데이터와 맵핑하고, 상기 모바일 애플리케이션 프로그램 코드는, 상기 BT 광고 채널로부터 추출된 상기 키를 사용하여 룩업을 수행하여 상기 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 식별하고, 상기 타겟 IoT 디바이스를 온보딩하기 위해 상기 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 사용하는, 시스템.

### 청구항 2

제1항에 있어서, 상기 연관된 데이터는, 상기 타겟 IoT 디바이스를 구성하고 상기 IoT 디바이스를 IoT 서비스 상의 사용자의 계정에 추가하기 위한 명령들을 사용자에게 제공하기 위해 상기 모바일 애플리케이션 프로그램 코드에 의해 사용 가능한, 시스템.

### 청구항 3

제2항에 있어서, 상기 연관된 데이터는 상기 명령들을 사용자에게 제공하기 위해 비디오, 오디오, 텍스트, 및/또는 이미지들을 식별하는 하나 이상의 네트워크 링크를 포함하는, 시스템.

### 청구항 4

제3항에 있어서, 상기 키 사전은 JSON 데이터 포맷으로 저장되는, 시스템.

### 청구항 5

제2항에 있어서, 다수의 키들이 다수의 BT 광고 채널들로부터 추출된 경우, 상기 모바일 애플리케이션 프로그램 코드는 상기 타겟 IoT 디바이스의 키를 식별하기 위해 특정 키들을 우선순위화 및/또는 필터링하는 것인, 시스템.

### 청구항 6

제5항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 상기 IoT 서비스 상의 사용자의 계정과 이미 연관된 IoT 디바이스들의 키들을 필터링하는 것인, 시스템.

### 청구항 7

제6항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 각각의 BT 광고 채널들 각각의 측정된 신호 강도에 기반하여 상기 키들을 우선순위화하는 것인, 시스템.

### 청구항 8

제1항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 초기 온보딩 프로세스가 이용 가능하지 않거나 성공적이지 않을 때에만 상기 모바일 디바이스로 하여금 상기 BT 광고 채널을 청취하여 상기 키를 추출하게 하는 것인, 시스템.

### 청구항 9

방법으로서,

타겟 사물 인터넷(IoT) 디바이스에 의해, 블루투스(BT) 광고 채널을 통해 상기 타겟 IoT 디바이스의 모델과 연관된 키를 전송하는 단계;

모바일 디바이스에 의해, 모바일 애플리케이션 프로그램 코드에 응답하여 상기 BT 광고 채널을 청취하고, 상기 키를 추출하는 단계;

상기 BT 광고 채널로부터 추출된 키를 사용하며 상기 모바일 디바이스 상에 설치된 키 사전에서 룩업을 수행하고, 복수의 키들 각각을 IoT 디바이스 모델 및 연관된 데이터와 맵핑하는 단계;

상기 키 사전으로부터 상기 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 식별하는 단계; 및

상기 타겟 IoT 디바이스를 온보딩하기 위해 상기 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 사용하는 단계를 포함하는 방법.

**청구항 10**

제9항에 있어서, 상기 연관된 데이터는, 상기 타겟 IoT 디바이스를 구성하고 상기 IoT 디바이스를 IoT 서비스 상의 사용자의 계정에 추가하기 위한 명령들을 사용자에게 제공하기 위해 상기 모바일 애플리케이션 프로그램 코드에 의해 사용 가능한, 방법.

**청구항 11**

제10항에 있어서, 상기 연관된 데이터는 상기 명령들을 사용자에게 제공하기 위해 비디오, 오디오, 텍스트, 및/또는 이미지들을 식별하는 하나 이상의 네트워크 링크를 포함하는, 방법.

**청구항 12**

제11항에 있어서, 상기 키 사전은 JSON 데이터 포맷으로 저장되는, 방법.

**청구항 13**

제10항에 있어서, 다수의 키들이 다수의 BT 광고 채널들로부터 추출되는 경우, 상기 모바일 애플리케이션 프로그램 코드는 상기 타겟 IoT 디바이스의 키를 식별하기 위해 특정 키들을 우선순위화 및/또는 필터링하는 것인, 방법.

**청구항 14**

제13항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 상기 IoT 서비스 상의 상기 사용자의 계정과 이미 연관된 IoT 디바이스들의 키들을 필터링하는 것인, 방법.

**청구항 15**

제14항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 각각의 BT 광고 채널들 각각의 측정된 신호 강도에 기반하여 상기 키들을 우선순위화하는 것인, 방법.

**청구항 16**

제9항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 초기 온보딩 프로세스가 이용 가능하지 않거나 성공적이지 않은 경우에만 상기 모바일 디바이스로 하여금 상기 BT 광고 채널을 청취하여 상기 키를 추출하게 하는 것인, 방법.

**청구항 17**

시스템으로서,

키를 블루투스(BT) 광고 채널을 통해 전송하는 타겟 사물 인터넷(IoT) 디바이스 - 상기 키는 상기 타겟 IoT 디바이스의 모델과 연관됨 -;

모바일 디바이스에 설치된 때 상기 모바일 디바이스로 하여금 상기 BT 광고 채널을 청취하여 상기 키를 추출하

게 하는 모바일 애플리케이션 프로그램 코드; 및

상기 모바일 애플리케이션 프로그램 코드에 접속되며 상기 모바일 디바이스에 키 사전을 설치 및/또는 업데이트하는 IoT 서비스를 포함하고, 상기 키 사전은 복수의 키 각각을 상기 IoT 디바이스 모델 및 연관된 데이터와 매핑하고,

상기 모바일 애플리케이션 프로그램 코드는, 상기 BT 광고 채널로부터 추출된 키를 사용하여 룩업을 수행하여 상기 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 식별하고, 상기 타겟 IoT 디바이스를 온보딩하기 위해 상기 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 사용하는, 시스템.

**청구항 18**

제17항에 있어서, 상기 연관된 데이터는, 상기 타겟 IoT 디바이스를 구성하고 상기 IoT 디바이스를 IoT 서비스 상의 사용자의 계정에 추가하기 위한 명령들을 사용자에게 제공하기 위해 상기 모바일 애플리케이션 프로그램 코드에 의해 사용 가능한, 시스템.

**청구항 19**

제18항에 있어서, 상기 연관된 데이터는 사용자에게 명령으로서 제공될 상기 IoT 서비스 상의 비디오, 오디오, 텍스트, 및/또는 이미지들을 식별하는 하나 이상의 네트워크 링크를 포함하는, 시스템.

**청구항 20**

제19항에 있어서, 상기 키 사전은 JSON 데이터 포맷으로 저장되는, 시스템.

**청구항 21**

제18항에 있어서, 다수의 키들이 다수의 BT 광고 채널들로부터 추출된 경우, 상기 모바일 애플리케이션 프로그램 코드는 상기 타겟 IoT 디바이스의 키를 식별하기 위해 특정 키들을 우선순위화 및/또는 필터링하는 것인, 시스템.

**청구항 22**

제21항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 상기 IoT 서비스 상의 사용자의 계정과 이미 연관된 IoT 디바이스들의 키들을 필터링하는 것인, 시스템.

**청구항 23**

제22항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 각각의 BT 광고 채널들 각각의 측정된 신호 강도에 기반하여 상기 키들을 우선순위화하는 것인, 시스템.

**청구항 24**

제17항에 있어서, 상기 모바일 애플리케이션 프로그램 코드는 초기 온보딩 프로세스가 이용 가능하지 않거나 성공적이지 않을 때에만 상기 모바일 디바이스로 하여금 상기 BT 광고 채널을 청취하여 상기 키를 추출하게 하는 것인, 시스템.

## *발명의 설명*

### *기 술 분 야*

[0001] 기술분야

[0002] 본 발명은 일반적으로 컴퓨터 시스템의 분야에 관한 것이다. 더 구체적으로는, 본 발명은 머신-러닝 기반 IoT 디바이스 프로비저닝 시스템 및 방법에 관한 것이다.

### *배 경 기 술*

[0003] 배경기술

[0004] "사물 인터넷"은 인터넷 기반구조 내의 고유하게 식별 가능한 임베디드 디바이스들의 상호접속을 지칭한다. 궁

극적으로, IoT는, 사실상 임의의 타입의 물리적인 물건이 그 자체 또는 그의 주변에 대한 정보를 제공할 수 있고 그리고/또는 인터넷을 통하여 클라이언트 디바이스를 통해 원격으로 제어될 수 있는 새로운 광범위한 타입의 애플리케이션을 생성할 것으로 예상된다.

## *도면의 간단한 설명*

[0005] 아래의 도면들과 관련된 아래의 상세한 설명으로부터 본 발명의 더 양호한 이해를 달성할 수 있다.

**도 1a 및 도 1b**는 IoT 시스템 아키텍처의 상이한 실시예들을 예시한다.

**도 2**는 본 발명의 일 실시예에 따른 IoT 디바이스를 예시한다.

**도 3**은 본 발명의 일 실시예에 따른 IoT 허브를 예시한다.

**도 4a 및 도 4b**는 IoT 디바이스들로부터 데이터를 제어 및 수집하고 통지를 생성하기 위한 본 발명의 실시예들을 예시한다.

**도 5**는 IoT 디바이스들로부터 데이터를 수집하고 IoT 허브 및/또는 IoT 서비스로부터 통지를 생성하기 위한 본 발명의 실시예들을 예시한다.

**도 6**은 중개 모바일 디바이스가 고정 IoT 디바이스로부터 데이터를 수집하고 데이터를 IoT 허브에 제공하는 시스템의 일 실시예를 예시한다.

**도 7**은 본 발명의 일 실시예에서 구현되는 중개 접속 로직을 예시한다.

**도 8**은 본 발명의 일 실시예에 따른 방법을 예시한다.

**도 9a**는 프로그램 코드 및 데이터 업데이트들이 IoT 디바이스에 제공되는 일 실시예를 예시한다.

**도 9b**는 프로그램 코드 및 데이터 업데이트들이 IoT 디바이스에 제공되는 방법의 일 실시예를 예시한다.

**도 10**은 보안 아키텍처의 일 실시예의 고수준 도면을 예시한다.

**도 11**은 IoT 디바이스에 키를 저장하는 데 가입자 식별 모듈(SIM)이 사용되는 아키텍처의 일 실시예를 예시한다.

**도 12a**는 IoT 디바이스가 바코드 또는 QR 코드를 사용하여 등록되는 일 실시예를 예시한다.

**도 12b**는 바코드 또는 QR 코드를 사용하여 페어링이 수행되는 일 실시예를 예시한다.

**도 13**은 IoT 허브를 사용하여 SIM을 프로그래밍하는 방법의 일 실시예를 예시한다.

**도 14**는 IoT 디바이스를 IoT 허브 및 IoT 서비스에 등록하는 방법의 일 실시예를 예시한다.

**도 15**는 IoT 디바이스로 전송될 데이터를 암호화하는 방법의 일 실시예를 예시한다.

**도 16a 및 도 16b**는 IoT 서비스와 IoT 디바이스 사이에서 데이터를 암호화하는 본 발명의 상이한 실시예들을 예시한다.

**도 17**은 보안 키 교환을 수행하고, 공통 비밀을 생성하고, 비밀을 사용하여 키 스트림을 생성하는 본 발명의 실시예들을 예시한다.

**도 18**은 본 발명의 일 실시예에 따른 패킷 구조를 예시한다.

**도 19**는 IoT 디바이스와 정식으로 페어링함이 없이 IoT 디바이스에 데이터를 기입하고 그로부터 데이터를 판독하기 위해 일 실시예에서 사용되는 기술을 예시한다.

**도 20**은 본 발명의 일 실시예에서 사용되는 예시적인 커맨드 패킷 세트를 예시한다.

**도 21**은 커맨드 패킷들을 사용하는 예시적인 트랜잭션 시퀀스를 예시한다.

**도 22**는 본 발명의 일 실시예에 따른 방법을 나타낸다.

**도 23a 내지 도 23c**는 본 발명의 일 실시예에 따른 보안 페어링 방법을 예시한다.

**도 24**는 데이터 전송 조건을 식별하기 위해 광고 간격을 조정하는 본 발명의 일 실시예를 예시한다.

**도 25**는 본 발명의 일 실시예에 따른 방법을 예시한다.

**도 26a 내지 도 26c**는 다수의 IoT 허브가 IoT 디바이스에 데이터/커맨드를 송신하려고 시도하는 일 실시예의 동작을 예시한다.

**도 27**은 본 발명의 일 실시예에 따른 방법을 예시한다.

**도 28**은 보안 IoT 디바이스 프로비저닝을 위한 시스템의 일 실시예를 예시한다.

**도 29**는 본 발명의 일 실시예에 따른 방법을 예시한다.

**도 30**은 복수의 IoT 디바이스에 대한 흐름 제어를 수행하기 위한 시스템의 일 실시예를 예시한다.

**도 31**은 본 발명의 일 실시예에 따른 방법을 예시한다.

**도 32**는 애플리케이션 속성들, 시스템 속성들, 및 우선순위 통지 속성들을 관리하기 위한 시스템의 일 실시예를 예시한다.

**도 33 및 도 34**는 익명 계정을 생성 및 변환하기 위한 본 발명의 실시예들을 예시한다.

**도 35**는 IoT 디바이스들을 익명으로 구성 및 제어하기 위한 방법 및 사용자 인터페이스를 예시한다.

**도 36a 및 도 36b**는 제1 사용자가 IoT 디바이스들을 구성하고 IoT 디바이스들을 제2 사용자의 계정으로 전송하는 일 실시예를 예시한다.

**도 37a 및 도 37b**와 **도 38a 및 도 38b**는 본 발명의 다양한 실시예들을 예시하는 트랜잭션 선도이다.

**도 39**는 제1 사용자(사용자 A)가 하나 이상의 IoT 디바이스를 제2 사용자(사용자 B)에게 대출하는 일련의 트랜잭션을 예시한다.

**도 40**은 머신-러닝(ML) 기반 IoT 디바이스 검출 및 프로비저닝을 위한 프로세스의 일 실시예를 예시한다.

**도 41**은 머신-러닝(ML) 기반 IoT 디바이스 검출 및 프로비저닝을 위한 아키텍처의 일 실시예를 예시한다.

**도 42**는 본 발명의 일 실시예에 따른 방법을 예시한다.

**도 43은** 블루투스 광고 패킷으로부터 수신된 키를 사용하여 설정 정보를 동적으로 검색하는 일 실시예를 예시한다.

**도 44**는 본 발명의 일 실시예에 따른 방법을 예시한다.

## *발명을 실시하기 위한 구체적인 내용*

[0006] 아래의 설명에서는 아래에 설명되는 본 발명의 실시예들에 대한 완전한 이해를 제공하기 위해 여러 특정 세부사항들이 설명의 목적으로 기재된다. 그러나 본 발명의 실시예들은 이러한 특정 세부사항들 중 일부가 없이도 실시될 수 있다는 것이 당업자에게 명백할 것이다. 다른 경우에서, 잘 알려진 구조 및 디바이스는 본 발명의 실시예의 기본 원리를 불명확하게 하는 것을 피하기 위해 블록선도 형태로 도시된다.

[0007] 본 발명의 일 실시예는 개발자가 새로운 IoT 디바이스 및 애플리케이션을 설계 및 구축하기 위해 활용할 수 있는 사물 인터넷(IoT) 플랫폼을 포함한다. 특히, 일 실시예는 IoT 디바이스들을 인터넷에 연결되게 하는 미리 정의된 네트워킹 프로토콜 스택 및 IoT 허브를 포함하여, IoT 디바이스들을 위한 기반 하드웨어/소프트웨어 플랫폼을 포함한다. 부가적으로, 일 실시예는 IoT 허브들 및 접속된 IoT 디바이스들에 아래에 설명되는 바와 같이 액세스하고 그들을 관리할 수 있는 IoT 서비스를 포함한다. 부가적으로, IoT 플랫폼의 일 실시예는 IoT 서비스, 허브, 및 접속된 디바이스들에 액세스하고 그들을 구성하기 위한 (예를 들어, 클라이언트 디바이스에서 실행되는) IoT 앱 또는 웹 애플리케이션을 포함한다. 기존의 온라인 소매상 및 다른 웹사이트 운영자는 고유 IoT 기능을 기존의 사용자 기반에 쉽게 제공하기 위해 본 명세서에 설명된 IoT 플랫폼을 활용할 수 있다.

[0008] 도 1a는 본 발명의 실시예가 구현될 수 있는 아키텍처 플랫폼의 개요를 예시한다. 특히, 예시된 실시예는 인터넷(220)을 통해 IoT 서비스(120)에 통신 가능하게 스스로 연결된 중앙 IoT 허브(110)에 로컬 통신 채널(130)을 통해 통신 가능하게 연결된 복수의 IoT 디바이스(101 내지 105)를 포함한다. IoT 디바이스(101 내지 105) 각각은 로컬 통신 채널들(130) 각각을 인에이블하기 위해 (예를 들어, 아래에 설명되는 페어링 기술을 사용하여) 초기에 IoT 허브(110)에 페어링될 수 있다. 일 실시예에서, IoT 서비스(120)는 각각의 사용자의 IoT 디바이스로

부터 수집된 사용자 계정 정보 및 데이터를 유지하기 위한 최종 사용자 데이터베이스(122)를 포함한다. 예를 들어, IoT 디바이스가 센서(예를 들어, 온도 센서, 가속도계, 열 센서, 모션 검출기 등)를 포함하는 경우, 데이터베이스(122)는 IoT 디바이스(101 내지 105)에 의해 수집된 데이터를 저장하도록 계속 업데이트될 수 있다. 데이터베이스(122)에 저장된 데이터는 그런 다음에는 사용자의 디바이스(135)에 설치된 IoT 앱 또는 브라우저를 통해(또는 데스크톱 또는 다른 클라이언트 컴퓨터 시스템을 통해) 최종 사용자에 의해 그리고 (예를 들어, IoT 서비스(120)에 가입한 웹사이트(130)와 같은) 웹 클라이언트에 의해 액세스 가능하게 될 수 있다.

[0009] IoT 디바이스(101 내지 105)에는 그들 및 그들의 주변에 대한 정보를 수집하고 그 수집된 정보를 IoT 허브(110)를 통해 IoT 서비스(120), 사용자 디바이스(135), 및/또는 외부 웹사이트(130)에 제공하기 위한 다양한 타입의 센서가 탑재될 수 있다. IoT 디바이스(101 내지 105) 중 일부는 IoT 허브(110)를 통해 전송된 제어 커맨드에 응답하여 지정된 기능을 수행할 수 있다. IoT 디바이스(101 내지 105)에 의해 수집된 정보 및 제어 커맨드의 다양한 특정 예가 아래에 제공된다. 아래에 설명된 일 실시예에서, IoT 디바이스(101)는, 사용자 선택을 기록하고 사용자 선택을 IoT 서비스(120) 및/또는 웹사이트에 전송하도록 설계된 사용자 입력 디바이스이다.

[0010] 일 실시예에서, IoT 허브(110)는 4G(예를 들어, 모바일 WiMAX, LTE) 또는 5G 셀룰러 데이터 서비스와 같은 셀룰러 서비스(115)를 통해 인터넷(220)으로의 접속을 확립하기 위한 셀룰러 라디오를 포함한다. 대안적으로 또는 부가적으로, IoT 허브(110)는 (예를 들어, 인터넷 서비스를 최종 사용자에게 제공하는 인터넷 서비스 제공자를 통해) IoT 허브(110)를 인터넷에 연결시키는 WiFi 액세스 포인트 또는 라우터(116)를 통해 WiFi 접속을 확립하기 위한 WiFi 라디오를 포함할 수 있다. 물론, 본 발명의 기본 원리는 임의의 특정 타입의 통신 채널 또는 프로토콜로 제한되지 않는다는 것에 유의하여야 한다.

[0011] 일 실시예에서, IoT 디바이스(101 내지 105)는 배터리 전력으로 장기간(예를 들어, 수년) 동안 동작할 수 있는 초 저전력 디바이스이다. 전력을 보전하기 위해, 로컬 통신 채널(130)은 블루투스 저에너지(LE)와 같은 저전력 무선 통신 기술을 사용하여 구현될 수 있다. 이러한 실시예에서, IoT 디바이스(101 내지 105) 각각 및 IoT 허브(110)에는 블루투스 LE 라디오 및 프로토콜 스택이 탑재된다.

[0012] 언급된 바와 같이, 일 실시예에서, IoT 플랫폼은 사용자가 접속된 IoT 디바이스(101 내지 105), IoT 허브(110), 및/또는 IoT 서비스(120)에 액세스하고 그들을 구성할 수 있도록 하기 위한, 사용자 디바이스(135)에서 실행되는 IoT 앱 또는 웹 애플리케이션을 포함한다. 일 실시예에서, 상기 앱 또는 웹 애플리케이션은 웹사이트(130)의 운영자에 의해 그의 사용자 베이스에 IoT 기능을 제공하도록 설계될 수 있다. 예시된 바와 같이, 웹사이트는 각 사용자에 관련된 계정 기록을 포함하는 사용자 데이터베이스(131)를 유지할 수 있다.

[0013] **도 1b**는 복수의 IoT 허브(110, 111, 190)에 대한 추가 접속 옵션을 예시한다. 이 실시예에서, 단일 사용자는 단일 사용자 구내(180)(예를 들어, 사용자의 집 또는 회사)에 다수의 허브(110, 111)가 현장에 설치되게 할 수 있다. 이는 예를 들어 IoT 디바이스(101 내지 105) 모두를 접속시키는 데 필요한 무선 범위를 확장시키기 위해 행해질 수 있다. 나타낸 바와 같이, 사용자가 다수의 허브(110, 111)를 갖는 경우, 그 허브들은 로컬 통신 채널(예를 들어, Wifi, 이더넷, 전력 라인 네트워킹 등)을 통해 접속될 수 있다. 일 실시예에서, 허브(110, 111) 각각은 셀룰러(115) 또는 WiFi(116) 접속(**도 1b**에 명시적으로 도시되지 않음)을 통해 IoT 서비스(120)로의 직접 접속을 확립할 수 있다. 대안적으로 또는 부가적으로, IoT 허브(110)와 같은 IoT 허브들 중 하나는 (IoT 허브(110)와 IoT 허브(111)를 연결하는 점선으로 표시된 바와 같이) IoT 허브(111)와 같은 사용자 구내(180)의 다른 IoT 허브들 모두에 접속 및/또는 로컬 서비스를 제공하는 "마스터" 허브로 동작할 수 있다. 예를 들어, 마스터 IoT 허브(110)는 IoT 서비스(120)로의 직접 접속을 확립하기 위한 유일한 IoT 허브일 수 있다. 일 실시예에서, "마스터" IoT 허브(110)에만 IoT 서비스(120)로의 접속을 확립하기 위한 셀룰러 통신 인터페이스가 탑재된다. 그렇기 때문에, IoT 서비스(120)와 다른 IoT 허브(111) 사이의 모든 통신은 마스터 IoT 허브(110)를 통해 흐를 것이다. 이러한 역할에서, 마스터 IoT 허브(110)는 다른 IoT 허브(111)와 (예를 들어, 가능한 경우 일부 데이터 요청들을 로컬에서 서비스하는) IoT 서비스(120) 사이에서 교환되는 데이터에 대해 필터링 동작을 수행하기 위한 추가 프로그램 코드를 제공받을 수 있다.

[0014] IoT 허브(110, 111)가 어떻게 접속되는지에 관계없이, 일 실시예에서, IoT 서비스(120)는 앱이 설치된 사용자 디바이스(135)를 통해 액세스 가능한 단일의 포괄적인 사용자 인터페이스(및/또는 브라우저 기반 인터페이스) 하에서 허브를 사용자와 논리적으로 연관시키고 부착된 IoT 디바이스(101 내지 105) 모두를 결합시킬 것이다.

[0015] 이러한 실시예에서, 마스터 IoT 허브(110) 및 하나 이상의 슬레이브 IoT 허브(111)는 WiFi 네트워크(116), 이더넷 네트워크, 및/또는 사용 전력-라인 통신(PLC) 네트워킹일 수 있는 로컬 네트워크를 통해 접속될 수 있다(예를 들어, 여기서 네트워크의 전부 또는 일부가 사용자의 전력 라인을 통해 구동됨). 부가적으로, IoT 허브

(110, 111)에 대해, IoT 디바이스(101 내지 105) 각각은, 몇몇 예를 들자면, WiFi, 이더넷, PLC, 또는 블루투스 LE와 같은 임의의 타입의 로컬 네트워크 채널을 사용하여 IoT 허브(110, 111)와 상호접속될 수 있다.

[0016] **도 1b**는 제2 사용자 구내(181)에 설치된 IoT 허브(190)도 보여주고 있다. 사실상 제한되지 않는 수의 그러한 IoT 허브(190)가 전세계의 사용자 구내의 IoT 디바이스(191, 192)로부터 데이터를 수집하도록 설치 및 구성될 수 있다. 일 실시예에서, 2개의 사용자 구내(180, 181)가 동일한 사용자에 대해 구성될 수 있다. 예를 들어, 하나의 사용자 구내(180)는 사용자의 주된 집일 수 있고, 다른 사용자 구내(181)는 사용자의 별장일 수 있다. 그러한 경우에, IoT 서비스(120)는 앱이 설치된 사용자 디바이스(135)를 통해 액세스 가능한 단일의 포괄적인 사용자 인터페이스(및/또는 브라우저 기반 인터페이스) 하에서 IoT 허브(110, 111, 190)를 사용자와 논리적으로 연관시키고 부착된 IoT 디바이스(101 내지 105, 191, 192) 모두를 결합시킬 것이다.

[0017] **도 2**에 예시된 바와 같이, IoT 디바이스(101)의 예시적인 실시예는 프로그램 코드 및 데이터(201 내지 203)를 저장하기 위한 메모리(210) 및 프로그램 코드를 실행하고 데이터를 처리하기 위한 저전력 마이크로제어기(200)를 포함한다. 메모리(210)는 동적 랜덤 액세스 메모리(DRAM)와 같은 휘발성 메모리일 수 있거나, 플래시 메모리와 같은 비휘발성 메모리일 수 있다. 일 실시예에서, 비휘발성 메모리는 영속적인 저장을 위해 사용될 수 있고, 휘발성 메모리는 런타임 시에 프로그램 코드 및 데이터의 실행을 위해 사용될 수 있다. 또한, 메모리(210)는 저전력 마이크로제어기(200) 내에 통합될 수 있거나, 버스 또는 통신 패브릭(fabric)을 통해 저전력 마이크로제어기(200)에 결합될 수 있다. 본 발명의 기본 원리는 메모리(210)의 임의의 특정 구현으로 제한되지 않는다.

[0018] 예시된 바와 같이, 프로그램 코드는 IoT 디바이스(201)에 의해 수행될 기능들의 애플리케이션별 세트를 정의하는 애플리케이션 프로그램 코드(203)와, IoT 디바이스(101)의 애플리케이션 개발자가 활용할 수 있는 미리 정의된 빌딩 블록(building block)들의 세트를 포함하는 라이브러리 코드(202)를 포함할 수 있다. 일 실시예에서, 라이브러리 코드(202)는 각각의 IoT 디바이스(101)와 IoT 허브(110) 사이의 통신을 가능하게 하기 위한 통신 프로토콜 스택(201)과 같은, IoT 디바이스를 구현하는 데 요구되는 기본 기능들의 세트를 포함한다. 언급된 바와 같이, 일 실시예에서, 통신 프로토콜 스택(201)은 블루투스 LE 프로토콜 스택을 포함한다. 이러한 실시예에서, 블루투스 LE 라디오 및 안테나(207)가 저전력 마이크로제어기(200) 내에 통합될 수 있다. 그러나, 본 발명의 기본 원리는 임의의 특정 통신 프로토콜로 제한되지 않는다.

[0019] **도 2**에 도시된 특정 실시예는 사용자 입력을 수신하고 사용자 입력을 저전력 마이크로제어기에 제공하기 위한 복수의 입력 디바이스 또는 센서(210)도 포함하며, 저전력 마이크로제어기는 애플리케이션 코드(203) 및 라이브러리 코드(202)에 따라 사용자 입력을 처리한다. 일 실시예에서, 입력 디바이스들 각각은 최종 사용자에게 피드백을 제공하기 위한 LED(209)를 포함한다.

[0020] 부가적으로, 예시된 실시예는 저전력 마이크로제어기에 전력을 공급하기 위한 배터리(208)를 포함한다. 일 실시예에서, 비충전식 코인 셀 배터리가 사용된다. 그러나, 대안적인 실시예에서, 통합형 재충전식 배터리가 사용될 수 있다(예를 들어, IoT 디바이스를 AC 전력 공급부(도시되지 않음)에 접속시킴으로써 재충전 가능함).

[0021] 오디오를 생성하기 위한 스피커(205)도 제공된다. 일 실시예에서, 저전력 마이크로제어기(299)는 스피커(205)에서 오디오를 생성하기 위해 (예를 들어, MPEG-4/어드밴스드 오디오 코딩(AAC) 스트림과 같은) 압축된 오디오 스트림을 디코딩하기 위한 오디오 디코딩 로직을 포함한다. 대안적으로, 저전력 마이크로제어기(200) 및/또는 애플리케이션 코드/데이터(203)는 최종 사용자가 입력 디바이스(210)를 통해 선택을 입력할 때 그 최종 사용자에게 언어 피드백을 제공하기 위한 오디오의 디지털 샘플링된 단편(snippet)을 포함할 수 있다.

[0022] 일 실시예에서, IoT 디바이스(101)에는 이 IoT 디바이스(101)를 설계한 목적의 대상인 특정 애플리케이션에 기반하여 하나 이상의 다른/대안적인 I/O 디바이스 또는 센서(250)가 포함될 수 있다. 예를 들어, 온도, 압력, 습도 등을 측정하기 위한 환경 센서가 포함될 수 있다. IoT 디바이스가 보안 디바이스로 사용되는 경우, 보안 센서 및/또는 도어 록 오프너가 포함될 수 있다. 물론, 이들 예는 단지 예시의 목적으로 제공되는 것이다. 본 발명의 기본 원리는 IoT 디바이스의 임의의 특정 타입으로 제한되지 않는다. 사실, 라이브러리 코드(202)가 탑재된 저전력 마이크로제어기(200)의 고도로 프로그래밍 가능한 속성을 고려해 볼 때, 애플리케이션 개발자는 사실상 임의의 타입의 IoT 애플리케이션을 위한 저전력 마이크로제어기와 인터페이싱하도록 한 새로운 애플리케이션 코드(203) 및 새로운 I/O 디바이스(250)를 쉽게 개발할 수 있다.

[0023] 일 실시예에서, 저전력 마이크로제어기(200)는 또한 통신을 암호화하고/하거나 서명을 생성하기 위한 암호화 키를 저장하기 위한 보안 키 저장소를 포함한다. 대안적으로, 상기 키는 가입자 식별 모듈(SIM)에서 보안될 수

있다.

[0024] 일 실시예에서, IoT 디바이스를 사실상 전력을 전혀 소비하고 있지 않는 초 저전력 상태에서 깨우기 위한 웨이크업 수신기(207)가 포함된다. 일 실시예에서, 웨이크업 수신기(207)는 **도 3**에 도시된 바와 같이 IoT 허브(110)에 구성된 웨이크업 송신기(307)로부터 수신된 웨이크업 신호에 응답하여 IoT 디바이스(101)로 하여금 이러한 저전력 상태를 빠져나가게 하도록 구성된다. 특히, 일 실시예에서, 송신기(307) 및 수신기(207)는 테슬라 코일과 같은 전기 공진 변압기 회로를 함께 형성한다. 동작 시에, 허브(110)가 IoT 디바이스(101)를 매우 낮은 전력 상태에서 깨어나게 할 필요가 있을 때, 에너지가 송신기(307)로부터 라디오 주파수 신호를 통해 수신기(207)로 전송된다. 에너지 전달 때문에, IoT 디바이스(101)는 자신이 저전력 상태에 있을 때에는 전력을 사실상 전혀 소비하지 않도록 구성될 수 있는데, 왜냐하면 이 디바이스는 (디바이스들을 네트워크 신호를 통해 깨어나게 하는 네트워크 프로토콜에서와 같이) 허브로부터의 신호를 계속 "청취"할 필요가 없기 때문이다. 오히려, IoT 디바이스(101)의 마이크로제어기(200)는 송신기(307)로부터 수신기(207)로 전기적으로 전송된 에너지를 사용하여 전력이 실제로 차단된 후에 웨이크업을 하도록 구성될 수 있다.

[0025] **도 3**에 예시된 바와 같이, IoT 허브(110)는 프로그램 코드 및 데이터(305)를 저장하기 위한 메모리(317)와, 프로그램 코드를 실행하고 데이터를 처리하기 위한 마이크로제어기와 같은 하드웨어 로직(301)도 포함한다. 광역 네트워크(WAN) 인터페이스(302)와 안테나(310)가 IoT 허브(110)를 셀룰러 서비스(115)에 결합시킨다. 대안적으로, 위에서 언급된 바와 같이, IoT 허브(110)는 근거리 네트워크 통신 채널을 설정하기 위한 WiFi 인터페이스(및 WiFi 안테나) 또는 이더넷 인터페이스와 같은 로컬 네트워크 인터페이스(도시되지 않음)도 포함할 수 있다. 일 실시예에서, 하드웨어 로직(301)은 또한 통신을 암호화하고 서명을 생성/검증하기 위한 암호화 키를 저장하기 위한 보안 키 저장소를 포함한다. 대안적으로, 상기 키는 가입자 식별 모듈(SIM)에서 보안될 수 있다.

[0026] 로컬 통신 인터페이스(303)와 안테나(311)가 IoT 디바이스(101 내지 105) 각각과의 로컬 통신 채널을 확립한다. 위에서 언급된 바와 같이, 일 실시예에서, 로컬 통신 인터페이스(303)/안테나(311)는 블루투스 LE 표준을 구현한다. 그러나, 본 발명의 기본 원리는 IoT 디바이스(101 내지 105)와 로컬 통신 채널을 확립하기 위한 임의의 특정 프로토콜로 제한되지 않는다. WAN 인터페이스(302) 및/또는 로컬 통신 인터페이스(303)는 **도 3**에서는 별개의 유닛으로 예시되어 있지만 하드웨어 로직(301)과 동일한 칩 내에 임베딩될 수 있다.

[0027] 일 실시예에서, 프로그램 코드 및 데이터는 로컬 통신 인터페이스(303) 및 WAN 인터페이스(302)를 통해 통신하기 위한 별개의 스택을 포함할 수 있는 통신 프로토콜 스택(308)을 포함한다. 부가적으로, IoT 허브가 새로운 IoT 디바이스들과 페어링할 수 있도록 하기 위해 디바이스 페어링 프로그램 코드 및 데이터(306)가 메모리에 저장될 수 있다. 일 실시예에서, 각각의 새로운 IoT 디바이스(101 내지 105)는 페어링 프로세스 동안 IoT 허브(110)로 전달되는 고유 코드를 할당받는다. 예를 들어, 고유 코드는 IoT 디바이스 상의 바코드에 임베딩될 수 있으며, 바코드 판독기(106)에 의해 판독될 수 있거나 로컬 통신 채널(130)을 통해 전달될 수 있다. 대안적인 실시예에서, 고유 ID 코드는 IoT 디바이스에 자기적으로 임베딩되며, IoT 허브는 IoT 디바이스(101)가 IoT 허브(110)로부터 수 인치 내에서 이동하는 경우에 코드를 검출하기 위한 라디오 주파수 ID(RFID) 또는 근거리장 통신(NFC) 센서와 같은 자기 센서를 갖는다.

[0028] 일 실시예에서, 일단 고유 ID가 전달되면, IoT 허브(110)는 로컬 데이터베이스(도시되지 않음)에 질의하고/하거나, 코드가 수용 가능한지를 검증하기 위해 해시(hash)를 수행하고/하거나, ID 코드를 확인하기 위해 IoT 서비스(120), 사용자 디바이스(135), 및/또는 웹사이트(130)와 통신함으로써 고유 ID를 검증할 수 있다. 일단 검증되면, 일 실시예에서, IoT 허브(110)는 IoT 디바이스(101)를 페어링하고, (언급된 바와 같이, 비휘발성 메모리를 포함할 수 있는) 메모리(317)에 페어링 데이터를 저장한다. 일단 페어링이 완료되면, IoT 허브(110)는 본 명세서에 설명된 다양한 IoT 기능을 수행하기 위해 IoT 디바이스(101)와 접속할 수 있다.

[0029] 일 실시예에서, IoT 서비스(120)를 운영하는 조직은 개발자가 새로운 IoT 서비스를 용이하게 설계할 수 있도록 하기 위해 IoT 허브(110) 및 기본적인 하드웨어/소프트웨어 플랫폼을 제공할 수 있다. 특히, 개발자는 IoT 허브(110)를 제공받는 것 외에 허브(110) 내에서 실행되는 프로그램 코드 및 데이터(305)를 업데이트하기 위한 소프트웨어 개발 키트(SDK)를 제공받을 수 있다. 부가적으로, IoT 디바이스(101)에 대해, SDK는 다양한 상이한 타입의 애플리케이션(101)의 설계를 용이하게 하기 위하여 기반 IoT 하드웨어(예를 들어, 도 2에 도시된 저전력 마이크로제어기(200) 및 다른 컴포넌트)에 대해 설계된 광범위한 세트의 라이브러리 코드(202)를 포함할 수 있다. 일 실시예에서, SDK는 개발자가 IoT 디바이스에 대한 입력 및 출력을 지정하기만 하면 되는 그래픽 설계 인터페이스를 포함한다. IoT 디바이스(101)를 허브(110) 및 서비스(120)에 접속할 수 있게 하는 통신 스택(201)을 포함하여 모든 네트워킹 코드가 이미 개발자를 위해 준비되어 있다. 부가적으로, 일 실시예에서, SDK

는 모바일 디바이스(예를 들어, 아이폰 및 안드로이드 디바이스)를 위한 앱의 설계를 용이하게 하기 위한 라이브러리 코드 기반을 포함한다.

[0030] 일 실시예에서, IoT 허브(110)는 IoT 디바이스(101 내지 105)와 IoT 서비스(120) 사이의 데이터의 연속적인 양방향 스트림을 관리한다. IoT 디바이스(101 내지 105)로의/로부터의 업데이트가 실시간으로 요구되는(예를 들어, 사용자가 보안 디바이스 또는 환경 측정의 현재 상태를 볼 필요가 있는) 상황에서, IoT 허브는 정기 업데이트를 사용자 디바이스(135) 및/또는 외부 웹사이트들(130)에 제공하기 위한 개방형 TCP 소켓을 유지할 수 있다. 업데이트를 제공하는 데 사용되는 특정 네트워킹 프로토콜은 기본 애플리케이션의 필요에 기반하여 미세조정될 수 있다. 예를 들어, 연속적인 양방향 스트림을 갖는 것이 타당하지 않을 수 있는 일부 경우에서, 간단한 요청/응답 프로토콜이 정보를 필요 시 수집하는 데 사용될 수 있다.

[0031] 일 실시예에서, IoT 허브(110) 및 IoT 디바이스(101 내지 105) 둘 모두는 네트워크를 통해 자동적으로 업그레이드 가능하다. 특히, 새로운 업데이트가 IoT 허브(110)에게 이용 가능한 경우, 그 IoT 허브는 IoT 서비스(120)로부터 업데이트를 자동적으로 다운로드하여 설치할 수 있다. 그 IoT 허브는 업데이트된 코드를 먼저 로컬 메모리에 복사하고, 구동하고, 구형 프로그램 코드를 교체하기 전에 업데이트를 검증할 수 있다. 유사하게, 업데이트가 IoT 디바이스(101 내지 105) 각각에게 이용 가능한 경우, 업데이트는 초기에 IoT 허브(110)에 의해 다운로드되어서 IoT 디바이스(101 내지 105) 각각으로 내보내질 수 있다. 그 후, 각각의 IoT 디바이스(101 내지 105)는 IoT 허브에 대해 위에서 설명된 것과 유사한 방식으로 업데이트를 적용하고 업데이트의 결과를 다시 IoT 허브(110)에 보고할 수 있다. 업데이트가 성공적이면, IoT 허브(110)는 자신의 메모리로부터 업데이트를 삭제하고, (예를 들어, 그것이 각각의 IoT 디바이스에 대한 새로운 업데이트를 계속 체크할 수 있도록) 각각의 IoT 디바이스에 설치된 코드의 최신 버전을 기록할 수 있다.

[0032] 일 실시예에서, IoT 허브(110)는 A/C 전원을 통해 전력을 공급받는다. 특히, IoT 허브(110)는 A/C 전력 코드를 통해 공급된 A/C 전압을 더 낮은 DC 전압으로 변압시키기 위한 변압기를 갖는 전력 유닛(390)을 포함할 수 있다.

[0033] **도 4a**는 IoT 시스템을 사용하여 범용 원격 제어 동작을 수행하기 위한 본 발명의 일 실시예를 예시한다. 특히, 이 실시예에서, 한 세트의 IoT 디바이스들(101 내지 103)에는 (몇 개만 예로 들자면) 에어컨/히터(430), 조명 시스템(431), 및 시청각 장비(432)를 포함한 다양한 상이한 타입의 전자 장비를 제어하기 위해 원격 제어 코드를 전송하기 위한 적외선(IR) 및/또는 라디오 주파수(RF) 블라스터들(401 내지 403)이 각각 탑재된다. **도 4a**에 도시된 실시예에서, IoT 디바이스들(101 내지 103)에는 후술되는 바와 같이 그들이 제어하는 디바이스들의 동작을 검출하기 위한 센서들(404 내지 406)이 각각 탑재된다.

[0034] 예를 들어, IoT 디바이스(101) 내의 센서(404)는 현재의 온도/습도를 감지하고, 그에 응답하여 현재의 원하는 온도에 기반하여 에어컨/히터(430)를 제어하기 위한 온도 및/또는 습도 센서일 수 있다. 이 실시예에서, 에어컨/히터(430)는 원격 제어 디바이스(전형적으로 자체가 임베딩된 온도 센서를 갖는 리모트 컨트롤)를 통해 제어되도록 설계된 것이다. 일 실시예에서, 사용자는 사용자 디바이스(135) 상에 설치된 앱 또는 브라우저를 통해 IoT 허브(110)에 원하는 온도를 제공한다. IoT 허브(110) 상에서 실행되는 제어 로직(412)은 센서(404)로부터 현재 온도/습도 데이터를 수신하고, 그에 응답하여 원하는 온도/습도에 따라 IR/RF 블라스터(401)를 제어하기 위해 IoT 디바이스(101)에 커맨드를 송신한다. 예를 들어, 온도가 원하는 온도보다 낮으면, 제어 로직(412)은 커맨드를 IR/RF 블라스터(401)를 통해 에어컨/히터로 송신하여 (예를 들어, 에어컨을 끄거나 히터를 켬으로써) 온도를 높일 수 있다. 커맨드는 IoT 허브(110)의 데이터베이스(413) 내에 저장된 필요한 원격 제어 코드를 포함할 수 있다. 대안적으로 또는 추가적으로, IoT 서비스(421)는 지정된 사용자 기본 기반설정 및 저장된 제어 코드(422)에 하여 전자 장비(430 내지 432)를 제어하기 위한 제어 로직(421)을 구현할 수 있다.

[0035] 예시된 예의 IoT 디바이스(102)는 조명(431)을 제어하는 데 사용된다. 특히, IoT 디바이스(102) 내의 센서(405)는 조명 설비(431)(또는 다른 조명 장치)에 의해 생성되는 광의 현재 밝기를 검출하도록 구성된 광센서 또는 광검출기일 수 있다. 사용자는 사용자 디바이스(135)를 통해 IoT 허브(110)에 원하는 조명 레벨(온 또는 오프의 지시를 포함함)을 지정할 수 있다. 이에 응답하여, 제어 로직(412)은 IR/RF 블라스터(402)에 커맨드를 송신하여, (431)의 현재 밝기 레벨을 제어할 것이다(예를 들어, 현재 밝기가 너무 낮으면 조명을 높이거나 현재 밝기가 너무 높으면 조명을 낮추거나; 전등을 단순히 켜거나 또는 끔).

[0036] 예시된 예의 IoT 디바이스(103)는 시청각 장비(432)(예를 들어, 텔레비전, A/V 수신기, 케이블/위성 수신기, 애플(Apple)TV™ 등)를 제어하도록 구성된다. IoT 디바이스(103) 내의 센서(406)는 현재의 주위 볼륨 레벨을 검출하기 위한 오디오 센서(예를 들어, 마이크로폰 및 관련 로직)이고/이거나, 텔레비전에 의해 생성된 광에 기반

하여 (예를 들어, 지정된 스펙트럼 내의 광을 측정함으로써) 텔레비전이 켜짐 또는 꺼짐 상태인지를 검출하기 위한 광센서일 수 있다. 대안적으로, 센서(406)는 검출된 온도에 기반하여 오디오 장비가 켜짐 또는 꺼짐 상태인지를 검출하기 위해 시청각 장비에 접속된 온도 센서를 포함할 수 있다. 다시 한번, 제어 로직(412)은 사용자 디바이스(135)를 통한 사용자 입력에 응답하여 커맨드를 IoT 디바이스(103)의 IR 블라스터(403)를 통해 시청각 장비로 전송할 수 있다.

[0037] 전술한 내용은 단지 본 발명의 일 실시예의 예시적인 예에 불과하다는 점에 유의해야 한다. 본 발명의 기본 원리는 IoT 디바이스에 의해 제어될 임의의 특정 타입의 센서 또는 장비로 제한되지 않는다.

[0038] IoT 디바이스들(101 내지 103)이 블루투스 LE 접속을 통해 IoT 허브(110)에 연결되는 실시예에서, 센서 데이터 및 커맨드는 블루투스 LE 채널을 통해 전송된다. 그러나, 본 발명의 기본 원리는 블루투스 LE 또는 임의의 다른 통신 표준으로 제한되지 않는다.

[0039] 일 실시예에서, 각각의 전자 장비를 제어하는 데 필요한 제어 코드는 IoT 허브(110)의 데이터베이스(413) 및/또는 IoT 서비스(120)의 데이터베이스(422)에 저장된다. **도 4b**에 예시된 바와 같이, 제어 코드는 IoT 서비스(120)에 유지되는 상이한 장비들에 대한 제어 코드들(422)의 마스터 데이터베이스로부터 IoT 허브(110)로 제공될 수 있다. 최종 사용자는 사용자 디바이스(135)에서 실행되는 앱 또는 브라우저를 통해 제어될 전자(또는 다른) 장비의 타입을 지정할 수 있고, IoT 허브 상의 원격 제어 코드 학습 모듈(491)은 그에 응답하여 IoT 서비스(120)의 원격 제어 코드 데이터베이스(492)에서 (예를 들어, 고유 ID를 갖는 각각의 전자 장비를 식별하는) 필요한 IR/RF 코드를 검색할 수 있다.

[0040] 또한, 일 실시예에서, IoT 허브(110)에는 원격 제어 코드 학습 모듈(491)이 전자 장비와 함께 제공된 원래의 리모트 컨트롤(495)로부터 직접 새로운 원격 제어 코드를 "학습"하는 것을 가능하게 하는 IR/RF 인터페이스(490)가 탑재된다. 예를 들어, 에어컨(430)과 함께 제공된 원래의 리모트 컨트롤에 대한 제어 코드가 원격 제어 데이터베이스에 포함되어 있지 않으면, 사용자는 사용자 디바이스(135)의 앱/브라우저를 통해 IoT 허브(110)와 상호 작용하여, 원래의 리모트 컨트롤에 의해 생성된 다양한 제어 코드(예를 들어, 온도 증가, 온도 감소 등)를 IoT 허브(110)에게 교시할 수 있다. 원격 제어 코드는 학습되면 IoT 허브(110) 상의 제어 코드 데이터베이스(413)에 저장되고/되거나, IoT 서비스(120)로 역전송되어 중앙 원격 제어 코드 데이터베이스(492)에 포함될 수 있다(그리고 후속하여 동일한 에어컨 유닛(430)을 갖는 다른 사용자에 의해 사용됨).

[0041] 일 실시예에서, IoT 디바이스들(101 내지 103) 각각은 극도로 작은 폼 팩터를 가지며, 양면 테이프, 작은 네일, 자기 부착물 등을 사용하여 그들 각각의 전자 장비(430 내지 432) 상에 또는 그 근처에 부착될 수 있다. 공기 조화기(430)와 같은 장비의 제어를 위해, 센서(404)가 가정 내의 주변 온도를 정확하게 측정할 수 있도록 IoT 디바이스(101)를 충분히 멀리 배치하는 것이 바람직할 것이다(예를 들어, 공기 조화기에 IoT 디바이스를 직접 배치하는 것은, 공기 조화기가 작동할 때 너무 낮거나 히터가 작동할 때 너무 높은 온도 측정을 초래할 것이다). 대조적으로, 조명을 제어하는 데 사용되는 IoT 디바이스(102)는 센서(405)가 현재 조명 레벨을 검출하기 위해 조명 설비(431) 상에 또는 그 부근에 배치될 수 있다.

[0042] 설명된 바와 같은 일반적인 제어 기능을 제공하는 것 외에도, IoT 허브(110) 및/또는 IoT 서비스(120)의 일 실시예는 각각의 전자 장비의 현재 상태와 관련된 통지를 최종 사용자에게 송신한다. 문자 메시지 및/또는 앱 전용 통지일 수 있는 통지는 이어서 사용자의 모바일 디바이스(135)의 디스플레이에 표시될 수 있다. 예를 들어, 사용자의 에어컨이 장기간 동안 켜져 있었지만 온도가 변하지 않은 경우, IoT 허브(110) 및/또는 IoT 서비스(120)는 에어컨이 적절히 기능하고 있지 않다는 통지를 사용자에게 전송할 수 있다. 사용자가 집에 있지 않고(이는 모션 센서를 통해 또는 사용자의 현재 검출된 위치에 기반하여 검출될 수 있음), 센서(406)가 시청각 장비(430)가 켜져 있다는 것을 나타내거나, 센서(405)가 전등이 켜져 있다는 것을 나타내는 경우, 사용자가 시청각 장비(432) 및/또는 전등(431)을 끄기를 원하는지를 묻는 통지가 사용자에게 전송될 수 있다. 임의의 장비 타입에 대해 동일한 타입의 통지가 전송될 수 있다.

[0043] 사용자가 통지를 수신하면, 그 사용자는 사용자 디바이스(135)의 앱 또는 브라우저를 통해 전자 장비(430 내지 432)를 원격 제어할 수 있다. 일 실시예에서, 사용자 디바이스(135)는 터치스크린 디바이스이고, 앱 또는 브라우저는 장비(430 내지 432)를 제어하기 위해 사용자가 선택할 수 있는 버튼을 갖는 리모트 컨트롤의 이미지를 표시한다. 사용자는 통지를 수신하면 그래픽 리모트 컨트롤을 열어서 다양한 상이한 장비들을 끄거나 조정할 수 있다. IoT 서비스(120)를 통해 접속되는 경우, 사용자의 선택은 IoT 서비스(120)로부터 IoT 허브(110)로 전송될 수 있고, 이어서 IoT 허브(110)는 제어 로직(412)을 통해 장비를 제어할 것이다. 대안적으로, 사용자 입력은 사용자 디바이스(135)로부터 IoT 허브(110)로 직접 전송될 수 있다.

[0044] 일 실시예에서, 사용자는 전자 장비(430 내지 432)에 대한 다양한 자동 제어 기능을 수행하도록 IoT 허브(110) 상의 제어 로직(412)을 프로그래밍할 수 있다. 제어 로직(412)은 전술한 바와 같이 원하는 온도, 밝기 레벨, 및 볼륨 레벨을 유지하는 것 이외에도 소정 조건이 검출되면 전자 장비를 자동으로 끌 수 있다. 예를 들어, 제어 로직(412)은 사용자가 집에 없고 에어컨이 기능하고 있지 않다는 것을 검출하게 되면 자동으로 에어컨을 끌 수 있다. 유사하게, 사용자가 집에 없고, 센서(406)가 시청각 장비(430)가 켜져 있음을 나타내거나 센서(405)가 전등이 켜져 있음을 나타내면, 제어 로직(412)은 시청각 장비 및 전등을 각각 끄도록 하는 커맨드를 IR/RF 블라스터(403, 402)를 통해 자동 송신할 수 있다.

[0045] **도 5**는 전자 장비(530, 531)를 모니터링하기 위한 센서(503, 504)가 탑재된 IoT 디바이스(104, 105)의 추가 실시예를 예시한다. 특히, 이 실시예의 IoT 디바이스(104)는 스토브가 켜진 채로 있을 때를 검출하기 위해 스토브(530) 상에 또는 그 부근에 배치될 수 있는 온도 센서(503)를 포함한다. 일 실시예에서, IoT 디바이스(104)는 온도 센서(503)에 의해 측정된 현재 온도를 IoT 허브(110) 및/또는 IoT 서비스(120)로 전송한다. 스토브가 (예를 들어, 측정된 온도에 기반하여) 임계 기간을 초과하여 켜져 있는 것으로 검출되면, 제어 로직(512)은 사용자에게 스토브(530)가 켜져 있음을 알리는 통지를 최종 사용자의 디바이스(135)로 전송할 수 있다. 또한, 일 실시예에서, IoT 디바이스(104)는, 사용자로부터 명령을 수신하는 것에 응답하여 또는 (제어 로직(512)이 사용자에 의해 그렇게 하도록 프로그래밍된 경우) 자동으로 스토브를 끄기 위한 제어 모듈(501)을 포함할 수 있다. 일 실시예에서, 제어 로직(501)은 스토브(530)로의 전기 또는 가스를 차단하는 스위치를 포함한다. 그러나, 다른 실시예에서, 제어 로직(501)은 스토브 자체 내에 통합될 수 있다.

[0046] **도 5**는 또한 세탁기 및/또는 건조기와 같은 소정 타입의 전자 장비의 모션을 검출하기 위한 모션 센서(504)를 갖는 IoT 디바이스(105)를 예시한다. 사용될 수 있는 다른 센서는 주위 볼륨 레벨을 검출하기 위한 오디오 센서(예를 들어, 마이크로폰 및 로직)이다. 전술한 다른 실시예에서와 같이, 이 실시예는 소정의 지정된 조건이 충족되면(예를 들어, 모션이 장기간 동안 검출되어 세탁기/건조기가 꺼지지 않았음을 나타내는 경우) 최종 사용자에게 통지를 송신할 수 있다. **도 5**에 도시되지 않지만, IoT 디바이스(105)에는 자동으로 그리고/또는 사용자 입력에 응답하여 (예를 들어, 전기/가스를 절환시킴으로써) 세탁기/건조기(531)를 끄는 제어 모듈도 탑재될 수 있다.

[0047] 일 실시예에서, 제어 로직 및 스위치를 갖는 제1 IoT 디바이스는 사용자의 집 안의 모든 전원을 끄도록 구성될 수 있고, 제어 로직 및 스위치를 갖는 제2 IoT 디바이스는 사용자의 집 안의 모든 가스를 끄도록 구성될 수 있다. 이어서, 센서를 갖는 IoT 디바이스가 사용자의 집에 있는 전자 또는 가스 구동 장비 상에 또는 그 부근에 배치될 수 있다. 사용자가 특정 장비(예를 들어, 스토브(530))가 켜진 채로 있다는 통지를 받으면, 사용자는 손상을 방지하기 위해 집 안의 모든 전기 또는 가스를 끄도록 하는 명령을 전송할 수 있다. 대안적으로, IoT 허브(110) 및/또는 IoT 서비스(120) 내의 제어 로직(512)은 그러한 상황에서 전기 또는 가스를 자동으로 끄도록 구성될 수 있다.

[0048] 일 실시예에서, IoT 허브(110)와 IoT 서비스(120)는 주기적인 간격으로 통신한다. IoT 서비스(120)가 (예를 들어, 지정된 지속시간 동안 IoT 허브로부터 요청 또는 응답을 수신하지 못함으로써) IoT 허브(110)에 대한 접속이 끊긴 것을 검출하면, (예를 들어, 문자 메시지 또는 앱 전용 통지를 전송함으로써) 이러한 정보를 최종 사용자의 디바이스(135)로 전달할 것이다.

[0049] **중개 디바이스를 통해 데이터를 통신하기 위한 장치 및 방법**

[0050] 상기에 언급된 바와 같이, 블루투스 LE와 같은 IoT 디바이스들을 상호접속하는 데 사용되는 무선 기술들은 일반적으로 단거리 기술들이기 때문에, IoT 구현을 위한 허브가 IoT 디바이스의 범위 밖에 있는 경우, IoT 디바이스는 데이터를 IoT 허브로(그리고 이와 반대로도) 전송할 수 없을 것이다.

[0051] 이러한 결함을 해결하기 위해, 본 발명의 일 실시예는 IoT 허브의 무선 범위 밖에 있는 IoT 디바이스가 모바일 디바이스들이 범위 내에 있을 때 하나 이상의 모바일 디바이스와 주기적으로 접속하기 위한 메커니즘을 제공한다. IoT 디바이스는 일단 접속되면, IoT 허브에 제공될 필요가 있는 임의의 데이터를 모바일 디바이스로 송신할 수 있으며, 이어서 모바일 디바이스는 데이터를 IoT 허브로 전송한다.

[0052] **도 6**에 예시된 바와 같이, 일 실시예는 IoT 허브(110), IoT 허브(110)의 범위 밖에 있는 IoT 디바이스(601), 및 모바일 디바이스(611)를 포함한다. 범위 밖에 있는 IoT 디바이스(601)는 데이터를 수집 및 전달할 수 있는 임의의 형태의 IoT 디바이스를 포함할 수 있다. 예를 들어, IoT 디바이스(601)는 냉장고에서 이용 가능한 식품들, 그 식품들을 소비하는 사용자들, 및 현재 온도를 모니터링하기 위해 냉장고 내에 구성된 데이터 수집

디바이스를 포함할 수 있다. 물론, 본 발명의 기본 원리는 IoT 디바이스의 임의의 특정 타입으로 제한되지 않는다. 본 명세서에서 설명되는 기술들은, 단지 몇 가지 예를 들자면, 스마트 미터, 스토브, 세탁기, 건조기, 조명 시스템, HVAC 시스템, 및 시청각 장비에 대한 데이터를 수집 및 송신하는 데 사용되는 것들을 포함한 임의의 타입의 IoT 디바이스를 사용하여 구현될 수 있다.

[0053] 더욱이, **도 6**에 예시된 모바일 디바이스(611)는 데이터를 통신 및 저장할 수 있는 임의의 형태의 모바일 디바이스일 수 있다. 예를 들어, 일 실시예에서, 모바일 디바이스(611)는 본 명세서에서 설명되는 기술들을 촉진하기 위한 앱이 설치된 스마트폰이다. 다른 실시예에서, 모바일 디바이스(611)는 목걸이 또는 팔찌에 부착된 통신 토큰, 스마트워치, 또는 피트니스 디바이스와 같은 웨어러블 디바이스를 포함한다. 스마트폰 디바이스를 소유하지 않은 노년 사용자들 또는 다른 사용자들에게 웨어러블 토큰이 특히 유용할 수 있다.

[0054] 동작 시, 범위 밖에 있는 IoT 디바이스(601)는 모바일 디바이스(611)와의 접속을 주기적으로 또는 계속적으로 체크할 수 있다. (예를 들어, 사용자가 냉장고의 부근 내에서 움직인 결과로) 접속이 확립되면, IoT 디바이스(601) 상의 임의의 수집된 데이터(605)가 모바일 디바이스(611)의 임시 데이터 저장소(615)로 자동 전송된다. 일 실시예에서, IoT 디바이스(601) 및 모바일 디바이스(611)는 BTLE와 같은 저전력 무선 표준을 사용하여 로컬 무선 통신 채널을 확립한다. 그러한 경우, 모바일 디바이스(611)는 공지된 페어링 기술들을 이용하여 초기에 IoT 디바이스(601)와 페어링될 수 있다.

[0055] 일단 데이터가 임시 데이터 저장소로 전송된 상태에서, (예를 들어, 사용자가 IoT 허브(110)의 범위 내에서 걸을 때) 일단 IoT 허브(110)와의 통신이 확립되면, 모바일 디바이스(611)는 데이터를 전송할 것이다. 이어서, IoT 허브는 데이터를 중앙 데이터 저장소(413)에 저장하고/하거나, 데이터를 인터넷을 통해 하나 이상의 서비스 및/또는 다른 사용자 디바이스로 전송할 수 있다. 일 실시예에서, 모바일 디바이스(611)는 상이한 타입의 통신 채널(잠재적으로는 WiFi와 같은 보다 높은 출력의 통신 채널)을 사용하여 데이터를 IoT 허브(110)에 제공할 수 있다.

[0056] 범위 밖의 IoT 디바이스(601), 모바일 디바이스(611), 및 IoT 허브는 모두가 본 명세서에서 설명되는 기술들을 구현하기 위한 프로그램 코드 및/또는 로직으로 구성될 수 있다. **도 7**에 예시된 바와 같이, 예를 들어, 본 명세서에서 설명되는 동작들을 수행하기 위해, IoT 디바이스(601)는 중개 접속 로직 및/또는 애플리케이션으로 구성될 수 있고, 모바일 디바이스(611)는 중개 접속 로직/애플리케이션으로 구성될 수 있고, IoT 허브(110)는 중개 접속 로직/애플리케이션(721)으로 구성될 수 있다. 각각의 디바이스의 중개 접속 로직/애플리케이션은 하드웨어, 소프트웨어, 또는 이들의 임의의 조합으로 구현될 수 있다. 일 실시예에서, IoT 디바이스(601)의 중개 접속 로직/애플리케이션(701)은 (디바이스 앱으로서 구현될 수 있는) 모바일 디바이스 상의 중개 접속 로직/애플리케이션(711)과의 접속을 검색 및 설정하여 데이터를 임시 데이터 저장소(615)로 전송한다. 이어서, 모바일 디바이스(611) 상의 중개 접속 로직/애플리케이션(701)은 데이터를 IoT 허브 상의 중개 접속 로직/애플리케이션으로 전송하고, IoT 허브는 데이터를 중앙 데이터 저장소(413)에 저장한다.

[0057] **도 7**에 예시된 바와 같이, 각각의 디바이스 상의 중개 접속 로직/애플리케이션(701, 711, 721)은 가까운 곳에 있는 응용에 기반하여 구성될 수 있다. 예를 들어, 냉장고의 경우, 접속 로직/애플리케이션(701)은 소수의 패킷을 주기적으로 전송하기만 하면 될 수 있다. 다른 응용들(예를 들어, 온도 센서들)의 경우, 접속 로직/애플리케이션(701)은 더 빈번한 업데이트들을 전송하는 것이 필요할 수 있다.

[0058] 일 실시예에서, 모바일 디바이스(611)보다는 IoT 디바이스(601)가, IoT 허브(110)의 범위 내에 위치되는 하나 이상의 중개 IoT 디바이스와의 무선 접속을 확립하도록 구성될 수 있다. 이 실시예에서, IoT 허브의 범위 밖에 있는 임의의 IoT 디바이스들(601)은 다른 IoT 디바이스들을 사용하여 "체인"을 형성함으로써 허브에 링크될 수 있다.

[0059] 또한, **도 6 및 도 7**에는 간략화를 위해 단일 모바일 디바이스(611)만이 예시되지만, 일 실시예에서, 상이한 사용자들의 그러한 모바일 디바이스들 다수가 IoT 디바이스(601)와 통신하도록 구성될 수 있다. 더욱이, 다수의 다른 IoT 디바이스들에 대해 동일한 기술이 구현될 수 있고, 이에 의해 집 전체에 걸쳐 하나의 중개 디바이스 데이터 수집 시스템이 형성될 수 있다.

[0060] 더욱이, 일 실시예에서, 본 명세서에서 설명되는 기술들은 다양한 상이한 타입의 적절한 데이터를 수집하는 데 사용될 수 있다. 예를 들어, 일 실시예에서, 모바일 디바이스(611)가 IoT 디바이스(601)와 접속할 때마다, 사용자의 아이덴티티가, 수집된 데이터(605)에 포함될 수 있다. 이러한 방식으로, IoT 시스템은 집 안의 상이한 사용자들의 거동을 추적하는 데 사용될 수 있다. 예를 들어, 수집된 데이터(605)는 냉장고에서 사용되는 경우

에는 냉장고 옆을 지나가는 각각의 사용자, 냉장고를 여는 각각의 사용자, 및 각각의 사용자에 의해 소비되는 특정 식품들의 아이덴티티를 포함할 수 있다. 상이한 타입의 데이터가 다른 타입의 IoT 디바이스들로부터 수집될 수 있다. 시스템은 이 데이터를 사용하여, 예를 들어, 어떤 사용자가 옷을 세탁하고, 어떤 사용자가 주어진 날에 TV를 시청하고, 각 사용자가 잠에 들고 일어나는 시간 등을 결정할 수 있다. 이 크라우드-소싱된 데이터 전부는 그 후 IoT 허브의 데이터 저장소(413) 내에서 컴파일되고/되거나 외부 서비스 또는 사용자에게 포워딩될 수 있다.

[0061] 본 명세서에서 설명되는 기술들의 또 다른 유익한 응용은 도움을 필요로 할 수 있는 노년 사용자들을 모니터링하는 것이다. 이러한 응용을 위해, 모바일 디바이스(611)는 사용자의 집의 상이한 방들에서 정보를 수집하기 위한, 노년 사용자에 의해 착용되는 매우 작은 토큰일 수 있다. 예를 들어, 사용자가 냉장고를 열 때마다, 이러한 데이터가 수집된 데이터(605)에 포함되어 토큰을 통해 IoT 허브(110)로 전송될 것이다. 이어서, IoT 허브는 그 데이터를 하나 이상의 외부 사용자(예를 들어, 노년 사용자를 돌보는 자식들 또는 다른 사람들)에게 제공할 수 있다. 데이터가 지정된 기간(예를 들어, 12시간) 동안 수집되지 않은 경우, 이것은 노년 사용자가 집 주위에서 움직이지 않았고/않았거나 냉장고를 열지 않았음을 의미한다. 이어서, IoT 허브(110)나 혹은 이 IoT 허브에 접속된 외부 서비스가 경고 통지를 이러한 다른 사람들에게 전송하여, 그들에게 노년 사용자를 확인해 봐야 한다는 것을 알릴 수 있다. 또한, 수집된 데이터(605)는 사용자가 음식을 소비하고 있는지 그리고 식료품 가게에 가는 것이 필요한지, 노년 사용자가 TV를 시청하고 있는지 그리고 얼마나 자주 시청하는지, 노년 사용자가 옷을 세탁하는 빈도 등과 같은 다른 적절한 정보를 포함할 수 있다.

[0062] 다른 구현에서, 세탁기, 냉장고, HVAC 시스템 등과 같은 전자 디바이스에 문제가 있는 경우, 수집된 데이터는 교체가 필요한 부품에 대한 표시를 포함할 수 있다. 그러한 경우, 문제 해결을 요청이 담긴 통지가 기술자에게 전송될 수 있다. 그러면 기술자는 필요한 교체 부품을 갖고서 집에 도착할 수 있다.

[0063] 본 발명의 일 실시예에 따른 방법이 **도 8**에 도시된다. 본 방법은 위에 기술된 아키텍처들의 맥락 내에서 구현될 수 있지만, 임의의 특정 아키텍처로 제한되지 않는다.

[0064] 801에서, IoT 허브의 범위 밖에 있는 IoT 디바이스가 데이터(예를 들어, 냉장고 문의 개방, 사용된 식품 등)를 주기적으로 수집한다. 802에서, IoT 디바이스는 (예를 들어, BTLE 표준에 의해 지정된 것들과 같은, 접속을 설정하기 위한 표준 로컬 무선 기술들을 사용하여) 모바일 디바이스와의 접속을 주기적으로 또는 계속적으로 점검한다. 모바일 디바이스에 대한 접속이 확립된 것으로 802에서 결정되면, 803에서, 수집된 데이터가 모바일 디바이스로 전송된다. 804에서, 모바일 디바이스는 데이터를 IoT 허브, 외부 서비스, 및/또는 사용자에게 전송한다. 언급된 바와 같이, 모바일 디바이스는 (예를 들어, WiFi 링크를 통해) 이미 접속된 경우에는 즉시 데이터를 송신할 수 있다.

[0065] 일 실시예에서, 본 명세서에서 설명되는 기술들은 IoT 디바이스들로부터 데이터를 수집하는 것 외에도, 데이터를 IoT 디바이스들에 업데이트하거나 달리 제공하는 데 사용될 수 있다. 일례가 **도 9a**에 도시되어 있는데, 이 도면은 IoT 디바이스(601)(또는 그러한 IoT 디바이스들의 그룹)에 설치될 필요가 있는 프로그램 코드 업데이트들(901)을 갖는 IoT 허브(110)를 도시한다. 프로그램 코드 업데이트들은 IoT 디바이스가 사용자에 의해 요구되는 대로 동작하는 데 필요한 시스템 업데이트들, 패치들, 구성 데이터, 및 임의의 다른 데이터를 포함할 수 있다. 일 실시예에서, 사용자는 IoT 디바이스(601)에 대한 구성 옵션들을 모바일 디바이스 또는 컴퓨터를 통해 지정할 수 있으며, 이어서 그 옵션들은 IoT 허브(110) 상에 저장되고, 본 명세서에서 설명되는 기술들을 사용하여 IoT 디바이스에 제공된다. 구체적으로, 일 실시예에서, IoT 허브(110) 상의 중개 접속 로직/애플리케이션(721)은 모바일 디바이스(611) 상의 중개 접속 로직/애플리케이션(711)과 통신하여, 프로그램 코드 업데이트들을 임시 저장소(615) 내에 저장한다. 모바일 디바이스(611)가 IoT 디바이스(601)의 범위에 들어간 때, 모바일 디바이스(611) 상의 중개 접속 로직/애플리케이션(711)은 IoT 디바이스(601) 상의 중개 접속 로직/애플리케이션(701)과 접속하여 프로그램 코드 업데이트들을 디바이스에 제공한다. 일 실시예에서, 그러면 IoT 디바이스(601)는 자동화 업데이트 프로세스로 들어가서 새로운 프로그램 코드 업데이트들 및/또는 데이터를 설치할 수 있다.

[0066] IoT 디바이스를 업데이트하기 위한 방법이 **도 9b**에 도시된다. 본 방법은 위에 기술된 시스템 아키텍처의 맥락 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0067] 900에서, 새로운 프로그램 코드 또는 데이터 업데이트들이 (예를 들어, 인터넷을 통해 모바일 디바이스에 결합된) IoT 허브 및/또는 외부 서비스 상에서 이용 가능해진다. 901에서, 모바일 디바이스가 IoT 디바이스를 대신하여 프로그램 코드 또는 데이터 업데이트들을 받고 저장한다. 902에서, IoT 디바이스 및/또는 모바일 디바이

스는 접속이 확립되었는지를 결정하기 위해 주기적으로 점검한다. 접속이 확립된 것으로 903에서 결정되면, 904에서, 업데이트들이 IoT 디바이스로 전송되어 설치된다.

[0068] **개선된 보안을 위한 실시예들**

[0069] 일 실시예에서, 각각의 IoT 디바이스(101)의 저전력 마이크로제어기(200) 및 IoT 허브(110)의 저전력 로직/마이크로제어기(301)는 후술되는 실시예에 의해 사용되는 암호화 키를 저장하기 위한 보안 키 저장소를 포함한다(예를 들어, **도 10 내지 도 15** 및 관련 텍스트 참조). 대안적으로, 상기 키는 아래에서 논의되는 바와 같이 가입자 식별 모듈(SIM)에서 보안될 수 있다.

[0070] **도 10**은 IoT 서비스(120)와 IoT 허브(110)와 IoT 디바이스(101, 102) 간의 통신을 암호화하기 위한 공개 키 기반구조(PKI) 기술 및/또는 대칭 키 교환/암호화 기술을 사용하는 고수준 아키텍처를 예시한다.

[0071] 공개/비공개 키 쌍을 사용하는 실시예가 먼저 설명될 것이고, 이어서 대칭 키 교환/암호화 기술을 사용하는 실시예가 설명될 것이다. 특히, PKI를 사용하는 실시예에서는, 고유 공개/비공개 키 쌍이 각각의 IoT 디바이스(101, 102), 각각의 IoT 허브(110), 및 IoT 서비스(120)와 관련된다. 일 실시예에서, 새로운 IoT 허브(110)가 설정된 때, 그 허브의 공개 키가 IoT 서비스(120)에 제공되고, 새로운 IoT 디바이스(101)가 설정될 때, 그 디바이스의 공개 키가 IoT 허브(110)와 IoT 서비스(120) 모두에 제공된다. 디바이스들 사이에서 공개 키를 안전하게 교환하기 위한 다양한 기술이 아래에서 설명된다. 일 실시예에서, 임의의 수신 디바이스가 서명을 확인함으로써 공개 키의 유효성을 검증할 수 있도록, 모든 수신 디바이스에 알려진(즉, 인증서 형태의) 마스터 키에 의해 모든 공개 키가 서명된다. 따라서, 단지 원시 공개 키만을 교환하기보다는 이러한 인증서가 교환될 것이다.

[0072] 예시된 바와 같이, 일 실시예에서, 각각의 IoT 디바이스(101, 102)는 각각의 디바이스의 비공개 키를 보안 저장하기 위한 보안 키 저장소(1001, 1003)를 각각 포함한다. 이어서, 보안 로직(1002, 1304)은 안전하게 저장된 비공개 키를 사용하여 본 명세서에 설명된 암호화/해독 동작을 수행한다. 유사하게, IoT 허브(110)는 IoT 허브 비공개 키 및 IoT 디바이스(101, 102) 및 IoT 서비스(120)의 공개 키를 저장하기 위한 보안 저장소(1011)를 포함할 뿐만 아니라, 키를 사용하여 암호화/해독 동작을 수행하기 위한 보안 로직(1012)도 포함한다. 마지막으로, IoT 서비스(120)는 그 자신의 비공개 키, 다양한 IoT 디바이스 및 IoT 허브의 공개 키를 보안 저장하기 위한 보안 저장소(1021)와, 이 키들을 사용하여 IoT 허브 및 디바이스와의 통신을 암호화/해독하기 위한 보안 로직(1013)을 포함할 수 있다. 일 실시예에서, IoT 허브(110)가 IoT 디바이스로부터 공개 키 인증서를 수신한 때, IoT 허브는 (예를 들어, 전술한 바와 같은 마스터 키를 사용하여 서명을 확인함으로써) 그 인증서를 검증할 수 있고, 이어서 그 인증서 내에서 공개 키를 추출하여 그 공개 키를 그것의 보안 키 저장소(1011)에 저장할 수 있다.

[0073] 예로서, 일 실시예에서, IoT 서비스(120)가 커맨드 또는 데이터(예를 들어, 도어를 열기 위한 커맨드, 센서를 판독하기 위한 요청, IoT 디바이스에 의해 처리/표시될 데이터 등)를 IoT 디바이스(101)로 송신할 필요가 있을 때, 보안 로직(1013)은 IoT 디바이스(101)의 공개 키를 사용하여 데이터/커맨드를 암호화하여 암호화된 IoT 디바이스 패킷을 생성한다. 일 실시예에서, 보안 로직은 이어서 IoT 허브(110)의 공개 키를 사용하여 IoT 디바이스 패킷을 암호화하여 IoT 허브 패킷을 생성하고 IoT 허브 패킷을 IoT 허브(110)로 송신한다. 일 실시예에서, 서비스(120)는 암호화된 메시지를 그것의 비공개 키 또는 상기에 언급된 마스터 키로 서명하여, 디바이스(101)가 신뢰 소스로부터 변경되지 않은 메시지를 수신하고 있는지를 검증할 수 있도록 한다. 이어서, 디바이스(101)는 비공개 키 및/또는 마스터 키에 대응하는 공개 키를 사용하여 서명을 확인할 수 있다. 전술한 바와 같이, 공개/비공개 키 암호화 대신에 대칭 키 교환/암호화 기술이 사용될 수 있다. 이들 실시예에서, 하나의 키를 비공개적으로 저장하고 대응하는 공개 키를 다른 디바이스에 제공하기보다는, 디바이스들 각각에, 암호화에 사용되고 서명을 확인하는 데 사용되는 동일한 대칭 키의 사본을 제공할 수 있다. 대칭 키 알고리즘의 일례는 진보된 암호화 표준(AES)이지만, 본 발명의 기본 원리는 임의의 타입의 특정 대칭 키로 제한되지 않는다.

[0074] 대칭 키 구현을 사용하여, 각각의 디바이스(101)는 IoT 허브(110)와 대칭 키를 교환하기 위해 보안 키 교환 프로토콜에 들어간다. 동적 대칭 키 프로비저닝 프로토콜(DSKPP)과 같은 보안 키 프로비저닝 프로토콜이 보안 통신 채널을 통해 키를 교환하는 데 사용될 수 있다(예를 들어, RFC(Request for Comments) 6063 참조). 그러나, 본 발명의 기본 원리들은 임의의 특정 키 프로비저닝 프로토콜로 한정되지 않는다.

[0075] 일단 대칭 키가 교환되면, 대칭 키는 통신을 암호화하기 위해 각각의 디바이스(101) 및 IoT 허브(110)에 의해 사용될 수 있다. 유사하게, IoT 허브(110) 및 IoT 서비스(120)는 보안 대칭 키 교환을 수행한 다음, 교환된 대칭 키를 사용하여 통신을 암호화할 수 있다. 일 실시예에서, 새로운 대칭 키가 디바이스(101)와 허브(110) 사

이에서 그리고 허브(110)와 IoT 서비스(120) 사이에서 주기적으로 교환된다. 일 실시예에서, 새로운 대칭 키가 디바이스(101)와 허브(110)와 서비스(120) 사이에서 각각의 새로운 통신 세션을 이용하여 교환된다(예를 들어, 새로운 키가 생성되고 각각의 통신 세션 동안 안전하게 교환된다). 일 실시예에서, IoT 허브 내의 보안 모듈(1012)이 신뢰되는 경우, 서비스(120)는 허브 보안 모듈(1312)과 세션 키를 협상할 수 있고, 이어서 보안 모듈(1012)은 각각의 디바이스(120)와 세션 키를 협상할 것이다. 이어서, 서비스(120)로부터의 메시지가 디바이스(101)로의 전송을 위해 재암호화되기 전에 허브 보안 모듈(1012)에서 해독 및 검증될 것이다.

[0076] 일 실시예에서, 허브 보안 모듈(1012)에 대한 손상(compromise)을 방지하기 위해, 설치 시에 디바이스(101)와 서비스(120) 사이에서 1회(영구) 설치 키가 협상될 수 있다. 메시지를 디바이스(101)로 전송할 때, 서비스(120)는 먼저 이 디바이스 설치 키로 암호화/MAC하고, 이어서 허브의 세션 키로 암호화/MAC할 수 있다. 이어서, 허브(110)는 암호화된 디바이스 블롭(device blob)을 검증 및 추출하여, 이를 디바이스로 전송할 것이다.

[0077] 본 발명의 일 실시예에서, 재생 공격을 방지하기 위한 카운터 메커니즘이 구현된다. 예를 들어, 디바이스(101)로부터 허브(110)로의(또는 그 반대로의) 각각의 연속적인 통신이 계속 증가하는 카운터 값을 할당받을 수 있다. 허브(110) 및 디바이스(101) 둘 모두는 이 값을 추적하고 이 값이 디바이스들 간의 각각의 연속적인 통신에서 정확한지를 검증할 것이다. 동일한 기술이 허브(110)와 서비스(120) 사이에서 구현될 수 있다. 이러한 방식으로 카운터를 사용하는 것은 (카운터 값이 부정확할 것이기 때문에) 각각의 디바이스들 간의 통신을 스푸핑(spoofing)하는 것을 더욱 어렵게 할 것이다. 그러나, 이것 없이도, 서비스와 디바이스 사이의 공유 설치 키가 모든 디바이스에 대한 네트워크(허브) 전반적 공격을 방지할 것이다.

[0078] 일 실시예에서, 공개/비공개 키 암호화를 사용할 때, IoT 허브(110)는 이의 비공개 키를 사용하여, IoT 허브 패킷을 해독하고 암호화된 IoT 디바이스 패킷을 생성해서 이를 관련 IoT 디바이스(101)로 전송한다. 이어서, IoT 디바이스(101)는 이의 비공개 키를 사용하여 IoT 디바이스 패킷을 해독하여, IoT 서비스(120)로부터 시작되는 커맨드/데이터를 생성한다. 이어서, IoT 디바이스는 데이터를 처리하고/하거나 커맨드를 실행할 수 있다. 대칭 암호화를 사용하는 경우, 각각의 디바이스는 공유 대칭 키로 암호화하고 해독할 것이다. 어느 경우에나, 각각의 송신 디바이스는 또한 메시지를 해당 디바이스의 비공개 키로 서명하여, 수신 디바이스가 그것의 신빙성을 검증할 수 있도록 한다.

[0079] IoT 디바이스(101)로부터 IoT 허브(110)로의 그리고 IoT 서비스(120)로의 통신을 암호화하는 데 상이한 키 세트가 사용될 수 있다. 예를 들어, 공개/비공개 키 배열을 사용하는 경우, 일 실시예에서, IoT 디바이스(101) 상의 보안 로직(1002)은 IoT 허브(110)의 공개 키를 사용하여, IoT 허브(110)로 전송되는 데이터 패킷을 암호화한다. 이어서, IoT 허브(110) 상의 보안 로직(1012)은 IoT 허브의 비공개 키를 사용하여 데이터 패킷을 해독할 수 있다. 유사하게, IoT 디바이스(101) 상의 보안 로직(1002) 및/또는 IoT 허브(110) 상의 보안 로직(1012)은 IoT 서비스(120)의 공개 키를 사용하여, IoT 서비스(120)로 전송되는 데이터 패킷을 암호화할 수 있다(데이터 패킷은 이어서 IoT 서비스(120) 상의 보안 로직(1013)에 의해 서비스의 비공개 키를 사용하여 해독될 수 있다). 대칭 키를 사용하는 경우, 디바이스(101)와 허브(110)는 어느 한 대칭 키를 공유할 수 있는 반면, 허브와 서비스(120)는 다른 한 대칭 키를 공유할 수 있다.

[0080] 위의 설명에서는 소정의 특정 세부 내용들이 기재되어 있지만, 본 발명의 기본 원리는 다양한 상이한 암호화 기술을 사용하여 구현될 수 있다는 점에 유의해야 한다. 예를 들어, 상기에 논의된 일부 실시예는 비대칭 공개/비공개 키 쌍을 사용하지만, 대안적인 실시예는 다양한 IoT 디바이스(101, 102)와 IoT 허브(110)와 IoT 서비스(120) 사이에서 안전하게 교환되는 대칭 키를 사용할 수 있다. 더욱이, 일부 실시예에서는 데이터/커맨드 그 자체가 암호화되는 것이 아니라 키를 사용하여 데이터/커맨드(또는 다른 데이터 구조)에 대한 서명을 생성한다. 이어서, 수신자는 그것의 키를 사용하여 서명을 확인할 수 있다.

[0081] **도 11**에 예시된 바와 같이, 일 실시예에서, 각각의 IoT 디바이스(101) 상의 보안 키 저장소는 프로그래밍 가능 가입자 식별 모듈(SIM)(1101)을 사용하여 구현된다. 이 실시예에서, IoT 디바이스(101)는 초기에는 프로그래밍되지 않은 SIM 카드(1101)를 IoT 디바이스(101) 상의 SIM 인터페이스(1100) 내에 설치한 상태로 최종 사용자에게 제공될 수 있다. SIM에 하나 이상의 암호화 키의 세트를 프로그래밍하기 위해, 사용자는 프로그래밍 가능 SIM 카드(1101)를 SIM 인터페이스(500)에서 꺼내서 IoT 허브(110) 상의 SIM 프로그래밍 인터페이스(1102) 안에 삽입한다. 이어서, IoT 허브 상의 프로그래밍 로직(1125)이 SIM 카드(1101)를 안전하게 프로그래밍하여 IoT 디바이스(101)를 IoT 허브(110) 및 IoT 서비스(120)에 대해 등록/페어링한다. 일 실시예에서, 공개/비공개 키 쌍이 프로그래밍 로직(1125)에 의해 무작위로 생성될 수 있고, 이어서 쌍의 공개 키는 IoT 허브의 보안 저장 디바

이스(411)에 저장될 수 있는 반면, 비공개 키는 프로그래밍 가능 SIM(1101) 내에 저장될 수 있다. 게다가, 프로그래밍 로직(525)은 IoT 허브(110), IoT 서비스(120), 및/또는 임의의 다른 IoT 디바이스(101)의 공개 키를 (IoT 디바이스(101) 상의 보안 로직(1302)이 발신 데이터를 암호화하는 데 사용하도록) SIM 카드(1401)에 저장할 수 있다. 일단 SIM(1101)이 프로그래밍되면, 새로운 IoT 디바이스(101)는 SIM을 보안 식별자로 사용하여(예를 들어, SIM을 사용하여 디바이스를 등록하기 위한 기존 기술을 사용하여) IoT 서비스(120)로 프로비저닝될 수 있다. 프로비저닝 후에, IoT 허브(110) 및 IoT 서비스(120) 둘 모두는 IoT 디바이스(101)와의 통신을 암호화할 때 사용될 IoT 디바이스의 공개 키의 사본을 안전하게 저장할 것이다.

[0082] **도 11**과 관련하여 전술한 기술은 새로운 IoT 디바이스를 최종 사용자에게 제공할 때 엄청난 유연성을 제공한다. 사용자가 (현재 행해지고 있는 바와 같이) 판매/구매 시에 각각의 SIM을 특정 서비스 제공자에 직접 등록할 것을 요구하기보다는, 최종 사용자가 SIM을 IoT 허브(110)를 통해 직접 프로그래밍할 수 있고, 프로그래밍의 결과는 IoT 서비스(120)로 안전하게 전달될 수 있다. 결과적으로, 새로운 IoT 디바이스(101)가 온라인 또는 지역 소매상으로부터 최종 사용자에게 판매될 수 있고, 나중에 IoT 서비스(120)로 안전하게 프로비저닝될 수 있다.

[0083] 등록 및 암호화 기술이 SIM(가입자 식별 모듈)의 특정 맥락 내에서 위에 설명되었지만, 본 발명의 기본 원리는 "SIM" 디바이스로 제한되지 않는다. 오히려, 본 발명의 기본 원리는 암호화 키 세트를 저장하기 위한 보안 저장소를 갖는 임의의 타입의 디바이스를 사용하여 구현될 수 있다. 또한, 위의 실시예는 탈착식 SIM 디바이스를 포함하지만, 일 실시예에서, SIM 디바이스는 탈착식이 아니라, IoT 디바이스 그 자체가 IoT 허브(110) 상의 프로그래밍 인터페이스(1102) 내에 삽입될 수 있다.

[0084] 일 실시예에서, 사용자가 SIM(또는 다른 디바이스)을 프로그래밍할 것을 필요로 하기보다는, SIM은 최종 사용자에게 배포되기 전에 IoT 디바이스(101) 내에 미리 프로그래밍된다. 이 실시예에서, 사용자가 IoT 디바이스(101)를 설정할 때, IoT 허브(110)/IoT 서비스(120)와 새로운 IoT 디바이스(101) 사이에서 암호화 키를 안전하게 교환하는 데 본 명세서에 설명된 다양한 기술이 사용될 수 있다.

[0085] 예를 들어, **도 12a**에 예시된 바와 같이, 각각의 IoT 디바이스(101) 또는 SIM(401)은 IoT 디바이스(101) 및/또는 SIM(1001)을 고유하게 식별하는 바코드 또는 QR 코드(1501)와 함께 패키징될 수 있다. 일 실시예에서, 바코드 또는 QR 코드(1201)는 IoT 디바이스(101) 또는 SIM(1001)에 대한 공개 키의 인코딩된 표현을 포함한다. 대안적으로, 바코드 또는 QR 코드(1201)는 IoT 허브(110) 및/또는 IoT 서비스(120)가 공개 키를 식별하거나 생성하는 데 사용될 수 있다(예를 들어, 보안 저장소에 이미 저장된 공개 키에 대한 포인터로 사용될 수 있다). 바코드 또는 QR 코드(601)는 (**도 12a**에 도시된 바와 같이) 별도의 카드 상에 인쇄될 수 있거나 IoT 디바이스 그 자체 상에 직접 인쇄될 수 있다. 바코드가 어디에 인쇄되는지에 관계없이, 일 실시예에서, IoT 허브(110)에는 바코드를 판독하고 결과적인 데이터를 IoT 허브(110) 상의 보안 로직(1012) 및/또는 IoT 서비스(120) 상의 보안 로직(1013)에 제공하기 위한 바코드 판독기(206)가 탑재된다. 이어서, IoT 허브(110) 상의 보안 로직(1012)은 IoT 디바이스에 대한 공개 키를 그것의 보안 키 저장소(1011) 내에 저장할 수 있고, IoT 서비스(120) 상의 보안 로직(1013)은 공개 키를 (후속 암호화된 통신에 사용되도록) 그것의 보안 저장소(1021) 내에 저장할 수 있다.

[0086] 일 실시예에서, 바코드 또는 QR 코드(1201)에 포함되는 데이터는 또한 IoT 서비스 제공자에 의해 설계된 IoT 앱 또는 브라우저 기반 애플릿이 설치된 (예를 들어, 아이폰 또는 안드로이드 디바이스와 같은) 사용자 디바이스(135)를 통해 캡처될 수 있다. 일단 캡처되면, 바코드 데이터는 (예를 들어, 보안 소켓 계층(SSL) 접속과 같은) 보안 접속을 통해 IoT 서비스(120)로 안전하게 전달될 수 있다. 바코드 데이터는 또한 보안 로컬 접속을 통해(예를 들어, 로컬 WiFi 또는 블루투스 LE 접속을 통해) 클라이언트 디바이스(135)로부터 IoT 허브(110)로 제공될 수 있다.

[0087] IoT 디바이스(101) 상의 보안 로직(1002) 및 IoT 허브(110) 상의 보안 로직(1012)은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합을 사용하여 구현될 수 있다. 예를 들어, 일 실시예에서, 보안 로직(1002, 1012)은 IoT 디바이스(101)와 IoT 허브(110) 사이에 로컬 통신 채널(130)을 설정하는 데 사용되는 칩(예를 들어, 로컬 채널(130)이 블루투스 LE인 경우에 블루투스 LE 칩) 내에 구현된다. 보안 로직(1002, 1012)의 특정 위치에 관계없이, 일 실시예에서, 보안 로직(1002, 1012)은 소정 타입의 프로그램 코드를 실행하기 위한 보안 실행 환경을 확립하도록 설계된다. 이는 예를 들어 트러스트존(TrustZone) 기술(일부 ARM 프로세서 상에서 이용 가능함) 및/또는 신뢰 실행 기술(Trusted Execution Technology)(인텔(Intel)에 의해 설계됨)을 사용하여 구현될 수 있다. 물론, 본 발명의 기본 원리는 임의의 특정 타입의 보안 실행 기술로 제한되지 않는다.

[0088] 일 실시예에서, 바코드 또는 QR 코드(1501)는 각각의 IoT 디바이스(101)를 IoT 허브(110)와 페어링하는 데 사용될 수 있다. 예를 들어, 블루투스 LE 디바이스를 페어링하기 위해 현재 사용되는 표준 무선 페어링 프로세스를

사용하기보다는, 바코드 또는 QR 코드(1501) 내에 임베딩되는 페어링 코드가 IoT 허브를 대응하는 IoT 디바이스와 페어링하기 위해 IoT 허브(110)에 제공될 수 있다.

[0089] **도 12b**는 IoT 허브(110) 상의 바코드 판독기(206)가 IoT 디바이스(101)와 관련된 바코드/QR 코드(1201)를 캡처하는 일 실시예를 예시한다. 언급된 바와 같이, 바코드/QR 코드(1201)는 IoT 디바이스(101) 상에 직접 인쇄될 수 있거나, IoT 디바이스(101)가 제공된 별개의 카드 상에 인쇄될 수 있다. 어느 경우에나, 바코드 판독기(206)는 바코드/QR 코드(1201)로부터 페어링 코드를 판독하고 로컬 통신 모듈(1280)에 페어링 코드를 제공한다. 일 실시예에서, 로컬 통신 모듈(1280)은 블루투스 LE 칩 및 관련 소프트웨어이지만, 본 발명의 기본 원리는 임의의 특정 프로토콜 표준으로 제한되지 않는다. 페어링 코드가 수신되면, 그 페어링 코드는 페어링 데이터(1285)를 포함하는 보안 저장소에 저장되고, IoT 디바이스(101)와 IoT 허브(110)가 자동적으로 페어링된다. IoT 허브가 이러한 방식으로 새로운 IoT 디바이스와 페어링될 때마다, 그러한 페어링을 위한 페어링 데이터는 보안 저장소(685) 내에 저장된다. 일 실시예에서, IoT 허브(110)의 로컬 통신 모듈(1280)은 페어링 코드를 수신하면 그 페어링 코드를 IoT 디바이스(101)와의 로컬 무선 채널을 통한 통신을 암호화하기 위한 키로서 사용할 수 있다.

[0090] 유사하게, IoT 디바이스(101) 측에서, 로컬 통신 모듈(1590)은 로컬 보안 저장 디바이스(1595) 내에 IoT 허브와의 페어링을 지시하는 페어링 데이터를 저장한다. 페어링 데이터(1295)는 바코드/QR 코드(1201)에서 식별된 미리 프로그래밍된 페어링 코드를 포함할 수 있다. 페어링 데이터(1295)는 보안 로컬 통신 채널을 설정하는 데 필요한, IoT 허브(110) 상의 로컬 통신 모듈(1280)로부터 수신된 페어링 데이터(예를 들어, IoT 허브(110)와의 통신을 암호화하기 위한 추가 키)도 포함할 수 있다.

[0091] 따라서, 바코드/QR 코드(1201)는 페어링 코드가 무선으로 송신되지 않기 때문에 현재 무선 페어링 프로토콜보다 훨씬 더 안전한 방식으로 로컬 페어링을 수행하는 데 사용될 수 있다. 또한, 일 실시예에서, 페어링에 사용되는 동일한 바코드/QR 코드(1201)는 IoT 디바이스(101)로부터 IoT 허브(110)로의 그리고 IoT 허브(110)로부터 IoT 서비스(120)로의 보안 접속을 구축하기 위한 암호화 키를 식별하는 데 사용될 수 있다.

[0092] 본 발명의 일 실시예에 따른 SIM 카드를 프로그래밍하는 방법이 **도 13**에 예시되어 있다. 본 방법은 전술한 시스템 아키텍처 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0093] 1301에서, 사용자는 블랭크(blank) SIM 카드를 갖는 새로운 IoT 디바이스를 받고, 1602에서, 사용자는 블랭크 SIM 카드를 IoT 허브 안에 삽입한다. 1303에서, 사용자는 블랭크 SIM 카드에 하나 이상의 암호화 키의 세트를 프로그래밍한다. 예를 들어, 전술한 바와 같이, 일 실시예에서, IoT 허브는 공개/비공개 키 쌍을 무작위로 생성하고, SIM 카드 상에 비공개 키를 저장하고 그것의 로컬 보안 저장소에 공개 키를 저장할 수 있다. 또한, 1304에서, 적어도 공개 키가 IoT 서비스로 전송되어, 그 공개 키가 IoT 디바이스를 식별하고 IoT 디바이스와의 암호화된 통신을 확립하는 데 사용될 수 있다. 전술한 바와 같이, 일 실시예에서, "SIM" 카드 이외의 프로그래밍 가능 디바이스가 **도 13**에 도시된 방법에서 SIM 카드와 동일한 기능을 수행하는 데 사용될 수 있다.

[0094] 새로운 IoT 디바이스를 네트워크 안에 통합하는 방법이 **도 14**에 예시되어 있다. 본 방법은 전술한 시스템 아키텍처 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0095] 1401에서, 사용자는 암호화 키가 미리 할당된 새로운 IoT 디바이스를 수신한다. 1402에서, 키는 IoT 허브에 안전하게 제공된다. 전술한 바와 같이, 일 실시예에서, 이것은 디바이스에 할당된 공개/비공개 키 쌍의 공개 키를 식별하기 위해 IoT 디바이스와 관련된 바코드를 판독하는 것을 포함한다. 바코드는 IoT 허브에 의해 직접 판독되거나 모바일 디바이스를 통해 앱 또는 브라우저를 통해 캡처될 수 있다. 대안적인 실시예에서, 블루투스 LE 채널, 근거리장 통신(NFC) 채널, 또는 보안 WiFi 채널과 같은 보안 통신 채널이 IoT 디바이스와 IoT 허브 사이에 확립되어 키를 교환하도록 할 수 있다. 키는 전송되는 방식에 상관없이 일단 수신되면 IoT 허브 디바이스의 보안 키 저장소에 저장된다. 전술한 바와 같이, 키를 저장하고 보호하는 데에는 보안 엔클레이브(Secure Enclave), 신뢰 실행 기술(TXT), 및/또는 트러스트존과 같은 다양한 보안 실행 기술이 IoT 허브에서 사용될 수 있다. 또한, 803에서, 키는 IoT 서비스로 안전하게 송신되며, IoT 서비스는 키를 그 자신의 보안 키 저장소에 저장한다. 이어서, IoT 서비스는 키를 사용하여 IoT 디바이스와의 통신을 암호화할 수 있다. 다시 한번, 교환은 인증서/서명된 키를 사용하여 구현될 수 있다. 허브(110) 내에서, 저장된 키의 변경/추가/제거를 방지하는 것이 특히 중요하다.

[0096] 공개/비공개 키를 사용하여 커맨드/데이터를 IoT 디바이스로 안전하게 전달하는 방법이 **도 15**에 예시되어 있다. 본 방법은 전술한 시스템 아키텍처 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0097] 1501에서, IoT 서비스는 IoT 디바이스 공개 키를 사용하여 데이터/커맨드를 암호화하여 IoT 디바이스 패킷을 생성한다. IoT 서비스는 이어서 IoT 허브의 공개 키를 사용하여 IoT 디바이스 패킷을 암호화하여 IoT 허브 패킷을 생성한다(예를 들어, IoT 디바이스 패킷 주위에 IoT 허브 래퍼(wrapper)를 생성함). 1502에서, IoT 서비스는 IoT 허브 패킷을 IoT 허브로 송신한다. 1503에서, IoT 허브는 IoT 허브의 비공개 키를 사용하여 IoT 허브 패킷을 해독하여 IoT 디바이스 패킷을 생성한다. 1504에서, 그것은 이어서 IoT 디바이스 패킷을 IoT 디바이스로 송신하고, IoT 디바이스는 1505에서 IoT 디바이스 비공개 키를 사용하여 IoT 디바이스 패킷을 해독하여 데이터/커맨드를 생성한다. 1506에서, IoT 디바이스는 데이터/커맨드를 처리한다.

[0098] 대칭 키를 사용하는 실시예에서, 대칭 키 교환은 각 디바이스들 사이에서(예를 들어, 각 디바이스와 허브 사이에서 그리고 허브와 서비스 사이에서) 협상될 수 있다. 일단 키 교환이 완료되면, 각 전송 디바이스는 데이터를 수신 디바이스로 송신하기 전에 대칭 키를 사용하여 각각의 송신을 암호화 및/또는 서명한다.

[0099] **사물 인터넷(IoT) 시스템에서 보안 통신 채널을 확립하기 위한 장치 및 방법**

[0100] 본 발명의 일 실시예에서, 데이터의 암호화 및 해독은, (예를 들어, 사용자의 모바일 디바이스(611) 및/또는 IoT 허브(110)와 같은) 통신 채널을 지원하는 데 사용되는 중간 디바이스들에 관계없이, IoT 서비스(120)와 각각의 IoT 디바이스(101) 사이에서 수행된다. IoT 허브(110)를 통해 통신하는 일 실시예가 **도 16a**에 예시되고, IoT 허브를 필요로 하지 않는 다른 실시예가 **도 16b**에 예시된다.

[0101] 먼저 **도 16a**를 참조하면, IoT 디바이스(101)와 IoT 서비스(120) 사이의 통신을 암호화/해독하기 위해, IoT 서비스(120)는 "서비스 세션 키들"(1650)의 세트를 관리하는 암호화 엔진(1660)을 포함하고 각 IoT 디바이스(101)는 "디바이스 세션 키들"(1651)의 세트를 관리하는 암호화 엔진(1661)을 포함한다. 암호화 엔진들은 본 명세서에서 설명되는 보안/암호화 기술들을 수행할 때 세션 공개/비공개 키 쌍을 생성하고 그 쌍의 비공개 세션 키로의 액세스를 방지하기 위한 하드웨어 보안 모듈(1630, 1631) 및 도출된 비밀을 사용하여 키 스트림을 (무엇보다도 특히) 생성하기 위한 키 스트림 생성 모듈(1640, 1641)을 포함한 상이한 하드웨어 모듈들에 의존할 수 있다. 일 실시예에서, 서비스 세션 키들(1650) 및 디바이스 세션 키들(1651)은 관련된 공개/비공개 키 쌍들을 포함한다. 예를 들어, 일 실시예에서, IoT 디바이스(101) 상의 디바이스 세션 키들(1651)은 IoT 서비스(120)의 공개 키 및 IoT 디바이스(101)의 비공개 키를 포함한다. 아래에서 상세히 논의되는 바와 같이, 일 실시예에서, 보안 통신 세션을 확립하기 위해, 공개/비공개 세션 키 쌍들(1650, 1651)은, 각각, 각각의 암호화 엔진(1660, 1661)이 동일한 비밀을 생성하는 데 사용되며, 이 동일한 비밀은 이어서 SKGM들(1640, 1641)이 IoT 서비스(120)와 IoT 디바이스(101) 사이의 통신을 암호화 및 해독하기 위한 키 스트림을 생성하는 데 사용된다. 본 발명의 일 실시예에 따른 비밀의 생성 및 사용과 관련된 추가 세부사항들이 아래에 제공된다.

[0102] **도 16a**에서, 일단 키들(1650, 1651)을 사용하여 비밀이 생성되면, 클라이언트는 메시지를 클리어 트랜잭션(Clear transaction)(1611)으로 표시된 바와 같이 항상 IoT 서비스(120)를 통해 IoT 디바이스(101)로 전송할 것이다. 본 명세서에서 사용된 바와 같은 "클리어"는 기본 메시지가 본 명세서에서 설명되는 암호화 기술들을 사용하여 암호화되지 않는다는 것을 나타내려는 것이다. 그러나, 예시된 바와 같이, 일 실시예에서는 통신을 보호하기 위해 클라이언트 디바이스(611)와 IoT 서비스(120) 사이에 보안 소켓 계층(SSL) 채널 또는 다른 보안 채널(예를 들어, 인터넷 프로토콜 보안(IPSEC) 채널)이 확립된다. 이어서, 1602에서, IoT 서비스(120) 상의 암호화 엔진(1660)은 생성된 비밀을 사용하여 메시지를 암호화하고 암호화된 메시지를 IoT 허브(110)로 송신한다. 일 실시예에서는, 비밀을 사용하여 메시지를 직접 암호화하기보다는, 비밀과 카운터 값을 사용하여 각 메시지 패킷을 암호화하는 데 사용되는 키 스트림을 생성한다. 이 실시예의 세부사항들이 **도 17**과 관련하여 아래에서 설명된다.

[0103] 예시된 바와 같이, SSL 접속 또는 다른 보안 채널이 IoT 서비스(120)와 IoT 허브(110) 사이에 확립될 수 있다. 1603에서, IoT 허브(110)(일 실시예에서는 메시지를 해독하는 능력을 갖지 않음)는 암호화된 메시지를 (예를 들어, 블루투스 저에너지(BTLE) 통신 채널을 통해) IoT 디바이스로 송신한다. 이어서, IoT 디바이스(101) 상의 암호화 엔진(1661)은 비밀을 사용하여 메시지를 해독하고 메시지 내용을 처리할 수 있다. 비밀을 사용하여 키 스트림을 생성하는 실시예에서, 암호화 엔진(1661)은 비밀과 카운터 값을 사용하여 키 스트림을 생성한 다음 키 스트림을 메시지 패킷의 해독에 사용할 수 있다.

[0104] 메시지 자체는 IoT 서비스(120)와 IoT 디바이스(101) 사이의 임의의 형태의 통신을 포함할 수 있다. 예를 들어, 메시지는 IoT 디바이스(101)에 측정을 행하고 그 결과를 다시 클라이언트 디바이스(611)에 보고하는 것과 같은 특정 기능을 수행하도록 명령하는 커맨드 패킷을 포함할 수 있거나, IoT 디바이스(101)의 동작을 구성하는

구성 데이터를 포함할 수 있다.

[0105] 응답이 요구되는 경우, IoT 디바이스(101) 상의 암호화 엔진(1661)은 1604에서 비밀 또는 도출된 키 스트림을 사용하여 응답을 암호화하고 암호화된 응답을 IoT 허브(110)로 송신하며, 이 IoT 허브는 1605에서 응답을 IoT 서비스(120)로 전송한다. 이어서, IoT 서비스(120) 상의 암호화 엔진(1660)은 1606에서 비밀 또는 도출된 키 스트림을 사용하여 응답을 해독하고 (예를 들어, SSL 또는 다른 보안 통신 채널을 통해) 해독된 응답을 클라이언트 디바이스(611)로 송신한다.

[0106] **도 16b**는 IoT 허브를 필요로 하지 않는 실시예를 예시한다. 오히려, 이 실시예에서, IoT 디바이스(101)와 IoT 서비스(120) 사이의 통신은 (예를 들어, **도 6 내지 도 9b**와 관련하여 전술한 실시예들에서와 같이) 클라이언트 디바이스(611)를 통해 일어난다. 이 실시예에서, 클라이언트 디바이스(611)는 IoT 디바이스(101)로 메시지를 전송하기 위해, 1611에서, 메시지의 암호화되지 않은 버전을 IoT 서비스(120)로 송신한다. 1612에서, 암호화 엔진(1660)은 비밀 또는 도출된 키 스트림을 사용하여 메시지를 암호화하고 암호화된 메시지를 다시 클라이언트 디바이스(611)로 송신한다. 이어서, 1613에서, 클라이언트 디바이스(611)는 암호화된 메시지를 IoT 디바이스(101)에 전송하고 암호화 엔진(1661)은 비밀 또는 도출된 키 스트림을 사용하여 메시지를 해독한다. 이어서, IoT 디바이스(101)는 본 명세서에서 설명되는 바와 같이 메시지를 처리할 수 있다. 응답이 요구되는 경우, 암호화 엔진(1661)은 1614에서 비밀을 사용하여 응답을 암호화하고 암호화된 응답을 클라이언트 디바이스(611)로 송신하며, 이 클라이언트 디바이스는 1615에서 암호화된 응답을 IoT 서비스(120)로 전송한다. 이어서, 1616에서, 암호화 엔진(1660)은 응답을 해독하고 해독된 응답을 클라이언트 디바이스(611)로 송신한다.

[0107] **도 17**은 IoT 서비스(120)와 IoT 디바이스(101) 사이에서 초기에 수행될 수 있는 키 교환 및 키 스트림 생성을 예시한다. 일 실시예에서, 이러한 키 교환은 IoT 서비스(120) 및 IoT 디바이스(101)가 새로운 통신 세션을 확립할 때마다 수행될 수 있다. 대안적으로, 지정된 기간(예컨대, 하루, 일주일 등) 동안 키 교환이 수행될 수 있고 교환된 세션 키들이 사용될 수 있다. **도 17**에는 간략화를 위해 중간 디바이스가 도시되지 않지만, 통신은 IoT 허브(110) 및/또는 클라이언트 디바이스(611)를 통해 일어날 수 있다.

[0108] 일 실시예에서, IoT 서비스(120)의 암호화 엔진(1660)은 세션 공개/비공개 키 쌍을 생성하기 위해 (예를 들어, 아마존(Amazon)(등록상표)에 의해 제공되는 CloudHSM과 같은 것일 수 있는) HSM(1630)에 커맨드를 전송한다. HSM(1630)은 후속하여 키 쌍 중 비공개 세션 키로의 액세스를 방지할 수 있다. 유사하게, IoT 디바이스(101) 상의 암호화 엔진은 세션 공개/비공개 키 쌍을 생성하고 쌍의 세션 비공개 키로의 액세스를 방지하는 (예컨대, 아트멜 코포레이션(Atmel Corporation)(등록상표)으로부터의 Atecc508 HSM과 같은) HSM(1631)에 커맨드를 송신할 수 있다. 물론, 본 발명의 기본 원리들은 임의의 특정 타입의 암호화 엔진 또는 제조자로 제한되지 않는다.

[0109] 일 실시예에서, 1701에서, IoT 서비스(120)는 HSM(1630)을 사용하여 생성된 그의 세션 공개 키를 IoT 디바이스(101)로 송신한다. IoT 디바이스는 그의 HSM(1631)을 사용하여 그 자신의 세션 공개/비공개 키 쌍을 생성하고, 1702에서 그의 키 쌍 중 공개 키를 IoT 서비스(120)로 송신한다. 일 실시예에서, 암호화 엔진들(1660, 1661)은 타원 곡선 공개-비공개 키 쌍을 갖는 두 당사자가 공유 비밀을 확립할 수 있게 하는 익명 키 협약인 ECDH(Elliptic Curve Diffie-Hellman) 프로토콜을 사용한다. 일 실시예에서, 1703에서, IoT 서비스(120)의 암호화 엔진(1660)은 이들 기술을 사용하여 IoT 디바이스 세션 공개 키 및 그 자신의 세션 비공개 키를 사용하여 비밀을 생성한다. 유사하게, 1704에서, IoT 디바이스(101)의 암호화 엔진(1661)은 IoT 서비스(120) 세션 공개 키 및 그 자신의 세션 비공개 키를 사용하여 동일한 비밀을 독립적으로 생성한다. 더 구체적으로, 일 실시예에서, IoT 서비스(120) 상의 암호화 엔진(1660)은 공식 '비밀 = IoT 디바이스 세션 공개 키 * IoT 서비스 세션 비공개 키'에 따라 비밀을 생성하며, 여기서 '*'는 IoT 디바이스 세션 공개 키가 IoT 서비스 세션 비공개 키와 점 곱셈된다(point-multiplied)는 것을 의미한다. IoT 디바이스(101) 상의 암호화 엔진(1661)은 IoT 서비스 세션 공개 키가 IoT 디바이스 세션 비공개 키와 점 곱셈되는, '비밀 = IoT 서비스 세션 공개 키 * IoT 디바이스 세션 비공개 키'인 공식에 따라 비밀을 생성한다. 마침내, IoT 서비스(120)와 IoT 디바이스(101) 둘 다가 아래에 설명되는 바와 같이 통신을 암호화하는 데 사용될 동일한 비밀을 생성하였다. 일 실시예에서, 암호화 엔진들(1660, 1661)은 비밀을 생성하기 위한 상기 동작들을 수행하기 위해 각각 KSGM들(1640, 1641)과 같은 하드웨어 모듈에 의존한다.

[0110] 일단 비밀이 결정되면, 그 비밀은 암호화 엔진들(1660, 1661)이 데이터를 직접 암호화 및 해독하는 데 사용될 수 있다. 대안적으로, 일 실시예에서, 암호화 엔진들(1660, 1661)은 KSGM들(1640, 1641)에 각각의 데이터 패킷을 암호화/해독하기 위해 비밀을 사용하여 새로운 키 스트림을 생성하도록(즉, 새로운 키 스트림 데이터 구조가 각각의 패킷에 대해 생성됨) 하는 커맨드를 전송한다. 특히, 키 스트림 생성 모듈(1640, 1641)의 일 실시예는

카운터 값이 각각의 데이터 패킷에 대해 증분되어 비밀과 조합되어 사용되어서 키 스트림을 생성하게 되는 GCM(Galois/Counter Mode)을 구현한다. 따라서, 데이터 패킷을 IoT 서비스(120)로 송신하기 위해, IoT 디바이스(101)의 암호화 엔진(1661)은 비밀 및 현재 카운터 값을 사용하여, KSGM들(1640, 1641)로 하여금 새로운 키 스트림을 생성하게 하고 다음 키 스트림의 생성을 위해 카운터 값을 증분시키게 한다. 이어서, 새로 생성된 키 스트림은 IoT 서비스(120)로 송신되기 전에 데이터 패킷을 암호화하는 데 사용된다. 일 실시예에서, 키 스트림은 암호화된 데이터 패킷을 생성하기 위해 데이터와 XOR된다. 일 실시예에서, IoT 디바이스(101)는 카운터 값을 암호화된 데이터 패킷과 함께 IoT 서비스(120)로 송신한다. 이어서, IoT 서비스 상의 암호화 엔진(1660)은 KSGM(1640)과 통신하며, 이 KSGM은 수신된 카운터 값과 비밀을 사용하여 (동일한 비밀 및 카운터 값이 사용되기 때문에 동일한 키 스트림이어야 하는) 키 스트림을 생성하고 생성된 키 스트림을 사용하여 데이터 패킷을 해독한다.

[0111] 일 실시예에서, IoT 서비스(120)로부터 IoT 디바이스(101)로 송신되는 데이터 패킷들은 동일한 방식으로 암호화된다. 구체적으로, 카운터가 각각의 데이터 패킷에 대해 증분되어 비밀과 함께 사용되어서 새로운 키 스트림을 생성한다. 이어서, 키 스트림은 데이터를 암호화하는 데 사용되며(예를 들어, 데이터와 키 스트림의 XOR을 수행), 암호화된 데이터 패킷은 카운터 값과 함께 IoT 디바이스(101)로 송신된다. 이어서, IoT 디바이스(101) 상의 암호화 엔진(1661)은 데이터 패킷을 해독하는 데 사용되는 동일한 키 스트림을 생성하는 데 카운터 값과 비밀을 사용하는 KSGM(1641)과 통신한다. 따라서, 이 실시예에서, 암호화 엔진들(1660, 1661)은 그들 자신의 카운터 값들을 사용하여 키 스트림을 생성하여데이터를 암호화하고, 암호화된 데이터 패킷들과 함께 수신된 카운터 값들을 사용하여 키 스트림을 생성하여서 데이터를 해독한다.

[0112] 일 실시예에서, 각각의 암호화 엔진(1660, 1661)은 그가 다른 것으로부터 수신한 최종 카운터 값을 기록하며, 카운터 값이 비순차적으로 수신되는지 또는 그 카운터 값이 한 번을 초과하여 수신되는지를 검출하는 시퀀싱 로직을 포함한다. 카운터 값이 비순차적으로 수신되는 경우, 또는 그 카운터 값이 한 번을 초과하여 수신되는 경우, 이는 재생 공격이 시도되고 있음을 나타낼 수 있다. 이에 응답하여, 암호화 엔진들(1660, 1661)은 통신 채널로부터 접속 해제될 수 있고/있거나 보안 경고를 생성할 수 있다.

[0113] **도 18**은 4바이트 카운터 값(1800), 가변 크기의 암호화된 데이터 필드(1801), 및 6바이트 태그(1802)를 포함하는, 본 발명의 일 실시예에서 사용되는 예시적인 암호화된 데이터 패킷을 예시한다. 일 실시예에서, 태그(1802)는 (일단 해독되면) 해독된 데이터를 검증하기 위한 체크섬 값을 포함한다.

[0114] 언급된 바와 같이, 일 실시예에서, IoT 서비스(120)와 IoT 디바이스(101) 사이에서 교환되는 세션 공개/비공개 키 쌍들(1650, 1651)은 주기적으로 그리고/또는 각각의 새로운 통신 세션의 개시에 응답하여 생성될 수 있다.

[0115] 본 발명의 일 실시예는 IoT 서비스(120)와 IoT 디바이스(101) 사이의 세션들을 인증하기 위한 추가 기술들을 구현한다. 특히, 일 실시예에서, 마스터 키 쌍, 공장 키 쌍들의 세트, 및 IoT 서비스 키 쌍들의 세트, 및 IoT 디바이스 키 쌍들의 세트를 포함하는 공개/비공개 키 쌍들의 계층 구조가 사용된다. 일 실시예에서, 마스터 키 쌍은 모든 다른 키 쌍들에 대한 신뢰 루트를 포함하고, (예를 들어, 본 명세서에 설명되는 IoT 시스템들을 구현하는 조직의 제어 하에) 단일의 매우 안전한 위치에 유지된다. 마스터 비공개 키는 공장 키 쌍들과 같은 다양한 다른 키 쌍들을 통해 서명들을 생성(그리고 이에 의해 인증)하는 데 사용될 수 있다. 이어서, 서명들은 마스터 공개 키를 사용하여 검증될 수 있다. 일 실시예에서, IoT 디바이스들을 제조하는 각각의 공장은 그 자신의 공장 키 쌍을 할당받아서 그 키 쌍을 IoT 서비스 키들 및 IoT 디바이스 키들을 인증하는 데 사용할 수 있다. 예를 들어, 일 실시예에서, 공장 비공개 키는 IoT 서비스 공개 키들 및 IoT 디바이스 공개 키들을 통해 서명을 생성하는 데 사용된다. 이어서, 이러한 서명은 대응하는 공장 공개 키를 사용하여 검증될 수 있다. 이러한 IoT 서비스/장치 공개 키는 **도 16a 및 도 16b**와 관련하여 위에서 설명한 "세션" 공개/비공개 키와 동일하지 않다는 점에 유의한다. IoT 서비스/디바이스 키 쌍들이 영구적인(즉, 공장에서 생성된) 동안, 상술된 세션 공개/비공개 키들은 임시적인(즉, 서비스/디바이스 세션을 위해 생성된) 것이다.

[0116] 마스터 키들과 공장 키들과 서비스/디바이스 키들 사이의 전술한 관계들을 염두에 두고, 본 발명의 일 실시예는 다음의 동작들을 수행하여 IoT 서비스(120)와 IoT 디바이스(101) 사이에 인증 및 보안의 추가 계층들을 제공한다:

[0117] A. 일 실시예에서, IoT 서비스(120)는 초기에 다음을 포함하는 메시지를 생성한다:

[0118] 1. IoT 서비스의 고유 ID:

[0119] • IoT 서비스의 일련번호;

[0120] • 타임스탬프;

[0121] • 이 고유 ID에 서명하는 데 사용되는 공장 키의 ID;

[0122] • 고유 ID의 클래스(즉, 서비스);

[0123] • IoT 서비스의 공개 키

[0124] • 고유 ID를 통한 서명.

[0125] 2. 다음을 포함하는 공장 인증서:

[0126] • 타임스탬프;

[0127] • 인증서에 서명하는 데 사용되는 마스터 키의 ID

[0128] • 공장 공개 키

[0129] • 공장 인증서의 서명

[0130] 3. (**도 16a 및 도 16b**와 관련하여 전술한 바와 같은) IoT 서비스 세션 공개 키

[0131] 4. IoT 서비스 세션 공개 키 서명(예를 들어, IoT 서비스의 비공개 키로 서명됨)

[0132] B. 일 실시예에서, 메시지는 (후술하는) 협상 채널 상에서 IoT 디바이스로 전송된다. IoT 디바이스는 메시지를 분석하고:

[0133] 1. (메시지 페이로드에 존재하는 경우에만) 공장 인증서의 서명을 검증하고

[0134] 2. 고유 ID에 의해 식별된 키를 사용하여 고유 ID의 서명을 검증하고

[0135] 3. 고유 ID로부터의 IoT 서비스의 공개 키를 사용하여 IoT 서비스 세션 공개 키 서명을 검증하고

[0136] 4. IoT 서비스의 공개 키뿐만 아니라 IoT 서비스의 세션 공개 키를 저장하고

[0137] 5. IoT 디바이스 세션 키 쌍을 생성한다.

[0138] C. 이어서, IoT 디바이스는 다음을 포함하는 메시지를 생성한다:

[0139] 1. IoT 디바이스의 고유 ID

[0140] • IoT 디바이스 일련번호

[0141] • 타임스탬프

[0142] • 이 고유 ID에 서명하는 데 사용되는 공장 키의 ID

[0143] • 고유 ID의 클래스(즉, IoT 디바이스)

[0144] • IoT 디바이스의 공개 키

[0145] • 고유 ID의 서명

[0146] 2. IoT 디바이스의 세션 공개 키

[0147] 3. IoT 디바이스의 키로 서명된 (IoT 디바이스 세션 공개 키 + IoT 서비스 세션 공개 키)의 서명

[0148] D. 이 메시지는 IoT 서비스로 역전송된다. IoT 서비스는 메시지를 분석하고:

[0149] 1. 공장 공개 키를 사용하여 고유 ID의 서명을 검증하고

[0150] 2. IoT 디바이스의 공개 키를 사용하여 세션 공개 키들의 서명을 검증하고

[0151] 3. IoT 디바이스의 세션 공개 키를 저장하고

[0152] E. 이어서, IoT 서비스는 IoT 서비스 키로 서명된 (IoT 디바이스 세션 공개 키 + IoT 서비스 세션 공개 키)의 서명을 포함하는 메시지를 생성한다.

[0153] F. IoT 디바이스는 메시지를 분석하고:

[0154] 1. IoT 서비스의 공개 키를 사용하여 세션 공개 키들의 서명을 검증하고

[0155] 2. IoT 디바이스 세션 비공개 키 및 IoT 서비스의 세션 공개 키로부터 키 스트림을 생성하고

[0156] 3. 이어서, IoT 디바이스는 "메시징 이용 가능" 메시지를 전송한다.

[0157] G. 이어서, IoT 서비스는 다음을 수행한다:

[0158] 1. IoT 서비스 세션 비공개 키 및 IoT 디바이스의 세션 공개 키로부터 키 스트림을 생성하고

[0159] 2. 다음을 포함하는 메시징 채널 상에서 새로운 메시지를 생성하고:

[0160] • 난수 2바이트 값을 생성하고 저장하고

[0161] • (아래에 논의되는) 부메랑 속성 Id 및 난수 값을 갖는 속성 메시지를 설정한다.

[0162] H. IoT 디바이스는 메시지를 수신하고:

[0163] 1. 메시지를 해독하려고 시도하고

[0164] 2. 표시된 속성 Id에 대해 동일한 값을 가진 업데이트를 방출한다.

[0165] I. IoT 서비스는 메시지 페이로드가 부메랑 속성 업데이트를 포함하고 있음을 인식하고:

[0166] 1. 그의 페어링된 상태를 참으로 설정하고

[0167] 2. 협상자 채널 상에서 페어링 완료 메시지를 전송한다.

[0168] J. IoT 디바이스는 메시지를 수신하고 그의 페어링된 상태를 참으로 설정한다.

[0169] 상기의 기술들은 "IoT 서비스" 및 "IoT 디바이스"와 관련하여 설명되지만, 본 발명의 기본 원리들은 사용자 클라이언트 디바이스들, 서버들 및 인터넷 서비스들을 포함하는 임의의 2개의 디바이스 간의 보안 통신 채널을 확립하도록 구현될 수 있다.

[0170] 상기의 기술들은 (비밀이 한쪽 당사자로부터 다른 당사자에게 송신되는 현재의 블루투스 페어링 기술들과는 대조적으로) 비공개 키들이 결코 무선으로 공유되지 않기 때문에 매우 안전하다. 전체 대화를 듣는 공격자는 공유 비밀을 생성하기에 충분하지 않은 공개 키들만을 가질 것이다. 이러한 기술들은 또한 서명된 공개 키들을 교환함으로써 중간자 공격을 방지한다. 또한, GCM 및 별개의 카운터들이 각각의 디바이스에서 사용되기 때문에, 임의의 종류의 "재생 공격"(중간자가 데이터를 캡처하고 그것을 다시 전송함)이 방지된다. 일부 실시예들은 또한 비대칭 카운터들을 사용함으로써 재생 공격을 방지한다.

[0171] **디바이스들을 정식으로 페어링함이 없이 데이터 및 커맨드를 교환하는 기술**

[0172] GATT는 일반 속성 프로파일의 두문자어로서, 이는 2개의 블루투스 저에너지(BTLE) 디바이스가 데이터를 앞뒤로 전송하는 방식을 정의한다. GATT는 표의 각각의 엔트리에 대한 16 비트 특성 ID들을 사용하는 간단한 탐색표(lookup table) 내에 서비스들, 특성들, 및 관련 데이터를 저장하는 데 사용되는 속성 프로토콜(ATT)이라고 하는 일반 데이터 프로토콜을 사용한다. "특성들"은 때때로 "속성들"로 지칭된다는 점에 유의한다.

[0173] 블루투스 디바이스들 상에서, 가장 일반적으로 사용되는 특성은 (특성 ID 10752 (0x2A00)을 갖는) 디바이스 "이름"이다. 예를 들어, 블루투스 디바이스는 그의 주변의 다른 블루투스 디바이스들을, GATT를 사용하여 그러한 다른 블루투스 디바이스들에 의해 공개된 "이름" 특성을 판독함으로써 식별할 수 있다. 따라서, 블루투스 디바이스는 디바이스들을 정식으로 페어링/본딩함이 없이 데이터를 교환할 수 있는 고유한 능력을 갖는다("페어링"과 "본딩"은 때때로 상호 교환적으로 사용되며; 본 논의의 나머지는 용어 "페어링"을 사용할 것임에 유의한다).

[0174] 본 발명의 일 실시예는 BTLE 인에이블드(BTLE-enabled) IoT 디바이스들과 정식으로 페어링함이 없이 이들 디바

이스와 통신하기 위해 이러한 능력을 이용한다. 각각의 개별 IoT 디바이스와의 페어링은 각각의 디바이스와 페어링하는 데 필요한 시간의 양 때문에, 그리고 한 번에 단지 하나의 페어링된 접속만이 설정될 수 있기 때문에 극히 비효율적일 것이다.

[0175] **도 19**는 블루투스(BT) 디바이스(1910)가 페어링된 BT 접속을 정식으로 설정함이 없이 IoT 디바이스(101)의 BT 통신 모듈(1901)과 네트워크 소켓 추상화를 설정하는 하나의 특정 실시예를 예시한다. BT 디바이스(1910)는 **도 16a**에 도시된 바와 같은 IoT 허브(110) 및/또는 클라이언트 디바이스(611)에 포함될 수 있다. 예시된 바와 같이, BT 통신 모듈(1901)은 특성 ID들, 그러한 특성 ID들과 관련된 이름들 및 그러한 특성 ID들에 대한 값들의 리스트를 포함하는 데이터 구조를 유지한다. 각각의 특성에 대한 값은 현재 BT 표준에 따라 특성 ID에 의해 식별되는 20바이트 버퍼 내에 저장될 수 있다. 그러나, 본 발명의 기본 원리들은 임의의 특정 버퍼 크기로 제한되지 않는다.

[0176] **도 19**의 예에서, "이름" 특성은 "IoT 디바이스 14"의 특정 값을 할당받는 BT 정의 특성이다. 본 발명의 일 실시예는 BT 디바이스(1910)와 보안 통신 채널을 협상하는 데 사용될 추가 특성들의 제1 세트 및 BT 디바이스(1910)와의 암호화된 통신에 사용될 추가 특성들의 제2 세트를 지정한다. 특히, 예시된 예에서 특성 ID <65532>에 의해 식별되는 "협상 기입" 특성은 발신 협상 메시지들을 송신하는 데 사용될 수 있고, 특성 ID <65533>에 의해 식별되는 "협상 판독" 특성은 착신 협상 메시지들을 수신하는 데 사용될 수 있다. "협상 메시지들"은 본 명세서에서 설명되는 바와 같은 보안 통신 채널을 확립하기 위해 BT 디바이스(1910) 및 BT 통신 모듈(1901)에 의해 사용되는 메시지들을 포함할 수 있다. 예로서, **도 17**에서, IoT 디바이스(101)는 "협상 판독" 특성 <65533>을 통해 IoT 서비스 세션 공개 키(1701)를 수신할 수 있다. 키(1701)는 IoT 서비스(120)로부터 BTLE 인에이블드 IoT 허브(110) 또는 클라이언트 디바이스(611)로 송신될 수 있으며, 이어서 BTLE 인에이블드 IoT 허브(110) 또는 클라이언트 디바이스(611)는 GATT를 사용하여 특성 ID <65533>에 의해 식별되는 협상 판독 값 버퍼에 키(1701)를 기입할 수 있다. 이어서, IoT 디바이스 애플리케이션 로직(1902)은 특성 ID <65533>에 의해 식별되는 값 버퍼로부터 키(1701)를 판독하고 그 키를 전술한 바와 같이 처리할 수 있다(예를 들어, 그 키를 사용하여 비밀을 생성하고, 비밀을 사용하여 키 스트림을 생성하고, 기타 등등).

[0177] 키(1701)가 20바이트(일부 현재 구현들에서의 최대 버퍼 크기)보다 큰 경우, 그 키는 20바이트 부분들에 기입될 수 있다. 예를 들어, 처음 20바이트는 BT 통신 모듈(1903)에 의해 특성 ID <65533>에 기입되고 IoT 디바이스 애플리케이션 로직(1902)에 의해 판독될 수 있으며, 이어서 IoT 디바이스 애플리케이션 로직(1902)은 특성 ID <65532>에 의해 식별되는 협상 기입 값 버퍼에 수신 확인 메시지를 기입할 수 있다. GATT를 사용하여, BT 통신 모듈(1903)은 특성 ID <65532>로부터 이 수신 확인을 판독하고 그에 응답하여 키(1701)의 다음 20바이트를 특성 ID <65533>에 의해 식별되는 협상 판독 값 버퍼에 기입할 수 있다. 이러한 방식으로, 특성 ID <65532> 및 <65533>에 의해 정의되는 네트워크 소켓 추상화가 보안 통신 채널을 확립하는 데 사용되는 협상 메시지들을 교환하도록 확립된다.

[0178] 일 실시예에서, 일단 보안 통신 채널이 설정되면, (IoT 디바이스(101)로부터 암호화된 데이터 패킷들을 송신하기 위한) 특성 ID <65534> 및 (IoT 디바이스에 의해 암호화된 데이터 패킷들을 수신하기 위한) 특성 ID <65533>을 사용하여 제2 네트워크 소켓 추상화가 설정된다. 즉, BT 통신 모듈(1903)은 (예를 들어, **도 16a**의 암호화된 메시지(1603)와 같은) 송신할 암호화된 데이터 패킷을 갖는 경우에는 특성 ID <65533>에 의해 식별되는 메시지 판독 값 버퍼를 사용하여 암호화된 데이터 패킷을 한 번에 20바이트씩 기입하기 시작한다. 이어서, IoT 디바이스 애플리케이션 로직(1902)은 판독 값 버퍼로부터 암호화된 데이터 패킷을 한 번에 20바이트씩 판독하여, 특성 ID <65532>에 의해 식별되는 기입 값 버퍼를 통해 필요에 따라 수신 확인 메시지들을 BT 통신 모듈(1903)로 전송할 것이다.

[0179] 일 실시예에서, 후술하는 GET, SET 및 UPDATE의 커맨드들은 2개의 BT 통신 모듈(1901, 1903) 사이에서 데이터 및 커맨드를 교환하는 데 사용된다. 예를 들어, BT 통신 모듈(1903)은, 특성 ID <65533>을 식별하고, IoT 디바이스 애플리케이션 로직(1902)에 의해 후속 판독될 수 있는 특성 ID <65533>에 의해 식별되는 값 필드/버퍼 안에 기입할 SET 커맨드를 포함하는 패킷을 전송할 수 있다. IoT 디바이스(101)로부터 데이터를 검색하기 위해, BT 통신 모듈(1903)은 특성 ID <65534>에 의해 식별되는 값 필드/버퍼로 지향되는 GET 커맨드를 송신할 수 있다. GET 커맨드에 응답하여, BT 통신 모듈(1901)은 특성 ID <65534>에 의해 식별되는 값 필드/버퍼로부터의 데이터를 포함하는 UPDATE 패킷을 BT 통신 모듈(1903)로 송신할 수 있다. 또한, UPDATE 패킷들은 IoT 디바이스(101) 상에서의 특정 속성의 변경들에 응답하여 자동으로 송신될 수 있다. 예를 들어, IoT 디바이스가 조명 시스템과 관련되고 사용자가 전등을 끄는 경우, 조명 애플리케이션과 관련된 온/오프 속성에 대한 변경을 반영하

기 위해 UPDATE 패킷이 전송될 수 있다.

[0180] **도 20**은 본 발명의 일 실시예에 따른 GET, SET, 및 UPDATE에 대해 사용되는 예시적인 패킷 포맷들을 예시한다. 일 실시예에서, 이러한 패킷들은 협상에 이어서 메시지 기입 <65534> 및 메시지 판독 <65533> 채널들을 통해 송신된다. GET 패킷(2001)에서, 제1 1바이트 필드는 패킷을 GET 패킷으로서 식별하는 값(0X10)을 포함한다. 제2 1바이트 필드는 현재 GET 커맨드를 고유하게 식별하는(즉, GET 커맨드가 관련된 현재 트랜잭션을 식별하는) 요청 ID를 포함한다. 예를 들어, 서비스 또는 디바이스로부터 송신되는 GET 커맨드의 각각의 인스턴스는 상이한 요청 ID를 할당받을 수 있다. 이것은 예를 들어 카운터를 증가시키고 카운터 값을 요청 ID로서 사용함으로써 행해질 수 있다. 그러나, 본 발명의 기본 원리들은 요청 ID를 설정하기 위한 임의의 특정 방식으로 제한되지 않는다.

[0181] 2바이트 속성 ID는 패킷이 지향되는 애플리케이션별 속성을 식별한다. 예를 들어, GET 커맨드가 **도 19**에 예시된 IoT 디바이스(101)로 전송되고 있는 경우, 속성 ID는 요청되는 특정한 애플리케이션별 값을 식별하는 데 사용될 수 있다. 위의 예로 되돌아가면, GET 커맨드는 전등이 켜졌는지 또는 꺼졌는지(예를 들어, 1 = 온, 0 = 오프)를 식별하는 값을 포함하는 조명 시스템의 전력 상태와 같은 애플리케이션별 속성 ID로 보내질 수 있다. IoT 디바이스(101)가 도어와 관련된 보안 장치인 경우, 값 필드는 도어의 현재 상태(예를 들어, 1 = 열림, 0 = 닫힘)를 식별할 수 있다. GET 커맨드에 응답하여, 속성 ID에 의해 식별되는 현재 값을 포함하는 응답이 송신될 수 있다.

[0182] **도 20**에 예시된 SET 패킷(2002) 및 UPDATE 패킷(2003)은 또한 패킷의 타입(즉, SET 및 UPDATE)을 식별하는 제1 1바이트 필드, 요청 ID를 포함하는 제2 1바이트 필드, 및 애플리케이션 정의 속성을 식별하는 2바이트 속성 ID 필드를 포함한다. 또한, SET 패킷은 n바이트 값 데이터 필드에 포함된 데이터를 길이를 식별하는 2바이트 길이 값을 포함한다. 값 데이터 필드는 IoT 디바이스에서 실행될 커맨드 및/또는 소정의 방식으로(예를 들어, 원하는 파라미터를 설정하기 위해, IoT 디바이스의 전원을 끄기 위해, 기타 등등) IoT 디바이스의 동작을 구성하기 위한 구성 데이터를 포함할 수 있다. 예를 들어, IoT 디바이스(101)가 팬의 속도를 제어하는 경우, 값 필드는 현재 팬 속도를 반영할 수 있다.

[0183] UPDATE 패킷(2003)은 SET 커맨드의 결과들의 업데이트를 제공하기 위해 송신될 수 있다. UPDATE 패킷(2003)은 SET 커맨드의 결과들과 관련된 데이터를 포함할 수 있는 n바이트 값 데이터 필드의 길이를 식별하기 위한 2바이트 길이 값 필드를 포함한다. 또한, 1바이트 업데이트 상태 필드가 업데이트되는 변수의 현재 상태를 식별할 수 있다. 예를 들어, SET 커맨드가 IoT 디바이스에 의해 제어되는 전등을 끄려고 시도한 경우, 업데이트 상태 필드는 전등이 성공적으로 꺼졌는지를 나타낼 수 있다.

[0184] **도 21**은 SET 및 UPDATE 커맨드들을 포함하는 IoT 서비스(120)와 IoT 디바이스(101) 사이의 예시적인 트랜잭션 시퀀스를 예시한다. IoT 허브 및 사용자의 모바일 디바이스와 같은 중개 디바이스들은 본 발명의 기본 원리들을 모호하게 하는 것을 피하기 위해 도시되지 않는다. 2101에서, SET 커맨드(2101)는 IoT 서비스로부터 IoT 디바이스(101)로 송신되고 BT 통신 모듈(1901)에 의해 수신되며, 이 BT 통신 모듈은 2102에서 그에 응답하여 특성 ID에 의해 식별되는 GATT 값 버퍼를 업데이트한다. SET 커맨드는 2103에서 저전력 마이크로제어기(MCU)(200)에 의해(또는 **도 19**에 도시된 IoT 디바이스 애플리케이션 로직(1902)과 같은 저전력 MCU 상에서 실행되는 프로그램 코드에 의해) 값 버퍼로부터 판독된다. 2104에서, MCU(200) 또는 프로그램 코드는 SET 커맨드에 응답하여 동작을 수행한다. 예를 들어, SET 커맨드는 새로운 온도와 같은 새로운 구성 파라미터를 지정하는 속성 ID를 포함할 수 있거나, (IoT 디바이스가 "켜짐" 또는 저전력 상태로 들어가게 하기 위한) 온/오프와 같은 상태 값을 포함할 수 있다. 따라서, 2104에서, 새로운 값이 IoT 디바이스 안에 설정되고, UPDATE 커맨드가 2105에서 반환되며, 2106에서 실제 값이 GATT 값 필드에서 업데이트된다. 일부 경우에, 실제 값은 원하는 값과 동일할 것이다. 다른 경우에, 업데이트된 값은 상이할 수 있다(즉, 이는 IoT 디바이스(101)가 소정 타입의 값들을 업데이트하는 데 시간이 걸릴 수 있기 때문이다). 마지막으로, 2107에서, GATT 값 필드로부터의 실제 값을 포함하는 UPDATE 커맨드가 다시 IoT 서비스(120)로 송신된다.

[0185] **도 22**는 본 발명의 일 실시예에 따른 IoT 서비스와 IoT 디바이스 사이에 보안 통신 채널을 구현하기 위한 방법을 예시한다. 본 방법은 위에 기술된 네트워크 아키텍처들의 맥락 내에서 구현될 수 있지만, 임의의 특정 아키텍처로 제한되지 않는다.

[0186] 2201에서, IoT 서비스는 타원 곡선 디지털 서명 알고리즘(ECDSA) 인증서들을 사용하여 IoT 허브와 통신하기 위한 암호화된 채널을 생성한다. 2202에서, IoT 서비스는 세션 비밀을 사용하여 IoT 디바이스 패킷들 내의 데이터/커맨드를 암호화하여 암호화된 디바이스 패킷을 생성한다. 상기에 언급된 바와 같이, 세션 비밀은 IoT 디바

이스 및 IoT 서비스에 의해 독립적으로 생성될 수 있다. 2203에서, IoT 서비스는 암호화된 디바이스 패킷을 암호화된 채널을 통해 IoT 허브로 송신한다. 2204에서, 해독함이 없이, IoT 허브는 암호화된 디바이스 패킷을 IoT 디바이스로 전달한다. 2205에서, IoT 디바이스는 세션 비밀을 사용하여 암호화된 디바이스 패킷을 해독한다. 언급된 바와 같이, 일 실시예에서, 이것은 (암호화된 디바이스 패킷과 함께 제공된) 비밀 및 카운터 값을 사용하여 키 스트림을 생성한 후에 키 스트림을 사용하여 패킷을 해독함으로써 달성될 수 있다. 이어서, 2206에서, IoT 디바이스는 디바이스 패킷 내에 포함된 데이터 및/또는 커맨드를 추출하여 처리한다.

[0187] 따라서, 표준 페어링 기술들을 사용하여 BT 디바이스들을 정식으로 페어링함이 없이, 위의 기술들을 사용하여, 2개의 BT 인에이블드 디바이스 사이에 양방향 보안 네트워크 소켓 추상화가 설정될 수 있다. 이러한 기술들은 위에서는 IoT 서비스(120)와 통신하는 IoT 디바이스(101)와 관련하여 설명되었지만, 본 발명의 기본 원리들은 임의의 2개의 BT 인에이블드 디바이스 사이에서 보안 통신 채널을 협상하고 확립하도록 구현될 수 있다.

[0188] **도 23a 내지 도 23c**는 본 발명의 일 실시예에 따른 디바이스들을 페어링하기 위한 상세한 방법을 예시한다. 본 방법은 위에 기술된 시스템 아키텍처의 맥락 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0189] 2301에서, IoT 서비스는 IoT 서비스의 일련번호 및 공개 키를 포함하는 패킷을 생성한다. 2302에서, IoT 서비스는 공장 비공개 키를 사용하여 패킷에 서명한다. 2303에서, IoT 서비스는 패킷을 암호화된 채널을 통해 IoT 허브로 전송하고, 2304에서 IoT 허브는 패킷을 암호화되지 않은 채널을 통해 IoT 디바이스로 전송한다. 2305에서, IoT 디바이스는 패킷의 서명을 검증하고, 2306에서 IoT 디바이스는 IoT 디바이스의 일련번호 및 공개 키를 포함하는 패킷을 생성한다. 2307에서, IoT 디바이스는 공장 비공개 키를 사용하여 패킷에 서명하고, 2308에서 IoT 디바이스는 패킷을 암호화되지 않은 채널을 통해 IoT 허브로 전송한다.

[0190] 2309에서, IoT 허브는 패킷을 암호화된 채널을 통해 IoT 서비스로 전송하고, 2310에서 IoT 서비스는 패킷의 서명을 검증한다. 2311에서, IoT 서비스는 세션 키 쌍을 생성하고, 2312에서 IoT 서비스는 세션 공개 키를 포함하는 패킷을 생성한다. 이어서, IoT 서비스는 2313에서 IoT 서비스 비공개 키로 패킷에 서명하고, 2314에서 IoT 서비스는 패킷을 암호화된 채널을 통해 IoT 허브로 전송한다.

[0191] **도 23b**를 참조하면, IoT 허브는 2315에서 패킷을 암호화되지 않은 채널을 통해 IoT 디바이스로 전송하고, 2316에서 IoT 디바이스는 패킷의 서명을 검증한다. 2317에서, IoT 디바이스는 (예를 들어, 전술한 기술을 사용하여) 세션 키 쌍을 생성하고, 2318에서 IoT 디바이스 세션 공개 키를 포함하는 IoT 디바이스 패킷이 생성된다. 2319에서, IoT 디바이스는 IoT 디바이스 비공개 키로 IoT 디바이스 패킷에 서명한다. 2320에서, IoT 디바이스는 패킷을 암호화되지 않은 채널을 통해 IoT 허브로 전송하고, 2321에서 IoT 허브는 패킷을 암호화된 채널을 통해 IoT 서비스로 전송한다.

[0192] 2322에서, IoT 서비스는 (예를 들어, IoT 디바이스 공개 키를 사용하여) 패킷의 서명을 검증하고, 2323에서 IoT 서비스는 (위에서 상세히 설명된 바와 같이) IoT 서비스 비공개 키 및 IoT 디바이스 공개 키를 사용하여 세션 비밀을 생성한다. 2324에서, IoT 디바이스는 (역시, 전술한 바와 같이) IoT 디바이스 비공개 키 및 IoT 서비스 공개 키를 사용하여 세션 비밀을 생성하고, 2325에서 IoT 디바이스는 난수를 생성하고 그것을 세션 비밀을 사용하여 암호화한다. 2326에서, IoT 서비스는 암호화된 패킷을 암호화된 채널을 통해 IoT 허브로 전송한다. 2327에서, IoT 허브는 암호화된 패킷을 암호화되지 않은 채널을 통해 IoT 디바이스로 전송한다. 2328에서, IoT 디바이스는 세션 비밀을 사용하여 패킷을 해독한다.

[0193] **도 23c**를 참조하면, IoT 디바이스는 2329에서 세션 비밀을 사용하여 패킷을 재암호화하고, 2330에서 IoT 디바이스는 암호화된 패킷을 암호화되지 않은 채널을 통해 IoT 허브로 전송한다. 2331에서, IoT 허브는 암호화된 패킷을 암호화된 채널을 통해 IoT 서비스로 전송한다. IoT 서비스는 2332에서 세션 비밀을 사용하여 패킷을 해독한다. 2333에서, IoT 서비스는 난수가 그가 전송한 난수와 일치하는지를 검증한다. 이어서, IoT 서비스는 2334에서 페어링이 완료되었음을 지시하는 패킷을 전송하고, 2335에서 모든 후속 메시지들이 세션 비밀을 사용하여 암호화된다.

[0194] **데이터 전송 조건을 식별하기 위해 패킷 간격 타이밍을 수정하는 장치 및 방법**

[0195] 블루투스 저에너지(BTLE) 디바이스는 디바이스들 간의 접속을 확립하기 위해 "광고 간격"으로 분리된 광고 패킷들을 전송한다. BTLE 주변 디바이스는 이 주변의 모든 디바이스에 광고 간격을 사용하여 광고 패킷들을 브로드캐스팅한다. 이어서, 수신 BTLE 디바이스는 이 정보에 따라 동작하거나 더 많은 정보를 수신하도록 접속할 수 있다.

[0196] BTLE에 대한 2.4 ㎓ 스펙트럼은 2402 ㎒에서 2480 ㎒까지에 걸쳐 있고, 0에서 39까지 번호가 매겨진 40개의 1 ㎒ 와이드 채널을 사용한다. 각 채널은 2 ㎒씩 분리된다. 채널 37, 채널 38, 채널 39는 광고 패킷을 전송하는 데에만 사용된다. 나머지 채널은 접속 중 데이터 교환에 사용된다. BTLE 광고 동안, BTLE 주변 디바이스는 패킷들을 3개의 광고 채널에서 차례로 송신한다. 디바이스 또는 비콘을 스캐닝하는 중앙 디바이스는 광고 패킷에 대한 그러한 채널들을 청취할 것이며, 이는 그것이 인근 디바이스를 발견하도록 돕는다. 채널 37, 채널 38, 채널 39는 의도적으로 2.4 ㎓ 스펙트럼에 걸쳐 분산된다(즉, 채널 37과 채널 39는 대역의 첫 번째 및 마지막 채널이고 채널 38은 중간에 있다). 임의의 단일 광고 채널이 차단되는 경우, 다른 채널들은 수 MHz의 대역폭만큼 분리되어 있기 때문에 자유로울 가능성이 있다.

[0197] IoT 디바이스가 송신될 데이터를 가질 때, IoT 디바이스는 통상적으로는 데이터가 전송될 준비가 되었음을 나타내는 플래그를 그의 광고 패킷의 일부로 포함할 것이다. 본 발명의 일 실시예에서, IoT 디바이스는 이러한 플래그를 사용하기보다는 그가 보류 중인 데이터를 갖고 있음을 나타내기 위해 광고 간격을 조정한다. 예를 들어, T가 데이터가 보류 중이 아닌 때의 광고 패킷들 간의 시간인 경우, 0.75T, 0.5T, 또는 1.25T와 같은 상이한 광고 간격을 선택하여 데이터가 보류 중임을 나타낼 수 있다. 일 실시예에서, 2개의 상이한 간격은 애플리케이션의 특정 요구에 기반하여 프로그래밍 가능하고, 어떤 간격이 어떤 상태를 의미하는지를 결정하는 것을 더 어렵게 만든다.

[0198] **도 24**는 BTLE 통신 인터페이스(2410)가 데이터 송신 준비가 되었을 때 광고 간격을 조정하는 광고 간격 선택 로직(2411)을 포함하는 IoT 디바이스(101)의 일 실시예를 예시한다. 또한, IoT 허브(110) 상의 BTLE 통신 인터페이스(2420)는 광고 간격의 변화를 검출하고 확인 응답을 제공하며 데이터를 수신하는 광고 간격 검출 로직(2421)을 포함한다.

[0199] 특히, 예시된 실시예에서, IoT 디바이스(101) 상의 애플리케이션(2401)은 IoT 디바이스가 전송될 데이터를 가지고 있음을 나타낸다. 이에 응답하여, 광고 간격 선택 로직(2411)은 광고 간격을 수정하여 IoT 허브(110)에 데이터가 송신될 것임을 통지한다(예를 들어, 간격을 0.75T 또는 어떤 다른 값으로 변경). 광고 간격 검출 로직(2421)이 이러한 변경을 검출하면, BTLE 통신 인터페이스(2420)는 IoT 디바이스(101)의 BTLE 통신 인터페이스(2410)에 접속하여 데이터를 수신할 준비가 되었음을 나타낸다. 이어서, IoT 디바이스(101)의 BTLE 통신 인터페이스(2410)는 IoT 허브의 BTLE 통신 인터페이스(2420)에 데이터를 송신한다. 이어서, IoT 허브는 데이터를 IoT 서비스(120) 및/또는 사용자의 클라이언트 디바이스(도시되지 않음)로 통과시킬 수 있다. 데이터가 송신된 후에, 이어서 광고 간격 선택 로직(2411)은 통상의 광고 간격(예를 들어, AI=T)으로 되돌아갈 수 있다.

[0200] 본 발명의 일 실시예에서, 전술한 하나 이상의 보안/암호화 기술을 사용하여 IoT 디바이스(101)와 IoT 서비스(120) 간에 보안 통신 채널이 확립된다(예를 들어, **도 16a 내지 도 23c** 및 관련 텍스트 참조). 예를 들어, 일 실시예에서, IoT 서비스(120)는 IoT 디바이스(101)와 IoT 서비스(120) 사이의 모든 통신을 암호화하기 위해 전술한 바와 같이 IoT 디바이스(101)와의 키 교환을 수행한다.

[0201] 본 발명의 일 실시예에 따른 방법이 **도 25**에 도시된다. 본 방법은 위에 기술된 시스템 아키텍처의 맥락 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0202] 2500에서, IoT 디바이스는 광고 패킷을 생성할 때 표준 광고 간격을 사용한다(예를 들어, 시간(T)으로 분리됨). IoT 디바이스는, 2501에서 결정된, 전송할 데이터가 있을 때까지 2502에서 표준 광고 간격을 유지한다. 이어서, 2503에서, IoT 디바이스는 광고 간격을 전환하여 송신할 데이터가 있음을 나타낸다. 2504에서, IoT 허브 또는 다른 네트워크 디바이스는 IoT 디바이스와의 접속을 확립함으로써, IoT 디바이스가 자신의 데이터를 송신할 수 있게 한다. 마지막으로, 2505에서, IoT 디바이스는 자신의 보류 중인 데이터를 IoT 허브로 송신한다.

[0203] 본 명세서에서 광고 간격 기술이 BTLE 프로토콜의 맥락 내에서 설명되지만, 본 발명의 기본 원리는 BTLE에 제한되지 않는다는 점에 주의해야 한다. 실제로, 본 발명의 기본 원리는 디바이스들 사이의 무선 통신을 확립하기 위한 광고 간격을 선택하는 임의의 시스템에서 구현될 수 있다.

[0204] 또한, 상기의 많은 실시예에서는 전용 IoT 허브(110)가 예시되어 있지만, 본 발명의 기본 원리를 준수하는 데 전용 IoT 허브 하드웨어 플랫폼이 요구되는 것은 아니다. 예를 들어, 상술한 다양한 IoT 허브는 아이폰(iPhone)® 및 안드로이드(Android)® 디바이스와 같은 다양한 다른 네트워킹 디바이스에서 실행되는 소프트웨어로 구현될 수 있다. 실제로, 위에서 논의한 IoT 허브는 IoT 디바이스와 통신하고(예를 들어, BTLE 또는 다른 로컬 무선 프로토콜을 사용) 인터넷을 통해 (예를 들어, WiFi 또는 셀룰러 데이터 접속을 사용하여 IoT 서비스로의) 접속을 확립할 수 있는 임의의 디바이스에서 구현될 수 있다.

[0205] **IoT 허브를 IoT 디바이스에 접속시킬 때 무선 트래픽을 감소시키는 시스템 및 방법**

[0206] 다수의 IoT 허브가 특정 위치에서 구성될 때, 단일 IoT 디바이스는 범위 내의 각각의 IoT 허브와 접속할 능력을 가질 수 있다. 언급한 바와 같이, IoT 디바이스는 IoT 허브가 커맨드 및/또는 데이터를 송신하기 위해 그에 접속될 수 있도록 "접속 가능한" 범위 내의 임의의 IoT 허브들에 통지하기 위해 광고 채널을 사용할 수 있다. 다수의 IoT 허브가 IoT 디바이스의 범위 내에 있을 때, IoT 서비스는 이들 IoT 허브 각각을 통해 IoT 디바이스로 어드레싱되는 커맨드/데이터를 송신하려고 시도할 수 있고, 이에 의해 (예를 들어, 다중 전송들로 인한 간섭으로 인하여) 무선 대역폭이 낭비되고 성능이 감소된다.

[0207] 이 문제를 해결하기 위해, 본 발명의 일 실시예는 특정 IoT 허브가 IoT 디바이스에 성공적으로 접속되면 다른 IoT 허브는 커맨드/데이터를 송신하려는 시도를 중단하라는 통지를 받게 되는 것을 보장하는 기술을 구현한다. 이 실시예는 모두 IoT 디바이스(101)의 범위 내에 있는 IoT 허브들(110 내지 112)의 예시적인 세트를 도시하는 **도 26a 내지 도 26c**와 관련하여 설명될 것이다. 그 결과, IoT 디바이스(101)의 보안 무선 통신 모듈(2610)은 IoT 허브들(110 내지 112) 각각의 보안 무선 통신 모듈들(2650 내지 2652)을 보고 그에 접속할 수 있다. 일 실시예에서, 보안 무선 통신 모듈은 전술한 보안 BTLE 모듈을 포함한다. 그러나, 본 발명의 기본 원리들은 임의의 특정 무선 표준으로 제한되지 않는다.

[0208] **도 26a**에 예시된 바와 같이, 일 실시예에서, IoT 디바이스(101)의 보안 무선 통신 모듈(2610)은 그에 "접속 가능"하다(즉, 범위 내의 임의의 디바이스가 그에 접속될 수 있다)는 것을 나타내는 광고 비콘을 인근 무선 통신 디바이스들에 주기적으로 송신하는 광고 제어 로직(2610)을 포함한다. 이어서, 광고 비콘을 수신하는 임의의 IoT 허브들(110 내지 112)은 IoT 디바이스(101)를 인식하고, 보안 무선 통신 모듈들(2650 내지 2652)은 커맨드/데이터가 IoT 서비스에 의해 IoT 디바이스(101)로 어드레싱되었을 때 IoT 디바이스(101)의 보안 무선 통신 모듈(2610)에 접속할 수 있다.

[0209] **도 26b**에 예시된 바와 같이, 일 실시예에서, IoT 서비스가 IoT 디바이스(101)에 대한 데이터/커맨드를 가질 때, IoT 서비스는 데이터/커맨드를 특정 위치 내의 IoT 허브들(110 내지 112) 모두(예를 들어, 사용자의 계정과 관련된 및/또는 IoT 디바이스(101)의 범위 내의 모든 IoT 허브)로 송신할 수 있다. 예시된 바와 같이, 이어서, IoT 허브들(110 내지 112) 각각은 커맨드/데이터를 제공하기 위해 IoT 디바이스(101)와 접속하려고 시도할 수 있다.

[0210] **도 26c**에 예시된 바와 같이, 일 실시예에서, 단일의 IoT 허브(111)만이 IoT 디바이스(101)에 성공적으로 접속하고 IoT 디바이스(101)에 의한 처리를 위한 커맨드/데이터를 제공할 것이다. BTLE와 같은 소정 무선 통신 프로토콜에서, 일단 접속이 이루어지면, 보안 무선 통신 모듈(2610)은 광고 비콘을 송신하는 것을 중단할 것이다. 그렇기 때문에, 다른 IoT 허브들(110, 112)은 IoT 디바이스(101)가 IoT 허브(111)로부터 데이터를 성공적으로 수신했다는 것을 알 수 있는 어떠한 방법도 갖지 않을 것이며, 커맨드/데이터를 송신하려고 계속 시도할 것이고, 그에 따라 무선 대역폭을 소비하고 간섭을 생성한다.

[0211] 이러한 제한을 해결하기 위해, 보안 무선 통신 모듈(2610)의 일 실시예는, IoT 허브(111)의 보안 무선 통신 모듈(2651)과의 성공적인 접속을 검출할 때, 광고 제어 모듈(2612)이 광고 비콘을 계속 송신하게 하는 접속 관리자(2611)를 포함한다. 그러나, IoT 디바이스(101)가 "접속 가능"을 나타내는 대신에, 새로운 광고 비콘은 IoT 디바이스(101)가 "접속 불가"를 나타낸다. 일 실시예에서, "접속 불가" 표시에 응답하여, IoT 허브들(110, 112)의 보안 무선 통신 모듈들(2650, 2652)은 IoT 디바이스에 커맨드/데이터를 송신하려는 시도를 중단함으로써 불필요한 무선 트래픽을 감소시킬 것이다.

[0212] 상기 기술들은 기존의 무선 프로토콜들 외에 쉽게 구현될 수 있는 기술들을 사용하여 바람직하지 않은 무선 트래픽에 대한 훌륭한 해결책을 제공한다. 예를 들어, 일 실시예에서, "접속 가능" 및 "접속 불가" 표시는 BTLE 표준의 맥락 내에서 구현된다. 그러나, 언급한 바와 같이, 본 발명의 기본 원리는 다양한 상이한 무선 네트워크 프로토콜을 사용하여 구현될 수 있다.

[0213] 본 발명의 일 실시예에 따른 방법이 **도 27**에 도시된다. 본 방법은 위에 기술된 시스템 아키텍처의 맥락 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0214] 2701에서, 커맨드 및/또는 데이터는 2개 이상의 IoT 허브를 통해 IoT 서비스로부터 송신된다. 예를 들어, 사용자는 IoT 서비스에 접속된 사용자의 모바일 디바이스 상의 앱을 통해 IoT 디바이스를 제어하려고 시도하고 있을 수 있다. 2702에서, IoT 허브는 IoT 디바이스에 접속하려고 시도하고 IoT 허브들 중 하나가 성공적으로 접속하고 IoT 디바이스에 커맨드/데이터를 제공한다. 언급한 바와 같이, IoT 허브는 IoT 디바이스가 광고 비콘에서

"접속 가능" 표시를 송신하는 결과로서 IoT 디바이스를 인식할 수 있다.

[0215] 2703에서, 성공적인 접속에 응답하여, IoT 디바이스는 "접속 불가" 광고 비콘을 송신하기 시작하고, 이에 의해 IoT 디바이스가 더 이상 접속 가능하지 않음을 범위 내의 임의의 IoT 허브에 알린다. 2704에서, 다른 IoT 허브들은 "접속 불가" 비콘을 수신하면 커맨드/데이터를 IoT 디바이스로 송신하려는 시도를 중단한다.

[0216] **보안 사물 인터넷(IoT) 디바이스 프로비저닝을 위한 시스템 및 방법**

[0217] 전술한 바와 같이, 일 실시예에서, 디바이스가 IoT 허브에 광고할 때, 그 디바이스는 IoT 디바이스를 고유하게 식별하기 위해 허브 및 IoT 서비스가 사용하는 8바이트의 "디바이스 ID"를 사용한다. 디바이스 ID는, 판독되고 시스템에 IoT 디바이스를 프로비저닝/등록하기 위해 IoT 서비스로 송신되는, IoT 디바이스에 인쇄된 고유 바코드 또는 QR 코드 내에 포함될 수 있다. 프로비저닝/등록되면, 디바이스 ID는 시스템의 IoT 디바이스를 어드레싱하는 데 사용된다.

[0218] 이 구현과 관련한 한 가지 보안 문제는, 바코드/QR 코드 데이터가 암호화 없이 송신될 수 있기 때문에, 디바이스 ID의 무선 전송을 스니핑(sniffing)하여 시스템을 손상시킬 수 있어서, 다른 사용자가 디바이스 ID를 자신의 계정과 관련시킬 수 있게 한다는 점이다.

[0219] 이러한 문제를 해결하기 위해, 일 실시예에서, "관련 ID"는 각 디바이스 ID와 관련되고 프로비저닝 프로세스 중에 사용되어, 해당 디바이스 ID가 명확히 결코 전송되지 않도록 하는 것을 보장한다. **도 28**에 예시된 바와 같이, 이 실시예에서, 관련 ID(2812)는 IoT 디바이스(101) 상에 인쇄된 바코드/QR 코드에 포함되는 반면에 디바이스 ID(2811)는 IoT 서비스(120)와의 보안 통신을 보장하기 위해 전술한 기술을 구현하는 보안 무선 통신 모듈(2810) 내에 안전하게 유지된다. 일 실시예에서, 관련 ID(2812)는 디바이스 ID와 같이 8바이트 ID이고 IoT 디바이스마다 고유하다. 새로운 IoT 디바이스(101)가 시스템에 프로비저닝될 때, 사용자는 IoT 앱 또는 애플리케이션이 설치된 사용자 디바이스(135)를 이용하여 관련 ID(2812)를 포함하는 바코드/QR 코드를 스캔한다. 대안적으로, 또는 부가적으로, IoT 허브(110)는 관련 ID를 포함하는 바코드/QR 코드를 캡처하는 데 사용될 수 있다.

[0220] 어느 경우이든, 관련 ID는 각각의 관련 ID와 각각의 디바이스 ID 사이의 관련성을 포함하는 디바이스 데이터베이스(2851)에서 탐색을 수행하는 IoT 서비스(120) 상의 디바이스 프로비저닝 모듈(2850)로 송신된다. 디바이스 프로비저닝 모듈(2850)은 관련 ID(2812)를 사용하여 디바이스 ID(2811)를 식별하고, 이어서 디바이스 ID를 사용하여 시스템에 새로운 IoT 디바이스(101)를 프로비저닝한다. 특히, 디바이스 ID가 디바이스 데이터베이스(2851)로부터 결정되면, 디바이스 프로비저닝 모듈(2850)은 디바이스 ID(2811)를 사용하여 IoT 허브(110)가 IoT 디바이스(101)와 통신하도록 인가하는 커맨드를 IoT 허브(110)(사용자 디바이스(135)를 포함할 수 있음)로 송신한다.

[0221] 일 실시예에서, 관련 ID(2812)는 IoT 디바이스(101)가 제조될 때(즉, 보안 무선 통신 모듈(2810)이 프로비저닝될 때) 공장에서 생성된다. 이어서, 디바이스 ID(2811) 및 관련 ID(2812) 둘 모두가 IoT 서비스에 제공될 수 있고 디바이스 데이터베이스(2851) 내에 저장될 수 있다. 예시된 바와 같이, 디바이스 데이터베이스(2851)는 각각의 디바이스가 프로비저닝되었는지 여부를 명시하는 표시를 포함할 수 있다. 예로서, 이는 IoT 디바이스(101)가 프로비저닝됨을 표시하는 제1 값(예를 들어, 1) 및 IoT 디바이스가 프로비저닝되지 않음을 표시하는 제2 값(예를 들어, 0)을 갖는 2진 값일 수 있다. 시스템이 IoT 디바이스(101)를 프로비저닝/등록하면, IoT 서비스(120)와 IoT 디바이스(101) 사이의 통신이 전술한 보안 기술을 사용하여 보호되기 때문에 디바이스 ID가 사용될 수 있다.

[0222] 일 실시예에서, 사용자가 IoT 디바이스를 판매할 때, 사용자는 IoT 서비스(120)에 로그인하고 사용자의 계정으로부터 IoT 디바이스를 해제함으로써 디바이스 ID를 해제할 수 있다. 이어서, 새로운 사용자는 본 명세서에 설명된 디바이스 프로비저닝 기술을 사용하여 IoT 디바이스를 프로비저닝하고 IoT 디바이스를 자신의 계정과 관련시킬 수 있다.

[0223] 본 발명의 일 실시예에 따른 방법이 **도 29**에 도시된다. 본 방법은 위에 기술된 시스템 아키텍처의 맥락 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0224] 2901에서, 디바이스 ID와 IoT 디바이스의 관련 ID 사이의 관련성이 생성된다(예를 들어, IoT 디바이스가 제조되는 공장에서). 관련 ID는 IoT 디바이스에 스탬핑된 바코드/QR 코드 내에 임베딩될 수 있다. 2902에서, 디바이스 ID와 관련 ID 사이의 관련성이 IoT 서비스에 저장된다. 2903에서, 사용자는 새로운 IoT 디바이스를 구입하고 (예를 들어, 앱 또는 애플리케이션이 설치된 사용자의 모바일 디바이스를 통해 또는 바코드 판독기를 갖는

IoT 허브를 통해) 관련 ID를 포함하는 바코드/QR 코드를 스캔한다.

[0225] 2904에서, 관련 ID는 IoT 서비스로 송신되고, 2905에서, 관련 ID는 디바이스 ID를 식별하는 데 사용된다. 2906에서, IoT 디바이스는 디바이스 ID를 사용하여 프로비저닝된다. 예를 들어, IoT 디바이스 데이터베이스는 이 특정 디바이스 ID가 프로비저닝되었음을 표시하도록 업데이트될 수 있고, IoT 서비스는 그 디바이스 ID를 IoT 허브로 전하여서, 그 IoT 허브에게 새로운 IoT 디바이스와 통신하도록 명령할 수 있다.

[0226] **사물 인터넷(IoT) 시스템 내에서 흐름 제어를 수행하는 시스템 및 방법**

[0227] 로컬 무선 네트워크 트래픽은 주어진 위치 내의 IoT 디바이스들의 수에 의거하여 증가할 것이다. 또한, 일부의 경우에서, IoT 디바이스는 IoT 디바이스에 의해 수행되고 있는 기능을 고려할 때 합당한 것보다 더 많은 데이터를 송신하고 있을 수 있다. 예를 들어, IoT 디바이스의 소프트웨어/하드웨어가 오작동할 수 있거나 IoT 디바이스가 해킹될 수 있고, 이로 인해 IoT 디바이스가 불필요한 데이터를 IoT 서비스로 계속해서 전송하게 된다.

[0228] 본 발명의 일 실시예는 지정된 데이터 임계치가 특정 IoT 디바이스에 의해 도달되었을 때 IoT 허브에서 흐름 제어를 수행하여 데이터 트래픽을 효과적으로 무시함으로써 이러한 문제를 해결한다. 일 실시예에서, 각 IoT 디바이스에는, 해당 IoT 디바이스에게 전송이 허용된 기간에 걸친 데이터 양을 표시하는 흐름 제어 파라미터들의 지정된 세트가 구성된다. 흐름 제어 파라미터들은 IoT 디바이스의 타입에 기반할 수 있다. 예를 들어, 도어 록 및 온도 조절기와 같은 소정 IoT 디바이스는 통상적으로는 짧은 데이터 패킷만을 주기적으로 전송할 것이고, 비디오 카메라와 같은 다른 IoT 디바이스는 상당히 더 많은 양의 데이터를 가능하기로는 비주기적으로 전송할 수 있다. 따라서, 흐름 제어 파라미터들은 문제의 IoT 디바이스의 예상 동작에 기반하여 충분한 양의 대역폭을 제공하도록 설정될 수 있다. 일 실시예에서, 각 IoT 디바이스는 해당 IoT 디바이스의 데이터 요구에 기반하여 특정 흐름 제어 "클래스"에 할당된다.

[0229] 그러한 실시예가 **도 30**에 예시되어 있는데, 이 도면은 상이한 세트의 흐름 제어 파라미터들(3015, 3031, 3041)로 각각 구성된 보안 무선 통신 모듈(2810, 3030, 3040)을 갖는 복수의 IoT 디바이스(101 내지 103)를 도시한다. 일 실시예에서, 흐름 제어 파라미터들은 각각의 IoT 디바이스가 지정된 기간에 걸쳐 송신할 것으로 예상되는 데이터의 양 및/또는 빈도를 지정한다(예를 들어, 0.25 Mbyte/시간, 50 Mbyte/시간, 100 Mbyte/일, 10회 통신 시도/일 등). 일 실시예에서, 흐름 제어 파라미터들(3015, 3031, 3041)은, 예시된 바와 같이, IoT 디바이스 데이터베이스(2851) 내의 디바이스별 흐름 제어 파라미터들(3020)의 세트를 관리하는 디바이스 관리 모듈(3021)을 포함하는 IoT 서비스(120)에 의해 지정될 수 있다. 예를 들어, 각각의 IoT 디바이스에 대한 데이터 전송 요구가 결정되면, 디바이스별 흐름 제어 파라미터들(3020)은 이러한 요구를 반영하도록 업데이트될 수 있다.

[0230] 언급된 바와 같이, 일 실시예에서, 디바이스 데이터베이스(2851)는 복수의 상이한 흐름 제어 "클래스"(예를 들어, 시청각 디바이스, 온도 디바이스, 제어 디바이스, 보안 디바이스 등)에 대한 데이터 전송 요구를 포함한다. 새로운 IoT 디바이스가 시스템에 도입될 때, 그 IoT 디바이스는 그 IoT 디바이스의 요구 및/또는 IoT 디바이스의 타입에 기반하여 특정 흐름 제어 클래스와 관련된다.

[0231] 디바이스별 흐름 제어 파라미터들(3020)은 로컬 데이터베이스 내에 디바이스별 흐름 제어 파라미터들(3010)의 사본을 저장하는 흐름 제어 관리 로직(2811)을 포함하는 IoT 허브들(110)에 분배될 수 있다. 일 실시예에서, 흐름 제어 관리(2811)는 각각의 IoT 디바이스(101 내지 103)로부터 수신되고/되거나 그로 송신되는 데이터 트래픽의 양을 모니터링할 수 있다. 데이터 트래픽의 양이 (디바이스별 흐름 제어 파라미터들(3010)에 의해 표시된 바와 같은) 지정된 임계치에 도달하면, IoT 허브(110)는 IoT 디바이스에게 일정 기간 동안 전송을 중단하도록 명령할 수 있고/있거나 단순히 IoT 디바이스로부터 트래픽을 차단할 수 있다.

[0232] 특정 IoT 디바이스가 지정된 임계치보다 높은 레벨에서 전송/수신하고 있으면, 이는 IoT 디바이스가 오작동 중이라는 것을 나타낼 수 있다. 그렇기 때문에, 일 실시예에서, IoT 서비스(120)는 IoT 디바이스를 리셋하라는 커맨드를 송신할 수 있다. 디바이스가 여전히 임계치보다 높은 레벨에서 통신하고 있는 경우, IoT 서비스(120)는 IoT 디바이스에 패치와 같은 소프트웨어 업데이트를 송신할 수 있다. 업데이트된 소프트웨어가 설치되면, IoT 디바이스가 리셋되고 새로운 소프트웨어로 초기화된다. 또한, IoT 디바이스가 오작동하는 것을 사용자에게 알리기 위한 통지가 IoT 서비스로부터 사용자 디바이스로 전송될 수 있다.

[0233] 일 실시예에서, IoT 허브(110)는 데이터 통신 임계치에 도달했다는 사실에도 불구하고 소정 타입의 데이터 트래픽은 허용할 수 있다. 예를 들어, 일 실시예에서, IoT 허브(110)는 소정의 IoT 디바이스가 그의 임계치에 도달했더라도 소정 타입의 "높은 우선순위" 통지는 허용할 것이다. 예로서, IoT 디바이스가 도어 록 또는 도어 엔

트리 검출기인 경우, 소정 조건 하에서(예를 들어, 집이 모니터링되고 있을 때), IoT 허브(110)는 해당 IoT 디바이스가 사용되고 있는 도어를 누군가가 열었음을 표시하는 데이터를 통과시킬 수 있다. 유사하게, IoT 디바이스가 열 및/또는 연기 검출기인 경우, IoT 허브(110)는 (예를 들어, 온도가 임계 값에 도달했기 때문에) 알람 조건을 표시하는 데이터를 통과시킬 수 있다. (예를 들어, 잠재적으로 위험한 조건을 표현하는 것과 같은) 다양한 다른 타입의 "높은 우선순위" 통지가 현재의 흐름 제어 상태에 무관하게 IoT 허브(110)에 의해 통과될 수 있다. 일 실시예에서, 이들 "높은 우선순위" 통지는 후술되는 바와 같이 상이한 속성들을 사용하여 식별된다.

[0234] 본 발명의 일 실시예에 따른 방법이 **도 31**에 도시된다. 본 방법은 위에 기술된 시스템 아키텍처의 맥락 내에서 구현될 수 있지만, 임의의 특정 시스템 아키텍처로 제한되지 않는다.

[0235] 3101에서, 각각의 IoT 디바이스에 대해 흐름 제어 파라미터들이 지정된다. 일 실시예에서, IoT 디바이스는 그와 관련된 흐름 제어 파라미터들의 지정된 세트를 갖는 특정 IoT 디바이스 "클래스"에 할당될 수 있다. 3102에서, 흐름 제어 파라미터들은 IoT 시스템 내의 IoT 허브에 저장된다. 일 실시예에서, 각각의 허브는 IoT 디바이스 파라미터들(예를 들어, 로컬에서 프로비저닝된 IoT 디바이스들에 대한 그 파라미터들만) 모두의 서브세트를 저장할 수 있다.

[0236] IoT 허브가 3103에서, 특정 IoT 디바이스가 결정된 지정된 흐름 제어 파라미터들 밖에서 동작하고 있음을 검출하면, 3104에서, IoT 허브는 해당 IoT 디바이스와의 추가적인 통신을 일시적으로 억제(예를 들어, IoT 디바이스와 IoT 서비스 사이의 통신을 차단)할 것이다. 또한, 언급한 바와 같이, IoT 서비스 및/또는 IoT 허브는 IoT 디바이스를 재부팅하고/하거나 IoT 디바이스에 소프트웨어 업데이트를 설치함으로써 문제를 해결하기 위한 조치를 취할 수 있다.

[0237] **속성 클래스들을 사용하여 사물 인터넷(IoT) 디바이스들 및 트래픽을 관리하는 시스템 및 방법**

[0238] 상이한 IoT 디바이스들이 주어진 위치에서 상이한 기능들을 수행하는 데 사용될 수 있다. 예를 들어, 소정 IoT 디바이스가 온도 및 상태(예를 들어, 온/오프 상태)와 같은 데이터를 수집하고, 이 데이터가 최종 사용자에 의해 액세스되고/되거나 다양한 타입의 경고 조건들을 생성하는 데 사용될 수 있는 IoT 서비스에 이 데이터를 다시 보고하는 데 사용될 수 있다. 이 구현을 가능하게 하기 위해, 본 발명의 일 실시예는 수집된 데이터, 시스템 데이터, 및 다른 형태의 데이터를 상이한 타입의 속성 클래스들을 사용하여 관리한다.

[0239] **도 32**는 직렬 주변기기 인터페이스(SPI) 버스와 같은 직렬 인터페이스(3216)를 통해 마이크로컨트롤러 유닛(MCU)(3215)과 통신하는 보안 무선 통신 모듈(3218)을 포함하는 IoT 디바이스의 일 실시예를 예시한다. 보안 무선 통신 모듈(3218)은 전술한 기술들을 사용하여 IoT 서비스(120)와의 보안 통신을 관리하고, MCU(3215)는 IoT 디바이스(101)의 애플리케이션별 기능을 수행하기 위해 프로그램 코드를 실행한다.

[0240] 일 실시예에서, IoT 디바이스에 의해 수집된 데이터 및 IoT 디바이스에 관련된 시스템 구성을 관리하는 데 다양한 상이한 속성 클래스들이 사용된다. 특히, **도 32**에 도시된 예에서, 속성들은 애플리케이션 속성들(3210), 시스템 속성들(3211), 및 우선순위 통지 속성들(3212)을 포함한다. 일 실시예에서, 애플리케이션 속성들(3210)은 IoT 디바이스(101)에 의해 수행되는 애플리케이션별 기능과 관련된 속성들을 포함한다. 예를 들어, IoT 디바이스가 보안 센서를 포함하는 경우, 애플리케이션 속성들(3210)은 도어 또는 윈도우가 개방되었는지 여부를 지시하는 2진 값을 포함할 수 있다. IoT 디바이스가 온도 센서를 포함하면, 애플리케이션 속성들(3210)은 현재 온도를 지시하는 값을 포함할 수 있다. 사실상 무제한 수의 다른 애플리케이션별 속성들이 정의될 수 있다. 일 실시예에서, MCU(3215)는 애플리케이션별 프로그램 코드를 실행하고, 애플리케이션별 속성들(3210)로의 액세스만 제공받는다. 예를 들어, 애플리케이션 개발자는 보안 무선 통신 모듈(3218)을 갖는 IoT 디바이스(101)를 구입하여 MCU(3215)에 의해 실행될 애플리케이션 프로그램 코드를 설계할 수 있다. 결과적으로, 애플리케이션 개발자는 애플리케이션 속성들에는 액세스할 필요가 있겠지만 아래에 설명되는 다른 타입의 속성들에는 액세스할 필요가 없을 것이다.

[0241] 일 실시예에서, 시스템 속성들(3211)은 IoT 디바이스(101) 및 IoT 시스템에 대한 동작 및 구성 속성들을 정의하는 데 사용된다. 예를 들어, 시스템 속성들은 네트워크 구성 설정(예를 들어, 위에서 논의된 흐름 제어 파라미터들과 같은), 디바이스 ID, 소프트웨어 버전, 광고 간격 선택, 보안 구현 특징부(위에 설명된 바와 같은 것), 및 IoT 디바이스(101)가 IoT 서비스와 안전하게 통신할 수 있게 하는 데 필요한 다양한 다른 저수준 변수를 포함할 수 있다.

[0242] 일 실시예에서, 우선순위 통지 속성들(3212)의 세트가 이들 속성들과 관련된 중요성 또는 심각성의 수준에 기반하여 정의된다. 예를 들어, 특정 속성이 임계치에 도달하는 온도 값과 같은 위험한 조건(예를 들어, 사용자가

실수로 스토브를 켜진 채로 나둔 경우 또는 사용자의 집 안의 열 센서가 트리거될 때)과 관련된 경우, 이 속성은 우선순위 통지 속성 클래스에 할당될 수 있다. 전술한 바와 같이, 우선순위 통지 속성들은 다른 속성들과는 상이하게 처리될 수 있다. 예를 들어, 특정 우선순위 통지 속성이 임계치에 도달할 때, IoT 허브는 IoT 허브에 의해 구현되고 있는 현재의 흐름 제어 메커니즘에 관계없이, IoT 서비스에 속성의 값을 전달할 수 있다. 일 실시예에서, 우선순위 통지 속성들은 또한 IoT 서비스를 트리거하여 (예를 들어, 사용자에게 잠재적으로 위험한 조건을 경고하기 위해) 사용자의 집 또는 사업장 내의 알람 조건들 및/또는 사용자에 대한 통지를 생성할 수 있다.

[0243] **도 32**에 예시된 바와 같이, 일 실시예에서, 애플리케이션 속성들(3210), 시스템 속성들(3211), 및 우선순위 통지 속성들(3212)의 현재 상태는 IoT 서비스(120) 상의 디바이스 데이터베이스(2851) 내에 복제/미러링된다. 예를 들어, 속성들 중 한 속성의 변경이 IoT 디바이스(101)에서 업데이트될 때, 보안 무선 통신 모듈(3218)은 이 변경을 IoT 서비스(120) 상의 디바이스 관리 로직(3021)으로 전달하고, 디바이스 관리 로직은 디바이스 데이터베이스(2851) 내의 속성의 값을 응답적으로 업데이트한다. 또한, 사용자가 IoT 서비스 상에서 속성들 중 한 속성을 업데이트(예를 들어, 원하는 온도와 같은 현재 상태 또는 조건을 조정)하는 경우, 속성 변경은 디바이스 관리 로직(3021)으로부터 보안 무선 통신 모듈(3218)로 전송될 것이고, 무선 통신 모듈 이어서 자신의 속성의 로컬 사본을 업데이트할 것이다. 이러한 방식으로, 속성들은 IoT 디바이스(101)와 IoT 서비스(120) 사이에서 일관되게 유지된다. 또한, 속성들에는 IoT 앱 또는 애플리케이션이 설치된 사용자 디바이스를 통해 그리고/또는 하나 이상의 외부 서비스(3270)에 의해 IoT 서비스(120)로부터 액세스될 수 있다. 언급한 바와 같이, IoT 서비스(120)는 다양한 상이한 속성 클래스들로의 액세스를 제공하기 위해 애플리케이션 프로그래밍 인터페이스(API)를 노출시킬 수 있다.

[0244] 또한, 일 실시예에서, 우선순위 통지 처리 로직(3022)은 우선순위 통지 속성(3212)에 관련된 통지의 수신에 응답하여 규칙 기반 동작을 수행할 수 있다. 예를 들어, 우선순위 통지 속성이 위험한 상태(예를 들어, 다리미 또는 스토브가 사용자에 의해 켜진 채로 있는 것과 같은 상태)를 표시하면, 우선순위 통지 처리 로직(3022)은 위험한 디바이스를 끄려고 시도하는 규칙들의 세트를 구현(예를 들어, 가능한 경우 디바이스에 "끔" 커맨드를 전송)할 수 있다. 일 실시예에서, 우선순위 통지 처리 로직(3022)은 (예를 들어, 위험한 디바이스가 "켜짐" 상태에 있는 때에 사용자가 집을 떠난 것으로 검출되는 경우) 위험한 디바이스를 끌지 여부를 결정하기 위해 사용자의 현재 위치와 같은 다른 관련 데이터를 활용할 수 있다. 또한, 우선순위 통지 처리 로직(3022)은 사용자의 클라이언트 디바이스에 경고 상태를 전송하여 사용자에게 해당 상태를 통지할 수 있다. 잠재적으로 위험하거나 또는 이와 다른 바람직하지 않은 상태를 해결하려고 시도하기 위해 우선순위 통지 처리 로직(3022)에 의해 다양한 다른 타입의 규칙 세트들이 구현될 수 있다.

[0245] **도 32**에는 BTLE 속성들(3205)의 세트 및 속성 어드레스 디코더(3207)도 도시되어 있다. 일 실시예에서, BTLE 속성들(3205)은 **도 19 및 도 20**과 관련하여 전술한 바와 같이 판독 및 기입 포트를 확립하는 데 사용될 수 있다. 속성 어드레스 디코더(3207)는 각각의 속성과 관련된 고유 ID 코드를 판독하여 어느 속성이 수신/송신되고 있는지를 결정하고 그에 맞춰 속성을 처리한다(예를 들어, 속성이 보안 무선 통신 모듈(3218) 내에 저장된 곳을 식별한다).

[0246] **익명 IoT 기기 계정 및 IoT 기기 이체/대여 장치 및 방법**

[0247] 사용자 계정 생성은 IoT 시스템의 신규 가입자 확보에 상당한 장벽이 된다. 예를 들어, 잠재적 가입자는 상당한 양의 개인 및/또는 금융 정보(예를 들어, 전화번호, 신용카드 데이터, 페이팔 계정 정보 등)가 요구되는 경우에는 온라인 IoT 서비스에 대한 가입을 연기 또는 거절하는 쪽을 선택할 수 있다.

[0248] 본 발명의 일 실시예는 계정 생성을 자동화함으로써 이러한 한계를 해결한다. 특히, 일 실시예는 IoT 디바이스가 IoT 서비스로 초기에 프로비저닝될 때 익명 사용자 계정이 생성되는 아키텍처 및 프로세스를 구현한다. 익명 사용자 계정은 사용자가 IoT 디바이스와 연관된 QR 코드를 처음으로 스캔할 때 자동으로 생성될 수 있다. 후속해서, IoT 서비스는 사용자가 IoT 구현을 시도할 기회를 가진 후에 비익명의 사용자 계정이 비침입적 방식으로 개설되도록 하기 위해 사용자로부터의 추가 정보를 요청할 수 있다.

[0249] 따라서, 일 실시예에서, 자동 계정 생성은 사용자로 하여금 그 사용자와 연계된 어떠한 정보도 없이 QR 코드를 스캐닝한 후 수초 이내에 IoT 디바이스를 점검할 수 있게 한다. IoT 디바이스는 (예를 들어, 전술한 보안/암호화 기술들을 사용하여) IoT 서비스가 계정에 대해 생성하는 인증된 보안 세션과 느슨하게 연결된다. 일 실시예에서, 사용자가 클라이언트 디바이스 상의 IoT 앱을 닫거나 상기 세션이 사용자로의 임의의 링크가 이루어지기 전에 만료되면, 계정은 고아 계정이 된다. 고아 계정과 연관된 IoT 디바이스는 IoT 디바이스가 클라이언트 디

바이스 상의 IoT 앱을 통해 다시 연관될 때에는 새로운 계정에 링크될 수 있다. 새로운 사용자로의 IoT 디바이스 전송은 각각의 새로운 사용자에게 새로운 디바이스 경험을 제공하기 위해 임의의 변경들을 소거할 수 있다.

[0250] **도 33**은 본 발명의 실시예들이 구현될 수 있는 아키텍처를 예시한다. 동작 시, 모바일 디바이스(135)의 사용자는 하나 이상의 IoT 디바이스들(101-102)의 관련 ID들(2812A, 2812B)을 캡처하기 위해 모바일 디바이스(135)의 카메라를 사용하기 위한 카메라 인터페이스를 제공하는 IoT 앱(또는 브라우저 실행 가능 코드)을 개방한다. 언급된 바와 같이, 관련 ID들(2812A, 2812B)은 QR 코드들, 바코드들, 또는 IoT 디바이스들(101, 102) 상에 디스플레이된 임의의 다른 형태의 광학 코딩으로서 인코딩될 수 있다. 대안적으로, 일 실시예에서, 관련 ID들은 NFC(Near Field Communication) 또는 BTLE와 같은 무선 프로토콜을 통해 전달될 수 있다. **도 28**에 관하여 이미 설명한 바와 같이, 관련 ID(2812)는 각 관련 ID와 디바이스 ID들 사이의 링크를 포함하는 디바이스 데이터베이스(2851)에서 룩업을 수행하는 IoT 서비스(120) 상의 디바이스 프로비저닝 모듈(2850)로 전송될 수 있다. 디바이스 프로비저닝 모듈(2850)은 관련 ID(2812)를 사용하여 디바이스 ID(2811)를 식별하고, 이어서 디바이스 ID를 사용하여 시스템에 새로운 IoT 디바이스(101)를 프로비저닝하고 IoT 서비스(120)와의 보안 연결을 확립한다. 추가적인 세부사항은 위의 **도 28 및 도 29**에 대한 설명을 참조한다.

[0251] **도 33**으로 돌아가서, 본 발명의 일 실시예는 사용자 계정을 관리하기 위한 계정 관리 모듈(3350) 및 IoT 디바이스를 관리하기 위한 디바이스 관리 모듈(3315)을 포함한다. 예시된 실시예에서, 계정 관리 모듈(3350)은 익명 계정들을 생성하고 그 익명 계정들을 본 명세서에 설명된 바와 같은 하나 이상의 IoT 디바이스와 연관시키기 위한 익명 계정 생성 로직(3311)을 포함한다. 일단 익명 계정 생성 로직(3311)이 익명 계정을 생성하면, 해당 계정의 사용자는 비익명 계정을 갖는 다른 사용자들과 동일한 방식으로(익명 계정에는 아래에 설명된 바와 같이 특정 제한이 있을 수 있겠지만) IoT 디바이스들로부터의 데이터를 제어 및 수신할 수 있다.

[0252] 어떤 후속 시점에서(예를 들어, 사용자가 IoT 디바이스들을 테스트할 기회를 제공받은 후), 사용자 디바이스(135) 상의 IoT 앱은 제한된 양의 개인 정보(예를 들어, 이메일 주소, 전화번호 등)를 제공함으로써 계정을 개설할 옵션을 사용자에게 제시할 수 있다. 일단 사용자가 그 정보를 입력하면, 계정 변환 로직(3312)은 데이터베이스를 비익명 계정으로 변환하고 사용자 데이터를 데이터베이스에 추가한다. 예를 들어, **도 33**에서, 데이터베이스(3351)는 초기에 익명 계정과 연관된 4개의 상이한 IoT 디바이스들에 대응하는 4개의 엔트리를 포함한다.

[0253] 일단 사용자가 몇몇 지정된 양의 개인 정보를 입력하면, 계정 변환 로직(3312)은 **도 34**에 도시된 바와 같이 이들 엔트리들을 비익명 계정 엔트리들로 변환하고, 해당 계정과 관련된 사용자 데이터를 데이터베이스(3351)에 저장한다. 데이터베이스(3351)는 설명의 간략화를 위해 테이블로 예시되지만, 실제 데이터베이스 구조는 다수의 상호 연관된 테이블들을 포함할 수 있고/있거나 다른 유형의 데이터 저장 구조들을 사용할 수 있다. 본 발명의 기본 원리들은 어떤 특정 데이터베이스 또는 데이터 저장 배열로 제한되지 않는다.

[0254] 사용자 계정이 비익명으로 된 후, IoT 서비스/IoT 앱은 필요에 따라 사용자에게 추가 정보를 촉구할 수 있다. 예를 들어, 사용자가 수수료를 요하는 서비스 또는 트랜잭션에 액세스하기로 선택하는 경우, IoT 앱은 사용자에게 금융 계정 정보(예를 들어, 신용 카드, 페이팔 계정 등)를 입력하라는 프롬프트를 표시할 수 있고, 그런 다음에는 그 정보를 데이터베이스(3351)에 안전하게 저장할 수 있다.

[0255] 본 발명의 일 실시예는 제1 사용자가 제2 사용자를 위한 복수의 IoT 디바이스들을 구성한 다음 그 IoT 디바이스들을 IoT 서비스(120) 상의 제2 사용자의 익명 또는 비익명 계정으로 전송할 수 있게 하도록 특별히 맞춤형으로 구성된다. 예를 들어, 제1 사용자는 (예를 들어, 설명된 바와 같이 관련 ID들을 캡처함으로써) 제2 사용자를 위한 복수의 IoT 디바이스들을 구성하기 위해 IoT 앱을 갖는 제1 모바일 디바이스(제1 사용자 자신의 디바이스)를 사용하는 전문 설치자일 수 있다. 모든 IoT 디바이스들이 IoT 서비스(120) 상에 프로비저닝되고 제1 사용자의 계정과 연관되면, 제1 사용자는 IoT 앱을 통해 코드가 생성되게 할 수 있고, 그 후 그 코드는 IoT 디바이스들을 제2 사용자의 계정으로 전송하는 데 사용될 수 있다. 일 실시예에서, IoT 앱을 통해 (예를 들어, IoT 앱 내에서부터 디바이스 전송을 위한 메뉴 항목으로 내비게이팅하여) 제출된 제1 사용자의 요청에 응답하여 IoT 서비스(120)에 의해 QR 코드가 생성되고/되거나 제공된다. 코드를 제2 사용자에게 이메일로 보내거나 또는 문자로 보내는 것과 같은 다른 메커니즘도 코드를 제2 사용자에게 제공하는 데 사용될 수 있다.

[0256] 코드가 어떻게 생성되거나 전달되는지에 관계없이, 일 실시예에서, 코드는 IoT 서비스(120)에서 제2 사용자를 대신하여 제1 사용자에 의해 구성된 복수의 새로운 IoT 디바이스들과 연관된다. 일 실시예에서, IoT 디바이스들의 구성이 완료되면, 제1 사용자는 제1 모바일 디바이스의 스크린 상에 QR 코드를 디스플레이한다. 그러면 제2 사용자는 IoT 앱을 열고 QR 코드를 스캔한다(QR 코드는, 언급된 바와 같이, IoT 서비스(120) 또는 제1 모바일 디바이스에 의해 생성되어 IoT 서비스(120)로 전송될 수 있다).

[0257] 일 실시예에서, 코드를 검증한 후에, IoT 서비스(120) 상의 계정 변환 모듈(3312)은 IoT 디바이스들을 제1 사용자의 계정으로부터 제2 사용자의 계정으로 전달한다. 하나의 특정 구현예에서, 제1 사용자의 계정은 익명이고 제2 사용자의 계정은 (초기에는) 익명이거나 비익명이지만, 본 발명의 기본 원칙은 이러한 구현예에 제한되지 않는다. 예를 들어, 특정 구현예 여하에 따라, 제1 사용자의 계정과 제2 사용자의 계정 모두가 익명 계정, 비익명 계정, 또는 이들의 조합일 수 있다.

[0258] 예시적인 IoT 앱을 위한 방법 및 사용자 인터페이스의 일 실시예가 **도 35**와 관련하여 기술될 것이다. 이 예에서는 사용자가 (예를 들어, 전술한 바와 같이) IoT 서비스에 안전하게 접속되도록 설계된 새로운 IoT 디바이스를 갖고 있지만 IoT 서비스에 대한 계정은 아직 개설하지 않은 것으로 가정한다.

[0259] 3501에서, 사용자는 IoT 디바이스로부터 QR 코드를 스캔하기 위한 명령들을 디스플레이하는 모바일 디바이스 상의 IoT 앱을 개시한다. 사용자에게 IoT 허브가 설치되어 있지 않으면, IoT 앱은 실행될 때 IoT 허브의 기능을 수행한다(예를 들어, 전술한 바와 같이 IoT 디바이스를 IoT 서비스에 안전하게 접속한다). 모바일 디바이스 OS는 3502에서 사용자에게 IoT 앱에 의한 카메라의 사용을 인가하라는 프롬프트를 표시한다. 3503에서, 사용자는 IoT 디바이스 상의 QR 코드의 이미지를 캡처한다. 사용자가 QR 코드에 초점을 맞추는 데 도움이 되도록 GUI 내에 정사각형 캡처 영역의 윤곽을 나타낸다. 3504에서, 사용자는 IoT 디바이스를 제어(예를 들어, 본 실시예에서는 ON/OFF)하기 위한 그래픽 제어들의 세트를 제공받고, 또한 (IoT 디바이스의 유형 및 구현예에 따라) IoT 디바이스에 의해 수집되고 IoT 서비스를 통해 안전하게 전송되는 데이터도 제공받을 수 있다. 따라서, 이 실시예에서, 사용자는 IoT 서비스에 대한 계정을 개설하고 임의의 개인 정보를 입력하기 전에 IoT 디바이스로의 안전한 양방향 액세스를 획득한다. 일 실시예에서, 이는 사용자가 QR 코드를 스캔할 때 익명 계정을 생성하고 IoT 디바이스 ID/관련 ID를 익명 계정과 연관시키는 익명 계정 생성 모듈(3311)에 의해 달성된다.

[0260] 3504에서 "+" 아이콘을 선택하면, 사용자는 3505에 나타낸 것과 같은 다양한 옵션들 - 이들 중 일부는 3506에서 사용자에게 개인 정보를 제출할 것을 요구할 수 있음 - 을 포함하는 메뉴를 제시받는다. 예를 들어, 사용자는 이메일 주소 또는 전화번호를 입력할 수 있다. 이에 응답하여, 계정 변환 로직(3312)은 **도 34**와 관련하여 전술한 바와 같이 익명 계정을 비익명 계정으로 변환할 수 있다.

[0261] 제1 사용자가 제2 사용자를 대신하여 IoT 디바이스들을 구성하는 방법 및 예시적인 사용자 인터페이스가 **도 36a 및 도 36b**와 관련하여 설명될 것이다. 먼저 **도 36a**를 참조하면, 3601에서, 제1 사용자는 제1 IoT 디바이스를 스캔하고, 그 결과 3602에 나타낸 제어 그래픽들이 생성된다. 제1 사용자는 3603에서 메뉴를 열고, 3604에서 다른 IoT 디바이스를 추가하는 선택을 한다. 제2 디바이스를 스캐닝할 때, 2 세트의 제어 그래픽 세트(각 디바이스마다 하나씩)가 3605에 나타내어진다. 화살표에 나타낸 바와 같이, 제1 사용자는 메뉴로 되돌아가서 디바이스들을 추가하는 것을 계속할 수 있다. 각 디바이스가 추가됨에 따라, IoT 서비스(120)는 전술한 바와 같이 업데이트된다. 예를 들어, 제1 사용자가 비익명 계정을 갖는 경우, IoT 디바이스들 각각에 대한 데이터가 데이터베이스(3351) 내에 저장되고 제1 사용자의 계정과 연관된다. 익명 계정이 사용되고 있는 경우, 제1 사용자는 개인 정보(예를 들어, 이메일 주소 또는 전화번호)를 입력함으로써 비익명 계정으로 전환할 수 있다.

[0262] **도 36b**로 돌아가서, 모든 IoT 디바이스가 구성된 경우, 제1 사용자는 3606에서 메뉴에서 "전송" 옵션을 선택하여 IoT 디바이스를 제2 사용자의 계정으로 전송할 수 있다. 예시된 바와 같이, 다양한 옵션들이 전송을 위해 이용 가능할 수 있다. 일 실시예(옵션 1)에서는 QR 코드가 전술한 바와 같이 제1 사용자의 모바일 디바이스에 디스플레이된다. QR 코드는 제1 사용자가 "전송" 메뉴 항목을 선택하는 것에 응답하여 IoT 서비스 상의 계정 변환 로직(3312)에 의해 생성될 수 있다. QR 코드가 생성되면, 그 코드는 데이터베이스(3351) 내에서, 제1 사용자에 의해 구성된 IoT 디바이스들의 세트와 연관될 수 있다. 3608에서, 제1 사용자는 IoT 앱을 사용하여 그 QR 코드를 스캔할 수 있고, 이어서 IoT 앱은 그 코드를 IoT 서비스로 전송할 것이다. 이어서 계정 변환 로직(3312)은 그 코드를 확인하게 되면 그 코드 및 제1 사용자의 계정과 연관된 IoT 디바이스들을 제2 사용자의 계정으로 전달하고, 그 결과 3609에 나타낸 제어 그래픽들(이 실시예에서는 2개의 IoT 디바이스에 대해 2 세트의 제어 그래픽 세트)이 생성된다.

[0263] 대안적으로 또는 추가적으로, 3610에서, 제1 사용자는 제2 사용자의 이메일 주소 또는 계정 번호와 같은 제2 사용자와 연관된 식별 정보를 입력하거나, 제2 사용자에게 QR 코드를 (예를 들어, 이메일 또는 문자를 통해) 수동으로 보낼 수 있다. 제2 사용자가 IoT 앱을 통해 IoT 서비스에 액세스하여 필요한 정보를 제공할 때, 3611에서, 제2 사용자에게는 IoT 디바이스들이 제2 사용자의 계정으로 전송될 때와 동일한 제어 그래픽 세트가 제공된다.

[0264] **도 37a 및 도 37b**는 사용자 A(3701) 및 사용자 B(3702)가 계정 관리 모듈(3350) 및 디바이스 관리 모듈(3315)

과의 통신을 통해 IoT 디바이스 관리 동작들을 구현하는 본 발명의 상이한 실시예들을 예시하는 트랜잭션 선도이다. 각 사용자는 예시된 트랜잭션들을 수행하는 컴퓨팅 디바이스에 설치된 IoT 앱을 갖는 것으로 가정된다.

[0265] **도 37a 및 도 37b**의 실시예에서, 사용자 A(3701)는 사용자 B(3702)를 대신하여 디바이스에 대한 구성을 추가하고, 구성하고, 지속시킨다. 이 실시예에서, 사용자 A는 익명 사용자 계정으로 작업을 수행하지만 다수의 디바이스들을 구성하고 지속시키는 능력은 여전히 갖는다. 본 명세서에서 사용되는 바와 같이, 구성을 지속시킨다는 것은 디바이스들의 구성이 디바이스 전송을 존속시킨다는 것을 의미한다. 예를 들어, 전문 설치자(사용자 A)가 최종 사용자의 디바이스 계정(사용자 B)의 계정 사용자 이름과 암호를 공유하지 않고도 IoT 온도 조절기를 설치할 수 있으며 IoT 온도 조절기에 Wifi 자격 증명 및 일정을 구성할 수 있다. 최종 사용자가 사전 구성된 IoT 디바이스를 시작하려면 디바이스를 연결하기만 하면 된다. 다른 실시예에서, 사용자 A는 비익명 사용자 계정으로 IoT 디바이스들을 구성하는 작업을 수행한 다음에 그 구성을 사용자 B에게 전달할 수 있다.

[0266] **도 37a**에서, 사용자 A(3701)는 초기에 IoT 앱을 사용하여 (예를 들어, 전술한 바와 같이 QR 코드를 캡처함으로써) 새로운 IoT 디바이스를 추가한다. "디바이스 추가(add device)" 커맨드는 사용자 A에 대한 익명 계정을 생성하고 "디바이스 추가"에 대한 커맨드를 디바이스 관리 모듈(3315)로 전송하는 계정 관리 모듈(3350)에 의해 수신되고, 디바이스 관리 모듈은 새로운 IoT 디바이스를 데이터베이스에 추가하고 계정 관리 모듈(3350)에 통지한다. 언급된 바와 같이, 일 실시예에서, 이는 디바이스 ID를 식별하기 위해 관련 ID를 사용함으로써 그리고 디바이스 ID를 계정 관리 모듈(3350) 및/또는 사용자 A(3701)에 안전하게 전송함으로써 달성된다.

[0267] 그런 다음, 사용자 A는 IoT 디바이스의 구성을 수행하고, 디바이스 관리 모듈(3315)은 그 구성 데이터를 설정하고 사용자 A(3701)에게 통지한다. 구성이 완료된 때, 사용자 A는 디바이스 관리 모듈(3315)로 하여금 구성된 IoT 디바이스들 각각에 대한 구성을 지속시키도록 하는 지속 명령을 (예를 들어, IoT 앱 사용자 인터페이스를 통해) 생성한다. 사용자 A의 분리된 익명 계정은 후속하여 삭제될 수 있지만, IoT 디바이스 구성들은 디바이스 관리 모듈(3315)에 의해 지속된다.

[0268] 언급된 바와 같이, 이러한 IoT 디바이스 구성들은, 사용자 B의 IoT 앱에 의해 입력/캡처된 때에 사용자 B에게 구성된 IoT 디바이스들을 전송할, 데이터베이스 내의 코드와 연관될 수 있다. 개별 IoT 디바이스의 경우, 코드는 디바이스 ID/관련 ID일 수 있다. 따라서, **도 37b**에 도시된 바와 같이, 사용자 B(3702)는 계정 관리 모듈(3350)로 전송되는 IoT 앱을 통해 "디바이스 추가" 커맨드를 생성한다. 이 커맨드는 IoT 디바이스를 식별하고, 이에 응하여, 계정 관리 모듈(3350)은 사용자 B(3702)에 대한 익명 계정을 생성한다. 디바이스 관리 모듈(3315)은 커맨드를 수신하면, 지속된 디바이스 구성들을 사용자 B의 계정으로 전송하고, 통지를 계정 관리 모듈(3350) 및 사용자 B의 IoT 앱으로 전송한다.

[0269] 일 실시예에서, 계정이 이메일 또는 전화번호에 링크되면(즉, 비익명 계정이 되면), 해당 계정과 연관된 디바이스들의 더 이상의 전송은 사용자의 동의 없이는 허용되지 않는다. 따라서, 사용자 B(3702)가 전화번호를 사용하여 통지를 전송한 때, 계정 관리 모듈(3350)은 그 계정과 연관된 IoT 디바이스들을 안전하게 하기 위해 그 계정을 비익명 계정으로 변환하고 디바이스 관리 모듈(3315)에 통지를 전송한다. 디바이스 관리 모듈(3315)은 디바이스들을 잠그고 계정 관리 모듈(3350)에 통지한다.

[0270] **도 38a 및 도 38b**는 사용자 A(3701)가 익명 계정에 추가된 IoT 디바이스를 추가하는(예를 들어, IoT 디바이스의 QR 코드를 스캔하는) 예를 예시한다. 후속하여, 사용자 B(3702)는 동일한 IoT 디바이스를 추가한다. IoT 디바이스가 익명 계정과 연관되기 때문에, 디바이스 관리 모듈(3315)은 사용자 B에 대해 생성된 새로운 계정으로 디바이스를 전송한다. 그런 다음에 사용자 B(3702)가 이메일 주소를 익명 계정과 연관시킨다. 계정 관리 모듈(3350)은 사용자 B에게 비익명 계정이 생성되었음을 통지하고, 이에 응하여 계정을 비익명 계정으로 변환한다. IoT 장치를 안전하게 하기 위해 디바이스 관리 모듈(3315)에 메시지를 보낸다. 디바이스 관리 모듈(3315)은 IoT 장치를 잠그고 계정 관리 모듈에 통지한다. 후속하여, 계정 관리 모듈(3350)은 사용자 A의 고아 계정을 삭제한다. **도 38b**에 예시된 바와 같이, 일단 디바이스 관리 모듈(3315)이 사용자 B의 계정을 대신하여 디바이스를 잠그면, 디바이스를 추가하기 위한 사용자 A에 의한 후속 시도는 성공하지 못할 것이다.

[0271] 일 실시예에서, 사용자에게 추가적인 IoT 디바이스/서비스 이점을 제공하기 위해, 클라이언트 상의 IoT 앱은 사용자가 자신의 익명 계정을 비익명 연결 계정으로 전환하는 것을 장려할 수 있다. 사용자는 계정을 이메일 또는 전화번호에 연결함으로써, 클라이언트 세션이 만료된 후에도 계정을 복구할 수 있는 능력을 제공받는다. 계정에 전화나 이메일을 추가하면 통지, 다른 사용자와의 공유, 결제 영수증 발송, 및/또는 디바이스 규칙 생성과 같은 혜택을 받는 것도 가능해질 수 있다.

[0272] 일 실시예에서, 전술한 아키텍처는 IoT 앱을 설치하는 사용자들과 일시적으로 연관될 수 있는 공공 사용 IoT 디바이스들을 허용한다. 제한이 아닌 예로서, 공공 사용 IoT 디바이스에는 주차 미터, 자판기, 및/또는 자율 주행차가 포함될 수 있다. 위에서 논의된 바와 같이, 본 발명의 일 실시예는 익명 계정의 생성을 통해 IoT 디바이스의 즉각적인 사용을 제공하고, 또한 전화번호 또는 이메일 주소로 영수증을 전송하기 위한 계정 링크를 제공한다. 이 구현예는 계정 채택 및 데이터 분석에 도움이 될 것이다.

[0273] 일 실시예에서, 디바이스가 전화번호 또는 이메일에 링크될 때, 사용자는 디바이스를 다른 계정들과 "공유"할 수 있다. 예를 들어, **도 35**의 3505에서 GUI 메뉴는 "공유" 옵션을 포함한다. 제1 사용자는 이 옵션을 선택해서 제2 사용자에 대한 식별 정보(예를 들어, 이메일 주소, 전화번호)를 입력하여 제2 사용자와 하나 이상의 IoT 디바이스를 공유할 수 있다. 공유할 때, IoT 디바이스는 양쪽 사용자에 의해 제어될 수 있다.

[0274] 대조적으로, 일 실시예에서, 전술한 바와 같은 자동 계정 생성(즉, 사용자가 사용자 데이터의 수신에 응답하면 익명 계정을 생성)과 조합된 "이전" 옵션(예를 들어, **도 36b**의 3606 참조)은 소유자가 해당 디바이스의 직접적인 제어를 다른 사용자 계정에게 포기하지만 계정 관리 로직(3350)에 의해서는 해당 IoT 디바이스의 소유자로서 유지되는(따라서 대여된 IoT 디바이스를 철회할 수 있는) 디바이스를 쉽게 대여할 수 있게 한다. 일 실시예는 다양한 형태의 "이전들"을 구현한다. 예를 들어, 영구 이전은 IoT 디바이스들을 제 1 사용자의 계정으로부터 제 2 사용자의 계정으로 영구적으로 이전할 수 있다. 일단 이전되면, 제1 사용자는 제2 사용자가 IoT 디바이스들을 다시 제1 사용자에게 이전하지 않는 한 IoT 디바이스들의 제어를 회복하지 않는다. 대조적으로, 아래에서 논의되는 실시예들은 임시 이전(즉, 디바이스 대여)을 구현하고, 여기서 제1 사용자는 IoT 디바이스들의 제어를 특정 기간 동안 제2 사용자에게 이전한다. 일 실시예에서, 제1 사용자는 IoT 디바이스들이 제2 사용자에게 대여된 때에는 그 IoT 디바이스들에 액세스하지 못하게 된다.

[0275] 구체적으로, **도 39**는 제1 사용자(사용자 A)가 하나 이상의 IoT 장치를 제2 사용자(사용자 B)에게 대여하는 일련의 트랜잭션을 예시한다. 이 실시예에서, IoT 디바이스(들)의 소유자인 사용자 A는 디바이스 이전 동작을 인가하는 계정 관리 로직(3350)에 "디바이스 대여" 커맨드를 전송한다. 언급한 바와 같이, "대여"는 일시적인 디바이스 이전으로 정의될 수 있다. 그런 다음, 디바이스 관리 로직(3315)은 IoT 디바이스들을 잠금해제하여, IoT 디바이스(들)가 다른 계정(익명 또는 비익명)과 연관될 수 있게 한다.

[0276] 사용자 B는 후속하여 (예를 들어, QR 코드들을 캡처함으로써) IoT 디바이스(들) 중 하나 이상을 추가한다. 계정 관리 로직(3350)은 그에 응답하여 동작을 인가하고 사용자 B에 대한 계정(사용자 B가 계정을 갖고 있지 않은 경우에는 익명 계정)을 생성한다. 디바이스 관리 로직(3315)은 계정 관리 로직으로부터 인가를 수신하면 디바이스들을 사용자 B의 계정에 추가하고 확인을 생성한다. 그런 다음, 사용자 B는 이전된 IoT 디바이스들 중 임의의 것을 보고 제어할 수 있다. 반면에, 사용자 A는 일시적으로 IoT 장치들을 보거나 제어하지 못하게 된다.

[0277] 따라서, 상기에 설명된 바와 같이 디바이스 대여를 이용함으로써, 디바이스 구성들은 지속될 수 있지만, IoT 디바이스들에 액세스되는 현재 사용자는 IoT 디바이스들을 오로지 임의의 시간에 제어하는 것만 보장받는다. 이러한 실시예는 프라이버시가 우려되는 상황에서 특히 유용할 수 있다. 예를 들어, IoT 디바이스들이 설비된 집이 (예를 들어, 에어비앤비 또는 유사 서비스를 이용하여) 임대될 때, 임차인들은 IoT 디바이스들(예를 들어, 온도 조절기, 보안 디바이스, 비디오 카메라 등)로의 임시 액세스가 필요할 수 있다. 임차인은 그 집에 있는 동안에는 부동산 소유자가 IoT 디바이스들에 액세스를 유지했는지 우려할 수 있다. 본 명세서에 설명된 기술들을 사용함으로써, 부동산 소유자는 보안 디바이스들 및 비디오 카메라들을 포함한 모든 IoT 디바이스들이 임차인에게 일시적으로 이전되었고 이로써 임대 기간 동안에는 부동산 소유자가 액세스하지 못한다는 것을 임차인에게 보증할 수 있다.

[0278] 디바이스 대여는 특정 사용자 계정에 대여하거나 어느 누구에게(예를 들어, 익명 사용자 계정) 대여함으로써 달성될 수 있다. 전술한 실시예들은 디바이스 이전 프로세스를 단순화한다. 부동산 소유자는 어떠한 세입자에게도 어떤 계정 정보도 교환하지 않아도 간단히 대여를 할 수 있다. 그러면 세입자는 추가 명령 없이도 IoT 디바이스들의 제어를 취할 수 있을 것이다.

[0279] 본 명세서에 설명된 익명 계정의 익명성으로 인해, 패스워드의 필요성은 본질적으로 제거된다. 이메일이나 전화번호에 액세스되지 않는 다수의 디바이스에 동시에 로그인해야 할 필요성은 여전히 있다. 이 기능을 추가하기 위해 사용자는 패스워드를 설정하고 이메일이 아닌 사용자 아이디를 입력할 수 있다.

[0280] **기계 학습(ML) 기반 IoT 디바이스 프로비저닝 시스템 및 방법**

[0281] 새로운 IoT 디바이스를 설치하기 위해, 사용자는 일반적으로 IoT 디바이스 앱을 설치 및 실행하는데, 이는 스크

립트에 기반하여 사용자에게 설치 프로세스를 안내한다. 설치 프로세스의 특정 단계에서, IoT 디바이스 앱은 사용자에게 IoT 디바이스의 표면 또는 디스플레이로부터 QR 코드 또는 기타 광학 코드를 캡처하라는 프롬프트를 표시한다. 광학 코드가 캡처되면, 이 식별 정보는 IoT 디바이스 앱 및/또는 IoT 서비스가 IoT 디바이스를 고유하게 식별할 수 있게 하고, (예를 들어, 다양한 실시예들에 대해 위에서 설명된 바와 같이) IoT 디바이스와 안전하게 통신할 수 있게 하고, 그리고/또는 IoT 디바이스를 IoT 서비스에 등록할 수 있게 한다.

[0282] 그러나, 현재의 구현들은 사용자가 광학 코드를 찾을 수 없을 때 불충분한 해결책을 제공한다. 일부 구현에서, 예를 들어, IoT 디바이스 앱은 일련의 드롭-다운 선택 필드들을 통해 또는 모델 번호의 알려진 부분을 입력함으로써 사용자에게 IoT 디바이스 모델을 선택하도록 요청한다. 이는 잠재적으로 많은 수의 IoT 디바이스 모델들 - 이들 중 일부는 외관이 유사할 수 있고/있거나 유사한 기능을 가질 수 있음 - 이 동일한 회사에 의해 제조되는 것을 감안할 때 난제가 될 수 있다.

[0283] 이러한 제한들을 해결하기 위해, 본 발명의 실시예들은 IoT 디바이스 모델을 식별하는 데 객체 인식 기술을 사용한다. 특히, 이 설정 프로세스 단계에서, IoT 디바이스 앱은 사용자에게 IoT 디바이스의 적어도 하나의 이미지를 캡처하도록 요청한다. IoT 디바이스 앱은 이미지(들)를 IoT 서비스에 포워딩하고, IoT 서비스는 사전 훈련된 ML 기반 이미지 인식 엔진을 사용하여 이미지 인식을 수행한다. 그런 다음에 IoT 서비스는 IoT 디바이스 모델 번호를 나타내는 정보를 IoT 디바이스 앱으로 전송하고, IoT 디바이스 앱은 그 정보를 사용하여 설정 프로세스를 완료한다.

[0284] 제한이 아닌 예로서, **도 40**은 3개의 상이한 IoT 디바이스, 즉 벽 스위치(102), 비디오 카메라(102), 및 도어 록(103)을 예시한다. 이 디바이스들 중 하나를 설정할 때 광학 코드가 이용 가능하지 않으면, 사용자 디바이스(135)에서 실행되는 IoT 디바이스 앱은 사용자에게 이러한 특정 IoT 디바이스(101, 102, 또는 103)의 하나 이상의 이미지들을 캡처하라는 프롬프트를 표시한다.

[0285] IoT 디바이스 앱은 이미지 데이터(4010A)를 보안 접속을 통해 IoT 서비스(120) 상의 디바이스 프로비저닝 로직(4000)으로 전송한다. 일 실시예에서, 디바이스 프로비저닝 로직(4000)은 설정 프로세스 동안 IoT 디바이스(101 내지 103)와 IoT 서비스(120) 사이의 인터페이스로서 동작한다. 이미지 사전 처리(4015)는 상이한 유형의 사용자 디바이스들로부터 수신된 이미지 데이터를 정규화하기 위해 수행될 수 있다. 예를 들어, 입력 이미지는 특정 해상도로 스케일링될 수 있고/있거나 이미지 파일은 ML 기반 객체 인식 엔진(4050)에 의한 처리를 위한 특정 특성들(예를 들어, 특정 비트 깊이, 컬러 공간 등)을 갖는 새로운 포맷으로 변환될 수 있다.

[0286] 정규화된 이미지 데이터(4010B)는 (예를 들어, **도 41**과 관련하여 아래에서 설명되는 바와 같이) IoT 서비스(120)를 운영하는 실체에 의해 판매되는 IoT 디바이스들의 이미지들로 훈련된 ML 기반 객체 인식 엔진(4050)에 제공된다. ML 기반 객체 인식 엔진(4050)에 의해 식별된 디바이스 모델(4040)은 디바이스 프로비저닝 로직(4000)에 전달되며, 디바이스 프로비저닝 로직은 그에 응하여 디바이스 모델 및/또는 추가적인 설정 명령들(4041)을 사용자 디바이스 상의 IoT 앱으로 전달한다. 예를 들어, 일단 IoT 디바이스가 식별되면, 사용자는 (IoT 디바이스 앱을 통해) 일련의 버튼을 누르거나 IoT 디바이스를 특정 횟수만큼 켜고 및 꺼서 IoT 디바이스를 특정 설정 모드(이 모드에 있지 않은 경우)로 진입시키도록 하는 요청을 받을 수 있다.

[0287] **도 41**은 훈련 시퀀스의 예를 예시한다. 특히, 훈련 로직(4070)은 이미지들 및 연관된 메타데이터를 포함하는 훈련 데이터(4075)를 사용하여 ML 기반 객체 인식 엔진(4050)을 훈련시킨다. 훈련 로직(4070)은 각 이미지 및 각 이미지 중의 IoT 디바이스를 식별하는 연관된 메타데이터를 제공할 수 있고, ML 기반 객체 인식 엔진(4050)은 그 제공된 것을 각 IoT 디바이스의 객체 특성들을 "학습"하는 데 사용한다. 일부 실시예들에서, 특정한 양의 학습이 달성되면, ML 기반 객체 인식 엔진(4050)은 결과(예를 들어, 식별된 IoT 디바이스들 또는 후보들)를 훈련 모듈(4070)에 표시할 수 있고, 훈련 모듈은 ML 기반 객체 인식 엔진(4050)에 (예를 들어, 정확한 IoT 디바이스가 식별되었는지 여부를 표시하는) 피드백을 제공할 수 있고, 이 피드백은 ML 기반 객체 인식 엔진(4050)이 추가적인 학습을 하는 데 사용된다.

[0288] 새로운 동적으로 획득된 훈련 데이터(4077)가 필드로부터 수집되고 새로운 훈련 데이터(4075)가 IoT 디바이스 제조자에 의해 제공되므로, ML 기반 객체 인식 엔진(4050)은 일단 동작하면 계속 학습할 수 있다. 예를 들어, 사용자가 사용자 디바이스(4090)로부터 IoT 디바이스의 이미지를 캡처하고 IoT 디바이스가 식별되었을 때마다, 이 정보는 훈련 로직(4070)이 ML 기반 객체 인식 엔진(4050)을 추가로 가르치는 데 사용될 수 있다. 또한, 새로운 IoT 디바이스 제품들이 이용 가능해짐에 따라, 이들 제품들과 연관된 훈련 데이터(4075)는 ML 기반 객체 인식 엔진(4050)이 새로운 IoT 디바이스들과 연관된 객체 특성들을 학습할 수 있도록 훈련 로직(4070)에 직접 제공될 수 있다.

[0289] 이러한 방식으로, ML 기반 객체 인식 엔진(4050)은 상이한 이미지들 중의 객체 특성들에 기반하여 상이한 IoT 디바이스들을 구별하는 방법을 학습한다. 누적된 학습은 IoT 서비스(120) 상에 ML 데이터(4055)로 저장된다. 일 구현예에서, 훈련 데이터(4075)는 설치 상태와 미설치 상태 모두(예를 들어, 벽 박스에 설치된 벽 스위치 대 포장에서 제거되었지만 아직 설치되지 않은 동일한 벽 스위치)에서의 각 IoT 디바이스 모델의 이미지들을 포함하고, 따라서 IoT 디바이스들은 상이한 설치 단계들에서 식별될 수 있다.

[0290] **도 40 및 도 41**은 ML 기반 객체 인식 엔진(4050)이 IoT 서비스(120) 상에 있는 구현예를 예시하지만, 일부 실시예들은 ML 기반 객체 인식 컴포넌트들을 개별 사용자 디바이스들(4090) 상에(예를 들어, IoT 디바이스 앱들 내에) 포함할 수도 있다. 예를 들어, ML 기반 객체 인식 엔진(4050)에 의해 생성된 ML 데이터(4055)는 IoT 디바이스 앱들에 포함되고 지속적으로 업데이트될 수 있다. 그런 다음에 IoT 디바이스 앱 내의 객체 인식 엔진은 IoT 서비스(120)와 통신하지 않고도 ML 데이터(4055)를 사용하여 IoT 디바이스들을 로컬 방식으로 인식할 수 있다. 이러한 실시예들은, 예를 들어, IoT 서비스(120)에 대한 네트워크 연결이 이용 불가능한 경우에 유용할 수 있다. 일부 실시예들에서, 이미지 인식의 일부는 (예를 들어, 인식이 비교적 쉽게 행해질 수 있는 경우들에 있어서는) 사용자 디바이스(4090) 상에서 수행되는 반면, 다른 부분은 (예를 들어, 더 집중적인 동작들이 필요할 때에는) IoT 서비스(120) 상의 ML 기반 객체 인식 엔진(4050)에 의해 수행된다.

[0291] 다양한 형태의 ML 기반 객체 인식이 본 발명의 기본 원리들에 따라 수행될 수 있다. 예를 들어, 기계 학습 기술은 표현 학습을 갖는 인공 신경망들을 기반으로 할 수 있다. 이는 컨볼루션 신경망 및 다른 형태의 딥 러닝 배열을 포함할 수 있지만 이에 한정되지는 않는다. 또한, 기계 학습 엔진들은 감독 학습, 준감독 학습, 비감독 학습, 또는 이들의 임의의 조합을 사용하여(예를 들어, 상이한 훈련 단계들에서 상이한 유형들의 기계 학습을 사용하여) 훈련될 수 있다.

[0292] 본 발명의 다양한 실시예에 따른 방법이 **도 42**에 도시된다. 본 방법은 본 명세서에 기술된 아키텍처들의 맥락 내에서 구현될 수 있지만 임의의 특정 IoT 서비스 또는 기계 학습 아키텍처로 제한되지 않는다.

[0293] 4201에서, 사용자는 사용자의 모바일 디바이스에서 IoT 디바이스 앱을 실행함으로써 새로운 IoT 디바이스(또는 이전에 소유된 IoT 디바이스)를 설정하기 위한 프로세스를 개시한다. 4202에서, IoT 디바이스 앱은 사용자가 바코드 또는 QR 코드와 같은 기계 판독 가능 광학 라벨을 스캔하도록 요청받는 설정 프로세스의 단계에 도달한다. 광학 라벨이 이용 가능하면, 사용자는 광학 라벨을 스캔하고, 설정 프로세스는 전술한 다양한 실시예들과 관련하여 설명된 바와 같이 계속된다(예를 들어, IoT 디바이스와 IoT 서비스 사이에 보안 통신 채널을 확립하고 IoT 디바이스를 등록한다).

[0294] 4203에서, 사용자는 광학 라벨이 이용 가능하지 않음을 IoT 앱을 통해 표시하고, 4204에서, IoT 디바이스 앱은 사용자에게 IoT 디바이스의 사진을 캡처하라는 프롬프트를 표시한다. 일 실시예에서, IoT 앱은, OS에 의해 모바일 디바이스 카메라로의 액세스가 제공될 때 사용자에게 이미지를 캡처하기 위해 카메라를 IoT 디바이스로부터 특정 거리만큼 호버링하게 하도록 명령하는 통합형 사진 캡처 컴포넌트를 포함한다.

[0295] 4205에서, 사진 데이터(또는 사진 데이터의 선택 부분들)가 IoT 서비스로 전송되고, 4206에서, IoT 서비스는 ML 기반 객체 인식을 수행하여 IoT 디바이스 모델을 식별한다. 언급한 바와 같이, 사용자는 정확한 검출 결과의 가능성이 지정된 임계치(예를 들어, 80%, 75% 등)를 초과하도록 하기 위해 필요에 따라 추가적인 사진을 찍으라는 지시를 받을 수 있다.

[0296] 일단 식별되면, 4207에서, IoT 서비스는 IoT 디바이스 모델의 표시 및/또는 IoT 디바이스를 특정된 설정 모드에 배치하기 위한 프로세스와 같은 추가 정보를 전송한다(그러나, 후자의 정보는 IoT 디바이스에 저장되고, IoT 디바이스 모델의 식별 시에 액세스될 수 있다). 마지막으로, 4208에서, IoT 디바이스 앱은 IoT 디바이스 모델에 기반하여 설정 프로세스를 계속 실행한다.

[0297] **블루투스 광고 채널을 이용한 IoT 디바이스 식별 및 초기화 시스템 및 방법**

[0298] 전술한 바와 같이, BTLE(Bluetooth Low Energy) 또는 BT 5(이하, "BT") 디바이스와 같은 저전력 블루투스 디바이스들은 광고 패킷을 전송하여 연결 정보를 브로드캐스팅하고 다른 BT 디바이스들과 접속을 확립한다. 본 발명의 실시예들은 모바일 앱에 특정 유형의 주변 디바이스들의 "인식"을 제공하기 위해 BT 광고 패킷들을 사용하여 IoT 디바이스들과 모바일 앱 사이의 상호작용 시스템을 구현한다. 이러한 인식을 통해, 모바일 앱은 사용자에게 디바이스 온보딩 프로세스 동안 IoT 디바이스별 이미지, 비디오, 오디오, 및/또는 명령을 제공한다.

[0299] 다양한 이유로, 최종 사용자들은 온보딩 프로세스 동안 IoT 디바이스들과 물리적으로 상호작용하라는 요구를 받

을 수 있다. 상황은 사용 설명서 및/또는 기타 문서의 분실, 이전 소유자의 계정에 여전히 묶여 있을 수 있는 이전에 소유되었다가 반환된 디바이스(RMA)의 구매, 또는 (일반적으로 디바이스로의 물리적 액세스를 보장하기 위해) 온보딩 전에 "설정 모드"에 놓여 있어야 하는 IoT 디바이스의 구매를 포함할 수 있지만 이에 한정되지는 않는다. 실제로, 이는 일반적으로는 다수의 버튼들을 함께 누르는 형태, 단일 버튼들을 몇 초 동안 누르고 있는 형태, 전원을 여러 번 켜고 끄는 형태, 또는 설정 프로세스로 들어가기 위해 IoT 디바이스의 기본 펌웨어를 트리거하는 몇몇 다른 일련의 상호작용들의 형태를 취한다.

[0300] 그러나, 다양한 유형의 IoT 디바이스들이 물리적 하드웨어 형태의 가상 무한 어레이를 취할 수 있기 때문에, 최종 사용자들에게 임의의 단일 IoT 디바이스와 상호작용하는 방법을 정확하게 알려주는 표준화된 방법은 없다. 서류는 원래의 포장에 넣어질 수 있지만 온보딩 과정 중에 분실되거나 아니면 사용할 수 없는 경우가 종종 있다. 또한, 가능한 하드웨어 모든 구성에 대한 사용자 정의 선도를 서면으로든 온라인 경험으로든 작성하고, 검증하고, 인쇄하고, 포함하는 작업은 일반적으로 실행 가능하지 않다.

[0301] 다른 해결책에는 최종 사용자에게 다양한 가능성 목록에서 IoT 장치를 선택하도록 요청하는 것이 포함된다. 많은 디바이스가 있는 대규모 배포의 경우, 이는 불쾌한 사용자 경험을 초래한다. 많은 IoT 디바이스 유형들이 서로 비슷해 보이고, 사용자 오류 가능성이 높고, 온보딩 좌절은 반품률 상승으로 이어진다.

[0302] **도 43**을 참조하면, 본 발명의 일 실시예는 사용자의 모바일 디바이스(135) 상에서 실행되는 모바일 앱(1902)에 IoT 디바이스(101)와 같은 인근 디바이스들의 특정 유형에 대한 "인식"을 제공하기 위해 BT 광고 채널(4360)을 사용하는 이러한 제한을 해결한다. 구체적으로, 온보딩 로직(4301)을 포함하는 애플리케이션(4307)은 펌웨어로부터 로딩되어 IoT 디바이스(101) 상에서 실행(예를 들어, IoT 디바이스(101)의 마이크로컨트롤러 또는 다른 프로세서 상에서 실행)될 수 있다.

[0303] 일 실시예에서, IoT 디바이스(101)는 BT 5와 같은 저전력 블루투스 프로토콜을 구현하는 블루투스 모듈(4380)을 포함한다. 앞서 설명된 바와 같이, BT 광고 모드에 있을 때, BT 모듈(4380)은 31바이트까지의 길이일 수 있는 광고 페이로드를 갖는 패킷들을 주기적으로 전송한다. 본 발명의 일 실시예에서, BT 광고 시퀀스 동안, 블루투스 모듈(4380)은 IoT 디바이스(101)의 모델을 고유하게 식별하는 "키"(4350)를 BT 광고 채널(4360)을 통해 전송한다. 키(4350)는 BT 광고 데이터 공간(예를 들어, 최대 31바이트 길이) 내에 적합하도록 크기가 정해지고 제조 동안 BT 모듈(4380)에 프로그래밍되는 수치 값일 수 있다. 대안적으로, 키(4350)는 IoT 디바이스 펌웨어(또는 다른 비휘발성 스토리지)에 프로그래밍되고 온보딩 로직(4301)을 통해 BT 모듈(4380)에 제공될 수 있다.

[0304] 키가 어떻게 포맷되는지에 관계없이, 키는 모바일 앱(4308)에 응답하여 모바일 디바이스(135)의 BT 모듈(4381)을 통해 수신된다. 예시된 바와 같이, 모바일 앱(4308)은 온보딩 프로세스 동안 IoT 서비스(120) 상에서 실행되는 디바이스 온보딩 로직(4321)에 안전하게 접속될 수 있다. 일 구현예에서, 모바일 앱(4308)은 다양한 블루투스 광고 "키들"을 이미지, 텍스트, 비디오, 및/또는 오디오 자산을 포함하는 IoT 디바이스 문서와 연관시키는 키 사전(4310)을 다운로드하고/하거나 미리 설치되어, 사용자에게 IoT 디바이스(101)의 온보딩과 관련된 명령들(4388)을 제공한다. 예를 들어, IoT 디바이스 모델이 키로 식별되면, 명령들(4388)은 일련의 버튼 누름들 또는 다른 IoT 디바이스 입력들을 식별하여, IoT 디바이스(101)로 하여금 그에 IoT 서비스(120) 상의 사용자의 계정이 성공적으로 구성되고 연관될 수 있는 설정 모드로 진입하게 할 수 있다.

[0305] 하나의 특정 구현예에서, 키 사전(4310)은 JSON-포맷 사전이지만, 본 발명의 기본 원리들은 임의의 특정 데이터 포맷으로 제한되지 않는다. 새로운 IoT 디바이스들이 출시됨에 따라, 새로운 사전 데이터 및 연관된 키들이 IoT 서비스(120) 상의 사전 데이터베이스(4320)에 추가될 수 있고, 요구되는 경우 모바일 앱들(4308)에 제공될 수 있다.

[0306] 일 실시예에서, 최종 사용자와 IoT 디바이스(101) 사이의 하드웨어 상호작용이 필요할 때, 모바일 앱(4308)은 BT 모듈(4381)을 통해 그 영역을 스캔하여, 이러한 특수 키를 광고하는 블루투스 디바이스를 찾을 것이다. IoT 디바이스들이 발견되지 않은 경우, IoT 디바이스들에게 IoT 디바이스에 더 가깝게 이동하도록 하고 IoT 디바이스의 켜짐이 보장되도록 하라는 프롬프트를 표시하는 일반 정보가 사용자에게 보일 수 있다. 단 하나의 IoT 디바이스(101)만이 발견되면(즉, 수신된 특정 키(4350)에 기반하여 검출되면), 모바일 앱(4308)은 키 사전(4310)에 의해 특정된 바와 같은 IoT 디바이스의 키에 기반하여 커스텀 명령들(4388)을 디스플레이할 것이다.

[0307] 일 실시예에서, 키 사전(4310)은 IoT 디바이스 온보딩 프로세스를 통해 사용자를 안내하기 위해 비디오, 오디오, 이미지들, 및/또는 텍스트를 포함하는 디바이스별 온보딩 명령들(4388)을 가리키는 키별 하이퍼링크들 또는 다른 어드레스들을 포함할 수 있다. 키 사전(4310)은 또한 모바일 디바이스(135) 상에 로컬로 저장된 텍

스트 또는 다른 콘텐츠를 포함하거나 이에 링크될 수 있다.

[0308] 다수의 IoT 디바이스들로부터의 키들이 모바일 디바이스(135)에 의해 검출된 경우, 모바일 앱(4308)의 일 실시예는 커스텀 명령들(4388)을 디스플레이할 것이다. 예를 들어, IoT 디바이스들의 우선순위화된 목록(예를 들어, 가장 강한 블루투스 신호를 갖는 IoT 디바이스들을 목록 상단에 둠)이 블루투스 신호 강도에 기반하여 제공될 수도 있다. 초기에 보인 IoT 디바이스가 타겟 IoT 디바이스(101)가 아닌 경우에는 추가적인 옵션들이 모바일 앱(4308)의 사용자 인터페이스를 통해 사용자에게 보일 수 있다. 일 구현예에서, 사용자의 계정과 이미 연관된 임의의 IoT 디바이스들은 이용 가능한 IoT 디바이스들의 목록으로부터 필터링된다.

[0309] 타겟 IoT 디바이스(101)가 식별될 때, 모바일 애플리케이션(4308)은 키(4350)를 사용하여 키 사전(4310)에 액세스하여 IoT 디바이스 모델을 식별하고 온보딩 명령들(4388)을 생성할 것이다. 전술된 바와 같이, 온보딩 명령들(4388)은 사용자에게 설정 흐름을 안내하기 위해 드로잉 이미지들, 비디오, 오디오, 및/또는 텍스트를 포함할 수 있다.

[0310] 본 발명의 일 실시예에 따른 방법이 **도 44**에 예시된다. 본 방법은 위에 기술된 시스템 아키텍처의 맥락 내에서 구현될 수 있지만, 임의의 특정 아키텍처로 제한되지 않는다.

[0311] 4401에서, 사용자는 모바일 디바이스에서 IoT 디바이스 앱을 실행하고, (예를 들어, 사용자-선택 가능 메뉴 옵션들을 통해) 새로운 IoT 디바이스의 온보드를 표시한다. IoT 디바이스가 식별될 수 있는 경우, 4402에서 결정되면, 표준 온보딩 프로세스가 사용될 수 있고, 4410에서, IoT 디바이스 앱은 IoT 디바이스 및 IoT 서비스와 통신하여 IoT 디바이스에 IoT 서비스 상의 사용자 계정을 구성하고 연관시킨다.

[0312] IoT 디바이스가 표준 기술들을 통해 식별될 수 없는 경우, 모바일 디바이스는 4403에서 IoT 디바이스 앱에 응답하여 BT 광고 채널들을 청취한다. 다수의 키들이 식별되고 4404에서 결정되면, 4411에서, 키들/IoT 디바이스들의 목록은 전술한 바와 같이 우선순위화 및/또는 필터링된다. 예를 들어, 가장 큰 신호 강도 값을 갖는 키/IoT 디바이스는 우선순위 목록의 상단에 배치될 수 있고/있거나, 사용자의 계정과 이미 연관된 임의의 IoT 디바이스들은 그 목록으로부터 필터링될 수 있다.

[0313] 일단 타겟 IoT 디바이스가 식별되면, IoT 디바이스 앱은 타겟 IoT 디바이스 모델 및/또는 연관된 데이터를 식별하기 위해 키 사전에서의 룩업을 수행한다. 예를 들어, 키 사전은, 4406에서 모바일 디바이스를 통해 제공되는 비디오, 오디오, 텍스트, 및/또는 이미지들을 포함하여 IoT 디바이스 모델에 관련된 다양한 형태의 교육 콘텐츠로의 링크들을 포함할 수 있다. 4407에서, 구성 프로세스가 완료된 후, 타겟 IoT 디바이스는 완전히 온보딩되고 IoT 서비스 상의 사용자의 계정과 연관된다.

[0314] 본 발명의 실시예들은 위에서 설명된 다양한 단계들을 포함할 수 있다. 단계들은 범용 또는 특수 목적 프로세서로 하여금 단계들을 수행하게 하는 데 사용될 수 있는 기계 실행 가능 명령들로 구현될 수 있다. 대안적으로, 이들 단계들은 단계들을 수행하기 위한 하드와이어드(hardwired) 로직을 포함하는 특수 하드웨어 요소들에 의해, 또는 프로그래밍된 컴퓨터 요소들 및 맞춤형 하드웨어 요소들의 임의의 조합에 의해 수행될 수 있다.

[0315] 본원에 설명되는 바와 같이, 명령들은 특정 동작들을 수행하도록 구성된 주문형 반도체(ASIC)와 같은 하드웨어의 특정 구성들, 또는 비일시적 컴퓨터 판독 가능 매체에서 구현되는 메모리에 저장된 미리 결정된 기능 또는 소프트웨어 명령들을 갖는 특정 구성들을 지칭할 수 있다. 따라서, 도면에 도시된 기술들은 하나 이상의 전자 디바이스들(예를 들어, 최종 스테이션, 네트워크 요소 등) 상에 저장되고 그것 상에서 실행되는 코드 및 데이터를 사용하여 구현될 수 있다. 이러한 전자 장치들은 비일시적 컴퓨터 기계 판독 가능 저장 매체(예를 들어, 자기 디스크, 광 디스크, 랜덤 액세스 메모리, 읽기전용 메모리, 플래시 메모리 장치, 위상 변조 메모리) 및 일시적 컴퓨터 기계 판독 가능 통신 매체(예를 들어, 전기적, 광학적, 음향적 또는 다른 형태의 전파된 신호들, 예컨대, 캐리어 파, 적외선 신호, 디지털 신호 등)와 같은 컴퓨터 기계 판독 가능 매체를 사용하여 코드 및 데이터를 저장하고 (내부적으로 그리고/또는 네트워크를 통해 다른 전자 장치들과) 통신한다.

[0316] 부가적으로, 그러한 전자 디바이스들은 전형적으로 하나 이상의 저장 디바이스(비일시적 기계 판독 가능 저장 매체), 사용자 입력/출력 디바이스(예를 들어, 키보드, 터치스크린, 및/또는 디스플레이), 및 네트워크 연결부와 같은 하나 이상의 다른 컴포넌트에 결합된 하나 이상의 프로세서들의 세트를 포함한다.

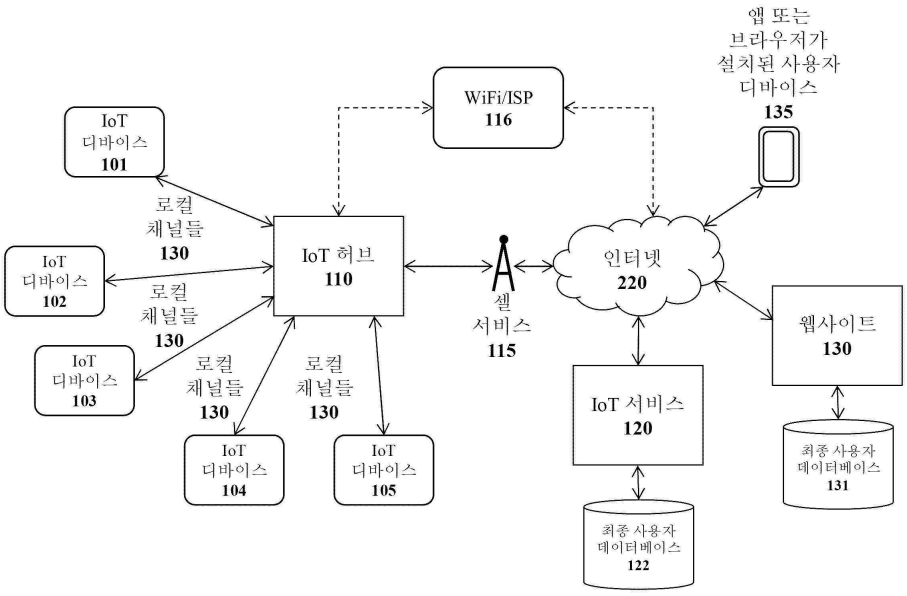
[0317] 프로세서들의 세트와 다른 컴포넌트들의 결합은 전형적으로 하나 이상의 버스들 및 브리지들(또한 버스 제어기들로 지칭됨)을 통해 이루어진다. 저장 디바이스 및 네트워크 트래픽을 반송하는 신호들은 각각 하나 이상의 기계 판독 가능 저장 매체들 및 기계 판독 가능 통신 매체들을 나타낸다. 따라서, 주어진 전자 디바이스의 저

장 디바이스는 전형적으로 그 전자 디바이스의 하나 이상의 프로세서들의 세트 상에서의 실행을 위한 코드 및/또는 데이터를 저장한다. 물론, 본 발명의 실시예의 하나 이상의 부분들이 소프트웨어, 펌웨어, 및/또는 하드웨어의 상이한 조합들을 사용하여 구현될 수 있다.

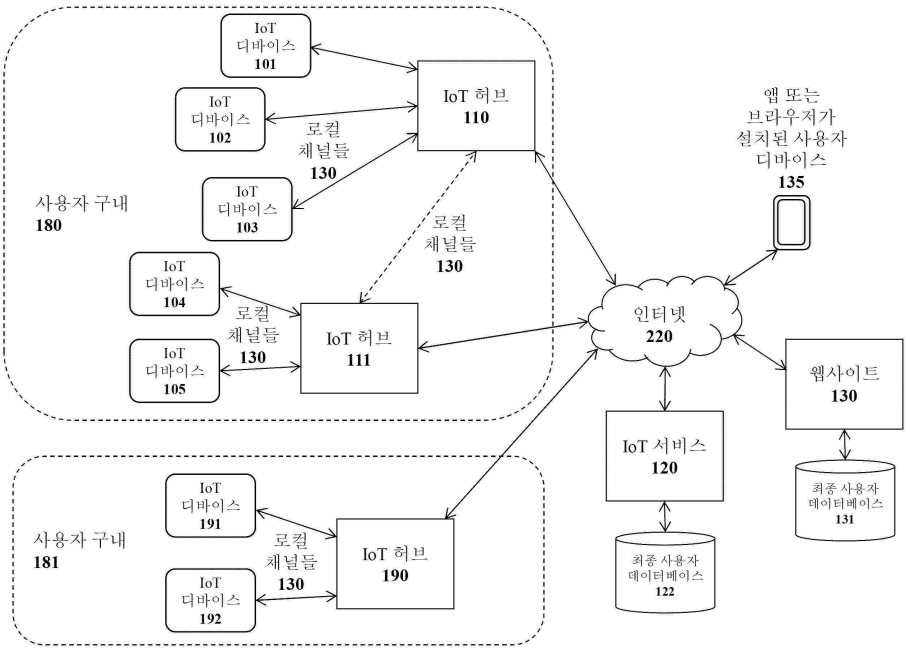
[0318] 이 상세한 설명 전반에 걸쳐, 설명을 위해, 본 발명의 완전한 이해를 제공하기 위해 다수의 구체적인 세부사항들이 제시되었다. 그러나, 본 발명은 이러한 특정 세부사항들 중 일부 없이도 실시될 수 있다는 것이 통상의 기술자에게 명백할 것이다. 특정 경우에서는, 본 발명의 요지를 모호하게 하는 것을 피하기 위해 잘 알려진 구조 및 기능은 상세히 설명되지 않았다. 따라서, 본 발명의 범위 및 사상은 다음의 청구범위에 의해 판단되어야 한다.

## 도면

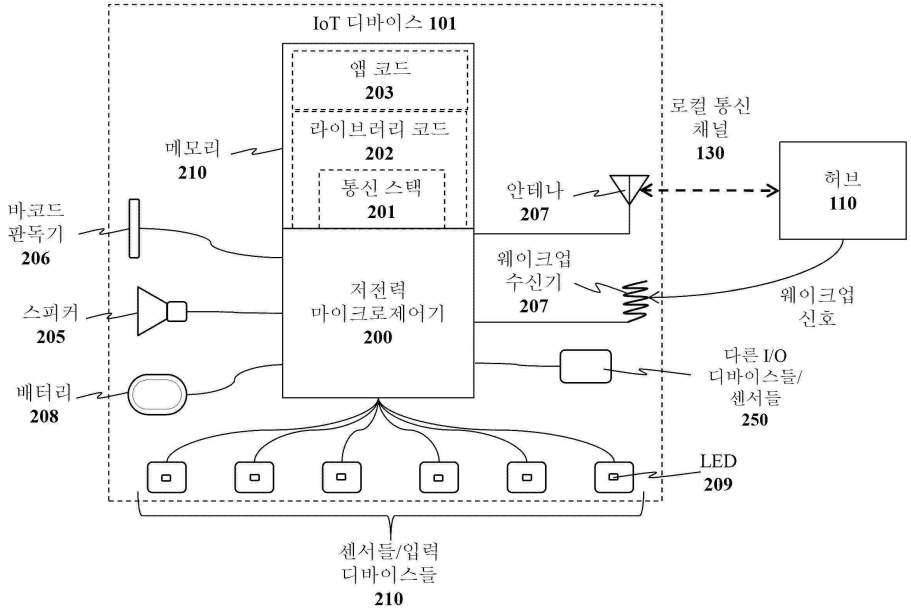
### 도면1a



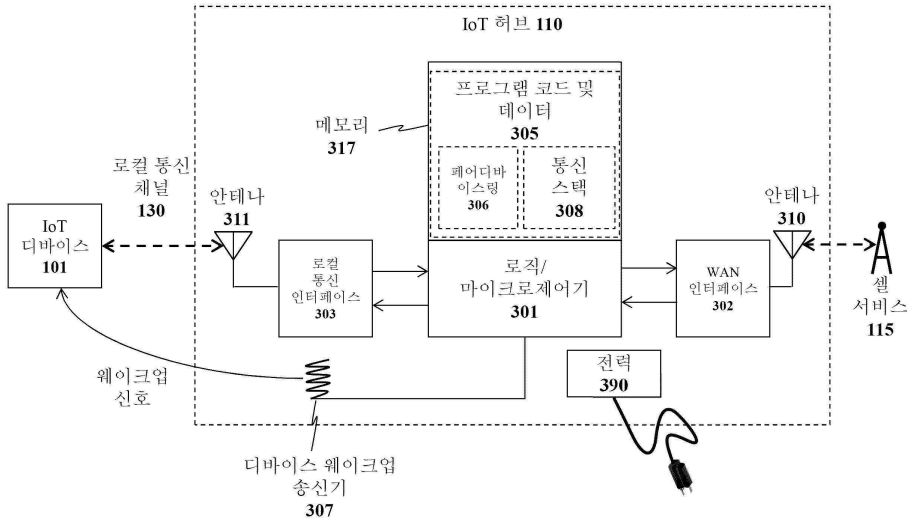

### 도면1b



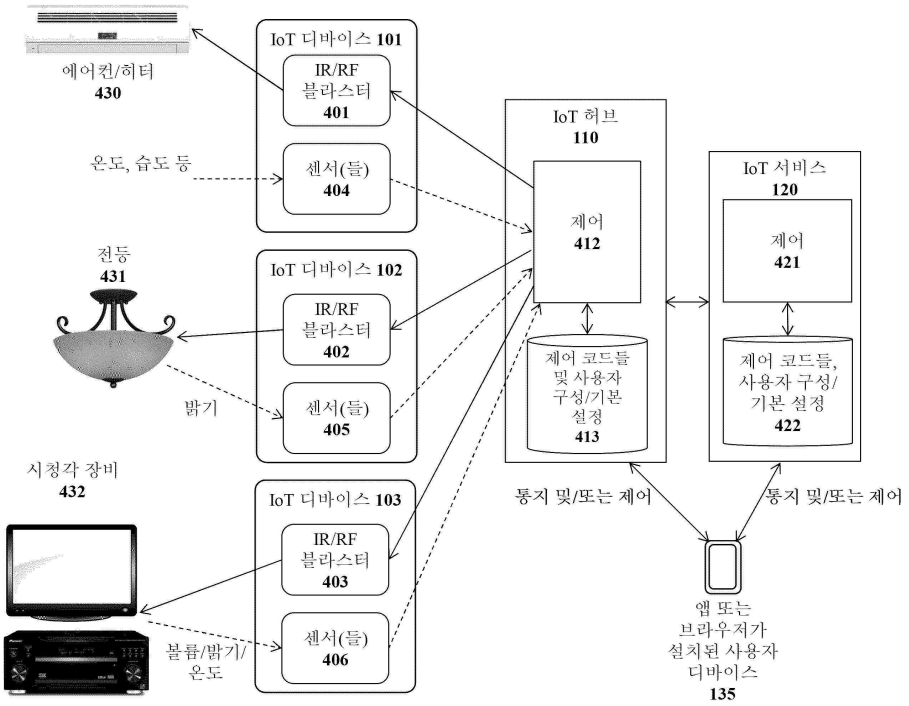
도면2


IoT 디바이스 101
앱 코드
203
라이브러리 코드
202
통신 스택
201
메모리
210
로컬 통신
채널
130
허브
110
안테나
207
바코드
판독기
206
웨이크업
수신기
207
웨이크업
신호
스피커
205
저전력
마이크로제어기
200
다른 I/O
디바이스들/
센서들
250
배터리
208
LED
209
센서들/입력
디바이스들
210

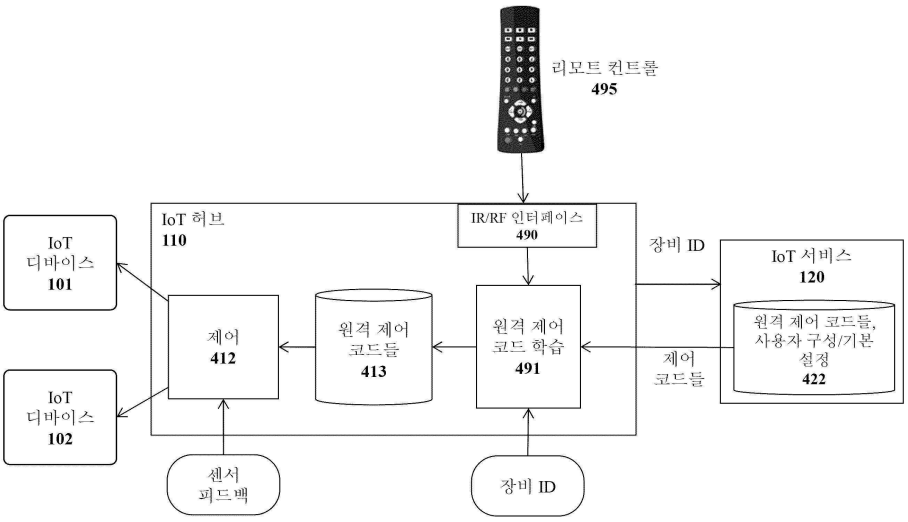
도면3


IoT 허브 110
프로그램 코드 및
데이터
305
메모리
317
페어디바
이스링
306
통신
스택
308
로컬 통신
채널
130
안테나
311
안테나
310
IoT
디바이스
101
로컬
통신
인터페이스
303
로직/
마이크로제어기
301
WAN
인터페이스
302
셀
서비스
115
웨이크업
신호
전력
390
디바이스 웨이크업
송신기
307

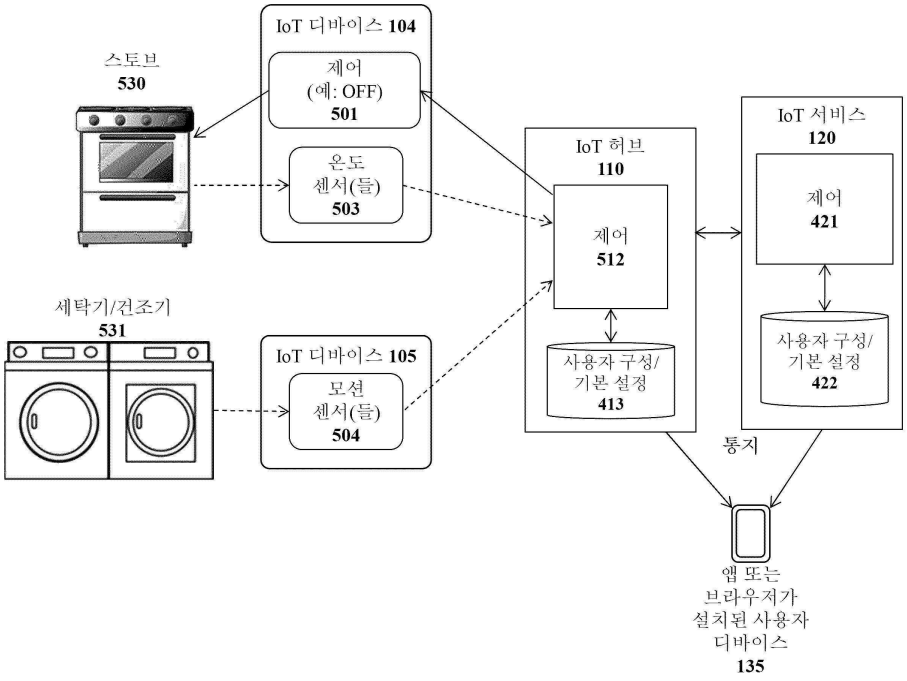
도면4a


에어컨/히터
430
IoT 디바이스 101
IR/RF 블라스터
401
온도, 습도 등
센서(들)
404
IoT 허브
110
제어
412
IoT 서비스
120
제어
421
전등
431
IoT 디바이스 102
IR/RF 블라스터
402
제어 코드들 및 사용자 구성/기본 설정
413
제어 코드들, 사용자 구성/기본 설정
422
밝기
센서(들)
405
시청각 장비
432
IoT 디바이스 103
IR/RF 블라스터
403
통지 및/또는 제어
통지 및/또는 제어
센서(들)
406
볼륨/밝기/온도
앱 또는 브라우저가 설치된 사용자 디바이스
135

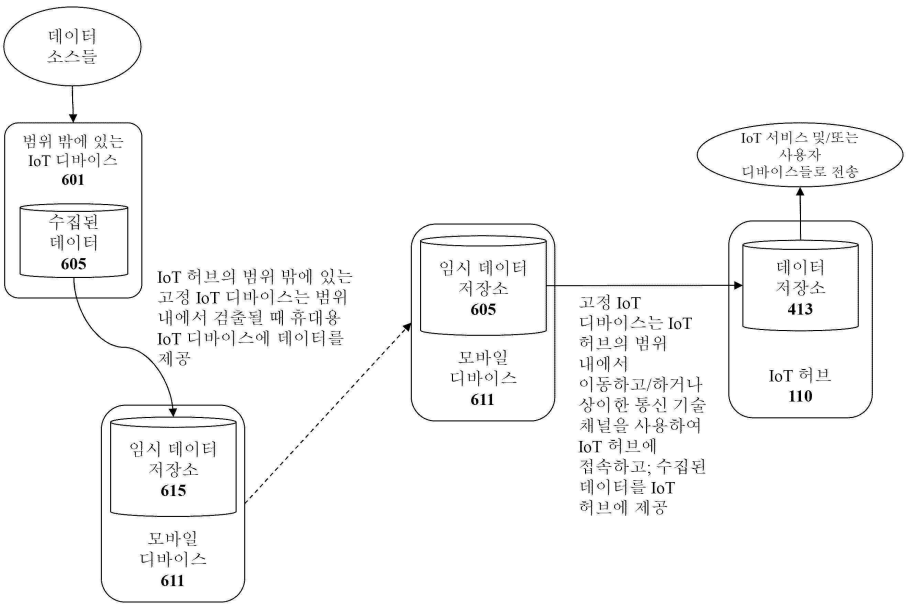
도면4b


리모트 컨트롤
495
IoT 허브
110
IR/RF 인터페이스
490
장비 ID
IoT 서비스
120
IoT 디바이스
101
제어
412
원격 제어 코드들
413
원격 제어 코드 학습
491
원격 제어 코드들, 사용자 구성/기본 설정
422
제어 코드들
IoT 디바이스
102
센서 피드백
장비 ID

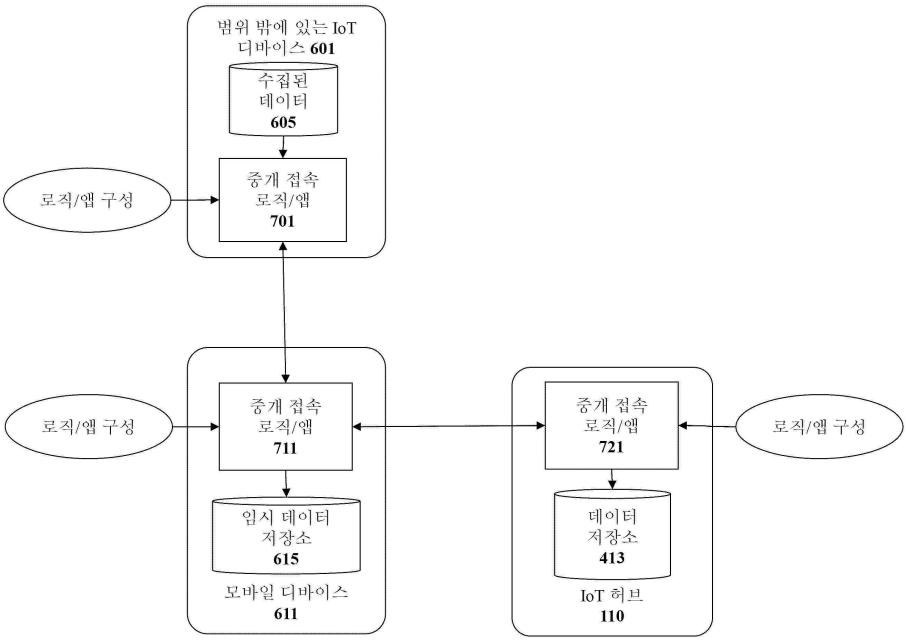
도면5


스토브
530
IoT 디바이스 104
제어
(예: OFF)
501
온도
센서(들)
503
세탁기/건조기
531
IoT 디바이스 105
모션
센서(들)
504
IoT 허브
110
제어
512
사용자 구성/
기본 설정
413
IoT 서비스
120
제어
421
사용자 구성/
기본 설정
422
통지
앱 또는
브라우저가
설치된 사용자
디바이스
135

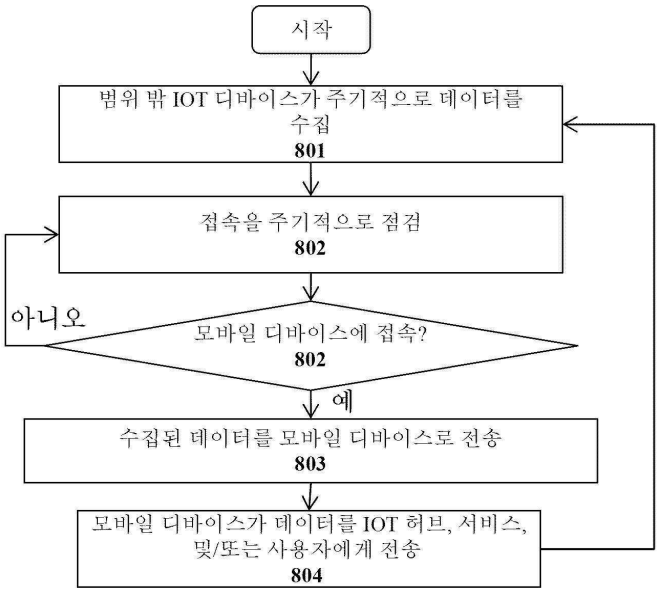
도면6


데이터
소스들
범위 밖에 있는
IoT 디바이스
601
수집된
데이터
605
IoT 허브의 범위 밖에 있는
고정 IoT 디바이스는 범위
내에서 검출될 때 휴대용
IoT 디바이스에 데이터를
제공
임시 데이터
저장소
615
모바일
디바이스
611
임시 데이터
저장소
605
모바일
디바이스
611
고정 IoT
디바이스는 IoT
허브의 범위
내에서
이동하고/하거나
상이한 통신 기술
채널을 사용하여
IoT 허브에
접속하고; 수집된
데이터를 IoT
허브에 제공
IoT 서비스 및/또는
사용자
디바이스들로 전송
데이터
저장소
413
IoT 허브
110

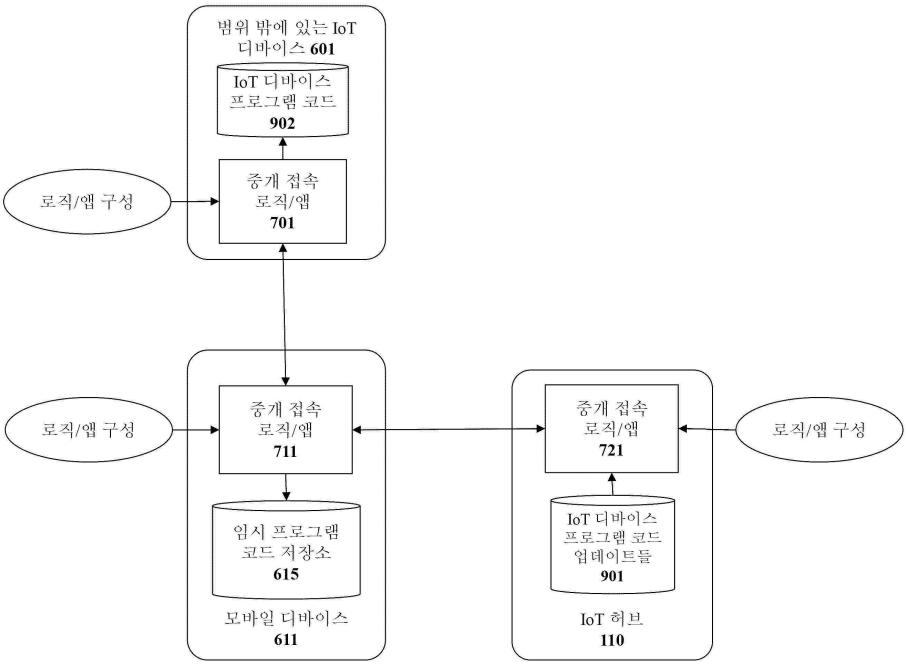
도면7


범위 밖에 있는 IoT 디바이스 601
수집된 데이터 605
중개 접속 로직/앱 701
로직/앱 구성
중개 접속 로직/앱 711
로직/앱 구성
임시 데이터 저장소 615
모바일 디바이스 611
중개 접속 로직/앱 721
로직/앱 구성
데이터 저장소 413
IoT 허브 110

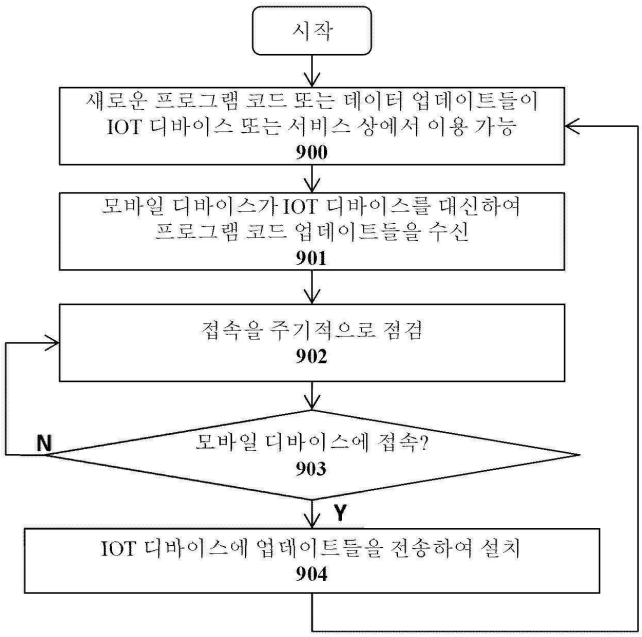
도면8


시작
범위 밖 IOT 디바이스가 주기적으로 데이터를 수집 801
접속을 주기적으로 점검 802
아니오
모바일 디바이스에 접속? 802
예
수집된 데이터를 모바일 디바이스로 전송 803
모바일 디바이스가 데이터를 IOT 허브, 서비스, 및/또는 사용자에게 전송 804

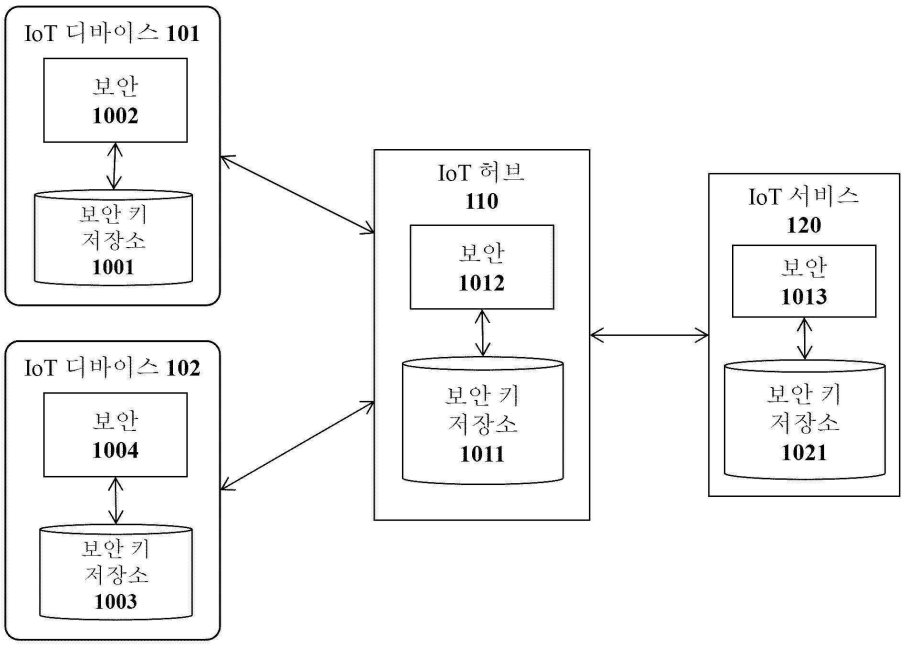
도면9a


범위 밖에 있는 IoT 디바이스 601
IoT 디바이스 프로그램 코드 902
중개 접속 로직/앱 701
로직/앱 구성
로직/앱 구성
중개 접속 로직/앱 711
임시 프로그램 코드 저장소 615
모바일 디바이스 611
중개 접속 로직/앱 721
IoT 디바이스 프로그램 코드 업데이트들 901
IoT 허브 110
로직/앱 구성

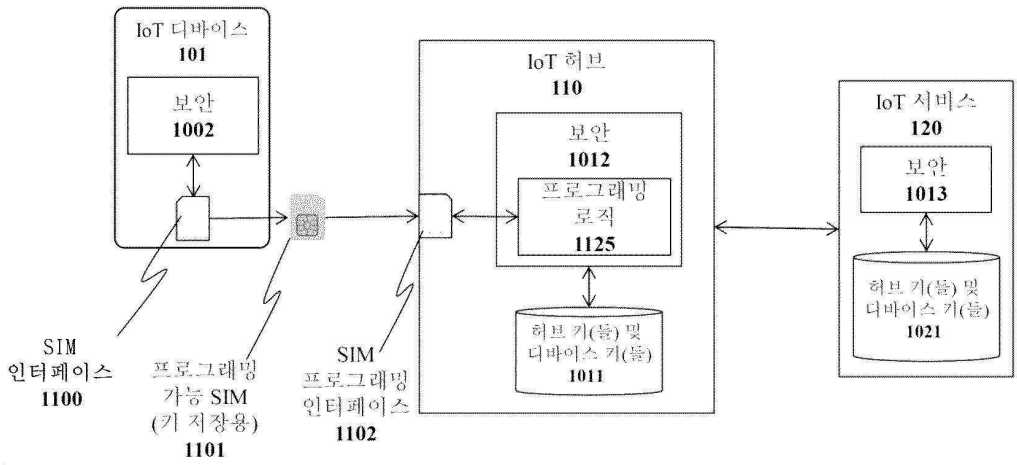
도면9b


시작
새로운 프로그램 코드 또는 데이터 업데이트들이 IOT 디바이스 또는 서비스 상에서 이용 가능 900
모바일 디바이스가 IOT 디바이스를 대신하여 프로그램 코드 업데이트들을 수신 901
접속을 주기적으로 점검 902
모바일 디바이스에 접속? 903
N
Y
IOT 디바이스에 업데이트들을 전송하여 설치 904

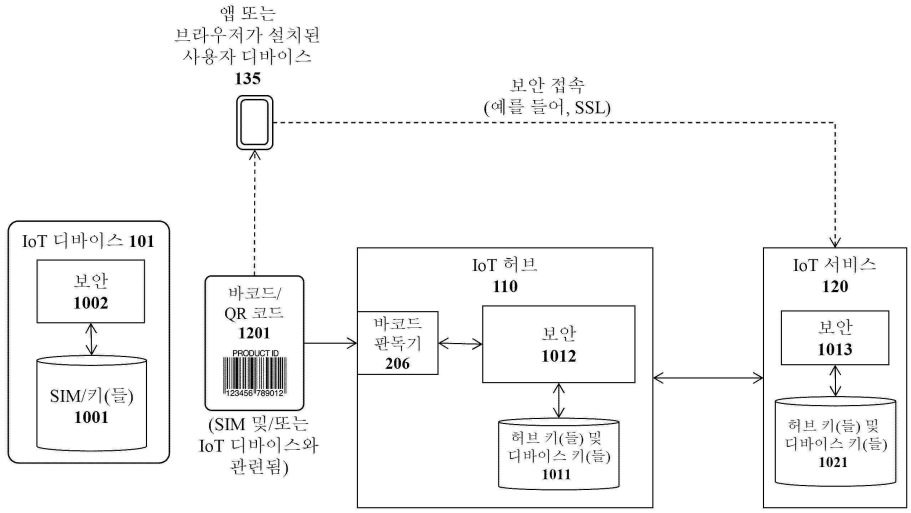
도면10


IoT 디바이스 101
보안
1002
보안 키
저장소
1001
IoT 디바이스 102
보안
1004
보안 키
저장소
1003
IoT 허브
110
보안
1012
보안 키
저장소
1011
IoT 서비스
120
보안
1013
보안 키
저장소
1021

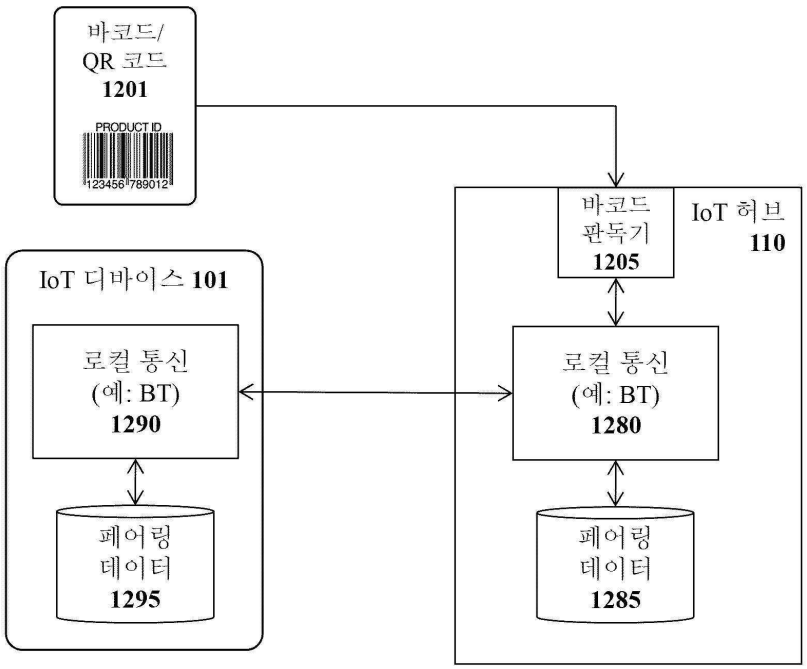
도면11


IoT 디바이스
101
보안
1002
SIM
인터페이스
1100
프로그래밍
가능 SIM
(키 저장용)
1101
SIM
프로그래밍
인터페이스
1102
IoT 허브
110
보안
1012
프로그래밍
로직
1125
허브 키(들) 및
디바이스 키(들)
1011
IoT 서비스
120
보안
1013
허브 키(들) 및
디바이스 키(들)
1021

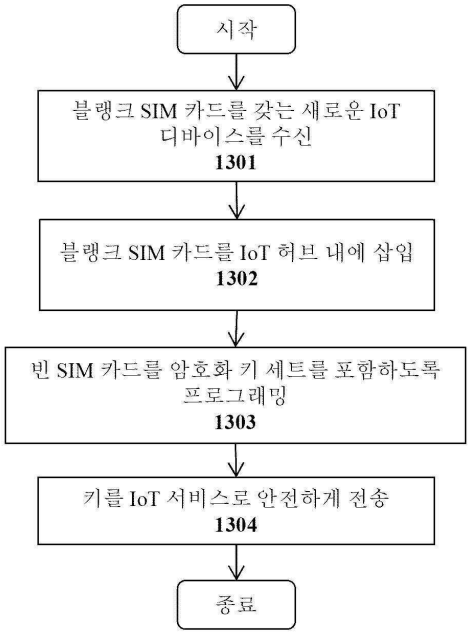
도면12a


앱 또는
브라우저가 설치된
사용자 디바이스
135
보안 접속
(예를 들어, SSL)
IoT 디바이스 101
보안
1002
SIM/키(들)
1001
바코드/
QR 코드
1201
PRODUCT ID
123456 789012
(SIM 및/또는
IoT 디바이스와
관련됨)
IoT 허브
110
바코드
판독기
206
보안
1012
허브 키(들) 및
디바이스 키(들)
1011
IoT 서비스
120
보안
1013
허브 키(들) 및
디바이스 키(들)
1021

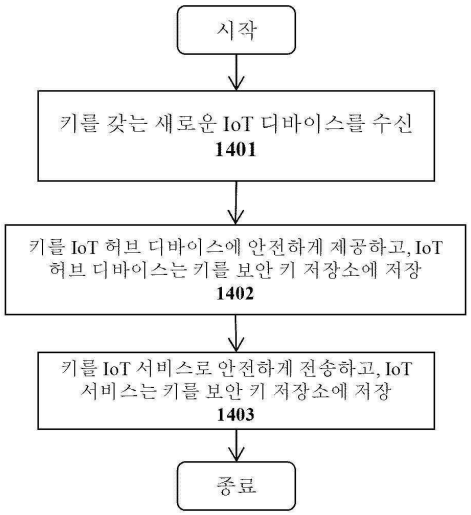
도면12b


바코드/
QR 코드
1201
PRODUCT ID
123456 789012
바코드
판독기
1205
IoT 허브
110
IoT 디바이스 101
로컬 통신
(예: BT)
1290
로컬 통신
(예: BT)
1280
페어링
데이터
1295
페어링
데이터
1285

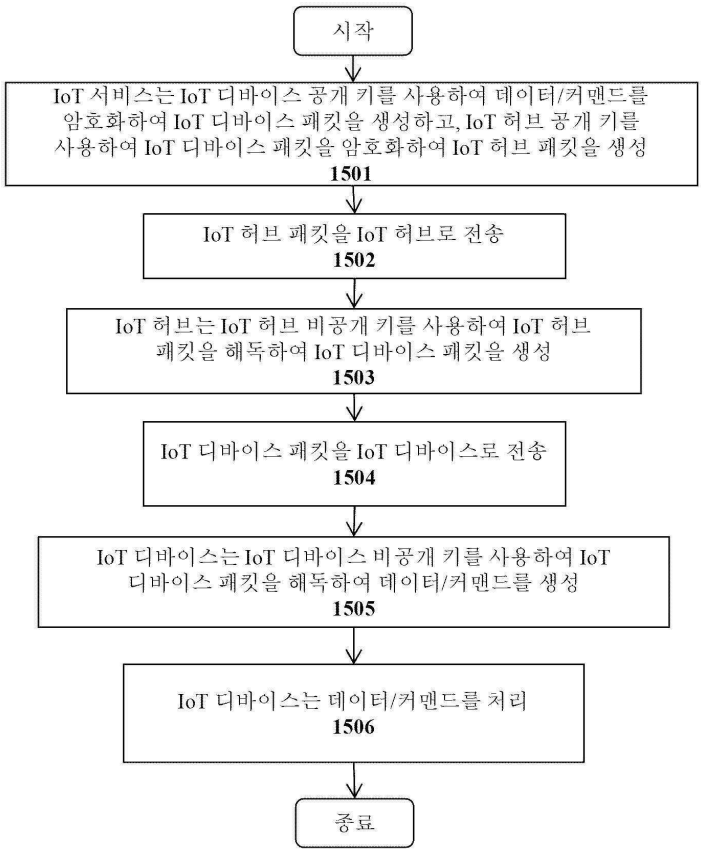
도면13


시작
블랭크 SIM 카드를 갖는 새로운 IoT 디바이스를 수신
1301
블랭크 SIM 카드를 IoT 허브 내에 삽입
1302
빈 SIM 카드를 암호화 키 세트를 포함하도록 프로그래밍
1303
키를 IoT 서비스로 안전하게 전송
1304
종료

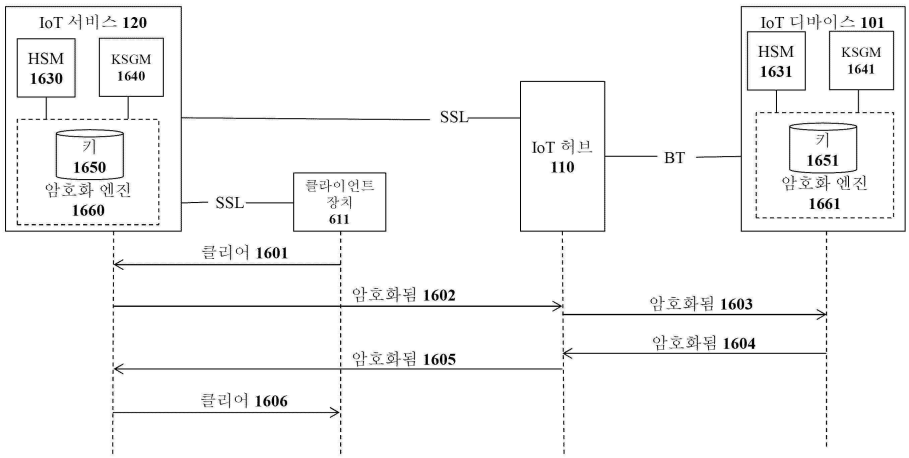
도면14


시작
키를 갖는 새로운 IoT 디바이스를 수신
1401
키를 IoT 허브 디바이스에 안전하게 제공하고, IoT 허브 디바이스는 키를 보안 키 저장소에 저장
1402
키를 IoT 서비스로 안전하게 전송하고, IoT 서비스는 키를 보안 키 저장소에 저장
1403
종료

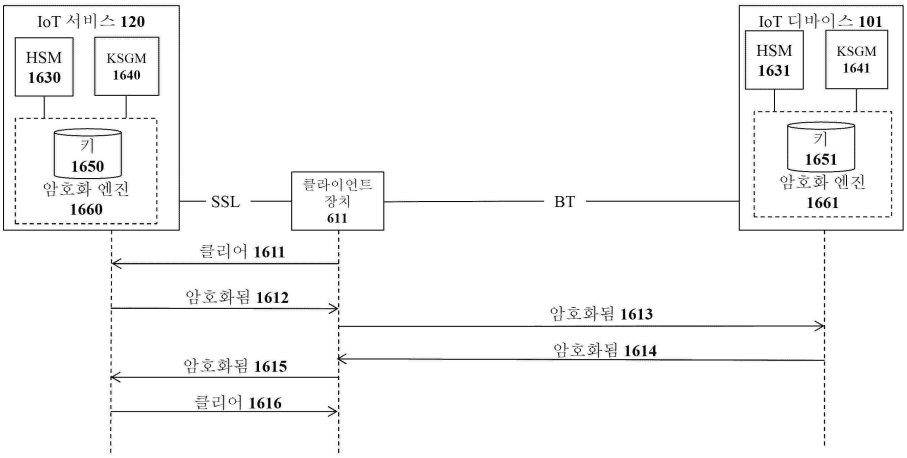
도면15


시작
IoT 서비스는 IoT 디바이스 공개 키를 사용하여 데이터/커맨드를 암호화하여 IoT 디바이스 패킷을 생성하고, IoT 허브 공개 키를 사용하여 IoT 디바이스 패킷을 암호화하여 IoT 허브 패킷을 생성
1501
IoT 허브 패킷을 IoT 허브로 전송
1502
IoT 허브는 IoT 허브 비공개 키를 사용하여 IoT 허브 패킷을 해독하여 IoT 디바이스 패킷을 생성
1503
IoT 디바이스 패킷을 IoT 디바이스로 전송
1504
IoT 디바이스는 IoT 디바이스 비공개 키를 사용하여 IoT 디바이스 패킷을 해독하여 데이터/커맨드를 생성
1505
IoT 디바이스는 데이터/커맨드를 처리
1506
종료

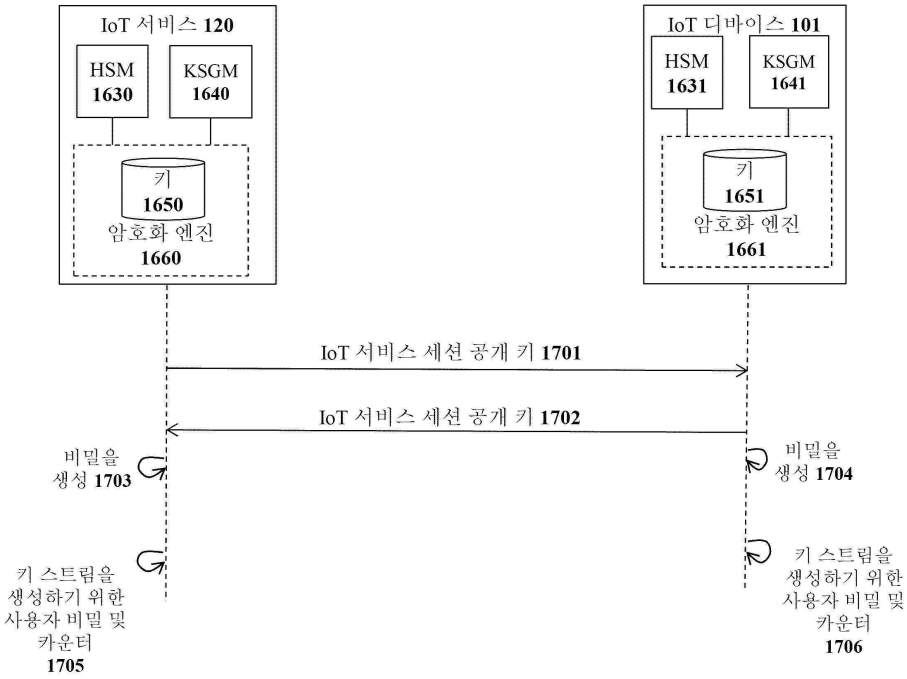
도면16a


IoT 서비스 120
HSM 1630
KSGM 1640
키 1650
암호화 엔진 1660
SSL
SSL
클라이언트 장치 611
IoT 허브 110
BT
IoT 디바이스 101
HSM 1631
KSGM 1641
키 1651
암호화 엔진 1661
클리어 1601
암호화됨 1602
암호화됨 1603
암호화됨 1604
암호화됨 1605
클리어 1606

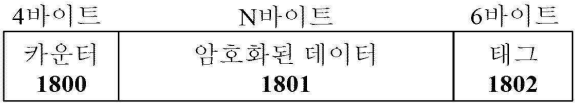
도면16b



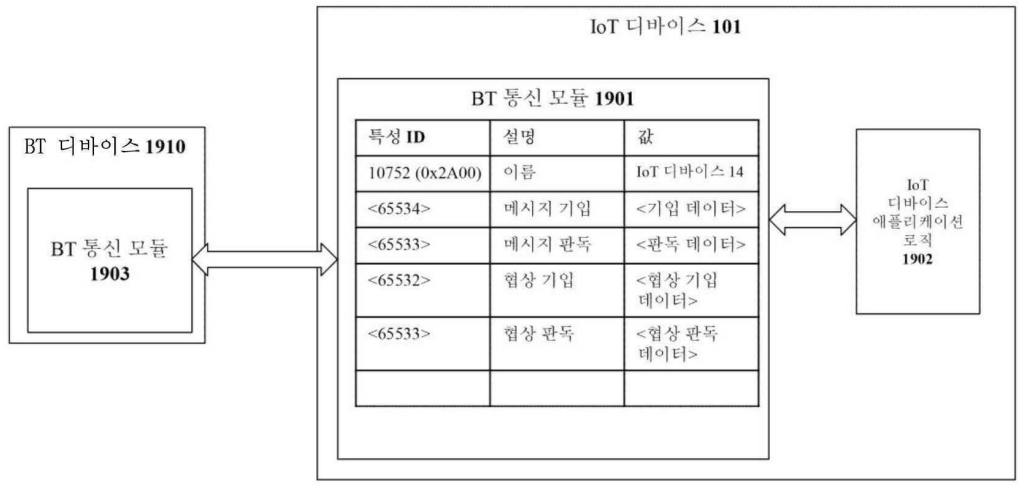

도면17



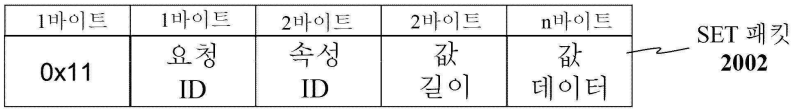

도면18

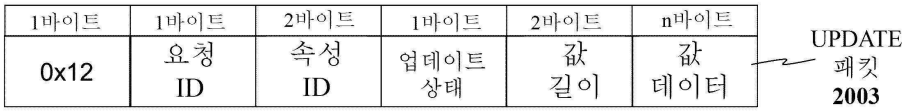


도면19

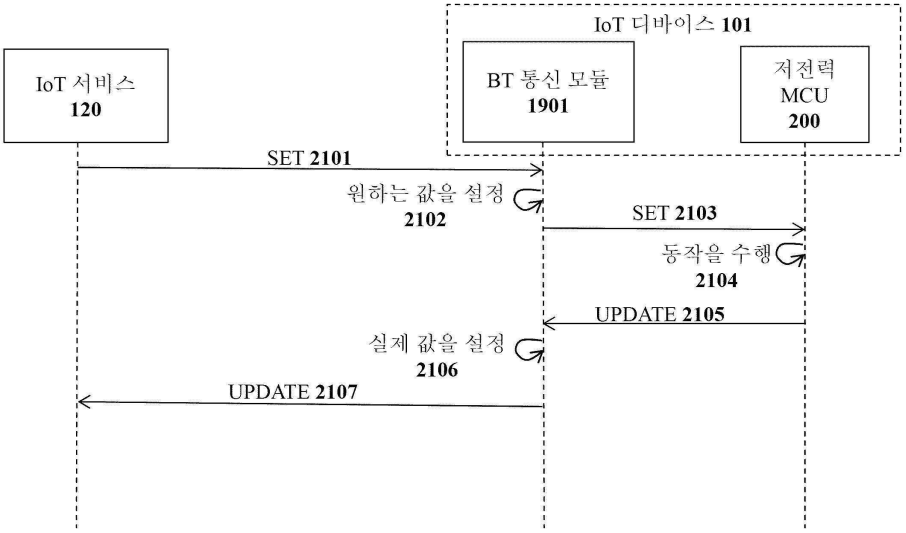



도면20

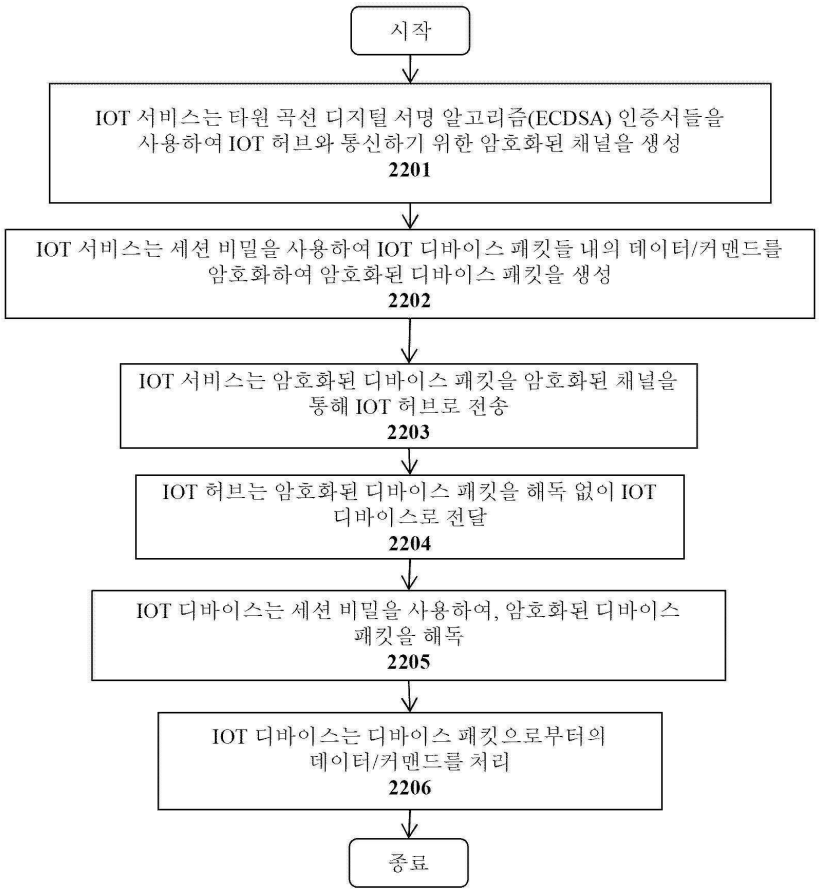



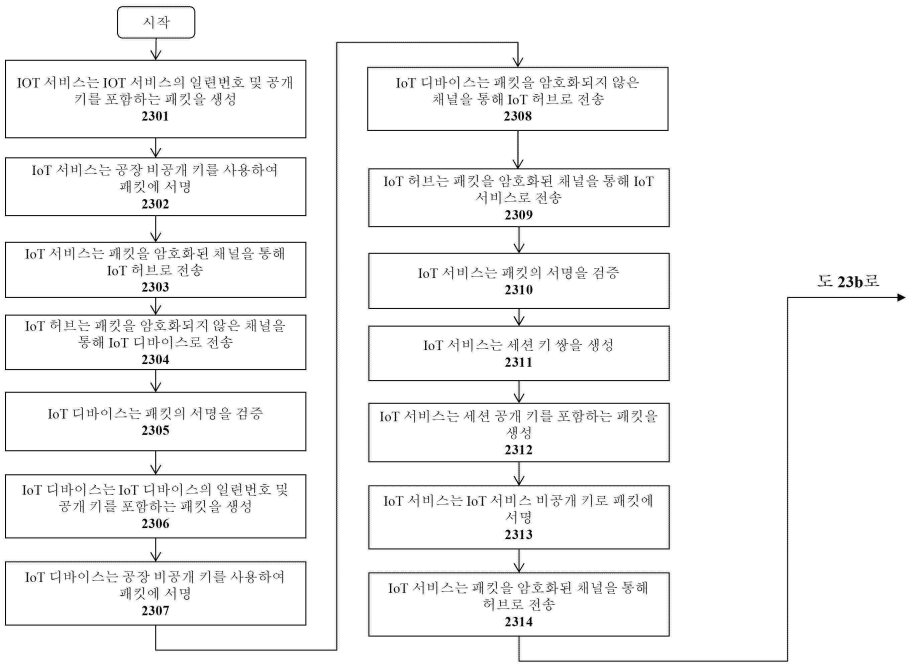

도면21



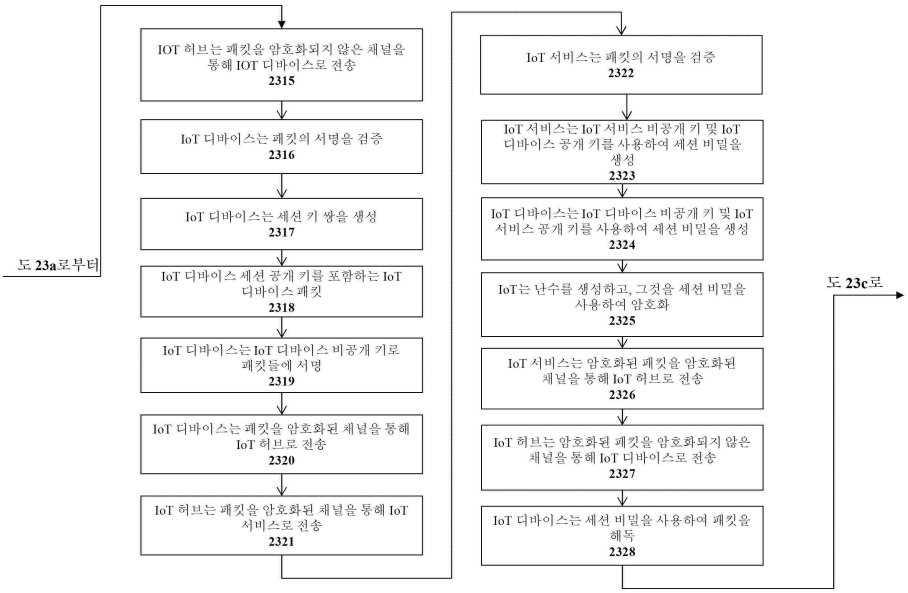
도면22


시작
IOT 서비스는 타원 곡선 디지털 서명 알고리즘(ECDSA) 인증서들을 사용하여 IOT 허브와 통신하기 위한 암호화된 채널을 생성
2201
IOT 서비스는 세션 비밀을 사용하여 IOT 디바이스 패킷들 내의 데이터/커맨드를 암호화하여 암호화된 디바이스 패킷을 생성
2202
IOT 서비스는 암호화된 디바이스 패킷을 암호화된 채널을 통해 IOT 허브로 전송
2203
IOT 허브는 암호화된 디바이스 패킷을 해독 없이 IOT 디바이스로 전달
2204
IOT 디바이스는 세션 비밀을 사용하여, 암호화된 디바이스 패킷을 해독
2205
IOT 디바이스는 디바이스 패킷으로부터의 데이터/커맨드를 처리
2206
종료

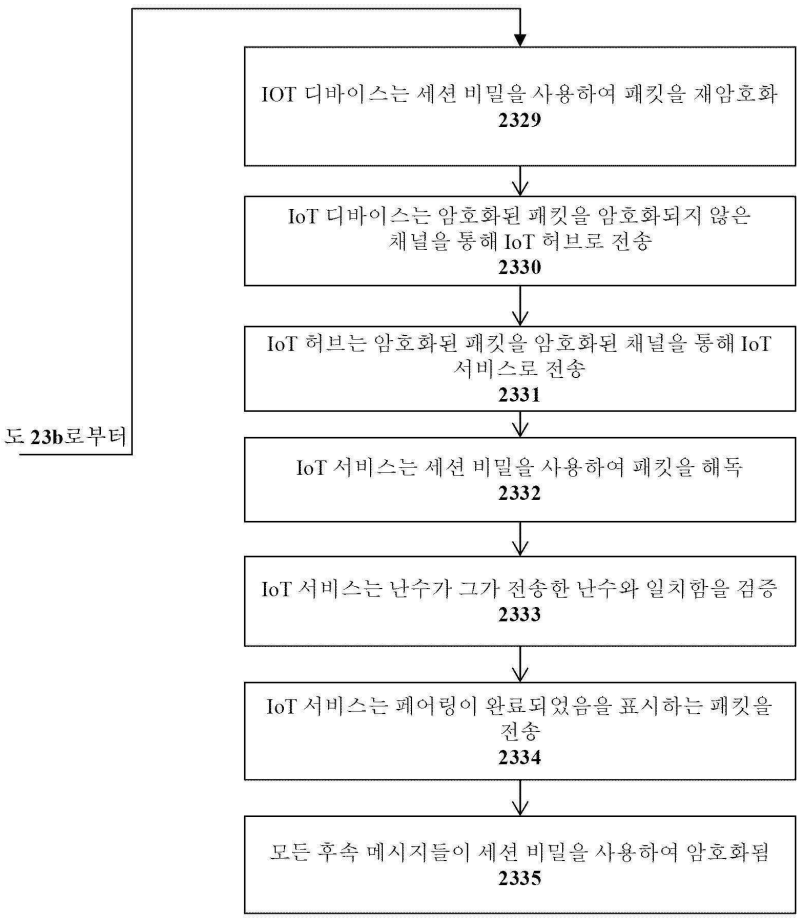
도면23a


시작
IOT 서비스는 IOT 서비스의 일련번호 및 공개 키를 포함하는 패킷을 생성
2301
IoT 서비스는 공장 비공개 키를 사용하여 패킷에 서명
2302
IoT 서비스는 패킷을 암호화된 채널을 통해 IoT 허브로 전송
2303
IoT 허브는 패킷을 암호화되지 않은 채널을 통해 IoT 디바이스로 전송
2304
IoT 디바이스는 패킷의 서명을 검증
2305
IoT 디바이스는 IoT 디바이스의 일련번호 및 공개 키를 포함하는 패킷을 생성
2306
IoT 디바이스는 공장 비공개 키를 사용하여 패킷에 서명
2307
IoT 디바이스는 패킷을 암호화되지 않은 채널을 통해 IoT 허브로 전송
2308
IoT 허브는 패킷을 암호화된 채널을 통해 IoT 서비스로 전송
2309
IoT 서비스는 패킷의 서명을 검증
2310
IoT 서비스는 세션 키 쌍을 생성
2311
IoT 서비스는 세션 공개 키를 포함하는 패킷을 생성
2312
IoT 서비스는 IoT 서비스 비공개 키로 패킷에 서명
2313
IoT 서비스는 패킷을 암호화된 채널을 통해 허브로 전송
2314
도 23b로

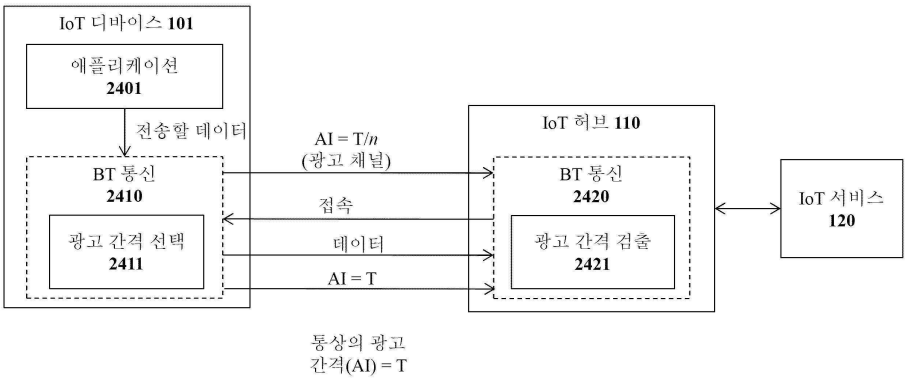
도면23b


도 23a로부터
IOT 허브는 패킷을 암호화되지 않은 채널을 통해 IOT 디바이스로 전송
2315
IoT 디바이스는 패킷의 서명을 검증
2316
IoT 디바이스는 세션 키 쌍을 생성
2317
IoT 디바이스 세션 공개 키를 포함하는 IoT 디바이스 패킷
2318
IoT 디바이스는 IoT 디바이스 비공개 키로 패킷들에 서명
2319
IoT 디바이스는 패킷을 암호화된 채널을 통해 IoT 허브로 전송
2320
IoT 허브는 패킷을 암호화된 채널을 통해 IoT 서비스로 전송
2321
IoT 서비스는 패킷의 서명을 검증
2322
IoT 서비스는 IoT 서비스 비공개 키 및 IoT 디바이스 공개 키를 사용하여 세션 비밀을 생성
2323
IoT 디바이스는 IoT 디바이스 비공개 키 및 IoT 서비스 공개 키를 사용하여 세션 비밀을 생성
2324
IoT는 난수를 생성하고, 그것을 세션 비밀을 사용하여 암호화
2325
IoT 서비스는 암호화된 패킷을 암호화된 채널을 통해 IoT 허브로 전송
2326
IoT 허브는 암호화된 패킷을 암호화되지 않은 채널을 통해 IoT 디바이스로 전송
2327
IoT 디바이스는 세션 비밀을 사용하여 패킷을 해독
2328
도 23c로

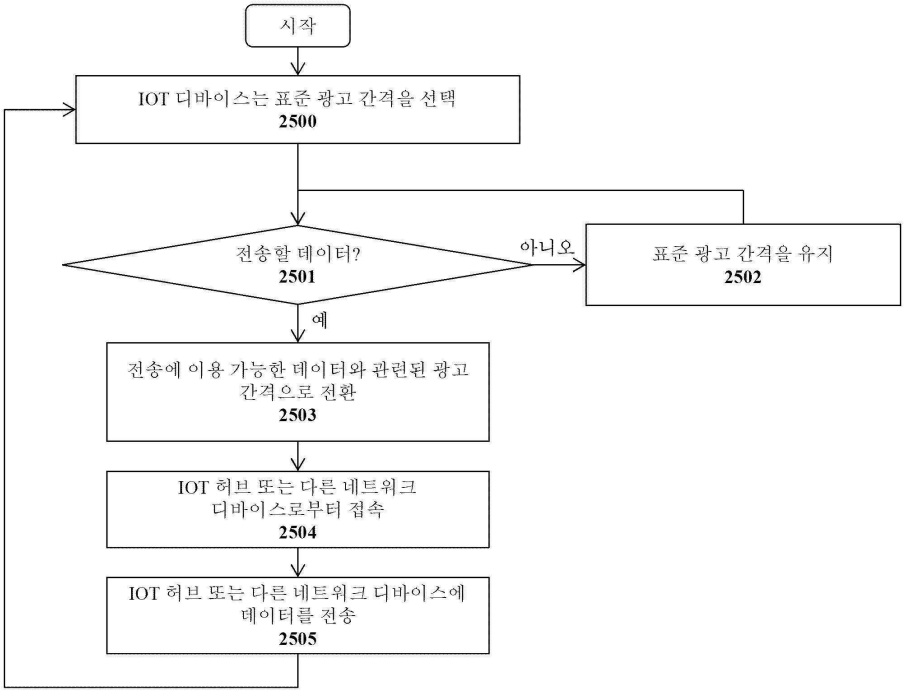
도면23c


도 23b로부터
IOT 디바이스는 세션 비밀을 사용하여 패킷을 재암호화
2329
IoT 디바이스는 암호화된 패킷을 암호화되지 않은 채널을 통해 IoT 허브로 전송
2330
IoT 허브는 암호화된 패킷을 암호화된 채널을 통해 IoT 서비스로 전송
2331
IoT 서비스는 세션 비밀을 사용하여 패킷을 해독
2332
IoT 서비스는 난수가 그가 전송한 난수와 일치함을 검증
2333
IoT 서비스는 페어링이 완료되었음을 표시하는 패킷을 전송
2334
모든 후속 메시지들이 세션 비밀을 사용하여 암호화됨
2335

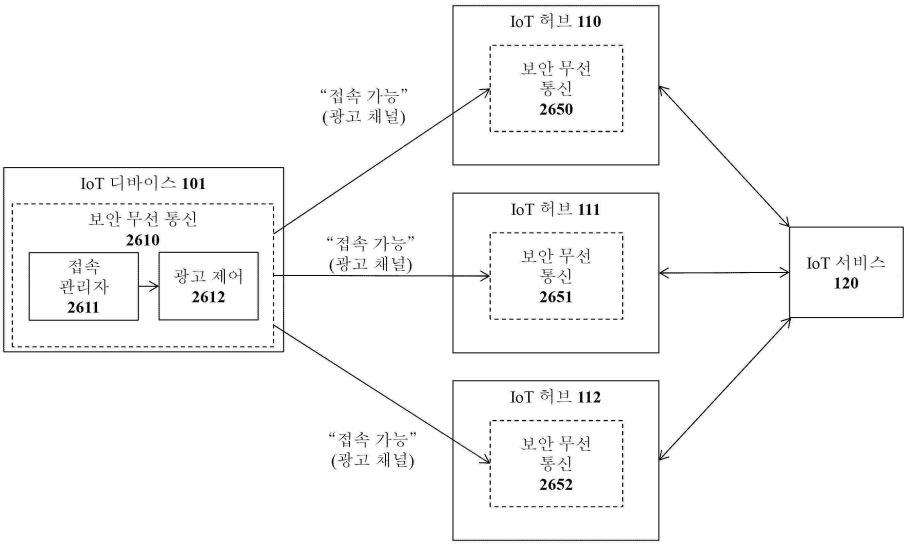
도면24


IoT 디바이스 101
애플리케이션
2401
전송할 데이터
BT 통신
2410
광고 간격 선택
2411
AI = T/n
(광고 채널)
접속
데이터
AI = T
IoT 허브 110
BT 통신
2420
광고 간격 검출
2421
IoT 서비스
120
통상의 광고
간격(AI) = T

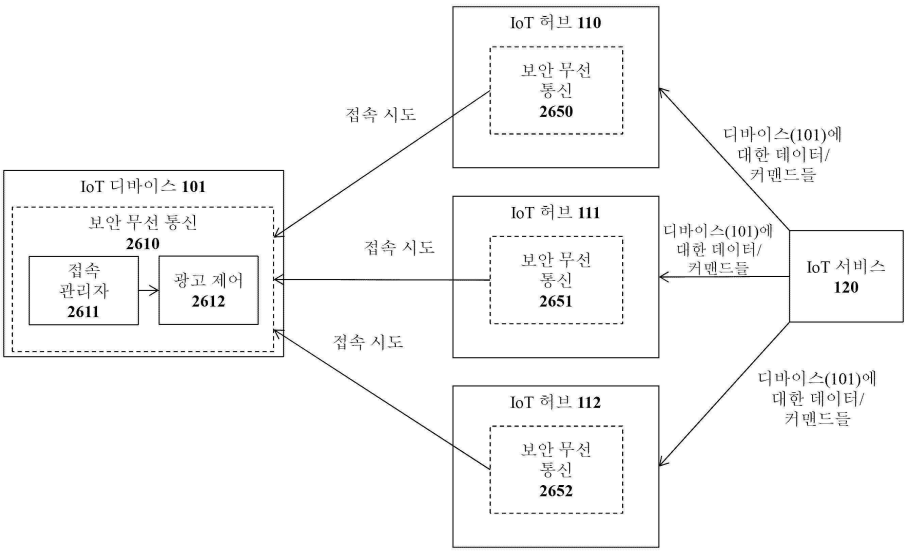
도면25


시작
IOT 디바이스는 표준 광고 간격을 선택
2500
전송할 데이터?
2501
아니오
표준 광고 간격을 유지
2502
예
전송에 이용 가능한 데이터와 관련된 광고 간격으로 전환
2503
IOT 허브 또는 다른 네트워크 디바이스로부터 접속
2504
IOT 허브 또는 다른 네트워크 디바이스에 데이터를 전송
2505

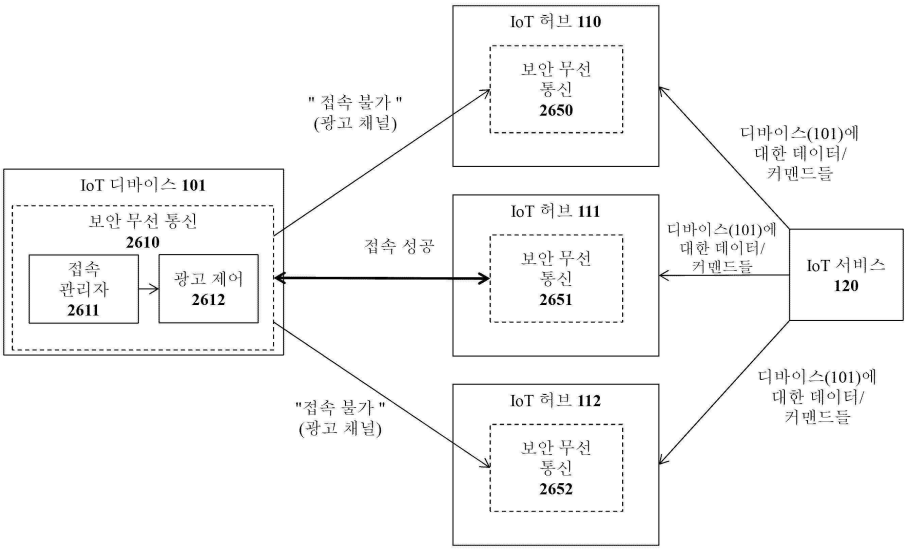
도면26a


IoT 허브 110
보안 무선 통신 2650
"접속 가능" (광고 채널)
IoT 디바이스 101
보안 무선 통신 2610
접속 관리자 2611
광고 제어 2612
IoT 허브 111
보안 무선 통신 2651
"접속 가능" (광고 채널)
IoT 서비스 120
IoT 허브 112
보안 무선 통신 2652
"접속 가능" (광고 채널)

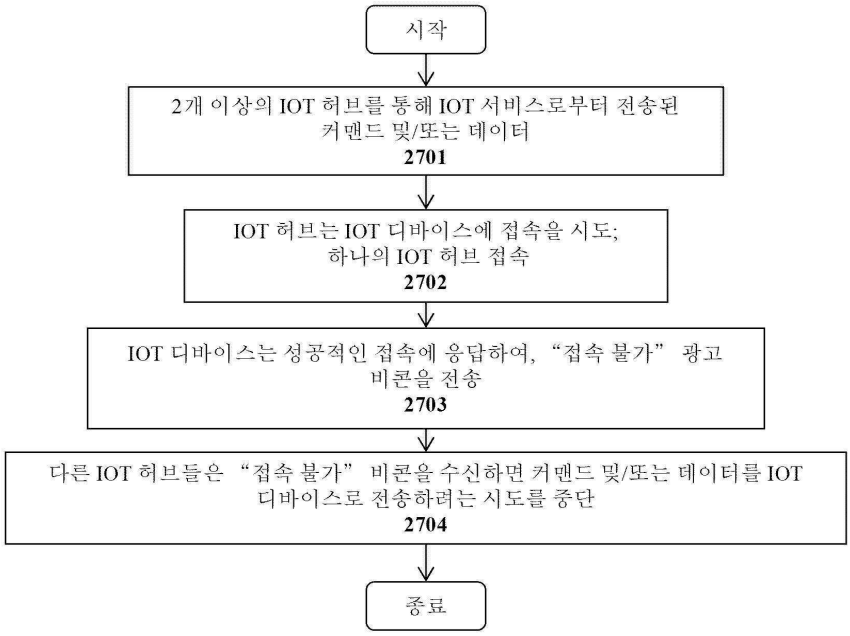
도면26b


IoT 허브 110
보안 무선 통신 2650
접속 시도
디바이스(101)에 대한 데이터/ 커맨드들
IoT 디바이스 101
보안 무선 통신 2610
접속 관리자 2611
광고 제어 2612
IoT 허브 111
보안 무선 통신 2651
접속 시도
디바이스(101)에 대한 데이터/ 커맨드들
IoT 서비스 120
접속 시도
IoT 허브 112
보안 무선 통신 2652
디바이스(101)에 대한 데이터/ 커맨드들

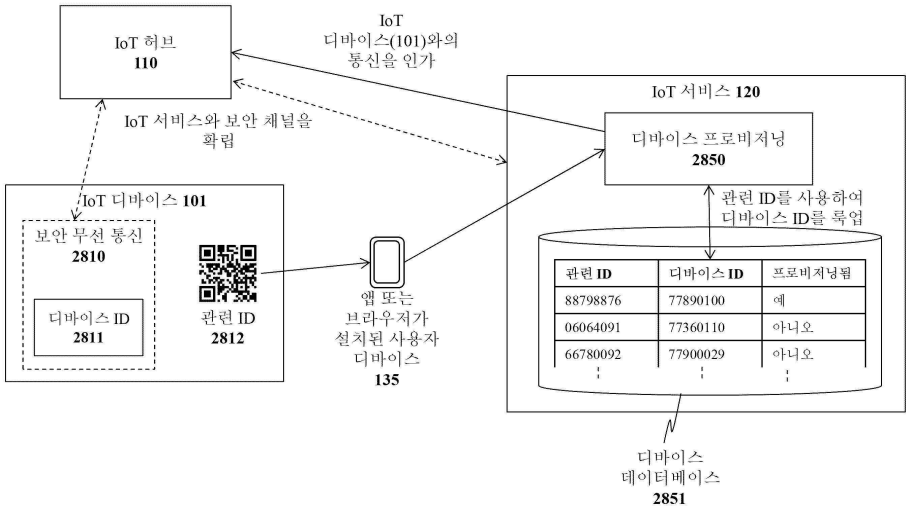
도면26c


IoT 허브 110
보안 무선 통신 2650
" 접속 불가 " (광고 채널)
디바이스(101)에 대한 데이터/ 커맨드들
IoT 디바이스 101
보안 무선 통신 2610
접속 관리자 2611
광고 제어 2612
접속 성공
IoT 허브 111
보안 무선 통신 2651
디바이스(101)에 대한 데이터/ 커맨드들
IoT 서비스 120
"접속 불가 " (광고 채널)
IoT 허브 112
보안 무선 통신 2652
디바이스(101)에 대한 데이터/ 커맨드들

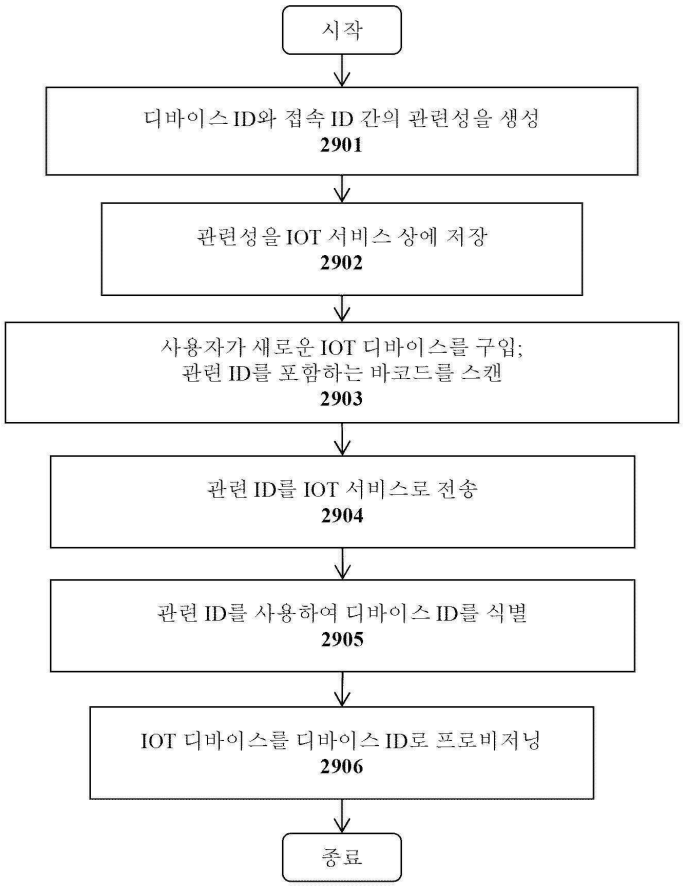
도면27


시작
2개 이상의 IOT 허브를 통해 IOT 서비스로부터 전송된 커맨드 및/또는 데이터 2701
IOT 허브는 IOT 디바이스에 접속을 시도; 하나의 IOT 허브 접속 2702
IOT 디바이스는 성공적인 접속에 응답하여, “접속 불가” 광고 비콘을 전송 2703
다른 IOT 허브들은 “접속 불가” 비콘을 수신하면 커맨드 및/또는 데이터를 IOT 디바이스로 전송하려는 시도를 중단 2704
종료

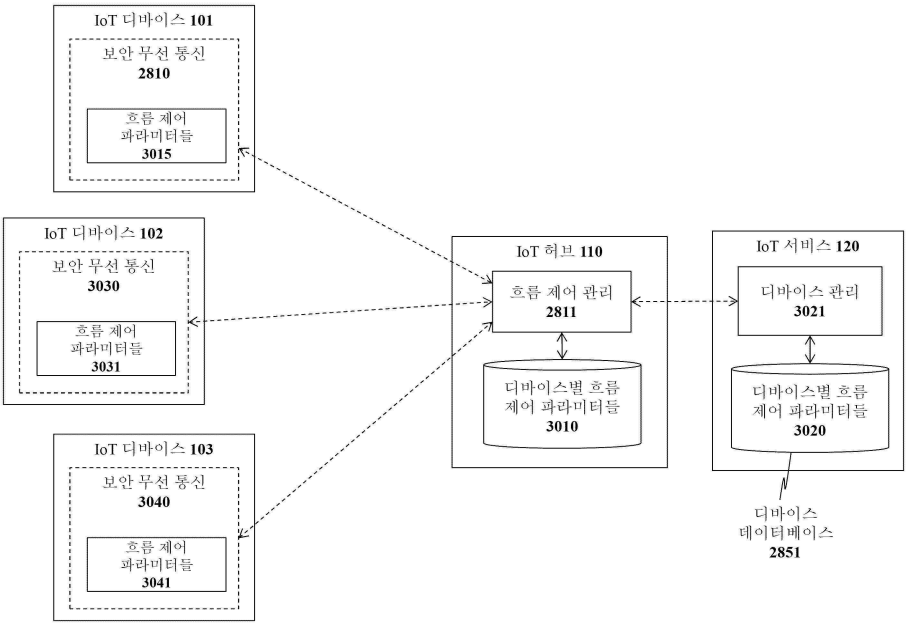
도면28


IoT 허브
110
IoT
디바이스(101)와의
통신을 인가
IoT 서비스와 보안 채널을
확립
IoT 서비스 120
디바이스 프로비저닝
2850
관련 ID를 사용하여
디바이스 ID를 룩업
IoT 디바이스 101
보안 무선 통신
2810
디바이스 ID
2811
관련 ID
2812
앱 또는
브라우저가
설치된 사용자
디바이스
135
관련 ID
디바이스 ID
프로비저닝됨
88798876
77890100
예
06064091
77360110
아니오
66780092
77900029
아니오
디바이스
데이터베이스
2851

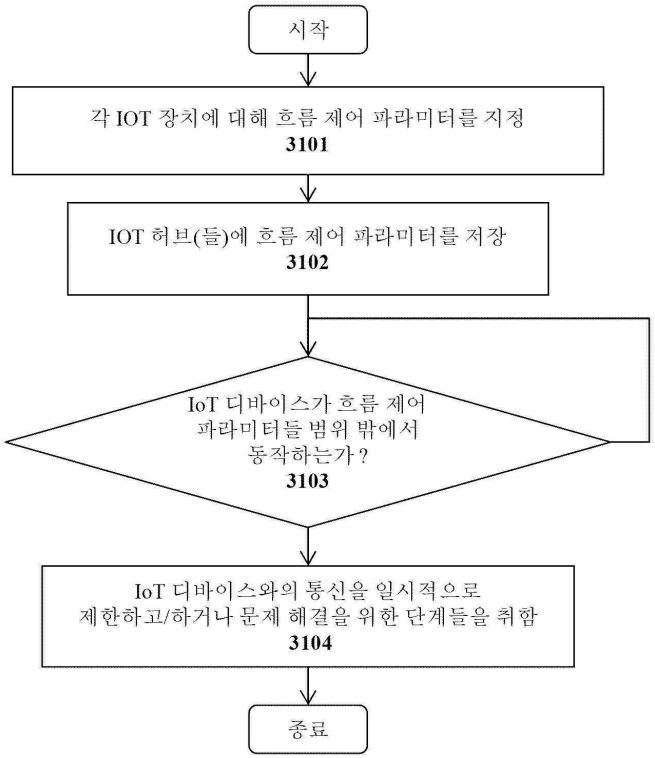
도면29


시작
디바이스 ID와 접속 ID 간의 관련성을 생성
2901
관련성을 IOT 서비스 상에 저장
2902
사용자가 새로운 IOT 디바이스를 구입;
관련 ID를 포함하는 바코드를 스캔
2903
관련 ID를 IOT 서비스로 전송
2904
관련 ID를 사용하여 디바이스 ID를 식별
2905
IOT 디바이스를 디바이스 ID로 프로비저닝
2906
종료

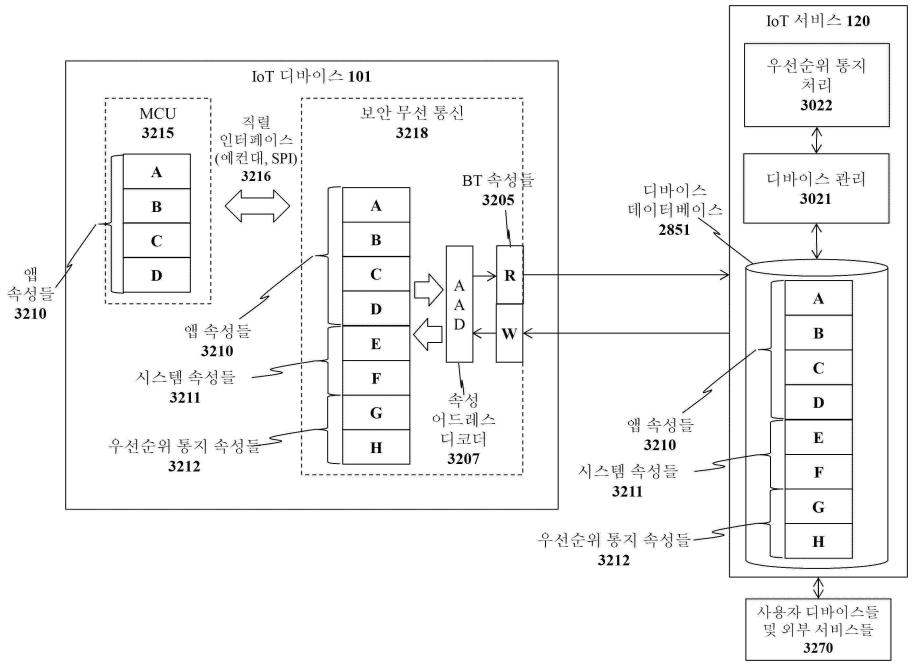
도면30


IoT 디바이스 101
보안 무선 통신
2810
흐름 제어
파라미터들
3015
IoT 디바이스 102
보안 무선 통신
3030
흐름 제어
파라미터들
3031
IoT 디바이스 103
보안 무선 통신
3040
흐름 제어
파라미터들
3041
IoT 허브 110
흐름 제어 관리
2811
디바이스별 흐름
제어 파라미터들
3010
IoT 서비스 120
디바이스 관리
3021
디바이스별 흐름
제어 파라미터들
3020
디바이스
데이터베이스
2851

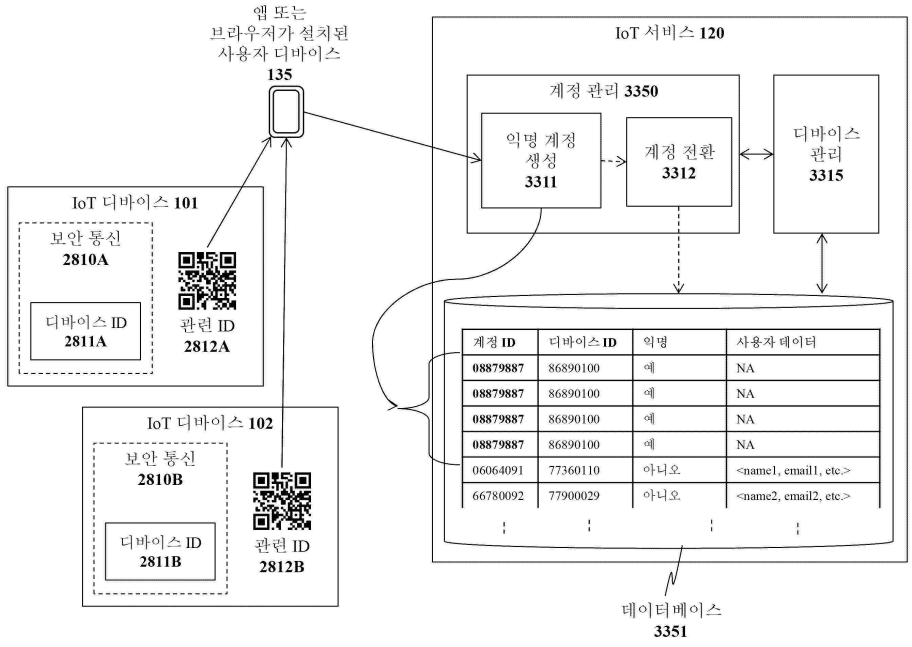
도면31


시작
각 IOT 장치에 대해 흐름 제어 파라미터를 지정
3101
IOT 허브(들)에 흐름 제어 파라미터를 저장
3102
IoT 디바이스가 흐름 제어
파라미터들 범위 밖에서
동작하는가?
3103
IoT 디바이스와의 통신을 일시적으로
제한하고/하거나 문제 해결을 위한 단계들을 취함
3104
종료

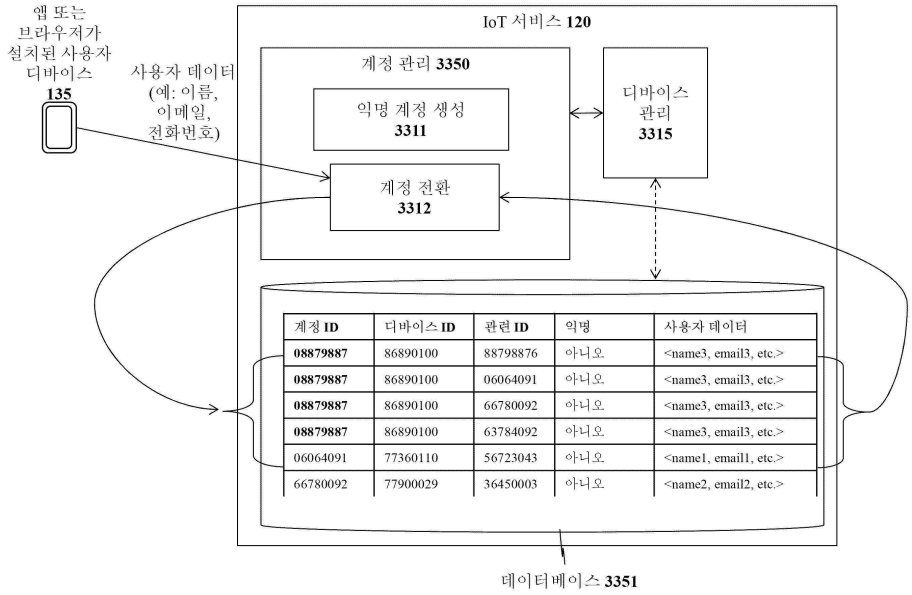
도면32


IoT 서비스 120
우선순위 통지 처리
3022
디바이스 관리
3021
디바이스 데이터베이스
2851
A
B
C
D
E
F
G
H
앱 속성들
3210
시스템 속성들
3211
우선순위 통지 속성들
3212
사용자 디바이스들 및 외부 서비스들
3270
IoT 디바이스 101
MCU
3215
직렬 인터페이스 (에컨대, SPI)
3216
보안 무선 통신
3218
BT 속성들
3205
A
D
R
W
속성 어드레스 디코디
3207

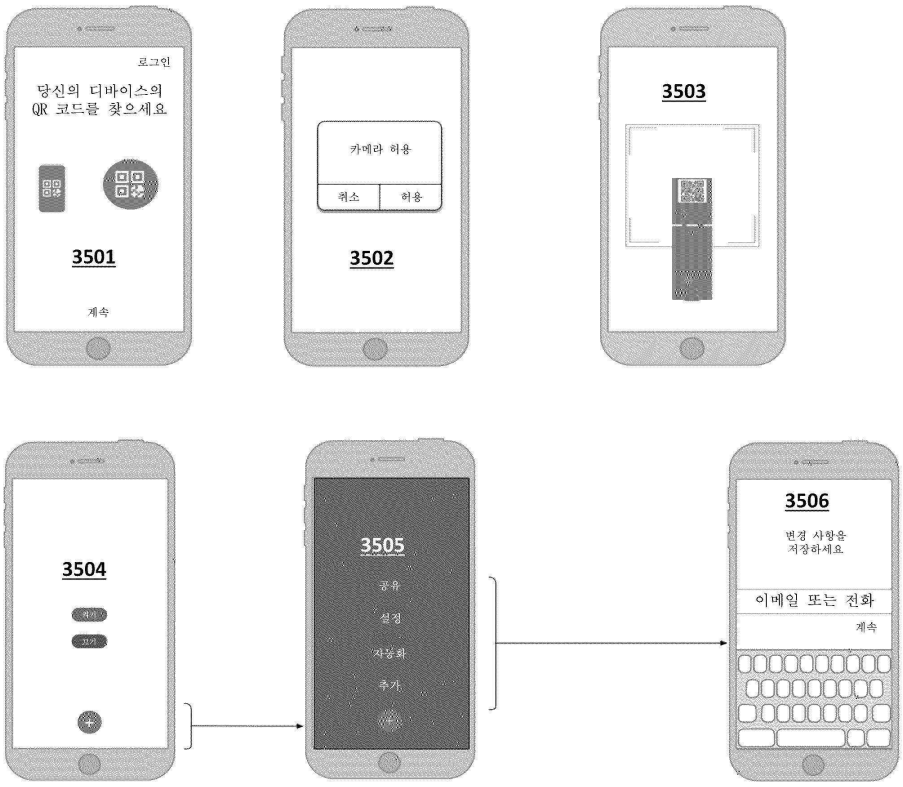
도면33


앱 또는 브라우저가 설치된 사용자 디바이스
135
IoT 서비스 120
계정 관리 3350
익명 계정 생성
3311
계정 전환
3312
디바이스 관리
3315
IoT 디바이스 101
보안 통신
2810A
디바이스 ID
2811A
관련 ID
2812A
IoT 디바이스 102
보안 통신
2810B
디바이스 ID
2811B
관련 ID
2812B
계정 ID
디바이스 ID
익명
사용자 데이터
08879887
86890100
예
NA
06064091
77360110
아니오
<name1, email1, etc.>
66780092
77900029
아니오
<name2, email2, etc.>
데이터베이스
3351

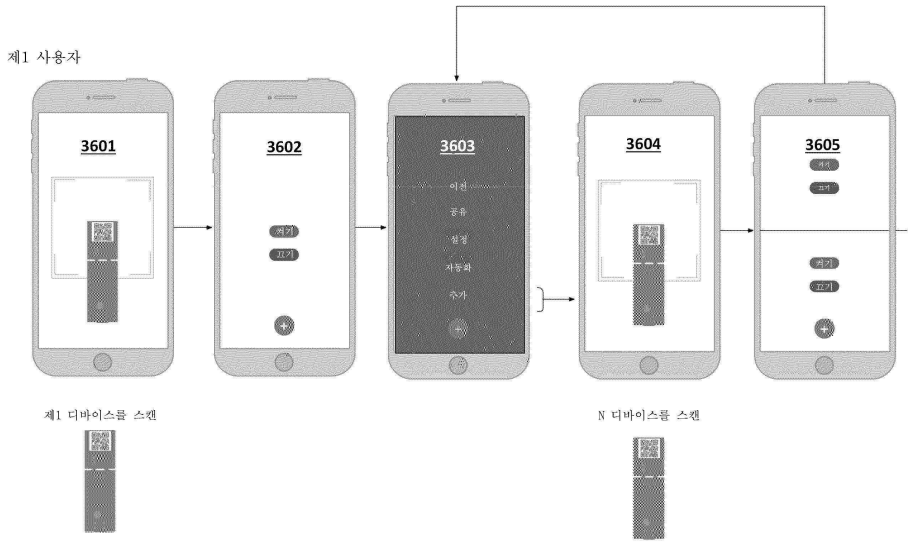
도면34


웹 또는
브라우저가
설치된 사용자
디바이스
135
사용자 데이터
(예: 이름,
이메일,
전화번호)
IoT 서비스 120
계정 관리 3350
익명 계정 생성
3311
계정 전환
3312
디바이스
관리
3315
계정 ID
디바이스 ID
관련 ID
익명
사용자 데이터
08879887 86890100 88798876 아니오 <name3, email3, etc.>
08879887 86890100 06064091 아니오 <name3, email3, etc.>
08879887 86890100 66780092 아니오 <name3, email3, etc.>
08879887 86890100 63784092 아니오 <name3, email3, etc.>
06064091 77360110 56723043 아니오 <name1, email1, etc.>
66780092 77900029 36450003 아니오 <name2, email2, etc.>
데이터베이스 3351

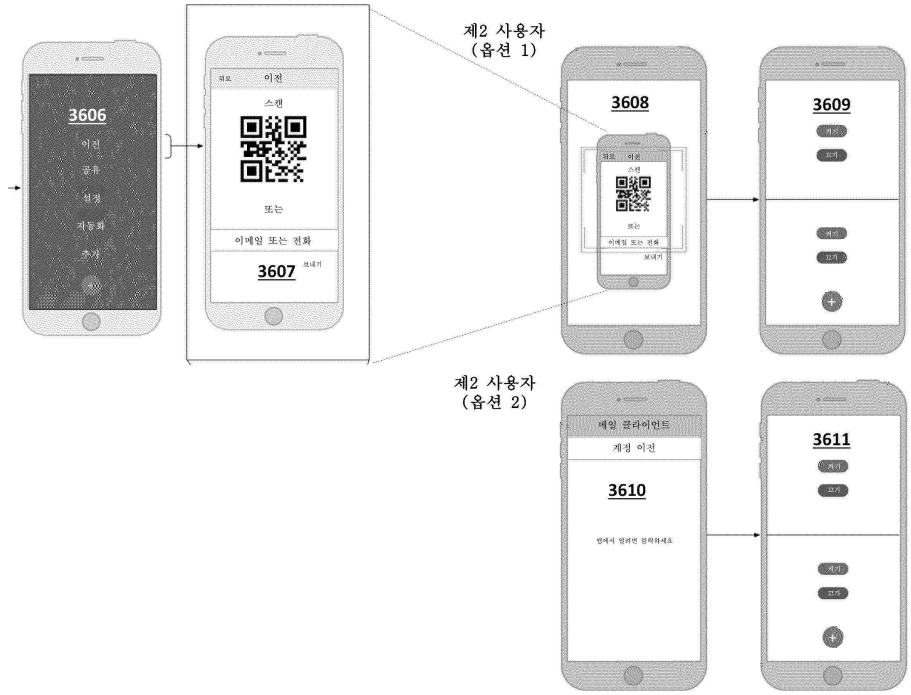
도면35


로그인
당신의 디바이스의
QR 코드를 찾으세요
3501
계속
카메라 허용
취소
허용
3502
3503
3504
3505
공유
설정
자동화
추가
3506
변경 사항을
저장하세요
이메일 또는 전화
계속

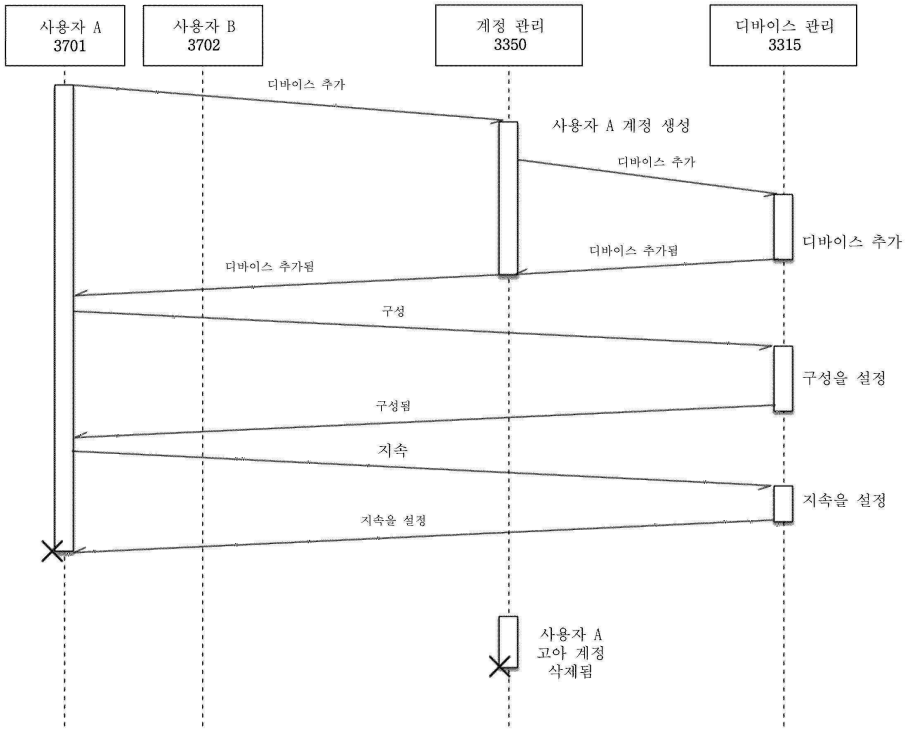
도면36a


제1 사용자
3601
3602
3603
3604
3605
제1 디바이스를 스캔
N 디바이스를 스캔

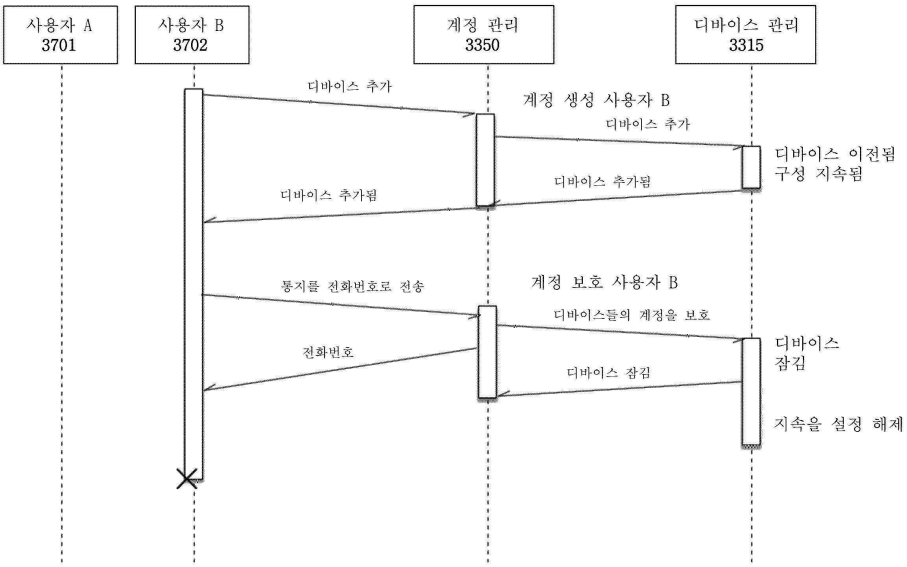
도면36b


3606
3607
제2 사용자
(옵션 1)
3608
3609
제2 사용자
(옵션 2)
3610
3611

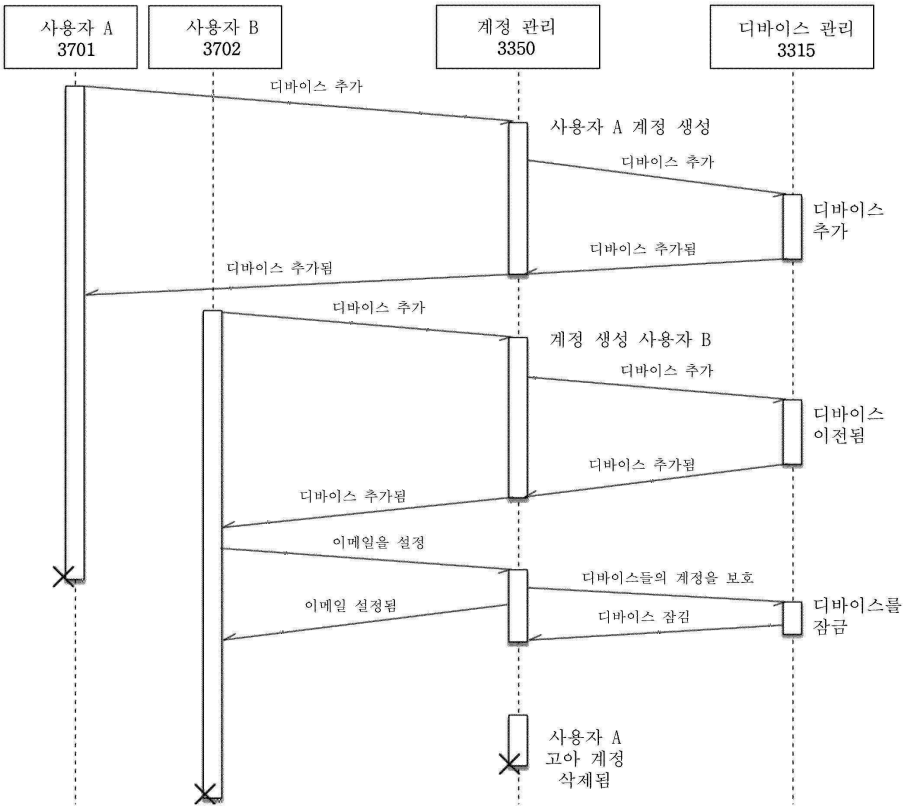
도면37a


사용자 A
3701
사용자 B
3702
계정 관리
3350
디바이스 관리
3315
디바이스 추가
사용자 A 계정 생성
디바이스 추가
디바이스 추가
디바이스 추가됨
디바이스 추가됨
구성
구성을 설정
구성됨
지속
지속을 설정
지속을 설정
사용자 A
고아 계정
삭제됨

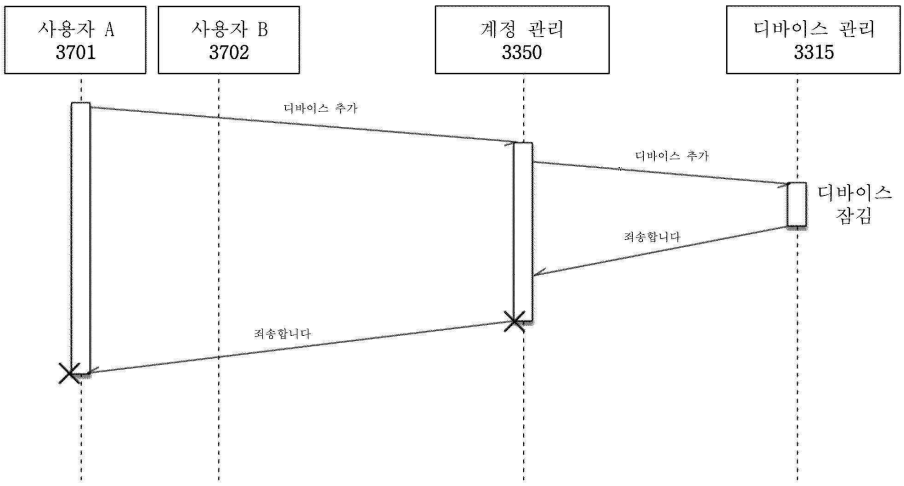
도면37b


사용자 A
3701
사용자 B
3702
계정 관리
3350
디바이스 관리
3315
디바이스 추가
계정 생성 사용자 B
디바이스 추가
디바이스 이전됨
구성 지속됨
디바이스 추가됨
디바이스 추가됨
통지를 전화번호로 전송
계정 보호 사용자 B
디바이스들의 계정을 보호
디바이스
잠김
전화번호
디바이스 잠김
지속을 설정 해제

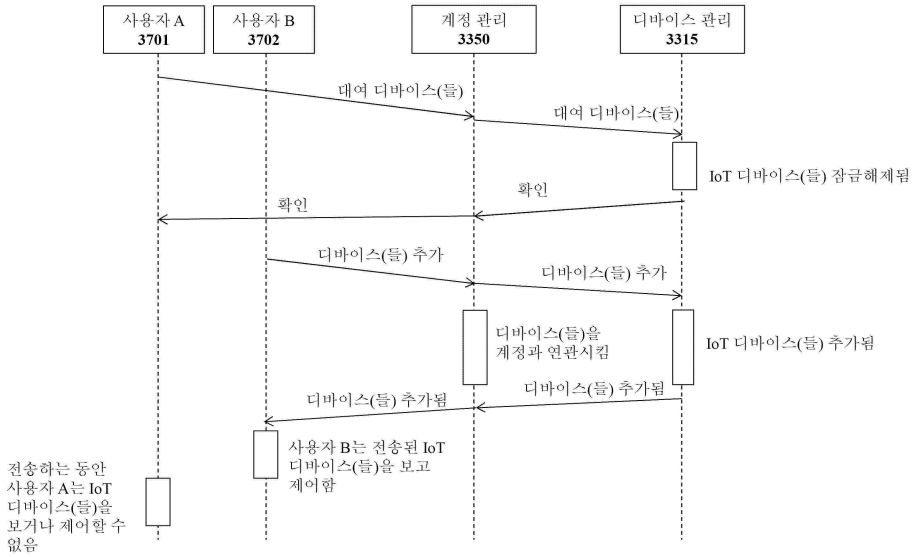
도면38a


사용자 A
3701
사용자 B
3702
계정 관리
3350
디바이스 관리
3315
디바이스 추가
사용자 A 계정 생성
디바이스 추가
디바이스
추가
디바이스 추가됨
디바이스 추가됨
디바이스 추가
계정 생성 사용자 B
디바이스 추가
디바이스
이전됨
디바이스 추가됨
디바이스 추가됨
이메일을 설정
디바이스들의 계정을 보호
디바이스를
잠금
이메일 설정됨
디바이스 잠김
사용자 A
고아 계정
삭제됨

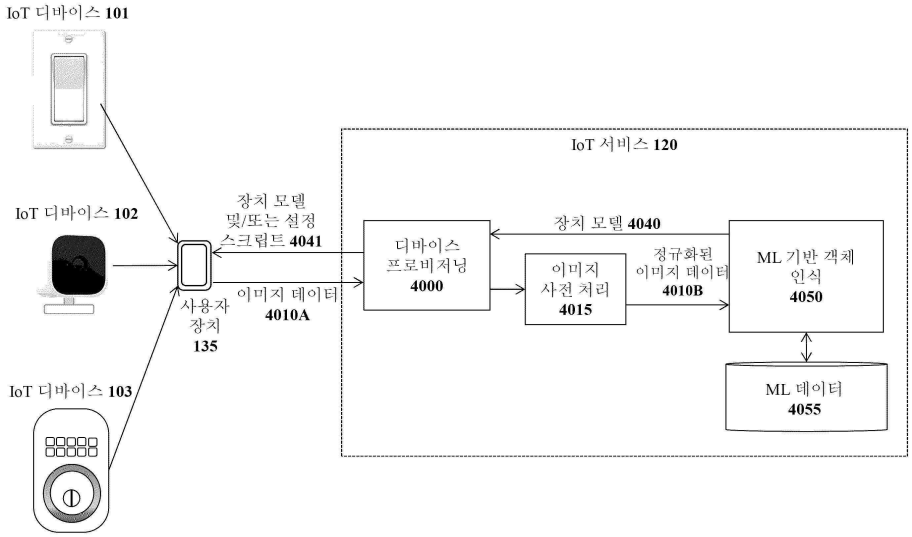
도면38b


사용자 A
3701
사용자 B
3702
계정 관리
3350
디바이스 관리
3315
디바이스 추가
디바이스 추가
디바이스
잠김
죄송합니다
죄송합니다

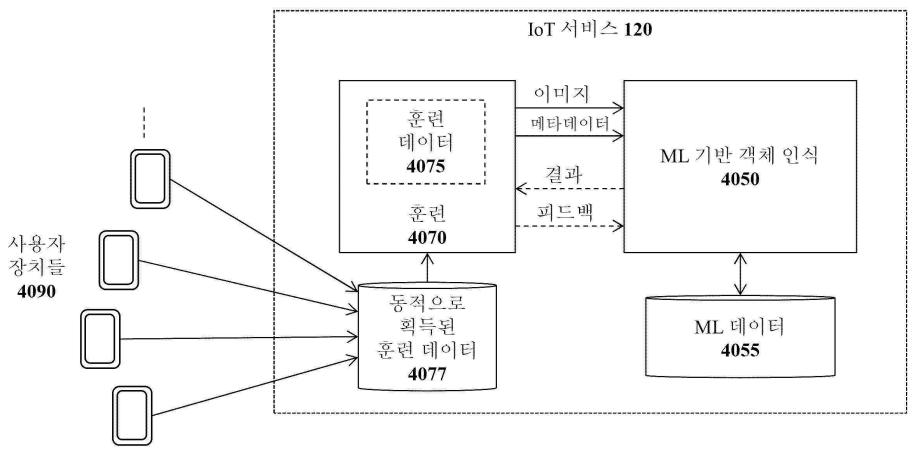
도면39



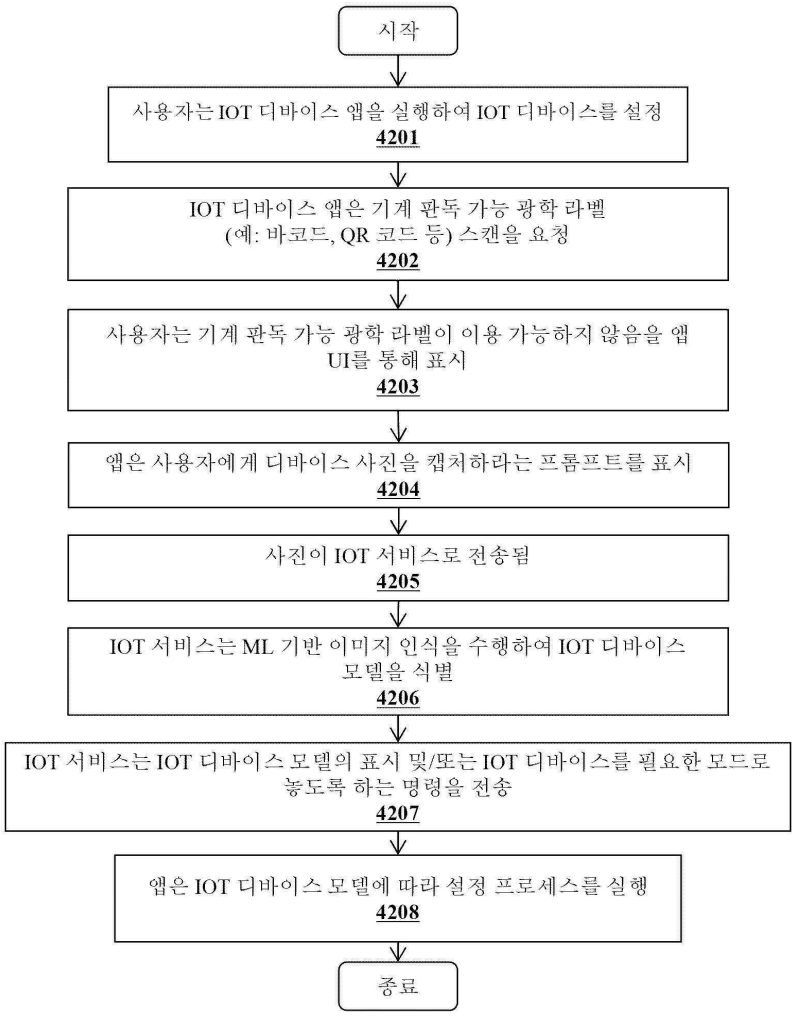

도면40



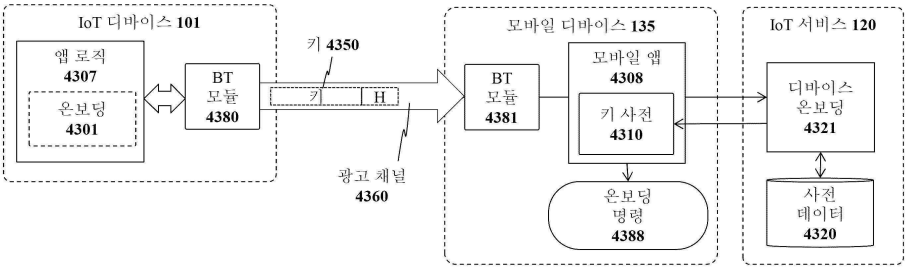

도면41



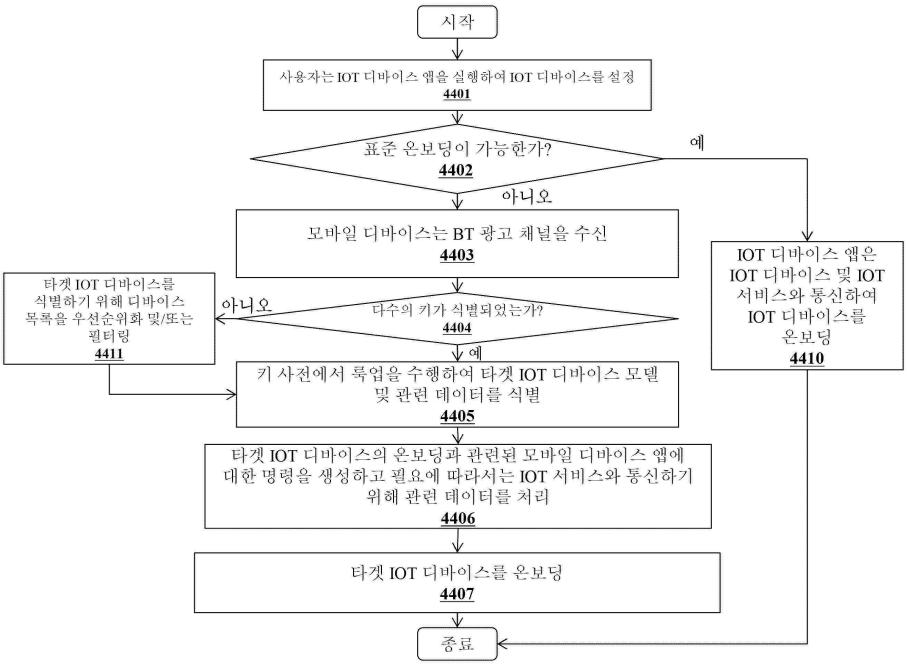
도면42


시작
사용자는 IOT 디바이스 앱을 실행하여 IOT 디바이스를 설정
4201
IOT 디바이스 앱은 기계 판독 가능 광학 라벨
(예: 바코드, QR 코드 등) 스캔을 요청
4202
사용자는 기계 판독 가능 광학 라벨이 이용 가능하지 않음을 앱
UI를 통해 표시
4203
앱은 사용자에게 디바이스 사진을 캡처하라는 프롬프트를 표시
4204
사진이 IOT 서비스로 전송됨
4205
IOT 서비스는 ML 기반 이미지 인식을 수행하여 IOT 디바이스
모델을 식별
4206
IOT 서비스는 IOT 디바이스 모델의 표시 및/또는 IOT 디바이스를 필요한 모드로
놓도록 하는 명령을 전송
4207
앱은 IOT 디바이스 모델에 따라 설정 프로세스를 실행
4208
종료

도면43


IoT 디바이스 101
앱 로직
4307
온보딩
4301
BT
모듈
4380
키 4350
키
H
광고 채널
4360
모바일 디바이스 135
BT
모듈
4381
모바일 앱
4308
키 사전
4310
온보딩
명령
4388
IoT 서비스 120
디바이스
온보딩
4321
사전
데이터
4320

도면44


시작
사용자는 IOT 디바이스 앱을 실행하여 IOT 디바이스를 설정
4401
표준 온보딩이 가능한가?
4402
예
아니오
모바일 디바이스는 BT 광고 채널을 수신
4403
IOT 디바이스 앱은 IOT 디바이스 및 IOT 서비스와 통신하여 IOT 디바이스를 온보딩
4410
타겟 IOT 디바이스를 식별하기 위해 디바이스 목록을 우선순위화 및/또는 필터링
4411
아니오
다수의 키가 식별되었는가?
4404
예
키 사전에서 룩업을 수행하여 타겟 IOT 디바이스 모델 및 관련 데이터를 식별
4405
타겟 IOT 디바이스의 온보딩과 관련된 모바일 디바이스 앱에 대한 명령을 생성하고 필요에 따라서는 IOT 서비스와 통신하기 위해 관련 데이터를 처리
4406
타겟 IOT 디바이스를 온보딩
4407
종료