(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년10월29일
(11) 등록번호 10-2723973
(24) 등록일자 2024년10월25일

(51) 국제특허분류(Int. Cl.)
H04L 9/08 (2006.01) *G06F 21/60* (2013.01)
H04L 65/40 (2022.01) *H04L 9/40* (2022.01)
H04W 12/02 (2021.01) *H04W 4/70* (2018.01)
H04W 84/14 (2009.01)
(52) CPC특허분류
H04L 9/0841 (2013.01)
G06F 21/606 (2013.01)
(21) 출원번호 10-2018-7002323
(22) 출원일자(국제) 2016년07월01일
심사청구일자 2021년06월25일
(85) 번역문제출일자 2018년01월24일
(65) 공개번호 10-2018-0025903
(43) 공개일자 2018년03월09일
(86) 국제출원번호 PCT/US2016/040819
(87) 국제공개번호 WO 2017/007725
국제공개일자 2017년01월12일
(30) 우선권주장
14/791,371 2015년07월03일 미국(US)
14/791,373 2015년07월03일 미국(US)
(56) 선행기술조사문헌
KR1020130118940 A*
Alfred J. Menezes 외 2명, Handbook of Applied Cryptography, CRC Press (1996.)*
Wireless Transport Layer Security: Version 06-Apr-2001, WAP Forum (2001.)*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
어페로, 인크.
미국, 94022, 캘리포니아주, 로스앨토스, 엘 카미노 리얼 4970, 스위트 210
(72) 발명자
브리트, 조
미국 94022 캘리포니아주 로스앨토스 엘 카미노 리얼 4970 스위트 210
자카리아, 오마르
미국 94022 캘리포니아주 로스앨토스 엘 카미노 리얼 4970 스위트 210
지머맨, 스코트
미국 94022 캘리포니아주 로스앨토스 엘 카미노 리얼 4970 스위트 210
(74) 대리인
특허법인(유)남아이피그룹, 특허법인 남앤남

전체 청구항 수 : 총 14 항 심사관 : 양종필

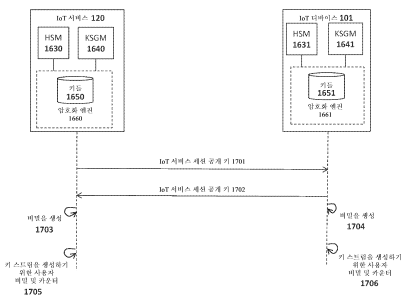
(54) 발명의 명칭 **사물 인터넷(IoT) 시스템에서 보안 통신 채널을 설정하기 위한 장치 및 방법**

(57) 요 약

IoT 디바이스들과 IoT 서비스 사이의 보안 통신을 위한 장치 및 방법이 설명된다. 예를 들어, 시스템의 일 실시예는 IoT 허브 또는 모바일 사용자 디바이스를 통해 IoT 디바이스와의 통신을 설정하기 위한 사물 인터넷(IoT) 서비스; 서비스 공개 키 및 서비스 비공개 키를 생성하기 위한 키 생성 로직을 포함하는, IoT 서비스 상의 제1

(뒷면에 계속)

대 표 도



암호화 엔진; 디바이스 공개 키 및 디바이스 비공개 키를 생성하기 위한 키 생성 로직을 포함하는, IoT 디바이스 상의 제2 암호화 엔진을 포함하며; 제1 암호화 엔진은 서비스 공개 키를 제2 암호화 엔진으로 송신하고, 제2 암호화 엔진은 디바이스 공개 키를 제1 암호화 엔진으로 송신하며; 제1 암호화 엔진은 디바이스 공개 키 및 서비스 비공개 키를 사용하여 비밀을 생성하고; 제2 암호화 엔진은 서비스 공개 키 및 디바이스 비공개 키를 사용하여 동일한 비밀을 생성하고; 일단 비밀이 생성되면, 제1 암호화 엔진 및 제2 암호화 엔진은 비밀을 사용하여 또는 비밀로부터 도출된 데이터 구조를 사용하여 제1 암호화 엔진과 제2 암호화 엔진 사이에서 송신되는 데이터 패킷들을 암호화하고 해독한다.

(52) CPC특허분류
H04L 63/0428 (2013.01)
H04L 63/062 (2013.01)
H04L 67/12 (2022.05)
H04L 9/0877 (2013.01)
H04W 12/02 (2021.01)
H04W 4/70 (2018.02)
H04W 84/14 (2013.01)

명 세 서

청구범위

청구항 1

시스템으로서,

IoT 허브 또는 모바일 사용자 디바이스를 통해 IoT 디바이스와의 통신을 설정하기 위한 사물 인터넷(IoT) 서비스;

서비스 공개 키 및 서비스 비공개 키를 생성하기 위한 키 생성 로직을 포함하는, 상기 IoT 서비스 상의 제1 암호화 회로;

디바이스 공개 키 및 디바이스 비공개 키를 생성하기 위한 키 생성 로직을 포함하는, 상기 IoT 디바이스 상의 제2 암호화 회로

을 포함하며,

상기 제1 암호화 회로는 상기 서비스 공개 키를 상기 제2 암호화 회로로 송신하고, 상기 제2 암호화 회로는 상기 디바이스 공개 키를 상기 제1 암호화 회로로 송신하며,

상기 제1 암호화 회로는 상기 디바이스 공개 키 및 상기 서비스 비공개 키를 사용하여 비밀을 생성하고,

상기 제2 암호화 회로는 상기 서비스 공개 키 및 상기 디바이스 비공개 키를 사용하여 동일한 비밀을 생성하고,

일단 상기 비밀이 생성되면, 상기 제1 암호화 회로 및 상기 제2 암호화 회로는 상기 비밀로부터 도출된 데이터 구조들을 사용하여 상기 제1 암호화 회로와 상기 제2 암호화 회로 사이에서 송신되는 데이터 패킷들을 암호화하고 해독하고,

상기 비밀로부터 도출된 상기 데이터 구조들은 상기 제1 암호화 회로에 의해 생성되는 제1 키 스트림 및 상기 제2 암호화 회로에 의해 생성되는 제2 키 스트림을 포함하고,

상기 제1 암호화 회로는 각각의 데이터 패킷이 상기 제2 암호화 회로로 송신되는 것에 응답하여 상기 제1 암호화 회로와 관련된 제1 카운터를 증가시키고, 상기 제2 암호화 회로는 각각의 데이터 패킷이 상기 제1 암호화 회로로 송신되는 것에 응답하여 상기 제2 암호화 회로와 관련된 제2 카운터를 증가시키고, 그리고

상기 제1 암호화 회로는 상기 제1 카운터의 현재 카운터 값 및 상기 비밀을 사용하여 상기 제1 키 스트림을 생성하고, 상기 제2 암호화 회로는 상기 제2 카운터의 현재 카운터 값 및 상기 비밀을 사용하여 상기 제2 키 스트림을 생성하는, 시스템.

청구항 2

제1항에 있어서, 상기 키 생성 로직은 하드웨어 보안 모듈(HSM)을 포함하는, 시스템.

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

제1항에 있어서, 상기 제1 암호화 회로는 상기 제1 카운터의 현재 카운터 값 및 상기 비밀을 사용하여 상기 제1 키 스트림을 생성하기 위한 타원 곡선 방법(ECM) 모듈을 포함하고, 상기 제2 암호화 회로는 상기 제2 카운터의

현재 카운터 값 및 상기 비밀을 사용하여 상기 제2 키 스트림을 생성하기 위한 ECM 모듈을 포함하는, 시스템.

청구항 7

제1항에 있어서, 상기 제1 암호화 회로는 상기 제1 키 스트림을 사용하여 제1 데이터 패킷을 암호화하여 제1 암호화된 데이터 패킷을 생성하고, 상기 제1 암호화된 데이터 패킷을 상기 제1 카운터의 현재 카운터 값과 함께 상기 제2 암호화 회로로 송신하는, 시스템.

청구항 8

제7항에 있어서, 상기 제2 암호화 회로는 상기 제1 카운터의 상기 현재 카운터 값 및 상기 비밀을 사용하여 상기 제1 키 스트림을 생성하고, 상기 제1 키 스트림을 사용하여 상기 제1 암호화된 데이터 패킷을 해독하는, 시스템.

청구항 9

제7항에 있어서, 상기 제1 데이터 패킷을 암호화하는 것은 상기 제1 키 스트림을 상기 제1 데이터 패킷과 XOR하여 상기 제1 암호화된 데이터 패킷을 생성하는 것을 포함하는, 시스템.

청구항 10

제8항에 있어서, 상기 IoT 디바이스는 상기 IoT 디바이스를 상기 IoT 허브 또는 상기 모바일 사용자 디바이스에 통신 가능하게 결합하기 위한 블루투스 저에너지(BTLE) 통신 인터페이스를 포함하며, 상기 IoT 허브 또는 상기 모바일 사용자 디바이스는 인터넷을 통해 상기 IoT 서비스에 통신 가능하게 결합되는, 시스템.

청구항 11

컴퓨터 구현 방법으로서,

IoT 허브 또는 모바일 사용자 디바이스를 통해 사물 인터넷(IoT) 서비스와 IoT 디바이스 사이의 통신을 설정하는 단계;

상기 IoT 서비스 상의 제1 암호화 회로의 키 생성 로직에 의해 서비스 공개 키 및 서비스 비공개 키를 생성하는 단계;

상기 IoT 디바이스 상의 제2 암호화 회로의 키 생성 로직에 의해 디바이스 공개 키 및 디바이스 비공개 키를 생성하는 단계;

상기 서비스 공개 키를 상기 제1 암호화 회로로부터 상기 제2 암호화 회로로 송신하고, 상기 디바이스 공개 키를 상기 제2 암호화 회로로부터 상기 제1 암호화 회로로 송신하는 단계;

상기 디바이스 공개 키 및 상기 서비스 비공개 키를 사용하여 비밀을 생성하는 단계;

상기 서비스 공개 키 및 상기 디바이스 비공개 키를 사용하여 동일한 비밀을 생성하는 단계; 및

상기 비밀로부터 도출된 데이터 구조들을 사용하여 상기 제1 암호화 회로와 상기 제2 암호화 회로 사이에서 송신되는 데이터 패킷들을 암호화하고 해독하는 단계를 포함하고,

상기 비밀로부터 도출된 상기 데이터 구조들은 상기 제1 암호화 회로에 의해 생성되는 제1 키 스트림 및 상기 제2 암호화 회로에 의해 생성되는 제2 키 스트림을 포함하고,

상기 제1 암호화 회로는 각각의 데이터 패킷이 상기 제2 암호화 회로로 송신되는 것에 응답하여 상기 제1 암호화 회로와 관련된 제1 카운터를 증가시키고, 상기 제2 암호화 회로는 각각의 데이터 패킷이 상기 제1 암호화 회로로 송신되는 것에 응답하여 상기 제2 암호화 회로와 관련된 제2 카운터를 증가시키고, 그리고

상기 제1 암호화 회로는 상기 제1 카운터의 현재 카운터 값 및 상기 비밀을 사용하여 상기 제1 키 스트림을 생성하고, 상기 제2 암호화 회로는 상기 제2 카운터의 현재 카운터 값 및 상기 비밀을 사용하여 상기 제2 키 스트림을 생성하는, 컴퓨터 구현 방법.

청구항 12

제11항에 있어서, 상기 키 생성 로직은 하드웨어 보안 모듈(HSM)을 포함하는, 컴퓨터 구현 방법.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

제11항에 있어서, 상기 제1 암호화 회로는 상기 제1 카운터의 현재 카운터 값 및 상기 비밀을 사용하여 상기 제1 키 스트림을 생성하기 위한 타원 곡선 방법(ECM) 모듈을 포함하고, 상기 제2 암호화 회로는 상기 제2 카운터의 현재 카운터 값 및 상기 비밀을 사용하여 상기 제2 키 스트림을 생성하기 위한 ECM 모듈을 포함하는, 컴퓨터 구현 방법.

청구항 17

제11항에 있어서, 상기 제1 암호화 회로는 상기 제1 키 스트림을 사용하여 제1 데이터 패킷을 암호화하여 제1 암호화된 데이터 패킷을 생성하고, 상기 제1 암호화된 데이터 패킷을 상기 제1 카운터의 현재 카운터 값과 함께 상기 제2 암호화 회로로 송신하는, 컴퓨터 구현 방법.

청구항 18

제17항에 있어서, 상기 제2 암호화 회로는 상기 제1 카운터의 상기 현재 카운터 값 및 상기 비밀을 사용하여 상기 제1 키 스트림을 생성하고, 상기 제1 키 스트림을 사용하여 상기 제1 암호화된 데이터 패킷을 해독하는, 컴퓨터 구현 방법.

청구항 19

제17항에 있어서, 상기 제1 데이터 패킷을 암호화하는 것은 상기 제1 키 스트림을 상기 제1 데이터 패킷과 XOR하여 상기 제1 암호화된 데이터 패킷을 생성하는 것을 포함하는, 컴퓨터 구현 방법.

청구항 20

제18항에 있어서, 상기 IoT 디바이스는 상기 IoT 디바이스를 상기 IoT 허브 또는 상기 모바일 사용자 디바이스에 통신 가능하게 결합하기 위한 블루투스 저에너지(BTLE) 통신 인터페이스를 포함하며, 상기 IoT 허브 또는 상기 모바일 사용자 디바이스는 인터넷을 통해 상기 IoT 서비스에 통신 가능하게 결합되는, 컴퓨터 구현 방법.

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 일반적으로 컴퓨터 시스템의 분야에 관한 것이다. 더 구체적으로, 본 발명은 IoT 시스템에서 보안 통신 채널을 설정하기 위한 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] "사물 인터넷"은 인터넷 기반구조 내의 고유하게 식별 가능한 임베디드 디바이스들의 상호접속을 지칭한다. 궁극적으로, 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.

[0003] IoT 개발 및 채택은 접속성, 전력, 및 표준화의 결여에 관련된 이슈로 인해 느렸다. 예를 들어, IoT 개발 및 채택에 대한 하나의 장애물은 개발자가 새로운 IoT 디바이스 및 서비스를 설계 및 제공하도록 허용하기 위한 어떠한 표준 플랫폼도 존재하지 않는다는 것이다. IoT 시장에 진입하기 위해, 개발자는 원하는 IoT 구현을 지원하는 데 요구되는 네트워크 프로토콜 및 기반구조, 하드웨어, 소프트웨어 및 서비스를 포함하여 처음부터 끝까지 전체 IoT 플랫폼을 설계해야 한다. 결과적으로, IoT 디바이스의 각각의 제공자는 IoT 디바이스를 설계하고 접속하기 위한 독점적인 기술을 사용하여, 다수의 타입의 IoT 디바이스의 채택을 최종 사용자에게 부담이 되게 한다. IoT 채택에 대한 다른 장애물은 IoT 디바이스들을 접속하고 전력공급하는 것과 연관된 어려움이다. 예를 들어, 냉장고, 차고 도어 오프너, 환경 센서, 집 보안 센서/제어기 등과 같은 기기를 접속시키는 것은, 각각의 접속된 IoT 디바이스에 전력공급하기 위한 전기 소스를 요구하고, 그러한 전기 소스는 종종 편리하게 위치되어 있지 않다.

[0004] 존재하는 다른 문제는 블루투스 LE와 같은 IoT 디바이스들을 상호접속하는 데 사용되는 무선 기술들이 일반적으로 단거리 기술들이라는 점이다. 따라서, IoT 구현을 위한 데이터 수집 허브가 IoT 디바이스의 범위 밖에 있는 경우, IoT 디바이스는 데이터를 IoT 허브로 전송할 수 없을 것이다(그리고 그 반대도 마찬가지다). 그 결과, IoT 디바이스가 범위 밖에 있는 IoT 허브(또는 다른 IoT 디바이스)에 데이터를 제공하는 것을 가능하게 하는 기술들이 필요하다.

도면의 간단한 설명

[0005] 아래의 도면들과 관련된 아래의 상세한 설명으로부터 본 발명의 더 양호한 이해가 얻어질 수 있다.

도 1a 및 도 1b는 IoT 시스템 아키텍처의 상이한 실시예들을 예시한다.

도 2는 본 발명의 일 실시예에 따른 IoT 디바이스를 예시한다.

도 3은 본 발명의 일 실시예에 따른 IoT 허브를 예시한다.

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

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

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

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

도 8은 본 발명의 일 실시예에 따른 방법을 예시한다.

도 9a는 프로그램 코드 및 데이터 업데이트들이 IoT 디바이스에 제공되는 실시예를 예시한다.

도 9b는 프로그램 코드 및 데이터 업데이트들이 IoT 디바이스에 제공되는 방법의 실시예를 예시한다.

도 10은 보안 아키텍처의 일 실시예의 고레벨 도면을 예시한다.

도 11은 IoT 디바이스 상에 키를 저장하기 위해 가입자 식별 모듈(SIM)이 사용되는 아키텍처의 일 실시예를 예시한다.

도 12a는 IoT 디바이스가 바코드 또는 QR 코드를 사용하여 등록되는 일 실시예를 예시한다.

도 12b는 바코드 또는 QR 코드를 사용하여 페어링이 수행되는 일 실시예를 예시한다.

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

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

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

도 16a 및 도 16b는 IoT 서비스와 IoT 디바이스 사이에서 데이터를 암호화하기 위한 본 발명의 상이한 실시예들을 예시한다.

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

도 18은 본 발명의 일 실시예에 따른 패킷 구조를 예시한다.

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

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

도 21은 커맨드 패킷들을 사용하는 예시적인 트랜잭션 시퀀스를 예시한다.

도 22는 본 발명의 일 실시예에 따른 방법을 예시한다.

도 23a 내지 도 23c는 본 발명의 일 실시예에 따른 보안 페어링을 위한 방법을 예시한다.

발명을 실시하기 위한 구체적인 내용

[0006] 아래의 설명에서, 설명의 목적으로, 아래에 설명되는 본 발명의 실시예의 완전한 이해를 제공하기 위해 다수의 특정 상세들이 기재된다. 그러나, 본 발명의 실시예는 이러한 특정 상세들 중 일부 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 블록도 형태로 도시된다.

[0007] 본 발명의 일 실시예는 새로운 IoT 디바이스 및 애플리케이션을 설계 및 구축하기 위해 개발자에 의해 이용될 수 있는 사물 인터넷(IoT) 플랫폼을 포함한다. 특히, 일 실시예는 IoT 디바이스들이 그것을 통해 인터넷에 결합되는 미리 정의된 네트워킹 프로토콜 스택 및 IoT 허브를 포함한 IoT 디바이스들을 위한 기반 하드웨어/소프트웨어 플랫폼을 포함한다. 부가적으로, 일 실시예는 IoT 허브들 및 접속된 IoT 디바이스들이 그것을 통해 아래에 설명되는 바와 같이 액세스 및 관리될 수 있는 IoT 서비스를 포함한다. 부가적으로, IoT 플랫폼의 일 실시예는 IoT 서비스, 허브 및 접속된 디바이스들에 액세스하고 그들을 구성하기 위한 (예를 들어, 클라이언트 디

바이스 상에서 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 레버리징할 수 있다.

[0008] 도 1a는 본 발명의 실시예가 구현될 수 있는 아키텍처 플랫폼의 개요를 예시한다. 특히, 예시된 실시예는 그 자체가 인터넷(220)을 통해 IoT 서비스(120)에 통신 가능하게 결합된 중앙 IoT 허브(110)에 로컬 통신 채널(130)을 통해 통신 가능하게 결합된 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) IoT 허브(110)에 초기에 페어링될 수 있다. 일 실시예에서, IoT 서비스(120)는 각각의 사용자의 IoT 디바이스로부터 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 포함하면, 데이터베이스(122)는 IoT 디바이스(101 내지 105)에 의해 수집된 데이터를 저장하도록 계속 업데이트될 수 있다. 이어서, 데이터베이스(122)에 저장된 데이터는 사용자의 디바이스(135) 상에 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 수집된 정보를 IoT 허브(110)를 통해 IoT 서비스(120), 사용자 디바이스(135) 및/또는 외부 웹사이트(130)에 제공하기 위한 다양한 타입들의 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 아래에서 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)에 대한 접속을 설정하기 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 결합시키는 WiFi 액세스 포인트 또는 라우터(116)를 통해 WiFi 접속을 설정하기 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 구성하도록 허용하기 위해 사용자 디바이스(135) 상에서 실행되는 IoT 앱 또는 웹 애플리케이션을 포함한다. 일 실시예에서, 앱 또는 웹 애플리케이션은 IoT 기능을 그의 사용자 기반에 제공하도록 웹사이트(130)의 운영자에 의해 설계될 수 있다. 예시된 바와 같이, 웹사이트는 각각의 사용자에 관련된 계정 기록을 포함하는 사용자 데이터베이스(131)를 유지할 수 있다.

[0013] 도 1b는 복수의 IoT 허브(110, 111, 190)에 대한 추가의 접속 옵션들 예시한다. 이러한 실시예에서, 단일 사용자는 단일 사용자 구내(premises)(180)(예를 들어, 사용자의 집 또는 사업체)에 현장 설치된 다수의 허브(110, 111)를 가질 수 있다. 이것은 예를 들어 IoT 디바이스(101 내지 105) 모두를 접속시키기 위해 필요한 무선 범위를 확장시키기 위해 행해질 수 있다. 표시된 바와 같이, 사용자가 다수의 허브(110, 111)를 갖는 경우, 그것들은 로컬 통신 채널(예를 들어, Wifi, 이더넷, 전력 라인 네트워킹 등)을 통해 접속될 수 있다. 일 실시예에서, 허브(110, 111) 각각은 (도 1b에 명시적으로 도시되지 않은) 셀룰러(115) 또는 WiFi(116) 접속을 통해 IoT 서비스(120)에 대한 직접 접속을 설정할 수 있다. 대안적으로 또는 부가적으로, IoT 허브(110)와 같은 IoT 허브들 중 하나는 (IoT 허브(110)와 IoT 허브(111)를 연결하는 점선에 의해 표시된 바와 같이) IoT 허브(111)와 같은, 사용자 구내(180) 상의 다른 IoT 허브들 모두에 접속성 및/또는 로컬 서비스를 제공하는 "마스터" 허브로서 동작할 수 있다. 예를 들어, 마스터 IoT 허브(110)는 IoT 서비스(120)에 대한 직접 접속을 설정하기 위한 유일한 IoT 허브일 수 있다. 일 실시예에서, "마스터" IoT 허브(110)에만 IoT 서비스(120)에 대한 접속을 설정하기 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 로컬식으로 일부 데이터 요청들을 서비스하는) IoT 서비스(120) 사이에서 교환되는 데이터에 대해 필터링 동작을 수행하기 위한 부가적인 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 1b는 또한 제2 사용자 구내(181)에 설치된 IoT 허브(190)를 도시한다. 사실상 제한되지 않는 수의 그러한 IoT 허브(190)가 전세계의 사용자 구내에서 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 2에 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 비-휘발성 메모리일 수 있다. 일 실시예에서, 비-휘발성 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 애플리케이션-특정 세트를 정의하는 애플리케이션 프로그램 코드(203), 및 IoT 디바이스(101)의 애플리케이션 개발자에 의해 이용될 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 인에이블하기 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)는 저전력 마이크로제어기(200) 내에 통합될 수 있다. 그러나, 본 발명의 기본 원리는 임의의 특정 통신 프로토콜로 제한되지 않는다.

[0019] 도 2에 도시된 특정 실시예는 또한 사용자 입력을 수신하고 사용자 입력을 저전력 마이크로제어기에 제공하기 위한 복수의 입력 디바이스 또는 센서(210)를 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, 비-충전 가능 코인 셀 배터리가 사용된다. 그러나, 대안적인 실시예에서, 통합된 재충전 가능 배터리가 사용될 수 있다(예를 들어, IoT 디바이스를 AC 전력 공급부(도시되지 않음)에 접속시킴으로써 재충전 가능함).

[0021] 오디오를 생성하기 위한 스피커(205)가 또한 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 사용자가 입력 디바이스(210)를 통해 선택을 입력할 때 언어 피드백을 최종 사용자에게 제공하기 위한 오디오의 디지털 샘플링된 단편(snippet)을 포함할 수 있다.

[0022] 일 실시예에서, 하나 이상의 다른/대안적인 I/O 디바이스 또는 센서(250)가, IoT 디바이스(101)가 그것을 위해

설계되는 특정 애플리케이션에 기초하여 IoT 디바이스(101) 상에 포함될 수 있다. 예를 들어, 온도, 압력, 습도 등을 측정하기 위해 환경 센서가 포함될 수 있다. IoT 디바이스가 보안 디바이스로서 사용되는 경우 보안 센서 및/또는 도어록 오프너가 포함될 수 있다. 물론, 이들 예는 단지 예시의 목적으로 제공된다. 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 인터페이싱하기 위해 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, 키는 가입자 식별 모듈(SIM)에서 보안될 수 있다.

[0024] 일 실시예에서, IoT 디바이스가 사실상 어떠한 전력도 소비하고 있지 않은 초 저전력 상태로부터 그 IoT 디바이스를 웨이크하기 위해 웨이크업 수신기(207)가 포함된다. 일 실시예에서, 웨이크업 수신기(207)는 도 3에 도시된 바와 같이 IoT 허브(110) 상에 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 매우 낮은 전력 상태로부터 IoT 디바이스(101)를 웨이크할 필요가 있을 때 에너지가 라디오 주파수 신호를 통해 송신기(307)로부터 수신기(207)로 송신된다. 에너지 전달 때문에, IoT 디바이스(101)는 그것이 그것의 저전력 상태에 있을 때 사실상 어떠한 전력도 소비하지 않도록 구성될 수 있는데, 왜냐하면 (디바이스가 네트워크 신호를 통해 어웨이크되도록 허용하는 네트워크 프로토콜에서 그러한 바와 같이) 그것이 허브로부터의 신호를 계속 "청취"할 필요가 없기 때문이다. 오히려, IoT 디바이스(101)의 마이크로제어기(200)는 송신기(307)로부터 수신기(207)로 전기적으로 송신된 에너지를 사용함으로써 사실상 전력 차단된 후에 웨이크 업하도록 구성될 수 있다.

[0025] 도 3에 예시된 바와 같이, IoT 허브(110)는 또한 프로그램 코드 및 데이터(305)를 저장하기 위한 메모리(317), 및 프로그램 코드를 실행하고 데이터를 처리하기 위한 마이크로제어기와 같은 하드웨어 로직(301)을 포함한다. 광역 네트워크(WAN) 인터페이스(302) 및 안테나(310)가 IoT 허브(110)를 셀룰러 서비스(115)에 결합시킨다. 대안적으로, 위에서 언급된 바와 같이, IoT 허브(110)는 또한 근거리 네트워크 통신 채널을 설정하기 위한 WiFi 인터페이스(및 WiFi 안테나) 또는 이더넷 인터페이스와 같은 로컬 네트워크 인터페이스(도시되지 않음)를 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, 키는 가입자 식별 모듈(SIM)에서 보안될 수 있다.

[0026] 로컬 통신 인터페이스(303) 및 안테나(311)가 IoT 디바이스(101 내지 105) 각각과 로컬 통신 채널을 설정한다. 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 설정하기 위한 임의의 특정 프로토콜로 제한되지 않는다. 도 3에서 별개의 유닛으로서 예시되지만, WAN 인터페이스(302) 및/또는 로컬 통신 인터페이스(303)는 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 디바이스와 페어링하도록 허용하기 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)에 통신되는 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 통신될 수 있다. 대안적인 실시예에서, 고유 ID 코드는 IoT 디바이스 상에 자기적으로 임베딩되며, IoT 허브는 IoT 디바이스(101)가 IoT 허브(110)로부터 수 인치 이내로 이동될 때 코드를 검출하기 위한 라디오 주파수 ID(RFID) 또는 근거리장 통신(NFC) 센서와 같은 자기 센서를 갖는다.

[0028] 일 실시예에서, 일단 고유 ID가 통신되면, 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 확인되면, 일 실시예에서, IoT 허브(110)는 IoT 디바이스(101)를 페어링하고, (언급된 바와 같이, 비-휘발성 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 구동하는 조직은 개발자가 새로운 IoT 서비스를 용이하게 설계하도록 허용하기 위해 IoT 허브(110) 및 기본적인 하드웨어/소프트웨어 플랫폼을 제공할 수 있다. 특히, IoT 허브(110)에 더하여, 개발자는 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 단지 IoT 디바이스에 대한 입력 및 출력만을 지정할 필요가 있는 그래픽 설계 인터페이스를 포함한다. IoT 디바이스(101)가 허브(110) 및 서비스(120)에 접속하도록 허용하는 통신 스택(201)을 포함한 모든 네트워킹 코드가 이미 개발자를 위해 제 위치에 있다. 부가적으로, 일 실시예에서, SDK는 또한 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 기초하여 미세조정될 수 있다. 예를 들어, 연속적인 양방향 스트림을 갖는 것이 타당하지 않을 수 있는 일부 경우에, 간단한 요청/응답 프로토콜이 필요할 경우 정보를 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, 그것은 IoT 서비스(120)로부터 업데이트를 자동적으로 다운로드 및 설치할 수 있다. 그것은 먼저 업데이트된 코드를 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 다운로드되고 IoT 디바이스(101 내지 105) 각각에 푸시 아웃될 수 있다. 그 후, 각각의 IoT 디바이스(101 내지 105)는 IoT 허브에 대해 위에서 설명된 것과 유사한 방식으로 업데이트를 적용하고 업데이트의 결과를 IoT 허브(110)에 다시 보고할 수 있다. 업데이트가 성공적이면, IoT 허브(110)는 그것의 메모리로부터 업데이트를 삭제하고 (예를 들어, 그것이 각각의 IoT 디바이스에 대한 새로운 업데이트를 계속 체크할 수 있도록) 각각의 IoT 디바이스 상에 설치된 코드의 최신 버전을 기록할 수 있다.

[0032] 일 실시예에서, IoT 허브(110)는 A/C 전력을 통해 전력공급된다. 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 4a는 IoT 시스템을 사용하여 범용 원격 제어 동작을 수행하기 위한 본 발명의 일 실시예를 예시한다. 특히, 이 실시예에서, IoT 디바이스들(101 내지 103)의 세트에는 (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 송신하기 위한 적외선(IR) 및/또는 라디오 주파수(RF) 블라스터들(401 내지 403)이 각각 탑재된다. 도 4a에 도시된 실시예에서, IoT 디바이스들(101 내지 103)에는 또한 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 기초하여 에어컨/히터(430)를 제어하기 위한 온도 및/또는 습도 센서일 수 있다. 이 실시예에서, 에어컨/히터(430)는 원격 제어 디바이스(전형적으로 그 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 IR/RF 블라스터(401)를 통해 에어컨/히터에 커맨드를 송신하여 (예를 들어, 에어컨을 턴오프하거나 히터를 턴온함으로써) 온도를 높일 수 있다. 커맨드는 IoT 허브(110) 상의 데이터베이스(413) 내에 저장된 필요한 원격 제어 코드를 포함할 수 있다. 대안적으로 또는 추가적으로, IoT 서비스(421)는 지정된 사용자 선호 및 저장된 제어 코드(422)에 기초하여 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, 발광체(431)의 현재 밝기 레벨을 제어할 것이다(예를 들어, 현재 밝기가 너무 낮으면 조명을 높이거나 현재 밝기가 너무 높으면 조명을 낮추거나; 단순히 발광체를 턴온 또는 턴오프함).

[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) 및/또는 텔레비전에 의해 생성된 광에 기초하여(예를 들어, 지정된 스펙트럼 내의 광을 측정함으로써) 텔레비전이 온 또는 오프 상태인지를 검출하기 위한 광센서일 수 있다. 대안적으로, 센서(406)는 검출된 온도에 기초하여 오디오 장비가 온 또는 오프 상태인지를 검출하기 위해 시청각 장비에 접속된 온도 센서를 포함할 수 있다. 다시 한번, 사용자 디바이스(135)를 통한 사용자 입력에 응답하여, 제어 로직(412)은 IoT 디바이스(103)의 IR 블라스터(403)를 통해 시청각 장비에 커맨드를 송신할 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 결합되는 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) 상의 데이터베이스(413) 및/또는 IoT 서비스(120) 상의 데이터베이스(422)에 저장된다. 도 4b에 예시된 바와 같이, 제어 코드는 IoT 서비스(120) 상에서 유지되는 상이한 장비들에 대한 제어 코드들(422)의 마스터 데이터베이스로부터 IoT 허브(110)에 제공될 수 있다. 최종 사용자는 사용자 디바이스(135) 상에서 실행되는 앱 또는 브라우저를 통해 제어될 전자(또는 다른) 장비의 타입을 지정할 수 있으며, 그에 응답하여 IoT 허브 상의 원격 제어 코드 학습 모듈(491)은 IoT 서비스(120) 상의 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) 상의 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 코드가 학습되면, 이들은 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 극히 작은 폼 팩터를 가지며, 양면 테이프, 작은 못, 자석 부착 등을 사용하여 그들 각자의 전자 장비(430 내지 432) 상에 또는 그 부근에 부착될 수 있다. 에어컨(430)과 같은 장비의 제어를 위해, 센서(404)가 집 안의 주위 온도를 정확하게 측정할 수 있도록 IoT 디바이스(101)를 충분히 멀리 배치하는 것이 바람직할 것이다(예를 들어, 에어컨 상에 직접 IoT 디바이스를 배치하는 것은 에어컨이 작동 중일 때 너무 낮거나 히터가 작동 중일 때 너무 높을 온도 측정치를 초래할 것임). 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. 텍스트 메시지 및/또는 앱 특유 통지일 수 있는 통지는 이어서 사용자의 모바일 디바이스(135)의 디스플레이 상에 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 기초하여 검출될 수 있음), 센서(406)가 시청각 장비(430)가 켜져 있다는 것을 지시하거나, 센서(405)가 발광체가 켜져 있다는 것을 지시하는 경우, 사용자가

시청각 장비(432) 및/또는 발광체(431)를 턴오프하기를 원하는지를 묻는 통지가 사용자에게 전송될 수 있다. 임의의 장비 타입에 대해 동일한 타입의 통지가 전송될 수 있다.

[0043] 사용자가 통지를 수신하면, 그/그녀는 사용자 디바이스(135) 상의 앱 또는 브라우저를 통해 전자 장비(430 내지 432)를 원격 제어할 수 있다. 일 실시예에서, 사용자 디바이스(135)는 터치 스크린 디바이스이고, 앱 또는 브라우저는 장비(430 내지 432)를 제어하기 위해 사용자가 선택할 수 있는 버튼을 갖는 리모트 컨트롤의 이미지를 표시한다. 통지를 수신하면, 사용자는 그래픽 리모트 컨트롤을 열고 다양한 상이한 장비를 턴오프하거나 조정할 수 있다. IoT 서비스(120)를 통해 접속되는 경우, 사용자의 선택은 IoT 서비스(120)로부터 IoT 허브(110)로 전송될 수 있으며, 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. 전술한 바와 같이 원하는 온도, 밝기 레벨 및 볼륨 레벨을 유지하는 것 이외에, 제어 로직(412)은 소정 조건이 검출되면 전자 장비를 자동으로 턴오프할 수 있다. 예를 들어, 제어 로직(412)이 사용자가 집에 없고 에어컨이 기능하고 있지 않다는 것을 검출하면, 그것은 자동으로 에어컨을 턴오프할 수 있다. 유사하게, 사용자가 집에 없고, 센서(406)가 시청각 장비(430)가 켜져 있음을 나타내거나 센서(405)가 발광체가 켜져 있음을 나타내면, 제어 로직(412)은 IR/RF 블라스터(403, 402)를 통해 커맨드를 자동 송신하여, 시청각 장비 및 발광체를 각각 턴오프할 수 있다.

[0045] 도 5는 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 송신한다. 스토브가 (예를 들어, 측정된 온도에 기초하여) 임계 기간을 초과하여 켜져 있는 것으로 검출되면, 제어 로직(512)은 사용자에게 스토브(530)가 켜져 있음을 알리는 통지를 최종 사용자의 디바이스(135)로 송신할 수 있다. 또한, 일 실시예에서, IoT 디바이스(104)는, 사용자로부터 지시를 수신하는 것에 응답하여 또는 (제어 로직(512)이 사용자에 의해 그렇게 하도록 프로그래밍된 경우) 자동으로, 스토브를 턴오프하기 위한 제어 모듈(501)을 포함할 수 있다. 일 실시예에서, 제어 로직(501)은 스토브(530)에 대한 전기 또는 가스를 차단하는 스위치를 포함한다. 그러나, 다른 실시예에서, 제어 로직(501)은 스토브 자체 내에 통합될 수 있다.

[0046] 도 5는 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 턴오프되지 않았음을 나타내는 경우) 최종 사용자에게 통지를 송신할 수 있다. 도 5에 도시되지 않지만, IoT 디바이스(105)에는 자동으로 그리고/또는 사용자 입력에 응답하여 (예를 들어, 전기/가스를 스위치 오프함으로써) 세탁기/건조기(531)를 턴오프하는 제어 모듈이 또한 탑재될 수 있다.

[0047] 일 실시예에서, 제어 로직 및 스위치를 갖는 제1 IoT 디바이스는 사용자의 집 안의 모든 전력을 턴오프하도록 구성될 수 있고, 제어 로직 및 스위치를 갖는 제2 IoT 디바이스는 사용자의 집 안의 모든 가스를 턴오프하도록 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 턴오프하기 위한 커맨드를 전송할 수 있다. 대안적으로, IoT 허브(110) 및/또는 IoT 서비스(120) 내의 제어 로직(512)은 그러한 상황에서 전기 또는 가스를 자동으로 턴오프하도록 구성될 수 있다.

[0048] 일 실시예에서, IoT 허브(110) 및 IoT 서비스(120)는 주기적인 간격으로 통신한다. IoT 서비스(120)가 (예를 들어, 지정된 지속 기간 동안 IoT 허브로부터 요청 또는 응답을 수신하지 못함으로써) IoT 허브(110)에 대한 접속이 손실된 것을 검출하면, (예를 들어, 텍스트 메시지 또는 앱 특유 통지를 전송함으로써) 이러한 정보를 최종 사용자의 디바이스(135)로 통신할 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 디바이스

는 데이터를 IoT 허브로 송신할 수 없을 것이다(그리고 그 반대도 마찬가지다).

[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. 일단 접속되면, IoT 디바이스는 IoT 허브에 제공될 필요가 있는 임의의 데이터를 모바일 디바이스로 송신할 수 있으며, 이어서 모바일 디바이스는 데이터를 IoT 허브로 전송한다.

[0052] 도 6에 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 통신할 수 있는 임의의 형태의 IoT 디바이스를 포함할 수 있다. 예를 들어, IoT 디바이스(601)는 냉장고에서 이용 가능한 음식 아이템들, 음식 아이템들을 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 6에 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 접속성을 주기적으로 또는 계속적으로 체크할 수 있다. (예를 들어, 사용자가 냉장고의 부근 내에서 이동하는 결과로서) 접속이 설정되면, IoT 디바이스(601) 상의 임의의 수집된 데이터(605)가 모바일 디바이스(611) 상의 임시 데이터 저장소(615)로 자동 송신된다. 일 실시예에서, IoT 디바이스(601) 및 모바일 디바이스(611)는 BTLE와 같은 저전력 무선 표준을 사용하여 로컬 무선 통신 채널을 설정한다. 그러한 경우, 모바일 디바이스(611)는 공지된 페어링 기술들을 이용하여 초기에 IoT 디바이스(601)와 페어링될 수 있다.

[0055] 일단 데이터가 임시 데이터 저장소로 전송되면, (예를 들어, 사용자가 IoT 허브(110)의 범위 내에서 걸을 때) 일단 IoT 허브(110)와의 통신이 설정되면 모바일 디바이스(611)는 데이터를 송신할 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 고전력 통신 채널)을 사용하여 데이터를 IoT 허브(110)에 제공할 수 있다.

[0056] 범위 밖에 있는 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 7에 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 디바이스 상의 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 전송하며, IoT 허브는 데이터를 중앙 데이터 저장소(413)에 저장한다.

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

[0058] 모바일 디바이스(611)보다는, 일 실시예에서, IoT 디바이스(601)는 IoT 허브(110)의 범위 내에 위치되는 하나 이상의 중개 IoT 디바이스와의 무선 접속을 설정하도록 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 7에는 간략함을 위해 단일 모바일 디바이스(611)만이 예시되지만, 일 실시예에서, 상이한 사용자들의 다수의 그러한 모바일 디바이스가 IoT 디바이스(601)와 통신하도록 구성될 수 있다. 더욱이, 동일한 기술들이 다수의 다른 IoT 디바이스에 대해 구현될 수 있으며, 이에 의해 집 전체에 걸친 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, 냉장고 안에서 사용되는 경우, 수집된 데이터(605)는 냉장고 옆을 지나가는 각각의 사용자, 냉장고를 여는 각각의 사용자, 및 각각의 사용자에 의해 소비되는 특정 음식 아이템들의 아이덴티티를 포함할 수 있다. 상이한 타입의 데이터가 다른 타입의 IoT 디바이스들로부터 수집될 수 있다. 이러한 데이터를 사용하여, 시스템은 예를 들어 어느 사용자가 옷을 세탁하는지, 어느 사용자가 주어진 날에 TV를 시청하는지, 각각의 사용자가 자러 가고 일어나는 시간 등을 결정할 수 있다. 이어서, 이러한 크라우드 소싱된(crowd-sourced) 데이터 모두가 IoT 허브의 데이터 저장소(413) 내에 집계되고/되거나 외부 서비스 또는 사용자에게 전송될 수 있다.

[0061] 본 명세서에서 설명되는 기술들의 다른 유리한 응용은 도움을 필요로 할 수 있는 노년 사용자들을 모니터링하는 것이다. 이러한 응용을 위해, 모바일 디바이스(611)는 사용자의 집의 상이한 방들에서 정보를 수집하기 위해 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 데이터를 하나 이상의 외부 사용자(예를 들어, 노년 사용자를 돌보는 자식들 또는 다른 개인들)에게 제공할 수 있다. 데이터가 지정된 기간(예를 들어, 12시간) 동안 수집되지 않은 경우, 이것은 노년 사용자가 집 주위에서 이동하지 않았고/않았거나 냉장고를 열지 않았음을 의미한다. 이어서, IoT 허브(110) 또는 IoT 허브에 접속된 외부 서비스가 경고 통지를 이러한 다른 개인들에게 송신하여, 그들이 노년 사용자를 체크해야 한다는 것을 그들에게 알릴 수 있다. 또한, 수집된 데이터(605)는 사용자에 의해 소비되고 있는 음식 및 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 관한 문제가 있는 경우, 수집된 데이터는 교체가 필요한 부품의 지시를 포함할 수 있다. 그러한 경우, 문제 해결의 요청과 함께 통지가 기술자에게 전송될 수 있다. 이어서, 기술자는 필요한 교체 부품을 갖고서 집에 도착할 수 있다.

[0063] 본 발명의 일 실시예에 따른 방법이 도 8에 예시된다. 방법은 상기에 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 음식 아이템 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 접속성을 주기적으로 또는 계속적으로 체크한다. 모바일 디바이스에 대한 접속이 설정된 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] IoT 디바이스들로부터 데이터를 수집하는 것에 더하여, 일 실시예에서, 본 명세서에서 설명되는 기술들은 데이터를 IoT 디바이스들에 업데이트하거나 달리 제공하는 데 사용될 수 있다. 일례가 도 9a에 도시되며, 이 도면은 IoT 디바이스(601)(또는 그러한 IoT 디바이스들의 그룹) 상에 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 모바일 디바이스 또는 컴퓨터를 통해 IoT 디바이스(601)에 대한 구성 옵션들을 지정할 수 있으며, 이어서 이들은 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 들어갈 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, 이어서, IoT 디바이스(601)는 자동화된 업데이트 프로세스에 들어가서, 새로운 프로그램 코드 업데이트들 및/또는 데이터를 설치할 수 있다.

[0066] IoT 디바이스를 업데이트하기 위한 방법이 도 9b에 도시된다. 방법은 상기에 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 이용가능해진다. 901에서, 모바일 디바이스는 IoT 디바이스를 대신하여 프로그램 코드 또는 데이터 업데이트들을 수신 및 저장한다. IoT 디바이스 및/또는 모바일 디바이스는 902에서 접속이 설정되었는지를 결정하기 위해 주기적으로 체크한다. 접속이 설정된 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, 키는 아래에서 논의되는 바와 같이 가입자 식별 모듈(SIM)에서 보안될 수 있다.

[0070] 도 10은 IoT 서비스(120), IoT 허브(110) 및 IoT 디바이스(101, 102) 간의 통신을 암호화하기 위해 공개 키 기반구조(PKI) 기술 및/또는 대칭 키 교환/암호화 기술을 사용하는 고레벨 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 설정될 때, 그것의 공개 키가 IoT 서비스(120)에 제공되고, 새로운 IoT 디바이스(101)가 설정될 때, 그것의 공개 키가 IoT 허브(110) 및 IoT 서비스(120) 둘 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)뿐만 아니라, 키를 사용하여 암호화/해독 동작을 수행하기 위한 보안 로직(1012)을 포함한다. 마지막으로, IoT 서비스(120)는 그 자신의 비공개 키, 다양한 IoT 디바이스 및 IoT 허브의 공개 키를 보안 저장하기 위한 보안 저장소(1021), 및 키를 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 수신할 때, IoT 허브는 (예를 들어, 전술한 바와 같이 마스터 키를 사용하여 서명을 확인함으로써) 그것을 검증할 수 있고, 이어서 그것 내부로부터 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)는 그것이 신뢰 소스로부터 변경되지 않은 메시지를 수신하고 있는지를 검증할 수 있다. 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, 디바이스들은 각각 암호화에 사용되고 서명을 확인하는 데 사용되는 동일한 대칭 키의 사본을 제공받을 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 프

로토콜에 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 원리는 임의의 특정 키 프로비저닝 프로토콜로 제한되지 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 수행 한 다음, 교환된 대칭 키를 사용하여 통신을 암호화할 수 있다. 일 실시예에서, 새로운 대칭 키가 디바이스(101)와 허브(110) 사이에서 그리고 허브(110)와 IoT 서비스(120) 사이에서 주기적으로 교환된다. 일 실시예에서, 새로운 대칭 키가 디바이스(101), 허브(110) 및 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 송신을 위해 재암호화되기 전에 허브 보안 모듈(1012)에서 해독 및 검증될 것이다.

[0076] 일 실시예에서, 허브 보안 모듈(1012) 상의 타협(compromise)을 방지하기 위해, 설치 시에 1회(영구) 설치 키가 디바이스(101)와 서비스(120) 사이에서 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 위해 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 더 어렵게 할 것이다. 그러나, 이것 없이도, 서비스와 디바이스 간의 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 그것의 비공개 키를 사용하여 IoT 허브 패킷을 해독하고 암호화된 IoT 디바이스 패킷을 생성하여, 이것을 관련 IoT 디바이스(101)로 송신한다. 이어서, IoT 디바이스(101)는 그것의 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 사용하여, 각각의 디바이스는 공유 대칭 키를 사용하여 암호화하고 해독할 것이다. 어느 경우에나, 각각의 송신 디바이스는 또한 메시지를 그것의 비공개 키로 서명할 수 있어서, 수신 디바이스는 그것의 진정성을 검증할 수 있다.

[0079] IoT 디바이스(101)로부터 IoT 허브(110)로의 그리고 IoT 서비스(120)로의 통신을 암호화하기 위해 상이한 키 세트가 사용될 수 있다. 예를 들어, 공개/비공개 키 배열을 사용하여, 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) 및 허브(110)는 하나의 대칭 키를 공유할 수 있는 반면, 허브 및 서비스(120)는 상이한 대칭 키를 공유할 수 있다.

[0080] 소정의 특정 상세가 위의 설명에서 기재되지만, 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), IoT 허브(110) 및 IoT 서비스(120) 사이에서 안전하게 교환되는 대칭 키를 사용할 수 있다. 더욱이, 일부 실시예에서, 데이터/커맨드 그 자체가 암호화되는 것이 아니라, 키가 데이터/커맨드(또는 다른 데이터 구조)에 대한 서명을 생성하는 데 사용된다. 이어서, 수신자는 그것의 키를 사용하여 서명을 확인할 수 있다.

[0081] 도 11에 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 처음에, IoT 디바이스(101) 상의 SIM 인터페이스(1100) 내에 안착되는 프로그래밍되지 않은 SIM 카드(1101)와 함께 최종 사용자에

게 제공될 수 있다. 하나 이상의 암호화 키의 세트를 갖도록 SIM을 프로그래밍하기 위해, 사용자는 SIM 인터페이스(500)로부터 프로그래밍 가능 SIM 카드(1101)를 취하여, 그것을 IoT 허브(110) 상의 SIM 프로그래밍 인터페이스(1102) 안에 삽입한다. 이어서, IoT 허브 상의 프로그래밍 로직(1125)은 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 디바이스(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)에 의해 발신 데이터를 암호화하는 데 사용되도록) SIM 카드(1401) 상에 저장할 수 있다. 일단 SIM(1101)이 프로그래밍되면, 새로운 IoT 디바이스(101)는 SIM을 보안 식별자로서 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 11과 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, SIM은 IoT 허브(110)를 통해 최종 사용자에 의해 직접 프로그래밍될 수 있고 프로그래밍의 결과는 IoT 서비스(120)로 안전하게 통신될 수 있다. 결과적으로, 새로운 IoT 디바이스(101)는 온라인 또는 로컬 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 상황에서 전술되지만, 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 실시예가 이동 가능한 SIM 디바이스를 포함하지만, 일 실시예에서, SIM 디바이스는 이동 가능한 것이 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 요구하기보다는, SIM은 최종 사용자에게 배포되기 전에 IoT 디바이스(101) 내에 미리 프로그래밍된다. 이 실시예에서, 사용자가 IoT 디바이스(101)를 설정할 때, 본 명세서에서 설명되는 다양한 기술은 IoT 허브(110)/IoT 서비스(120)와 새로운 IoT 디바이스(101) 사이에서 암호화 키를 안전하게 교환하는 데 사용될 수 있다.

[0085] 예를 들어, 도 12a에 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)에 의해 공개 키를 식별하거나 생성하는 데 사용될 수 있다(예를 들어, 보안 저장소에 이미 저장된 공개 키에 대한 포인터로서 사용될 수 있다). 바코드 또는 QR 코드(601)는 (도 12a에 도시된 바와 같이) 별도의 카드 상에 인쇄될 수 있거나 IoT 디바이스 그 자체 상에 직접 인쇄될 수 있다. 바코드가 인쇄되는 곳에 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 통신될 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 위치

에 관계없이, 일 실시예에서, 보안 로직(1002, 1012)은 소정 타입의 프로그램 코드를 실행하기 위한 보안 실행 환경을 설정하도록 설계된다. 이것은 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 12b는 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. 일단 페어링 코드가 수신되면, 그것은 페어링 데이터(1285)를 포함하는 보안 저장소에 저장되고, IoT 디바이스(101) 및 IoT 허브(110)는 자동적으로 페어링된다. IoT 허브가 이러한 방식으로 새로운 IoT 디바이스와 페어링될 때마다, 그러한 페어링을 위한 페어링 데이터는 보안 저장소(685) 내에 저장된다. 일 실시예에서, 일단 IoT 허브(110)의 로컬 통신 모듈(1280)이 페어링 코드를 수신하면, 그것은 IoT 디바이스(101)와의 로컬 무선 채널을 통한 통신을 암호화하기 위한 키로서 코드를 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 또한 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)를 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 13에 예시되어 있다. 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 수신하고, 1602에서 사용자는 블랭크 SIM 카드를 IoT 허브 안에 삽입한다. 1303에서, 사용자는 하나 이상의 암호화 키의 세트를 갖도록 블랭크 SIM 카드를 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 송신되어, 그것은 IoT 디바이스를 식별하고 IoT 디바이스와의 암호화된 통신을 설정하는 데 사용될 수 있다. 전술한 바와 같이, 일 실시예에서, "SIM" 카드 이외의 프로그래밍 가능 디바이스가 도 13에 도시된 방법에서 SIM 카드와 동일한 기능을 수행하는 데 사용될 수 있다.

[0094] 새로운 IoT 디바이스를 네트워크 안에 통합하는 방법이 도 14에 예시되어 있다. 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 설정되어 키를 교환할 수 있다. 키가 송신되는 방식에 상관없이, 일단 수신되면, 그것은 IoT 허브 디바이스의 보안 키 저장소에 저장된다. 전술한 바와 같이, 보안 엔클레이브(Secure Enclave), 신뢰 실행 기술(TXT) 및/또는 트러스트존과 같은 다양한 보안 실행 기술이 키를 저장하고 보호하기 위해 IoT 허브 상에서 사용될 수 있다. 또한, 803에서, 키는 IoT 서비스로 안전하게 송신되며, IoT 서비스는 그 자신의 보안 키 저장소에 키를

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 통신하는 방법이 도 15에 예시되어 있다. 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. 이어서 그것은 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 각각의 디바이스들 사이에서(예를 들어, 각각의 디바이스와 허브 사이에서 그리고 허브와 서비스 사이에서) 협상될 수 있다. 일단 키 교환이 완료되면, 각각의 송신 디바이스는 데이터를 수신 디바이스로 송신하기 전에 대칭 키를 사용하여 각각의 송신을 암호화 및/또는 서명한다.

[0099] 사물 인터넷(IoT) 시스템에서 보안 통신 채널을 설정하기 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 16a에 예시되고, IoT 허브를 필요로 하지 않는 다른 실시예가 도 16b에 예시된다.

[0101] 먼저 도 16a를 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, 각각의 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 (다른 것들 중에서도) 세션 공개/비공개 키 쌍을 생성하고 그 쌍의 비공개 세션 키에 대한 액세스를 방지하기 위한 하드웨어 보안 모듈(1630, 1631) 및 도출된 비밀을 사용하여 키 스트림을 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 설정하기 위해, 공개/비공개 세션 키 쌍들(1650, 1651)은, 각각, 각각의 암호화 엔진(1660, 1661)에 의해 동일한 비밀을 생성하는 데 사용되며, 이 동일한 비밀은 이어서 SKGM들(1640, 1641)에 의해 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 추가의 상세들이 아래에 제공된다.

[0102] 도 16a에서, 일단 키들(1650, 1651)을 사용하여 비밀이 생성되면, 클리어 트랜잭션(Clear transaction)(1611)에 의해 지시된 바와 같이, 클라이언트는 항상 IoT 서비스(120)를 통해 IoT 디바이스(101)에 메시지들을 전송할 것이다. 본 명세서에서 사용된 바와 같은 "클리어"는 기본 메시지가 본 명세서에서 설명되는 암호화 기술들을 사용하여 암호화되지 않는다는 것을 지시하도록 의도된다. 그러나, 예시된 바와 같이, 일 실시예에서, 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 설정된다. 이어서, 1602에서, IoT 서비스(120) 상의 암호화 엔진(1660)은 생성된 비밀을 사용하여 메시지를 암호화하고 암호화된 메시지를 IoT 허브(110)로 송신한다. 비밀을 사용하여 메시지를 직접 암호화하기보다는, 일 실시예에서, 비밀 및 카운터 값은 각각의 메시지 패킷을 암호화하는 데 사용되는 키 스트림을 생성하는 데 사용된다. 이 실시예의 상세들이 도 17과 관련하여 아래에서 설명된다.

[0103] 예시된 바와 같이, SSL 접속 또는 다른 보안 채널이 IoT 서비스(120)와 IoT 허브(110) 사이에 설정될 수 있다. 1603에서, IoT 허브(110)(일 실시예에서 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 비밀 및 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 측정을 행하고 그 결과를 다시 클라이언트 디바이스(611)에 보고하는 것과 같은 특정 기능을 수행하도록 IoT 디바이스(101)에 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 16b는 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 9b와 관련하여 전술한 실시예들에서와 같이) 클라이언트 디바이스(611)를 통해 일어난다. 이 실시예에서, IoT 디바이스(101)로 메시지를 송신하기 위해, 클라이언트 디바이스(611)는 1611에서 메시지의 암호화되지 않은 버전을 IoT 서비스(120)로 송신한다. 암호화 엔진(1660)은 1612에서 비밀 또는 도출된 키 스트림을 사용하여 메시지를 암호화하고 암호화된 메시지를 다시 클라이언트 디바이스(611)로 송신한다. 이어서, 클라이언트 디바이스(611)는 1613에서 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)에 전송한다. 이어서, 1616에서, 암호화 엔진(1660)은 응답을 해독하고 해독된 응답을 클라이언트 디바이스(611)로 송신한다.

[0107] 도 17은 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 설정할 때마다 수행될 수 있다. 대안적으로, 지정된 기간(예컨대, 하루, 일주일 등) 동안 키 교환이 수행될 수 있고 교환된 세션 키들이 사용될 수 있다. 간략화를 위해 도 17에는 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 쌍의 비공개 세션 키에 대한 액세스를 방지할 수 있다. 유사하게, IoT 디바이스(101) 상의 암호화 엔진은 세션 공개/비공개 키 쌍을 생성하고 쌍의 세션 비공개 키에 대한 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 쌍의 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 설정하도록 허용하는 익명 키 협약인 ECDH(Elliptic Curve Diffie-Hellman) 프로토콜을 사용한다. 일 실시예에서, 이들 기술을 사용하여, 1703에서, IoT 서비스(120)의 암호화 엔진(1660)은 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 포인트 곱셈된다는 것을 의미한다. IoT 디바이스(101) 상의 암호화 엔진(1661)은 IoT 서비스 세션 공개 키가 IoT 디바이스 세션 비공개 키와 포인트 곱셈되는, 공식 '비밀 = IoT 서비스 세션 공개 키 * IoT 디바이스 세션 비공개 키'에 따라 비밀을 생성한다. 마침내, IoT 서비스(120) 및 IoT 디바이스(101)는 둘 모두가 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 하

드웨어 모듈에 의존한다.

[0110] 일단 비밀이 결정되면, 그것은 암호화 엔진들(1660, 1661)에 의해 데이터를 직접 암호화 및 해독하는 데 사용될 수 있다. 대안적으로, 일 실시예에서, 암호화 엔진들(1660, 1661)은 각각의 데이터 패킷을 암호화/해독하기 위해 비밀을 사용하여 새로운 키 스트림을 생성하도록(즉, 새로운 키 스트림 데이터 구조가 각각의 패킷에 대해 생성됨) KSGM들(1640, 1641)에 커맨드들을 전송한다. 특히, 키 스트림 생성 모듈(1640, 1641)의 일 실시예는 GCM(Galois/Counter Mode)을 구현하며, 이 모드에서는 카운터 값이 각각의 데이터 패킷에 대해 증가되고, 키 스트림을 생성하기 위해 비밀과 조합되어 사용된다. 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)이 새로운 키 스트림을 생성하고 다음 키 스트림의 생성을 위해 카운터 값을 증가시키게 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 암호화된 데이터 패킷과 함께 카운터 값을 IoT 서비스(120)로 송신한다. 이어서, IoT 서비스 상의 암호화 엔진(1660)은 KSGM(1640)과 통신하며, 이 KSGM은 수신된 카운터 값 및 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 증가되고, 새로운 키 스트림을 생성하기 위해 비밀과 함께 사용된다. 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 카운터 값 및 비밀을 사용하여 데이터 패킷을 해독하는 데 사용되는 동일한 키 스트림을 생성하는 KSGM(1641)과 통신한다. 따라서, 이 실시예에서, 암호화 엔진들(1660, 1661)은 그들 자신의 카운터 값들을 사용하여 키 스트림을 생성하여서 데이터를 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 동일한 카운터 값이 한 번을 초과하여 수신되는지를 검출하는 시퀀싱 로직을 포함한다. 카운터 값이 비순차적으로 수신되는 경우, 또는 동일한 카운터 값이 한 번을 초과하여 수신되는 경우, 이것은 재생 공격이 시도되고 있음을 지시할 수 있다. 이에 응답하여, 암호화 엔진들(1660, 1661)은 통신 채널로부터 접속 해제될 수 있고/있거나, 보안 경보를 생성할 수 있다.

[0113] 도 18은 4 바이트 카운터 값(1800), 가변 크기의 암호화된 데이터 필드(1801) 및 6 바이트 태그(1802)를 포함하는 본 발명의 일 실시예에서 사용되는 예시적인 암호화된 데이터 패킷을 예시한다. 일 실시예에서, 태그(1802)는 (일단 해독되면) 해독된 데이터를 확인하기 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 신뢰의 루트를 포함하고, (예를 들어, 본 명세서에 설명되는 IoT 시스템들을 구현하는 조직의 제어하에) 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 IoT 서비스 키들 및 IoT 디바이스 키들을 인증하기 위해 후속 사용될 수 있는 그 자신의 공장 키 쌍을 할당받는다. 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/디바이스 공개 키들은 도 16a 및 도 16b와 관련하여 위에서 설명한 "세션" 공개/비공개 키들과 동일하지 않음에 유의한다. 전술한 세션 공개/비공개 키들은 일시적인 반면(즉, 서비스/디바이스 세션에 대해 생성됨), IoT 서비스/디바이스 키 쌍들은 영구적이다(즉, 공장에서 생성됨).

[0116] 마스터 키들, 공장 키들, 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 추가적인 계층들을 제

공한다:

[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 16b와 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 검증한다

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

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

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

[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 검증한다

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

[0151] 3. IoT 디바이스의 세션 공개 키를 저장한다

[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 검증한다

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

[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 생성한다

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

[0160] • 난수 2 바이트 값을 생성하고 저장한다

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

[0162] H. IoT 디바이스는 메시지를 수신하고:

[0163] 1. 메시지를 해독하려고 시도한다

[0164] 2. 지시된 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 설정한다

[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 설정하도록 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 디바이스 상에서 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 커맨드들을 교환하기 위한 기술

[0172] GATT는 일반 속성 프로파일의 두문자어이며, 그것은 2개의 블루투스 저에너지(BTLE) 디바이스가 데이터를 앞뒤로 전송하는 방식을 정의한다. 그것은 표의 각각의 엔트리에 대한 16 비트 특성 ID들을 사용하는 간단한 탐색 표 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 IoT 디바이스들과 정식으로 페어링함이 없이 이들 디바이스와 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 19는 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 16a에 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 20 바이트 버퍼 내에 저장될 수 있다. 그러나, 본 발명의 기본 원리들은 임의의 특정 버퍼 크기로 제한되지 않는다.

[0176] 도 19의 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 설정하기 위해 BT 디바이스(1910) 및 BT 통신 모듈(1901)에 의해 사용되는 메시지들을 포함할 수 있다. 예로서, 도 17에서, 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 그것을 전술한 바와 같이 처리할 수 있다(예컨대, 그것을 사용하여 비밀을 생성하고, 비밀을 사용하여 키 스트림을 생성하고, 기타 등등).

[0177] 키(1701)가 20 바이트(일부 현재 구현들에서의 최대 버퍼 크기)보다 큰 경우, 그것은 20 바이트 부분들에 기입될 수 있다. 예를 들어, 처음 20 바이트는 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 20 바이트를 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 설정하는 데 사용되는 협상 메시지들을 교환하도록 설정된다.

[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)이 (예컨대, 도 16a의 암호화된 메시지(1603)와 같은) 송신할 암호화된 데이터 패킷을 가질 때, 그것은 특성 ID <65533>에 의해 식별되는 메시지 판독 값 버퍼를 사용하여 암호화된 데이터 패킷을 한 번에 20 바이트씩 기입하기 시작한다. 이어서, IoT 디바이스 애플리케이션 로직(1902)은 판독 값 버퍼로부터 암호화된 데이터 패킷을 한 번에 20 바이트씩 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) 사이에 데이터 및 커맨드들을 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 발광체를 턴온하는 경우, 조명 애플리케이션과 관련된 온/오프 속성에 대한 변경을 반영하기 위해 UPDATE 패킷이 전송될 수 있다.

[0180] 도 20은 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 1 바이트 필드는 패킷을 GET 패킷으로서 식별하는 값(0X10)을 포함한다. 제2 1 바이트 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] 2 바이트 속성 ID는 패킷이 지향되는 애플리케이션 고유 속성을 식별한다. 예를 들어, GET 커맨드가 도 19에 예시된 IoT 디바이스(101)로 전송되고 있는 경우, 속성 ID는 요청되는 특정 애플리케이션 고유 값을 식별하는 데 사용될 수 있다. 위의 예로 되돌아가면, GET 커맨드는 발광체가 파워 온되는지 또는 파워 오프되는지(예컨대, 1 = 온, 0 = 오프)를 식별하는 값을 포함하는 조명 시스템의 전력 상태와 같은 애플리케이션 고유 속성 ID로 지향될 수 있다. IoT 디바이스(101)가 도어와 관련된 보안 장치인 경우, 값 필드는 도어의 현재 상태(예컨대, 1 = 열림, 0 = 닫힘)를 식별할 수 있다. GET 커맨드에 응답하여, 속성 ID에 의해 식별되는 현재 값을 포함하는 응답이 송신될 수 있다.

[0182] 도 20에 예시된 SET 패킷(2002) 및 UPDATE 패킷(2003)은 또한 패킷의 타입(즉, SET 및 UPDATE)을 식별하는 제1 1 바이트 필드, 요청 ID를 포함하는 제2 1 바이트 필드, 및 애플리케이션 정의 속성을 식별하는 2 바이트 속성 ID 필드를 포함한다. 또한, SET 패킷은 n 바이트 값 데이터 필드에 포함된 데이터를 길이를 식별하는 2 바이트 길이 값을 포함한다. 값 데이터 필드는 IoT 디바이스 상에서 실행될 커맨드 및/또는 소정의 방식으로(예컨대, 원하는 파라미터를 설정하기 위해, IoT 디바이스를 파워 다운시키기 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 n 바이트 값 데이터 필드의 길이를 식별하기 위한 2 바이트 길이 값 필드를 포함한다. 또한, 1 바이트 업데이트 상태 필드가 업데이트되는 변수의 현재 상태를 식별할 수 있다. 예를 들어, SET 커맨드가 IoT 디바이스에 의해 제어되는 발광체를 턴오프하려고 시도한 경우, 업데이트 상태 필드는 발광체가 성공적으로 턴오프되었는지를 지시할 수 있다.

[0184] 도 21은 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 19에 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 "온" 또는 저전력 상태에 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 22는 본 발명의 일 실시예에 따른 IoT 서비스와 IoT 디바이스 사이에 보안 통신 채널을 구현하기 위한 방법을 예시한다. 방법은 전술한 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/커맨드들을 암호화하여 암호화된 디바이스 패킷을 생성한다. 상기에 언급된 바와 같이, 세션 비밀은 IoT 디바이스 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. 22-5에서, 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 커맨드들을 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 IoT 서비스(120)와 통신하는 IoT 디바이스(101)와 관련하여 전술되지만, 본 발명의 기본 원리들은 임의의 2개의 BT 인에이블드 디바이스 사이에서 보안 통신 채널을 협상하고 설정하도록 구현될 수 있다.

[0188] 도 23a 내지 도 23c는 본 발명의 일 실시예에 따른 디바이스들을 페어링하기 위한 상세한 방법을 예시한다. 방법은 상기에 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 23b를 참조하면, IoT 허브는 2315에서 패킷을 암호화되지 않은 채널을 통해 IoT 디바이스로 전송하고, 2316에서 IoT 디바이스는 패킷의 서명을 검증한다. 2317에서, IoT 디바이스는 (예컨대, 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 (예컨대, 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 23c를 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 일치함을 검증한다. 이어서, IoT 서비스는 2334에서 페어링이 완료되었음을 지시하는 패킷을 전송하고, 2335에서 모든 후속 메시지들이 세션 비밀을 사용하여 암호화된다.

[0194] 본 발명의 실시예는 위에서 설명된 다양한 단계를 포함할 수 있다. 단계는 범용 또는 특수-목적 프로세서로 하

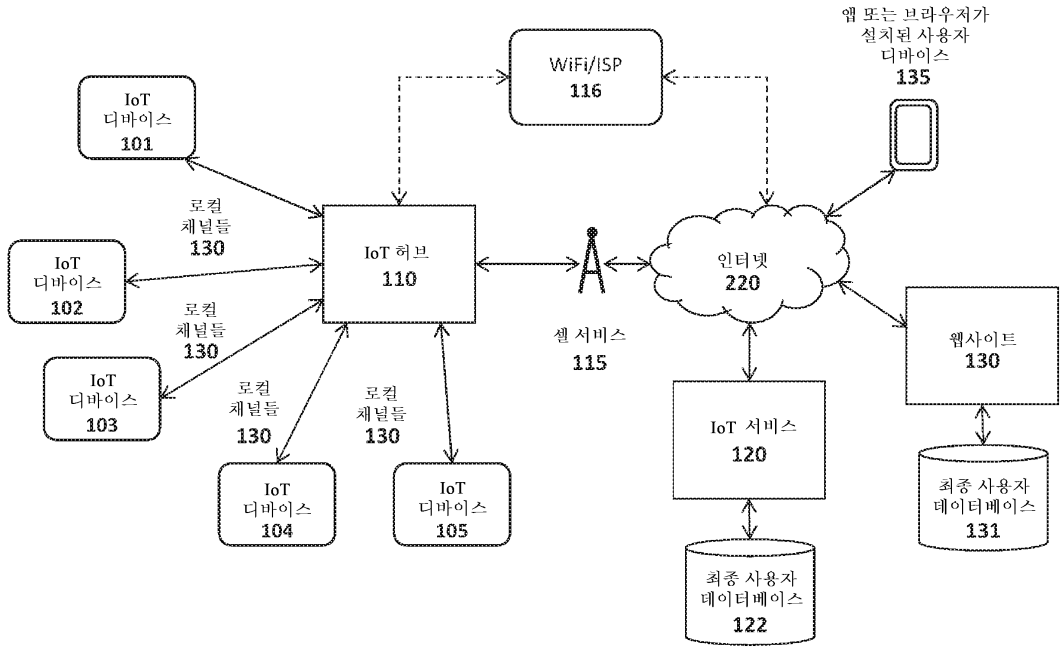
여금 그 단계를 수행하게 하기 위해 사용될 수 있는 기계-실행 가능 명령어들로 구현될 수 있다. 대안적으로, 이들 단계는 단계를 수행하기 위한 하드와이어드 로직(hardwired logic)을 포함하는 특정 하드웨어 컴포넌트들에 의해, 또는 프로그래밍된 컴퓨터 컴포넌트와 맞춤형 하드웨어 컴포넌트의 임의의 조합에 의해 수행될 수 있다.

[0195] 본 명세서에 설명된 바와 같이, 명령어는, 소정의 동작을 수행하도록 구성되거나, 비일시적인 컴퓨터 판독 가능 매체에 수록되는 메모리에 저장된 소프트웨어 명령어 또는 미리 결정된 기능을 갖는 주문형 집적 회로(ASIC)와 같은 하드웨어의 특정한 구성을 지칭할 수 있다. 따라서, 도면에 도시된 기술은 하나 이상의 전자 디바이스(예를 들어, 최종 스테이션, 네트워크 요소 등) 상에 저장되고 그것 상에서 실행되는 코드 및 데이터를 사용하여 구현될 수 있다. 그러한 전자 디바이스는 비일시적 컴퓨터 기계 판독 가능 저장 매체(예를 들어, 자기 디스크, 광 디스크, 랜덤 액세스 메모리, 판독 전용 메모리, 플래시 메모리 디바이스, 상변화 메모리) 및 일시적 컴퓨터 기계 판독 가능 통신 매체(예를 들어, 전기, 광학, 음향 또는 다른 형태의 전파 신호 - 예를 들어, 반송파, 적외선 신호, 디지털 신호 등)와 같은 컴퓨터 기계 판독 가능 매체를 사용하여 코드 및 데이터를 저장하고 (내부적으로 그리고/또는 네트워크를 통해 다른 전자 디바이스와) 통신한다. 부가적으로, 그러한 전자 디바이스는 전형적으로 하나 이상의 저장 디바이스(비일시적 기계 판독 가능 저장 매체), 사용자 입력/출력 디바이스(예를 들어, 키보드, 터치스크린, 및/또는 디스플레이), 및 네트워크 접속부와 같은 하나 이상의 다른 컴포넌트에 결합된 하나 이상의 프로세서의 세트를 포함한다. 프로세서의 세트와 다른 컴포넌트의 결합은 전형적으로 하나 이상의 버스 및 브리지(또한 버스 제어기로 지칭됨)를 통해 이루어진다. 저장 디바이스 및 네트워크 트래픽을 반송하는 신호는 각각 하나 이상의 기계 판독 가능 저장 매체 및 기계 판독 가능 통신 매체를 대표한다. 따라서, 주어진 전자 디바이스의 저장 디바이스는 전형적으로 그 전자 디바이스의 하나 이상의 프로세서의 세트 상에서의 실행을 위한 코드 및/또는 데이터를 저장한다. 물론, 본 발명의 실시예의 하나 이상의 부분이 소프트웨어, 펌웨어, 및/또는 하드웨어의 상이한 조합을 사용하여 구현될 수 있다.

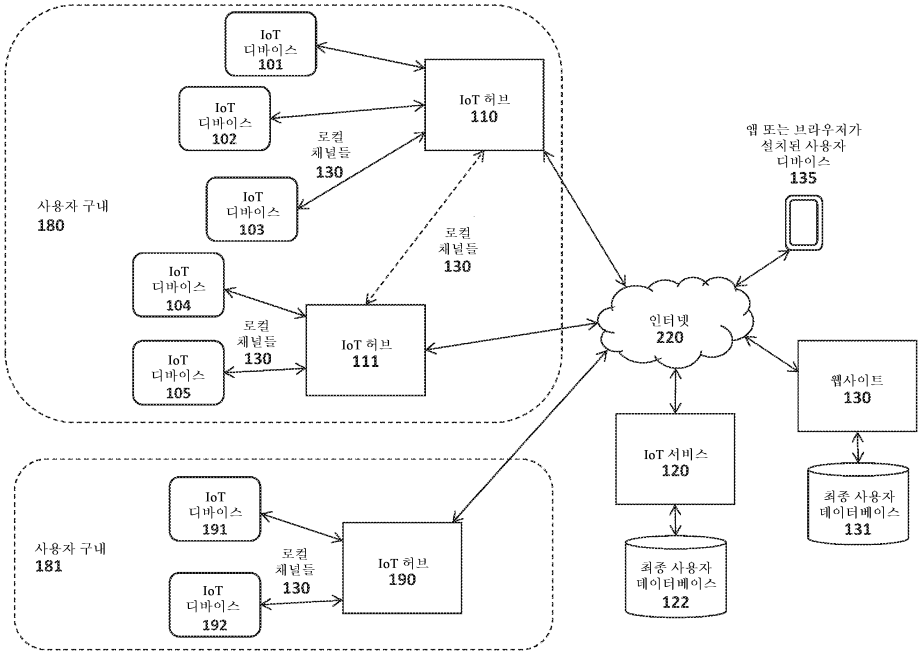
[0196] 이러한 상세한 설명 전반에 걸쳐, 설명의 목적으로, 본 발명의 완전한 이해를 제공하기 위해 다수의 특정 상세가 기술되었다. 그러나, 본 발명은 이러한 특정 상세들 중 일부 없이도 실시될 수 있다는 것이 당업자에게 명백할 것이다. 소정의 경우에, 잘 알려진 구조 및 기능은 본 발명의 주제를 불명확하게 하는 것을 피하기 위해 정성 들여 상세히 설명되지 않았다. 따라서, 본 발명의 범주 및 사상은 후속하는 청구범위의 관점에서 판단되어야 한다.

도면

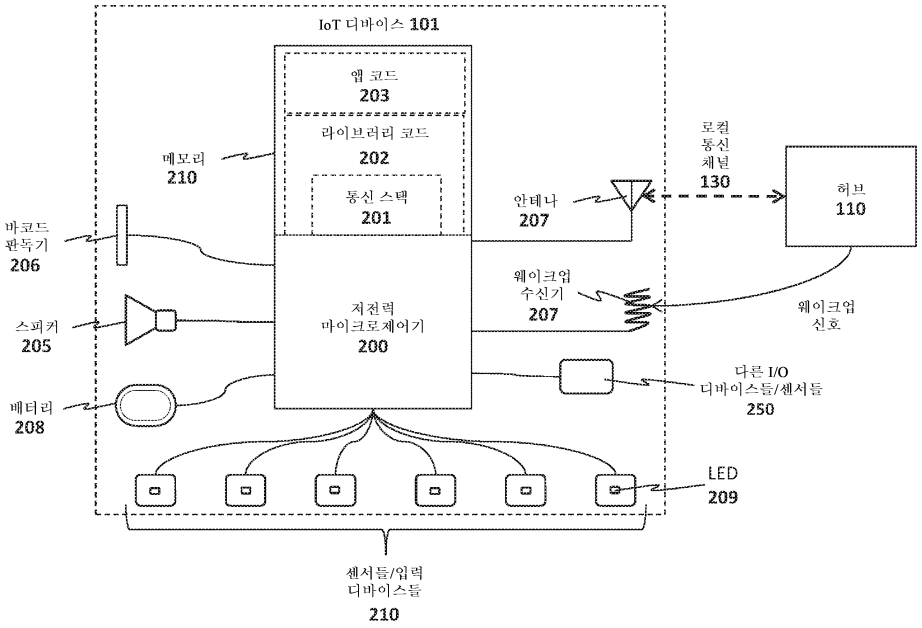
도면1a



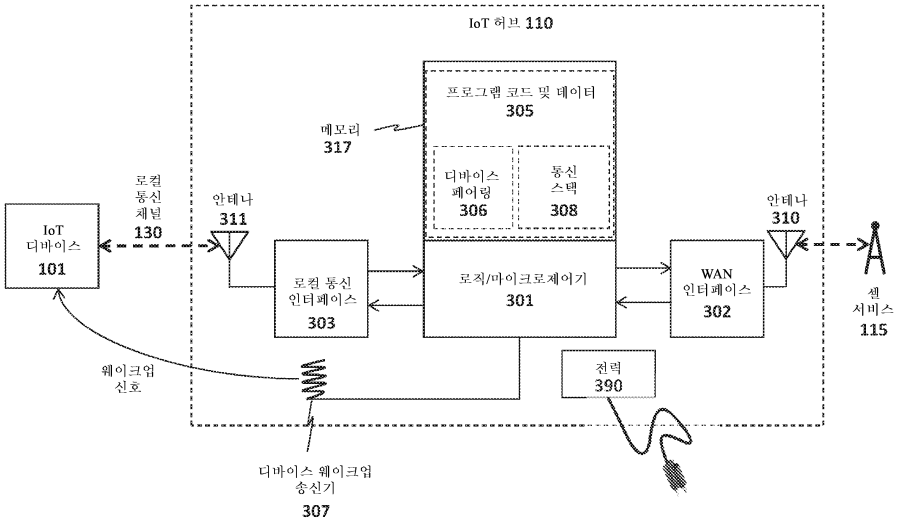
도면1b


IoT 디바이스 101
IoT 디바이스 102
IoT 디바이스 103
IoT 디바이스 104
IoT 디바이스 105
로컬 채널들 130
IoT 허브 110
사용자 구내 180
로컬 채널들 130
로컬 채널들 130
IoT 허브 111
앱 또는 브라우저가 설치된 사용자 디바이스 135
인터넷 220
웹사이트 130
IoT 서비스 120
최종 사용자 데이터베이스 131
최종 사용자 데이터베이스 122
IoT 디바이스 191
IoT 디바이스 192
로컬 채널들 130
IoT 허브 190
사용자 구내 181

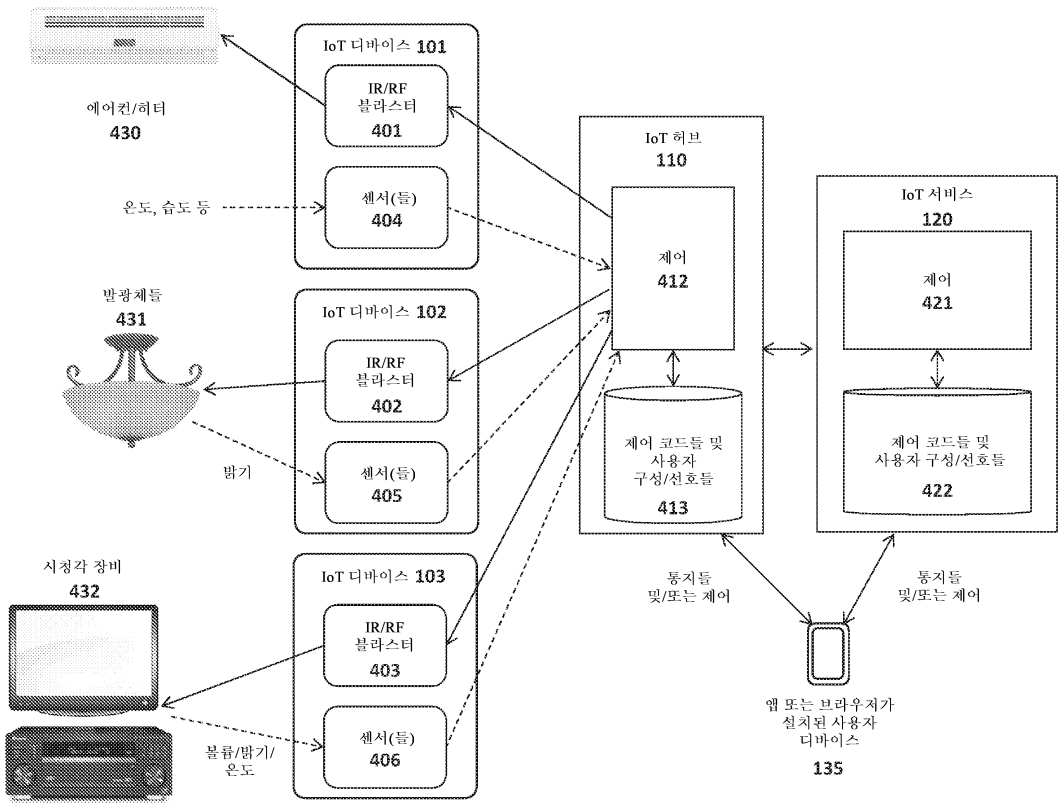
도면2


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

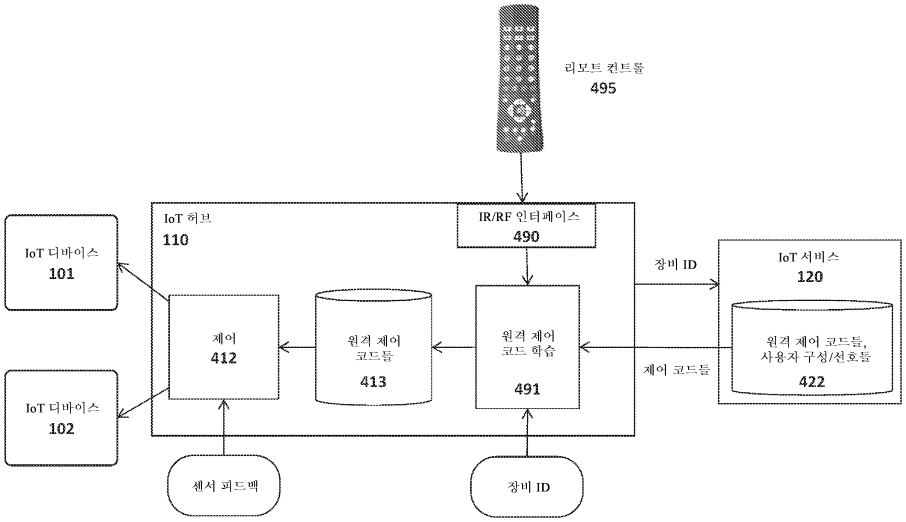
도면3


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

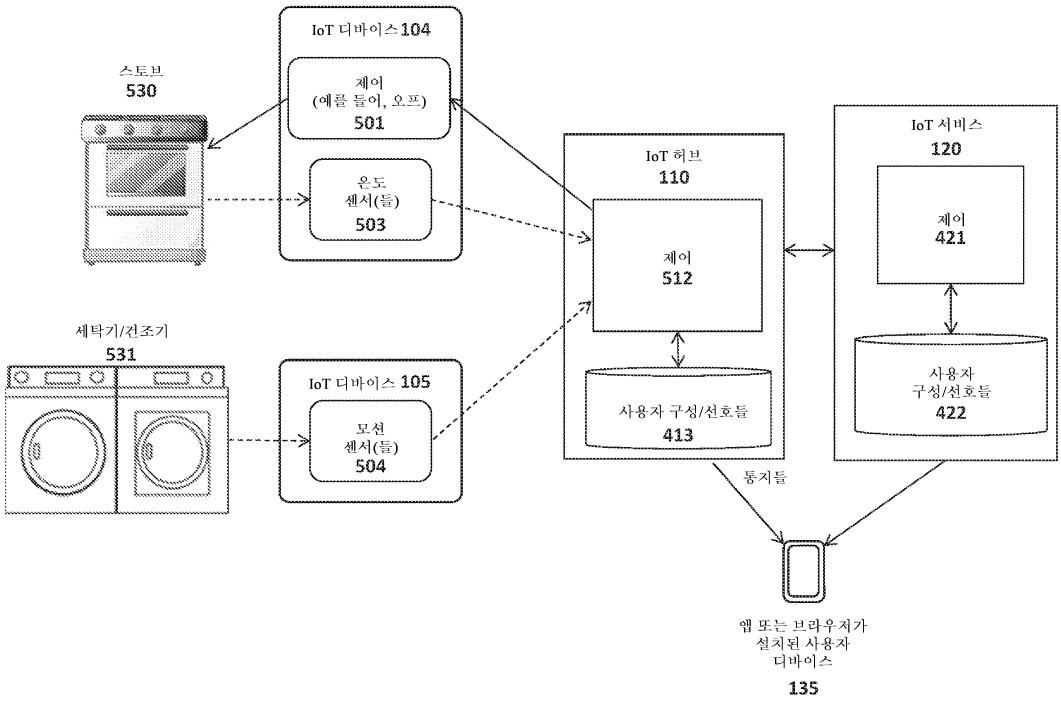
도면4a


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

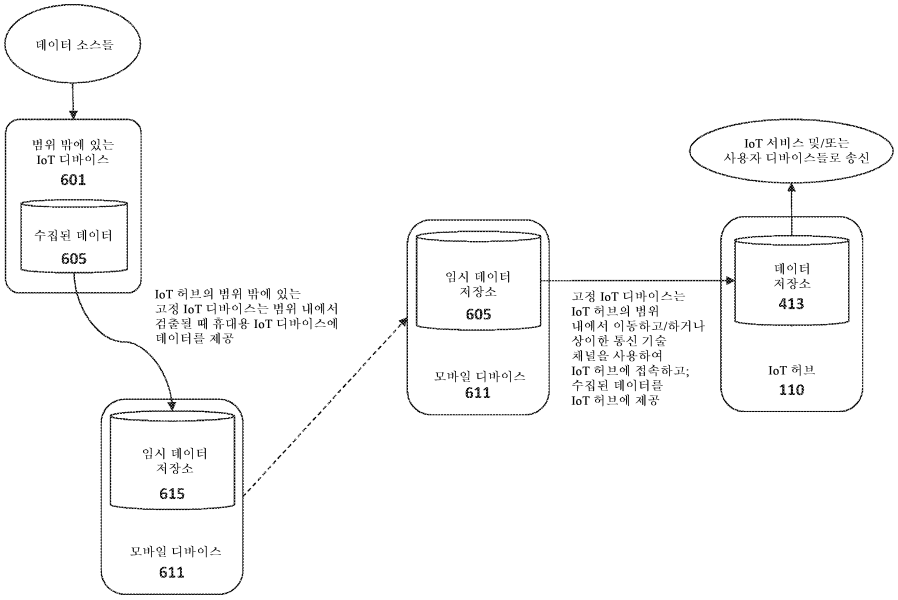
도면4b


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

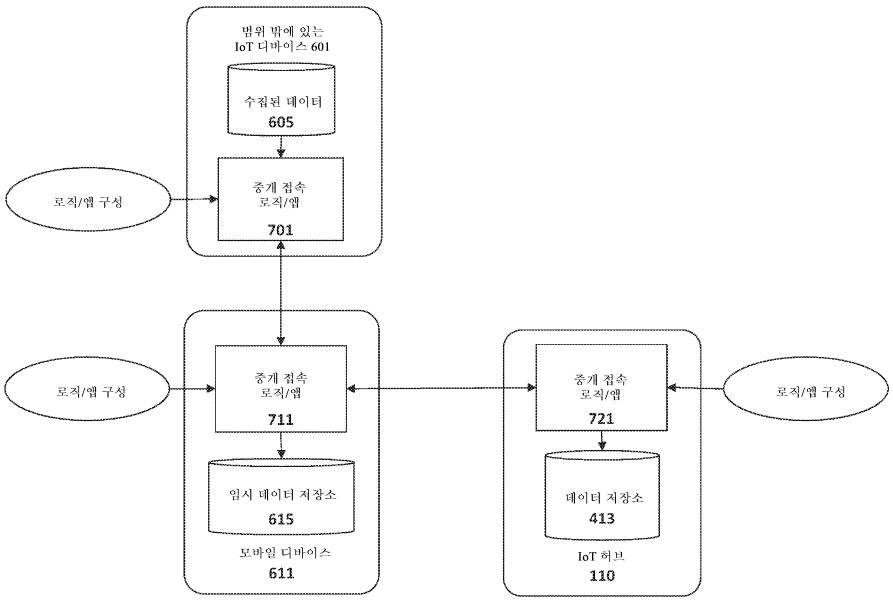
도면5


스토브
530
IoT 디바이스 104
제어
(예를 들어, 오프)
501
온도
센서(들)
503
세탁기/건조기
531
IoT 디바이스 105
모션
센서(들)
504
IoT 허브
110
제어
512
사용자 구성/선호들
413
IoT 서비스
120
제어
421
사용자
구성/선호들
422
통지들
앱 또는 브라우저가
설치된 사용자
디바이스
135

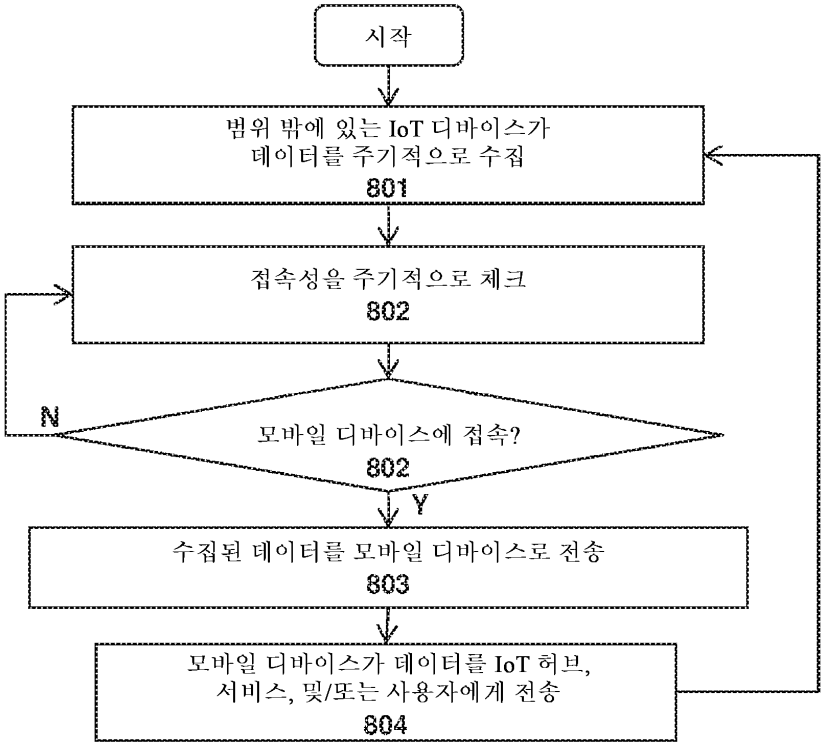
도면6


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
데이터
저장소
413
IoT 허브
110
IoT 서비스 및/또는
사용자 디바이스들로 송신

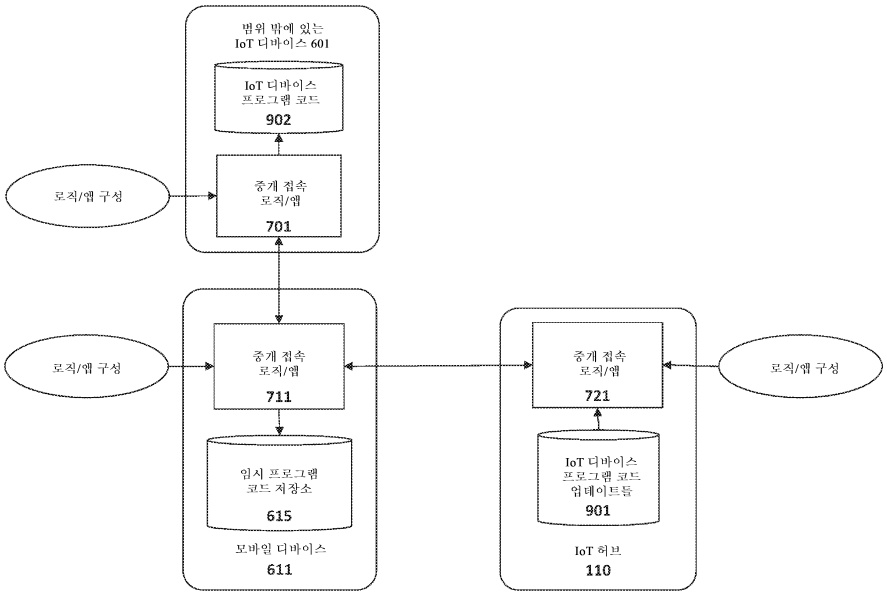
도면7


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

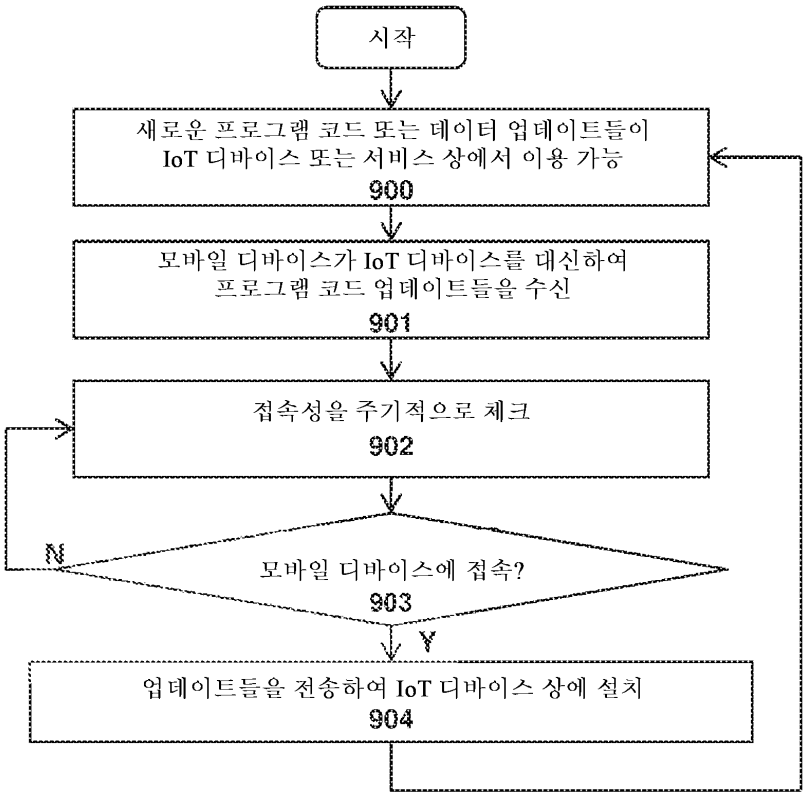
도면8


시작
범위 밖에 있는 IoT 디바이스가
데이터를 주기적으로 수집
801
접속성을 주기적으로 체크
802
N
모바일 디바이스에 접속?
802
Y
수집된 데이터를 모바일 디바이스로 전송
803
모바일 디바이스가 데이터를 IoT 허브,
서비스, 및/또는 사용자에게 전송
804

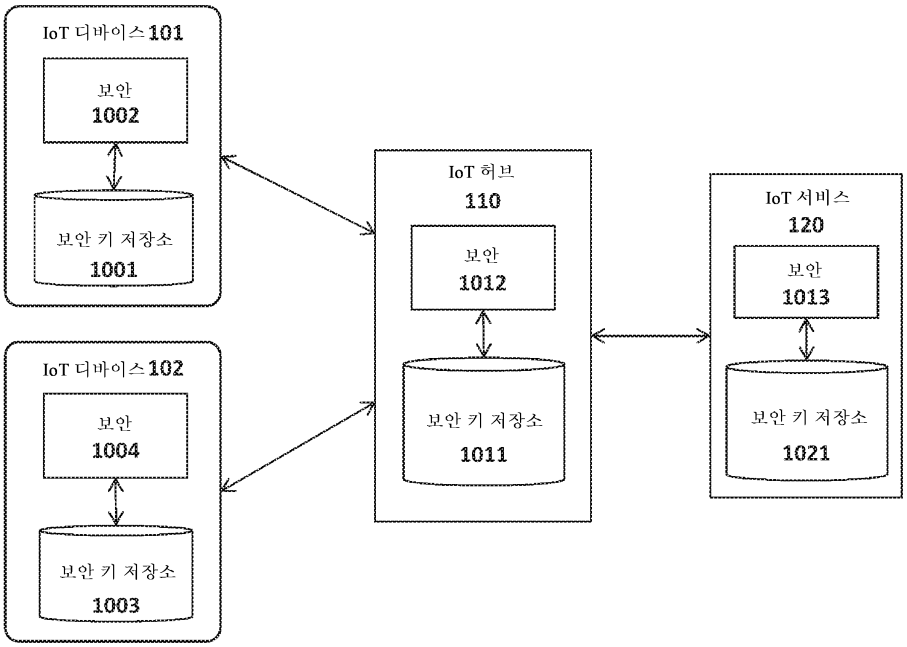
도면9a


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

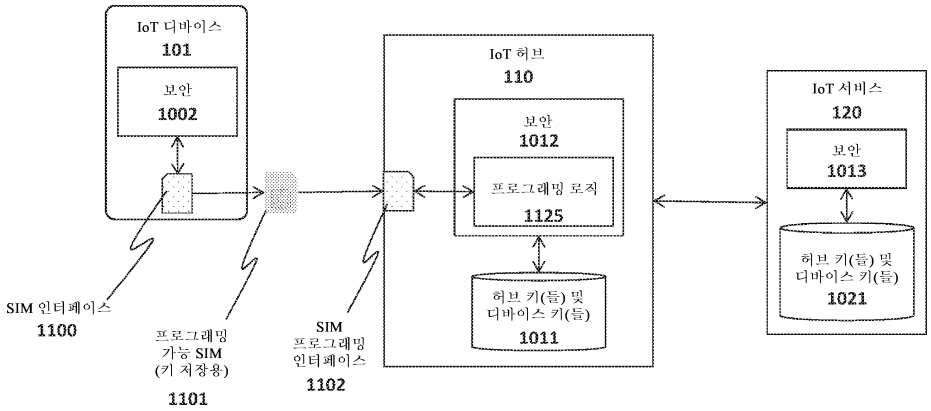
도면9b


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
접속성을 주기적으로 체크
902
모바일 디바이스에 접속?
903
N
Y
업데이트들을 전송하여 IoT 디바이스 상에 설치
904

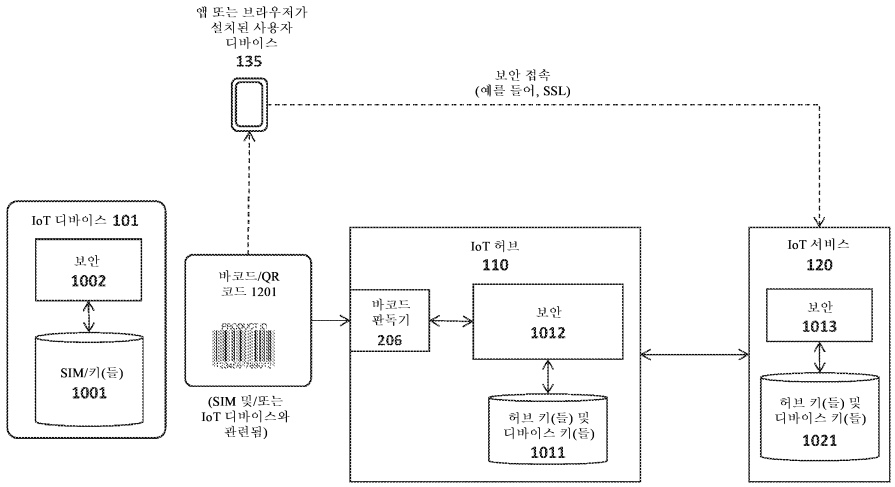
도면10


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

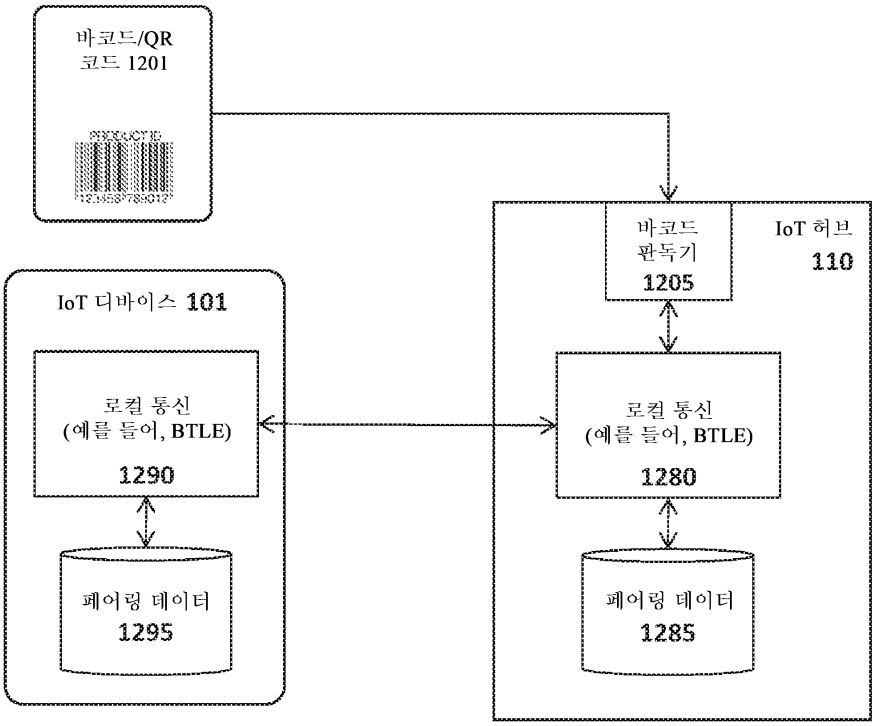
도면11


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

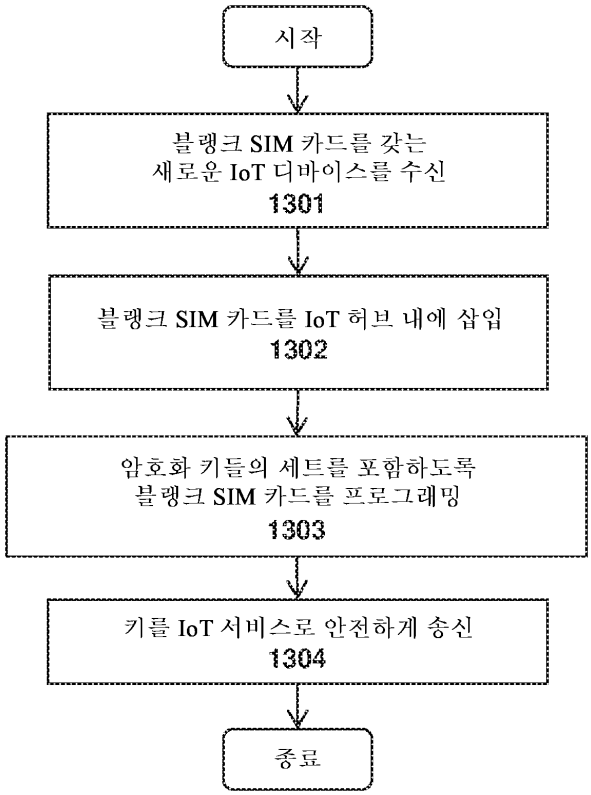
도면12a


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
(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

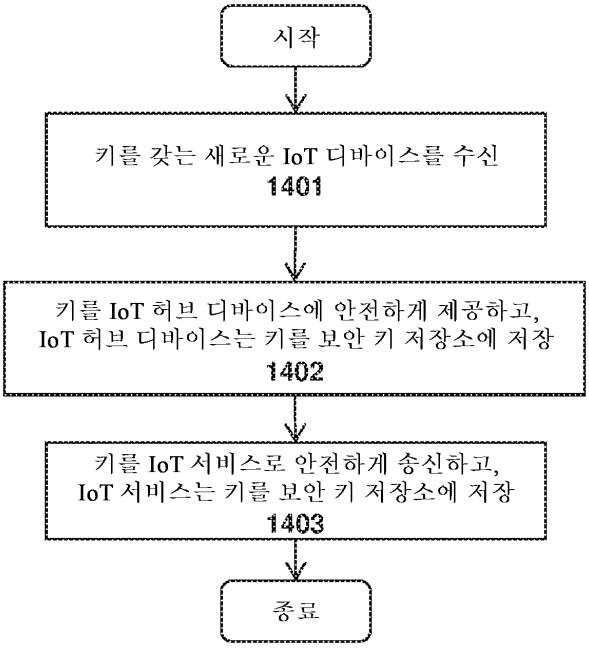
도면12b


바코드/QR
코드 1201
바코드
판독기
1205
IoT 허브
110
IoT 디바이스 101
로컬 통신
(예를 들어, BTLE)
1290
로컬 통신
(예를 들어, BTLE)
1280
페어링 데이터
1295
페어링 데이터
1285

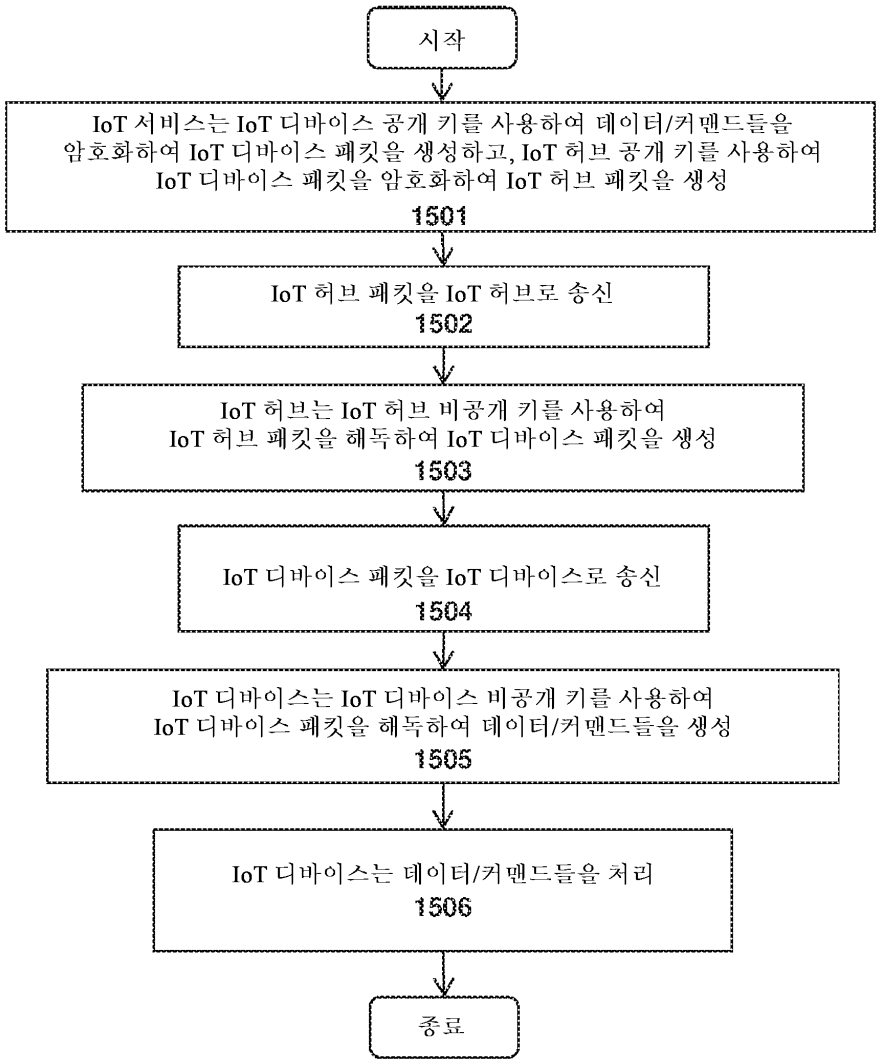
도면13


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

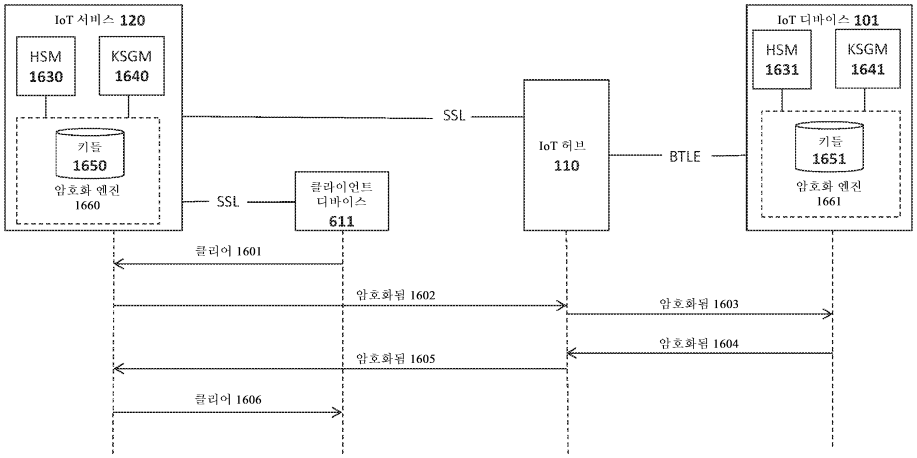
도면14


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 송신하고,
IoT 서비스는 키를 보안 키 저장소에 저장
1403
종료

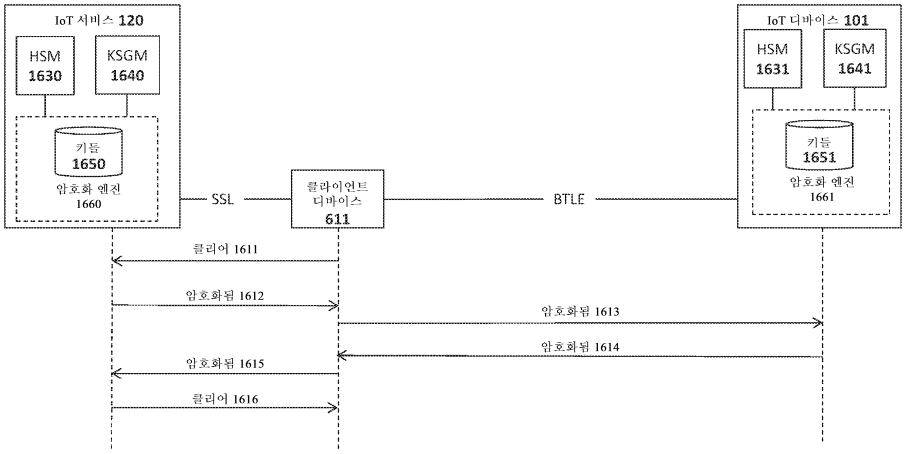
도면15


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

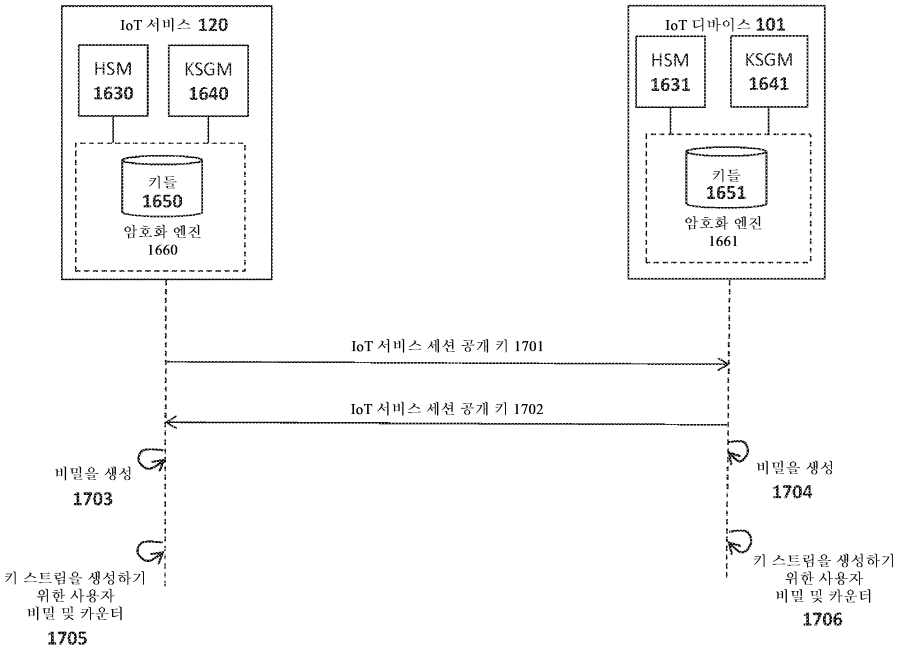
도면16a


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

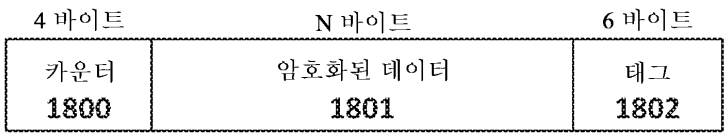
도면16b



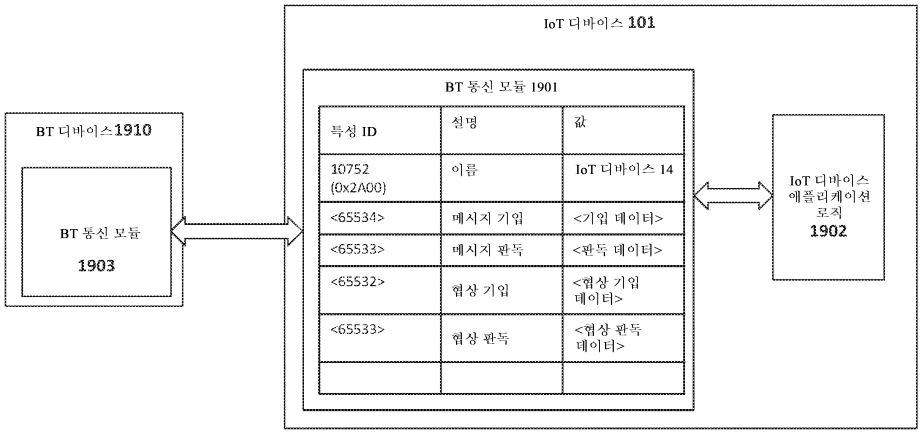
도면17



도면18



도면19



도면20

1 바이트	1 바이트	2 바이트
0x10	요청 ID	속성 ID

GET 패킷 2001

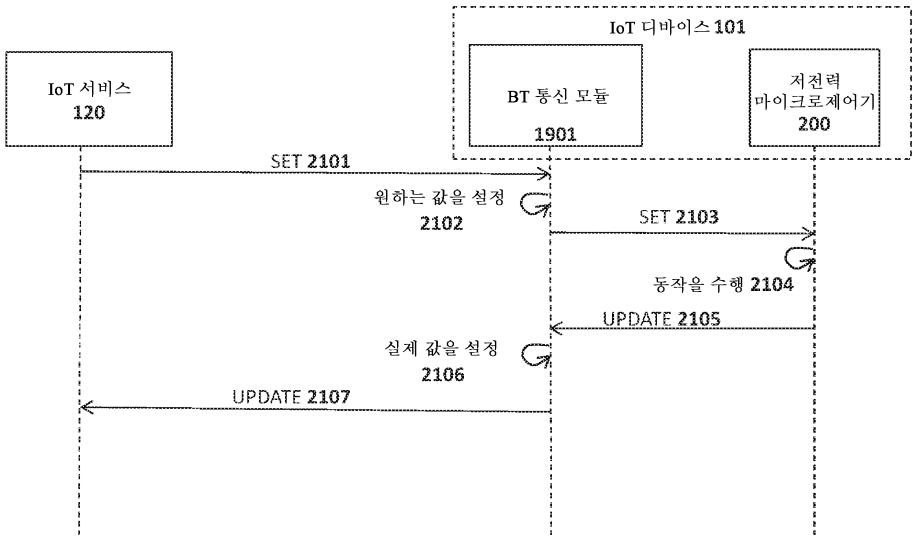
1 바이트	1 바이트	2 바이트	2 바이트	n 바이트
0x11	요청 ID	속성 ID	값 길이	값 데이터

SET 패킷 2002

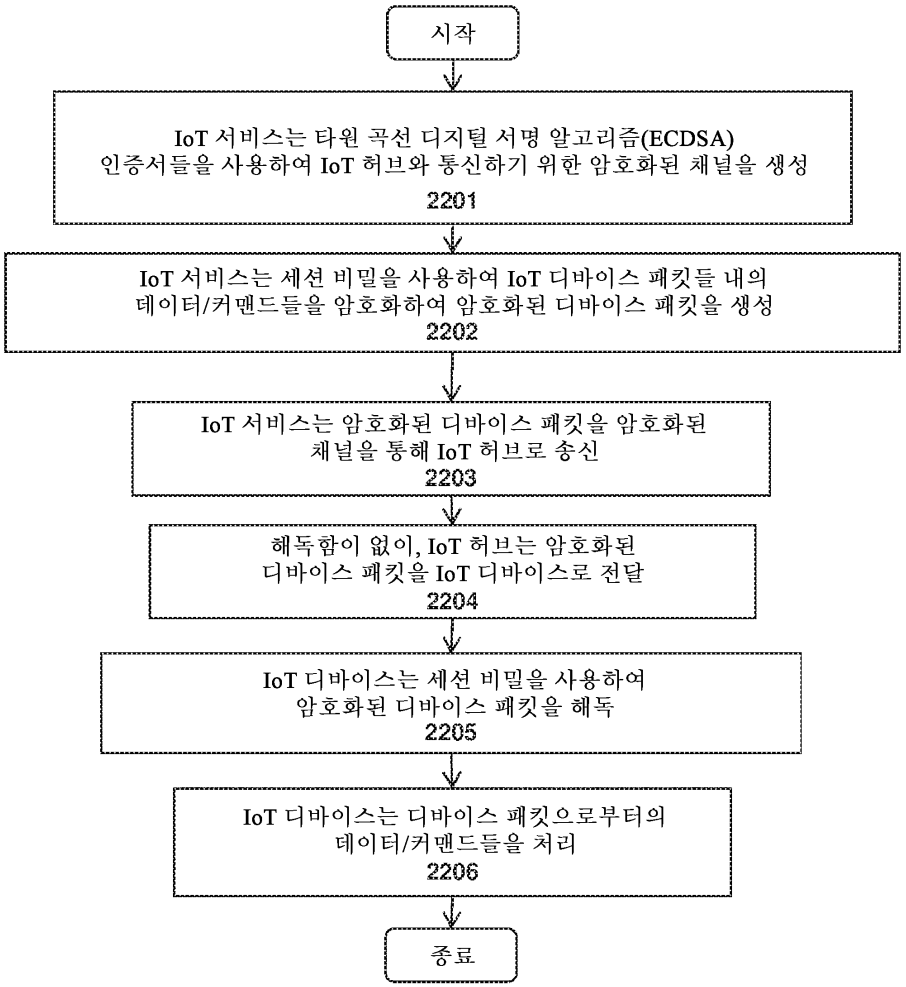
1 바이트	1 바이트	2 바이트	1 바이트	2 바이트	n 바이트
0x12	요청 ID	속성 ID	업데이트 상태	값 길이	값 데이터

UPDATE 패킷 2003

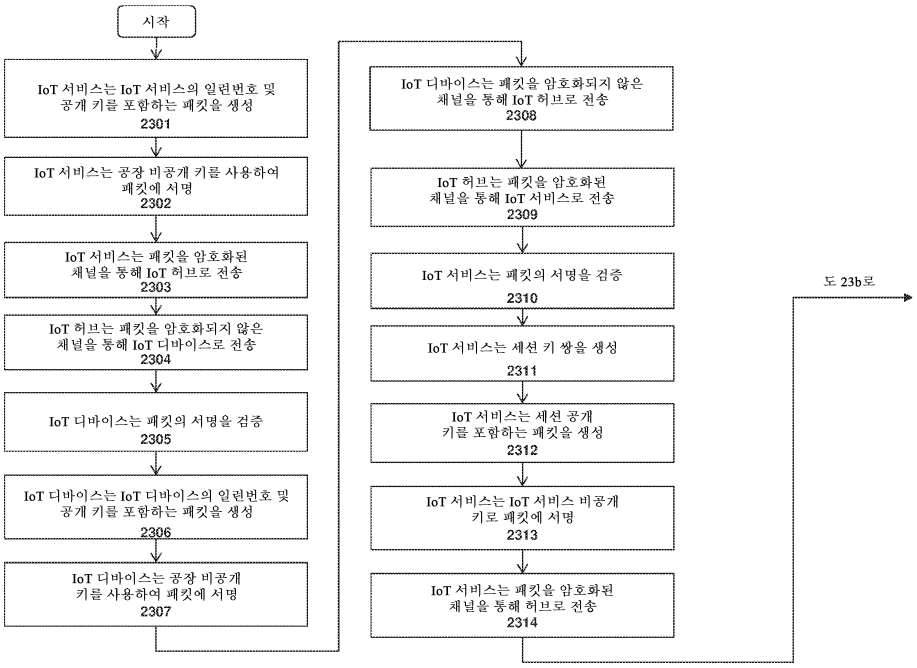
도면21



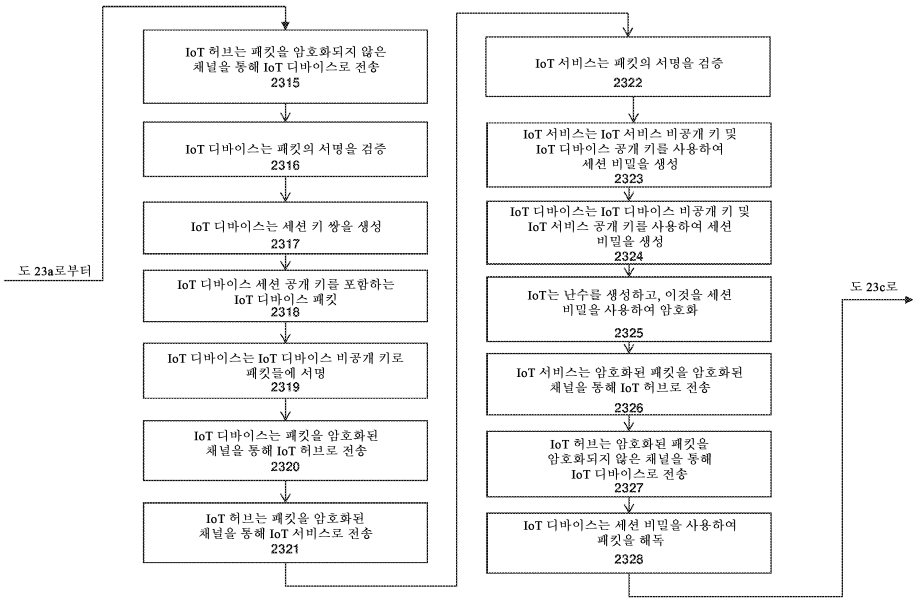
도면22


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/커맨드들을 암호화하여 암호화된 디바이스 패킷을 생성
2202
IoT 서비스는 암호화된 디바이스 패킷을 암호화된 채널을 통해 IoT 허브로 송신
2203
해독함이 없이, IoT 허브는 암호화된 디바이스 패킷을 IoT 디바이스로 전달
2204
IoT 디바이스는 세션 비밀을 사용하여 암호화된 디바이스 패킷을 해독
2205
IoT 디바이스는 디바이스 패킷으로부터의 데이터/커맨드들을 처리
2206
종료

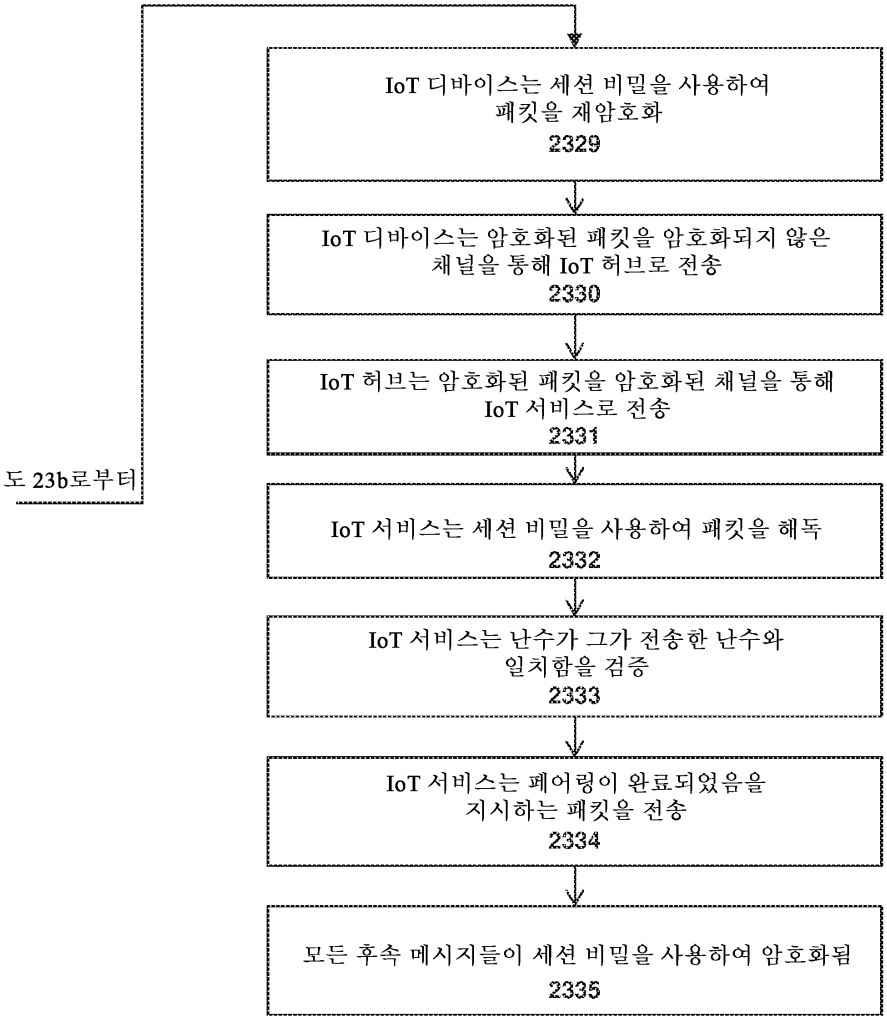
도면23a


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, 이것을 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 지시하는 패킷을 전송
2334
모든 후속 메시지들이 세션 비밀을 사용하여 암호화됨
2335