(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6926085号
(P6926085)

(45)発行日　令和3年8月25日(2021.8.25)　(24)登録日　令和3年8月6日(2021.8.6)

(51)Int.Cl.		F I		
G06F 13/00	(2006.01)	G06F	13/00	357A
H04L 9/08	(2006.01)	H04L	9/00	601C
H04L 9/32	(2006.01)	H04L	9/00	673B
G06F 21/45	(2013.01)	G06F	21/45	
H04Q 9/00	(2006.01)	G06F	13/00	358A

請求項の数 5　(全 54 頁)　最終頁に続く

(21)出願番号　特願2018-531069(P2018-531069)
(86)(22)出願日　平成28年12月14日(2016.12.14)
(65)公表番号　特表2019-502206(P2019-502206A)
(43)公表日　平成31年1月24日(2019.1.24)
(86)国際出願番号　PCT/US2016/066443
(87)国際公開番号　WO2017/106224
(87)国際公開日　平成29年6月22日(2017.6.22)
審査請求日　令和1年12月16日(2019.12.16)
(31)優先権主張番号　14/967,820
(32)優先日　平成27年12月14日(2015.12.14)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　14/967,870
(32)優先日　平成27年12月14日(2015.12.14)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　515290745
アフェロ　インコーポレイテッド
Ａｆｅｒｏ，　Ｉｎｃ．
アメリカ合衆国、９４０２２、カリフォルニア州、ロスアルトス、エル・カミノ・レアル、４９７０、スイート　２１０
(74)代理人　100094569
弁理士　田中　伸一郎
(74)代理人　100088694
弁理士　弟子丸　健
(74)代理人　100103610
弁理士　▲吉▼田　和彦
(74)代理人　100067013
弁理士　大塚　文昭

最終頁に続く

(54)【発明の名称】安全なモノのインターネット（ＩｏＴ）デバイスプロビジョニングのシステム及び方法

(57)【特許請求の範囲】
【請求項１】
方法であって、
ＩｏＴサービスにより、新しいモノのインターネット（ＩｏＴ）デバイス識別（ＩＤ）コードと関連付けＩＤコードとの間の関連付けを生成することであって、前記新しいＩｏＴデバイスＩＤコード及び前記関連付けＩＤコードは各々等しい長さのコードである、生成することと、
前記ＩｏＴサービスにより、前記ＩｏＴサービスのＩｏＴデバイスデータベースに前記関連付けを記憶することであって、前記ＩｏＴデバイスデータベースは、ＩｏＴデバイスがプロビジョニングされなかったことを示す第１の値及びＩｏＴデバイスがプロビジョニングされたことを示す第２の値を含む、記憶することと、
前記ＩｏＴサービスにより、新しいＩｏＴデバイス上に印刷されるバーコード又はＱＲコード(登録商標)を提供することであって、前記バーコード又はＱＲコード(登録商標)は前記関連付けＩＤコードを符号化し、前記新しいＩｏＴデバイスは安全な通信モジュールに前記新しいＩｏＴデバイスＩＤコードを記憶し、前記安全な通信モジュールはプログラム可能な加入者識別モジュール（ＳＩＭ）を含む、提供することと、
ＩｏＴハブにより、Ｂｌｕｅｔｏｏｔｈ(登録商標)　Ｌｏｗ　Ｅｎｅｒｇｙ（ＢＴＬＥ）リンクを介して、前記新しいＩｏＴデバイスとローカル通信チャネルを確立することであって、前記新しいＩｏＴデバイスはその上に印刷された前記バーコード又はＱＲコード(登録商標)を含む、確立することと、

前記ＩｏＴハブにより、前記関連付けＩＤコードを前記新しいＩｏＴデバイスから決定するために、前記バーコード又はＱＲコード(登録商標)を光学的に読み取ることと、
前記ＩｏＴハブにより、安全な通信チャネルを介して前記関連付けＩＤコードを前記ＩｏＴサービスに送信することであって、前記ＩｏＴサービスは、前記新しいＩｏＴデバイスＩＤコードを決定するために前記関連付けＩＤコードを使用して前記ＩｏＴデバイスデータベース内でルックアップを実施する、送信することと、
前記ＩｏＴサービスにより、前記新しいＩｏＴデバイスＩＤコードに基づいて前記ＩｏＴサービス上で暗号鍵を識別することと、
前記ＩｏＴサービスにより、前記暗号鍵及び楕円曲線暗号化を使用して前記新しいＩｏＴデバイスと暗号化通信チャネルを確立することと、
前記ＩｏＴサービスで前記新しいＩｏＴデバイスをプロビジョニングすることと、
前記ＩｏＴサービスにより、前記新しいＩｏＴデバイスがプロビジョニングされた後に前記ＩｏＴハブが前記新しいＩｏＴデバイスと通信することを許可することと、
前記ＩｏＴサービスにより、前記新しいＩｏＴデバイスがプロビジョニングされたことを示すために前記ＩｏＴデバイスデータベースを更新することと、
を含む方法。
【請求項２】
前記新しいＩｏＴデバイスをプロビジョニングすることは、前記ＩｏＴサービスから前記ＩｏＴハブへ前記新しいＩＯＴデバイスＩＤコードを安全に送信して、前記ＩｏＴハブに前記新しいＩｏＴデバイスとの通信を許可するように指示することを含む、請求項１に記載の方法。
【請求項３】
前記ＱＲコード(登録商標)又はバーコードを光学的に読み取ることは、前記ＩｏＴサービスに安全に連結されたクライアントデバイス上のカメラで前記ＱＲコード(登録商標)又はバーコードをキャプチャすることを含む、請求項１に記載の方法。
【請求項４】
前記ＱＲコード(登録商標)又はバーコードを光学的に読み取ることは、前記クライアントデバイス上のアプリ若しくはアプリケーションを使用して実施される、請求項３に記載の方法。
【請求項５】
前記新しいＩｏＴデバイスと前記暗号化通信チャネルを確立することは、
前記ＩｏＴサービスと前記新しいＩｏＴデバイスとの間の通信を、前記ＩｏＴハブ又はモバイルユーザデバイスを通して確立することと、
サービス公開鍵及びサービス秘密鍵を前記ＩｏＴサービス上の第１の暗号化エンジンの鍵生成ロジックによって生成することと、
デバイス公開鍵及びデバイス秘密鍵を前記新しいＩｏＴデバイス上の第２の暗号化エンジンの鍵生成ロジックによって生成することと、
前記サービス公開鍵を前記第１の暗号化エンジンから前記第２の暗号化エンジンに送信し、前記デバイス公開鍵を前記第２の暗号化エンジンから前記第１の暗号化エンジンに送信することと、
前記デバイス公開鍵及び前記サービス秘密鍵を使用してシークレットを生成することと、
前記サービス公開鍵及び前記デバイス秘密鍵を使用して同一の前記シークレットを生成することと、
前記シークレットを使用して又は前記シークレットから派生したデータ構造を使用して、前記第１の暗号化エンジンと前記第２の暗号化エンジンとの間で送信されるデータパケットを暗号化及び復号することと、
を含む、請求項１に記載の方法。
【発明の詳細な説明】
【技術分野】

【０００１】
本発明は、概して、コンピュータシステムの分野に関する。より具体的には、本発明は、安全なモノのインターネット（ＩｏＴ）デバイスプロビジョニングのためのシステム及び方法に関する。
【背景技術】
【０００２】
［関連技術の説明］
「モノのインターネット」は、インターネットインフラストラクチャ内に、一意的に識別可能に組み込まれたデバイスの相互接続を指す。最終的に、ＩｏＴは、事実上あらゆるタイプの物理的なモノが、それ自体若しくはその周囲についての情報を提供し得、及び／又はインターネットをわたってクライアントデバイスを介して遠隔制御され得る、広範囲の新しいタイプのアプリケーションをもたらすことが期待される。
【０００３】
接続性、電力、及び規格化の欠如に関連する問題のために、ＩｏＴ開発及び採用は遅れている。例えば、ＩｏＴ開発及び採用に対する１つの障害は、開発者が新しいＩｏＴデバイス及びサービスを設計して提供することを可能にする標準プラットフォームが存在しないことである。ＩｏＴ市場に参入するためには、開発者は、所望のＩｏＴ実装に対応するために必要なネットワークプロトコル及びインフラストラクチャ、ハードウェア、ソフトウェア、並びにサービスを含む、ＩｏＴプラットフォーム全体を一から設計する必要がある。その結果、ＩｏＴデバイスの各プロバイダは、ＩｏＴデバイスの設計と接続のために専有の技術を使用しており、エンドユーザにとって複数のタイプのＩｏＴデバイスの採用が厄介となっている。ＩｏＴの採用への別の障害は、ＩｏＴデバイスの接続及び給電に関連する困難である。例えば、冷蔵庫、ガレージドアオープナー、環境センサ、家庭用セキュリティセンサ／コントローラなどの接続機器は、接続された各ＩｏＴ機器に給電するための電源を必要とし、そのような電源はしばしば便利な位置に設けられていない。
【０００４】
存在する別の問題は、Ｂｌｕｅｔｏｏｔｈ(登録商標) ＬＥ（ＢＴＬＥ）などのＩｏＴデバイスを相互接続するために使用される無線技術が、概して、近距離技術であるということである。したがって、ＩｏＴ実装のためのデータ収集ハブがＩｏＴデバイスの範囲外にある場合、そのＩｏＴデバイスは、ＩｏＴハブにデータを送信することができない（逆もまた同様）。その結果として、ＩｏＴデバイスが、範囲外にあるＩｏＴハブ（又は他のＩｏＴデバイス）にデータを提供することを可能にする技術が必要とされる。
【０００５】
加えて、ＢＴＬＥなどの無線通信プロトコルに依存する現在のＩｏＴ実装は、適切なセキュリティ対策を提供しない。したがって、追加の技術が、ＩｏＴ実装においてセキュリティを向上させるために必要とされる。
【０００６】
本発明のより良好な理解は、以下の図面と併せた以下の詳細な説明から得ることができる。
【図面の簡単な説明】
【０００７】
【図１Ａ】ＩｏＴシステムアーキテクチャの異なる実施形態を例示する。
【図１Ｂ】ＩｏＴシステムアーキテクチャの異なる実施形態を例示する。
【図２】本発明の一実施形態によるＩｏＴデバイスを例示する。
【図３】本発明の一実施形態によるＩｏＴハブを例示する。
【図４Ａ】ＩｏＴデバイスからのデータを制御及び収集し、通知を生成するための本発明の実施形態を例示する。
【図４Ｂ】ＩｏＴデバイスからのデータを制御及び収集し、通知を生成するための本発明の実施形態を例示する。
【図５】ＩｏＴデバイスからのデータを収集し、ＩｏＴハブ及び／又はＩｏＴサービスか

らの通知を生成するための本発明の実施形態を例示する。
【図６】中間モバイルデバイスが固定ＩｏＴデバイスからデータを収集し、データをＩｏＴハブに提供する、システムの一実施形態を例示する。
【図７】本発明の一実施形態で実装される中間接続ロジックを例示する。
【図８】本発明の一実施形態による方法を例示する。
【図９Ａ】プログラムコード及びデータ更新がＩｏＴデバイスに提供される一実施形態を例示する。
【図９Ｂ】プログラムコード及びデータ更新がＩｏＴデバイスに提供される方法の一実施形態を例示する。
【図１０】セキュリティアーキテクチャの一実施形態の高レベル図を例示する。
【図１１】ＩｏＴデバイス上に鍵を記憶するために加入者識別モジュール（subscriber identity module）（ＳＩＭ）が使用されるアーキテクチャの一実施形態を例示する。
【図１２Ａ】バーコード又はＱＲコード(登録商標)を使用してＩｏＴデバイスが登録される一実施形態を例示する。
【図１２Ｂ】バーコード又はＱＲコード(登録商標)を使用してペアリングが実行される一実施形態を例示する。
【図１３】ＩｏＴハブを使用してＳＩＭをプログラミングするための方法の一実施形態を例示する。
【図１４】ＩｏＴデバイスをＩｏＴハブ及びＩｏＴサービスに登録するための方法の一実施形態を例示する。
【図１５】ＩｏＴデバイスに送信されるデータを暗号化するための方法の一実施形態を例示する。
【図１６Ａ】ＩｏＴサービスとＩｏＴデバイスとの間でデータを暗号化するための本発明の異なる実施形態を例示する。
【図１６Ｂ】ＩｏＴサービスとＩｏＴデバイスとの間でデータを暗号化するための本発明の異なる実施形態を例示する。
【図１７】安全な鍵交換を実行し、共通シークレットを生成し、シークレットを使用してキーストリームを生成するための本発明の実施形態を例示する。
【図１８】本発明の一実施形態によるパケット構造を例示する。
【図１９】ＩｏＴデバイスと正式にペアリングすることなくＩｏＴデバイスとの間でデータを読み書きするための一実施形態に用いられる技術を例示する。
【図２０】本発明の一実施形態で用いられるコマンドパケットの例示的なセットを例示する。
【図２１】コマンドパケットを使用したトランザクションの例示的なシーケンスを例示する。
【図２２】本発明の一実施形態による方法を例示する。
【図２３Ａ】本発明の一実施形態による安全なペアリングのための方法を例示する。
【図２３Ｂ】本発明の一実施形態による安全なペアリングのための方法を例示する。
【図２３Ｃ】本発明の一実施形態による安全なペアリングのための方法を例示する。
【図２４】データ送信状態を識別するためのアドバタイジング間隔を調節するための本発明の一実施形態を例示する。
【図２５】本発明の一実施形態による方法を例示する。
【図２６Ａ】複数のＩｏＴハブがデータ／コマンドをＩｏＴデバイスに送信しようとする一実施形態の動作を例示する。
【図２６Ｂ】複数のＩｏＴハブがデータ／コマンドをＩｏＴデバイスに送信しようとする一実施形態の動作を例示する。
【図２６Ｃ】複数のＩｏＴハブがデータ／コマンドをＩｏＴデバイスに送信しようとする一実施形態の動作を例示する。
【図２７】本発明の一実施形態による方法を例示する。
【図２８】安全なＩｏＴデバイスプロビジョニングのためのシステムの一実施形態を例示

する。
【図２９】本発明の一実施形態による方法を例示する。
【図３０】複数のＩｏＴデバイスのフロー制御を実行するためのシステムの一実施形態。
【図３１】本発明の一実施形態による方法を例示する。
【図３２】アプリケーション属性、システム属性、及び優先度通知属性を管理するためのシステムの一実施形態を例示する。
【発明を実施するための形態】
【０００８】
以下の説明では、説明を目的として、以下に記載される本発明の実施形態の完全な理解を提供するために、多数の特定の詳細が示される。しかしながら、本発明の実施形態がこれらの特定の詳細のうちのいくつかを用いずに実施され得ることは、当業者には明らかである。他の例では、本発明の実施形態の根本的な原理を不明瞭にすることを避けるために、周知の構造及びデバイスをブロック図の形態で示す。
【０００９】
本発明の一実施形態は、新しいＩｏＴデバイス及びアプリケーションを設計及び構築するために開発者によって利用され得るモノのインターネット（ＩｏＴ）プラットフォームを含む。具体的には、一実施形態は、既定のネットワーキングプロトコルスタックを含むＩｏＴデバイス、及びＩｏＴデバイスがインターネットに連結されるＩｏＴハブ用の基本ハードウェア／ソフトウェアプラットフォームを含む。加えて、一実施形態は、ＩｏＴサービスを含み、これを通じてＩｏＴハブ及び接続されたＩｏＴデバイスが、以下に説明するようにアクセスされ、管理され得る。加えて、ＩｏＴプラットフォームの一実施形態は、ＩｏＴサービス、ハブ、及び接続されたデバイスにアクセスし、それらを構成する、ＩｏＴアプリケーション又はウェブアプリケーション（例えば、クライアントデバイス上で実行される）を含む。既存のオンライン小売業者及び他のウェブサイトオペレータは、本明細書に記載されたＩｏＴプラットフォームを利用して、既存のユーザベースに独自のＩｏＴ機能を容易に提供することができる。
【００１０】
図１Ａは、本発明の実施形態を実装することができるアーキテクチャプラットフォームの概要を例示する。具体的には、図示の実施形態は、それ自体インターネット２２０を介してＩｏＴサービス１２０に通信可能に連結されている中央ＩｏＴハブ１１０に、ローカル通信チャネル１３０を介して通信可能に連結された複数のＩｏＴデバイス１０１～１０５を含む。ＩｏＴデバイス１０１～１０５のそれぞれは、ローカル通信チャネル１３０のそれぞれを有効にするために、ＩｏＴハブ１１０と最初にペアリングすることができる（例えば、後述するペアリング技術を使用して）。一実施形態では、ＩｏＴサービス１２０は、各ユーザのＩｏＴデバイスから収集されたユーザアカウント情報及びデータを維持するためのエンドユーザデータベース１２２を含む。例えば、ＩｏＴデバイスがセンサ（例えば、温度センサ、加速度計、熱センサ、動作検出器など）を含む場合、データベース１２２は、ＩｏＴデバイス１０１～１０５により収集されるデータを記憶するように継続的に更新され得る。次いで、データベース１２２内に記憶されたデータは、ユーザデバイス１３５上にインストールされたＩｏＴアプリケーション又はブラウザを介して（又はデスクトップ若しくは他のクライアントコンピュータシステムを介して）エンドユーザに、かつウェブクライアント（例えば、ＩｏＴサービス１２０に加入しているウェブサイト１３０など）に、アクセス可能にされてもよい。
【００１１】
ＩｏＴデバイス１０１～１０５には、それ自体及びその周辺に関する情報を収集し、収集された情報を、ＩｏＴハブ１１０を介してＩｏＴサービス１２０、ユーザデバイス１３５、及び／又は外部ウェブサイト１３０に提供するための様々なタイプのセンサが備わっていてもよい。ＩｏＴデバイス１０１～１０５のうちのいくつかは、ＩｏＴハブ１１０を介して送信される制御コマンドに応答して、指定された機能を実行することができる。ＩｏＴデバイス１０１～１０５によって収集される情報の様々な具体例及び制御コマンドが

以下に提供される。以下に説明する一実施形態では、ＩｏＴデバイス１０１は、ユーザ選択を記録し、ユーザ選択をＩｏＴサービス１２０及び／又はウェブサイトに送信するように設計されたユーザ入力デバイスである。

【００１２】

一実施形態では、ＩｏＴハブ１１０は、４Ｇ（例えば、モバイルＷｉＭＡＸ、ＬＴＥ）又は５Ｇセルラーデータサービスなどのセルラーサービス１１５を介してインターネット２２０への接続を確立するセルラー無線を含む。代替的に、又は加えて、ＩｏＴハブ１１０は、ＷｉＦｉアクセスポイント又はルータ１１６を介してＷｉＦｉ接続を確立するためのＷｉＦｉ無線を含むことができ、これは、ＩｏＴハブ１１０をインターネットに（例えば、エンドユーザにインターネットサービスを提供するインターネットサービスプロバイダを介して）連結する。当然のことながら、本発明の基本的な原理は、特定のタイプの通信チャネル又はプロトコルに限定されないことに留意すべきである。

【００１３】

一実施形態では、ＩｏＴデバイス１０１～１０５は、電池電力で長期間（例えば、数年）動作することができる超低電力デバイスである。電力を節約するために、ローカル通信チャネル１３０は、Ｂｌｕｅｔｏｏｔｈ(登録商標) Ｌｏｗ Ｅｎｅｒｇｙ（ＬＥ）などの低電力無線通信技術を使用して実装することができる。この実施形態では、ＩｏＴデバイス１０１～１０５及びＩｏＴハブ１１０のそれぞれには、Ｂｌｕｅｔｏｏｔｈ(登録商標) ＬＥ無線及びプロトコルスタックが備わっている。

【００１４】

上述したように、一実施形態では、ＩｏＴプラットフォームは、ユーザが、接続されたＩｏＴデバイス１０１～１０５、ＩｏＴハブ１１０、及び／又はＩｏＴサービス１２０にアクセスし、それらを構成することを可能にする、ユーザデバイス１３５上で実行されるＩｏＴアプリケーション又はウェブアプリケーションを含む。一実施形態では、アプリケーション又はウェブアプリケーションは、そのユーザベースにＩｏＴ機能を提供するように、ウェブサイト１３０のオペレータによって設計されてもよい。例示したように、ウェブサイトは、各ユーザに関連するアカウント記録を含むユーザデータベース１３１を維持することができる。

【００１５】

図１Ｂは、複数のＩｏＴハブ１１０～１１１、１９０に対する追加の接続オプションを例示する。この実施形態では、単一のユーザが、単一のユーザ構内１８０（例えば、ユーザの自宅又はビジネス）にオンサイトでインストールされた複数のハブ１１０～１１１を有することができる。これは、例えば、ＩｏＴデバイス１０１～１０５のすべてを接続するのに必要な無線範囲を拡張するために行われ得る。上述したように、ユーザが複数のハブ１１０、１１１を有する場合、それらは、ローカル通信チャネル（例えば、Ｗｉｆｉ、イーサネット(登録商標)、電力線ネットワーキングなど）を介して接続されてもよい。一実施形態では、ハブ１１０～１１１のそれぞれは、セルラー１１５又はＷｉＦｉ １１６接続（図１Ｂには明示されていない）を介してＩｏＴサービス１２０への直接接続を確立することができる。代替的に、又は加えて、ＩｏＴハブ１１０などのＩｏＴハブのうちの１つは、「マスター」ハブとして機能することができ、これは、ＩｏＴハブ１１１などのユーザ構内１８０上の他のすべてのＩｏＴハブに接続性及び／又はローカルサービスを提供する（ＩｏＴハブ１１０とＩｏＴハブ１１１を接続する点線で示すように）。例えば、マスターＩｏＴハブ１１０は、ＩｏＴサービス１２０への直接接続を確立する唯一のＩｏＴハブであってもよい。一実施形態では、「マスター」ＩｏＴハブ１１０のみに、ＩｏＴサービス１２０への接続を確立するためのセルラー通信インタフェースが備わっている。このように、ＩｏＴサービス１２０と他のＩｏＴハブ１１１との間のすべての通信は、マスターＩｏＴハブ１１０を通って流れる。この役割において、マスターＩｏＴハブ１１０には、他のＩｏＴハブ１１１とＩｏＴサービス１２０との間で交換されるデータ（例えば、可能であれば、いくつかのデータ要求にローカルでサービスする）に対してフィルタリング動作を実行するための追加のプログラムコードが提供され得る。

【００１６】
ＩｏＴハブ１１０～１１１がどのように接続されていようとも、一実施形態では、ＩｏＴサービス１２０は、ハブをユーザと論理的に関連付け、取り付けられたＩｏＴデバイス１０１～１０５のすべてを、インストールされたアプリケーション１３５（及び／又はブラウザベースのインタフェース）を有するユーザデバイスを介してアクセス可能な、単一の包括的なユーザインタフェースの下に結合する。
【００１７】
この実施形態では、マスターＩｏＴハブ１１０及び１つ以上のスレーブＩｏＴハブ１１１は、ＷｉＦｉネットワーク１１６、イーサネット(登録商標)ネットワーク、及び／又は電力線通信（power-line communications）（ＰＬＣ）ネットワーキング（例えば、ネットワークの全部若しくは一部がユーザの電力線を介して実行される）とすることができる、ローカルネットワークを介して接続してもよい。加えて、ＩｏＴハブ１１０～１１１に対して、ＩｏＴデバイス１０１～１０５のそれぞれは、いくつか例を挙げると、ＷｉＦｉ、イーサネット(登録商標)、ＰＬＣ、又はＢｌｕｅｔｏｏｔｈ(登録商標) ＬＥなどの、任意のタイプのローカルネットワークチャネルを使用して、ＩｏＴハブ１１０～１１１と相互接続してもよい。
【００１８】
図１Ｂはまた、第２のユーザ構内１８１にインストールされたＩｏＴハブ１９０を示す。実質的に無制限の数のそのようなＩｏＴハブ１９０は、世界中のユーザ構内のＩｏＴデバイス１９１～１９２からデータを収集するようにインストールされ、構成され得る。一実施形態では、２つのユーザ構内１８０～１８１は、同じユーザに対して構成されてもよい。例えば、一方のユーザ構内１８０がユーザの基本的なホームであり、他方のユーザ構内１８１がユーザのバケーションホームであってもよい。そのような場合、ＩｏＴサービス１２０は、ＩｏＴハブ１１０～１１１、１９０をユーザと論理的に関連付け、取り付けられたすべてのＩｏＴデバイス１０１～１０５、１９１～１９２を、単一の包括的なユーザインタフェースの下に結合し、インストールされたアプリケーション１３５（及び／又はブラウザベースのインタフェース）を有するユーザデバイスを介してアクセス可能にする。
【００１９】
図２に例示するように、ＩｏＴデバイス１０１の例示的な実施形態は、プログラムコード及びデータ２０１～２０３を記憶するメモリ２１０と、プログラムコードを実行しデータを処理する低電力マイクロコントローラ２００とを含む。メモリ２１０は、ダイナミックランダムアクセスメモリ（dynamic random access memory）（ＤＲＡＭ）などの揮発性メモリであってもよいし、フラッシュメモリなどの不揮発性メモリであってもよい。一実施形態では、不揮発性メモリを永続記憶に使用し、揮発性メモリをプログラムコードの実行及びデータの実行に使用することができる。更に、メモリ２１０は、低電力マイクロコントローラ２００内に統合されてもよく、バス又は通信ファブリックを介して低電力マイクロコントローラ２００に連結されてもよい。本発明の根本的な原理は、メモリ２１０のいかなる特定の実装にも限定されない。
【００２０】
例示したように、プログラムコードは、ＩｏＴデバイス１０１のアプリケーション開発者によって利用され得る既定のビルディングブロックのセットを含む、ＩｏＴデバイス２０１及びライブラリコード２０２によって実行される特定用途向けの機能セットを定義するアプリケーションプログラムコード２０３を含むことができる。一実施形態では、ライブラリコード２０２は、各ＩｏＴデバイス１０１とＩｏＴハブ１１０との間の通信を可能にするための通信プロトコルスタック２０１などのＩｏＴデバイスを実装するために必要とされる基本機能のセットを含む。上述したように、一実施形態では、通信プロトコルスタック２０１は、Ｂｌｕｅｔｏｏｔｈ(登録商標) ＬＥプロトコルスタックを含む。この実施形態では、Ｂｌｕｅｔｏｏｔｈ(登録商標) ＬＥ無線機及びアンテナ２０７は、低電力マイクロコントローラ２００内に統合されてもよい。しかしながら、本発明の基本原理

は、いかなる特定の通信プロトコルにも限定されない。

【００２１】

図２に示す特定の実施形態はまた、ユーザ入力を受信し、ユーザ入力を低電力マイクロコントローラに提供する複数の入力デバイス又はセンサ２１０を含み、低電力マイクロコントローラは、アプリケーションコード２０３及びライブラリコード２０２に従ってユーザ入力を処理する。一実施形態では、入力デバイスのそれぞれは、エンドユーザにフィードバックを提供するＬＥＤ　２０９を含む。

【００２２】

加えて、例示した実施形態は、低電力マイクロコントローラに電力を供給するための電池２０８を含む。一実施形態では、非充電式コイン型電池が使用される。しかしながら、別の実施形態では、統合された充電式電池を使用することができる（例えば、交流電源（図示せず）にＩｏＴデバイスを接続することによって再充電可能）。

【００２３】

オーディオを発生するためのスピーカ２０５も設けられている。一実施形態では、低電力マイクロコントローラ２９９は、スピーカ２０５上にオーディオを発生するために圧縮されたオーディオストリーム（例えば、ＭＰＥＧ－４／アドバンストオーディオコーディング（Advanced Audio Coding）（ＡＡＣ）ストリーム）を復号するための、オーディオ復号ロジックを含む。代替的に、低出力マイクロコントローラ２００及び／又はアプリケーションコード／データ２０３が、ユーザが入力デバイス２１０を介して選択を入力すると、エンドユーザに口頭のフィードバックを提供するための、デジタルでサンプリングされたオーディオスニペットを含むことができる。

【００２４】

一実施形態では、ＩｏＴデバイス１０１が設計される特定用途に基づいて、１つ以上の他の／代替のＩ／Ｏデバイス又はセンサ２５０が、ＩｏＴデバイス１０１に含まれてもよい。例えば、温度、圧力、湿度などを測定するために環境センサを含めることができる。ＩｏＴデバイスがセキュリティデバイスとして使用される場合には、セキュリティセンサ及び／又はドアロックオープナが含まれてもよい。当然のことながら、これらの例は、単に例示のために提供されている。本発明の基本原理は、いかなる特定のタイプのＩｏＴデバイスにも限定されない。実際に、ライブラリコード２０２が備わった低電力マイクロコントローラ２００の高度にプログラマブルな性質を考慮すると、アプリケーション開発者は、新しいアプリケーションコード２０３及び新しいＩ／Ｏデバイス２５０を容易に開発して、実質的に任意のタイプのＩｏＴアプリケーションのために低電力マイクロコントローラとインタフェースをとることができる。

【００２５】

一実施形態では、低電力マイクロコントローラ２００はまた、通信を暗号化するための、及び／又は署名を生成するための暗号鍵を記憶するための安全な鍵ストアを含む。代替的に、鍵は、加入者識別モジュール（ＳＩＭ）内に確保されてもよい。

【００２６】

一実施形態では、実質的に電力を消費していない超低電力状態からＩｏＴデバイスを起動させるために、ウェイクアップ受信機２０７が含まれる。一実施形態では、ウェイクアップ受信機２０７は、図３に示すように、ＩｏＴハブ１１０上に構成されたウェイクアップ送信機３０７から受信されたウェイクアップ信号に応答して、ＩｏＴデバイス１０１をこの低電力状態から出させるように構成される。具体的には、一実施形態では、送信機３０７と受信機２０７は共に、テスラコイルなどの電気共振トランス回路を形成する。動作中、ハブ１１０が非常に低い電力状態からＩｏＴデバイス１０１を復帰させる必要がある場合、エネルギは送信機３０７から受信機２０７への無線周波数信号を介して送信される。エネルギ移動の理由で、ＩｏＴデバイス１０１は、それが低電力状態にあるときには、ハブからの信号を継続的に「聞く」必要がないので、実質的に電力を消費しないように構成することができる（ネットワーク信号を介してデバイスを起動させることができる、ネットワークプロトコルの場合と同様に）。むしろ、ＩｏＴデバイス１０１のマイクロコン

トローラ２００は、送信機３０７から受信機２０７に電気的に送信されたエネルギを使用することによって、事実上パワーダウンされた後にウェイクアップするように構成することができる。

【００２７】

図３に例示するように、ＩｏＴハブ１１０はまた、プログラムコード及びデータ３０５を記憶するためのメモリ３１７と、プログラムコードを実行しデータを処理するためのマイクロコントローラなどのハードウェアロジック３０１とを含む。広域ネットワーク（wide area network）（ＷＡＮ）インタフェース３０２及びアンテナ３１０は、ＩｏＴハブ１１０をセルラーサービス１１５に連結する。代替的に、上述したように、ＩｏＴハブ１１０は、ローカルエリアネットワーク通信チャネルを確立するためにＷｉＦｉインタフェース（及びＷｉＦｉアンテナ）又はイーサネット(登録商標)インタフェースなどのローカルネットワークインタフェース（図示せず）を含むこともできる。一実施形態では、ハードウェアロジック３０１はまた、通信を暗号化するための、及び／又は署名を生成／検証するための暗号鍵を記憶するための安全な鍵ストアを含む。代替的に、鍵は、加入者識別モジュール（ＳＩＭ）内に確保されてもよい。

【００２８】

ローカル通信インタフェース３０３及びアンテナ３１１は、ＩｏＴデバイス１０１～１０５のそれぞれとのローカル通信チャネルを確立する。上述したように、一実施形態では、ローカル通信インタフェース３０３／アンテナ３１１はＢｌｕｅｔｏｏｔｈ(登録商標)ＬＥ規格を実装する。しかしながら、本発明の根底にある原理は、ＩｏＴデバイス１０１～１０５とのローカル通信チャネルを確立するためのいかなる特定のプロトコルにも限定されない。図３においては別個のユニットとして示されているが、ＷＡＮインタフェース３０２及び／又はローカル通信インタフェース３０３は、ハードウェアロジック３０１と同じチップ内に組み込まれてもよい。

【００２９】

一実施形態では、プログラムコード及びデータは、ローカル通信インタフェース３０３及びＷＡＮインタフェース３０２を介して通信するための別個のスタックを含むことができる通信プロトコルスタック３０８を含む。加えて、デバイスペアリングプログラムコード及びデータ３０６は、ＩｏＴハブを新しいＩｏＴデバイスとペアリングすることができるようにメモリに記憶され得る。一実施形態では、各新しいＩｏＴデバイス１０１～１０５には、ペアリングプロセス中にＩｏＴハブ１１０に通信される一意的なコードが割り当てられる。例えば、一意的なコードは、ＩｏＴデバイス上のバーコードに組み込まれてもよく、かつバーコードリーダ１０６によって読み取られてもよく、又はローカル通信チャネル１３０を介して通信されてもよい。別の実施形態では、一意的なＩＤコードがＩｏＴデバイスに磁気的に組み込まれ、ＩｏＴハブは、無線周波数ＩＤ（radio frequency ID）（ＲＦＩＤ）又は近距離通信（near field communication）（ＮＦＣ）センサなどの磁気センサを有し、ＩｏＴデバイス１０１がＩｏＴハブ１１０の数インチ内で移動するとき、コードを検出する。

【００３０】

一実施形態では、一意的なＩＤが通信されると、ＩｏＴハブ１１０は、ローカルデータベース（図示せず）に問い合わせること、コードが許容可能であることを検証するためにハッシュを実行すること、並びに／又はＩｏＴサービス１２０、ユーザデバイス１３５、及び／若しくはウェブサイト１３０と通信することによって、一意的なＩＤを検証して、ＩＤコードの妥当性を確認することができる。妥当性が確認されると、一実施形態では、ＩｏＴハブ１１０は、ＩｏＴデバイス１０１をペアリングし、メモリ３１７（これは、上述したように、不揮発性メモリを含むことができる）にペアリングデータを記憶する。ペアリングが完了すると、ＩｏＴハブ１１０は、本明細書に記載の様々なＩｏＴ機能を実行するためにＩｏＴデバイス１０１と接続することができる。

【００３１】

一実施形態では、ＩｏＴサービス１２０を実行する組織は、開発者が新しいＩｏＴサー

ビスを容易に設計できるように、ＩｏＴハブ１１０及び基本ハードウェア／ソフトウェアプラットフォームを提供することができる。具体的には、ＩｏＴハブ１１０に加えて、開発者には、ハブ１１０内で実行されるプログラムコード及びデータ３０５を更新するためのソフトウェア開発キット（software development kit）（ＳＤＫ）が提供されてもよい。加えて、ＩｏＴデバイス１０１については、ＳＤＫは、様々な異なるタイプのアプリケーション１０１の設計を容易にするために、ベースのＩｏＴハードウェア（例えば、低電力マイクロコントローラ２００及び図２に示す他の構成要素）用に設計された広範なライブラリコード２０２のセットを含んでもよい。一実施形態では、ＳＤＫは、開発者がＩｏＴデバイスの入力と出力を指定するだけでよいグラフィカルデザインインタフェースを含む。ＩｏＴデバイス１０１がハブ１１０及びサービス１２０に接続することを可能にする通信スタック２０１を含むネットワーキングコードはすべて、開発者のために既に配置されている。加えて、一実施形態では、ＳＤＫは、モバイルデバイス（例えば、ｉＰｈｏｎｅ（登録商標）及びＡｎｄｒｏｉｄ（登録商標）デバイス）用のアプリケーションの設計を容易にするライブラリコードベースも含む。

【００３２】

一実施形態では、ＩｏＴハブ１１０は、ＩｏＴデバイス１０１～１０５とＩｏＴサービス１２０との間のデータの連続的な双方向ストリームを管理する。ＩｏＴデバイス１０１～１０５への／からの更新がリアルタイムで要求される状況（例えば、ユーザがセキュリティデバイス又は環境測定値の現在の状態を見る必要がある状況）では、ＩｏＴハブは、ユーザデバイス１３５及び／又は外部のウェブサイト１３０に定期的な更新を提供するためにオープンＴＣＰソケットを維持することができる。更新を提供するために使用される特定のネットワーキングプロトコルは、基本用途のニーズに基づいて調整されてもよい。例えば、連続的な双方向ストリームを有することが理にかなっていない可能性がある場合、必要なときに情報を収集するために単純な要求／応答プロトコルを使用することができる。

【００３３】

一実施形態では、ＩｏＴハブ１１０及びＩｏＴデバイス１０１～１０５の両方が、ネットワークを介して自動的に更新可能である。具体的には、ＩｏＴハブ１１０について新しい更新が利用可能であるとき、ＩｏＴサービス１２０から更新を自動的にダウンロードしてインストールすることができる。それは、古いプログラムコードを交換する前に、まず、更新されたコードをローカルメモリにコピーし、実行して、更新を検証し得る。同様に、ＩｏＴデバイス１０１～１０５のそれぞれについて更新が利用可能である場合、更新は、ＩｏＴハブ１１０によって最初にダウンロードされ、ＩｏＴデバイス１０１～１０５のそれぞれにプッシュアウトされてもよい。各ＩｏＴデバイス１０１～１０５は、ＩｏＴハブに関して上述したのと同様の方法で更新を適用し、更新の結果をＩｏＴハブ１１０に報告することができる。更新が成功した場合、ＩｏＴハブ１１０は、更新をそのメモリから削除し、（例えば、各ＩｏＴデバイスについての新しい更新を確認し続けることができるように）それぞれのＩｏＴデバイスにインストールされているコードの最新バージョンを記録することができる。

【００３４】

一実施形態では、ＩｏＴハブ１１０は、Ａ／Ｃ電力を介して給電される。具体的には、ＩｏＴハブ１１０は、Ａ／Ｃ電源コードを介して供給されるＡ／Ｃ電圧をより低いＤＣ電圧に変換するための変圧器を備えた電源ユニット３９０を含むことができる。

【００３５】

図４Ａは、ＩｏＴシステムを使用してユニバーサル遠隔制御操作を実行するための、本発明の一実施形態を例示する。具体的には、この実施形態では、ＩｏＴデバイス１０１～１０３のセットには、（ほんの数例を挙げると）空気調節装置／ヒータ４３０、照明システム４３１、及び視聴覚機器４３２を含む、様々な異なるタイプの電子機器を制御する遠隔制御コードを送信するための、赤外線（infrared）（ＩＲ）及び／又は無線周波数（radio frequency）（ＲＦ）ブラスタ４０１～４０３がそれぞれ備わっている。図４Ａに示

される実施形態では、ＩｏＴデバイス１０１～１０３にはまた、以下に説明するように、それらが制御するデバイスの動作を検出するためのセンサ４０４～４０６がそれぞれ備わっている。

【００３６】

例えば、ＩｏＴデバイス１０１におけるセンサ４０４は、現在の温度／湿度を検知し、それに応じて、現在の所望の温度に基づき空気調節装置／ヒータ４３０を制御するための温度及び／又は湿度センサであってもよい。この実施形態では、空気調節装置／ヒータ４３０は、遠隔制御デバイス（典型的には、それ自体が温度センサをその中に組み込んだ遠隔制御装置）を介して制御されるように設計されるものである。一実施形態では、ユーザは、ユーザデバイス１３５上にインストールされたアプリケーション又はブラウザを介して、所望の温度をＩｏＴハブ１１０に提供する。ＩｏＴハブ１１０上で実行される制御ロジック４１２は、センサ４０４から現在の温度／湿度データを受信し、それに応じて、所望の温度／湿度に従ってＩＲ／ＲＦブラスタ４０１を制御するように、ＩｏＴデバイス１０１にコマンドを送信する。例えば、温度が所望の温度未満である場合、制御ロジック４１２は、温度を上げるように、ＩＲ／ＲＦブラスタ４０１を介して空気調節装置／ヒータにコマンドを送信してもよい（例えば、空気調節装置をオフにすることか、又はヒータをオンにすることのいずれかによって）。コマンドは、ＩｏＴハブ１１０上のデータベース４１３に記憶された必要な遠隔制御コードを含んでもよい。代替的に、又は加えて、ＩｏＴサービス４２１は、指定されたユーザ選好及び記憶された制御コード４２２に基づき電子機器４３０～４３２を制御するために、制御ロジック４２１を実装してもよい。

【００３７】

例示した実施例におけるＩｏＴデバイス１０２は、照明４３１を制御するために使用される。具体的には、ＩｏＴデバイス１０２のセンサ４０５は、照明設備４３１（又は他の照明装置）によってもたらされている光の現在の輝度を検出するように構成された光センサ又は光検出器であってもよい。ユーザは、ユーザデバイス１３５を介して、ＩｏＴハブ１１０に所望の照明レベル（オン又はオフの表示を含む）を指定してもよい。それに応答して、制御ロジック４１２は、照明４３１の現在の輝度レベルを制御するように、ＩＲ／ＲＦブラスタ４０２にコマンドを送信する（例えば、現在の輝度が低すぎる場合は照明を明るくするか、若しくは現在の輝度が高すぎる場合は照明を暗くするか、又は単純に照明をオン若しくはオフにする）。

【００３８】

例示した実施例におけるＩｏＴデバイス１０３は、視聴覚機器４３２（例えば、テレビ、Ａ／Ｖ受信機、ケーブル／衛星受信機、ＡｐｐｌｅＴＶＰＰ（商標）ＰＰなど）を制御するように構成される。ＩｏＴデバイス１０３のセンサ４０６は、現在の周囲音量レベルを検出するためのオーディオセンサ（例えば、マイクロホン及び関連ロジック）、並びに／又はテレビによって生成された光に基づき、（例えば、指定されたスペクトル内の光を測定することによって）テレビがオンであるか、それともオフであるかを検出するための光センサであってもよい。代替的に、センサ４０６は、検出された温度に基づき、オーディオ機器がオンであるか、それともオフであるかを検出するための、視聴覚機器に接続された温度センサを含んでもよい。この場合も、ユーザデバイス１３５を介したユーザ入力に応答して、制御ロジック４１２は、ＩｏＴデバイス１０３のＩＲブラスタ４０３を介して視聴覚機器にコマンドを送信してもよい。

【００３９】

上記が本発明の一実施形態の単なる例示した実施例であることに留意すべきである。本発明の基本原理は、ＩｏＴデバイスによって制御されるいかなる特定のタイプのセンサ又は機器にも限定されない。

【００４０】

ＩｏＴデバイス１０１～１０３がＢｌｕｅｔｏｏｔｈ(登録商標)　ＬＥ接続を介してＩｏＴハブ１１０に連結される実施形態では、センサデータ及びコマンドは、Ｂｌｕｅｔｏｏｔｈ(登録商標)　ＬＥチャネルを介して送信される。しかしながら、本発明の基本原理

は、Ｂｌｕｅｔｏｏｔｈ(登録商標)　ＬＥ又はいずれの他の通信標準にも限定されない。
【００４１】
一実施形態では、電子機器のそれぞれを制御するために必要とされる制御コードは、ＩｏＴハブ１１０上のデータベース４１３及び／又はＩｏＴサービス１２０上のデータベース４２２に記憶される。図４Ｂに例示するように、制御コードは、ＩｏＴサービス１２０上で維持される異なる機器に対して、制御コード４２２のマスターデータベースからＩｏＴハブ１１０に提供されてもよい。エンドユーザは、ユーザデバイス１３５上で実行されるアプリケーション又はブラウザを介して制御される電子（又は他の）機器のタイプを指定してもよく、それに応答して、ＩｏＴハブ上の遠隔制御コード学習モジュール４９１は、ＩｏＴサービス１２０上の遠隔制御コードデータベース４９２から、必要とされるＩＲ／ＲＦコードを取得してもよい（例えば、一意的なＩＤを有する各電子機器を識別する）。
【００４２】
加えて、一実施形態では、ＩｏＴハブ１１０には、遠隔制御コード学習モジュール４９１が、電子機器と共に提供された元の遠隔制御装置４９５から直接新しい遠隔制御コードを「学習」することを可能にする、ＩＲ／ＲＦインタフェース４９０が備わっている。例えば、空気調節装置４３０と共に提供された元の遠隔制御装置の制御コードが、遠隔制御データベースに含まれていない場合、ユーザは、ユーザデバイス１３５上のアプリケーション／ブラウザを介してＩｏＴハブ１１０と対話して、元の遠隔制御装置によって生成される様々な制御コードをＩｏＴハブ１１０に教えてもよい（例えば、温度を上げる、温度を下げるなど）。遠隔制御コードが学習されると、それらは、ＩｏＴハブ１１０上の制御コードデータベース４１３に記憶されてもよく、かつ／又は中央遠隔制御コードデータベース４９２に含められるように、ＩｏＴサービス１２０に送り返されてもよい（続いて、同じ空気調節装置ユニット４３０を有する他のユーザによって使用されてもよい）。
【００４３】
一実施形態では、ＩｏＴデバイス１０１～１０３のそれぞれは、極端に小さいフォームファクタを有し、両面テープ、小さい釘、磁気アタッチメントなどを使用して、それらの対応する電子機器４３０～４３２の上又は付近に取り付けられてもよい。空気調節装置４３０などの１つの機器を制御するために、ＩｏＴデバイス１０１を十分に離して配置し、センサ４０４が自宅内の周囲温度を正確に測定することができるようにすることが望ましい（例えば、空気調節装置上に直接ＩｏＴデバイスを配置すると、温度測定値は、空気調節装置が作動しているときは低すぎになり、ヒータが作動しているときは高すぎになるであろう）。対照的に、照明を制御するために使用されるＩｏＴデバイス１０２は、センサ４０５が現在の照明レベルを検出するために、照明設備４３１の上又は付近に配置されてもよい。
【００４４】
記載される一般的な制御機能を提供することに加えて、ＩｏＴハブ１１０及び／又はＩｏＴサービス１２０の一実施形態は、各電子機器の現在の状態に関連した通知をエンドユーザに送信する。通知は、テキストメッセージ及び／又はアプリケーション特有の通知であってもよく、次いで、通知は、ユーザのモバイルデバイス１３５のディスプレイ上に表示されてもよい。例えば、ユーザの空気調節装置が長期間オンであるが温度が変化していない場合、ＩｏＴハブ１１０及び／又はＩｏＴサービス１２０は、空気調節装置が適切に機能していないという通知をユーザに送信してもよい。ユーザが自宅におらず（このことは、動作センサを介して検出されてもよく、若しくはユーザの現在の検出された位置に基づいてもよい）、センサ４０６が、視聴覚機器４３０がオンであることを示すか、又はセンサ４０５が、照明がオンであることを示す場合、ユーザが視聴覚機器４３２及び／又は照明４３１をオフにすることを希望するか尋ねる通知がユーザに送信されてもよい。同じタイプの通知が、任意の機器のタイプに対して送信されてもよい。
【００４５】
ユーザが通知を受信すると、彼／彼女は、ユーザデバイス１３５上のアプリケーション

又はブラウザを介して電子機器４３０～４３２を遠隔制御してもよい。一実施形態では、ユーザデバイス１３５は、タッチスクリーンデバイスであり、アプリケーション又はブラウザは、機器４３０～４３２を制御するためのユーザが選択可能なボタンを含む遠隔制御装置の画像を表示する。通知を受信した後、ユーザは、グラフィカル遠隔制御装置を開き、様々な異なる機器をオフにするか、又は調節してもよい。ＩｏＴサービス１２０を介して接続されている場合、ユーザの選択は、ＩｏＴサービス１２０からＩｏＴハブ１１０に転送されてもよく、ＩｏＴハブ１１０は、次いで制御ロジック４１２を介して機器を制御することになる。代替的に、ユーザ入力は、ユーザデバイス１３５からＩｏＴハブ１１０に直接送信されてもよい。

【００４６】

一実施形態では、ユーザは、電子機器４３０～４３２に対して様々な自動制御機能を実行するように、ＩｏＴハブ１１０上の制御ロジック４１２をプログラミングしてもよい。上記の所望の温度、輝度レベル、及び音量レベルを維持することに加えて、制御ロジック４１２は、ある特定の条件が検出された場合に電子機器を自動的にオフにしてもよい。例えば、制御ロジック４１２が、ユーザが自宅にいないこと、及び空気調節装置が機能していないことを検出する場合、制御ロジック４１２は、空気調節装置を自動的にオフにしてもよい。同様に、ユーザが自宅におらず、センサ４０６が、視聴覚機器４３０がオンであることを示すか、又はセンサ４０５が、照明がオンであることを示す場合、制御ロジック４１２は、視聴覚機器及び照明をそれぞれオフにするように、ＩＲ／ＲＦブラスタ４０３及び４０２を介してコマンドを自動的に送信してもよい。

【００４７】

図５は、電子機器５３０及び５３１を監視するためのセンサ５０３及び５０４が備わった、ＩｏＴデバイス１０４及び１０５の追加の実施形態を例示する。具体的には、この実施形態のＩｏＴデバイス１０４は、コンロがオンのままであるときを検出するためにコンロ５３０の上又は付近に配置されてもよい、温度センサ５０３を含む。一実施形態では、ＩｏＴデバイス１０４は、温度センサ５０３によって測定された現在の温度をＩｏＴハブ１１０及び／又はＩｏＴサービス１２０に送信する。コンロが閾値期間を超えてオンであることが検出される場合（例えば、測定された温度に基づき）、制御ロジック５１２は、コンロ５３０がオンであることをユーザに通知する通知を、エンドユーザのデバイス１３５に送信してもよい。加えて、一実施形態では、ＩｏＴデバイス１０４は、ユーザからの命令を受信することに応答して、又は自動的に（制御ロジック５１２がそうするようにユーザによってプログラミングされる場合）、のいずれかによって、コンロをオフにするための制御モジュール５０１を含んでもよい。一実施形態では、制御ロジック５０１は、コンロ５３０への電気又はガスを遮断するためのスイッチを備える。しかしながら、他の実施形態では、制御ロジック５０１は、コンロ自体内に統合されてもよい。

【００４８】

図５はまた、洗濯機及び／又は乾燥機などのある特定のタイプの電子機器の動作を検出するための動作センサ５０４を有する、ＩｏＴデバイス１０５を例示する。使用され得る別のセンサは、周囲の音量レベルを検出するためのオーディオセンサ（例えば、マイクロホン及びロジック）である。上記の他の実施形態のように、この実施形態は、ある特定の指定された条件が満たされた場合、エンドユーザに通知を送信してもよい（例えば、動作が長期間検出され、洗濯機／乾燥機がオフになっていないことを示す場合）。図５に示されないが、ＩｏＴデバイス１０５にはまた、自動的に、かつ／又はユーザ入力に応答して、（例えば、電気／ガスをオフに切り替えることによって）洗濯機／乾燥機５３１をオフにするための制御モジュールが備わっていてもよい。

【００４９】

一実施形態では、制御ロジック及びスイッチを有する第１のＩｏＴデバイスは、ユーザの自宅内のすべての電力をオフにするように構成されてもよく、制御ロジック及びスイッチを有する第２のＩｏＴデバイスは、ユーザの自宅内のすべてのガスをオフにするように構成されてもよい。次いで、センサを有するＩｏＴデバイスは、ユーザの自宅内の電気又

はガス駆動の機器の上又は付近に位置付けられてもよい。特定の機器がオンのままである（例えば、コンロ５３０）ことをユーザが通知された場合、ユーザは、自宅内のすべての電気又はガスをオフにするコマンドを送信して、損害を防止してもよい。代替的に、ＩｏＴハブ１１０及び／又はＩｏＴサービス１２０の制御ロジック５１２は、そのような状況において電気又はガスを自動的にオフにするように構成されてもよい。

【００５０】

一実施形態では、ＩｏＴハブ１１０及びＩｏＴサービス１２０は、周期的な間隔で通信する。ＩｏＴサービス１２０が、ＩｏＴハブ１１０への接続が切れていることを検出する場合（例えば、指定された継続時間、ＩｏＴハブからの要求又は応答を受信していないことによって）、ＩｏＴサービス１２０は、この情報をエンドユーザのデバイス１３５に通信することになる（例えば、テキストメッセージ又はアプリケーション特有の通知を送信することによって）。

通信のための装置及び方法

中間デバイスを通じたデータ

【００５１】

上述したように、Ｂｌｕｅｔｏｏｔｈ(登録商標)　ＬＥなどのＩｏＴデバイスを相互接続するために使用される無線技術は概して、近距離技術であるため、ＩｏＴ実装のためのハブがＩｏＴデバイスの範囲外にある場合、ＩｏＴデバイスは、ＩｏＴハブにデータを送信することができない（逆もまた同様）。

【００５２】

この欠陥に対処するために、本発明の一実施形態は、モバイルデバイスが範囲内にあるとき、１つ以上のモバイルデバイスと周期的に接続するために、ＩｏＴハブの無線範囲外にあるＩｏＴデバイスのための機構を提供する。いったん接続されると、ＩｏＴデバイスは、ＩｏＴハブに提供される必要がある任意のデータをモバイルデバイスに送信することができ、次いでモバイルデバイスは、ＩｏＴハブにデータを転送する。

【００５３】

図６に例示するように、一実施形態は、ＩｏＴハブ１１０と、ＩｏＴハブ１１０の範囲外にあるＩｏＴデバイス６０１と、モバイルデバイス６１１とを含む。範囲外のＩｏＴデバイス６０１は、データを収集及び通信することが可能な任意の形態のＩｏＴデバイスを含んでもよい。例えば、ＩｏＴデバイス６０１は、冷蔵庫内の利用可能な食料品、食料品を消費するユーザ、及び現在の温度を監視するように、冷蔵庫内に構成されたデータ収集デバイスを備えてもよい。当然のことながら、本発明の基本原理は、いかなる特定のタイプのＩｏＴデバイスにも限定されない。本明細書に記載される技術は、ほんの数例を挙げると、スマートメータ、コンロ、洗濯機、乾燥機、照明システム、ＨＶＡＣシステム、及び視聴覚機器に関するデータを収集及び送信するために使用されるデバイスを含む、任意のタイプのＩｏＴデバイスを使用して実装されてもよい。

【００５４】

更に、動作中のモバイルデバイスである、図６に例示するＩｏＴデバイス６１１は、データを通信及び記憶することが可能な任意の形態のモバイルデバイスであってもよい。例えば、一実施形態では、モバイルデバイス６１１は、本明細書に記載される技術を促進するために、アプリケーションがその上にインストールされたスマートフォンである。別の実施形態では、モバイルデバイス６１１は、ネックレス若しくはブレスレットに取り付けられた通信トークン、スマートウォッチ、又はフィットネスデバイスなど、装着可能なデバイスを含む。装着可能なトークンは、スマートフォンデバイスを所有しない高齢のユーザ又は他のユーザにとって特に有用であり得る。

【００５５】

動作中、範囲外のＩｏＴデバイス６０１は、モバイルデバイス６１１との接続性を周期的又は連続的にチェックしてもよい。接続を確立した後（例えば、ユーザが冷蔵庫の近くを移動する結果として）、ＩｏＴデバイス６０１上の任意の収集されたデータ６０５が、モバイルデバイス６１１上の一時データリポジトリ６１５に自動的に送信される。一実施

形態では、ＩｏＴデバイス６０１及びモバイルデバイス６１１は、ＢＴＬＥなどの低電力無線標準を使用して、ローカル無線通信チャネルを確立する。そのような場合、モバイルデバイス６１１は、既知のペアリング技術を使用してＩｏＴデバイス６０１と最初にペアリングされてもよい。

【００５６】

いったんデータが一時データリポジトリに伝送されると、モバイルデバイス６１１は、ＩｏＴハブ１１０との通信が確立されるとデータを送信する（例えば、ユーザがＩｏＴハブ１１０の範囲内を歩くとき）。次いで、ＩｏＴハブは、中央データリポジトリ４１３にデータを記憶してもよく、かつ／又はインターネット上で、１つ以上のサービス及び／若しくは他のユーザデバイスにデータを送信してもよい。一実施形態では、モバイルデバイス６１１は、異なるタイプの通信チャネルを使用して、ＩｏＴハブ１１０にデータを提供してもよい（潜在的に、ＷｉＦｉなどのより高出力の通信チャネル）。

【００５７】

範囲外のＩｏＴデバイス６０１、モバイルデバイス６１１、及びＩｏＴハブはすべて、本明細書に記載される技術を実装するためのプログラムコード及び／又はロジックにより構成されてもよい。図７に例示するように、例えば、本明細書に記載される動作を実行するために、ＩｏＴデバイス６０１は、中間接続ロジック及び／又はアプリケーションにより構成されてもよく、モバイルデバイス６１１は、中間接続ロジック／アプリケーションにより構成されてもよく、ＩｏＴハブ１１０は、中間接続ロジック／アプリケーション７２１により構成されてもよい。各デバイス上の中間接続ロジック／アプリケーションは、ハードウェア、ソフトウェア、又はこれらの任意の組み合わせで実装されてもよい。一実施形態では、ＩｏＴデバイス６０１の中間接続ロジック／アプリケーション７０１は、モバイルデバイス上の中間接続ロジック／アプリケーション７１１（デバイスアプリケーションとして実装されてもよい）との接続を検索及び確立して、一時データリポジトリ６１５にデータを伝送する。次いで、モバイルデバイス６１１上の中間接続ロジック／アプリケーション７０１は、中央データリポジトリ４１３にデータを記憶するＩｏＴハブ上の中間接続ロジック／アプリケーションに、データを転送する。

【００５８】

図７に例示するように、各デバイス上の中間接続ロジック／アプリケーション７０１、７１１、７２１は、手元のアプリケーションに基づき構成されてもよい。例えば、冷蔵庫に関して、接続ロジック／アプリケーション７０１は、周期的ベースで少数のパケットを送信するだけでよい。他のアプリケーション（例えば、温度センサ）に対して、接続ロジック／アプリケーション７０１は、より頻繁な更新を送信する必要があり得る。

【００５９】

モバイルデバイス６１１よりはむしろ、一実施形態では、ＩｏＴデバイス６０１が、ＩｏＴハブ１１０の範囲内に位置する１つ以上の中間ＩｏＴデバイスとの無線接続を確立するように構成されてもよい。この実施形態では、ＩｏＴハブの範囲外の任意のＩｏＴデバイス６０１が、他のＩｏＴデバイスを使用して「チェーン」を形成することによってハブにリンクされてもよい。

【００６０】

加えて、簡潔にするために、単一のモバイルデバイス６１１のみが図６～７に例示されるが、一実施形態では、異なるユーザの複数のそのようなモバイルデバイスは、ＩｏＴデバイス６０１と通信するように構成されてもよい。更に、同じ技術が、複数の他のＩｏＴデバイスに対して実装されてもよく、それにより、自宅全体にわたって中間デバイスデータ収集システムを形成する。

【００６１】

更に、一実施形態では、本明細書に記載される技術は、様々な異なるタイプの関連データを収集するために使用されてもよい。例えば、一実施形態では、モバイルデバイス６１１がＩｏＴデバイス６０１と接続するたびに、ユーザの識別が、収集されたデータ６０５と共に含まれてもよい。このようにして、ＩｏＴシステムは、自宅内の異なるユーザの挙

動を追跡するために使用されてもよい。例えば、冷蔵庫内で使用される場合、収集されたデータ６０５は、冷蔵庫のそばを通る各ユーザ、冷蔵庫を開ける各ユーザ、及び各ユーザによって消費される特定の食料品の識別を含んでもよい。異なるタイプのデータが、他のタイプのＩｏＴデバイスから収集されてもよい。このデータを使用して、システムは、例えば、どのユーザが衣服を洗濯するのか、どのユーザが所与の日にテレビを観るのか、各ユーザが就寝及び起床する時間などを判定することが可能である。次いで、このクラウドソースデータのすべてが、ＩｏＴハブのデータリポジトリ４１３内にコンパイルされてもよく、かつ／又は外部サービス若しくはユーザに転送されてもよい。

【００６２】

本明細書に記載される技術の別の有益な用途は、補助を必要とし得る高齢のユーザを監視するためのものである。このアプリケーションに関して、モバイルデバイス６１１は、ユーザの自宅の異なる室内の情報を収集するために、高齢のユーザによって装着された非常に小型のトークンであってもよい。ユーザが冷蔵庫を開けるたびに、例えば、このデータは、収集されたデータ６０５と共に含まれ、トークンを介してＩｏＴハブ１１０に伝送される。次いで、ＩｏＴハブは、１つ以上の外部ユーザ（例えば、高齢のユーザを世話する子供又は他の個人）にデータを提供してもよい。データが指定された期間（例えば、１２時間）収集されていない場合、これは、高齢のユーザが自宅を動き回っていない、かつ／又は冷蔵庫を開けていないことを意味する。次いで、ＩｏＴハブ１１０又はＩｏＴハブに接続された外部サービスは、これらの他の個人にアラート通知を送信し、彼らに高齢のユーザを確認するべきであることを通知してもよい。加えて、収集されたデータ６０５は、ユーザによって消費されている食品、並びに食料品店に行くことが必要であるかどうか、高齢のユーザがテレビを観ているかどうか、及びどれほど頻繁に観ているか、高齢のユーザが衣服を洗濯する頻度などの他の関連情報を含んでもよい。

【００６３】

別の実装例において、洗濯機、冷蔵庫、ＨＶＡＣシステムなどの電子デバイスに問題がある場合、収集されたデータは、交換される必要がある部品の指示を含んでもよい。そのような場合、通知は、問題を解決するための要求と共に技術者に送信されてもよい。次いで、技術者は、必要とされる交換部品を持って自宅に到着し得る。

【００６４】

本発明の一実施形態に従った方法が図８に例示される。本方法は、上記のアーキテクチャとの関連で実装され得るが、いかなる特定のアーキテクチャにも限定されない。

【００６５】

８０１において、ＩｏＴハブの範囲外にあるＩｏＴデバイスは、データ（例えば、冷蔵庫の扉の開放、使用された食料品など）を周期的に収集する。８０２において、ＩｏＴデバイスは、モバイルデバイスとの接続性を周期的又は連続的にチェックする（例えば、ＢＴＬＥ標準によって指定されたものなど、接続を確立するための標準的なローカル無線技術を使用して）。８０２において、モバイルデバイスへの接続が確立され、判定された場合、８０３において、収集されたデータは、８０３においてモバイルデバイスに伝送される。８０４において、モバイルデバイスは、ＩｏＴハブ、外部サービス、及び／又はユーザにデータを伝送する。述べられたように、モバイルデバイスは、それが既に接続されている場合（例えば、ＷｉＦｉリンクを介して）、すぐにデータを送信し得る。

【００６６】

ＩｏＴデバイスからデータを収集することに加えて、一実施形態では、本明細書に記載される技術は、データを更新するか、又は別様にＩｏＴデバイスにデータを提供するために使用されてもよい。一例が図９Ａに示され、それは、ＩｏＴデバイス６０１（又はそのようなＩｏＴデバイスの群）上にインストールされる必要があるプログラムコード更新９０１を有する、ＩｏＴハブ１１０を示す。プログラムコード更新は、システム更新、パッチ、構成データ、及びＩｏＴデバイスがユーザの要求どおり動作するために必要とされる任意の他のデータを含んでもよい。一実施形態では、ユーザは、モバイルデバイス又はコンピュータを介してＩｏＴデバイス６０１に対する構成オプションを指定してもよく、そ

れらは次いで、本明細書に記載される技術を使用して、ＩｏＴハブ１１０上に記憶され、かつＩｏＴデバイスに提供される。具体的には、一実施形態では、ＩｏＴハブ１１０上の中間接続ロジック／アプリケーション７２１は、モバイルデバイス６１１上の中間接続ロジック／アプリケーション７１１と通信して、一時記憶装置６１５内にプログラムコード更新を記憶する。モバイルデバイス６１１がＩｏＴデバイス６０１の範囲に入るとき、モバイルデバイス６１１上の中間接続ロジック／アプリケーション７１１は、ＩｏＴデバイス６０１上の中間／接続ロジック／アプリケーション７０１と接続して、デバイスにプログラムコード更新を提供する。一実施形態では、ＩｏＴデバイス６０１は次いで、新しいプログラムコード更新及び／又はデータをインストールするための自動更新プロセスに入ってもよい。

【００６７】

ＩｏＴデバイスを更新するための方法が、図９Ｂに示される。本方法は、上記のシステムアーキテクチャとの関連で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【００６８】

９００において、新しいプログラムコード又はデータ更新は、ＩｏＴハブ及び／又は外部サービス（例えば、インターネット上でモバイルデバイスに連結された）上で利用可能になる。９０１において、モバイルデバイスは、ＩｏＴデバイスの代わりにプログラムコード又はデータ更新を受信及び記憶する。ＩｏＴデバイス及び／又はモバイルデバイスは、９０２において接続が確立されているかどうかを判定するために周期的にチェックする。９０３において接続が確立され、判定された場合、９０４において、更新は、ＩｏＴデバイスに伝送され、インストールされる。

改善されたセキュリティのための実施形態

【００６９】

一実施形態では、各ＩｏＴデバイス１０１の低電力マイクロコントローラ２００及びＩｏＴハブ１１０の低電力ロジック／マイクロコントローラ３０１は、以下に記載される実施形態によって使用される暗号鍵を記憶するための安全な鍵ストアを含む（例えば、図１０～図１５及び関連する文章を参照されたい）。代替的に、鍵は、後述するように、加入者識別モジュール（ＳＩＭ）内に確保されてもよい。

【００７０】

図１０は、ＩｏＴサービス１２０、ＩｏＴハブ１１０、並びにＩｏＴデバイス１０１及び１０２の間の通信を暗号化するために公開鍵インフラストラクチャ（public key infrastructure）（ＰＫＩ）技術及び／又は対称鍵交換／暗号化技術を使用する、高レベルアーキテクチャを例示する。

【００７１】

公開／秘密鍵ペアを使用する実施形態をまず説明し、続いて、対称鍵交換／暗号化技術を使用する実施形態を説明する。具体的には、ＰＫＩを使用するある実施形態において、一意的な公開／秘密鍵ペアが、各ＩｏＴデバイス１０１～１０２、各ＩｏＴハブ１１０、及びＩｏＴサービス１２０に関連付けられる。一実施形態では、新しいＩｏＴハブ１１０がセットアップされるとき、その公開鍵がＩｏＴサービス１２０に提供され、新しいＩｏＴデバイス１０１がセットアップされるとき、その公開鍵がＩｏＴハブ１１０及びＩｏＴサービス１２０の両方に提供される。デバイス間で公開鍵を安全に交換するための様々な技術を以下に説明する。一実施形態では、いかなる受信デバイスも、署名の妥当性を確認することによって公開鍵の妥当性を検証することができるように、すべての公開鍵が、受信デバイスのすべてに既知である親鍵（すなわち、一種の証明書）によって署名される。したがって、未加工の公開鍵を単に交換するのではなく、むしろこれらの証明書が交換されることになる。

【００７２】

例示したように、一実施形態では、各ＩｏＴデバイス１０１、１０２は、各デバイスの秘密鍵を記憶するセキュリティのために、それぞれ、安全な鍵ストア１００１、１００３

を含む。次いで、セキュリティロジック１００２、１３０４が、安全に記憶された秘密鍵を用いて、本明細書に記載される暗号化／解読動作を実行する。同様に、ＩｏＴハブ１１０は、ＩｏＴハブ秘密鍵、並びにＩｏＴデバイス１０１～１０２及びＩｏＴサービス１２０の公開鍵を記憶するための安全な記憶装置１０１１、並びに、鍵を使用して暗号化／解読動作を実行するためのセキュリティロジック１０１２を含む。最後に、ＩｏＴサービス１２０は、それ自体の秘密鍵、様々なＩｏＴデバイス及びＩｏＴハブの公開鍵を記憶するセキュリティのための安全な記憶装置１０２１、並びに鍵を使用してＩｏＴハブ及びデバイスとの通信を暗号化／解読するためのセキュリティロジック１０１３を含んでもよい。一実施形態では、ＩｏＴハブ１１０がＩｏＴデバイスから公開鍵証明書を受信すると、ＩｏＴハブ１１０は、それを（例えば、上記の親鍵を使用して署名の妥当性を確認することにより）検証し、次いで、その中から公開鍵を抽出し、その公開鍵をその安全な鍵ストア１０１１内に記憶することができる。

【００７３】

例として、一実施形態では、ＩｏＴサービス１２０が、コマンド又はデータ（例えば、ドアを開錠するコマンド、センサを読み取る要求、ＩｏＴデバイスにより処理／表示されるべきデータなど）をＩｏＴデバイス１０１に送信する必要があるとき、セキュリティロジック１０１３は、ＩｏＴデバイス１０１の公開鍵を使用してそのデータ／コマンドを暗号化して、暗号化されたＩｏＴデバイスパケットを生成する。一実施形態では、次いで、セキュリティロジック１０１３は、ＩｏＴハブ１１０の公開鍵を使用し、ＩｏＴデバイスパケットを暗号化して、ＩｏＴハブパケットを生成し、ＩｏＴハブパケットをＩｏＴハブ１１０に送信する。一実施形態では、デバイス１０１が、それが信頼されるソースから変更されていないメッセージを受信していることを検証することができるように、サービス１２０は、その秘密鍵又は上述の親鍵を用いて、暗号化されたメッセージに署名する。次いで、デバイス１０１は、秘密鍵及び／又は親鍵に対応する公開鍵を使用して、署名の妥当性を確認してもよい。上述したように、対称鍵交換／暗号化技術が、公開／秘密鍵暗号化の代わりに使用されてもよい。これらの実施形態では、１つの鍵をプライベートに記憶し、対応する公開鍵を他のデバイスに提供するのではなく、それぞれのデバイスに、暗号化のために、かつ署名の妥当性を確認するために使用されるものと同じ対称鍵のコピーを提供してもよい。対称鍵アルゴリズムの一例は高度暗号化標準（Advanced Encryption Standard）（ＡＥＳ）であるが、本発明の基本原理は、いかなるタイプの特定の対称鍵にも限定されない。

【００７４】

ある対称鍵実装形態を使用すると、各デバイス１０１は、ＩｏＴハブ１１０と対称鍵を交換するために、安全な鍵交換プロトコルに入る。動的対称鍵プロビジョニングプロトコル（Dynamic Symmetric Key Provisioning Protocol）（ＤＳＫＰＰ）などの安全な鍵プロビジョニングプロトコルが、安全な通信チャネルを介して鍵を交換するために使用され得る（例えば、コメント要求（Request for Comments）（ＲＦＣ）６０６３を参照されたい）。しかしながら、本発明の基本原理は、いかなる特定の鍵プロビジョニングプロトコルにも限定されるものではない。

【００７５】

対称鍵が交換されると、それらは、各デバイス１０１及びＩｏＴハブ１１０によって、通信を暗号化するために使用され得る。同様に、ＩｏＴハブ１１０及びＩｏＴサービス１２０は、安全な対称鍵交換を実行し、次いで、交換された対称鍵を使用して通信を暗号化し得る。一実施形態では、新しい対称鍵が、デバイス１０１とハブ１１０との間、及びハブ１１０とＩｏＴサービス１２０との間で定期的に交換される。一実施形態では、デバイス１０１、ハブ１１０、及びサービス１２０の間での新しい通信セッションのたびに、新しい対称鍵が交換される（例えば、通信セッションごとに新しい鍵が生成され、安全に交換される）。一実施形態では、ＩｏＴハブ内のセキュリティモジュール１０１２が信頼される場合、サービス１２０は、ハブセキュリティモジュール１３１２とセッション鍵を交渉し得、次いで、セキュリティモジュール１０１２が、各デバイス１２０とセッション鍵

を交渉することになる。次いで、サービス１２０からのメッセージは、ハブセキュリティモジュール１０１２で解読及び検証され、その後、デバイス１０１への送信のために再暗号化される。

【００７６】

一実施形態では、ハブセキュリティモジュール１０１２でのセキュリティ侵害を防止するために、１回限りの（恒久的な）インストール鍵が、インストール時にデバイス１０１とサービス１２０との間で交渉されてもよい。メッセージをデバイス１０１に送るとき、サービス１２０は、まずこのデバイスインストール鍵を用いて暗号化／ＭＡＣし、次いでハブのセッション鍵を用いてそれを暗号化／ＭＡＣし得る。次いで、ハブ１１０は、暗号化されたデバイスブロブを検証及び抽出し、それをデバイスに送ることになる。

【００７７】

本発明の一実施形態では、リプレイアタックを防止するためにカウンタ機構が実装される。例えば、デバイス１０１からハブ１１０へ（又は逆もまた同様）の連続する通信それぞれに、継続的に増加するカウンタ値が割り当てられ得る。ハブ１１０とデバイス１０１との両方がこの値を追跡し、デバイス間での連続する通信それぞれにおいてその値が正しいことを検証する。これと同じ技術が、ハブ１１０とサービス１２０との間に実装され得る。この方法でカウンタを使用すると、各デバイス間での通信を偽装することがより困難になるであろう（カウンタ値が誤ったものになるため）。しかしながら、これを用いずとも、サービスとデバイスとの間で共有されたインストール鍵は、すべてのデバイスに対するネットワーク（ハブ）規模の攻撃を防止するであろう。

【００７８】

一実施形態では、公開／秘密鍵暗号化を使用するとき、ＩｏＴハブ１１０は、その秘密鍵を使用してＩｏＴハブパケットを解読し、暗号化されたＩｏＴデバイスパケットを生成し、それを、関連付けられたＩｏＴデバイス１０１に送信する。次いで、ＩｏＴデバイス１０１は、その秘密鍵を使用してＩｏＴデバイスパケットを解読して、ＩｏＴサービス１２０を起点とするコマンド／データを生成する。次いで、ＩｏＴデバイス１０１は、データを処理し、かつ／又はコマンドを実行してもよい。対称暗号化を使用すると、各デバイスは、共有された対称鍵を用いて暗号化及び解読を行う。いずれかの場合であれば、各送信デバイスはまた、受信デバイスがメッセージの信頼性を検証することができるように、その秘密鍵を用いてメッセージに署名してもよい。

【００７９】

異なる鍵のセットが、ＩｏＴデバイス１０１からＩｏＴハブ１１０への通信及びＩｏＴサービス１２０への通信を暗号化するために使用されてもよい。例えば、ある公開／秘密鍵構成を使用すると、一実施形態では、ＩｏＴデバイス１０１上のセキュリティロジック１００２が、ＩｏＴハブ１１０の公開鍵を使用して、ＩｏＴハブ１１０に送信されたデータパケットを暗号化する。次いで、ＩｏＴハブ１１０上のセキュリティロジック１０１２は、ＩｏＴハブの秘密鍵を使用して、データパケットを解読し得る。同様に、ＩｏＴデバイス１０１上のセキュリティロジック１００２及び／又はＩｏＴハブ１１０上のセキュリティロジック１０１２は、ＩｏＴサービス１２０の公開鍵を使用して、ＩｏＴサービス１２０に送信されたデータパケットを暗号化し得る（これは次いで、ＩｏＴサービス１２０上のセキュリティロジック１０１３によって、サービスの秘密鍵を使用して解読され得る）。対称鍵を使用すると、デバイス１０１及びハブ１１０は、ある対称鍵を共有し得、一方でハブ及びサービス１２０は、異なる対称鍵を共有し得る。

【００８０】

上記の説明において、ある特定の具体的詳細が上に記載されているが、本発明の基本原理は様々な異なる暗号化技術を使用して実装され得ることに留意すべきである。例えば、上述した一部の実施形態は非対称の公開／秘密鍵ペアを使用するが、別の実施形態は、様々なＩｏＴデバイス１０１～１０２、ＩｏＴハブ１１０、及びＩｏＴサービス１２０の間で安全に交換される対称鍵を使用し得る。更に、一部の実施形態では、データ／コマンド自体は暗号化されないが、データ／コマンド（又は他のデータ構造）上の署名を生成する

ために鍵が使用される。次いで、受信者が、その鍵を使用して署名の妥当性を確認し得る。

【００８１】

図１１に例示するように、一実施形態では、各ＩｏＴデバイス１０１上の安全な鍵ストアは、プログラマブル加入者識別モジュール（ＳＩＭ）１１０１を使用して実装される。この実施形態では、ＩｏＴデバイス１０１は、ＩｏＴデバイス１０１上のＳＩＭインタフェース１１００内に据え付けられたプログラムされていないＳＩＭカード１１０１と共にエンドユーザに最初に提供され得る。１つ以上の暗号鍵のセットを用いてＳＩＭをプログラミングするために、ユーザは、プログラマブルＳＩＭカード１１０１をＳＩＭインタフェース５００から取り出し、それをＩｏＴハブ１１０上のＳＩＭプログラミングインタフェース１１０２に挿入する。次いで、ＩｏＴハブ上のプログラミングロジック１１２５が、ＩｏＴデバイス１０１をＩｏＴハブ１１０及びＩｏＴサービス１２０に登録／ペアリングするように、ＳＩＭカード１１０１を安全にプログラミングする。一実施形態では、公開／秘密鍵ペアは、プログラミングロジック１１２５によってランダムに生成されてもよく、次いで、このペアの公開鍵は、ＩｏＴハブの安全な記憶デバイス４１１内に記憶されてもよく、一方で秘密鍵は、プログラマブルＳＩＭ　１１０１内に記憶されてもよい。加えて、プログラミングロジック５２５は、ＩｏＴハブ１１０、ＩｏＴサービス１２０、及び／又は任意の他のＩｏＴデバイス１０１の公開鍵を、（ＩｏＴデバイス１０１上のセキュリティロジック１３０２による発信データの暗号化に使用するために）ＳＩＭカード１４０１上に記憶してもよい。いったんＳＩＭ　１１０１がプログラミングされると、新しいＩｏＴデバイス１０１に、ＳＩＭを安全な識別子として使用して（例えば、ＳＩＭを用いてデバイスを登録するための既存の技術を使用して）ＩｏＴサービス１２０がプロビジョニングされ得る。プロビジョニング後、ＩｏＴハブ１１０とＩｏＴサービス１２０との両方が、ＩｏＴデバイスの公開鍵のコピーを、ＩｏＴデバイス１０１との通信を暗号化する際に使用されるように安全に記憶する。

【００８２】

図１１に関して上述した技術は、新しいＩｏＴデバイスをエンドユーザに提供する際に多大な柔軟性を提供する。（現在行われているのと同様に）ユーザが販売／購入の際に各ＳＩＭを特定のサービスプロバイダに直接登録することを要するのではなく、ＳＩＭは、エンドユーザによりＩｏＴハブ１１０を介して直接プログラミングされてもよく、プログラミングの結果は、ＩｏＴサービス１２０に安全に通信され得る。それ故に、新しいＩｏＴデバイス１０１がオンライン又はローカルの小売業者からエンドユーザに販売され、後にＩｏＴサービス１２０が安全にプロビジョニングされ得る。

【００８３】

ＳＩＭ（加入者識別モジュール）という具体的な文脈において登録及び暗号化技術を上述したが、本発明の基本原理は「ＳＩＭ」デバイスに限定されない。むしろ、本発明の基本原理は、暗号鍵セットを記憶するための安全な記憶装置を有する、いかなるタイプのデバイスを使用して実装されてもよい。更に、上記の実施形態は取り外し可能なＳＩＭデバイスを含むのに対し、一実施形態では、ＳＩＭデバイスは取り外し可能でないが、ＩｏＴデバイス自体が、ＩｏＴハブ１１０のプログラミングインタフェース１１０２に挿入されてもよい。

【００８４】

一実施形態では、ユーザがＳＩＭ（又は他のデバイス）をプログラミングすることを要するのではなく、ＳＩＭは、エンドユーザへの流通前に、ＩｏＴデバイス１０１に予めプログラミングされる。この実施形態において、ユーザがＩｏＴデバイス１０１をセットアップするとき、本明細書に記載される様々な技術が、ＩｏＴハブ１１０／ＩｏＴサービス１２０と新しいＩｏＴデバイス１０１との間で暗号鍵を安全に交換するために使用され得る。

【００８５】

例えば、図１２Ａに例示するように、各ＩｏＴデバイス１０１又はＳＩＭ　４０１は、

ＩｏＴデバイス１０１及び／又はＳＩＭ　１００１を一意的に識別するバーコード又はＱＲコード(登録商標)１５０１と共に梱包されていてもよい。一実施形態では、バーコード又はＱＲコード(登録商標)１２０１は、ＩｏＴデバイス１０１又はＳＩＭ　１００１の公開鍵の符号化表現を含む。代替的に、バーコード又はＱＲコード(登録商標)１２０１は、ＩｏＴハブ１１０及び／又はＩｏＴサービス１２０によって、公開鍵を識別又は生成するために使用されてもよい（例えば、安全な記憶装置内に既に記憶されている公開鍵に対するポインタとして使用される）。バーコード又はＱＲコード(登録商標)６０１は、別個のカード上に（図１２Ａに示されるように）印刷されてもよく、又はＩｏＴデバイス自体上に直接印刷されてもよい。バーコードが印刷される場所に関わらず、一実施形態では、ＩｏＴハブ１１０には、バーコードを読み取り、得られたデータをＩｏＴハブ１１０上のセキュリティロジック１０１２及び／又はＩｏＴサービス１２０上のセキュリティロジック１０１３に提供するための、バーコードリーダ２０６が備わっている。次いで、ＩｏＴハブ１１０上のセキュリティロジック１０１２は、その安全な鍵ストア１０１１内にＩｏＴデバイスの公開鍵を記憶してもよく、ＩｏＴサービス１２０上のセキュリティロジック１０１３は、その安全な記憶装置１０２１内に公開鍵を（後の暗号化通信に使用するために）記憶してもよい。

【００８６】

一実施形態では、バーコード又はＱＲコード(登録商標)１２０１内に含まれるデータはまた、インストールされたＩｏＴアプリケーション又はＩｏＴサービスプロバイダにより設計されたブラウザベースのアプレットを用いて、ユーザデバイス１３５（例えば、ｉＰｈｏｎｅ(登録商標)又はＡｎｄｒｏｉｄデバイスなど）によりキャプチャされてもよい。キャプチャされると、バーコードデータは、安全な接続（例えば、セキュアソケットレイヤー（secure sockets layer）（ＳＳＬ）接続など）を介して、ＩｏＴサービス１２０に安全に通信され得る。バーコードデータはまた、安全なローカル接続を介して（例えば、ローカルＷｉＦｉ又はＢｌｕｅｔｏｏｔｈ(登録商標)　ＬＥ接続を介して）、クライアントデバイス１３５からＩｏＴハブ１１０に提供されてもよい。

【００８７】

ＩｏＴデバイス１０１上のセキュリティロジック１００２及びＩｏＴハブ１１０上のセキュリティロジック１０１２は、ハードウェア、ソフトウェア、ファームウェア、又はそれらの任意の組み合わせを使用して実装され得る。例えば、一実施形態では、セキュリティロジック１００２、１０１２は、ＩｏＴデバイス１０１とＩｏＴハブ１１０との間にローカル通信チャネル１３０を確立するために使用されるチップ（例えば、ローカルチャネル１３０がＢｌｕｅｔｏｏｔｈ(登録商標)　ＬＥである場合は、Ｂｌｕｅｔｏｏｔｈ(登録商標)　ＬＥチップ）内に実装される。セキュリティロジック１００２、１０１２の具体的な位置に関わらず、一実施形態では、セキュリティロジック１００２、１０１２は、ある特定のタイプのプログラムコードを実行するために安全な実行環境を確立するように設計される。これは、例えば、ＴｒｕｓｔＺｏｎｅ技術（一部のＡＲＭプロセッサで利用可能）及び／又はトラステッド・エグゼキューション・テクノロジー（Ｉｎｔｅｌにより設計）を使用することによって、実装され得る。当然のことながら、本発明の基本原理は、いかなる特定のタイプの安全な実行技術にも限定されない。

【００８８】

一実施形態では、バーコード又はＱＲコード(登録商標)１５０１は、各ＩｏＴデバイス１０１をＩｏＴハブ１１０とペアリングするために使用され得る。例えば、Ｂｌｕｅｔｏｏｔｈ(登録商標)　ＬＥデバイスをペアリングするために現在使用されている標準的な無線ペアリングプロセスを使用するのではなく、バーコード又はＱＲコード(登録商標)１５０１内に組み込まれたペアリングコードをＩｏＴハブ１１０に提供して、ＩｏＴハブを対応するＩｏＴデバイスとペアリングしてもよい。

【００８９】

図１２Ｂは、ＩｏＴハブ１１０上のバーコードリーダ２０６が、ＩｏＴデバイス１０１に関連付けられたバーコード／ＱＲコード(登録商標)１２０１をキャプチャする、一実施

形態を例示する。上述したように、バーコード／ＱＲコード(登録商標)１２０１は、ＩｏＴデバイス１０１上に直接印刷されてもよく、又はＩｏＴデバイス１０１と共に提供される別個のカード上に印刷されてもよい。いずれの場合においても、バーコードリーダ２０６は、バーコード／ＱＲコード(登録商標)１２０１からペアリングコードを読み取り、このペアリングコードをローカル通信モジュール１２８０に提供する。一実施形態では、ローカル通信モジュール１２８０は、Ｂｌｕｅｔｏｏｔｈ(登録商標)　ＬＥチップ及び関連付けられたソフトウェアであるが、本発明の基本原理は、いかなる特定のプロトコル標準にも限定されない。ペアリングコードが受信されると、それは、ペアリングデータ１２８５を含む安全な記憶装置内に記憶され、ＩｏＴデバイス１０１とＩｏＴハブ１１０とが自動的にペアリングされる。この方法でＩｏＴハブが新しいＩｏＴデバイスとペアリングされるたびに、そのペアリングに関するペアリングデータが、安全な記憶装置６８５内に記憶される。一実施形態では、ＩｏＴハブ１１０のローカル通信モジュール１２８０がペアリングコードを受信すると、それは、このコードを鍵として使用して、ローカル無線チャネルを介したＩｏＴデバイス１０１との通信を暗号化し得る。

【００９０】

同様に、ＩｏＴデバイス１０１側では、ローカル通信モジュール１５９０が、ＩｏＴハブとのペアリングを示すペアリングデータを、ローカルの安全な記憶デバイス１５９５内に記憶する。ペアリングデータ１２９５は、バーコード／ＱＲコード(登録商標)１２０１で識別される予めプログラミングされたペアリングコードを含んでもよい。ペアリングデータ１２９５はまた、安全なローカル通信チャネルを確立するために必要な、ＩｏＴハブ１１０上のローカル通信モジュール１２８０から受信されるペアリングデータ（例えば、ＩｏＴハブ１１０との通信を暗号化するための追加の鍵）を含んでもよい。

【００９１】

したがって、バーコード／ＱＲコード(登録商標)１２０１は、ペアリングコードが無線で送信されないため、現在の無線ペアリングプロトコルよりもはるかに安全な方法でローカルペアリングを実行するために使用され得る。加えて、一実施形態では、ペアリングに使用されるものと同じバーコード／ＱＲコード(登録商標)１２０１を使用して暗号鍵を識別し、ＩｏＴデバイス１０１からＩｏＴハブ１１０へ、かつＩｏＴハブ１１０からＩｏＴサービス１２０への安全な接続を構築することができる。

【００９２】

本発明の一実施形態によるＳＩＭカードをプログラミングするための方法が、図１３に例示される。本方法は、上述のシステムアーキテクチャ内で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【００９３】

１３０１において、ユーザは、空のＳＩＭカードを備えた新しいＩｏＴデバイスを受け取り、１６０２において、ユーザは、空のＳＩＭカードをＩｏＴハブに挿入する。１３０３において、ユーザは、１つ以上の暗号鍵のセットを用いて空のＳＩＭカードをプログラミングする。例えば、上述のように、一実施形態において、ＩｏＴハブは、公開／秘密鍵ペアをランダムに生成し、秘密鍵をＳＩＭカード上に、かつ公開鍵をそのローカルの安全な記憶装置内に記憶し得る。加えて、１３０４において、ＩｏＴデバイスを識別し、かつＩｏＴデバイスとの暗号化通信を確立するために使用され得るように、少なくとも公開鍵がＩｏＴサービスに送信される。上述したように、一実施形態では、「ＳＩＭ」カード以外のプログラマブルデバイスが、図１３に示される方法でＳＩＭカードと同じ機能を実行するために使用されてもよい。

【００９４】

新しいＩｏＴデバイスをネットワークに統合するための方法が、図１４に例示される。本方法は、上述のシステムアーキテクチャ内で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【００９５】

１４０１において、ユーザは、暗号鍵が予め割り当てられている新しいＩｏＴデバイス

を受け取る。１４０２において、この鍵がＩｏＴハブに安全に提供される。上述のように、一実施形態では、これは、ＩｏＴデバイスに関連付けられたバーコードを読み取って、デバイスに割り当てられた公開／秘密鍵ペアの公開鍵を識別することを伴う。バーコードは、ＩｏＴハブによって直接読み取られても、又はアプリケーション若しくはブラウザを介してモバイルデバイスによってキャプチャされてもよい。別の実施形態では、Ｂｌｕｅｔｏｏｔｈ(登録商標) ＬＥチャネル、近距離通信（ＮＦＣ）チャネル、又は安全なＷｉＦｉチャネルなどの安全な通信チャネルが、鍵の交換のためにＩｏＴデバイスとＩｏＴハブとの間に確立されてもよい。鍵の送信方法に関わらず、受信されると、鍵はＩｏＴハブデバイスの安全な鍵ストア内に記憶される。上述のように、セキュアエンクレーブ、トラステッド・エグゼキューション・テクノロジー（Trusted Execution Technology）（ＴＸＴ）、及び／又はＴｒｕｓｔｚｏｎｅなどの様々な安全な実行技術が、鍵の記憶及び保護のためにＩｏＴハブで使用され得る。加えて、８０３において、鍵はＩｏＴサービスに安全に送信され、ＩｏＴサービスは、この鍵をそれ自体の安全な鍵ストア内に記憶する。ＩｏＴサービスは次いで、この鍵を使用して、ＩｏＴデバイスとの通信を暗号化し得る。この場合も、この交換は、証明書／署名付き鍵を使用して実行されてもよい。ハブ１１０内では、記憶された鍵の改変／追加／除去を防止することが特に重要である。

【００９６】

公開／秘密鍵を使用してコマンド／データをＩｏＴデバイスに安全に通信するための方法が、図１５に例示される。本方法は、上述のシステムアーキテクチャ内で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【００９７】

１５０１において、ＩｏＴサービスは、ＩｏＴデバイス公開鍵を使用してデータ／コマンドを暗号化して、ＩｏＴデバイスパケットを作成する。次いで、ＩｏＴサービスは、ＩｏＴハブの公開鍵を使用し、このＩｏＴデバイスパケットを暗号化して、ＩｏＴハブパケットを作成する（例えば、ＩｏＴデバイスパケット周囲のＩｏＴハブラッパーを作成する）。１５０２において、ＩｏＴサービスは、ＩｏＴハブパケットをＩｏＴハブに送信する。１５０３において、ＩｏＴハブは、ＩｏＴハブの秘密鍵を使用してＩｏＴハブパケットを解読して、ＩｏＴデバイスパケットを生成する。次いで、１５０４において、ＩｏＴハブは、ＩｏＴデバイスパケットをＩｏＴデバイスに送信し、ＩｏＴデバイスは、１５０５において、ＩｏＴデバイス秘密鍵を使用してＩｏＴデバイスパケットを解読して、データ／コマンドを生成する。１５０６において、ＩｏＴデバイスは、データ／コマンドを処理する。

【００９８】

対称鍵を使用するある実施形態において、対称鍵交換は、各デバイス間（例えば、各デバイスとハブと、及びハブとサービスとの間）で交渉され得る。鍵交換が完了すると、各送信デバイスは、対称鍵を使用して各送信を暗号化し、かつ／又はそれに署名し、その後、データを受信デバイスに送信する。

モノのインターネット（ＩｏＴ）システムに安全な通信を確立するための装置及び方法

【００９９】

本発明の一実施形態では、通信チャネルをサポートするために使用される中間デバイス（例えば、ユーザのモバイルデバイス６１１及び／又はＩｏＴハブ１１０など）に関わらず、データの暗号化及び解読は、ＩｏＴサービス１２０と各ＩｏＴデバイス１０１との間で実行される。ＩｏＴハブ１１０を介して通信する一実施形態が図１６Ａに例示され、ＩｏＴハブを必要としない別の実施形態が図１６Ｂに例示される。

【０１００】

最初に図１６Ａで、ＩｏＴデバイス１０１とＩｏＴサービス１２０との間の通信を暗号化／解読するために、ＩｏＴサービス１２０は、「サービスセッション鍵」１６５０のセットを管理する暗号化エンジン１６６０を含み、各ＩｏＴデバイス１０１は、「デバイスセッション鍵」１６５１のセットを管理する暗号化エンジン１６６１を含む。暗号化エンジンは、本明細書に記載するセキュリティ／暗号化技術を実行するとき、セッション公開

／秘密鍵ペアを生成して、このペアのセッション秘密鍵へのアクセスを防止するための（他のものの中でも）ハードウェアセキュリティモジュール１６３０～１６３１、及び導出したシークレットを使用してキーストリームを生成するためのキーストリーム生成モジュール１６４０～１６４１を含む、異なるハードウェアモジュールに依拠することができる。一実施形態では、サービスセッション鍵１６５０及びデバイスセッション鍵１６５１は、関連する公開／秘密鍵ペアを含む。例えば、一実施形態では、ＩｏＴデバイス１０１上のデバイスセッション鍵１６５１は、ＩｏＴサービス１２０の公開鍵、及びＩｏＴデバイス１０１の秘密鍵を含む。以下に詳細に説明するように、一実施形態では、安全な通信セッションを確立するために、セッション公開／秘密鍵ペア１６５０及び１６５１がそれぞれの暗号化エンジン、それぞれ１６６０及び１６６１によって使用されて、同じシークレットを生成し、このシークレットは次にＳＫＧＭ　１６４０～１６４１によって使用されて、ＩｏＴサービス１２０とＩｏＴデバイス１０１との間の通信を暗号化及び解読するキーストリームを生成する。本発明の一実施形態によるシークレットの生成及び使用に関連付けられた追加の詳細は、以下に提供される。

【０１０１】

図１６Ａで、鍵１６５０～１６５１を使用してシークレットが生成されると、クライアントは、クリアトランザクション１６１１によって示されるように常にＩｏＴサービス１２０を介してＩｏＴデバイス１０１にメッセージを送信することになる。本明細書で使用されるとき「クリア」は、根本的なメッセージが本明細書に記載された暗号化技術を使用して暗号化されていないことを示すことを意味する。しかし、例示したように、一実施形態では、セキュアソケットレイヤー（ＳＳＬ）チャネル又は他の安全なチャネル（例えば、インターネットプロトコルセキュリティ（Internet Protocol Security）（ＩＰＳＥＣ）チャネル）は、通信を保護するためにクライアントデバイス６１１とＩｏＴサービス１２０との間で確立される。ＩｏＴサービス１２０上の暗号化エンジン１６６０は、次に、生成されたシークレットを使用してメッセージを暗号化して、１６０２で暗号化メッセージをＩｏＴハブ１１０に送信する。メッセージを直接暗号化するためにシークレットを使用するのではなく、一実施形態では、シークレット及びカウンタ値を使用して、キーストリームを生成し、このキーストリームを使用して、それぞれのメッセージパケットを暗号化する。この実施形態の詳細は、図１７に関して以下に説明する。

【０１０２】

例示したように、ＳＳＬ接続又は他の安全なチャネルは、ＩｏＴサービス１２０とＩｏＴハブ１１０との間で確立することができる。ＩｏＴハブ１１０（一実施形態ではメッセージを解読する能力を有さない）は、１６０３で（例えば、Ｂｌｕｅｔｏｏｔｈ(登録商標)　Ｌｏｗ　Ｅｎｅｒｇｙ（ＢＴＬＥ）通信チャネルを介して）暗号化メッセージをＩｏＴデバイスに送信する。ＩｏＴデバイス１０１上の暗号化エンジン１６６１は、次に、シークレットを使用してメッセージを解読して、メッセージコンテンツを処理することができる。キーストリームを生成するためにシークレットを使用する実施形態では、暗号化エンジン１６６１は、シークレット及びカウンタ値を使用してキーストリームを生成し、次にメッセージパケットの解読のためにキーストリームを使用することができる。

【０１０３】

メッセージ自体は、ＩｏＴサービス１２０とＩｏＴデバイス１０１との間の任意の形態の通信を含むことができる。例えば、メッセージは、測定を行ってその結果をクライアントデバイス６１１に通知して返すことなどの特定の機能を実行することをＩｏＴデバイス１０１に命令するコマンドパケットを含むことができる、又はＩｏＴデバイス１０１の動作を構成する構成データを含むことができる。

【０１０４】

応答が必要とされる場合、ＩｏＴデバイス１０１上の暗号化エンジン１６６１は、シークレット又は導出されたキーストリームを使用して、応答を暗号化し、１６０４で暗号化応答をＩｏＴハブ１１０に送信し、ＩｏＴハブ１１０は、１６０５で応答をＩｏＴサービス１２０に転送する。ＩｏＴサービス１２０上の暗号化エンジン１６６０は、次に、シー

クレット又は導出されたキーストリームを使用して応答を解読して、１６０６で（例えば、ＳＳＬ又は他の安全な通信チャネルを介して）解読された応答をクライアントデバイス６１１に送信する。

【０１０５】

図１６Ｂは、ＩｏＴハブを必要としない実施形態を例示する。むしろ、この実施形態では、ＩｏＴデバイス１０１とＩｏＴサービス１２０との間の通信は、クライアントデバイス６１１を介して行われる（例えば、図６～９Ｂに関して上述した実施形態におけるように）。この実施形態では、メッセージをＩｏＴデバイス１０１に送信するために、クライアントデバイス６１１は、１６１１でメッセージの非暗号化バージョンをＩｏＴサービス１２０に送信する。暗号化エンジン１６６０は、シークレット又は導出されたキーストリームを使用してメッセージを暗号化して、１６１２で暗号化メッセージをクライアントデバイス６１１に返送する。クライアントデバイス６１１は、次に、１６１３で暗号化メッセージをＩｏＴデバイス１０１に転送し、暗号化エンジン１６６１は、シークレット又は導出されたキーストリームを使用してメッセージを解読する。ＩｏＴデバイス１０１は、次に、本明細書に記載されたようにメッセージを処理することができる。応答が必要とされる場合、暗号化エンジン１６６１は、シークレットを使用して、応答を暗号化し、１６１４で暗号化応答をクライアントデバイス６１１に送信し、クライアントデバイス６１１は、１６１５で暗号化応答をＩｏＴサービス１２０に転送する。暗号化エンジン１６６０は、次に、応答を解読して、１６１６で解読された応答をクライアントデバイス６１１に送信する。

【０１０６】

図１７は、ＩｏＴサービス１２０とＩｏＴデバイス１０１との間で最初に実行することができる鍵交換及びキーストリーム生成を例示する。一実施形態では、この鍵交換は、ＩｏＴサービス１２０及びＩｏＴデバイス１０１が新しい通信セッションを確立するたびに実行することができる。代替的に、鍵交換を実行することができ、交換されたセッション鍵を指定された期間（例えば、一日、一週間など）使用することができる。簡潔にするために図１７に中間デバイスは示されていないが、通信は、ＩｏＴハブ１１０及び／又はクライアントデバイス６１１を介して行うことができる。

【０１０７】

一実施形態では、ＩｏＴサービス１２０の暗号化エンジン１６６０は、セッション公開／秘密鍵ペアを生成するために、コマンドをＨＳＭ　１６３０（例えば、Ａｍａｚｏｎ（登録商標）によって提供されるＣｌｏｕｄＨＳＭなどとすることができる）に送信する。ＨＳＭ　１６３０は、その後、このペアのセッション秘密鍵へのアクセスを防止することができる。同様に、ＩｏＴデバイス１０１上の暗号化エンジンは、セッション公開／秘密鍵ペアを生成してこのペアのセッション秘密鍵へのアクセスを防止するＨＳＭ　１６３１（例えば、Ａｔｍｅｌ　Ｃｏｒｐｏｒａｔｉｏｎ（登録商標）によるＡｔｅｃｃ５０８　ＨＳＭなどの）にコマンドを送信することができる。当然のことながら、本発明の基本原理は、いかなる特定のタイプの暗号化エンジン又は製造業者にも限定されない。

【０１０８】

一実施形態では、ＩｏＴサービス１２０は、１７０１で、ＨＳＭ　１６３０を使用して生成されたそのセッション公開鍵をＩｏＴデバイス１０１に送信する。ＩｏＴデバイスは、そのＨＳＭ　１６３１を使用して、それ自体のセッション公開／秘密鍵ペアを生成し、１７０２でそのペアの公開鍵をＩｏＴサービス１２０に送信する。一実施形態では、暗号化エンジン１６６０～１６６１は、楕円曲線Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎ（Elliptic curve Diffie-Hellman）（ＥＣＤＨ）プロトコルを使用し、このプロトコルは、楕円曲線公開－秘密鍵ペアを有する２つの当事者が共有シークレットを確立することができる匿名鍵の取り決めである。一実施形態では、これらの技術を使用して、１７０３で、ＩｏＴサービス１２０の暗号化エンジン１６６０は、ＩｏＴデバイスセッション公開鍵及びそれ自体のセッション秘密鍵を使用してシークレットを生成する。同様に、１７０４で、ＩｏＴデバイス１０１の暗号化エンジン１６６１は、ＩｏＴサービス１２０のセッション公開鍵

及びそれ自体のセッション秘密鍵を使用して同じシークレットを独自に生成する。より具体的には、一実施形態では、ＩｏＴサービス１２０上の暗号化エンジン１６６０は、シークレット＝ＩｏＴデバイスセッション公開鍵*ＩｏＴサービスセッション秘密鍵という式に従って、シークレットを生成し、ここで（*）は、ＩｏＴデバイスセッション公開鍵がＩｏＴサービスセッション秘密鍵によって点乗積されることを意味する。ＩｏＴデバイス１０１上の暗号化エンジン１６６１は、シークレット＝ＩｏＴサービスセッション公開鍵*ＩｏＴデバイスセッション秘密鍵という式に従って、シークレットを生成し、ＩｏＴサービスセッション公開鍵は、ＩｏＴデバイスセッション秘密鍵によって点乗積される。結局、ＩｏＴサービス１２０及びＩｏＴデバイス１０１は両方とも、以下に説明するように通信を暗号化するのに使用される同じシークレットを生成した。一実施形態では、暗号化エンジン１６６０～１６６１は、シークレットを生成するための上記の動作を実行するＫＳＧＭ、それぞれ１６４０～１６４１などのハードウェアモジュールに依拠する。

【０１０９】

シークレットが決定されると、シークレットは、暗号化エンジン１６６０及び１６６１によって使用されて、データを直接暗号化及び解読することができる。代替的に、一実施形態では、暗号化エンジン１６６０～１６６１は、コマンドをＫＳＧＭ　１６４０～１６４１に送信して、それぞれのデータパケットを暗号化／解読するためにシークレットを使用して新しいキーストリームを生成する（すなわち、それぞれのパケットに対して新しいキーストリームデータ構造が生成される）。具体的には、キーストリーム生成モジュール１６４０～１６４１の一実施形態は、それぞれのデータパケットに対してカウンタ値が増加され、キーストリームを生成するためにシークレットと組み合わせて使用される、Ｇａｌｏｉｓ／カウンタモード（Galois/Counter Mode）（ＧＣＭ）を実装する。したがって、データパケットをＩｏＴサービス１２０に送信するために、ＩｏＴデバイス１０１の暗号化エンジン１６６１は、シークレット及び現在のカウンタ値を使用して、ＫＳＧＭ　１６４０～１６４１に新しいキーストリームを生成させ、次のキーストリームを生成するためにカウンタ値を増加させる。次に、新たに生成されたキーストリームを使用して、データパケットを暗号化し、その後、ＩｏＴサービス１２０に送信される。一実施形態では、キーストリームは、データでＸＯＲされて、暗号化データパケットを生成する。一実施形態では、ＩｏＴデバイス１０１は、カウンタ値を暗号化データパケットと共にＩｏＴサービス１２０に送信する。ＩｏＴサービス上の暗号化エンジン１６６０は、次に、ＫＳＧＭ　１６４０と通信し、ＫＳＧＭ　１６４０は、受信したカウンタ値及びシークレットを使用して、キーストリーム（同じシークレット及びカウンタ値が使用されるので同じキーストリームでなければならない）を生成し、生成されたキーストリームを使用して、データパケットを解読する。

【０１１０】

一実施形態では、ＩｏＴサービス１２０からＩｏＴデバイス１０１に送信されるデータパケットは、同じ方法で暗号化される。具体的には、それぞれのデータパケットに対してカウンタが増加されて、シークレットと共に使用されて、新しいキーストリームを生成する。キーストリームは、次に、データを暗号化するために使用され（例えば、データ及びキーストリームのＸＯＲを実行して）、暗号化データパケットは、カウンタ値と共にＩｏＴデバイス１０１に送信される。ＩｏＴデバイス１０１上の暗号化エンジン１６６１は、次に、ＫＳＧＭ　１６４１と通信し、ＫＳＧＭ　１６４１は、カウンタ値及びシークレットを使用して、データパケットを解読するために使用される同じキーストリームを生成する。したがって、この実施形態では、暗号化エンジン１６６０～１６６１は、それら自体のカウンタ値を使用して、データを暗号化するキーストリームを生成し、暗号化データパケットと共に受信したカウンタ値を使用して、データを解読するキーストリームを生成する。

【０１１１】

一実施形態では、それぞれの暗号化エンジン１６６０～１６６１は、それが他方から受信した最後のカウンタ値を追跡し、カウンタ値がシーケンス外で受信されたか否か又は同

じカウンタ値が１回より多く受信されたか否かを検出するシーケンシングロジックを含む。カウンタ値がシーケンス外で受信された場合、又は同じカウンタ値が１回より多く受信された場合、これは、リプレイアタックが試みられていることを示し得る。それに応答して、暗号化エンジン１６６０～１６６１は、通信チャネルから接続を切ることができる、及び／又はセキュリティアラートを生成することができる。

【０１１２】

図１８は、４バイトのカウンタ値１８００と、可変サイズの暗号化データフィールド１８０１と、６バイトのタグ１８０２とを含む、本発明の一実施形態で用いられる例示的な暗号化データパケットを例示する。一実施形態では、タグ１８０２は、解読されたデータ（それが解読されたら）の妥当性を確認するチェックサム値を含む。

【０１１３】

上述したように、一実施形態では、ＩｏＴサービス１２０とＩｏＴデバイス１０１との間で交換されたセッション公開／秘密鍵ペア１６５０～１６５１は、定期的に、及び／又はそれぞれの新しい通信セッションの開始に応答して生成することができる。

【０１１４】

本発明の一実施形態は、ＩｏＴサービス１２０とＩｏＴデバイス１０１との間のセッションを認証するための追加の技術を実装する。具体的には、一実施形態では、親鍵ペア、工場鍵ペアのセット、並びにＩｏＴサービス鍵ペアのセット及びＩｏＴデバイス鍵ペアのセットを含む、公開／秘密鍵ペアの階層が使用される。一実施形態では、親鍵ペアは、他の鍵ペアのすべてに対する信頼のルートを含み、単一の高度に安全な場所に（例えば、本明細書に記載されたＩｏＴシステムを実装する組織の管理下に）維持される。マスター秘密鍵を使用して、工場鍵ペアなどの様々な他の鍵ペアの上に署名を生成する（及びそれによって認証する）ことができる。署名は、次に、マスター公開鍵を使用して検証することができる。一実施形態では、ＩｏＴデバイスを製造するそれぞれの工場は、それ自体の工場鍵ペアを割り当てられ、工場鍵ペアは、次に、ＩｏＴサービス鍵及びＩｏＴデバイス鍵を認証するために使用することができる。例えば、一実施形態では、工場秘密鍵を使用して、ＩｏＴサービス公開鍵及びＩｏＴデバイス公開鍵の上に署名を生成する。これらの署名は、次に、対応する工場公開鍵を使用して検証することができる。これらのＩｏＴサービス／デバイス公開鍵は、図１６Ａ～Ｂに関して上述した「セッション」公開／秘密鍵と同じではないことに留意されたい。上述したセッション公開／秘密鍵は、一時的であり（すなわち、サービス／デバイスセッションに対して生成される）、一方、ＩｏＴサービス／デバイス鍵ペアは、恒久的なものである（すなわち、工場で生成される）。

【０１１５】

親鍵、工場鍵、サービス／デバイス鍵の間の上述の関係を念頭に、本発明の一実施形態は、ＩｏＴサービス１２０とＩｏＴデバイス１０１との間の認証及びセキュリティの追加のレイヤを提供するために、以下の動作を実行する。

Ａ．一実施形態では、ＩｏＴサービス１２０は、最初に、以下を含むメッセージを生成する。

１．ＩｏＴサービスの一意的なＩＤ：
- ・ＩｏＴサービスのシリアルナンバー、
- ・タイムスタンプ、
- ・この一意的なＩＤに署名するために使用される工場鍵のＩＤ、
- ・一意的なＩＤ（すなわち、サービス）のクラス、
- ・ＩｏＴサービスの公開鍵、
- ・一意的なＩＤの上の署名。

２．以下を含む工場証明書：
- ・タイムスタンプ、
- ・証明書に署名するために使用される親鍵のＩＤ、
- ・工場公開鍵、
- ・工場証明書の署名。

３．ＩｏＴサービスセッション公開鍵（図１６Ａ～Ｂに関して上述したような）
４．ＩｏＴサービスセッション公開鍵署名（例えば、ＩｏＴサービスの秘密鍵で署名された）。
Ｂ．一実施形態では、メッセージは、交渉チャネル（以下に説明する）上でＩｏＴデバイスに送信される。ＩｏＴデバイスは、メッセージを解析して：
１．工場証明書の署名（メッセージペイロード内に存在する場合のみ）を検証する。
２．一意的なＩＤによって識別された鍵を使用して一意的なＩＤの署名を検証する。
３．一意的なＩＤからのＩｏＴサービスの公開鍵を使用してＩｏＴサービスセッション公開鍵署名を検証する。
４．ＩｏＴサービスの公開鍵、並びにＩｏＴサービスのセッション公開鍵を保存する。
５．ＩｏＴデバイスセッション鍵ペアを生成する。
Ｃ．ＩｏＴデバイスは、次に、以下を含むメッセージを生成する：
１．ＩｏＴデバイスの一意的なＩＤ、
・ＩｏＴデバイスのシリアルナンバー、
・タイムスタンプ、
・この一意的なＩＤに署名するために使用される工場鍵のＩＤ、
・一意的なＩＤ（すなわち、ＩｏＴデバイス）のクラス、
・ＩｏＴデバイスの公開鍵、
・一意的なＩＤの署名。
２．ＩｏＴデバイスのセッション公開鍵。
３．ＩｏＴデバイスの鍵で署名された（ＩｏＴデバイスセッション公開鍵＋ＩｏＴサービスセッション公開鍵）の署名。
Ｄ．このメッセージは、ＩｏＴサービスに返送される。ＩｏＴサービスは、メッセージを解析して：
１．工場公開鍵を使用して一意的なＩＤの署名を検証する。
２．ＩｏＴデバイスの公開鍵を使用してセッション公開鍵の署名を検証する。
３．ＩｏＴデバイスのセッション公開鍵を保存する。
Ｅ．ＩｏＴサービスは、次に、ＩｏＴサービスの鍵で署名された（ＩｏＴデバイスセッション公開鍵＋ＩｏＴサービスセッション公開鍵）の署名を含むメッセージを生成する。
Ｆ．ＩｏＴデバイスは、メッセージを解析して：
１．ＩｏＴサービスの公開鍵を使用してセッション公開鍵の署名を検証する。
２．ＩｏＴデバイスセッション秘密鍵及びＩｏＴサービスのセッション公開鍵からキーストリームを生成する。
３．ＩｏＴデバイスは、次に、「メッセージング利用可能」メッセージを送信する。
Ｇ．ＩｏＴサービスは、次に、以下を実行する：
１．ＩｏＴサービスセッション秘密鍵及びＩｏＴデバイスのセッション公開鍵からキーストリームを生成する。
２．以下を含めて、メッセージングチャネル上で新しいメッセージを作成する：
・ランダムな２バイト値を生成して記憶する。
・ブーメラン属性Ｉｄ（以下に説明する）及びランダム値を有する属性メッセージを設定する。
Ｈ．ＩｏＴデバイスは、メッセージを受信して：
１．メッセージを解読することを試みる。
２．示された属性Ｉｄ上と同じ値を有する更新を送信する。
Ｉ．ＩｏＴサービスは、メッセージペイロードがブーメラン属性更新を含むことを認識する：
１．そのペアリング状態を真に設定する。
２．交渉チャネル上でペアリング完了メッセージを送信する。
Ｊ．ＩｏＴデバイスは、メッセージを受信して、ＩｏＴデバイスのペアリング状態を真

に設定する。

【０１１６】

上述の技術は「ＩｏＴサービス」及び「ＩｏＴデバイス」に関して説明したが、本発明の基本原理は、ユーザのクライアントデバイス、サーバ、及びインターネットサービスを含む、任意の２つのデバイス間で安全な通信チャネルを確立するように実装することができる。

【０１１７】

上述の技術は、秘密鍵が無線で共有されない（シークレットが片方の当事者から他方に送信される現在のＢｌｕｅｔｏｏｔｈ(登録商標)ペアリング技術と対照的に）ので、高度に安全である。会話全体を聞いている攻撃者は、公開鍵を有するのみということになり、これは、共有シークレットを生成するために不十分である。これらの技術はまた、署名された公開鍵を交換することによる中間者攻撃を防止する。加えて、ＧＣＭ及び別個のカウンタがそれぞれのデバイス上で使用されるため、任意の種類の「リプレイアタック」（中間者がデータをキャプチャしてそれを再度送信する）が防止される。いくつかの実施形態はまた、非対称カウンタを使用することによりリプレイアタックを防止する。

デバイスを正式にペアリングすることなくデータ及びコマンドを交換するための技術

【０１１８】

ＧＡＴＴは、一般属性プロファイル（Generic Attribute Profile）に対する頭字語であり、これは、２つのＢｌｕｅｔｏｏｔｈ(登録商標) Ｌｏｗ Ｅｎｅｒｇｙ（ＢＴＬＥ）デバイスがデータを往復して伝送する方法を規定する。これは、属性プロトコル（Attribute Protocol）（ＡＴＴ）と呼ばれる一般データプロトコルを利用し、このプロトコルは、簡単なルックアップテーブルに、テーブルへの入力ごとに１６ビットの特性ＩＤを使用してサービス、特性、及び関連データを記憶するために使用される。一方で「特性」は、「属性」と呼ばれることもあることに留意されたい。

【０１１９】

Ｂｌｕｅｔｏｏｔｈ(登録商標)デバイス上で、最も一般的に使用される特性は、デバイスの「名前」（特性ＩＤ １０７５２（０ｘ２Ａ００）を有する）である。例えば、Ｂｌｕｅｔｏｏｔｈ(登録商標)デバイスは、その近傍内の他のＢｌｕｅｔｏｏｔｈ(登録商標)デバイスを、ＧＡＴＴを使用してこれらの他のＢｌｕｅｔｏｏｔｈ(登録商標)デバイスによって発行された「名前」特性を読み取ることにより、識別することができる。したがって、ブルートゥース(登録商標)デバイスは、デバイスを正式にペアリング／結合することなくデータを交換するための固有の能力を有する（「ペアリング」と「結合」とが時として交換可能に使用される点に留意されたい。この議論の残りは、用語「ペアリング」を使用することになる）。

【０１２０】

本発明の一実施形態は、ＢＴＬＥ対応ＩｏＴデバイスと、これらのデバイスと正式にペアリングすることなく通信するために、この能力を利用する。それぞれの個別のＩｏＴデバイスとのペアリングは、ペアリングするために必要とされる時間のため、及び同時に１つのペアリングされた接続のみを確立することができるため、著しく非効率であろう。

【０１２１】

図１９は、Ｂｌｕｅｔｏｏｔｈ(登録商標)（ＢＴ）デバイス１９１０が、ペアリングされたＢＴ接続を正式に確立することなくＩｏＴデバイス１０１のＢＴ通信モジュール１９０１とのネットワークソケットアブストラクションを確立する、特定の一実施形態を例示する。ＢＴデバイス１９１０は、図１６Ａに示すようなＩｏＴハブ１１０及び／又はクライアントデバイス６１１内に含めることができる。例示したように、ＢＴ通信モジュール１９０１は、特性ＩＤ、それらの特性ＩＤに関連付けられた名前、及びそれらの特性ＩＤに対する値のリストを含むデータ構造を維持する。それぞれの特性に対する値は、現在のＢＴ標準に従って特性ＩＤにより識別された２０バイトのバッファに記憶することができる。しかしながら、本発明の基本原理は、いかなる特定のバッファサイズにも限定されない。

【０１２２】
図１９の実施例では、「名前」特性は、「ＩｏＴデバイス１４」の特定の値を割り当てられたＢＴで規定された特性である。本発明の一実施形態は、ＢＴデバイス１９１０との安全な通信チャネルを交渉するために使用される追加の特性の第１のセット、及びＢＴデバイス１９１０との暗号化通信のために使用される追加の特性の第２のセットを指定する。具体的には、例示した実施例で特性ＩＤ＜６５５３２＞により識別された「交渉書込」特性は、発信交渉メッセージを送信するために使用することができ、特性ＩＤ＜６５５３３＞により識別された「交渉読取」特性は、受信交渉メッセージを受信するために使用することができる。「交渉メッセージ」は、本明細書に記載されたような安全な通信チャネルを確立するためにＢＴデバイス１９１０及びＢＴ通信モジュール１９０１によって使用されるメッセージを含むことができる。例として、図１７で、ＩｏＴデバイス１０１は、「交渉読取」特性＜６５５３３＞を介してＩｏＴサービスセッション公開鍵１７０１を受信することができる。鍵１７０１は、ＩｏＴサービス１２０からＢＴＬＥ対応ＩｏＴハブ１１０又はクライアントデバイス６１１に送信することができ、それらは、次に、ＧＡＴＴを使用して、特性ＩＤ＜６５５３３＞により識別された交渉読取値バッファに鍵１７０１を書込むことができる。ＩｏＴデバイスのアプリケーションロジック１９０２は、次に、特性ＩＤ＜６５５３３＞により識別された値バッファから鍵１７０１を読み取って、上述したようにそれを処理することができる（例えば、それを使用してシークレットを生成し、シークレットを使用してキーストリームを生成するなど）。

【０１２３】
鍵１７０１が２０バイト（一部の現在の実装形態での最大バッファサイズ）より大きい場合は、鍵は２０バイトの部分に書き込むことができる。例えば、最初の２０バイトは、ＢＴ通信モジュール１９０３によって特性ＩＤ＜６５５３３＞に書き込んで、ＩｏＴデバイスアプリケーションロジック１９０２によって読み取ることができ、ＩｏＴデバイスアプリケーションロジック１９０２は、次に、確認応答メッセージを特性ＩＤ＜６５５３２＞により識別された交渉書込値バッファに書込むことができる。ＧＡＴＴを使用して、ＢＴ通信モジュール１９０３は、この確認応答を特性ＩＤ＜６５５３２＞から読み取ることができ、それに応じて、鍵１７０１の次の２０バイトを特性ＩＤ＜６５５３３＞により識別された交渉読取値バッファに書込むことができる。この方法で、特性ＩＤ＜６５５３２＞及び＜６５５３３＞により規定されたネットワークソケットアブストラクションは、安全な通信チャネルを確立するために使用される交渉メッセージを交換するために確立される。

【０１２４】
一実施形態では、安全な通信チャネルが確立されると、特性ＩＤ＜６５５３４＞（ＩｏＴデバイス１０１から暗号化データパケットを送信するための）及び特性ＩＤ＜６５５３３＞（ＩｏＴデバイスにより暗号化データパケットを受信するための）を使用して、第２のネットワークソケットアブストラクションが確立される。すなわち、ＢＴ通信モジュール１９０３が送信する暗号化データパケット（例えば、図１６Ａの暗号化メッセージ１６０３などの）を有するとき、ＢＴ通信モジュール１９０３は、特性ＩＤ＜６５５３３＞により識別されたメッセージ読取値バッファを使用して一度に２０バイト、暗号化データパケットを書込み始める。ＩｏＴデバイスアプリケーションロジック１９０２は、次に、読取値バッファから一度に２０バイト、暗号化データパケットを読み取り、必要に応じて特性ＩＤ＜６５５３２＞により識別された書込値バッファを介して確認応答メッセージをＢＴ通信モジュール１９０３に送信することになる。

【０１２５】
一実施形態では、後述するＧＥＴ、ＳＥＴ、及びＵＰＤＡＴＥのコマンドを使用して、２つのＢＴ通信モジュール１９０１と１９０３との間でデータ及びコマンドを交換する。例えば、ＢＴ通信モジュール１９０３は、特性ＩＤ＜６５５３３＞を識別しＳＥＴコマンドを含むパケットを送信して、特性ＩＤ＜６５５３３＞により識別された値フィールド／バッファに書込むことができ、それは次に、ＩｏＴデバイスアプリケーションロジック１

９０２によって読み取ることができる。ＩｏＴデバイス１０１からデータを取得するために、ＢＴ通信モジュール１９０３は、特性ＩＤ＜６５５３４＞により識別された値フィールド／バッファに向けられたＧＥＴコマンドを送信することができる。ＧＥＴコマンドに応答して、ＢＴ通信モジュール１９０１は、特性ＩＤ＜６５５３４＞により識別された値フィールド／バッファからのデータを含むＵＰＤＡＴＥパケットをＢＴ通信モジュール１９０３に送信することができる。加えて、ＵＰＤＡＴＥパケットは、ＩｏＴデバイス１０１上の特定の属性の変化に応答して、自動的に送信することができる。例えば、ＩｏＴデバイスが照明システムに関連付けられていて、ユーザが照明をオンにする場合、ＵＰＤＡＴＥパケットを送信して、照明アプリケーションに関連付けられたオン／オフ属性にこの変化を反映することができる。

【０１２６】

図２０は、本発明の一実施形態による、ＧＥＴ、ＳＥＴ、及びＵＰＤＡＴＥ用に使用される例示的なパケット形式を例示する。一実施形態では、これらのパケットは、交渉の後に、メッセージ書込＜６５５３４＞及びメッセージ読取＜６５５３３＞チャネルを介して送信される。ＧＥＴパケット２００１では、最初の１バイトのフィールドは、パケットをＧＥＴパケットとして識別する値（０Ｘ１０）を含む。２番目の１バイトのフィールドは、現在のＧＥＴコマンドを一意的に識別する（すなわち、ＧＥＴコマンドが関連付けられた現在のトランザクションを識別する）要求ＩＤを含む。例えば、サービス又はデバイスから送信されたＧＥＴコマンドのそれぞれのインスタンスに、異なる要求ＩＤを割り当てることができる。これは、例えば、カウンタを増加させて、カウンタ値を要求ＩＤとして使用することにより、実行することができる。しかしながら、本発明の基本原理は、要求ＩＤを設定するためのいかなる特定の方法にも限定されるものではない。

【０１２７】

２バイトの属性ＩＤは、パケットが向けられたアプリケーション特有の属性を識別する。例えば、ＧＥＴコマンドが図１９に例示したＩｏＴデバイス１０１に送信されている場合、属性ＩＤを使用して、要求されている特定のアプリケーション特有の値を識別することができる。上述の実施例に戻って、ＧＥＴコマンドは、照明システムの電源状態などのアプリケーション特有の属性ＩＤに向けることができ、この属性ＩＤは、照明が電源がオン又はオフになっているかを識別する値（例えば、１＝オン、０＝オフ）を含む。ＩｏＴデバイス１０１がドアに関連付けられたセキュリティ装置である場合、値フィールドは、ドアの現在の状態（例えば、１＝開いている、０＝閉じている）を識別することができる。ＧＥＴコマンドに応答して、属性ＩＤにより識別された現在の値を含む応答を送信することができる。

【０１２８】

図２０に例示したＳＥＴパケット２００２及びＵＰＤＡＴＥパケット２００３もまた、パケットのタイプ（すなわち、ＳＥＴ及びＵＰＤＡＴＥ）を識別する最初の１バイトのフィールド、要求ＩＤを含む２番目の１バイトのフィールド、及びアプリケーションで定義された属性を識別する２バイトの属性ＩＤフィールドを含む。加えて、ＳＥＴパケットは、ｎバイトの値データフィールドに含まれたデータの長さを識別する２バイト長の値を含む。値データフィールドは、ＩｏＴデバイス上で実行されるコマンド、及び／又はなんらかの方法でＩｏＴデバイスの動作を構成する（例えば、所望のパラメータを設定する、ＩｏＴデバイスの電源を切るなど）構成データを含むことができる。例えば、ＩｏＴデバイス１０１がファンの速度を制御する場合、値フィールドは、現在のファンの速度を反映することができる。

【０１２９】

ＵＰＤＡＴＥパケット２００３は、ＳＥＴコマンドの結果の更新を提供するために送信することができる。ＵＰＤＡＴＥパケット２００３は、ＳＥＴコマンドの結果に関連したデータを含むことができるｎバイトの値データフィールドの長さを識別する、２バイト長の値フィールドを含む。加えて、１バイトの更新状態フィールドは、更新されている変数の現在の状態を識別することができる。例えば、ＳＥＴコマンドがＩｏＴデバイスにより

制御された照明をオフにすることを試みた場合、更新状態フィールドは、照明が正常にオフにされたか否かを示すことができる。

【０１３０】

図２１は、ＳＥＴ及びＵＰＤＡＴＥコマンドを伴うＩｏＴサービス１２０とＩｏＴデバイス１０１との間の例示的なトランザクションのシーケンスを例示する。ＩｏＴハブ及びユーザのモバイルデバイスなどの中間デバイスは、本発明の基本原理を不明瞭にすることを避けるために示されていない。２１０１で、ＳＥＴコマンド２１０１は、ＩｏＴサービスからＩｏＴデバイス１０１に送信されて、ＢＴ通信モジュール１９０１により受信され、ＢＴ通信モジュール１９０１は、それに応じて、２１０２で特性ＩＤにより識別されたＧＡＴＴ値バッファを更新する。ＳＥＴコマンドは、２１０３で低電力マイクロコントローラ（low power microcontroller）（ＭＣＵ）２００により（又は図１９に示すＩｏＴデバイスアプリケーションロジック１９０２などの低電力ＭＣＵ上で実行されているプログラムコードにより）値バッファから読み取られる。２１０４で、ＭＣＵ　２００又はプログラムコードは、ＳＥＴコマンドに応答して動作を実行する。例えば、ＳＥＴコマンドは、新しい温度などの新しい構成パラメータを指定する属性ＩＤを含むことができる、又はオン／オフなどの状態値（ＩｏＴデバイスを「オン」又は低電力状態に入らせるための）を含むことができる。したがって、２１０４で、新しい値がＩｏＴデバイスに設定され、２１０５でＵＰＤＡＴＥコマンドが返され、２１０６でＧＡＴＴ値フィールドの実際の値が更新される。場合により、実際の値は、所望の値に等しいであろう。他の場合では、更新された値は、異なることがある（すなわち、ＩｏＴデバイス１０１がある特定のタイプの値を更新するのに時間がかかることがあるため）。最終的に、２１０７で、ＧＡＴＴ値フィールドからの実際の値を含むＵＰＤＡＴＥコマンドがＩｏＴサービス１２０に返送される。

【０１３１】

図２２は、本発明の一実施形態によるＩｏＴサービスとＩｏＴデバイスとの間で安全な通信チャネルを実装するための方法を例示する。本方法は、上述のネットワークアーキテクチャとの関連で実装され得るが、いかなる特定のアーキテクチャにも限定されない。

【０１３２】

２２０１で、ＩｏＴサービスは、楕円曲線デジタル署名アルゴリズム（elliptic curve digital signature algorithm）（ＥＣＤＳＡ）証明書を使用してＩｏＴハブと通信するための暗号化チャネルを作成する。２２０２で、ＩｏＴサービスは、セッションシークレットを使用してＩｏＴデバイスパケット内のデータ／コマンドを暗号化して、暗号化デバイスパケットを作成する。上述したように、セッションシークレットは、ＩｏＴデバイス及びＩｏＴサービスによって独自に生成することができる。２２０３で、ＩｏＴサービスは、暗号化チャネルを介して暗号化デバイスパケットをＩｏＴハブに送信する。２２０４で、解読することなく、ＩｏＴハブは、暗号化デバイスパケットをＩｏＴデバイスに渡す。２２－５で、ＩｏＴデバイスは、セッションシークレットを使用して、暗号化デバイスパケットを解読する。上述したように、一実施形態では、これは、シークレット及びカウンタ値（暗号化デバイスパケットと共に提供される）を使用してキーストリームを生成し、次にキーストリームを使用してパケットを解読することにより実現することができる。２２０６で、ＩｏＴデバイスは、次に、デバイスパケットに含まれたデータ及び／又はコマンドを抽出して処理する。

【０１３３】

したがって、上述の技術を使用して、標準的なペアリング技術を使用して、ＢＴデバイスを正式にペアリングすることなく、２つのＢＴ対応デバイス間で双方向の安全なネットワークソケットアブストラクションを確立することができる。これらの技術は、ＩｏＴサービス１２０と通信するＩｏＴデバイス１０１に関して上述したが、本発明の基本原理は、任意の２つのＢＴ対応デバイス間で安全な通信チャネルを交渉して確立するように実装することができる。

【０１３４】

図２３Ａ～Ｃは、本発明の一実施形態によるデバイスをペアリングするための詳細な方法を例示する。本方法は、上述のシステムアーキテクチャとの関連で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【０１３５】

２３０１で、ＩｏＴサービスは、ＩｏＴサービスのシリアルナンバー及び公開鍵を含むパケットを作成する。２３０２で、ＩｏＴサービスは、工場秘密鍵を使用してパケットに署名する。２３０３で、ＩｏＴサービスは、暗号化チャネルを介してＩｏＴハブにパケットを送信し、２３０４で、ＩｏＴハブは、非暗号化チャネルを介してＩｏＴデバイスにパケットを転送する。２３０５で、ＩｏＴデバイスは、パケットの署名を検証し、２３０６で、ＩｏＴデバイスは、ＩｏＴデバイスのシリアルナンバー及び公開鍵を含むパケットを生成する。２３０７で、ＩｏＴデバイスは、工場秘密鍵を使用してパケットに署名し、２３０８で、ＩｏＴデバイスは、非暗号化チャネルを介してＩｏＴハブにパケットを送信する。

【０１３６】

２３０９で、ＩｏＴハブは、暗号化チャネルを介してパケットをＩｏＴサービスに転送し、２３１０で、ＩｏＴサービスは、パケットの署名を検証する。２３１１で、ＩｏＴサービスは、セッション鍵ペアを生成し、２３１２で、ＩｏＴサービスは、セッション公開鍵を含むパケットを生成する。ＩｏＴサービスは、次に、２３１３で、ＩｏＴサービス秘密鍵でパケットに署名し、２３１４で、ＩｏＴサービスは、暗号化チャネルを介してパケットをＩｏＴハブに送信する。

【０１３７】

図２３Ｂに移って、２３１５で、ＩｏＴハブは、非暗号化チャネルを介してパケットをＩｏＴデバイスに転送し、２３１６で、ＩｏＴデバイスは、パケットの署名を検証する。２３１７で、ＩｏＴデバイスは、セッション鍵ペアを生成し（例えば、上述の技術を使用して）、２３１８で、ＩｏＴデバイスセッション公開鍵を含むＩｏＴデバイスパケットが生成される。２３１９で、ＩｏＴデバイスは、ＩｏＴデバイス秘密鍵でＩｏＴデバイスパケットに署名する。２３２０で、ＩｏＴデバイスは、非暗号化チャネルを介してＩｏＴハブにパケットを送信し、２３２１で、ＩｏＴハブは、暗号化チャネルを介してＩｏＴサービスにパケットを転送する。

【０１３８】

２３２２で、ＩｏＴサービスは、パケットの署名を検証し（例えば、ＩｏＴデバイス公開鍵を使用して）、２３２３で、ＩｏＴサービスは、ＩｏＴサービス秘密鍵及びＩｏＴデバイス公開鍵を使用して、セッションシークレットを生成する（先に詳細に説明したように）。２３２４で、ＩｏＴデバイスは、ＩｏＴデバイス秘密鍵及びＩｏＴサービス公開鍵を使用して、セッションシークレットを生成し（また、上述したように）、２３２５で、ＩｏＴデバイスは、乱数を生成して、セッションシークレットを使用してその乱数を暗号化する。２３２６で、ＩｏＴサービスは、暗号化チャネルを介して暗号化パケットをＩｏＴハブに送信する。２３２７で、ＩｏＴハブは、非暗号化チャネルを介して暗号化パケットをＩｏＴデバイスに転送する。２３２８で、ＩｏＴデバイスは、セッションシークレットを使用してパケットを解読する。

【０１３９】

図２３Ｃに移って、２３２９で、ＩｏＴデバイスは、セッションシークレットを使用してパケットを再暗号化し、２３３０で、ＩｏＴデバイスは、非暗号化チャネルを介して暗号化パケットをＩｏＴハブに送信する。２３３１で、ＩｏＴハブは、暗号化チャネルを介して暗号化パケットをＩｏＴサービスに転送する。２３３２で、ＩｏＴサービスは、セッションシークレットを使用してパケットを解読する。２３３３で、ＩｏＴサービスは、乱数がＩｏＴサービスが送信した乱数と一致することを検証する。ＩｏＴサービスは、次に、２３３４で、ペアリングが完了したことを示すパケットを送信し、２３３５で、その後のメッセージはすべて、セッションシークレットを使用して暗号化される。

パケット間隔タイミングを修正してデータ転送状態を識別するための装置及び方法

【０１４０】
Ｂｌｕｅｔｏｏｔｈ(登録商標) Ｌｏｗ Ｅｎｅｒｇｙ（ＢＴＬＥ）デバイスは、「アドバタイジング間隔」で分割されているアドバタイジングパケットを送信して、デバイス間の接続を確立する。ＢＴＬＥ周辺デバイスは、アドバタイジング間隔を使用してアドバタイジングパケットを、周囲のすべてのデバイスに送信する。次いで、受信ＢＴＬＥデバイスは、この情報に基づいて行動する、又は更に情報を受信するために接続することができる。
【０１４１】
ＢＴＬＥの２．４ＧＨｚスペクトルは、２４０２ＭＨｚ～２４８０ＭＨｚに拡張し、０～３９の番号が付けられた４０　１ＭＨｚ幅チャネルを使用する。各チャネルは、２ＭＨｚで分割されている。チャネル３７、３８、及び３９は、アドバタイズメントパケットを送信するためにのみ使用される。残りは、接続中のデータ交換のために使用される。ＢＴＬＥアドバタイズメントの間、ＢＴＬＥ周辺デバイスは、３つのアドバタイジングチャネル上に順々にパケットを送信する。デバイス又はビーコンをスキャンするセントラルデバイスは、アドバタイジングパケットを待ってそれらのチャネルをリッスンし、近隣のデバイスを発見するのに役立つ。チャネル３７、３８、及び３９は、２．４ＧＨｚスペクトルにわたり意図的に広がっている（即ち、チャネル３７及び３９は、帯域における第１及び最後のチャネルであり、チャネル３８は、中間である）。任意の単一のアドバタイジングチャネルがブロックされた場合、他のチャネルは、数ＭＨｚの帯域幅で分割されているのでフリーになりやすい。
【０１４２】
ＩｏＴデバイスが、送信するべきデータを有するとき、そのＩｏＴデバイスは、データを送信する準備ができていることを示すために、通常はそのアドバタイズメントパケットの一部としてフラグを含む。本発明の一実施形態では、このフラグを使用するのではなくて、ＩｏＴデバイスは、保留データを有することを示すためにアドバタイジング間隔を調節する。例えば、Ｔがアドバタイズメントパケット間の時間である場合に、保留のデータがないとき、０．７５Ｔ、０．５Ｔ、又は１．２５Ｔなどの異なるアドバタイジング間隔が、データが保留であることを示すように選択され得る。一実施形態では、２つの異なる間隔は、アプリケーションの特定の要件に基づいてプログラム可能であり、どの間隔がどの状態を示すかを決定することを困難にする。
【０１４３】
図２４は、ＩｏＴデバイス１０１の一実施形態を示し、この実施形態では、ＢＴＬＥ通信インタフェース２４１０が、データが送信される準備ができているときにアドバタイジング間隔を調節するアドバタイジング間隔選択ロジック２４１１を含む。それに加えて、ＩｏＴハブ１１０のＢＴＬＥ通信インタフェース２４２０は、アドバタイジング間隔検出ロジック２４２１を含むことにより、アドバタイジング間隔の変化を検出し、確認応答を与え、データを受信する。
【０１４４】
特に、示している実施形態では、ＩｏＴデバイス１０１のアプリケーション２４０１は、送信されるべきデータが有することを示す。それに応じて、アドバタイジング間隔選択ロジック２４１１は、アドバタイジング間隔を修正することにより、データが送信されるべきこと（例えば、間隔を．７５Ｔ、又はなんらかの別の値に変えること等）をＩｏＴハブ１１０に通知する。アドバタイジング間隔検出ロジック２４２１が変化を検出すると、ＢＴＬＥ通信インタフェース２４２０は、ＩｏＴデバイス１０１のＢＴＬＥ通信インタフェース２４１０に接続して、それがデータを受信する準備ができていることを示す。ＩｏＴデバイス１０１のＢＴＬＥ通信インタフェース２４１０は、次いで、ＩｏＴハブのＢＴＬＥ通信インタフェース２４２０にデータを送信する。ＩｏＴハブは、次いで、自らを通してＩｏＴサービス１２０に、及び／又はユーザのクライアントデバイス（図示せず）にデータを渡してもよい。データが送信された後に、アドバタイジング間隔選択ロジック２４１１は、次いで、通常のアドバタイジング間隔（例えばＡＩ＝Ｔ）に戻ってもよい。

【０１４５】
本発明の一実施形態では、安全な通信チャネルが、上記のセキュリティ／暗号化技術のうちの１つ又は複数を使用して、ＩｏＴデバイス１０１とＩｏＴサービス１２０との間に確立される（例えば、図１６Ａ～２３Ｃ及び関連する本文を参照）。例えば、一実施形態では、ＩｏＴサービス１２０は、上記のようにＩｏＴデバイス１０１との鍵交換を実行して、ＩｏＴデバイス１０１とＩｏＴサービス１２０との間のすべての通信を暗号化する。
【０１４６】
本発明の一実施形態に従う方法が図２５に示されている。本方法は、上記のシステムアーキテクチャとの関連で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。
【０１４７】
２５００において、（例えば、時間Ｔによって分離された）アドバタイジングパケットを生成するときに、ＩｏＴデバイスは、標準のアドバタイジング間隔を使用する。ＩｏＴデバイスは、２５０２において、それが送るべきデータを有することが２５０１において決定されるまで、標準のアドバタイジング間隔を維持する。次いで、２５０３において、ＩｏＴデバイスは、アドバタイジング間隔を切り換えることにより、送信するべきデータを有することを示す。２５０４において、ＩｏＴハブ又は別のネットワークデバイスは、ＩｏＴデバイスとの接続を確立することにより、ＩｏＴデバイスがそれ自体のデータを送信するのを可能にする。最終的に、２５０５において、ＩｏＴデバイスは、それ自体の保留データをＩｏＴハブに送信する。
【０１４８】
アドバタイジング間隔技術がＢＴＬＥプロトコルと関連して本明細書に記載されているけれども、本発明の基礎原理は、ＢＴＬＥに限定されないことを留意すべきである。実際に、本発明の基礎原理は、デバイス同士の間に無線通信を確立するためのアドバタイジング間隔を選択する任意のシステムに実装されてもよい。
【０１４９】
それに加えて、専用のＩｏＴハブ１１０が上記の多くの実施形態に示されているけれども、専用のＩｏＴハブハードウェアプラットホームが本発明の基礎原理に従うのに必要ではない。例えば、上記の様々なＩｏＴハブは、ｉＰｈｏｎｅｓ（登録商標）及びＡｎｄｒｏｉｄ（登録商標）デバイス等の様々な別のネットワーキングデバイス内で実行されるソフトウェアとして実装されてもよい。実際に、上述したＩｏＴハブは、（例えば、ＢＴＬＥ又は別のローカル無線プロトコルを使用して）ＩｏＴデバイスと通信することができる、及び（例えば、ＷｉＦｉ又はセルラーのデータ接続を使用してＩｏＴサービスに）インターネットを介して接続を確立することができる任意のデバイスに実装されてもよい。
ＩｏＴハブをＩｏＴデバイスに接続する際に無線トラフィックを低減するためのシステム及び方法
【０１５０】
複数のＩｏＴハブが特定の場所に構成されると、単一のＩｏＴデバイスは、範囲内の各ＩｏＴハブと接続する能力を有し得る。上述したように、ＩｏＴデバイスは、ＩｏＴハブがＩｏＴデバイスに接続してコマンド及び／又はデータを送信することができるように、アドバタイジングチャネルを使用して、「接続可能」である範囲内の任意のＩｏＴハブに通知し得る。複数のＩｏＴハブが、ＩｏＴデバイスの範囲内にあるとき、ＩｏＴサービスは、これらのＩｏＴハブのそれぞれを介してＩｏＴデバイス宛てのコマンド／データを送信しようとすることによって、無線帯域幅を浪費し及び性能を低減する恐れがある（例えば、複数の伝送に起因した干渉のために）。
【０１５１】
この問題に対処するために、本発明の一実施形態は、ひとたび特定のＩｏＴハブがＩｏＴデバイスに正常に接続されると、他のＩｏＴハブはコマンド／データを送信する試みを中断するように通知されることを確実にする技術を実装する。この実施形態は、そのすべてがＩｏＴデバイス１０１の範囲内にあるＩｏＴハブ１１０～１１２の例示的なセットを

示す図２６Ａ～図２６Ｃに関して記述される。結果として、ＩｏＴデバイス１０１の安全な無線通信モジュール２６１０は、ＩｏＴハブ１１０～１１２のそれぞれの安全な無線通信モジュール２６５０～２６５２を見出し、それに接続することができる。一実施形態では、安全な無線通信モジュールは、上述した安全なＢＴＬＥモジュールを含む。しかしながら、本発明の基本原理は、いかなる特定の無線標準にも限定されない。

【０１５２】

図２６Ａに例示されるように、一実施形態では、ＩｏＴデバイス１０１の安全な無線通信モジュール２６１０は、定期的に近隣の無線通信デバイスに「接続可能」である（即ち、範囲内の任意のデバイスによって接続され得る）ことを示すアドバタイジングビーコンを送信するアドバタイジング制御ロジック２６１０を含む。次いで、アドバタイジングビーコンを受信する任意のＩｏＴハブ１１０～１１２は、ＩｏＴデバイス１０１を認識し、ＩｏＴサービスによってコマンド／データがＩｏＴデバイス１０１に宛てられると、安全な無線通信モジュール２６５０～２６５２は、ＩｏＴデバイス１０１の安全な無線通信モジュール２６１０に接続し得る。

【０１５３】

図２６Ｂに例示されるように、一実施形態では、ＩｏＴサービスがＩｏＴデバイス１０１に対するデータ／コマンドを有するとき、特定の場所内のすべてのＩｏＴハブ１１０～１１２にデータ／コマンドを送信し得る（例えば、ユーザのアカウントに関連した、かつ／又はＩｏＴデバイス１０１の範囲内のすべてのＩｏＴハブ）。例示したように、次いで、ＩｏＴハブ１１０～１１２のそれぞれは、コマンド／データを提供するためにＩｏＴデバイス１０１と接続しようとし得る。

【０１５４】

図２６Ｃに例示されるように、一実施形態では、単一のＩｏＴハブ１１１だけがＩｏＴデバイス１０１に正常に接続し、ＩｏＴデバイス１０１によって処理するためのコマンド／データを提供する。ＢＴＬＥなどの特定の無線通信プロトコルを使用して、ひとたび接続されると、安全な無線通信モジュール２６１０は、アドバタイジングビーコンの送信を中断する。したがって、他のＩｏＴハブ１１０、１１２は、ＩｏＴデバイス１０１がＩｏＴハブ１１１からデータを正常に受信したことを知る方法がなく、コマンド／データを送信しようとし続け、それによって無線帯域幅を消費し、干渉を引き起こす。

【０１５５】

この制限に対処するために、安全な無線通信モジュール２６１０の一実施形態は、ＩｏＴハブ１１１の安全な無線通信モジュール２６５１との正常な接続を検出すると、アドバタイジング制御モジュール２６１２にアドバタイジングビーコンの送信を継続させる接続マネージャ２６１１を含む。しかしながら、ＩｏＴデバイス１０１が「接続可能」であることを示す代わりに、新しいアドバタイジングビーコンは、ＩｏＴデバイス１０１が「接続不可能」であることを示す。一実施形態では、「接続不可能」の指示に応じて、ＩｏＴハブ１１０、１１２の安全な無線通信モジュール２６５０、２６５２は、ＩｏＴデバイスにコマンド／データを送信しようとする試みを中断し、それによって不要な無線トラフィックを低減させる。

【０１５６】

上述の技術は、既存の無線プロトコルの上に容易に実装され得る技術を使用して望ましくない無線トラフィックに洗練された解決策を提供する。例えば、一実施形態では、「接続可能」及び「接続不可能」の指示は、ＢＴＬＥ標準の関連の中で実装される。しかしながら、上述したように、本発明の基本原理は、多種多様の異なる無線ネットワークプロトコルを使用して実装され得る。

【０１５７】

本発明の一実施形態に従う方法が図２７に示されている。本方法は、上記のシステムアーキテクチャとの関連で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【０１５８】

２７０１において、コマンド及び／又はデータは、ＩｏＴサービスから２つ以上のＩｏＴハブを通って送信される。例えば、ユーザは、ＩｏＴサービスに接続されているユーザのモバイルデバイス上のアプリを介してＩｏＴデバイスを制御しようとしていてもよい。２７０２において、ＩｏＴハブは、ＩｏＴデバイスに接続しようと試み、そのＩｏＴハブのうちの１つが正常に接続し、コマンド／データをＩｏＴデバイスに提供する。上述したように、ＩｏＴハブは、ＩｏＴデバイスがアドバタイジングビーコンで「接続可能」の指示を送信した結果としてＩｏＴデバイスを認識し得る。

【０１５９】

２７０３において、正常な接続に応じて、そのＩｏＴデバイスは、「接続不可能」アドバタイジングビーコンを送信し始め、それによって範囲内のどのＩｏＴハブもＩｏＴデバイスがそれ以上接続可能でないことを通知する。２７０４において、「接続不可能」ビーコンを受信すると、他のＩｏＴハブは、コマンド／データをＩｏＴデバイスへ送信しようとする試みを中断する。

安全なモノのインターネット（ＩｏＴ）デバイスプロビジョニングのためのシステム及び方法

【０１６０】

上述したように、一実施形態では、デバイスは、ＩｏＴハブにアドバタイズするとき、ＩｏＴデバイスを個別に特定するためにハブ及びＩｏＴサービスが使用する８バイトの「デバイスＩＤ」を使用する。デバイスＩＤは、システムにＩｏＴデバイスをプロビジョニング／登録するために読み取られ、ＩｏＴサービスに送信されるＩｏＴデバイス上に印刷された固有のバーコード又はＱＲコード(登録商標)内に含まれ得る。ひとたびプロビジョニング／登録されると、デバイスＩＤは、システム内でＩｏＴデバイスのアドレスを指定するために使用される。

【０１６１】

この実装に対する１つのセキュリティ上の懸念は、バーコード／ＱＲコード(登録商標)データが暗号化せずに送信され得るので、デバイスＩＤの無線伝送を傍受してシステムに侵入することが可能であり得、それによって、別のユーザが、彼／彼女のアカウントとデバイスＩＤを関連付けることができるようになることである。

【０１６２】

一実施形態では、この懸念に対処するために、「関連付けＩＤ」は、それぞれのデバイスＩＤと関連付けられ、かつプロビジョニングプロセス中に使用されてデバイスＩＤが決して平文で送信されないことを確実にする。図２８に例示されるように、この実施形態では、関連付けＩＤ　２８１２は、ＩｏＴデバイス１０１上に印刷されたバーコード／ＱＲコード(登録商標)に含まれるが、デバイスＩＤ　２８１１は、上述した技術を実装してＩｏＴサービス１２０との安全な通信を確実にする安全な無線通信モジュール２８１０内に安全に維持されている。一実施形態では、関連付けＩＤ　２８１２は、デバイスＩＤと同じように８バイトのＩＤであり、ＩｏＴデバイスごとに固有である。システムにおいて、新しいＩｏＴデバイス１０１がプロビジョニングされると、ユーザは、インストールされたＩｏＴアプリ又はアプリケーションを有するユーザデバイス１３５を用いて、関連付けＩＤ　２８１２を含むバーコード／ＱＲコード(登録商標)をスキャンする。代替的に、又は追加的に、ＩｏＴハブ１１０を使用して、関連付けＩＤを含むバーコード／ＱＲコード(登録商標)をキャプチャしてもよい。

【０１６３】

いずれの場合においても、関連付けＩＤは、ＩｏＴサービス１２０上の、各関連付けＩＤと各デバイスＩＤとの間の関連付けを含むデバイスデータベース２８５１でルックアップを実行するデバイスプロビジョニングモジュール２８５０に送信される。デバイスプロビジョニングモジュール２８５０は、関連付けＩＤ　２８１２を使用してデバイスＩＤ２８１１を識別し、次いでデバイスＩＤを使用して、システム内の新しいＩｏＴデバイス１０１をプロビジョニングする。特に、ひとたびデバイスＩＤがデバイスデータベース２８５１から決定されると、デバイスプロビジョニングモジュール２８５０は、ＩｏＴハブ

１１０（ユーザデバイス１３５を含んでもよい）にコマンドを送信し、ＩｏＴハブ１１０がデバイスＩＤ　２８１１を用いてＩｏＴデバイス１０１と通信することを許可する。

【０１６４】

一実施形態では、関連付けＩＤ　２８１２は、ＩｏＴデバイス１０１が製造される際に（即ち、安全な無線通信モジュール２８１０がプロビジョニングされる際に）工場で生成される。次いで、デバイスＩＤ　２８１１及び関連付けＩＤ　２８１２の両方は、ＩｏＴサービスに提供され、デバイスデータベース２８５１内に記憶され得る。例示したように、デバイスデータベース２８５１は、それぞれのデバイスがプロビジョニングされたかどうかを特定する指示を含んでもよい。例として、これは、ＩｏＴデバイス１０１がプロビジョニングされていることを示す第１の値（例えば、１）及びＩｏＴデバイスがプロビジョニングされていないことを示す第２の値（例えば、０）による２進値であってもよい。ひとたびシステムが、ＩｏＴデバイス１０１をプロビジョニング／登録すると、ＩｏＴサービス１２０とＩｏＴデバイス１０１との間の通信は上述したセキュリティ技術を使用して保護されるため、デバイスＩＤを使用することができる。

【０１６５】

一実施形態では、ユーザがＩｏＴデバイスを売却する際に、ユーザは、ＩｏＴサービス１２０にログインし、ユーザのアカウントからＩｏＴデバイスをリリースすることによって、デバイスＩＤをリリースすることができる。次いで、新しいユーザは、本明細書に記載したデバイスプロビジョニング技術を使用して、ＩｏＴデバイスをプロビジョニングし、ＩｏＴデバイスを彼の／彼女のアカウントと関連付けることができる。

【０１６６】

本発明の一実施形態に従う方法が図２９に示されている。本方法は、上記のシステムアーキテクチャとの関連で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【０１６７】

２９０１において、関連付けは、（例えば、ＩｏＴデバイスが製造される工場で）ＩｏＴデバイスのデバイスＩＤと関連付けＩＤとの間で生成される。関連付けＩＤは、ＩｏＴデバイスに印されたバーコード／ＱＲコード(登録商標)内に組み込まれ得る。２９０２において、デバイスＩＤと関連付けＩＤとの間の関連付けは、ＩｏＴサービスに記憶される。２９０３において、ユーザは、新しいＩｏＴデバイスを購入し、関連付けＩＤを含むバーコード／ＱＲコード(登録商標)を（例えば、アプリ若しくはアプリケーションがそこにインストールされているユーザのモバイルデバイスを介して、又はバーコードリーダを備えるＩｏＴハブを介して）スキャンする。

【０１６８】

２９０４において、関連付けＩＤは、ＩｏＴサービスに送信され、そして２９０５において、この関連付けＩＤを使用してデバイスＩＤを識別する。２９０６において、ＩｏＴデバイスは、デバイスＩＤを使用してプロビジョニングされる。例えば、ＩｏＴデバイスデータベースは、この特定のデバイスＩＤがプロビジョニングされたことを示すために更新されてもよく、ＩｏＴサービスは、デバイスＩＤをＩｏＴハブに伝達し、ＩｏＴハブに新しいＩｏＴデバイスと通信するように指示することができる。

モノのインターネット（ＩｏＴ）システムにおいてフロー制御を実行するためのシステム及び方法

【０１６９】

ローカル無線ネットワークトラフィックは、所定の場所内のＩｏＴデバイスの数に基づいて増加することになる。更に、場合によっては、ＩｏＴデバイスは、ＩｏＴデバイスによって実行されている妥当な所定の機能よりも多くのデータを送信していてよい。例えば、ＩｏＴデバイス上のソフトウェア／ハードウェアは誤動作する恐れがあり、又はＩｏＴデバイスは侵入される恐れがあり、ＩｏＴデバイスがＩｏＴサービスに不必要なデータを継続的に送信する原因になる。

【０１７０】

本発明の一実施形態は、ＩｏＴハブで、特定のＩｏＴデバイスが指定したデータ閾値に到達したときにデータトラフィックを効果的に無視するフロー制御を実行することによって、これらの問題に対処する。一実施形態では、それぞれのＩｏＴデバイスは、ある期間にわたってＩｏＴデバイスが送信することを認められているデータ量を示すフロー制御パラメータの指定したセットで構成される。フロー制御パラメータは、ＩｏＴデバイスのタイプに基づいていてもよい。例えば、ドアの錠及びサーモスタットなどの特定のＩｏＴデバイスは、典型的にはデータのショートパケットのみを定期的に送信するべきであるのに対して、ビデオカメラなどの他のＩｏＴデバイスは、潜在的に非周期の方式ではるかに大きなデータ量を送信してもよい。したがって、フロー制御パラメータは、問題になっているＩｏＴデバイスの期待される動作に基づいて十分な量の帯域幅を提供するように設定されてもよい。一実施形態では、それぞれのＩｏＴデバイスは、そのＩｏＴデバイスのデータ要件に基づいて特定のフロー制御「クラス」に割り当てられている。

【０１７１】

そのような実施形態が図３０に例示されるとき、図は、それぞれ異なるフロー制御パラメータのセット３０１５、３０３１、３０４１で構成された安全な無線通信モジュール２８１０、３０３０、３０４０を有する複数のＩｏＴデバイス１０１～１０３を示す。一実施形態では、フロー制御パラメータは、周波数及び／又はそれぞれのＩｏＴデバイスが指定した期間にわたり送信することを期待されるデータ量を指定する（例えば、２５Ｍバイト／時間、５０Ｍバイト／時間、１００Ｍバイト／日、１０通信試行／日など）。一実施形態では、フロー制御パラメータ３０１５、３０３１、３０４１は、例示したように、ＩｏＴデバイスデータベース２８５１内のデバイスごとのフロー制御パラメータのセット３０２０を管理するデバイス管理モジュール３０２１を含むＩｏＴサービス１２０によって指定され得る。例えば、ひとたび各ＩｏＴデバイスに対するデータ伝送要件が決定されると、フローごとの制御パラメータ３０２０は、これらの要件を反映するように更新され得る。

【０１７２】

上述したように、一実施形態では、デバイスデータベース２８５１は、複数の異なるフロー制御「クラス」に関するデータ伝送要件を含む（例えば、視聴覚デバイス、温度デバイス、制御デバイス、セキュリティデバイスなど）。新しいＩｏＴデバイスがシステムに導入されると、次にそれはＩｏＴデバイスの要件及び／又はＩｏＴデバイスのタイプに基づいて特定のフロー制御クラスに関連付けられる。

【０１７３】

デバイスごとのフロー制御パラメータ３０２０は、ローカルデータベース内にデバイスごとのフロー制御パラメータ３０１０のコピーを記憶するフロー制御管理ロジック２８１１を含むＩｏＴハブ１１０に分散されてもよい。一実施形態では、フロー制御管理２８１１は、それぞれのＩｏＴデバイス１０１～１０３から受信した及び／又はこれらに送信したデータトラフィック量を監視し得る。データトラフィック量が指定した閾値に到達した場合は（デバイスごとのフロー制御パラメータ３０１０によって示されるように）、次に、ＩｏＴハブ１１０は、ＩｏＴデバイスに一定期間送信を中断するように指示してもよく、かつ／又はＩｏＴデバイスからのトラフィックを単純にブロックしてもよい。

【０１７４】

特定のＩｏＴデバイスが指定した閾値より上のレベルで送信／受信している場合、これはＩｏＴデバイスが誤動作していることを示す場合がある。したがって、一実施形態では、ＩｏＴサービス１２０は、コマンドを送信してＩｏＴデバイスをリセットしてもよい。デバイスが、依然として閾値より上のレベルで通信している場合、次にＩｏＴサービス１２０は、パッチなどのソフトウェア更新をＩｏＴデバイスに送信してもよい。ひとたび更新されたソフトウェアがインストールされると、ＩｏＴデバイスはリセットされ、新しいソフトウェアによって初期化される。加えて、通知は、ＩｏＴデバイスが誤動作していることをユーザに知らせるために、ＩｏＴサービスからユーザデバイスへ送信され得る。

【０１７５】

一実施形態では、ＩｏＴハブ１１０は、データ通信閾値に到達したという事実にもかかわらず、特定のタイプのデータトラフィックを許可し得る。例えば、一実施形態では、ＩｏＴハブ１１０は、たとえＩｏＴデバイスが閾値に到達したとしても特定のタイプの「優先度の高い」通知を許可することになる。例として、ＩｏＴデバイスがドアの錠又はドアエントリ検出器である場合は、次いで特定の条件下で（例えば、家が監視されているとき）、ＩｏＴハブ１１０は、ＩｏＴデバイスが使用されているドアを誰かが開けたことを示すデータを通過させ得る。同様に、ＩｏＴデバイスが、熱及び／又は煙検出器である場合は、次いで、ＩｏＴハブ１１０は、（例えば、温度が閾値に到達したので）アラーム状態を示すデータを通過させ得る。様々な他のタイプの「優先度の高い」通知（例えば、潜在的に危険な状態を示すものなど）は、現在のフロー制御の状態に関わらず、ＩｏＴハブ１１０によって通過され得る。一実施形態では、これらの「優先度の高い」通知は、後述される異なる属性を使用して識別される。

【０１７６】

本発明の一実施形態に従う方法が図３１に示されている。本方法は、上記のシステムアーキテクチャとの関連で実装され得るが、いかなる特定のシステムアーキテクチャにも限定されない。

【０１７７】

３１０１において、フロー制御パラメータは、各ＩｏＴデバイスに対して指定される。一実施形態では、またＩｏＴデバイスは、そこに関連付けられる指定したフロー制御パラメータのセットを有する特定のＩｏＴデバイス「クラス」に割り当てられ得る。３１０２において、フロー制御パラメータは、ＩｏＴシステム内のＩｏＴハブに記憶される。一実施形態では、それぞれのハブは、すべてのＩｏＴデバイスパラメータのサブセット（例えば、ローカルにプロビジョニングされたＩｏＴデバイスのパラメータのみ）を記憶し得る。

【０１７８】

ＩｏＴハブが、３１０３において判断される、特定のＩｏＴデバイスが指定したフロー制御パラメータの外で動作していることを検出した場合は、次に、３１０４において、ＩｏＴハブは、ＩｏＴデバイスとの更なる通信を一時的にやめることになる（例えば、ＩｏＴデバイスとＩｏＴサービスとの間の通信をブロックする）。加えて、上述したように、ＩｏＴサービス及び／又はＩｏＴハブは、ＩｏＴデバイスを再起動すること、及び／又はＩｏＴデバイスにソフトウェア更新をインストールすることによって問題を改善するための対策を施し得る。

属性クラスを使用してモノのインターネット（ＩｏＴ）デバイス及びトラフィックを管理するためのシステム及び方法

【０１７９】

異なるＩｏＴデバイスは、所定の場所において異なる機能を実行するために使用され得る。例えば、特定のＩｏＴデバイスを使用して、温度及び状態（例えば、オン／オフ状態）などのデータを収集し、このデータをＩｏＴサービスに戻して報告してもよく、そこでデータはエンドユーザによってアクセスされ得る、又は様々なタイプのアラート状態を生成するために使用され得る。この実装を可能にするために、本発明の一実施形態は、異なる属性クラスのタイプを使用して収集したデータ、システムデータ、及び他の形式のデータを管理する。

【０１８０】

図３２は、シリアルペリフェラルインタフェース（ＳＰＩ）バスなどのシリアルインタフェース３２１６を介して、マイクロコントローラユニット（ＭＣＵ）３２１５と通信する安全な無線通信モジュール３２１８を含むＩｏＴデバイスの一実施形態を例示する。安全な無線通信モジュール３２１８は、上述した技術を使用してＩｏＴサービス１２０との安全な通信を管理し、ＭＣＵ　３２１５は、ＩｏＴデバイス１０１の特定用途向け機能を実施するためのプログラムコードを実行する。

【０１８１】

一実施形態では、様々な異なる属性のクラスを使用して、ＩｏＴデバイスによって収集されたデータ及びＩｏＴデバイスに関するシステム構成を管理する。具体的には、図３２に示される例では、属性は、アプリケーション属性３２１０、システム属性３２１１、及び優先度通知属性３２１２を含む。一実施形態では、アプリケーション属性３２１０は、ＩｏＴデバイス１０１によって実施される特定用途向け機能に関連した属性を含む。例えば、ＩｏＴデバイスがセキュリティセンサを含む場合は、次にアプリケーション属性３２１０は、ドア又は窓が開けられたかどうかを示す２進値を含んでもよい。ＩｏＴデバイスが温度センサを含む場合は、次にアプリケーション属性３２１０は、現在の温度を示す値を含んでもよい。実質的に無制限の数の、他の特定用途向け属性を定義することができる。一実施形態では、ＭＣＵ　３２１５は、特定用途向けプログラムコードを実行し、特定用途向け属性３２１０へのアクセスを備えているだけである。例えば、アプリケーション開発者は、安全な無線通信モジュール３２１８と共にＩｏＴデバイス１０１を購入し、ＭＣＵ　３２１５によって実行されるアプリケーションプログラムコードを設計してもよい。結果的に、アプリケーション開発者は、アプリケーション属性へのアクセスを有することが必要になるが、後述される他の属性のタイプへのアクセスを有する必要はない。

【０１８２】

一実施形態では、システム属性３２１１は、ＩｏＴデバイス１０１及びＩｏＴシステムのための操作上及び構成の属性を定義するために使用される。例えば、システム属性は、ネットワーク構成設定（例えば、上述したフロー制御パラメータなど）、デバイスＩＤ、ソフトウェアバージョン、アドバタイジング間隔の選択、セキュリティ実装機能（上述のような）及びＩｏＴデバイス１０１をＩｏＴサービスと安全に通信できるようにするために必要な様々な他の低レベルの変数を含み得る。

【０１８３】

一実施形態では、優先度通知属性のセット３２１２は、それらの属性と関連付けられた重要度又は重大度のレベルに基づいて定義される。例えば、特定の属性が、閾値に到達する温度値などの危険な状態と関連付けられている場合は（例えば、ユーザが偶然にストーブをつけっぱなしにするとき、又はユーザの家の熱センサがトリガとなるとき）、この属性は、次いで優先度通知属性クラスに割り当てられ得る。上述したように、優先度通知属性は、他の属性とは異なって扱われ得る。例えば、特定の優先度通知属性が閾値に到達すると、ＩｏＴハブによって実装される現在のフロー制御機構に関わらず、ＩｏＴハブは、属性の値をＩｏＴサービスに渡し得る。一実施形態では、優先度通知属性がまたきっかけとなって、ＩｏＴサービスは、ユーザに対する通知及び／又はユーザの家若しくは企業内のアラーム状態（例えば、潜在的に危険な状態のユーザに警告する）を生成し得る。

【０１８４】

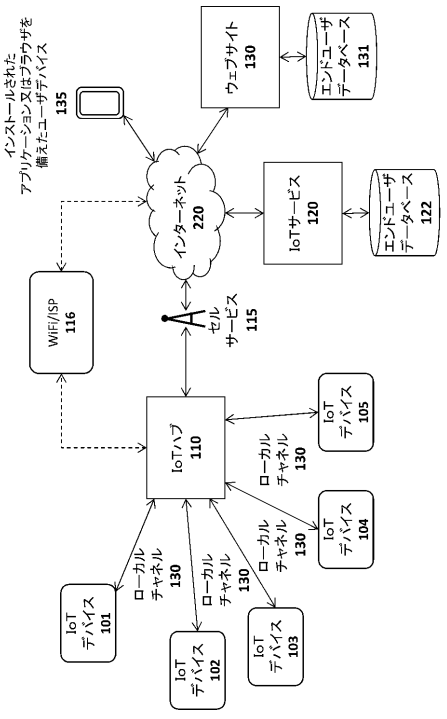
図３２に例示したように、一実施形態では、アプリケーション属性３２１０、システム属性３２１１、及び優先度通知属性３２１２の現在の状態は、ＩｏＴサービス１２０上のデバイスデータベース２８５１内で重複／ミラーリングされている。例えば、ＩｏＴデバイス１０１の属性のうちの１つにおける変更が更新されると、安全な無線通信モジュール３２１８は、変更をＩｏＴサービス１２０上のデバイス管理ロジック３０２１に伝達し、デバイス管理ロジックは、直ぐに反応してデバイスデータベース２８５１内の属性の値を更新する。加えて、ユーザがＩｏＴサービスの属性のうちの１つを更新すると（例えば、現在の状態又は望ましい温度などの条件を調節する）、属性変更は、デバイス管理ロジック３０２１から、安全な無線通信モジュール３２１８へ送信され、次にデバイス管理ロジックは属性のローカルコピーを更新する。このように、属性は、ＩｏＴデバイス１０１とＩｏＴサービス１２０との間で一貫性のある方式で維持される。属性はまた、インストールされたＩｏＴアプリ若しくはアプリケーションを有するユーザデバイスを介して、及び／又は１つ以上の外部サービス３２７０によってＩｏＴサービス１２０からアクセスされ得る。上述したように、ＩｏＴサービス１２０は、アプリケーションプログラミングインタフェース（ＡＰＩ）を公開して、様々な異なる属性のクラスへのアクセスを提供し得る。

【０１８５】
加えて、一実施形態では、優先度通知処理ロジック３０２２は、優先度通知属性３２１２に関する通知の受信に応じて、ルールベースの動作を実行してもよい。例えば、優先度通知属性が危険な状態を示す場合（例えば、ユーザによって残されるアイロン又はストーブなど）、次に優先度通知処理ロジック３０２２は、危険なデバイスをオフにしようと試みるルールのセットを実装してもよい（例えば、「オフ」コマンドを可能な場合デバイスに送信する）。一実施形態では、優先度通知処理ロジック３０２２は、危険なデバイスをオフにするかどうかを決定するために現在のユーザの場所など他の関連するデータを利用してもよい（例えば、危険なデバイスは「オン」状態にあるときユーザが家を出ているのを検出した場合）。加えて、優先度通知処理ロジック３０２２は、ユーザのクライアントデバイスにアラート状態を送信して、ユーザに状態を通知してもよい。様々な他のルールセットのタイプは、潜在的に危険な、ないしは望ましくない状態に対処しようと試みるために、優先度通知処理ロジック３０２２によって実装され得る。
【０１８６】
図３２には、ＢＴＬＥ属性３２０５及び属性アドレスデコーダ３２０７のセットもまた示される。一実施形態では、ＢＴＬＥ属性３２０５を使用して、図１９～図２０に関して上述したように読み取り及び書き込みポートを確立し得る。属性アドレスデコーダ３２０７は、各属性に関連した固有ＩＤコードを読み取って、どの属性が受信されている／送信されているかを決定し、それに応じて属性を処理する（例えば、属性が安全な無線通信モジュール３２１８内で記憶されている場所を識別する）。
【０１８７】
本発明の実施形態は、上で説明した様々な工程を含み得る。本工程は、汎用又は特殊目的のプロセッサに本工程を実行させるために使用され得る、機械実行可能な命令において具現化することができる。代替的に、これらの工程は、工程を実行するためのハードワイヤードロジックを含む特定のハードウェア構成要素によって、又はプログラミングされたコンピュータ構成要素及びカスタムハードウェア構成要素の任意の組み合わせによって、実行することができる。
【０１８８】
本明細書に記載される場合に、命令は、ある特定の動作を行うように構成されるか、又は所定の機能若しくはソフトウェア命令が非一時的コンピュータ可読媒体中に具現化されたメモリ内に記憶されている、特定用途向け集積回路（ＡＳＩＣ）などの、ハードウェアの特定の構成を指し得る。したがって、図面に示される技術は、１つ以上の電子デバイス（例えば、エンドステーション、ネットワーク要素など）上に記憶及び実行されるコード及びデータを使用して実装され得る。そのような電子デバイスは、非一時的コンピュータ機械可読記憶媒体（例えば、磁気ディスク、光ディスク、ランダムアクセスメモリ、読み出し専用メモリ、フラッシュメモリデバイス、相変化メモリ）、並びに一時的なコンピュータ機械可読通信媒体（例えば、搬送波、赤外線信号、デジタル信号などの電気的、光学的、音響的又は他の形態の伝搬信号）などのコンピュータ機械可読記憶媒体を使用して、コード及びデータを記憶及び（内部で及び／又はネットワークを介して他の電子デバイスと）通信する。加えて、そのような電子デバイスは、典型的に、１つ以上の記憶デバイス（非一時的機械可読記憶媒体）、ユーザ入力／出力デバイス（例えば、キーボード、タッチスクリーン、及び／又はディスプレイ）、並びにネットワーク接続などの、１つ以上の他の構成要素に連結された１つ以上のプロセッサのセットを含む。プロセッサの組と他の構成要素との連結は、典型的には、１つ以上のバス及びブリッジ（バスコントローラとも呼ばれる）を通じて行われる。記憶デバイスとネットワークトラフィックを運ぶ信号のそれぞれは、１つ以上の機械可読記憶媒体及び機械可読通信媒体を表す。したがって、所与の電子デバイスの記憶デバイスは、その電子デバイスの１つ以上のプロセッサのセット上で実行するためのコード及び／又はデータを、典型的に記憶する。当然のことながら、本発明の実施形態の１つ以上の部分は、ソフトウェア、ファームウェア、及び／又はハードウェアの異なる組み合わせを使用して実装されてもよい。

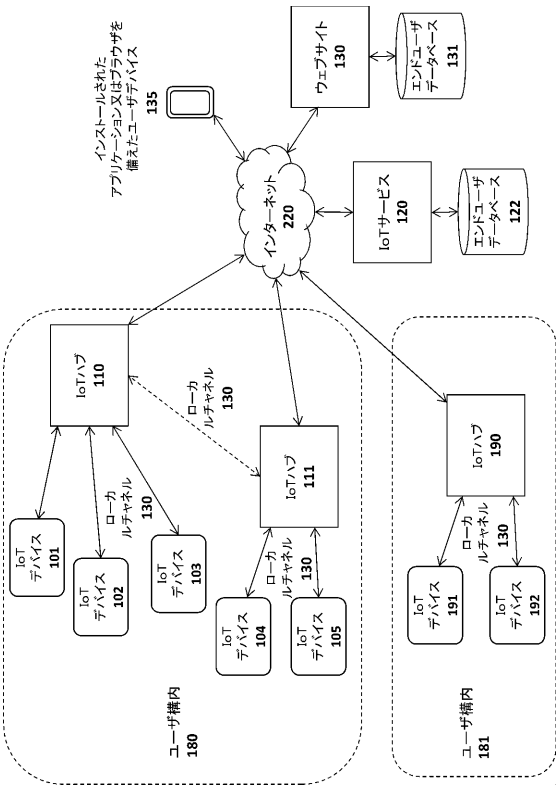
【０１８９】

この詳細な説明全体を通じて、説明を目的として、本発明の完全な理解を提供するために、多数の特定の詳細を記載した。しかしながら、本発明は、これらの具体的な詳細の一部がなくても実施され得ることは、当業者にとって明らかであろう。ある特定の例では、既知の構造及び機能は、本発明の主題を不明瞭にすることを回避するために、詳述しなかった。したがって、本発明の範囲及び趣旨は、以下の特許請求の範囲の観点から判断されるべきである。

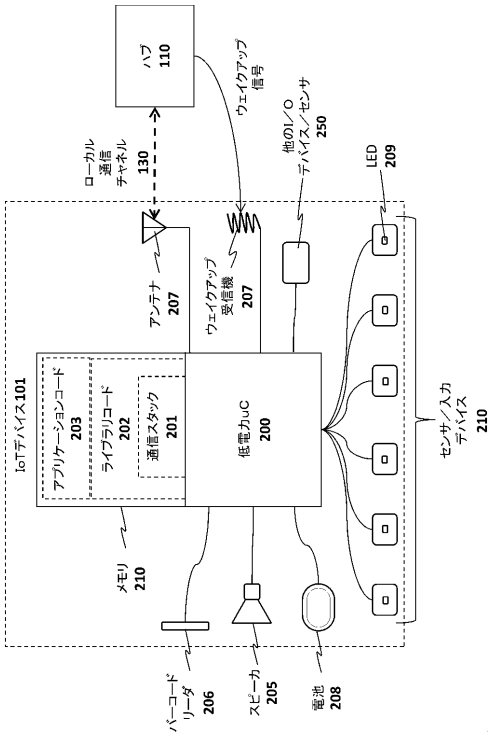
【図１Ａ】



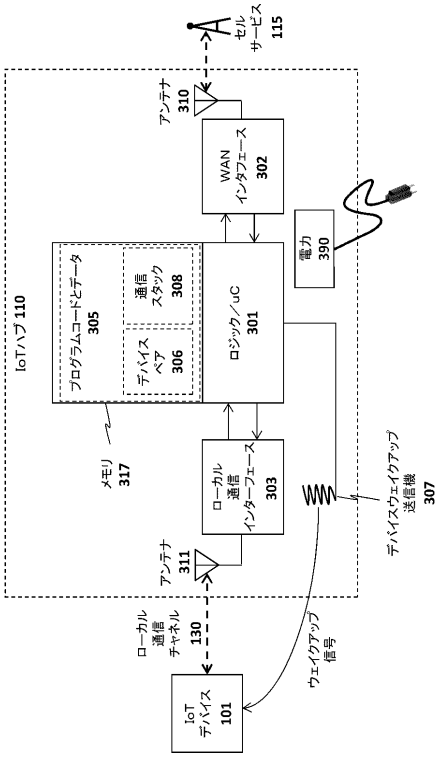
【図１Ｂ】



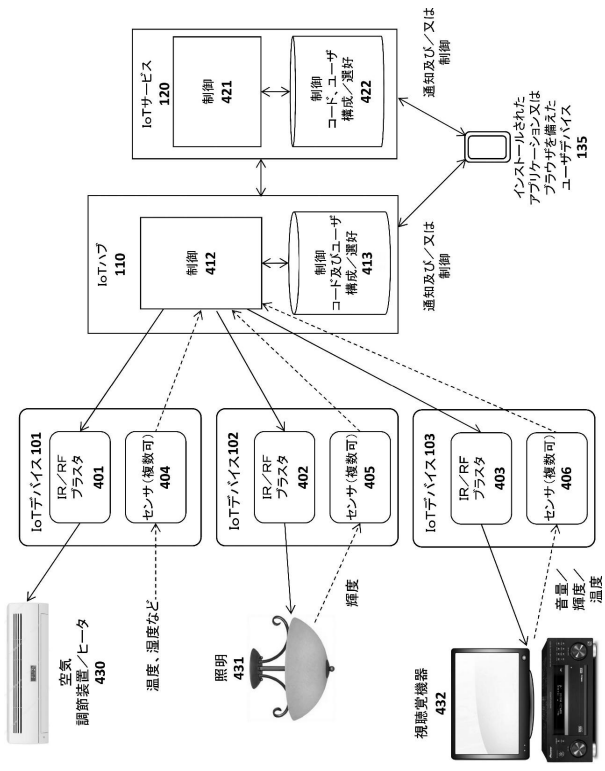
【図２】


ハブ
110
ウェイクアップ
信号
他のI／O
デバイス／センサ
250
LED
209
ローカル
通信
チャネル
130
アンテナ
207
ウェイクアップ
受信機
207
IoTデバイス101
アプリケーションコード
203
ライブラリコード
202
通信スタック
201
低電力uC
200
センサ／入力
デバイス
210
メモリ
210
バーコード
リーダ
206
スピーカ
205
電池
208

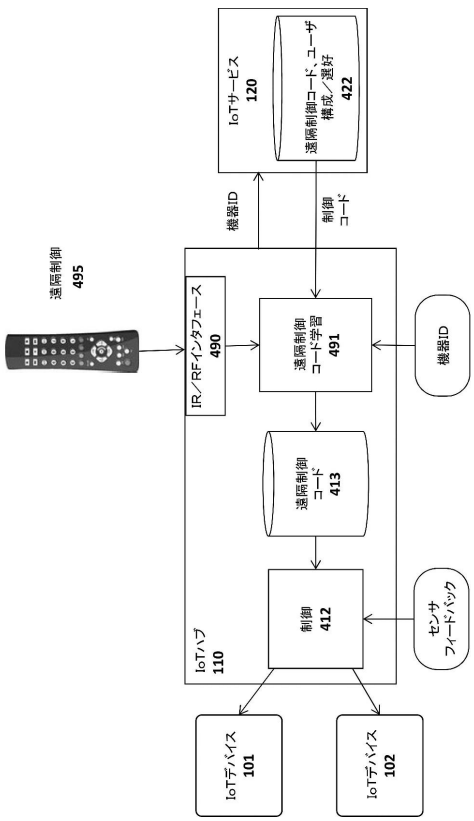
【図３】


セル
サービス
115
アンテナ
310
WAN
インタフェース
302
IoTハブ 110
プログラムコードとデータ
305
通信
スタック
308
デバイス
ペア
306
ロジック／uC
301
電力
390
メモリ
317
ローカル
通信
インターフェース
303
デバイスウェイクアップ
送信機
307
アンテナ
311
ローカル
通信
チャネル
130
ウェイクアップ
信号
IoT
デバイス
101

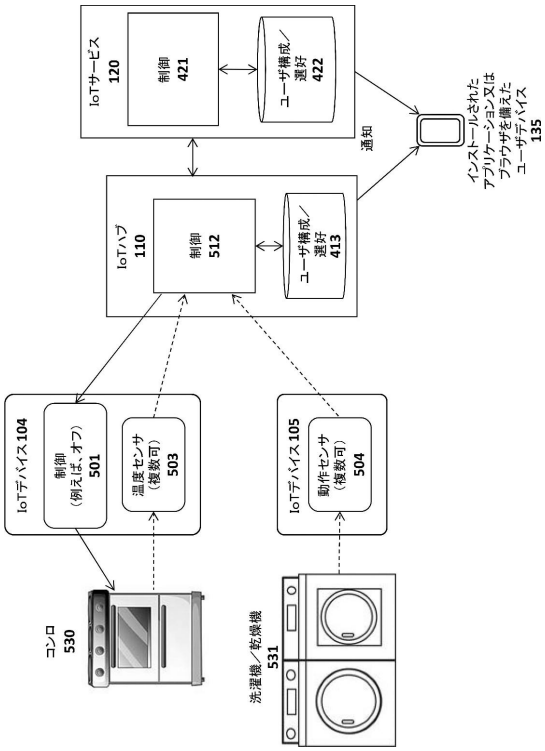
【図４Ａ】


IoTサービス
120
制御
421
制御
コード、ユーザ
構成／選好
422
通知及び／又は
制御
インストールされた
アプリケーション又は
ブラウザを備えた
ユーザデバイス
135
IoTハブ
110
制御
412
制御
コード及びユーザ
構成／選好
413
通知及び／又は
制御
IoTデバイス101
IR／RF
プラスタ
401
センサ（複数可）
404
IoTデバイス102
IR／RF
プラスタ
402
センサ（複数可）
405
IoTデバイス103
IR／RF
プラスタ
403
センサ（複数可）
406
空気
調節装置／ヒータ
430
温度、湿度など
照明
431
輝度
視聴覚機器
432
音量／
輝度／
温度

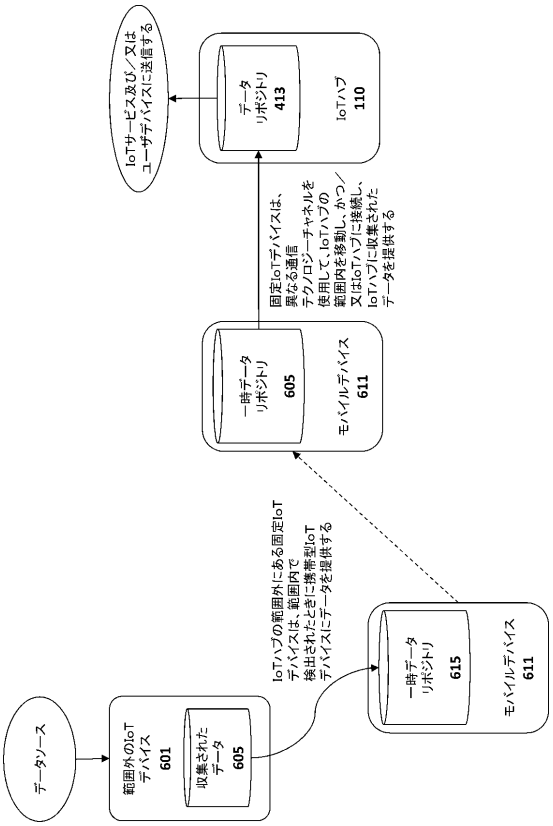
【図４Ｂ】


IoTサービス
120
遠隔制御コード、ユーザ
構成／選好
422
機器ID
制御
コード
遠隔制御
495
IR／RFインタフェース
490
遠隔制御
コード学習
491
機器ID
遠隔制御
コード
413
制御
412
センサ
フィードバック
IoTハブ
110
IoTデバイス
101
IoTデバイス
102

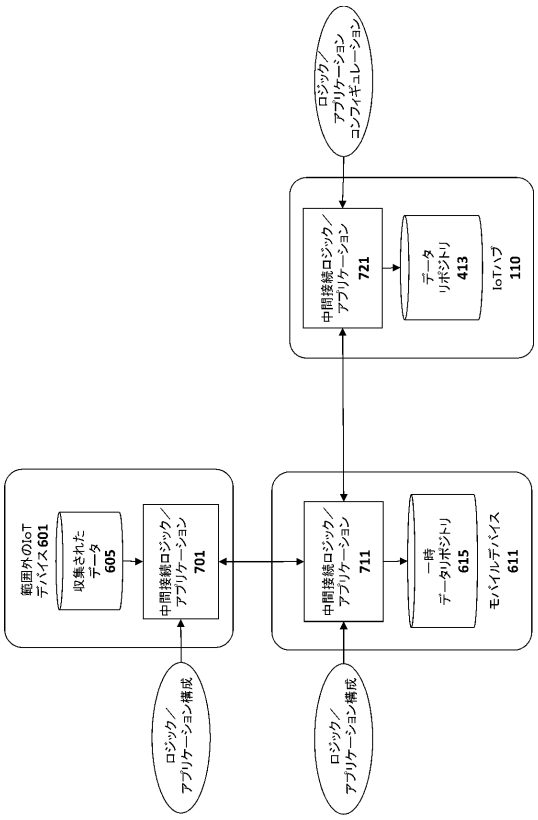
【図５】


IoTサービス
120
制御
421
ユーザ構成／
選好
422
通知
インストールされた
アプリケーション又は
ブラウザを備えた
ユーザデバイス
135
IoTハブ
110
制御
512
ユーザ構成／
選好
413
IoTデバイス104
制御
（例えば、オフ）
501
温度センサ
（複数可）
503
IoTデバイス105
動作センサ
（複数可）
504
コンロ
530
洗濯機／乾燥機
531

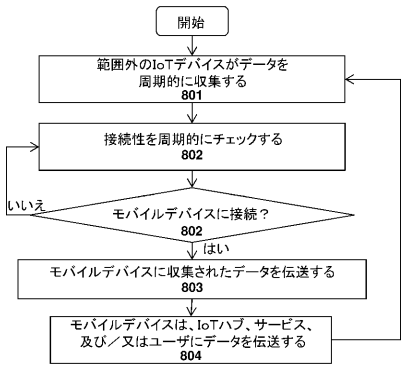
【図６】


IoTサービス及び／又は
ユーザデバイスに送信する
データ
リポジトリ
413
IoTハブ
110
固定IoTデバイスは、
異なる通信
テクノロジーチャネルを
使用して、IoTハブの
範囲内を移動し、かつ／
又はIoTハブに接続し、
IoTハブに収集された
データを提供する
一時データ
リポジトリ
605
モバイルデバイス
611
IoTハブの範囲外にある固定IoT
デバイスは、範囲内で
検出されたときに携帯型IoT
デバイスにデータを提供する
データソース
範囲外のIoT
デバイス
601
収集された
データ
605
一時データ
リポジトリ
615
モバイルデバイス
611

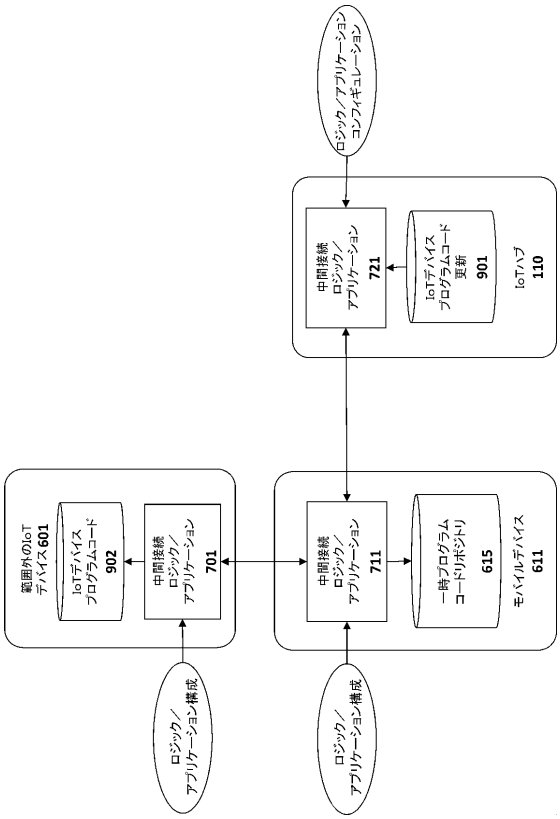
【図７】


ロジック／
アプリケーション
コンフィギュレーション
中間接続ロジック／
アプリケーション
721
データ
リポジトリ
413
IoTハブ
110
範囲外のIoT
デバイス601
収集された
データ
605
中間接続ロジック／
アプリケーション
701
中間接続ロジック／
アプリケーション
711
一時
データリポジトリ
615
モバイルデバイス
611
ロジック／
アプリケーション構成
ロジック／
アプリケーション構成

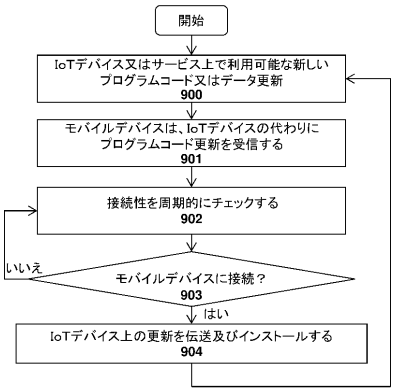
【図８】


開始
範囲外のIoTデバイスがデータを
周期的に収集する
801
接続性を周期的にチェックする
802
モバイルデバイスに接続？
802
いいえ
はい
モバイルデバイスに収集されたデータを伝送する
803
モバイルデバイスは、IoTハブ、サービス、
及び／又はユーザにデータを伝送する
804

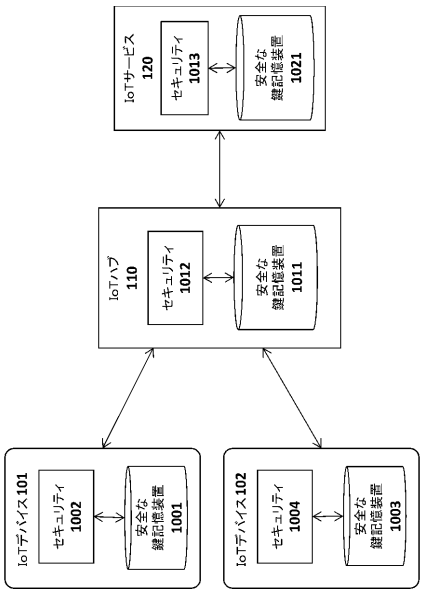
【図９Ａ】



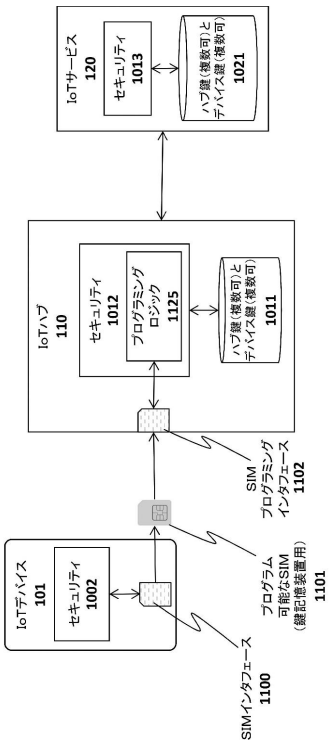
【図９Ｂ】



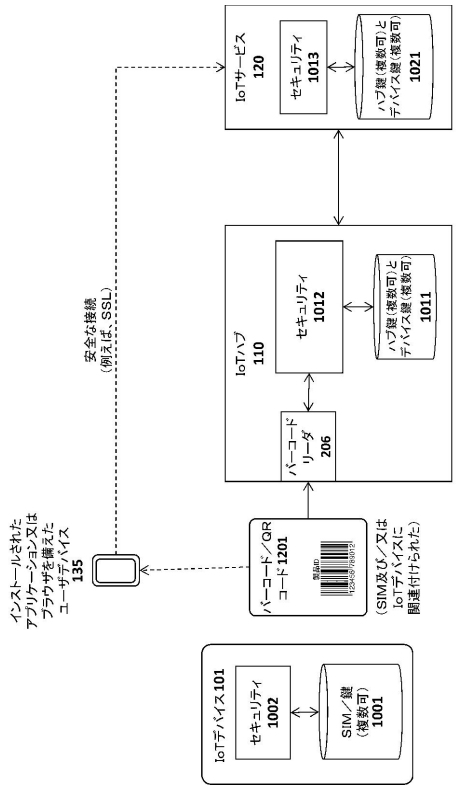
【図１０】



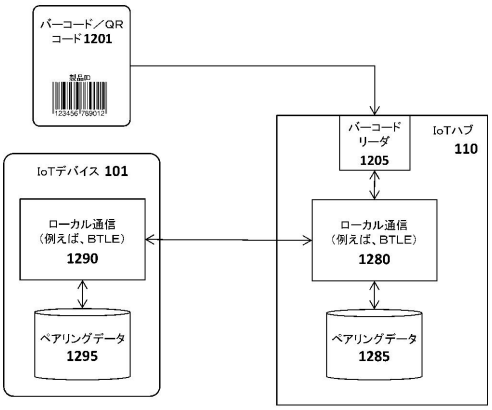
【図１１】



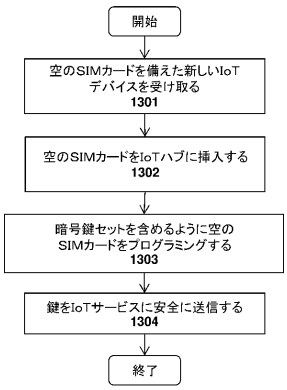
【図１２Ａ】



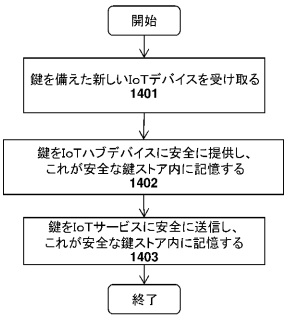
【図１２Ｂ】



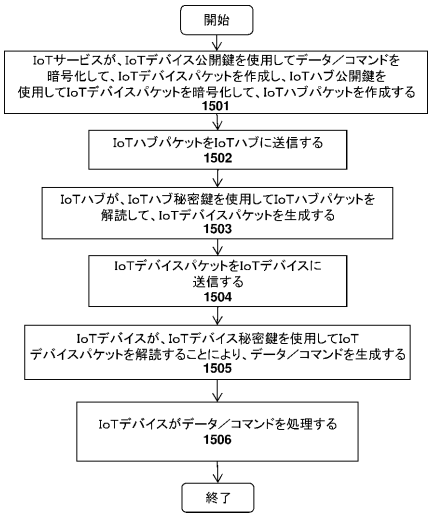
【図１３】



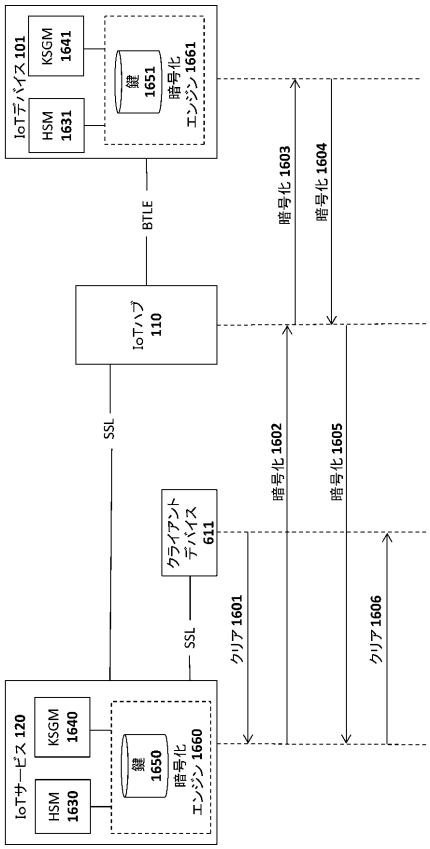
【図１４】



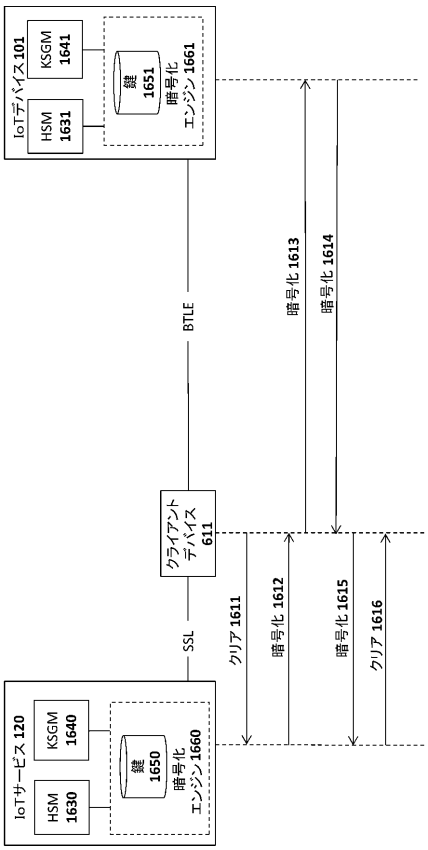
【図１５】



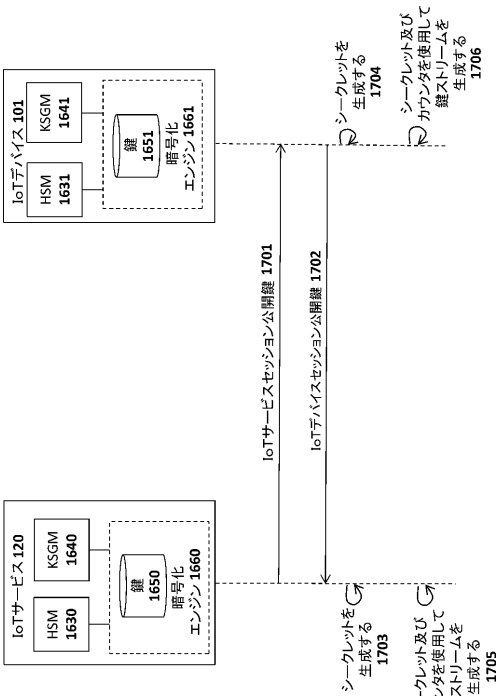
【図１６Ａ】



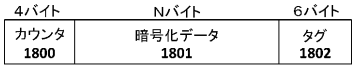
【図１６Ｂ】



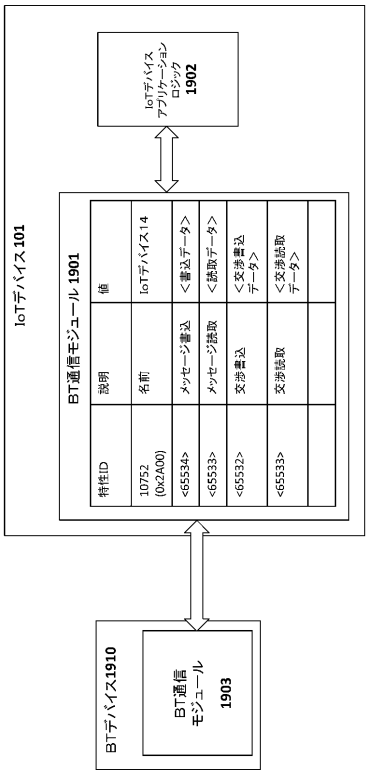
【図１７】



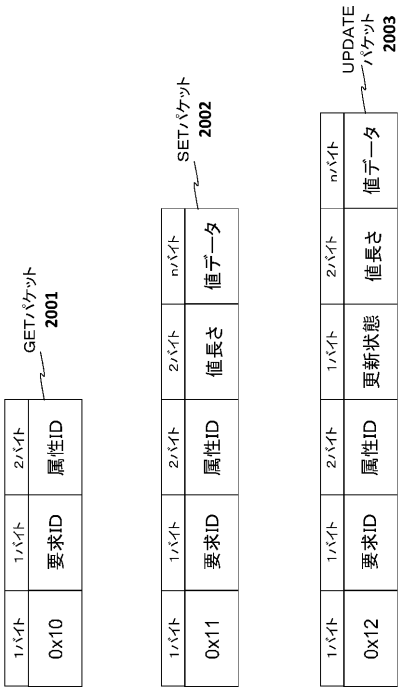
【図１８】



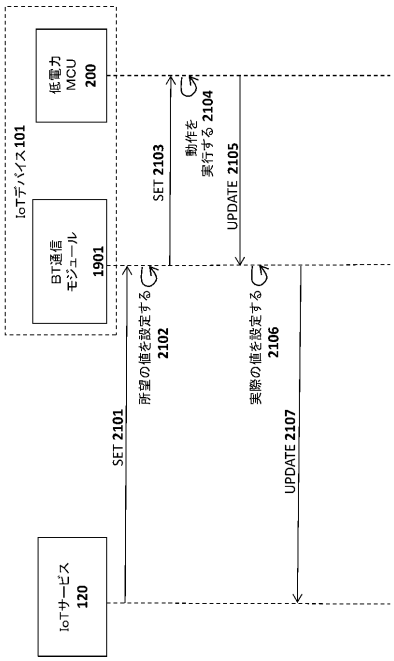
【図１９】



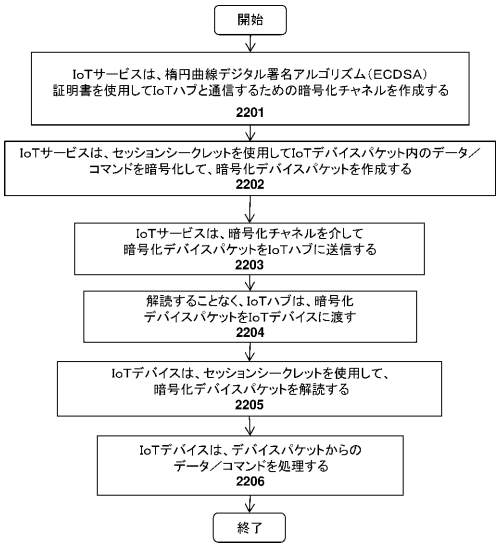
【図２０】



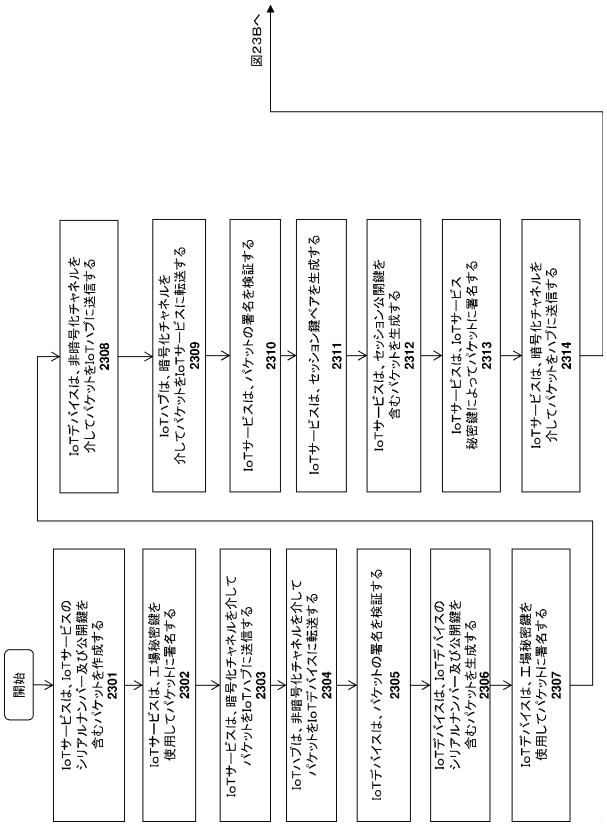
【図２１】



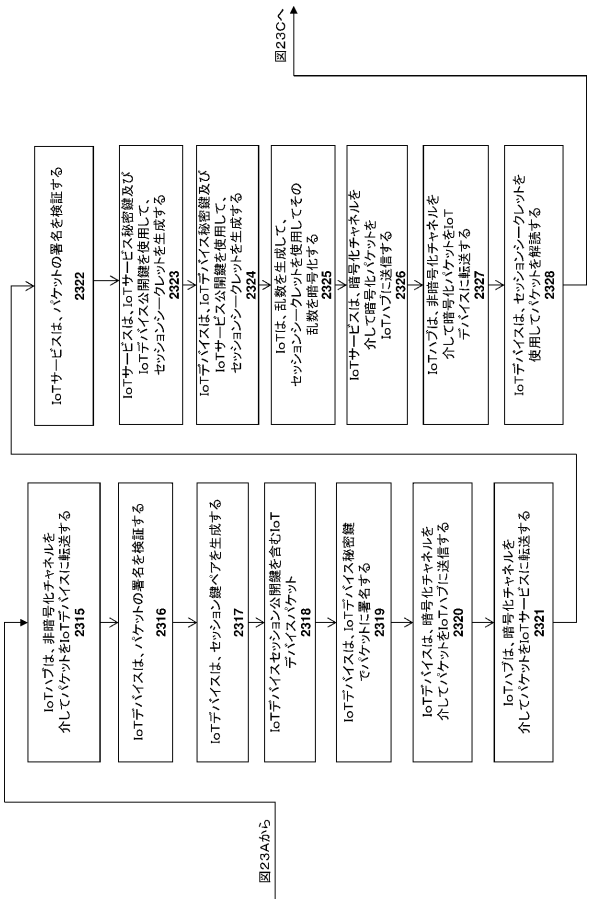
【図２２】



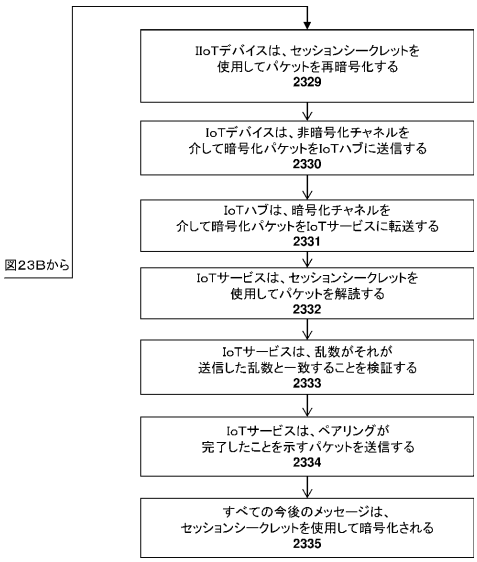
【図２３Ａ】



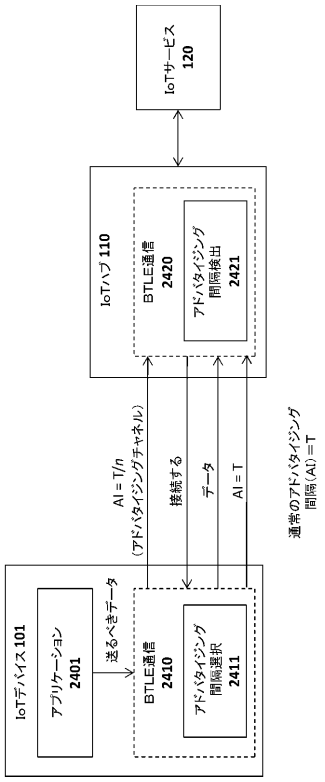
【図２３Ｂ】



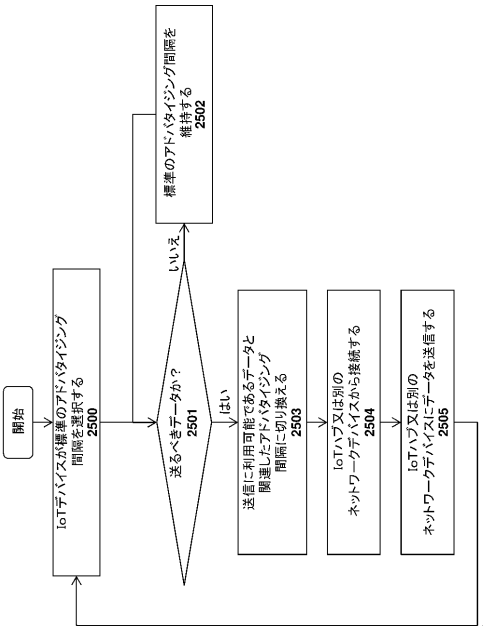
【図２３Ｃ】



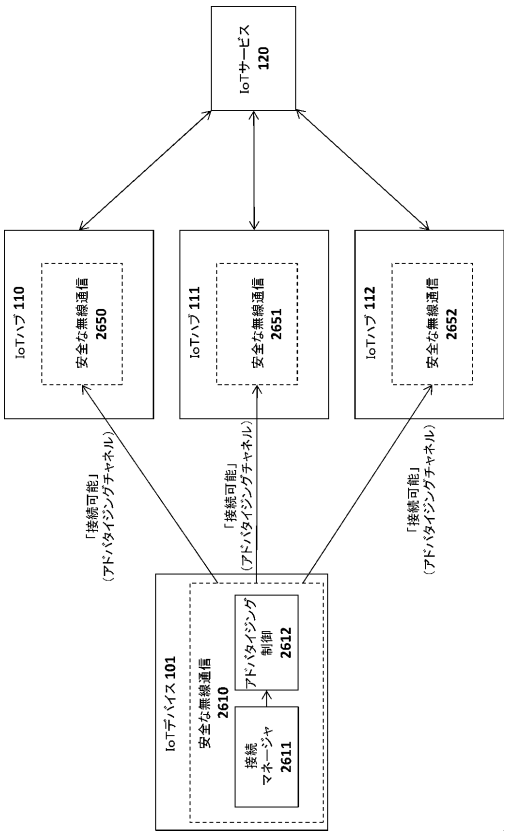
【図２４】



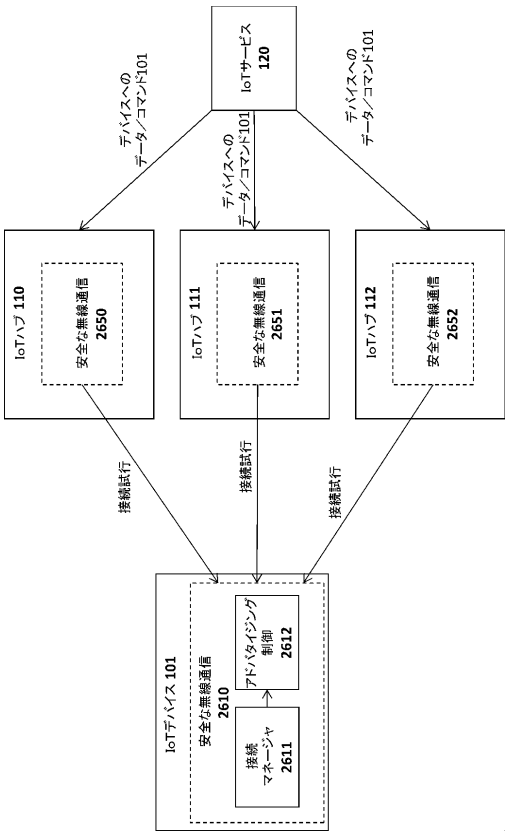
【図２５】



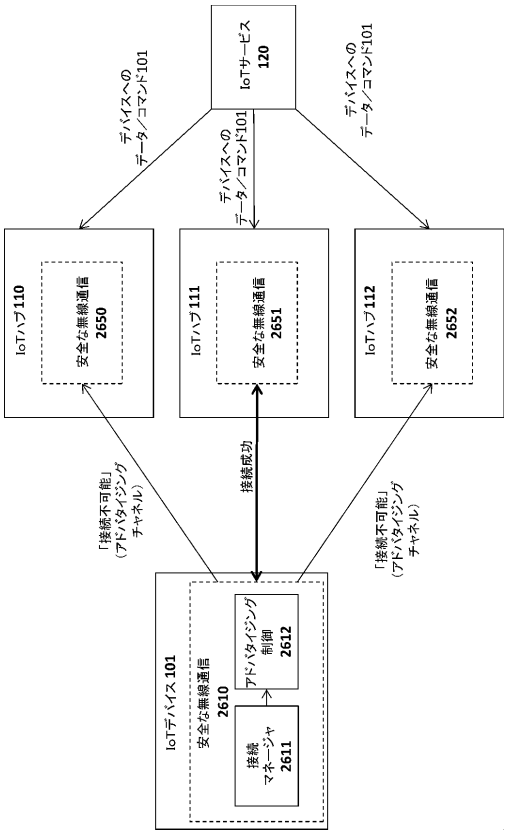
【図２６Ａ】



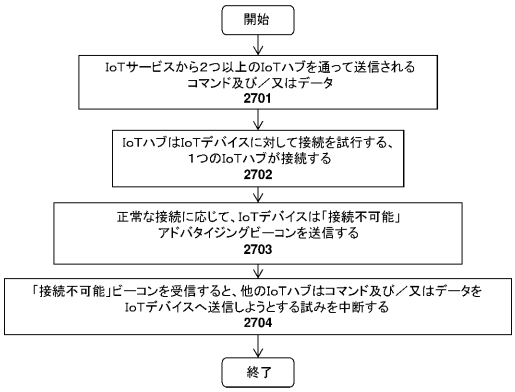
【図２６Ｂ】



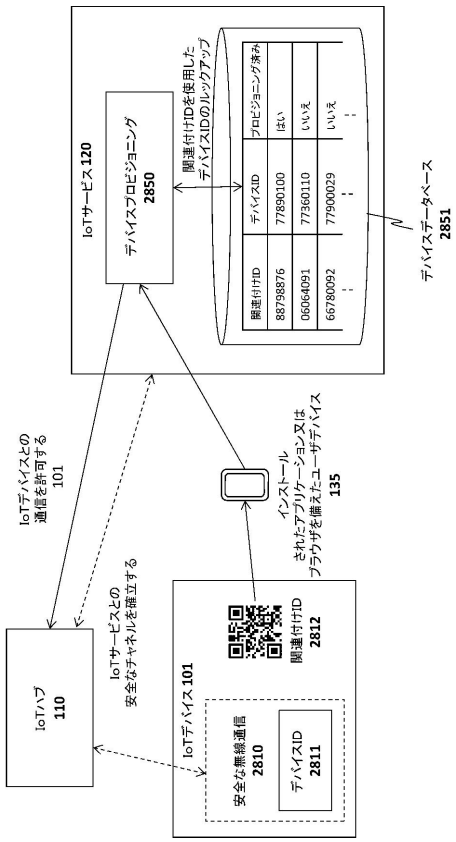
【図２６Ｃ】



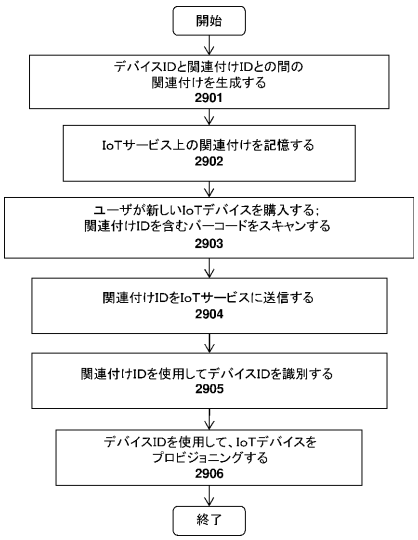
【図２７】



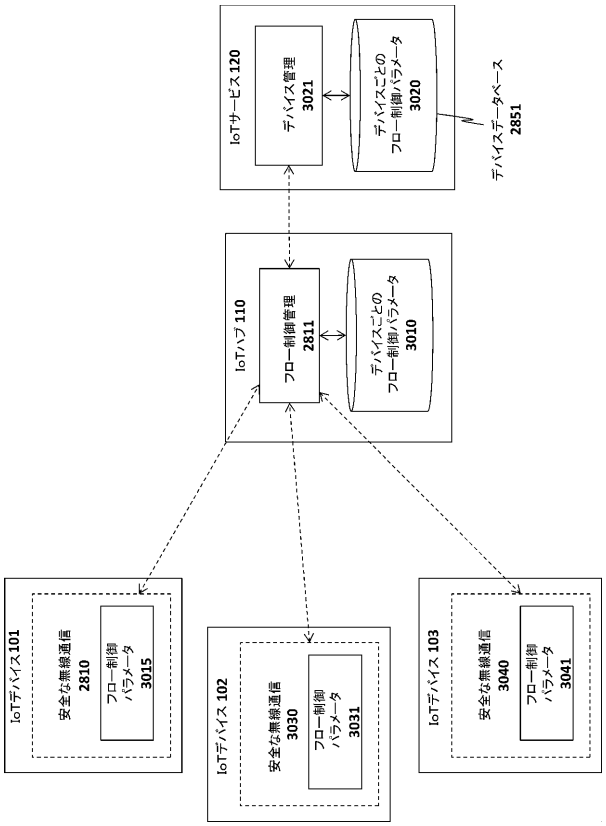
【図２８】



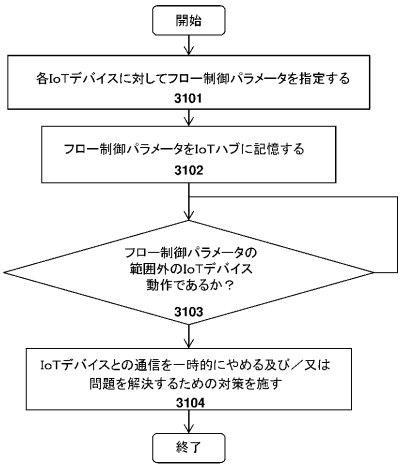
【図２９】



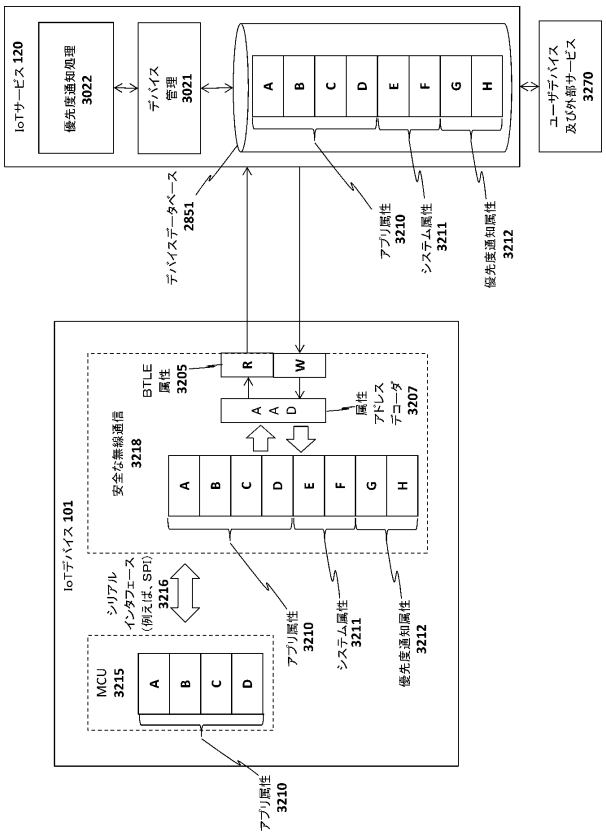
【図３０】


IoTサービス 120
デバイス管理 3021
デバイスごとのフロー制御パラメータ 3020
デバイスデータベース 2851
IoTハブ 110
フロー制御管理 2811
デバイスごとのフロー制御パラメータ 3010
IoTデバイス 101
安全な無線通信 2810
フロー制御パラメータ 3015
IoTデバイス 102
安全な無線通信 3030
フロー制御パラメータ 3031
IoTデバイス 103
安全な無線通信 3040
フロー制御パラメータ 3041

【図３１】


開始
各IoTデバイスに対してフロー制御パラメータを指定する 3101
フロー制御パラメータをIoTハブに記憶する 3102
フロー制御パラメータの範囲外のIoTデバイス動作であるか？ 3103
IoTデバイスとの通信を一時的にやめる及び／又は問題を解決するための対策を施す 3104
終了

【図３２】


IoTサービス 120
優先度通知処理 3022
デバイス管理 3021
A B C D E F G H
ユーザデバイス及び外部サービス 3270
デバイスデータベース 2851
アプリ属性 3210
システム属性 3211
優先度通知属性 3212
IoTデバイス 101
安全な無線通信 3218
BTLE属性 3205
R W
AAD
属性アドレスデコーダ 3207
シリアルインタフェース（例えば、SPI） 3216
MCU 3215
A B C D

フロントページの続き

(51)Int.Cl. FI
G06F 13/00 ＺＩＴ
G06F 13/00 ５１０Ｃ
H04Q 9/00 ３０１Ｄ

(31)優先権主張番号 14/967,964
(32)優先日 平成27年12月14日(2015.12.14)
(33)優先権主張国・地域又は機関
米国(US)

(74)代理人 100086771
弁理士 西島 孝喜
(74)代理人 100109070
弁理士 須田 洋之
(74)代理人 100109335
弁理士 上杉 浩
(74)代理人 100120525
弁理士 近藤 直樹
(74)代理人 100139712
弁理士 那須 威夫
(74)代理人 100196612
弁理士 鎌田 慎也
(72)発明者 ブリット ジョー
アメリカ合衆国 カリフォルニア州 ９４０２２ ロスアルトス エル カミノ リアル ４９７０ スイート ２１０
(72)発明者 ジマーマン スコット
アメリカ合衆国 カリフォルニア州 ９４０２２ ロスアルトス エル カミノ リアル ４９７０ スイート ２１０

審査官 木村 雅也

(56)参考文献 米国特許出願公開第２０１５／００６３１６４（ＵＳ，Ａ１）
米国特許出願公開第２０１５／０２２２６２１（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １３／００
Ｇ０６Ｆ ２１／４５
Ｈ０４Ｌ ９／０８
Ｈ０４Ｌ ９／３２
Ｈ０４Ｑ ９／００