(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7635225号
(P7635225)

(45)発行日　令和7年2月25日(2025.2.25)

(24)登録日　令和7年2月14日(2025.2.14)

(51)国際特許分類

		FI		
H04L 9/32 (2006.01)		H04L	9/32	200Z
H04L 9/08 (2006.01)		H04L	9/08	F
		H04L	9/08	D

請求項の数　13　（全51頁）

(21)出願番号　特願2022-525179(P2022-525179)
(86)(22)出願日　令和2年10月5日(2020.10.5)
(65)公表番号　特表2023-500259(P2023-500259A)
(43)公表日　令和5年1月5日(2023.1.5)
(86)国際出願番号　PCT/IB2020/059319
(87)国際公開番号　WO2021/084347
(87)国際公開日　令和3年5月6日(2021.5.6)
審査請求日　令和5年9月12日(2023.9.12)
(31)優先権主張番号　1915841.9
(32)優先日　令和1年10月31日(2019.10.31)
(33)優先権主張国・地域又は機関
英国(GB)

(73)特許権者　318001991
エヌチェーン ライセンシング アーゲー
スイス・６３００・ツーク・グラーフェナウヴェーク・６
(74)代理人　100107766
弁理士　伊東 忠重
(74)代理人　100070150
弁理士　伊東 忠彦
(74)代理人　100135079
弁理士　宮崎 修
(72)発明者　マッケイ，アレグザンダー
イギリス ダブリュー１ダブリュー ８エーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内

最終頁に続く

(54)【発明の名称】　ブロックチェーントランザクションを使用した通信プロトコル

(57)【特許請求の範囲】

【請求項１】

第１のネットワークに参加する許可をリクエスタに与えるためのコンピュータ実装方法であって、

前記第１のネットワークは、ブリッジノードのセットと、前記ブリッジノードのセットのうちの１つまたは複数によって制御可能なデバイスのセットとを含み、各ブリッジノードはブロックチェーンネットワークのそれぞれのノードでもあり、前記コンピュータ実装方法は、登録機関によって実行され、前記リクエスタはコンピュータ機器であり、かつ、前記登録機関はコンピュータ機器であり、前記方法は、

第１のブロックチェーントランザクションを生成するステップであり、前記第１のブロックチェーントランザクションは、前記登録機関の第１の公開鍵にリンクされた署名を含む入力と、第１の証明書を含む第１の出力と、前記登録機関の第２の公開鍵にロックされた第２の出力とを含み、前記第１の証明書は、前記リクエスタに割り当てられた識別子を含む、ステップと、

第２のブロックチェーントランザクションを生成するステップであり、前記第２のブロックチェーントランザクションは、前記第１のブロックチェーントランザクションの前記第２の出力を参照する入力を含み、かつ、前記登録機関の前記第２の公開鍵にリンクされた署名を含む、ステップと、

ブロックチェーンに含めるために、前記第２のブロックチェーントランザクションを、前記ブロックチェーンネットワークに送信するステップと、

前記ブロックチェーンに含めるために、前記第１のブロックチェーントランザクションを、前記ブロックチェーンネットワークに送信するステップと、
を含む、方法。
【請求項２】
前記第１の出力は、前記登録機関の前記第２の公開鍵にタイムロックされ、タイムロックは、所定の期間後まで前記第１の出力がロック解除されるのを防ぐ、
請求項１に記載の方法。
【請求項３】
前記第１の出力は、少なくとも前記登録機関の前記第２の公開鍵および異なる公開鍵にロックされる、
請求項１または２に記載の方法。
【請求項４】
前記方法は、
前記第１のブロックチェーントランザクションのトランザクション識別子を許可リクエスタに送信するステップを含む、
請求項１から３のいずれか一項に記載の方法。
【請求項５】
前記証明書は、暗号鍵で暗号化され、前記暗号鍵は、前記登録機関によって生成される、
請求項１から４のいずれか一項に記載の方法。
【請求項６】
前記方法は、
前記ネットワークに参加するための要求を前記リクエスタから受信するステップであり、前記要求は、１つまたは複数のクレデンシャルを含む、ステップと、
前記１つまたは複数のクレデンシャルに基づいて、前記要求を妥当性確認するステップであり、前記第１のブロックチェーントランザクションを前記生成するステップは、前記要求が有効であることを条件とするステップと、
を含む、請求項１から５のいずれか一項に記載の方法。
【請求項７】
前記ブリッジノードのセットは、マスタノードと、前記マスタノードによって制御可能な中間ノードのセットと、を含み、
前記登録機関は、前記マスタノードである、
請求項１から６のいずれか一項に記載の方法。
【請求項８】
前記ブリッジノードのセットは、マスタノードと、前記マスタノードによって制御可能な中間ノードのセットとを含み、前記リクエスタは、前記マスタノードである、
請求項１から７のいずれか一項に記載の方法。
【請求項９】
前記リクエスタは、前記ブロックチェーンネットワークのそれぞれのノードであり、かつ、
前記証明書は、許可リクエスタに割り当てられた公開鍵を含む、
請求項１から８のいずれか一項に記載の方法。
【請求項１０】
前記リクエスタは、前記第１のネットワークの１つまたは複数のブリッジノードによって制御可能なデバイスであり、前記方法は、
証明書のセットを前記リクエスタに送信するステップ、を含み、前記セット内の各証明書は、前記ブリッジノードのセットのうちのそれぞれの１つに送信されている、ステップ、を含む、
請求項１から７のいずれか一項に記載の方法。
【請求項１１】
前記方法は、

前記第１の証明書を前記ブリッジノードのセットのうちの１つまたは複数に送信するステップ、を含む、
請求項１から１０のいずれか一項に記載の方法。
【請求項１２】
１つまたは複数のメモリユニットを備えるメモリと、
１つまたは複数の処理ユニットを備える処理装置と、を備え、
前記メモリは、前記処理装置上で実行されるように構成されたコードを記憶し、前記コードは、前記処理装置上にあるときに、請求項１から１１のいずれか一項に記載の方法を実行するように構成される、
コンピュータ機器。
【請求項１３】
コンピュータ可読ストレージ上に具現化され、請求項１２に記載のコンピュータ機器上で実行されると、請求項１から１１のいずれか一項に記載の方法を実行するように構成されたコンピュータプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、例えば、要求エンティティがネットワークにアクセスするために、ネットワークに参加する許可を要求エンティティに与えるための方法に関する。
【背景技術】
【０００２】
ブロックチェーンは、分散型データ構造の形態を指し、ブロックチェーンの複製コピーが、ピアツーピア（Ｐ２Ｐ）ネットワーク内の複数のノードの各々において維持される。ブロックチェーンは、データブロックのチェーンを含み、各ブロックは１つまたは複数のトランザクションを含む。各トランザクションは、１つまたは複数のブロックにまたがり得るシーケンス内の先行するトランザクションを指し示し得る。トランザクションは、「マイニング」として知られるプロセスによって新しいブロックに含まれるためにネットワークにサブミットされ得、このプロセスは、複数のマイニングノードの各々が、「プルーフオブワーク」を実行しようと競うことと、すなわち、ブロックに含まれるのを待っている保留中のトランザクションのプールに基づいて暗号パズルを解くことを伴う。
【０００３】
従来、ブロックチェーンにおけるトランザクションは、デジタル資産、すなわち価値の蓄蔵として機能するデータを伝達するために使用される。しかしながら、ブロックチェーンは、ブロックチェーンの上に追加の機能を重ねるために活用することもできる。例えば、ブロックチェーンプロトコルは、トランザクションの出力における追加のユーザデータの格納を可能にし得る。最新のブロックチェーンでは、単一のトランザクション内に格納可能な最大データ容量が増えており、より複雑なデータを組み込むことが可能である。例えば、これを使用して、ブロックチェーンに電子文書を格納したり、さらにはオーディオまたはビデオデータを格納したりすることができる。
【０００４】
ネットワーク内の各ノードは、フォワード、マイニング、および格納という３つの役割のうちのいずれか１つ、２つ、またはすべてを担うことができる。フォワーディングノードは、ネットワークのノード全体にトランザクションを伝搬する。マイニングノードは、ブロックへのトランザクションのマイニングを実行する。ストレージノードは各々が、ブロックチェーンのマイニングされたブロックのそれら自体のコピーを格納する。ブロックチェーンにトランザクションを記録させるために、当事者は、伝搬されるべきネットワークのノードのうちの１つにトランザクションを送信する。トランザクションを受信するマイニングノードは、トランザクションを新しいブロックにマイニングしようと競い合い得る。各ノードは、同じノードプロトコルを尊重するように構成され、そのノードプロトコルには、トランザクションが有効であるための１つまたは複数の条件が含まれる。無効な

トランザクションは、伝搬もブロックへのマイニングもされない。トランザクションが妥当性確認（ｖａｌｉｄａｔｅ）され、それによってブロックチェーン上に受け入れられたと仮定すると、追加のユーザデータは、したがって、不変の公開記録としてＰ２Ｐネットワーク内のノードの各々に格納されたままになる。

【発明の概要】

【０００５】

モノのインターネット（ＩｏＴ）技術により、物理的デバイスのネットワークは、イベントを監視し、人間の介入なしにデータを交換することができる。ＩｏＴ技術の開発の動機としては、広範囲の産業にわたり従来の監視および制御方法に取って代わるリアルタイムのデータ収集および自動制御機構の必要性が挙げられる。ＩｏＴシステムは、大量のデータを生成し、ネットワークスケーラビリティ、強力なサイバーセキュリティ、信頼性のある接続性、および最小ネットワーク待ち時間を伴うシステムに依拠する。

【０００６】

現在、集中型アーキテクチャモデルは、ＩｏＴネットワーク内のノードを認証し、認可し、接続するために広く使用されている。このようなモデルは攻撃に対して脆弱であり、単一障害点として機能する。集中型システムが危険にさらされた場合、ＩｏＴネットワークにアクセスする許可が、悪意のあるデバイスに与えられる可能性、および／または既存のデバイスから除去される可能性がある。悪意のあるデバイスにＩｏＴネットワークへのアクセスが与えられた場合、そのデバイスは、例えば、機密データを得たり、またはネットワークを破壊したりすることができる。

【０００７】

ピアツーピア（Ｐ２Ｐ）アーキテクチャは、集中型アーキテクチャと比較して、より安全かつ効率的なソリューションを提供し、それによって、ネイバーは、それらの間で集中型ノードまたはエージェントを使用することなく、互いに直接対話する。ブロックチェーン技術は、安全なＰ２Ｐ通信の基礎であり、ＩｏＴシステムの開発に革命をもたらすことが期待されている。しかしながら、ＩｏＴデバイスのための次世代のブロックチェーンベースのシステムが実現される場合、ＩｏＴのためのブロックチェーンベースの制御方法は、オープンシステムに固有の課題を克服する必要がある。これらは、ブロックチェーン自体には固有でない可能性があるデータプライバシーおよびデバイス保護／制御機構を含む。

【０００８】

本明細書に開示される一態様によれば、第１のネットワークに参加する許可をリクエスタに与えるためのコンピュータ実装方法が提示され、第１のネットワークは、ブリッジノードのセットと、ブリッジノードのセットのうちの１つまたは複数によって制御可能なデバイスのセットとを含み、各ブリッジノードはブロックチェーンネットワークのそれぞれのノードでもあり、方法は、登録機関によって実行され、第１のブロックチェーントランザクションを生成することと、ここで、第１のブロックチェーントランザクションは、登録機関の第１の公開鍵にリンクされた署名を含む入力と、第１の証明書を含む第１の出力とを含み、第１の証明書は、リクエスタに割り当てられた識別子を含み、ブロックチェーンに含めるために第１のブロックチェーントランザクションをブロックチェーンネットワークに送信することとを含む。

【０００９】

第１のネットワーク（例えば、ＩｏＴネットワーク）は、１つまたは複数のブリッジノードと、ブリッジノードのうちの１つまたは複数によって制御され得る１つまたは複数のデバイスとを含む。ブリッジノードは、ブロックチェーンネットワークのノードでもある。すなわち、それらは、ＩｏＴネットワーク（例えば、他のネットワークノードおよびデバイスと通信するため）およびブロックチェーンネットワーク（例えば、トランザクションをブロックチェーンに送信し、ブロックチェーンに記録されたトランザクションを識別して読み出すため）の両方に接続することができるという意味で、ＩｏＴネットワークおよびブロックチェーンネットワークの一部である。これらのノードは、第１のネットワークとブロックチェーンネットワークとの間のゲートウェイまたはブリッジとして機能する

。それらはまた、ブロックチェーンネットワークのマイニングノード、フォワーディングノード、またはストレージノードの役割を果たす必要はないが、それも除外されるわけではない。いくつかの例では、第１のネットワークのデバイスのうちの１つまたは複数はまた、ブロックチェーンネットワークのノードであり得る。

【００１０】

登録機関（ＩｏＴネットワークのブリッジノードであってもそうでなくてもよい）は、ブロックチェーンネットワークのノードである。すなわち、登録機関はブロックチェーンに接続され、ブロックチェーンネットワークにトランザクションを送信するように構成される。登録機関は、リクエスタ（要求エンティティ）に証明書を付与する責任があり、これらの証明書は、リクエスタがネットワークに参加する許可を与える。ここで、リクエスタがネットワークに参加すると、リクエスタは、例えば、第１のネットワークに接続された他のノードおよび／またはデバイスとの通信などのタスクを実行することができる。登録機関は、新しいデバイスのクレデンシャルを検証（ｖｅｒｉｆｙ）し、デジタル証明書を発行することができる信頼できる当事者である。登録機関は、参入障壁として機能し、ＩｏＴネットワーク上に新しいデバイスが入るのを許可または拒否する。

【００１１】

第１のブロックチェーントランザクションは、トランザクションの出力において、リクエスタに発行された証明書を含む。証明書は、リクエスタに固有の識別子（「デバイスＩＤ」）を含む。デバイスＩＤは、擬似ランダムなバイト列であり得る。登録機関は、トランザクション（したがって証明書）をブロックチェーンネットワークにブロードキャストする。第１のブロックチェーントランザクションは、第１の入力内の署名によって、登録機関にリンクされる。署名は、登録機関の公開鍵にリンクされる。換言すると、署名は、登録機関の公開鍵に対応する秘密鍵に基づいて生成される。登録機関のみが秘密鍵の知識を有するので、登録機関のみがその署名でトランザクションに署名することができる。換言すると、証明書を偽造することはできない。したがって、証明書は、真正なエンティティに対してのみ付与され得る。リクエスタが、証明書の発行を受けて、第１のネットワークの他のノードまたはデバイスと通信するとき、それらのノードおよびデバイスは、リクエスタが証明書を発行されているかどうかをチェックすることができ、したがって、リクエスタが真正なエンティティであるかどうかをチェックすることができる。

【００１２】

いくつかの実施形態では、リクエスタはまた、ブロックチェーンネットワークのノードである。その場合、証明書は、要求ノードに割り当てられた公開鍵を含み得る。換言すると、公開鍵は、証明された公開鍵である。要求ノードが、その証明された公開鍵またはそれから導出された鍵を使用して他のノードと通信する（例えば、上記公開鍵のうちの１つで署名されたノードまたはブロックチェーンにトランザクションを送信する）場合、他のノードは、それらのトランザクションを送信するノードが真正なノードであると確信することができる。

【００１３】

本明細書で開示される別の態様によれば、第１のネットワークに参加する許可を要求するためのコンピュータ実装方法が提供され、第１のネットワークは、ブリッジノードのセットと、ブリッジノードのセットのうちの１つまたは複数によって制御可能なデバイスのセットとを含み、各ブリッジノードはブロックチェーンネットワークのそれぞれのノードでもあり、方法は、リクエスタによって実行され、第１のネットワークに参加する要求を登録機関に送信することと、第１の証明書を取得することとを含み、ここで、証明書は、登録機関によって発行され、リクエスタに割り当てられた識別子を含む。

【図面の簡単な説明】

【００１４】

本開示の実施形態の理解を助け、そのような実施形態がどのように実施され得るかを示すために、単なる例として、添付の図面を参照する。

【図１】図１は、ブロックチェーンを実装するためのシステムの概略ブロック図である。

【図２】図２は、ブロックチェーンに記録され得るトランザクションのいくつかの例を概略的に示す。
【図３】図３は、ブロックチェーンを実装するための別のシステムの概略ブロック図である。
【図４】図４は、出力ベースモデルのノードプロトコルにしたがってトランザクションを処理するためのノードソフトウェアの一部分の概略ブロック図である。
【図５】図５は、ＩｏＴネットワークとブロックチェーンネットワークとの間の重複を概略的に示す。
【図６】図６は、階層的ネットワークトポロジーを概略的に示す。
【図７ａ】図７ａは、部分的なコマンドトランザクションを概略的に示す。
【図７ｂ】図７ｂは、完全なコマンドトランザクションを概略的に示す。
【図８ａ】図８ａは、代替的な部分的なトランザクションを概略的に示す。
【図８ｂ】図８ｂは、代替的な完全なトランザクションを概略的に示す。
【図９】図９は、コマンド要求および応答サイクルを概略的に示す。
【図１０ａ】図１０ａは、サーバノードからスレーブノードに送信される部分的なコマンドトランザクションを概略的に示す。
【図１０ｂ】図１０ｂは、サーバノードからスレーブノードに送信される完全なコマンドトランザクションを概略的に示す。
【図１１ａ】図１１ａは、コマンド要求トランザクションを概略的に示す。
【図１１ｂ】図１１ｂは、コマンド承認トランザクションを概略的に示す。
【図１２ａ】図１２ａは、暗号化された部分的なコマンドトランザクションを概略的に示す。
【図１２ｂ】図１２ｂは、暗号化された完全なコマンドトランザクションを概略的に示す。
【図１３】図１３は、例示的なコマンドデータフォーマットを示す。
【図１４】図１４は、例示的なピアツーピア印刷システムを概略的に示す。
【図１５ａ】図１５ａは、ピアツーピア印刷システムで使用するための例示的なトランザクションを概略的に示す。
【図１５ｂ】図１５ｂは、ピアツーピア印刷システムで使用するための例示的なトランザクションを概略的に示す。
【図１５ｃ】図１５ｃは、ピアツーピア印刷システムで使用するための例示的なトランザクションを概略的に示す。
【図１６ａ】図１６ａは、証明書トランザクションおよび例示的な証明書フォーマットを概略的に示す。
【図１６ｂ】図１６ｂは、証明書トランザクションおよび例示的な証明書フォーマットを概略的に示す。
【発明を実施するための形態】
【００１５】
例示的なシステムの概要
図１は、一般的にブロックチェーン１５０を実装するための例示的なシステム１００を示す。システム１００は、典型的にはインターネットなど広域インターネットワークであるパケット交換ネットワーク１０１を含む。パケット交換ネットワーク１０１は、パケット交換ネットワーク１０１内にピアツーピア（Ｐ２Ｐ）オーバーレイネットワーク１０６を形成するように構成された複数のノード１０４を含む。各ノード１０４は、ピアのコンピュータ機器を含み、ノード１０４のうちの異なるものが異なるピアに属する。各ノード１０４は、１つまたは複数のプロセッサ、例えば１つまたは複数の中央処理装置（ＣＰＵ）、アクセラレータプロセッサ、特定用途向けプロセッサおよび／またはフィールドプログラマブルゲートアレイ（ＦＰＧＡ）を備える処理装置を含む。各ノードはまた、メモリ、すなわち、１つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備える。メモリは、１つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ソリッドステートドライブ（ＳＳＤ）、フラッシュメモリもしくはＥＥ

ＰＲＯＭなどの電子媒体、および／または光ディスクドライブなどの光学媒体を使用する１つまたは複数のメモリユニットを備え得る。

【００１６】

ブロックチェーン１５０は、データブロック１５１のチェーンを含み、ブロックチェーン１５０のそれぞれのコピーは、Ｐ２Ｐネットワーク１６０内の複数のノードの各々において維持される。チェーン内の各ブロック１５１は、１つまたは複数のトランザクション１５２を含み、この文脈におけるトランザクションは、データ構造の一種を指す。データ構造の性質は、トランザクションモデルまたは方式の一部として使用されるトランザクションプロトコルのタイプに依存する。所与のブロックチェーンは、典型的には、全体を通して１つの特定のトランザクションプロトコルを使用する。１つの一般的なタイプのトランザクションプロトコルでは、各トランザクション１５２のデータ構造は、少なくとも１つの入力と少なくとも１つの出力とを含む。各出力は、出力が暗号的にロックされている（ロック解除され、それによって償還または使用されるためにはそのユーザの署名を必要とする）ユーザ１０３に属するデジタル資産の量を表す額を指定する。各入力は、先行するトランザクション１５２の出力を指し示し、それによってトランザクションをリンクする。

【００１７】

ノード１０４のうちの少なくともいくつかは、トランザクション１５２をフォワードし、それによって伝搬するフォワーディングノード１０４Ｆの役割を引き受ける。ノード１０４のうちの少なくともいくつかは、ブロック１５１をマイニングするマイナー１０４Ｍの役割を引き受ける。ノード１０４のうちの少なくともいくつかは、ストレージノード１０４Ｓ（「フルコピー」ノードと呼ばれることもある）の役割を引き受け、その各々が、同じブロックチェーン１５０のそれぞれのコピーをそれぞれのメモリに格納する。各マイナーノード１０４Ｍはまた、ブロック１５１にマイニングされるのを待っているトランザクション１５２のプール１５４を維持する。所与のノード１０４は、フォワーディングノード１０４、マイナー１０４Ｍ、ストレージノード１０４Ｓ、またはこれらのうちの２つもしくはすべての任意の組合せであり得る。

【００１８】

所与の現在のトランザクション１５２ｊにおいて、入力（または各入力）は、トランザクションのシーケンスにおける先行するトランザクション１５２ｉの出力を参照するポインタを含み、この出力が現在のトランザクション１５２ｊにおいて償還または「使用」されるべきであることを指定する。一般に、先行するトランザクションは、プール１５４または任意のブロック１５１内の任意のトランザクションであり得る。先行するトランザクション１５２ｉは、現在のトランザクションが有効となるために存在および妥当性確認される必要があるが、先行するトランザクション１５２ｉは、現在のトランザクション１５２ｊが作成されるときまたはネットワーク１０６に送信されるときに必ずしも存在する必要はない。したがって、本明細書における「先行する（ｐｒｅｃｅｄｉｎｇ）」は、ポインタによってリンクされた論理シーケンスにおける先行するものを指し、必ずしも時間シーケンスにおける作成または送信の時間を指すものではなく、したがって、トランザクション１５２ｉ、１５２ｊが順不同に作成または送信されることを必ずしも除外するものではない（オーファントランザクションに関する以下の説明を参照）。先行するトランザクション１５２ｉは、先のトランザクション（ａｎｔｅｃｅｄｅｎｔ　ｔｒａｎｓａｃｔｉｏｎ）または先行したトランザクション（ｐｒｅｄｅｃｅｓｓｏｒ　ｔｒａｎｓａｃｔｉｏｎ）とも呼ばれる。

【００１９】

現在のトランザクション１５２ｊの入力はまた、先行するトランザクション１５２ｉの出力がロックされるユーザ１０３ａの署名を含む。次に、現在のトランザクション１５２ｊの出力は、新しいユーザ１０３ｂに暗号的にロックされ得る。したがって、現在のトランザクション１５２ｊは、先行するトランザクション１５２ｉの入力において定義された額を、現在のトランザクション１５２ｊの出力において定義されたように、新しいユーザ

１０３ｂに転送することができる。場合によっては、トランザクション１５２は、複数のユーザ（残り（ｃｈａｎｇｅ）を与えるためにそのうちの１人が元のユーザ１０３ａであり得る）間で入力額を分割するために複数の出力を有し得る。場合によっては、トランザクションはまた、１つまたは複数の先行するトランザクションの複数の出力からの額をまとめ、現在のトランザクションの１つまたは複数の出力に再分配するために複数の入力を有することができる。

【００２０】

上記は、「出力ベース」トランザクションプロトコルと称され得、未使用トランザクション出力（ＵＴＸＯ）タイププロトコルと称されることもある（ここでは出力はＵＴＸＯと称される）。ユーザの総残高は、ブロックチェーンに格納された任意の１つの数字で定義されるのではなく、代わりに、ユーザは、ブロックチェーン１５１内の多くの異なるトランザクション１５２全体に散在しているそのユーザのすべてのＵＴＸＯの値を照合するための特別な「ウォレット」アプリケーション１０５を必要とする。

【００２１】

トランザクションプロトコルの代替的なタイプは、アカウントベースのトランザクションモデルの一部として、「アカウントベース」プロトコルと称され得る。アカウントベースの場合、各トランザクションは、一連の過去のトランザクションにおける先行するトランザクションのＵＴＸＯを参照することによってではなく、絶対アカウント残高を参照することによって転送されるべき額を定義する。すべてのアカウントの現在の状態は、ブロックチェーンとは別にマイナーによって格納され、絶えず更新される。そのようなシステムでは、トランザクションは、アカウントの実行中のトランザクションタリー（「ポジション」とも呼ばれる）を使用して順序付けられる。この値は、送信者によってその暗号署名の一部として署名され、トランザクション参照計算の一部としてハッシュされる。加えて、トランザクションにおける任意選択のデータフィールドも署名することができる。このデータフィールドは、例えば前のトランザクションＩＤがデータフィールドに含まれている場合、前のトランザクションを指し示し得る。

【００２２】

いずれかのタイプのトランザクションプロトコルを用いて、ユーザ１０３が新しいトランザクション１５２ｊを成立させることを望む場合、ユーザは新しいトランザクションをユーザのコンピュータ端末１０２からＰ２Ｐネットワーク１０６のノード１０４（これは、今日では典型的にはサーバまたはデータセンタであるが、原理的には他のユーザ端末であってもよい）のうちの１つに送信する。このノード１０４は、ノード１０４の各々において適用されるノードプロトコルにしたがってトランザクションが有効であるかをチェックする。ノードプロトコルの詳細は、当該ブロックチェーン１５０において使用されているトランザクションプロトコルのタイプに対応し、全体としてトランザクションモデルを形成する。ノードプロトコルは、典型的には、新しいトランザクション１５２ｊ内の暗号署名が、トランザクション１５２の順序付けられたシーケンス内で前のトランザクション１５２ｉに依存する予想される署名と一致することをチェックすることをノード１０４に求める。出力ベースの場合、これは、新しいトランザクション１５２ｊの入力に含まれるユーザの暗号署名が、新しいトランザクションが使用する先行するトランザクション１５２ｉの出力において定義される条件と一致するかをチェックすることを含み得、この条件は、典型的には、新しいトランザクション１５２ｊの入力における暗号署名が、新しいトランザクションの入力が指し示す前のトランザクション１５２ｉの出力をロック解除することをチェックすることを少なくとも含む。いくつかのトランザクションプロトコルでは、条件は、入力および／または出力に含まれるカスタムスクリプトによって少なくとも部分的に定義され得る。代替的に、単にノードプロトコルのみによって固定されてもよく、またはこれらの組合せによるものであってもよい。いずれにしても、新しいトランザクション１５２ｊが有効である場合、現在のノードは、それをＰ２Ｐネットワーク１０６内のノード１０４のうちの１つまたは複数の他のノードにフォワードする。これらのノード１０４のうちの少なくともいくつかは、フォワーディングノード１０４Ｆとしても機能し、

同じノードプロトコルにしたがって同じテストを適用し、そのため新しいトランザクション１５２ｊを１つまたは複数のさらなるノード１０４にフォワードし、以下同様である。このようにして、新しいトランザクションはノード１０４のネットワーク全体に伝搬される。

【００２３】

出力ベースのモデルでは、所与の出力（例えば、ＵＴＸＯ）が使用されたかどうかの定義は、それがノードプロトコルにしたがって別の前方のトランザクション１５２ｊの入力によって有効に償還されたかどうかかである。トランザクションが有効であるための別の条件は、それが使用または償還しようと試みる先行する遷移１５２ｉの出力が、別の有効なトランザクションによってまだ使用／償還されていないことである。同様に、有効でない場合、トランザクション１５２ｊは、ブロックチェーンに伝搬も記録もされない。これは、使用者が同じトランザクションの出力を複数回使用しようとする二重支出を防止する。一方、アカウントベースのモデルは、アカウント残高を維持することによって二重支出を防止する。ここでも、トランザクション順序が定義されているので、アカウント残高は常に単一の定義された状態にある。

【００２４】

妥当性確認に加えて、ノード１０４Ｍのうちの少なくともいくつかはまた、「プルーフオブワーク」に支えられるマイニングとして知られるプロセスにおいてトランザクションのブロックを最初に作成しようと競い合う。マイニングノード１０４Ｍにおいて、新しいトランザクションが、ブロック内にまだ現れていない有効なトランザクションのプールに追加される。次いで、マイナーは、暗号パズルを解くことを試みることによって、トランザクション１５４のプールからトランザクション１５２の新しい有効ブロック１５１を組み立てようと競い合う。典型的には、これは、ノンスがトランザクション１５４のプールと連結されハッシュされたときにハッシュの出力が所定の条件を満たすような「ノンス」値を探索することを含む。例えば、所定の条件とは、ハッシュの出力が特定の所定の数の先行ゼロを有することであり得る。ハッシュ関数の特性は、その入力に対して予測不可能な出力を持つことである。したがって、この探索は、総当たりでしか実行することができないので、パズルを解こうとしている各ノード１０４Ｍでかなりの量の処理リソースを消費する。

【００２５】

最初にパズルを解いたマイナーノード１０４Ｍは、これをネットワーク１０６に公表し、後にネットワーク内の他のノード１０４によって容易にチェックすることができるその解を証明として提供する（ハッシュに対する解が与えられると、ハッシュの出力が条件を満たすことをチェックすることは簡単である）。勝者がパズルを解いたトランザクション１５４のプールは、次いで、ストレージノード１０４Ｓとして機能するノード１０４のうちの少なくともいくつかによって、そのような各ノードにおいて勝者が公表した解をチェックしたことに基づいて、ブロックチェーン１５０内に新しいブロック１５１として記録されるようになる。ブロックポインタ１５５はまた、チェーン内の前に作成されたブロック１５１ｎ－１を指し示す新しいブロック１５１ｎに割り当てられる。プルーフオブワークは、新たなブロック１５１を作成するのに多大な労力を要するので、二重支出のリスクを低減するのに役立ち、二重支出を含むブロックは他のノード１０４によって拒絶される可能性が高いので、マイニングノード１０４Ｍは、二重支出がそれらのブロックに含まれないようにインセンティブが与えられる。ブロック１５１は、作成されると、同じプロトコルにしたがってＰ２Ｐネットワーク１０６内の格納ノード１０４Ｓの各々で認識および維持されるので、修正することができない。ブロックポインタ１５５はまた、ブロック１５１にシーケンシャル順序を付与する。トランザクション１５２は、Ｐ２Ｐネットワーク１０６内の各ストレージノード１０４Ｓにおいて順序付けられたブロックに記録されるので、これは、トランザクションの不変の公開台帳を提供する。

【００２６】

任意の所与の時間にパズルを解こうと競い合う異なるマイナー１０４Ｍは、それらがい

つ解を探索し始めたかに応じて、任意の所与の時間におけるマイニングされていないトランザクションプール１５４の異なるスナップショットに基づいてそうすることができることに留意されたい。誰がそれぞれのパズルを最初に解いても、どのトランザクション１５２が次の新しいブロック１５１ｎに含まれるかを定義し、マイニングされていないトランザクションの現在のプール１５４が更新される。次いで、マイナー１０４Ｍは、新しく定義された未処理プール１５４からブロックを作成しようと競い合い続け、以下同様である。２人のマイナー１０４Ｍが互いに非常に短い時間内にパズルを解いて、ブロックチェーンの相反する見解が伝搬される場合に発生し得る任意の「フォーク」を解決するためのプロトコルも存在する。要するに、フォークのどちらのプロングが最も長く成長しても、最終的なブロックチェーン１５０となる。

【００２７】

ほとんどのブロックチェーンでは、勝利マイナー１０４Ｍには、（あるユーザから別のユーザにある額のデジタル資産を転送する通常のトランザクションとは対照的に）突如新しい量のデジタル資産を作成する特別なタイプの新しいトランザクションで自動的に報酬が与えられる。したがって、勝者ノードは、ある量のデジタル資産を「マイニング」したと言われる。この特別なタイプのトランザクションは、「生成」トランザクションと称されることがある。それは自動的に新しいブロック１５１ｎの一部を形成する。この報酬は、マイナー１０４Ｍがプルーフオブワーク競争に参加するためのインセンティブを与える。多くの場合、通常の（非生成）トランザクション１５２はまた、そのトランザクションが含まれたブロック１５１ｎを作成した勝利マイナー１０４Ｍにさらに報酬を与えるために、その出力の１つにおいて追加のトランザクション手数料を指定する。

【００２８】

マイニングに関与する計算リソースに起因して、典型的には、マイナーノード１０４Ｍの少なくとも各々は、１つまたは複数の物理サーバユニットを含むサーバの形態をとるか、またはデータセンタ全体の形態をとる。各フォワーディングノード１０４Ｍおよび／またはストレージノード１０４Ｓもまた、サーバまたはデータセンタの形態をとり得る。しかしながら、原則として、任意の所与のノード１０４は、一緒にネットワーク化されたユーザ端末またはユーザ端末のグループの形態をとることができる。

【００２９】

各ノード１０４のメモリは、そのそれぞれの１つまたは複数の役割を実行し、ノードプロトコルにしたがってトランザクション１５２を処理するために、ノード１０４の処理装置上で実行ように構成されたソフトウェアを記憶する。本明細書においてノード１０４に帰する任意のアクションは、それぞれのコンピュータ機器の処理装置上で実行されるソフトウェアによって実行され得ることが理解されよう。また、本明細書で使用される「ブロックチェーン」という用語は、一般に、技術の種類を指す一般的な用語であり、任意の特定の専有のブロックチェーン、プロトコルまたはサービスに限定されない。

【００３０】

消費ユーザの役割を果たす複数の当事者１０３の各々のコンピュータ機器１０２もネットワーク１０１に接続されている。これらは、トランザクションにおいて支払人および受取人として機能するが、他の当事者に代わってトランザクションのマイニングまたは伝搬に必ずしも参加するわけではない。それらは、マイニングプロトコルを必ずしも実行するわけではない。２人の当事者１０３およびそれらのそれぞれの機器１０２、すなわち、第１の当事者１０３ａおよびそのそれぞれのコンピュータ機器１０２ａ、ならびに第２の当事者１０３ｂおよびそのそれぞれのコンピュータ機器１０２ｂは、例示の目的で示されている。はるかに多くのそのような当事者１０３およびそれらのそれぞれのコンピュータ機器１０２が存在し、システムに参加し得るが、便宜上、それらは図示されていないことが理解されよう。各当事者１０３は、個人または組織であり得る。純粋に例示として、第１の当事者１０３ａは、本明細書ではアリスと称され、第２の当事者１０３ｂはボブと称されるが、これは限定的なものではなく、本明細書におけるアリスまたはボブへのいかなる言及も、それぞれ「第１の当事者」および「第２の当事者」と置き換えられ得ることが理

解されよう。

【００３１】

各当事者１０３のコンピュータ機器１０２は、１つまたは複数のプロセッサ、例えば、１つまたは複数のＣＰＵ、ＧＰＵ、他のアクセラレータプロセッサ、特定用途向けプロセッサ、および／またはＦＰＧＡを含むそれぞれの処理装置を含む。各当事者１０３のコンピュータ機器１０２は、メモリ、すなわち、１つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備える。このメモリは、１つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ＳＳＤ、フラッシュメモリもしくはＥＥＰＲＯＭなどの電子媒体、および／または光ディスクドライブなどの光学媒体を使用する１つまたは複数のメモリユニットを備え得る。各当事者１０３のコンピュータ機器１０２上のメモリは、処理装置上で実行するように構成された少なくとも１つのクライアントアプリケーション１０５のそれぞれのインスタンスを含むソフトウェアを記憶する。本明細書において所与の当事者１０３に帰する任意のアクションは、それぞれのコンピュータ機器１０２の処理装置上で実行されるソフトウェアを使用して実行され得ることが理解されよう。各当事者１０３のコンピュータ機器１０２は、少なくとも１つのユーザ端末、例えば、デスクトップもしくはラップトップコンピュータ、タブレット、スマートフォン、またはスマートウォッチなどのウェアラブルデバイスを備える。所与の当事者１０３のコンピュータ機器１０２はまた、ユーザ端末を介してアクセスされるクラウドコンピューティングリソースなどの１つまたは複数の他のネットワーク化されたリソースを備え得る。

【００３２】

クライアントアプリケーションまたはソフトウェア１０５は、最初に、適切な１つまたは複数のコンピュータ可読記憶媒体上で任意の所与の当事者１０３のコンピュータ機器１０２に提供され得、例えば、サーバからダウンロードされ得るか、またはリムーバブルＳＳＤ、フラッシュメモリキー、リムーバブルＥＥＰＲＯＭ、リムーバブル磁気ディスクドライブ、磁気フロッピーディスクもしくはテープ、ＣＤもしくはＤＶＤ　ＲＯＭなどの光ディスク、またはリムーバブル光学ドライブなどのリムーバブル記憶デバイス上で提供され得る。

【００３３】

クライアントアプリケーション１０５は、少なくとも「ウォレット」機能を備える。これは２つの主要な機能を有する。これらのうちの１つは、それぞれのユーザ当事者１０３が、ノード１０４のネットワーク全体に伝搬され、それによってブロックチェーン１５０に含まれることとなるトランザクション１５２を作成し、署名し、送信することを可能にすることである。もう１つは、それぞれの当事者に、その当事者が現在所有しているデジタル資産の額を報告することである。出力ベースのシステムでは、この第２の機能は、当該当事者に属するブロックチェーン１５０全体に散在している様々なトランザクション１５２の出力において定義された額を照合することを含む。

【００３４】

各コンピュータ機器１０２上のクライアントアプリケーション１０５のインスタンスは、Ｐ２Ｐネットワーク１０６のフォワーディングノード１０４Ｆのうちの少なくとも１つに動作可能に結合される。これにより、クライアント１０５のウォレット機能はトランザクション１５２をネットワーク１０６に送信することができる。クライアント１０５はまた、それぞれの当事者１０３が受信者である任意のトランザクションについてブロックチェーン１５０にクエリを行うために、ストレージノード１０４のうちの１つ、いくつか、またはすべてにコンタクトすることができる（または、実施形態では、ブロックチェーン１５０は、部分的にその公開可視性を通じてトランザクションにおける信頼を提供する公開施設であるので、実際にブロックチェーン１５０における他の当事者のトランザクションを検査する）。各コンピュータ機器１０２上のウォレット機能は、トランザクションプロトコルにしたがってトランザクション１５２を定式化し、送信するように構成される。各ノード１０４は、ノードプロトコルにしたがってトランザクション１５２を妥当性確認するように構成されたソフトウェアを実行し、フォワーディングノード１０４Ｆの場合に

は、トランザクション１５２をネットワーク１０６全体に伝搬させるためにそれらをフォワードするように構成される。トランザクションプロトコルおよびノードプロトコルは互いに対応し、所与のトランザクションプロトコルは所与のノードプロトコルと共に進行し、一緒に所与のトランザクションモデルを実装する。ブロックチェーン１５０内のすべてのトランザクション１５２に対して同じトランザクションプロトコルが使用される（ただし、トランザクションプロトコルは、その中のトランザクションの異なるサブタイプを可能にし得る）。同じノードプロトコルが、ネットワーク１０６内のすべてのノード１０４によって使用される（ただし、これは、トランザクションの異なるサブタイプを、そのサブタイプに対して定義された規則にしたがって異なって処理し、また、異なるノードが異なる役割を担い、したがってプロトコルの異なる対応する態様を実装することができる）。

【００３５】

述べたように、ブロックチェーン１５０は、ブロック１５１のチェーンを含み、各ブロック１５１は、前述したようなプルーフオブワークプロセスによって作成された１つまたは複数のトランザクション１５２のセットを含む。各ブロック１５１はまた、ブロック１５１へのシーケンシャル順序を定義するために、チェーン内の前に作成されたブロック１５１を指し示すブロックポインタ１５５を含む。ブロックチェーン１５０は、プルーフオブワークプロセスによって新しいブロックに含まれるのを待っている有効なトランザクション１５４のプールも含む。各トランザクション１５２（生成トランザクション以外）は、トランザクションのシーケンスへの順序を定義するように、前のトランザクションへ戻るポインタを含む（注意：トランザクション１５２のシーケンスは分岐することが可能である）。ブロック１５１のチェーンは、チェーン内の最初のブロックであった発生ブロック（Ｇｂ）１５３までずっと戻る。チェーン１５０内の早期にある１つまたは複数の元のトランザクション１５２は、先行するトランザクションではなく発生ブロック１５３を指し示していた。

【００３６】

所与の当事者１０３、例えばアリスが、ブロックチェーン１５０に含まれるべき新しいトランザクション１５２ｊを送信することを望むとき、アリスは、関連トランザクションプロトコルにしたがって（アリスのクライアントアプリケーション１０５内のウォレット機能を使用して）新しいトランザクションを定式化する。次いで、アリスは、クライアントアプリケーション１０５から、アリスが接続されている１つまたは複数のフォワーディングノード１０４Ｆのうちの１つにトランザクション１５２を送信する。例えば、これは、アリスのコンピュータ１０２に最も近いまたは最良に接続されたフォワーディングノード１０４Ｆであり得る。任意の所与のノード１０４が新しいトランザクション１５２ｊを受信すると、それはノードプロトコルおよびそのそれぞれの役割にしたがってそれを処理する。これは、新たに受信されたトランザクション１５２ｊが「有効」であるための特定の条件を満たすか否かを最初にチェックすることを含み、その例については、以下でより詳細に説明する。いくつかのトランザクションプロトコルでは、妥当性確認のための条件は、トランザクション１５２に含まれるスクリプトによってトランザクションごとに構成可能であり得る。代替的に、条件は、単にノードプロトコルの組込み特徴であってもよく、またはスクリプトとノードプロトコルとの組合せによって定義されてもよい。

【００３７】

新たに受信されたトランザクション１５２ｊが有効であると見なされるためのテストにパスすることを条件として（すなわち、それが「妥当性確認される」ことを条件として）、トランザクション１５２ｊを受信する任意のストレージノード１０４Ｓは、そのノード１０４Ｓにおいて維持されるブロックチェーン１５０のコピー内のプール１５４に新たな妥当性確認済みトランザクション１５２を追加する。さらに、トランザクション１５２ｊを受信する任意のフォワーディングノード１０４Ｆは、妥当性確認済みトランザクション１５２をＰ２Ｐネットワーク１０６内の１つまたは複数の他のノード１０４に前方に伝搬する。各フォワーディングノード１０４Ｆは同じプロトコルを適用するので、トランザクション１５２ｊが有効であると仮定すると、これは、それがすぐにＰ２Ｐネットワーク１

０６全体にわたって伝搬されることを意味する。
【００３８】
１つまたは複数のストレージノード１０４において維持されるブロックチェーン１５０のコピー内のプール１５４に認められると、マイナーノード１０４Ｍは、新しいトランザクション１５２を含むプール１５４の最新バージョンに対してプルーフオブワークパズルを解こうと競い始める（他のマイナー１０４Ｍは、プール１５４の古いビューに基づいてパズルを解こうと試みている可能性があるが、誰が最初に到達しても、次の新しいブロック１５１がどこで終わり新しいプール１５４がどこで開始するかを定義することとなり、最終的には、誰かが、アリスのトランザクション１５２ｊを含むプール１５４の一部についてパズルを解く）。新しいトランザクション１５２ｊを含むプール１５４に対してプルーフオブワークが行われると、それは不変的にブロックチェーン１５０内のブロック１５１のうちの１つの一部となる。各トランザクション１５２は、前のトランザクションへ戻るポインタを含むので、トランザクションの順序も不変的に記録される。
【００３９】
ＵＴＸＯベースのモデル
図２は、例示的なトランザクションプロトコルを示す。これは、ＵＴＸＯベースのプロトコルの一例である。トランザクション１５２（「Ｔｘ」と略記される）は、ブロックチェーン１５０の基本的なデータ構造である（各ブロック１５１は１つまたは複数のトランザクション１５２を含む）。以下では、出力ベースまたは「ＵＴＸＯ」ベースのプロトコルを参照して説明する。しかしながら、これはすべての可能な実施形態に限定されない。
【００４０】
ＵＴＸＯベースのモデルでは、各トランザクション（「Ｔｘ」）１５２は、１つまたは複数の入力２０２および１つまたは複数の出力２０３を含むデータ構造を含む。各出力２０３は、未使用トランザクション出力（ＵＴＸＯ）を含み得、これは、（ＵＴＸＯがまだ償還されていない場合）別の新しいトランザクションの入力２０２のソースとして使用され得る。ＵＴＸＯは、デジタル資産（価値の蓄蔵）の額を指定する。それはまた、他の情報の中でも、元となるトランザクションのトランザクションＩＤを含み得る。トランザクションデータ構造は、入力フィールド（複数可）２０２および出力フィールド（複数可）２０３のサイズを示すインジケータを含み得るヘッダ２０１も含み得る。ヘッダ２０１はまた、トランザクションのＩＤを含み得る。実施形態では、トランザクションＩＤは、（トランザクションＩＤ自体を除く）トランザクションデータのハッシュであり、マイナー１０４Ｍにサブミットされる生のトランザクション１５２のヘッダ２０１に格納される。
【００４１】
図２の各出力はＵＴＸＯとして示されているが、トランザクションは、追加的にまたは代替的に、１つまたは複数の使用不可能なトランザクション出力を含み得ることに留意されたい。
【００４２】
アリス１０３ａが、当該デジタル資産の額をボブ１０３ｂに転送するトランザクション１５２ｊを作成することを望むとする。図２では、アリスの新しいトランザクション１５２ｊは「Tx_1」とラベル付けされている。これは、シーケンス内の先行するトランザクション１５２ｉの出力２０３においてアリスにロックされたデジタル資産の額を取り、これのうちの少なくとも一部をボブに転送する。先行するトランザクション１５２ｉは、図２では「Tx_0」とラベル付けされている。Tx_0およびTx_1は、単なる任意のラベルである。それらは、Tx_0がブロックチェーン１５１内の最初のトランザクションであることも、Tx_1がプール１５４内のすぐ次のトランザクションであることも必ずしも意味するものではない。Tx_1は、アリスにロックされた未使用の出力２０３を依然として有する任意の先行する（すなわち先の）トランザクションを指し示すことができる。
【００４３】
先行するトランザクションTx_0は、アリスが新しいトランザクションTx_1を作成した時点では、または少なくともアリスがそれをネットワーク１０６に送信する時点までに

は、すでに妥当性確認されブロックチェーン１５０に含まれている可能性がある。それは、その時点でブロック１５１のうちの１つにすでに含まれていてもよいし、プール１５４で依然として待機していてもよく、その場合には、すぐに新しいブロック１５１に含まれることになる。代替的に、Tx_0およびTx_1を作成してネットワーク１０２に一緒に送信することができるか、またはノードプロトコルが「オーファン」トランザクションのバッファリングを可能にする場合には、Tx_0をTx_1の後に送信することさえもできる。トランザクションのシーケンスの文脈において本明細書で使用される「先行する」および「後続の」という用語は、トランザクション内で指定されているトランザクションポインタ（どのトランザクションがどの他のトランザクションを指し示すかなど）によって定義されるシーケンス内のトランザクションの順序を指す。それらは、同様に、「先行するもの」および「後続するもの」、または「先の」および「後の」、「親」および「子」などと置き換えられ得る。これは、それらの作成、ネットワーク１０６への送信、または任意の所与のノード１０４への到着の順序を必ずしも意味するものではない。それにもかかわらず、先行するトランザクション（先のトランザクションまたは「親」）を指し示す後続するトランザクション（後のトランザクションまたは「子」）は、親トランザクションが妥当性確認されるまでおよび妥当性確認されない限り妥当性確認されない。その親より前にノード１０４に到着する子は、オーファンとみなされる。それは、ノードプロトコルおよび／またはマイナー挙動に応じて、親を待つために特定の時間バッファされるかまたは破棄され得る。

【００４４】

先行するトランザクションTx_0の１つまたは複数の出力２０３のうちの１つは、本明細書では$UTXO_0$とラベル付けされた特定のＵＴＸＯを含む。各ＵＴＸＯは、ＵＴＸＯによって表されるデジタル資産の額を指定する値と、ロックスクリプトとを含み、ロックスクリプトは、後続のトランザクションが妥当性確認され、したがってＵＴＸＯが正常に償還されるために、後続のトランザクションの入力２０２内のロック解除スクリプトが満たさなければならない条件を定義する。典型的には、ロックスクリプトは、その額を特定の当事者（それが含まれるトランザクションの受益者）にロックする。すなわち、ロックスクリプトは、典型的には、後続のトランザクションの入力内のロック解除スクリプトに、先行するトランザクションがロックされる当事者の暗号署名が含まれるという条件を含むロック解除条件を定義する。

【００４５】

ロックスクリプト（通称ｓｃｒｉｐｔＰｕｂＫｅｙ）は、ノードプロトコルによって認識されるドメイン固有言語で書かれたコードの一部分である。そのような言語の特定の例は、「スクリプト」（大文字Ｓ）と呼ばれる。ロックスクリプトは、トランザクション出力２０３を使用するためにどの情報が必要とされるか、例えばアリスの署名の要件を指定する。ロック解除スクリプトはトランザクションの出力に現れる。ロック解除スクリプト（通称ｓｃｒｉｐｔＳｉｇ）は、ロックスクリプト基準を満たすのに必要な情報を提供するドメイン固有言語で書かれたコードの一部分である。例えば、ボブの署名を含み得る。ロック解除スクリプトは、トランザクションの入力２０２に現れる。

【００４６】

つまり、図示の例では、Tx_0の出力２０３内の$UTXO_0$は、$UTXO_0$が償還されるために（厳密には、$UTXO_0$を償還しようとする後続のトランザクションが有効となるために）アリスの署名$Sig\ P_A$を必要とするロックスクリプト［$Checksig\ P_A$］を含む。［$Checksig\ P_A$］は、アリスの公開鍵－秘密鍵ペアからの公開鍵P_Aを含む。Tx_1の入力２０２は、（例えば、実施形態ではトランザクションTx_0全体のハッシュであるそのトランザクションＩＤ、$TxID_0$によって）Tx_1を指し示すポインタを含む。Tx_1の入力２０２は、Tx_0の任意の他の可能な出力の中から、$UTXO_0$を識別するために、Tx_0内のそれを識別するインデックスを含む。Tx_1の入力２０２は、アリスが鍵ペアからのアリスの秘密鍵をデータの所定の部分（暗号では「メッセージ」と呼ばれることもある）に適用することによって作成された、アリスの暗号署名を含む

ロック解除スクリプト＜Ｓｉｇ　P_A＞をさらに含む。有効な署名を提供するためにどのデータ（または「メッセージ」）がアリスによって署名される必要があるかは、ロックスクリプトによって、またはノードプロトコルによって、またはこれらの組合せによって定義され得る。

【００４７】

新しいトランザクションTx_1がノード１０４に到着すると、ノードはノードプロトコルを適用する。これは、ロックスクリプトおよびロック解除スクリプトを一緒に実行して、ロック解除スクリプトがロックスクリプトで定義されている条件（この条件は１つまたは複数の基準を含み得る）を満たすかどうかをチェックすることを含む。実施形態では、これは２つのスクリプトを連結することを含む。

＜Ｓｉｇ　P_A＞＜P_A＞｜｜［Ｃｈｅｃｋｓｉｇ　P_A］

ここで、「｜｜」は連結を表し、「＜…＞」はデータをスタックに置くことを意味し、「［…］」はロック解除スクリプト（この例ではスタックベースの言語）で構成される関数である。同等に、スクリプトは、スクリプトを連結するのではなく、共通スタックを用いて次々に実行され得る。いずれにしても、一緒に実行されるとき、スクリプトは、Tx_0の出力内のロックスクリプトに含まれるようなアリスの公開鍵P_Aを使用して、Tx_1の入力内のロックスクリプトが、データの予期される部分に署名したアリスの署名を含むことを認証する。データの予想される部分自体（「メッセージ」）はまた、この認証を実行するためにTx_0命令に含まれる必要がある。実施形態では、署名されたデータは、Tx_0の全体を含む（つまり、平文のデータの署名された部分を指定する別個の要素は、すでに本質的に存在するので、含まれる必要はない）。

【００４８】

公開－秘密暗号法による認証の詳細は、当業者によく知られている。基本的に、アリスが自分の秘密鍵でメッセージを暗号化することによってメッセージに署名した場合、アリスの公開鍵および平文のメッセージ（暗号化されていないメッセージ）が与えられると、ノード１０４などの別のエンティティは、メッセージの暗号化バージョンがアリスによって署名されたものに違いないことを認証することができる。署名は、典型的には、メッセージをハッシュし、ハッシュに署名し、これを署名としてメッセージの平文バージョンにタグ付けすることを含み、これにより、公開鍵の任意の保持者が署名を認証することができる。

【００４９】

Tx_1内のロック解除スクリプトが、Tx_0のロックスクリプト内で指定されている１つまたは複数の条件を満たす場合（つまり、図示の例では、アリスの署名がTx_1内で提供され、認証された場合）、ノード１０４は、Tx_1が有効であると見なす。それがマイニングノード１０４Ｍである場合、これは、ワークオブプルーフを待つトランザクション１５４のプールにそれを追加することを意味する。それがフォワーディングノード１０４Ｆである場合、トランザクションTx_1をネットワーク１０６内の１つまたは複数の他のノード１０４にフォワードして、トランザクションTx_1がネットワーク全体に伝搬されるようにする。Tx_1が妥当性確認されてブロックチェーン１５０に含まれると、これは、Tx_0からの$UTXO_0$を使用済みと定義する。Tx_1は、未使用のトランザクション出力２０３を使用する場合にのみ有効であり得ることに留意されたい。別のトランザクション１５２によってすでに使用された出力を使用しようとする場合、Tx_1は、他のすべての条件が満たされたとしても無効になる。したがって、ノード１０４はまた、先行するトランザクションTx_0内の参照されたＵＴＸＯがすでに使用済みである（別の有効なトランザクションへの有効な入力をすでに形成している）かどうかをチェックする必要がある。これは、ブロックチェーン１５０がトランザクション１５２に定義された順序を課すことが重要である１つの理由である。実際には、所与のノード１０４は、どのトランザクション１５２内のどのＵＴＸＯ２０３が使用されたかをマーキングする別個のデータベースを維持し得るが、最終的には、ＵＴＸＯが使用されたかどうかを定義するものは、ブロックチェーン１５０内の別の有効なトランザクションへの有効な入力をすでに形成している

かどうかである。

【００５０】

ＵＴＸＯベースのトランザクションモデルでは、所与のＵＴＸＯが全体として使用される必要があることに留意されたい。ＵＴＸＯにおいて使用済みとして定義された額の一部を「後に残す」ことはできず、別の一部が使用される。しかしながら、次のトランザクションの複数の出力間でＵＴＸＯからの額を分割することはできる。例えば、$Ｔｘ_0$内の$ＵＴＸＯ_0$において定義された額は、$Ｔｘ_1$内の複数のＵＴＸＯ間で分割され得る。したがって、アリスが、$ＵＴＸＯ_0$において定義された額のすべてをボブに与えたくない場合、アリスは、リマインダを使用して、$Ｔｘ_1$の第２の出力において自分に残りを与えるか、または別の当事者に支払うことができる。

【００５１】

実際には、今日では、生成トランザクションの報酬だけでは、典型的には、マイニングを動機付けるのに十分ではないので、アリスは、通常、勝利マイナーに対する手数料を含む必要もある。アリスがマイナーに対する手数料を含めない場合、$Ｔｘ_0$は、マイナーノード１０４Ｍによって拒否される可能性が高く、したがって、技術的に有効であっても、それは依然として伝搬されず、ブロックチェーン１５０に含まれない（マイナープロトコルは、マイナー１０４Ｍが望まない場合にトランザクション１５２を受け入れることを強制しない）。いくつかのプロトコルでは、マイニング手数料は、それ自体の別個の出力２０３を必要としない（すなわち、別個のＵＴＸＯを必要としない）。代わりに、所与のトランザクション１５２の入力（複数可）２０２によって指し示される総額と出力（複数可）２０３で指定されている総額との間の差が自動的に勝利マイナー１０４に与えられる。例えば、$ＵＴＸＯ_0$へのポインタが$Ｔｘ_1$への唯一の入力であり、$Ｔｘ_1$は唯一の出力$ＵＴＸＯ_1$を有するとする。$ＵＴＸＯ_0$で指定されているデジタル資産の額が$ＵＴＸＯ_1$で指定されている額より大きい場合、その差が自動的に勝利マイナー１０４Ｍに贈られる。しかしながら、代替的にまたは追加的に、マイナー手数料がトランザクション１５２のＵＴＸＯ２０３のうちのそれ自体の１つにおいて明示的に指定され得ることは必ずしも除外されるものではなない。

【００５２】

所与のトランザクション１５２のすべての出力２０３で指定されている総額が、そのすべての入力２０２によって指し示された総額よりも大きい場合、これは、ほとんどのトランザクションモデルにおいて無効性の別の根拠であることにも留意されたい。したがって、そのようなトランザクションは、ブロック１５１に伝搬もマイニングもされない。

【００５３】

アリスおよびボブのデジタル資産は、ブロックチェーン１５０内のどこにでもある任意のトランザクション１５２においてそれらにロックされた未使用ＵＴＸＯから構成される。したがって、典型的には、所与の当事者１０３の資産は、ブロックチェーン１５０全体にわたる様々なトランザクション１５２のＵＴＸＯ全体に散在している。ブロックチェーン１５０内のどこにも、所与の当事者１０３の総残高を定義する数字は記憶されない。クライアントアプリケーション１０５におけるウォレット機能の役割は、それぞれの当事者にロックされ、別の前方のトランザクションでまだ使用されていない様々なＵＴＸＯすべての値を一緒に照合することである。これは、ストレージノード１０４Ｓのいずれか、例えば、それぞれの当事者のコンピュータ機器１０２に最も近いまたは最良に接続されたストレージノード１０４Ｓに記憶されたブロックチェーン１５０のコピーにクエリを行うことによって行うことができる。

【００５４】

スクリプトコードは、しばしば、概略的に表される（すなわち、正確な言語ではない）ことに留意されたい。例えば、［Ｃｈｅｃｋｓｉｇ $Ｐ_A$］と書いて［Ｃｈｅｃｋｓｉｇ $Ｐ_A$］＝ＯＰ＿ＤＵＰ ＯＰ＿ＨＡＳＨ１６０＜Ｈ（Ｐａ）＞ＯＰ＿ＥＱＵＡＬＶＥＲＩＦＹ ＯＰ＿ＣＨＥＣＫＳＩＧを意味し得る。「ＯＰ…」は、スクリプト言語の特定のオペコードを指す。ＯＰ＿ＣＨＥＣＫＳＩＧ（「Ｃｈｅｃｋｓｉｇ」とも呼ばれる）は、２

つの入力（署名および公開鍵）を取り、楕円曲線デジタル署名アルゴリズム（ＥＣＤＳＡ）を使用して署名の有効性を検証するスクリプトオペコードである。実行時に、署名（「ｓｉｇ」）の存在（ｏｃｃｕｒｒｅｎｃｅ）はスクリプトから除去されるが、ハッシュパズルなどの追加要件は、「ｓｉｇ」入力によって検証されたトランザクションに残る。別の例として、ＯＰ＿ＲＥＴＵＲＮは、トランザクション内にメタデータを記憶することができ、それによってメタデータをブロックチェーン１５０に不変に記録することができる、トランザクションの使用不可能な出力を作成するためのスクリプト言語のオペコードである。例えば、メタデータは、ブロックチェーンに格納することが望まれる文書を含み得る。

【００５５】

署名P_Aはデジタル署名である。実施形態において、これは、楕円曲線ｓｅｃｐ２５６ｋ１を使用するＥＣＤＳＡに基づく。デジタル署名は、特定のデータの一部分に署名する。実施形態では、所与のトランザクションについて、署名は、トランザクション入力の一部、およびトランザクション出力の全部または一部に署名する。署名された出力の特定の部分は、ＳＩＧＨＡＳＨフラグに依存する。ＳＩＧＨＡＳＨフラグは、どの出力が署名されるかを選択するために署名の最後に含まれる４バイトコードである（したがって、署名時に固定される）。

【００５６】

ロックスクリプトは、それぞれのトランザクションがロックされる当事者の公開鍵を含むという事実を指して、「ｓｃｒｉｐｔＰｕｂＫｅｙ」と呼ばれることがある。ロック解除スクリプトは、それが対応する署名を供給するという事実を指して「ｓｃｒｉｐｔＳｉｇ」と呼ばれることがある。しかしながら、より一般的には、ＵＴＸＯが償還されるための条件が署名を認証することを含むことは、ブロックチェーン１５０のすべてのアプリケーションにおいて必須ではない。より一般的には、スクリプト言語を使用して、任意の１つまたは複数の条件を定義することができる。したがって、より一般的な用語「ロックスクリプト」および「ロック解除スクリプト」が好まれ得る。

【００５７】

任意選択のサイドチャネル

図３は、ブロックチェーン１５０を実装するためのさらなるシステム１００を示す。システム１００は、追加の通信機能が含まれることを除いて、図１に関連して説明したものと実質的に同じである。アリスおよびボブのそれぞれのコンピュータ機器１０２ａ、１２０ｂ上のクライアントアプリケーションは、それぞれ、追加の通信機能を含む。すなわち、これは、（いずれかの当事者または第三者の指示で）アリス１０３ａがボブ１０３ｂとの別個のサイドチャネル３０１を確立することを可能にする。サイドチャネル３０１は、Ｐ２Ｐネットワークとは別でのデータの交換を可能にする。このような通信は、「オフチェーン」と称されることがある。例えば、これは、当事者の一方がトランザクションをネットワーク１０６にブロードキャストすることを選択するまで、トランザクションが（まだ）ネットワークＰ２Ｐ　１０６上に公開されたりチェーン１５０上に進んだりすることなくことなく、アリスとボブとの間でトランザクション１５２を交換するために使用され得る。代替的にまたは追加的に、サイドチャネル３０１は、鍵、交渉された額または条件、データコンテンツなどの任意の他のトランザクション関連データを交換するために使用され得る。

【００５８】

サイドチャネル３０１は、Ｐ２Ｐオーバーレイネットワーク１０６と同じパケット交換ネットワーク１０１を介して確立され得る。代替的にまたは追加的に、サイドチャネル３０１は、モバイルセルラーネットワークなどの異なるネットワーク、またはローカルワイヤレスネットワークなどのローカルエリアネットワーク、またはさらにはアリスのデバイス１０２１とボブのデバイス１０２ｂとの間の直接の有線またはワイヤレスリンクを介して確立され得る。一般に、本明細書のどこかで参照されるサイドチャネル３０１は、「オフチェーン」すなわちＰ２Ｐオーバーレイネットワーク１０６とは別でデータを交換する

ための１つまたは複数のネットワーキング技術または通信媒体を介した任意の１つまたは複数のリンクを含み得る。２つ以上のリンクが使用される場合、全体としてのオフチェーンリンクの束または集合は、サイドチャネル３０１と称され得る。したがって、アリスおよびボブがサイドチャネル３０１上で情報またはデータの特定の部分などを交換すると言われている場合、これは、これらのデータの部分のすべてが全く同じリンクまたは同じタイプのネットワーク上で送信されなければならないことを必ずしも意味するものではないことに留意されたい。

【００５９】

ノードソフトウェア

図４は、ＵＴＸＯまたは出力ベースのモデルの例において、Ｐ２Ｐネットワーク１０６の各ノード１０４上で実行されるノードソフトウェア４００の例を示す。ノードソフトウェア４００は、プロトコルエンジン４０１と、スクリプトエンジン４０２と、スタック４０３と、アプリケーションレベル決定エンジン４０４と、１つまたは複数のブロックチェーン関連機能モジュール４０５のセットとを含む。任意の所与のノード１０４において、これらは、マイニングモジュール４０５Ｍ、フォワーディングモジュール４０５Ｆ、および格納モジュール４０５Ｓ（ノードの１つまたは複数の役割に応じて）のうちのいずれか１つ、２つ、または３つすべてを含み得る。プロトコルエンジン４０１は、トランザクション１５２の異なるフィールドを認識し、ノードプロトコルにしたがってそれらを処理するように構成される。別の先行するトランザクション１５２ｍ－１（Tx_{m-1}）の出力（例えば、ＵＴＸＯ）を指し示す入力を有するトランザクション１５２ｍ（Tx_m）が受信されると、プロトコルエンジン４０１は、Tx_m内のロック解除スクリプトを識別し、それをスクリプトエンジン４０２に渡す。プロトコルエンジン４０１はまた、Tx_mの入力内のポインタに基づいて、Tx_{m-1}を識別し、取り出す。それは、Tx_{m-1}がまだブロックチェーン１５０上にない場合には保留中のトランザクションのそれぞれのノード自体のプール１５４から、またはTx_{m-1}がすでにブロックチェーン１５０上にある場合にはそれぞれのノードまたは別のノード１０４に格納されたブロックチェーン１５０内のブロック１５１のコピーからTx_{m-1}を取り出し得る。いずれにしても、スクリプトエンジン４０１は、Tx_{m-1}の指し示された出力におけるロックスクリプトを識別し、これをスクリプトエンジン４０２に渡す。

【００６０】

したがって、スクリプトエンジン４０２は、Tx_{m-1}のロックスクリプトと、Tx_mの対応する入力からのロック解除スクリプトとを有する。例えば、Tx_1およびTx_2が図４に示されているが、同じことが、Tx_0とTx_1などの任意のペアのトランザクションにも当てはまる。スクリプトエンジン４０２は、前述したように２つのスクリプトを一緒に実行し、これは、使用されているスタックベースのスクリプト言語（例えば、Ｓｃｒｉｐｔ）にしたがって、スタック４０３上にデータを配置し、そこからデータを取り出すことを含む。

【００６１】

スクリプトを一緒に実行することによって、スクリプトエンジン４０２は、ロック解除スクリプトがロックスクリプトにおいて定義された１つまたは複数の基準を満たすかどうか、すなわち、ロックスクリプトが含まれる出力を「ロック解除」するかどうかを決定する。スクリプトエンジン４０２は、この決定の結果をプロトコルエンジン４０１に返す。スクリプトエンジン４０２は、ロック解除スクリプトが対応するロックスクリプトで指定されている１つまたは複数の基準を満たすと決定した場合、結果「真」を返す。そうでなければ、結果「偽」を返す。

【００６２】

出力ベースのモデルでは、スクリプトエンジン４０２からの結果「真」は、トランザクションの有効性の条件のうちの１つである。典型的には、Tx_mの出力（複数可）で指定されているデジタル資産の総額が入力（複数可）によって指し示された総額を超えないこと、およびTx_{m-1}の指し示された出力が別の有効なトランザクションによってまだ使用

されていないことなど、同様に満たされなければならないプロトコルエンジン４０１によって評価される１つまたは複数のさらなるプロトコルレベル条件も存在する。プロトコルエンジン４０１は、スクリプトエンジン４０２からの結果を１つまたは複数のプロトコルレベル条件と共に評価し、それらがすべて真である場合にのみ、トランザクションＴｘｍを妥当性確認する。プロトコルエンジン４０１は、トランザクションが有効であるかどうかの指示をアプリケーションレベル決定エンジン４０４に出力する。Ｔｘｍが実際に妥当性確認されるという条件でのみ、決定エンジン４０４は、Ｔｘｍに関してそれぞれのブロックチェーン関連機能を実行するために、マイニングモジュール４０５Ｍおよびフォワーディングモジュール４０５Ｆの一方または両方を制御することを選択し得る。これは、マイニングモジュール４０５Ｍが、ブロック１５１にマイニングするためにＴｘｍをノードのそれぞれのプール１５４に追加すること、および／またはフォワーディングモジュール４０５ＦがＴｘｍをＰ２Ｐネットワーク１０６内の別のノード１０４にフォワードすることを含み得る。しかしながら、実施形態では、決定エンジン４０４は無効なトランザクションをフォワードまたはマイニングすることを選択しないが、これは、逆に、単に有効であるという理由で有効なトランザクションのマイニングまたはフォワードをトリガする義務があることを必ずしも意味するものではないことに留意されたい。任意選択で、実施形態では、決定エンジン４０４は、一方または両方の機能をトリガする前に、１つまたは複数の追加の条件を適用することができる。例えば、ノードがマイニングノード１０４Ｍである場合、決定エンジンは、トランザクションが有効であり、かつ十分なマイニング手数料を残しているという条件でのみトランザクションをマイニングすることを選択し得る。

【００６３】

本明細書における「真」および「偽」という用語は、単一の２進数（ビット）のみの形態で表される結果を返すことに必ずしも限定されないが、それは確かに１つの可能な実装形態であることにも留意されたい。より一般的には、「真」は、成功または肯定的な結果を示す任意の状態を指すことができ、「偽」は、不成功または非肯定的な結果を示す任意の状態を指すことができる。例えば、アカウントベースのモデル（図４には図示せず）では、「真」の結果は、ノード１０４による署名の暗黙的な（プロトコルレベルの）妥当性確認と、スマートコントラクトの追加の肯定的な出力との組合せによって示され得る（個々の結果の両方が真である場合、全体の結果が真を示すとみなされる）。

【００６４】

モノのインターネット

ＩｏＴは、日常の物理的なデバイスおよびオブジェクトへのインターネットの拡張である。計算処理能力およびインターネット接続性が組み込まれると、デバイスは、互いに通信および対話することができ、リモートでの監視および制御が可能になる。時間の経過とともに、ＩｏＴの定義は、機械学習、リアルタイム分析、および複数の技術の収束に起因して進化してきたが、ワイヤレスセンサネットワークおよび／または制御システムをサポートすることができるデバイスのシステムがＩｏＴを実現する可能性が高いことが一般的に認められている。

【００６５】

ＩｏＴシステムは、いくつかの課題に直面している。例えば、そのようなシステムのスケーラビリティおよびコストは、ＩｏＴシステムがそれらの最大の可能性を発揮するのを妨げ得る。集中型方式で接続および制御されるとき、ＩｏＴデバイスは、データを送信し、制御コマンドを受信するためのバックエンドインフラストラクチャを必要とする。これらのバックエンドインフラストラクチャは、サードパーティのクラウドサービスまたはオンプレミスサーバファームのいずれかでホストされる。次いで、ＩｏＴソリューションのスケーラビリティは、ＩｏＴサービスプロバイダの運用コストが法外に高くなる可能性のあるバックエンドサーバおよびデータセンタのスケーラビリティによって決定される。結果として、多くの提案されたＩｏＴソリューションは、費用効果的ではなく、日常的なシナリオでの使用には不適切である。ネットワーク待ち時間などの性能測定も、ＩｏＴ普及率を決定する重要な要因となる。

【００６６】
ＩｏＴシステムが直面する別の課題は、自動化と制御との間のトレードオフである。ＩｏＴソリューションは、日常的な電子デバイスへのリモートアクセスおよび制御を可能にするように設計される。ほとんどのＩｏＴソリューションは、完全なユーザ制御と、デバイスと他のＩｏＴソリューションコンポーネントとの間の自動化された通信とを両立させる。デバイスまたはＩｏＴシステムのいずれかが正常に機能しない場合、オーバーライド機構などの安全対策が講じられる必要がある。
【００６７】
別の課題は、サイバー攻撃からの脅威である。インターネットを介したデバイスの自動制御を可能にすることによって、ユーザは、２つの形態の潜在的なセキュリティリスクに曝される。１つは、インターネットを介してＩｏＴデバイスメタデータを送信することによって発生するプライバシーリスクである。例えば、盗聴者が家電製品などのデバイスからデータへのアクセスを得た場合、デバイス使用のパターンは、犯罪者、例えば、人が家にいるときを予測するために窃盗者によって使用され得る。第２のリスクは、攻撃者または他の第三者がＩｏＴデバイスの制御を獲得する可能性である。重機または危険物を操作するために使用されるような性能重視の制御ソフトウェアの場合、攻撃が破滅的な結果をもたらす恐れがある。
【００６８】
ＩｏＴシステムは、集中型または分散型および／またはハイブリッドであるように設計され得る。集中型ソリューションは、ボトルネックに悩まされるが、ＩｏＴシステム内の特権コンポーネントによるより高速でより信頼性の高い制御を可能にすることができる。状態更新の分散報告は、ＩｏＴソリューションがよりスケーラブルになることを可能にする。エッジコンピューティングは、重要なアプリケーションのネットワーク待ち時間を低減し、ＩｏＴシステムのクラウドへの依存を低下させ、大量のＩｏＴデータのより良好な管理を提供するのに役立つことができる。分散処理の増加は、集中型アーキテクチャおよび分散型アーキテクチャの利点をより良好に利用するためのシステムアーキテクチャにおける機会を浮き彫りにする。階層制御構造内で集中型システムと分散型システムとを組み合わせたハイブリッドシステムは、ユーザの安全性およびユーザビリティの目的を高め得る。
【００６９】
ブロックチェーン技術は、以下の理由で、ＩｏＴの将来において主要な役割を果たす可能性を有する：ブロックチェーンは、支払いおよび制御を１つのネットワークに統合することを可能にする；既存のインフラストラクチャを使用して、デバイス状態変化に関するメッセージをピギーバックすることができる；およびネットワーク上のデータの分散制御がより高速なユーザ－デバイス対話を可能にする。ブロックチェーン技術と組み合わせることで、物理的世界で役割を果たす従来のＩｏＴデバイスは、メッセージを送ると同時に価値の交換を行うことができる。公開ブロックチェーンは、グローバル支払いネットワークとしてだけではなく、ＩｏＴに関連付けられたリスクのいくつかに自動的に対処する強力な暗号セキュリティがそのプロトコルに組み込まれている汎用商品台帳としても機能する。
【００７０】
図５は、本開示の実施形態を実装するための例示的なシステム５００を示す。例示的なシステム５００は、１つまたは複数のエンドデバイス（すなわち、コンピューティングデバイス）５０２と、１つまたは複数のブリッジノード５０３（すなわち、ブロックチェーンクライアントアプリケーションを実行し、したがってブロックチェーンネットワーク１０６と第１のネットワーク５０１との間のブリッジとして機能するコンピューティングデバイス）とから構成される第１のネットワーク５０１を備える。明確にするために、第１のネットワーク５０１は、ＩｏＴネットワーク、すなわち、インターネットによって相互接続されたコンピューティングデバイスのネットワークと称される。典型的には、エンドデバイス５０２およびブリッジノード５０３は、日常のデバイスに埋め込まれる。エンド

デバイス５０２は、様々な形態、例えば、ユーザデバイス（例えば、スマートＴＶ、スマートスピーカ、玩具、ウェアラブルなど）、スマートアプライアンス（例えば、冷蔵庫、洗濯機、オーブンなど）、メータまたはセンサ（例えば、スマートサーモスタット、スマートライティング、セキュリティセンサなど）のうちの１つをとり得る。同様に、ブリッジノード５０３はまた、エンドデバイスがとり得るのと同じ形態を含み得るがそれに限定されない様々な形態をとり得る。ノード５０３は、専用サーバ機器、基地局、アクセスポイント、ルータなどの形態もとり得る。いくつかの例では、各デバイスは、固定ネットワーク（例えば、ＩＰ）アドレスを有し得る。例えば、エンドデバイスのうちの１つ、いくつか、またはすべては、モバイルデバイスとは対照的に、固定デバイス（例えば、スマートライト、またはスマートセントラルヒーティングコントローラなど）であり得る。

【００７１】

ＩｏＴネットワークは、パケット交換ネットワーク１０１、典型的には、インターネットなどの広域インターネットワークである。パケット交換ネットワーク１０１のノード５０３およびデバイス５０２は、パケット交換ネットワーク１０１内にピアツーピア（Ｐ２Ｐ）オーバーレイネットワーク５０１を形成するように構成される。各ノード５０３は、それぞれのコンピュータ機器を備え、各コンピュータ機器が、１つまたは複数のプロセッサ、例えば、１つまたは複数の中央処理装置（ＣＰＵ）、アクセラレータプロセッサ、特定用途向けプロセッサおよび／またはフィールドプログラマブルゲートアレイ（ＦＰＧＡ）を備えるそれぞれの処理装置を備える。各ノード５０３はまた、メモリ、すなわち、１つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備える。メモリは、１つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ソリッドステートドライブ（ＳＳＤ）、フラッシュメモリもしくはＥＥＰＲＯＭなどの電子媒体、および／または光ディスクドライブなどの光学媒体を使用する１つまたは複数のメモリユニットを備え得る。

【００７２】

ＩｏＴネットワークの各ノード５０３は、ブロックチェーンノード１０４でもある。これらのノード５０３は、第１のネットワーク５０１とブロックチェーンネットワーク１０６との間のブリッジ（ゲートウェイ）として機能するブリッジノード（ゲートウェイノード）として構成される。ブロックチェーンノード１０４は、「リスニングノード」であり得る。リスニングノードは、ブロックチェーンの完全なコピーを保持し、新しいトランザクションおよびブロックを妥当性確認および伝搬するが、新しいブロックを積極的にマイニングまたは生成しないクライアントアプリケーションを実行する。代替的に、ノードは「簡略化された支払い検証ノード」（ＳＰＶノード）であってもよい。ＳＰＶノードは、ビットコイントランザクションを生成してブロードキャストし、アドレスを間接的に監視することができるが、ブロックチェーンの完全なコピーを保持しない軽量クライアントを実行する。

【００７３】

ＩｏＴネットワークの各ノード５０３は、エンドデバイス５０２を直接または間接的に制御するように構成される。エンドデバイス５０２に直接接続されたノード５０３は、そのデバイスを直接制御することができる。エンドデバイス５０２に直接接続されていないノード５０３は、そのデバイスを間接的にのみ、例えば、１つまたは複数の中間ノードを介してエンドノードに制御メッセージをフォワードすることによって、制御することができる。各ノード５０３は、１つまたは複数のマイニングノード１０４Ｍに接続される。

【００７４】

図５はまた、ブロックチェーンネットワーク１０６のサブセットであるマイニングノード１０４Ｍのネットワーク５０４を示す。マイニングノードについては、図１～図３を参照して上述している。マイニングノード１０４Ｍは、有効なトランザクション（例えば、ＩｏＴノードから送信されたトランザクション）をブロックチェーン１５０にマイニングするように構成される。

【００７５】

図５に示すように、ノード５０３は、Ｐ２Ｐネットワーク５０１およびブロックチェーンＰ２Ｐネットワーク１０６の両方の一部を形成し、一方、マイニングノード１０４Ｍは、ブロックチェーンＰ２Ｐネットワーク１０６のみの一部を形成する。エンドデバイス５０２は、Ｐ２Ｐ　ＩｏＴネットワーク５０１のみの一部を形成するものとして図５に示されているが、エンドデバイス５０２がブロックチェーンノード１０４でもあり得ることを除外するものではない。

【００７６】

図６は、例示的なＩｏＴネットワーク５０１トポロジーを示す。ＩｏＴネットワーク５０１は、マスタノード５０３ａ、１つまたは複数の中間ノード５０３ｂ、５０３ｃの１つまたは複数のセット６０１、およびエンドデバイス５０２のセットを制御することができる。マスタノード５０２ａは、１つまたは複数の中間ノード５０３ｂ、５０３ｃを制御するように構成される。ＩｏＴネットワーク５０１が中間ノードの複数のセット（例えば、層）６０１ａ、６０１ｂを含む場合、マスタノード５０３ａは、中間ノード（「サーバノード」５０３ｂ）の第１のセット（層）６０１ａを直接制御し、中間ノードの１つまたは複数のさらなるセット（層）６０１ｂ（例えば、「スレーブノード」５０３ｃの層）を間接的に制御するように構成される。マスタノード５０３ａは、サーバノードおよびスレーブノードをオーバーライドおよび制御する能力を有する制御ノードである。各サーバノード５０３ｂは、スレーブノード５０３ｃを制御する能力を有するノードである。スレーブノード５０３ｃは、サーバノード５０３ｂおよびマスタノード５０３ａの制御下にあるノードである。一例として、エンドデバイス５０２ａに命令するために、マスタノード５０３ａは、サーバントノード５０３ｂを介してスレーブノード５０３ｃにコマンドを発行する。

【００７７】

図６の例示的なＩｏＴネットワークは、中間ノード（サーバノードおよびスレーブノード）の２つの層のみを示すが、他の例は、例えば、マスタノード５０３ａとサーバノード５０３ｂとの間、および／またはサーバノード５０３ｂとスレーブノード５０３ｃとの間に、中間ノードの１つまたは複数のさらなるセットを含んでいてもよい。図示のように、各ノードは、それぞれの接続６０２を介して１つまたは複数の他のノードに接続され、各エンドデバイス５０２は、それぞれの接続６０２を介して１つまたは複数のスレーブノードに接続される。１つまたは複数のノード（例えばマスタノード）は、以下では制御ノードと称される。各制御ノードは、コマンドを発行することによってアクションを実行するように他のノードに命令することができるノード５０３である。

【００７８】

ＩｏＴネットワークノード５０３は、機能の範囲、命令／特権の優越性、および／またはアクセスのスパンにおける階層に対応し得る。いくつかの実装形態では、ＳＰＶノードの階層セットは、図５および図６のマスタノード５０３ａ、サーバノード５０３ｂ、およびスレーブノード５０３ｃに対応する３つの階層レベルを有する「ＩｏＴコントローラ」を実装する。マスタノード５０３ａは１つまたは複数のサーバノード５０３ｂに命令し、各サーバノードは１つまたは複数のスレーブノード５０３ｃに命令する。各スレーブノード５０３ｃは、１つまたは複数のサーバノード５０３ｂから命令を受信する。すべてのスレーブノード５０３ｃは、１つまたは複数のＩｏＴエンドデバイス５０２と通信し、これらは、ＩｏＴコントローラ５０３とＩｏＴエンドデバイス５０２との間の通信の直接チャネルである。ＩｏＴコントローラ５０３の実行の状態は、ブロックチェーントランザクションＴｘに記録される。各ＩｏＴノード（マスタ、サーバ、またはスレーブ）は、対応するトランザクションＴｘを作成し、ブロックチェーンネットワーク１０６にブロードキャストする能力を有する。各スレーブノードは、エンドデバイス５０２からのトリガおよび／または確認信号を監視し、すべてのＩｏＴノード５０３は、ＩｏＴコントローラの全体的な論理を実行する目的で任意の他のＩｏＴノードと対話する能力を有する。

【００７９】

マスタノード、サーバノード（複数可）、およびスレーブノード（複数可）は、それぞ

れ独立してブロックチェーンネットワーク１０６上のノード１０４に接続し、ブロックチェーンウォレットを動作させ（例えばブロックチェーンアドレスを監視するため）、場合によっては完全なノードを実行することができる（これは必須ではないが）。マスタノード５０３ａは、他のＩｏＴノードのアクティビティをそれらの制御下で直接および間接的の両方で監視し、ブロックチェーントランザクションＴｘの形態でこれらのノードにコマンドを発行し、アラートに応答するように構成される。サーバノード５０３ｂは、サーバノード５０３ｂによって直接制御されないアドレスを含む複数のアドレスを監視するように構成される。サーバノード５０３ｂは、マスタノード５０３ａによってアクションを実行するようにコマンドされ得る。スレーブノード５０３ｃは、エンドデバイス５０２のアクティビティをそれらの制御下で直接監視するように構成される。スレーブノード５０３ｃは、サーバノード５０３ｂの直接コマンド下にあり、マスタノード５０３ａによってアクションを実行するようにコマンドされることもできる。スレーブノード５０３ｃは、エンドデバイス５０２のためのゲートウェイノード（すなわち、エンドデバイスとブロックチェーンネットワーク１０６との間のゲートウェイ）として機能する。エンドデバイス５０２は、近くのスレーブデバイスに接続するように構成される。それらは、オフチェーンメッセージングプロトコルを使用してエンドデバイス状態について報告する。

【００８０】

エンドデバイス５０２はＩｏＴノード５０３によって制御されるがそれ自体はＩｏＴノード５０３を制御しないという点でＩｏＴノード５０３とエンドデバイス５０２とは区別されるが、エンドデバイス５０２はブロックチェーンネットワーク１０６のノード１０４であってもよいことに留意されたい。すなわち、いくつかの例では、エンドデバイス５０２は、ブロックチェーンプロトコルクライアントまたはウォレットアプリケーションを動作させ得る。

【００８１】

ＩｏＴネットワーク５０１は、コマンドおよび制御階層をブロックチェーンネットワークインフラストラクチャの使用と組み合わせることによって、集中化と分散化とを両立させる。ネットワーク５０１のユーザは、クライアント－サーバならびにデバイス間のピアツーピア関係を含むそれら独自のマルチレベル制御階層を作成し得る。ネットワークアーキテクチャは、ＩｏＴネットワーク５０１、ブロックチェーンＰ２Ｐネットワーク１０４（すなわち、フルおよび軽量ブロックチェーンクライアント、例えば、マスタ、サーバントおよびスレーブノードは、ＳＰＶウォレットを動作させる軽量クライアントである）、およびブロックチェーンマイニングネットワーク５０４（ＩｏＴノードによって伝搬されるトランザクションを妥当性確認し、伝搬し、記憶するブロックチェーンＰ２Ｐネットワークのサブセット）という３つの層を含む。ブロックチェーンネットワーク１０６は、バックエンドインフラストラクチャとして機能し、ＩｏＴネットワーク５０１とブロックチェーンＰ２Ｐネットワーク１０６との間に重複が存在する。

【００８２】

許可プロトコル

本開示の実施形態は、ネットワーク５０１へのアクセスを要求するノード５０３またはデバイス５０２に、ネットワーク５０１に参加する許可を与えるためのプロトコルを提供する。ＩｏＴのコンテキストでは、新しいノード５０３は、登録機関（例えば、ネットワーク内の信頼できるエンティティ）によって提供されるオンチェーン偽造防止デジタル証明書を使用して、ＩｏＴネットワーク５０１上で許可される。プロトコルは、真正なノードのみがネットワークにアクセスし、および／またはネットワーク内の他のノードまたはデバイスを制御することができることを保証することによって、サイバー攻撃に関連する問題を解決する。

【００８３】

ＩｏＴネットワーク５０１に参加する許可は、登録機関によって与えられる（登録機関は、「許可付与機関」または「認証局」とも称され得る）。登録機関は、要求エンティティ（例えば、要求ノードまたは要求デバイス）にデジタル証明書を発行することを担う。

有効な証明書を有するエンティティは、ＩｏＴネットワーク５０１へのアクセスを有する。登録機関は、それぞれのコンピュータ機器を備え、各コンピュータ機器が、１つまたは複数のプロセッサ、例えば、１つまたは複数の中央処理装置（ＣＰＵ）、アクセラレータプロセッサ、特定用途向けプロセッサおよび／またはフィールドプログラマブルゲートアレイ（ＦＰＧＡ）を備えるそれぞれの処理装置を備える。登録機関のコンピューティング機器はまた、メモリ、すなわち、１つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備える。メモリは、１つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ソリッドステートドライブ（ＳＳＤ）、フラッシュメモリもしくはＥＥＰＲＯＭなどの電子媒体、および／または光ディスクドライブなどの光学媒体を使用する１つまたは複数のメモリユニットを備え得る。

【００８４】

要求エンティティがネットワーク５０１に参加する許可を与えるために、登録機関は、以下で「証明書トランザクション」と称されるブロックチェーントランザクションＴｘを生成する。例示的な証明書トランザクションを図１６ａに示す。証明書トランザクションＴｘは、１つまたは複数の入力および１つまたは複数の出力を含む。少なくとも１つの入力１５０１ａは、登録機関のデジタル署名を含む。すなわち、登録機関は、デジタル署名を生成することができる第１の秘密鍵（例えば、第１の秘密鍵－公開鍵ペア）を有し、登録機関は、そのデジタル署名を使用してトランザクションに署名する。例示的な証明書フォーマットを図１６ｂに示す。証明書トランザクションに署名することによって、登録機関は、トランザクションの出力（複数可）に含まれるデータを証明する。デジタル署名を、第１の秘密鍵の知識を有する登録機関によってのみ生成することができる。トランザクションはまた、登録機関によってリクエスタに発行されたデジタル証明書を含む第１の出力１５０２ａ（例えば、使用不可出力）を有する。デジタル証明書は、リクエスタに割り当てられた識別子を含む。識別子は、ＩｏＴネットワーク５０１内のリクエスタに固有である。リクエスタには識別子が割り当てられ、この識別子は一度発行されると固定されたままである必要があり、デバイスが発行されるあらゆる証明書に表示される。好ましくは、デバイス識別子は、証明書が生成されるときに割り当てられる。しかしながら、これは、リクエスタがすでにデバイス識別子を有していることを除外するものではなく、デバイス識別子は証明書に含まれることによって証明される。

【００８５】

生成されると、登録機関は、ブロックチェーン１５０に記録されるように証明書トランザクションをブロックチェーンネットワーク１０６の１つまたは複数のノード１０４に送信する。ブロックチェーン１５０に記録されると、リクエスタは、証明書を使用して、リクエスタがネットワーク５０１に参加する許可を与えられたことをネットワーク５０１の他のノードまたはデバイスに証明することができる。例えば、ネットワーク５０１の他のノード５０３と通信するとき、リクエスタは、証明書トランザクションひいては証明書を識別する情報を含むことができる。

【００８６】

図１～図３を参照すると、これらの例では、第１のノードはアリス１０３ａのコンピュータ機器１０２ａであり得、第２のノードはボブ１０３ｂのコンピュータ機器１０２ｂであり得る。

【００８７】

リクエスタがネットワーク５０１のノード５０３である（またはノードとしてネットワーク５０１に参加する許可を要求している）場合、証明書は、そのノードに割り当てられた一意の公開鍵を含み得る。公開鍵により、要求ノード５０３は、ネットワーク５０１に参加した時点で、ブロックチェーントランザクションを送受信することが可能になる。

【００８８】

秘密鍵は、秘密鍵の所有者のみが知っている秘密の数値である。例えば、秘密鍵は２５６ビットの文字列であり得る。公開鍵は、秘密鍵から導出され、共有することができる関連する公開値である。例えば、公開鍵は、秘密鍵とｓｅｃｐ２５６ｋ１楕円曲線生成点と

の楕円曲線乗算によって計算され得る。署名は、例えば、楕円曲線デジタル署名アルゴリズム（ＥＣＤＳＡ）を使用して生成されるものなどの暗号署名であり得る。代替的な署名方式、例えばラビン署名が使用され得る。

【００８９】

証明書トランザクションは、登録機関の第２の公開鍵にロックされた第２の出力１５０２ｂを含み得る。第２の公開鍵は、証明書トランザクションに署名する署名を生成するために使用される公開鍵と同じ公開鍵であっても、異なる公開鍵であってもよい。第２の出力１５０２ｂは、第２の公開鍵の知識が出力をロック解除するために必要とされるという意味で、第２の公開鍵にロックされる。例えば、第２の出力は第２の公開鍵のハッシュを含み得、後のトランザクションの入力によってロック解除されるためには、その入力は第２の公開鍵を含んでいなければならない。第２の出力１５０２ｂが第２のトランザクションの入力と並行して実行されるとき、入力において提供される第２の公開鍵はハッシュされ、第２の出力１５０２ｂに含まれているハッシュと比較される。２つのハッシュが一致する場合、第２の出力１５０２ｂはロック解除され得る（任意の追加の制約が満たされていることを条件とする）。

【００９０】

出力は、Ｐ２ＰＫＨ（ｐａｙ－ｔｏ－ｐｕｂｌｉｃ－ｋｅｙ－ｈａｓｈ）を介して公開鍵にロックされ得る。Ｐ２ＰＫＨは、公開鍵ハッシュに出力をロックするスクリプトパターンである。Ｐ２ＰＫＨ出力は、公開鍵ハッシュと一致する公開鍵に対して有効な署名を受信者が提供する場合に使用され得る。すなわち、Ｐ２ＰＫＨ出力は、必ずしもその順序ではないが、公開鍵のハッシュがＰ２ＰＫＨ出力内のアドレスと一致するような公開鍵ならびに公開鍵およびトランザクションメッセージに対して有効な署名という２つのアイテムを提供するように使用者にチャレンジする。

【００９１】

第２の出力１５０２ｂが登録機関の公開鍵にロックされているので、登録機関のみが証明書を失効させることができる。これにより、証明書が悪意のある当事者から失効されるのを防ぐ。

【００９２】

第２の出力１５０２ｂは、第２の公開鍵にタイムロックされ得る。タイムロックされた出力は、所定の期間後までロック解除することができない。例えば、登録機関は、証明書トランザクションにロック時間を含め得る。ロック時間は、証明書トランザクションの第２の出力１５０２ｂが、特定の時間（これは、例えば、Ｕｎｉｘ時間またはブロック高さによって指定され得る）後まで、後のトランザクションによって成功裏に使用されるのを防ぐ。ロック時間は、トランザクションの「ｎＬｏｃｋｔｉｍｅ」フィールドを使用して実装され得る。ｎＬｏｃｋｔｉｍｅは、出力が使用可能になるまでの最小時間を命じるトランザクションのパラメータである。ｎＬｏｃｋｔｉｍｅと組み合わせて、オペコード（例えば、ＯＰ＿ＣＨＥＣＫＬＯＣＫＴＩＭＥＶＥＲＩＦＹ（ＣＬＴＶ））は、第１のトランザクション上のｎＬｏｃｋＴｉｍｅがオペコードに提供された時間パラメータ以上でない限り、（スクリプト実行を失敗させることによって）後のトランザクションが第２の出力を使用するのを防ぐ。後のトランザクションは、そのｎＬｏｃｋＴｉｍｅが過去のものである場合にのみ有効ブロックに含まれ得るので、後のトランザクションが有効ブロックに含まれ得る前に、ＣＬＴＶベースのタイムロックが満了していることが保証される。

【００９３】

追加的にまたは代替的に、第２の出力１５０２ｂは「マルチシグネチャ」出力であってもよい。マルチシグネチャ出力は、複数の公開鍵、すなわち登録機関の第２の公開鍵および少なくとも１つの他の公開鍵にロックされる。他の公開鍵は、ネットワーク５０１の別のノード、またはＩｏＴネットワークの外部であるがブロックチェーンネットワーク１０６内にある第三者ノードの公開鍵であり得る。第２の出力１５０２ｂを使用しようとする後のトランザクションの入力は、第２の出力がロックされる公開鍵ごとに１つずつ、複数の署名を含まなければならない。

【００９４】
タイムロックは、登録機関が、合意された時間の前に、または異なるノード（例えばマスタノード）の許可なしに、証明書を失効させるのを防ぐ。マルチシグネチャ出力は、証明書が異なるノード（例えばマスタノード）の許可なしに失効されるのを防ぐ。どちらの技法も、最小証明書持続時間長を強制する。
【００９５】
各トランザクションは、ブロックチェーン１５０に記録されると、一意のトランザクション識別子ＴｘＩＤによって識別することができる。トランザクション識別子は、シリアル化されたトランザクションバイトの（ダブル）ＳＨＡ２５６ハッシュを計算することによって生成され得る。ＳＨＡ２５６の代わりに他のハッシュ関数を使用してもよい。登録機関は、証明書トランザクションのトランザクション識別子をリクエスタに送信し得る。これにより、リクエスタは、証明書トランザクションを識別し、したがって証明書トランザクション内の証明書を取得することが可能になる。代替的に、リクエスタは、登録機関のアドレスからブロックチェーン１５０に送信されたトランザクションをリッスンしてもよい。
【００９６】
リクエスタがノード（例えば、サーバントノード）としてネットワーク５０１に参加している場合、要求ノードは、トランザクション識別子を使用して、登録機関の第１の公開鍵を取得し、その第１の公開鍵から送信された１つまたは複数のさらなるトランザクション（すなわち、さらなる証明書トランザクション）を識別し得る。さらなるトランザクションは各々、ネットワーク５０１の１つまたは複数のさらなるノードまたはデバイスのそれぞれの証明書を含み得る。次いで、リクエスタは、それらの証明書を取得（例えば、ダウンロードおよび保存）し得る。証明書内の情報（例えば、デバイス識別子および／または公開鍵）は、ネットワーク５０１の他のノード５０３および／またはデバイス５０２と通信するために使用され得る。例えば、リクエスタは、ブロックチェーントランザクションを別のノード５０３に、そのノードの証明された公開鍵を使用して、例えば、証明された公開鍵にロックされたトランザクションに出力（例えば、Ｐ２ＰＫＨ出力）を含めることによって送信し得る。コマンドを受信すると、リクエスタは、証明書を使用して、コマンドが許可されたノード５０３またはデバイス５０２から発行されたかどうかをチェックすることができる。
【００９７】
リクエスタがブロックチェーンにアクセスできないエンドデバイス５０２としてネットワーク５０１に参加している場合、登録機関は、例えば有線接続、または例えばＢｌｕｅｔｏｏｔｈ、Ｗｉ－Ｆｉなどのワイヤレス接続を介してエンドデバイスに証明書を送信し得る。登録機関はまた、１つまたは複数の第２の証明書のセットを要求エンドデバイス５０２に送信し得る。各々がネットワーク５０１のそれぞれのノードまたはエンドデバイスに発行されるこれらの第２の証明書を使用して、要求エンドデバイスの通信が許可されたノード５０３およびデバイス５０２との間で行われることを保証することができる。
【００９８】
各証明書（第１および第２）は、証明書が発行されるノード５０３またはエンドデバイス５０２のネットワークアドレス（例えば、ＩＰアドレス）を含み得る。リクエスタは、許可された（すなわち、証明された）ノードのネットワークアドレスを使用して、そのノードと通信することができ、例えば、センサ読取り値またはコマンド確認応答を送信することができる。
【００９９】
登録機関は、リクエスタに発行された証明書をネットワーク５０１の１つまたは複数のノードおよび／またはエンドデバイスに送信し得る。これらのエンドデバイスは、証明書を使用してリクエスタと通信し、リクエスタがネットワーク５０１に参加する許可を与えられたかどうかを検証し得る。
【０１００】

第三者が証明書の内容（これは機密情報を含み得る）を閲覧するのを防ぐために、登録機関は証明書を暗号化し得る。例えば、証明書は、登録機関の公開鍵（これは登録機関の第１および／または第２の公開鍵と同じであってもなくてもよい）に基づく暗号鍵を使用して暗号化され得る。代替的に、暗号鍵は、登録機関によって生成された乱数であってもよい。

【０１０１】

登録機関のいくつかの例では、登録機関は、単に、すなわち明示的な要求を受信することなく、証明書をリクエスタに発行し得る。他の例では、リクエスタは最初に登録機関に要求を送信し得る。要求は、リクエスタの１つまたは複数のクレデンシャルを含み得る。例えば、クレデンシャルは、デバイスタイプ（例えば、ラップトップ、電話、オーブン、冷蔵庫など）、ノードタイプ（例えば、マスタノード、サーバントノード、スレーブノード、エンドデバイス）、ネットワークアドレス（例えば、ＩＰｖ６アドレスであり得るＩＰアドレス）などのうちの１つまたは複数を含み得る。登録機関または異なるノード（例えば、マスタノード）は、要求を妥当性確認し得る。妥当性確認された場合、登録機関は、第１のトランザクションを生成し、それをブロックチェーンネットワーク１０６に送信し得る。要求が許可されない場合、登録機関はトランザクションを生成しないであろう。

【０１０２】

場合によっては、リクエスタに発行された証明書を失効させる必要があり得る。例えば、リクエスタが、危険にさらされている可能性がある場合、または障害が発生している可能性がある場合である。証明書を失効させるために、登録機関は、第２のブロックチェーントランザクション（「失効トランザクション」）を生成する。失効トランザクションは、証明書トランザクションの第２の出力（すなわち、登録機関の第２の公開鍵にロックされた出力）を参照する入力を有する。入力は、第２の公開鍵にリンクされた署名を含む。証明書トランザクションの第２の出力がＰ２ＰＫＨ出力である場合、失効トランザクションの入力は、公開鍵のハッシュ（例えば、ＯＰ＿ＨＡＳＨ１６０）がＰ２ＰＫＨ出力内の公開鍵ハッシュと一致するような公開鍵を含んでいなければならない。Ｐ２ＰＫＨ出力は、必ずしもその順序ではないが、公開鍵のハッシュがＰ２ＰＫＨ出力内のアドレスと一致するような公開鍵ならびに公開鍵およびトランザクションメッセージに対して有効な署名という２つのアイテムを提供するように、使用者にチャレンジする。

【０１０３】

失効トランザクションは、１つまたは複数の出力、例えば、登録機関の第３の公開鍵（登録機関の第１および／または第２の公開鍵と同じであっても同じでなくてもよい）にロックされた出力を含み得る。次いで、登録機関は、ブロックチェーン１５０に記録されるように、失効トランザクションをブロックチェーンネットワーク１０６に送信する。失効トランザクションがブロックチェーン１５０に記録されると、証明書トランザクションは、未使用トランザクション出力（ＵＴＸＯ）セットから除去される。ＵＴＸＯは、別のブロックチェーントランザクションによって使用されていないブロックチェーントランザクションからの出力である。ネットワーク５０１上の異なるノードが、要求ノードに発行された証明書を識別しようと試みると、そのノードは、証明書を含む証明書トランザクションが使用済みであることを知り、証明書が失効しているものと解釈する。ネットワーク５０１のノードは、発行アドレスからおよび発行アドレス（すなわち、第２の公開鍵）に対して生成されたトランザクションを監視することによって、それらのピアリスト（すなわち、許可された／証明されたノードのリスト）を動的に更新することができる。

【０１０４】

ノード／デバイス証明書の有効性は、発行鍵が（マスタ鍵によって署名された証明書に含まれる）認識された発行鍵であること、証明書が所定のプロトコルにしたがって正しくフォーマットされていること、および証明書トランザクション内の使用可能な出力が未使用であることという３つの基準に依存し得る。証明書は、失効すると更新することができる。登録機関は、古い証明書内のＵＴＸＯを使用し、次いで、更新された情報を有する新しい証明書Ｔｘを作成する。次いで、登録機関は、新しい証明書アウトポイントロケーシ

ョンインデックスをＩｏＴネットワーク５０１上のデバイスにブロードキャストすることができる。これは、登録機関自身の（自己署名された）証明書にも当てはまる。

【０１０５】

上述したように、証明書は、リクエスタの一意の識別子を含む。証明書は、リクエスタの固有の公開鍵を含み得る。一般に、証明書は、以下のフィールドのうちの１つまたは複数を含み得る：

【表１】

フィールドサイズ（バイト）	フィールド名	データタイプ	記述
4	IOTプロトコル識別子	uint32_t	プロトコルを示すプレフィックス
1	ペイロードタイプ	uint16_t	メッセージが通常のメッセージであるか証明書であるかを示す1バイト識別子。
32	新しいデバイスID	char[32]	新しいデバイスの一意のデバイスID
33	新しいデバイス公開鍵	char[32]	（ブロックチェーンP2Pネットワーク上にある場合）ノードと通信するために使用されるsecp256k1（圧縮された）公開鍵
4	デバイスタイプ	uint16_t	デバイスタイプ（例えば、ラップトップ、電話、オーブン、冷蔵庫、街灯柱など）
4	IOTノードタイプ	uint16_t	ノードタイプ、例えば、マスタ、サーバント、スレーブ、エンドデバイス。
16 + 2	IPv6アドレス＋ポート	Char[16]	IPv6ネットワークアドレス。 ネットワークバイト順序。
4	UNIX時間作成日	uint16_t	デバイス作成日
4	UNIX時間証明書満了日	uint16_t	証明書が満了し、新しい証明書が発行されなければならないUNIX時間。
0-80	追加デバイス情報	char[]	追加デバイス情報（製造者情報を含む）。

このフォーマットの証明書は、例えばブロックチェーントランザクションの使用不可能な（ＯＰ＿ＲＥＴＵＲＮ）出力に符号化された１０４～１８４バイトのデータを必要とする。

【０１０６】

上述したように、ネットワーク５０１はマスタノード５０３を含み得る。いくつかの例では、登録機関がマスタノード５０３ａを含んでもよい。リクエスタはノード５０３またはエンドデバイス５０２であり得る。したがって、マスタノード５０３ａ自体が、ノード５０３またはデバイス５０２に証明書を発行し得る。追加のまたは代替的な例では、リクエスタはマスタノード５０３ａであってもよい。マスタノード５０３ａが登録機関およびリクエスタの両方である場合、マスタノード５０３ａは、自身の証明書に自己署名する。

【０１０７】

ＩｏＴネットワーク５０１へのアクセスが許可され、許可（またはブートストラップ）アルゴリズムを使用して新しいエンティティ（ノードまたはデバイス）を認可する。２つの例示的なアルゴリズムを以下に提供する。マスタノード５０３ａは、ネットワーク５０１上で与えられる許可を要求する任意の新しいデバイスのクレデンシャルを（直接または

間接的に）検証することができる登録機関によって制御され得る。次いで、マスタノード５０３ｓは、ネットワーク５０１上の他のすべてのノード５０３にブロードキャストされることになるオンチェーン証明書を発行することができる。ブロックチェーンウォレットを動作させないデバイスのためのブートストラップアルゴリズムは、それがブロックチェーントランザクションを介する代わりにＩＰアドレス間の通信に依拠するので、異なる。

【０１０８】

例示的な許可アルゴリズム（マスタ／サーバント／スレーブ）

ステップ１：ノードを実行するコンピューティングデバイスは、すべての関連製造情報を含むそのクレデンシャルを登録機関に登録する。登録機関は、マスタノード５０３ａを動作するエンティティと同じエンティティであり得る。しかしながら、通常のコマンドまたはメッセージに署名するために使用される公開鍵は、好ましくは、証明書トランザクションに署名するために使用される公開鍵とは異なるべきである。

【０１０９】

ステップ２：鍵ＰＫ$_{Issue}$を制御する登録機関は、クレデンシャルの真正性を妥当性確認し、デバイスがＩｏＴネットワーク５０１に入ることを認可されるべきかどうかを決定する。

【０１１０】

ステップ３：デバイスが認可される場合、登録機関は、ノードのための一意の証明書トランザクションを作成する。このトランザクションは、ＰＫ$_{Issue}$からＰＫ$_{Issue}$アドレスにブロードキャストされる。登録機関は、ＴｘＩＤを新しいノードに送信する。

【０１１１】

ステップ４：新しいノードは、トランザクションを見つけ、登録機関でＰＫ$_{Issue}$を識別する。新しいノードは、ＰＫ$_{Issue}$によって発行され、現在アクティブであるすべての証明書をダウンロードし、検証する。

【０１１２】

ステップ５：すでにネットワーク５０１上にあるサーバントノードおよびスレーブノードは、ＰＫ$_{Issue}$との間でブロードキャストされるトランザクションをリッスンするように構成される。それらが、新しいトランザクションを見て、新しいデバイス証明書をダウンロードし、評価すると、それらは、新しいノードと通信するように自身のウォレットを構成することができる。

【０１１３】

例示的な許可アルゴリズム（エンドデバイス）

ステップ１：エンドデバイス５０２は、すべての関連製造情報およびＩＰアドレスを含むそのクレデンシャルを登録機関に登録する。この場合も、登録機関は、マスタノード５０３ａを動作するエンティティと同じエンティティであり得る。しかしながら、通常のコマンドまたはメッセージに署名するために使用される公開鍵は、好ましくは、証明書トランザクションに署名するために使用される公開鍵とは異なるべきである。

【０１１４】

ステップ２：鍵ＰＫ$_{Issue}$を制御する登録機関は、クレデンシャルの真正性を妥当性確認し、デバイス５０２がＩｏＴネットワーク５０１に入ることを認可されるべきかどうかを決定する。

【０１１５】

ステップ３：デバイス５０２が認可される場合、登録機関は、デバイスのための一意のデバイス証明書トランザクションを作成する。このトランザクションは、ＰＫ$_{Issue}$によって制御されるアドレスとの間でブロードキャストされる。証明書データは、ＩＯＴノードのみが詳細を見ることができるように暗号化される。

【０１１６】

ステップ４：登録機関は、エンドデバイス５０２がピアリストを作成することを可能にする証明書のリストを送信する。このリストは、エンドデバイス５０２がネットワーク５０１上の他のノード５０３ｃに接続し、公開鍵階層を解釈することを可能にする。

【０１１７】
ステップ５：ネットワーク５０１上のノード５０３は、デバイス証明書を見ることができ、ＩＰ－ＩＰ（ＴＬＳ）でエンドデバイスと通信することができる。
【０１１８】
要求および応答プロトコル
本開示はまた、ネットワーク（例えば、ＩｏＴネットワーク）５０１のノードがブロックチェーントランザクションＴｘを使用してコマンド要求を発行し、それらのコマンド要求に基づいてデバイスに命令し、コマンド確認応答を発行するためのプロトコルを提供する。実施形態は、ＩｏＴネットワーク５０１に関して説明されるが、一般に、本開示の教示は、ブロックチェーンプロトコルクライアントアプリケーション１０５を動作させるノードと、それらのノードの少なくともサブセットによって制御可能なエンドデバイスとを備える任意のネットワークに適用され得る。
【０１１９】
ネットワーク５０１の第１のブリッジノード５０３（例えば、マスタノード５０３ａまたはサーバノード５０３ｂ）は、第１のノードによって署名された入力とコマンドデータを含む出力とを含む第１のトランザクションTx_1を生成する。コマンドデータは、制御されるエンドデバイス５０２の識別子と、エンドデバイス５０２を制御するためのコマンドメッセージとを含む。第１のノードは、コマンドの発信元であり得る。すなわち、第１のノードがコマンドデータを生成し得る。
【０１２０】
第１のノードは、エンドデバイス５０２を制御する第１のネットワーク５０１の第２のブリッジノード５０３（例えば、スレーブノード５０３ｃ）に第１のトランザクションTx_1を送信し得る。第１のトランザクションTx_1は、オフチェーンで、すなわちブロックチェーンに送信されることなく、送信され得る。例えば、第１のトランザクションTx_1は、例えばインターネットを介して第１のノードから第２のノードに直接送信され得る。例えば、第１のノードはサーバノード５０３ｂであり得、第２のノードはスレーブノード５０３ｃであり得る。代替的に、第１のトランザクションTx_1は、例えば、１つまたは複数の中間ノードを介して間接的に送信されてもよい。一例として、第１のトランザクションTx_1は、サーバノード５０３ｂを介してマスタノード５０３ａからスレーブノード５０３ｃに送信され得る。第２のノードは、有線またはワイヤレス接続を介して、例えばイーサネットまたはＷｉ－Ｆｉ接続を介して、エンドデバイス５０２に接続され得る。
【０１２１】
追加的にまたは代替的に、第１のノードは、ブロックチェーン１５０に記録されるように第１のトランザクションTx_1をブロックチェーンネットワーク１０６に送信し得る。これは、第１のトランザクションTx_1が有効なトランザクションであることに依拠する。後述するように、第１のトランザクションTx_1をブロックチェーンに送信しないことが好ましい場合もある。
【０１２２】
図１～図３を参照すると、これらの例では、第１のノードは、アリス１０３ａのコンピュータ機器１０２ａで構成され得、第２のノードはボブ１０３ｂのコンピュータ機器１０２ｂで構成され得る。先に説明したように、アリスおよびボブは、当事者の一方がトランザクションをネットワーク１０６にブロードキャストすることを選択するまで、トランザクションが（まだ）ブロックチェーンネットワーク１０６上に公開されたりチェーン１５０上に進んだりすることなくことなく、サイドチャネル（例えばサイドチャネル３０１）を使用してトランザクションを交換し得る。
【０１２３】
第２のノードは、第１のノードから直接または間接的に第１のトランザクションTx_1を取得し得、例えば、第１のトランザクションTx_1は、１つまたは複数の中間ノードを介して第２のノードにフォワードされ得る。第２のノードは、コマンドデータを使用して、コマンドデータ内のデバイス識別子（「デバイスＩＤ」）によって識別されるエンドデ

バイス５０２に制御命令を送信する。コマンドデータ内の制御メッセージは、エンドデバイス５０２の所望のアクションを定義し得る。制御メッセージは、第２のノードに、いくつかの可能な命令のうちの特定の１つをエンドデバイス５０２に送信させるように構成され得る。代替的に、第２のノードは、単一の命令をエンドデバイス５０２に送信するように構成されてもよく、すなわち、第２のノードは、同じ命令のみをエンドデバイスに送信する。これは、例えば、エンドデバイス５０２がセンサのような単純なデバイスであり、命令がセンサ読取り値の要求である場合である。

【０１２４】

コマンド（すなわち、エンドデバイスに対する命令）は、例えばＷｉ－Ｆｉを使用して、有線またはワイヤレス接続を介してオフチェーンでデバイスに送信され得る。代替的に、デバイスがネットワークのノードでもある場合、コマンドはブロックチェーントランザクションＴｘを介して送信されてもよい。

【０１２５】

いくつかの実施形態では、デバイスおよびコントローラ通信のための要求および応答サイクルは、第１および第２のノードによって実装され得る。要求（コマンド）は、コマンドデータ（例えば、ＯＰ＿ＲＥＴＵＲＮペイロード）を含む出力を含む部分的に完全なトランザクションとして発行される。応答（コマンドの確認応答）は、リクエスタと応答ノードの両方の署名を含むファイナライズされたトランザクションのブロードキャストである。メッセージ受信者は、コマンドデータ（例えば、ＯＰ＿ＲＥＴＵＲＮペイロード）を変更することはできないが、入力および出力を追加することができるので、トランザクションの展性（ｍａｌｌｅａｂｉｌｉｔｙ）は、この通信方法を可能にする。

【０１２６】

オペコードは、スクリプトエンジン４０２内で使用される命令シラブルまたはパーセルであり、データに対するスタックベースの動作および暗号化動作を実行するようにマイナー１０４Ｍに命令する。ここで、スクリプトエンジン４０２は、ブロックチェーントランザクションＴｘ内のスクリプトを妥当性確認するために使用される実行環境であり、スタック４０３は、集合に要素を追加する「プッシュ（ｐｕｓｈ）」と、直近に追加された要素を除去する「ポップ（ｐｏｐ）」という２つの主要な動作を有するデータ構造（要素の集合）である。オペコードは、スタック要素に対して動作を実行するように設計されている。トランザクションＴｘを妥当性確認するとき、スクリプトエンジン４０２は、ＯＰ＿ＲＥＴＵＲＮオペコードの後の出力スクリプト（ＳｃｒｉｐｔＰｕｂｋｅｙ）内のデータを実行することはない。実際には、これは、残りのスクリプトデータが任意であり得、出力自体が使用不可であることを意味する（１つのブロックチェーンプロトコルにおいて、出力非使用可能性（ｎｏｎ－ｓｐｅｎｄａｂｉｌｉｔｙ）を保証するためにＯＰ＿ＦＡＬＳＥオペコードがＯＰ＿ＲＥＴＵＲＮに先行する必要がある）。

【０１２７】

第１のノードから第２のノードに送信された第１のトランザクションTx_1は、第２の出力なしで送信され得る。すなわち、トランザクションは単一の出力を含む（出力はコマンドデータを含む）。部分的なトランザクションを完了するために、第２は、第１のトランザクションに入力および出力を追加することによってトランザクションを更新し得る。入力は、第２のノードの署名、すなわち第２のノードの秘密鍵を使用して生成された署名を含む。出力は、第２のノードの公開鍵にロックされた出力、例えばＰ２ＰＫＨ出力である。Ｐ２ＰＫＨ出力を使用するために、使用するトランザクションの入力は、公開鍵のハッシュ（例えば、ＯＰ＿ＨＡＳＨ１６０）がＰ２ＰＫＨ出力内の公開鍵ハッシュと一致するような公開鍵を含まなければならない。Ｐ２ＰＫＨ出力は、必ずしもその順序ではないが、公開鍵のハッシュがＰ２ＰＫＨ出力内のアドレスと一致するような公開鍵ならびに公開鍵およびトランザクションメッセージに対して有効な署名という２つのアイテムを提供するように使用者にチャレンジする。公開鍵は、署名を生成するために使用される秘密鍵に対応し得る。代替的に、署名は第１の公開鍵にリンクされてもよく、出力は異なる公開鍵にロックされてもよい。次いで、第２のノードは、完了したトランザクションをブロッ

クチェーンネットワーク１０６に送信し得る。完了したトランザクション（これらの実施形態ではコマンドトランザクションと称される）は、他のノード、例えば第１のノードが閲覧するためにブロックチェーン１５０において利用可能であり、デバイスによって実行されたコマンドの記録として機能する。すなわち、トランザクションがブロードキャストされると、独立したオブザーバは、どの公開鍵がコマンド／メッセージを発行したか、およびどの公開鍵がそれに応答したかを見ることができる。

【０１２８】

図７ａおよび図７ｂは、例示的な部分的な第１のトランザクションTx_1（部分的）および例示的な更新された第１のトランザクションTx_1（完全）を示す。部分的な第１のトランザクションは、単一の入力７０１ａと単一の出力７０２ａとを含む。更新された第１のトランザクションは、第２のノードによって追加された入力７０１ｂおよび出力７０２ｂを含む。ＳＩＧＨＡＳＨ＿ＳＩＮＧＬＥ署名タイプを使用して、所望のレベルのトランザクション展性を達成することができる。例えば、公開鍵PK_0を有するノードは、公開鍵PK_1を有するノードに命令を送信する。命令は、ＳＩＧＨＡＳＨ＿ＳＩＮＧＬＥ署名タイプ（図１０ａ）を使用して署名されたトランザクションの使用不可出力（例えば、ＯＰ＿ＲＥＴＵＲＮ出力）に符号化される。部分的に完全なトランザクションは有効である。命令が完了すると、PK_1を有する第２のノードは、それらのアドレスにロックされた出力を追加する。次いで、PK_1を有する第２のノードは、ＳＩＧＨＡＳＨ＿ＡＬＬ署名タイプ（図１０ｂ参照）を使用してトランザクション全体に署名することによってトランザクションをファイナライズする。

【０１２９】

代替的な実施形態では、第１のノードから第２のノードに送信される第１のトランザクションTx_1は、第２の出力と共に送信され得る。第２の出力は、第２のノードの公開鍵にロックされる。例えば、第２の出力は、第２のノードの公開鍵に対するＰ２ＰＫＨであり得る。

【０１３０】

第１のトランザクションTx_1を完了するために、第２のノードは、第１のトランザクションに入力を追加することによって第１のトランザクションを更新する。この時点で、第１のトランザクションTx_1は、２つの入力と２つの出力とを含む。第２の入力は、第２のノードの公開鍵を含む。第２の入力内の公開鍵は、第２の出力がロックされる公開鍵と同じであっても、同じでなくてもよい。完了すると、更新された第１のトランザクション（これらの実施形態ではコマンドトランザクションと称される）は、ブロックチェーン１５０に含めるためにブロックチェーンネットワーク１０６に送信される。コマンドトランザクションがブロードキャストされると、どの公開鍵がコマンド／メッセージを発行したかおよびどの公開鍵がそれに応答したかを、任意の独立したオブザーバが見ることができる。

【０１３１】

第２のノードの公開鍵にロックされた第２の出力は、第１のトランザクションの第１の入力によって参照されるデジタル資産の額よりも多いデジタル資産の額を転送し得る。その場合、第１のトランザクションTx_1は、ブロックチェーンネットワーク１０６の他のノードによって有効とみなされない部分的に完全なトランザクションである。すなわち、第１のトランザクションTx_1は、ブロックチェーンノードが従うコンセンサスルールを満たさないため、ブロックチェーン１５０のブロック１５２にマイニングされないであろう。第１のトランザクションTx_1を更新するとき、第２のノードは、第１および第２の入力によって参照されるデジタル資産の総計額が、第２の出力にロックされたデジタル資産の額よりも大きいことを保証しなければならないであろう。

【０１３２】

図８ａおよび図８ｂは、例示的な部分的な第１のトランザクションTx_1（部分的）および例示的な更新された第１のトランザクションTx_1（完全）を示す。第１のトランザクションは、第２のノードの公開鍵にロックされた第１の出力８０２ａおよび第２の出力

８０２ｂにコマンドデータを含む。更新された第１のトランザクションは、第２のノードによって追加された追加の入力８０１ｂを含む。PK_0を有する第１のノードが、PK_1のみを有する第２のノードによって実行されることを望む命令を、PK_1を有する第２のノードに送信する場合、それらは、両方の出力８０２ａ、８０２ｂをロックするが手数料を支払わない（したがって、マイニングも伝搬もされない）部分的に完全なトランザクションを送信することができる。PK_1にロックされたデジタル資産を償還するために、PK_1を有する第２のノードは、手数料を支払う入力８０１ｂを提供する必要がある。部分的に完全なトランザクションを使用してコマンドを発行するために、

【数１】

$$<Sig_{PK_0}>$$

のためのＳＩＧＨＡＳＨフラグは、ＳＩＧＨＡＳＨ＿ＪＡＮＵＳＡＬＣＡＮＰＡＹに設定され、コマンドデータと共にＯＰ＿ＲＥＴＵＲＮ出力を含む。これは、第１の出力８０２ａに含まれるコマンドデータが固定されている間に、誰でも追加の入力を加えることができることを意味する。コマンドを受信した公開鍵は、入力８０１ａ内の資金を償還するために追加の入力８０１ｂを追加することができる。新しい入力８０１ｂを保護し、さらなるトランザクションの展性を防ぐために、資金の受信者は、最小値（ダスト）入力を追加し、ＳＩＧＨＡＳＨ＿ＡＬＬを使用してトランザクション出力に署名する。

【０１３３】

ＳＩＧＨＡＳＨフラグは、トランザクションのどの部分に署名が署名しているかを示すためにトランザクション入力内の署名に追加されるフラグであることに留意されたい。デフォルトはＳＩＧＨＡＳＨ＿ＡＬＬ（ＳｃｒｉｐｔＳｉｇ以外のトランザクションのすべての部分が署名される）である。トランザクションの無署名部分は修正することができる。

【０１３４】

図９、図１０ａ、および図１０ｂを参照して、例示的な要求および応答アルゴリズムを以下に提供する。制御デバイス５０３ｂは、ネットワーク５０１上の他のノードと通信するように構成され、ネットワーク上の任意の他のノードへの最短通信経路を計算することができる。例えば、PK_{serv}は、PK_{slave}がｄｅｖｉｃｅ＿ＩＤを有するデバイスに最も近いコントローラであることを識別する。

【０１３５】

ステップ１：公開鍵PK_{serv}を有する制御デバイス５０３ｂが、公開鍵PK_{slave}を有する第２の制御デバイス５０３ｃに部分的なコマンドTx_1（図１０ａ参照）を送信する。トランザクションに含まれるＩｏＴメッセージは、コマンドおよびｄｅｖｉｃｅ＿ＩＤを有するターゲットデバイスを指定する。

【０１３６】

ステップ２：第２の制御デバイス（PK_{slave}）は、トランザクションの署名が有効であること、およびＩｏＴメッセージペイロード内に含まれるメッセージがネットワーク５０１の規則にしたがって有効であることをチェックする。

【０１３７】

ステップ３：第２の制御デバイス（PK_{slave}）は、オフチェーン通信（例えば、有線接続、Ｂｌｕｅｔｏｏｔｈ、ＩＰ－ｔｏ－ＩＰ）を介してコマンドメッセージ（「Ｍｓｇ」）をデバイス（ｄｅｖｉｃｅ＿ＩＤ）に送信する。

【０１３８】

ステップ４：コマンド要求されたアクションの完了時に、デバイス（ｄｅｖｉｃｅ＿ＩＤ）は、コマンド完了または確認応答メッセージ（「ａｃｋ」）を第２の制御デバイス（PK_{slave}）に送り返す。

【０１３９】

ステップ５：第２のコントローラ（PK_{slave}）は、第２の入力および署名を追加し、トランザクションをファイナライズする（図１０ｂ参照）。これは、第２のコントローラがコマンドの完了を確認したことを知らせる。

【０１４０】
ステップ６：第２のコントローラ（PK_{slave}）は、ファイナライズされたトランザクションをブロックチェーン（マイニング）ネットワーク５０４にブロードキャストする。
【０１４１】
いくつかの実施形態では、第１のトランザクションTx_1は、トランザクションではなく、コマンド要求トランザクションTx_1（要求）であり得る。すなわち、第１のノードは、エンドデバイス５０２を制御するための承認を求める要求をネットワークの２つ以上のノードに送信し得る。例えば、第１のノードは、マスタノード５０３ａおよび異なるブリッジノード５０３（例えば、別のサーバノード５０３ｂ）からの承認を要求するサーバノード５０３ｂであり得る。第１のトランザクションTx_1は、エンドデバイス５０２を制御するためのコマンドメッセージを含むコマンドデータを含む。しかしながら、第１のノードは、エンドデバイス５０２を制御する１つのノードに第１のトランザクションTx_1を送信する代わりに、コマンド要求を承認することができる２つ以上のノードに第１のトランザクションTx_1を送信する。すなわち、第１のノードは、ネットワーク５０１の２つ以上のノードにロックされた出力を含む。出力をロック解除するためには、後のトランザクションTx_2の入力において各ノードからの署名を提供しなければならない。第１のトランザクションTx_1は、ブロックチェーン１５０に含めるためにブロックチェーンネットワーク１６０にブロードキャストされる。２つ以上のノードが（それらの公開鍵または公開鍵アドレスに支払われたトランザクションをリッスンすることによって）トランザクションTx_2を見ると、２つ以上のノードは、コマンド要求の承認を望む場合、それぞれがコマンド承認トランザクションTx_2（承認）の入力に署名し、そのトランザクションTx_2をブロックチェーンネットワーク１０６にブロードキャストする。コマンド承認トランザクションTx_2は、コマンド要求トランザクションTx_1と同じコマンドデータと、エンドデバイス５０２を制御する第２のノードの公開鍵にロックされた追加の出力とを含む。
【０１４２】
図１１ａおよび図１１ｂは、例示的な第１および第２のトランザクションを示し、第１のトランザクションTx_1はコマンド要求トランザクションであり、第２のトランザクションTx_2はコマンド承認トランザクションである。第１のトランザクションTx_1は、第１のノードによって署名された入力１１０１ａと、コマンドデータを含む第１の出力１１０２ａと、１つは第３のノードの公開鍵であり１つは第４のノード（ＩｏＴネットワーク５０１のブリッジノード５０３でもある）の公開鍵という２つの異なる公開鍵にロックされた第２の出力１１０２ｂとを含む。第２のトランザクションTx_2は、第３および第４のノードによって署名された入力１１０１ｂと、第１の出力１１０２ａと、第２のノードの公開鍵にロックされた第２の出力１１０２ｃとを含む。マルチシグネチャスクリプトにより、コマンドに対するマルチファクタ承認が可能になる。場合によっては、コマンドの解釈または証明書の妥当性確認には、２つ以上のノード（例えば、ＩｏＴネットワークのノード）からの署名が必要であり得る。それはまた、使用可能な出力がＵＴＸＯセット内にあるかどうかをチェックすることを必要とし得る。例えば、公開鍵PK_0を有する第１のノードは、公開鍵PK_3を有する第２のノードにアクションを実行するように命令することを望み得る。セキュリティ要件として、このコマンドは、公開鍵PK_1およびPK_2をそれぞれ有する第３および第４のノードが承認することを必要とし得る。公開鍵PK_0を有する第１のノードは、マルチシグネチャアドレスに資金を送信する第１のトランザクションを作成する。第１のトランザクションは、承認を必要とするコマンドデータ（PK_3へのコマンドを符号化する）も含む。それに応答して、公開鍵PK_1およびPK_2を有するノード５０３は、両方が署名を提供する場合、マルチシグネチャアドレスからの資金を使用するオプションを有する。承認は、コマンド要求トランザクションにおける出力の使用として解釈される。これらの図は、Ｐ２ＭＳ（ｔｗｏ－ｏｆ－ｔｗｏ　ｐａｙ　ｔｏ　ｍｕｌｔｉ－ｓｉｇ）出力を示しているが、一般に、Ｐ２ＭＳ出力は、ｎ－ｏｆ－ｎ出力とすることができ、ここで、ｎは任意の整数であり、Ｐ２ＭＳは、支払人が出力を複数のア

ドレスにロックすることを可能にするスクリプトパターンのタイプである。使用されるために、出力は、公開鍵の指定されたセットからの１つまたは複数の署名を必要とし得る。

【０１４３】

いくつかの実施形態では、上述したように、第１のノードは、コマンド要求トランザクションTx_1を生成し得る。追加的にまたは代替的に、第１のノードは、ＩｏＴネットワーク５０１の異なるノードによって生成されたコマンド要求トランザクションに応答して、コマンド承認トランザクションTx_2を生成し得る。例えば、第１のノードは、要求に対する承認を与えることができるマスタノード５０３ａであり得る。その場合、ブロックチェーン１５０は、第１のノードの公開鍵およびＩｏＴネットワーク５０１の１つまたは複数のさらなるノード５０３の１つまたは複数のそれぞれの公開鍵にロックされた出力を含むコマンド要求トランザクションを含む。例えば、第１のノードは、第２のノードによって制御されるエンドデバイス５０２に発行されたコマンドを承認するノードであり得る。第１のノードがコマンドを承認する場合、第１のノードは、コマンド要求トランザクションの出力を参照するコマンド承認トランザクションに署名する。第１のノードが、コマンド承認トランザクションに署名する最後のノードである場合、第１のノードはトランザクションをブロックチェーンネットワーク１０６に送信する。ブロードキャストトランザクションTx_2は、コマンド承認要求であると解釈される。

【０１４４】

第２のノード（例えば、スレーブノード）がコマンドトランザクションまたはコマンド承認トランザクションを取得すると、第２のノードは、コマンド（Ｍｓｇ）を、コマンドデータ内のデバイス識別子（Ｄｅｖｉｃｅ＿ＩＤ）によって識別されるデバイスに送信する。いくつかの例では、デバイス５０２は、コマンドを受信したことおよび／またはコマンドを実行したことを示すために、確認応答メッセージ（Ａｃｋ）を第２のノードに送信し得る。これらの例では、第２のノードは、デバイスから確認応答を受信したことを条件として、第１のトランザクションを更新する（次いで、更新されたトランザクションをブロードキャストする）ことのみし得る。これは、エンドデバイスがコマンドを実行したことのさらなる補強証拠を提供する。

【０１４５】

ほとんどの日常的な小型電子デバイスが有するリソース制約に起因して、それらは、ブロックチェーン１５０を容易に監視することができず、かつ／またはそれらのすぐ近くのロケーションの外側のＩｏＴネットワークコンポーネントと通信することさえできない場合があり、したがって、エンドデバイス５０２の制御は、ローカルに（第２のノードからデバイスへ）かつオフチェーンで実行される。エンドデバイスへのメッセージおよびエンドデバイスからのメッセージは、追加のトランザクションメタデータを伴わない生のコマンドデータ（例えば、ＯＰ＿ＲＥＴＵＲＮペイロード）の形態をとり得る。これは、メッセージを含むデータパケットが小さいままであり、計算集約的な演算（楕円曲線数学など）が必要とされないことを保証する。

【０１４６】

いくつかの実施形態では、トランザクションに含まれるコマンドデータは暗号化される。コマンドデータは、乱数に基づく暗号鍵で暗号化され得るが、暗号鍵は、第１のノードの公開鍵および第２のノードの公開鍵に基づいて生成されることが好ましい。第１のノードおよび／または第２のノードの公開鍵は、証明された公開鍵（証明された公開鍵については後述する）であり得る。証明された公開鍵は、第１および／または第２のノードに発行されるそれぞれの証明書に含まれる。これらの公開鍵は、第１のトランザクションを生成および更新するために第１および第２のノードによって使用される公開鍵とは異なり得る。すなわち、署名を生成するためにまたはトランザクションの出力をロックするために使用される公開鍵（「トランザクション公開鍵」）は、暗号鍵を生成するために使用される公開鍵と同じではない可能性がある。他の例では、トランザクション公開鍵は、暗号鍵を生成するために使用され得る。

【０１４７】

コマンドデータを暗号化するために使用される暗号鍵は、第１および第２のノードによって独立して生成され得る。例えば、第１のノードは、第１のノードに知られている秘密鍵および第２のノードの公開鍵に基づいて暗号鍵を生成し得る。第２のノードは、第１のノードの秘密鍵に対応する公開鍵と第２のノードの公開鍵に対応する秘密鍵とに基づいて暗号鍵を生成し得る。これにより、第１のノードと第２のノードの両方がコマンドデータを復号できることが保証され、その結果、例えば、第２のノードは、デバイスに命令するためにコマンドメッセージにアクセスすることができる。

【０１４８】

任意選択的に、コマンドデータを含む第１のトランザクションの出力は、コマンドデータを暗号化するために使用される暗号鍵を含み得る。暗号化された鍵は、第２の異なる暗号鍵で暗号化され得る。第２の暗号鍵を知っている当事者は、暗号化された暗号鍵を復号し、暗号化されたコマンドデータを復号することができる。例えば、マスタノードなどのエンティティが、ネットワークのノードとの間で送信されるすべてのコマンドメッセージを閲覧することができることは有益であり得る。第２の暗号鍵は、第１のノードの公開鍵および異なるノード、例えばマスタノードの公開鍵に基づいて生成され得る。第１のノードの公開鍵は、第１のノードの証明された鍵またはトランザクション公開鍵であり得る。同様に、マスタノードの公開鍵は、マスタノードの証明された鍵またはトランザクション公開鍵であり得る。

【０１４９】

暗号化は対称であってもよい。対称暗号化は、インターネット上の通信に対してＨＴＴＰＳが提供するのと同程度のプライバシーをオンチェーンデータに提供する。これを行うために、マスタノードは、ローカルＩｏＴネットワーク上のノード間のすべての通常のメッセージを暗号化するために使用される秘密暗号鍵を作成する。各ＩｏＴトランザクションＯＰ＿ＲＥＴＵＲＮペイロード（すなわち、コマンドデータを含む出力）は、２つのデータチャンクを有し得る。第１に、ＢＩＥ１　ＥＣＩＥＳがＩｏＴメッセージを暗号化し、ここで、暗号鍵は、要求デバイスおよび応答デバイスの（証明された）公開鍵を使用して導出される（エンドツーエンド）。第２に、ＢＩＥ１　ＥＣＩＥＳがＩｏＴメッセージの暗号化／復号鍵を暗号化する。楕円曲線統合暗号化方式（ＥＣＩＥＳ）は、ディフィーヘルマン交換に基づく暗号化方式である。第２の暗号鍵（これはこのデータプッシュを暗号化するために使用される）は、マスタ公開鍵およびリクエスタ公開鍵を使用して導出される。追加のバイトプッシュは、それ自体が要求ノードとマスタとの間でＢＩＥ１を使用して暗号化されたＩｏＴメッセージのための復号鍵を含む暗号化されたペイロードの最後に追加される。これは、マスタノードが、ネットワーク上のデバイス間で送信された復号済みデータを閲覧することができることを保証する。他の暗号化技法、例えばＡＥＳ（Ａｍｅｒｉｃａｎ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）暗号化が使用されてもよい。

【０１５０】

図１２ａおよび図１２ｂは、暗号化されたペイロードデータを有する記述されたコマンドおよび応答トランザクションを示す。

【０１５１】

上述したように、第１および第２のノードは、トランザクションを生成および更新するときにトランザクション公開鍵を使用し得る。公開鍵は、ＩｏＴネットワーク上のノードを識別するために使用されるが、好ましくは、それらの公開鍵は、トランザクションに署名するために使用されるべきではない。例えば、各ノードは、登録機関によって発行された証明書に含まれる公開鍵を有し得る。証明された鍵（例えば、マスタノードによって署名された対応する証明書を有する公開鍵）の所有者は、自分の身元を第三者に隠し、トランザクションに署名するために使用することができる共有秘密を導出することができる。

【０１５２】

上述したように、ＩｏＴネットワーク５０１は、マスタノード５０３ａを備え得る。マスタノード５０３ａは、シード鍵を取得（例えば、生成）し、次いで、各々がシード鍵に基づく秘密鍵のセット（例えば、複数の秘密鍵）を生成し得る。次いで、マスタノード５

０３ａは、秘密鍵のセット（以下ではジョイント秘密鍵と称される）をネットワークのノード（例えば、サーバノードおよびスレーブノード）に送信し得る。各ノードは、シード秘密鍵ではなく、ジョイント秘密鍵の同じセットを受信する。

【０１５３】

各ノードは、それぞれの主秘密鍵、例えば、そのノードの証明された公開鍵に対応する秘密鍵を有する。マスタノード５０３ａを含む各ノードは、ジョイント秘密鍵のセットを使用して、対応する二次（またはトランザクション）秘密鍵のセットを生成する。トランザクション秘密鍵は、それぞれのノードの主秘密鍵を各ジョイント鍵に追加することによって生成される。各ノードについて、対応するトランザクション公開鍵のセットが、トランザクション秘密鍵のセットから生成され得る。

【０１５４】

第１のノードは、マスタノード５０３ａ、すなわち、ジョイント秘密鍵のセットを生成するノードであり得る。代替的に、第１のノードは、マスタノードからジョイント秘密鍵のセットを受信する中間ノード（例えば、サーバノードまたはスレーブノード）であってもよい。いくつかの例では、各ノードは、トランザクション公開鍵を１回だけ使用し得る。

【０１５５】

証明された鍵からトランザクション鍵を生成するための例示的な鍵マスキングアルゴリズムが以下に提供される。ローカルＩｏＴネットワーク５０１上のノード５０３はすべて、公開鍵を登録する証明書が発行されている。具体的には、サーバノードは証明された鍵PK_{Serv}を有し、マスタノードは秘密鍵sk_{Master}を有する証明された鍵PK_{Master}を有する。

【０１５６】

ステップ１：マスタノード５０３ａは、シードからマスタ拡張秘密鍵m_{joint}を生成する。mを使用して、ＩｏＴネットワーク５０１にわたって共有され、各ノードアドレスをマスキングするために使用される鍵のウォレットが生成される。ジョイントウォレットは以下のインデックスキーを有する：

【数２】

$$sk_{i,j}^{joint},\ PK_{i,j}^{joint} = sk_{i,j}^{joint} \cdot G$$

【０１５７】

ステップ２：マスタノード５０３ａは、オフチェーンエンドツーエンド暗号化方式（例えば、ＢＩＥ１　ＥＣＩＥＳ）を使用して、オフチェーンメッセージにおいてＩｏＴネットワーク上のノード他のノードとmを共有する。

【０１５８】

ステップ３：サーバノード５０３ｂがmを取得すると、ウォレット内の階層的決定論的鍵ペアのセットを導出することができる。

【０１５９】

ステップ４：ネットワーク５０１上の各ノード５０３は、それらの証明された鍵ペアからの秘密鍵を、ジョイントウォレットから生成された秘密鍵に追加することによって、それらのウォレット秘密鍵を生成する。例えば、マスタノードは、トランザクション署名鍵

【数３】

$$s_{i,j}^{M}$$

を生成し、ここで、以下である：

【数４】

$$sk_{i,j}^{M} = sk_{i,j}^{joint} + sk_{Master}$$

【０１６０】
ステップ５：各ノード５０３は、他のノード公開鍵をジョイントウォレットからの公開鍵に追加することによって、ＩｏＴネットワーク５０１上の他のノード５０３のすべての支払いエンドポイント（アドレス）を識別することができる。例えば、サーバノード５０３ｂは、マスタノードの支払いアドレス公開鍵
【数５】
$$PK_{i,j}^{M}$$

を導出することができき、ここで以下である：
【数６】
$$PK_{i,j}^{M} = PK_{i,j}^{joint} + PK_{Master},$$

【０１６１】
ＩｏＴネットワーク５０１上の各ノード５０３は、それ自体のウォレットを導出し、他のデバイスのアドレスを、それらがmおよび関連ＩｏＴ証明書のロケーションを知っている場合に監視することができる。
【０１６２】
暗号化および鍵マスキングの両方は、ローカルＩｏＴネットワーク５０１上のすべてのノード５０３に対するデバイス可視性を依然として保証しながら、第三者に漏洩されることからデバイスアクティビティデータを保護する。
【０１６３】
ＩｏＴネットワーク５０１のノード５０３によって送信される各トランザクションは、コマンドデータを含む出力を含む。出力および／またはコマンドデータは、出力がコマンドデータを含むことを示すためにプロトコルフラグを含み得る。これは、ＩｏＴデバイスおよび独立した第三者が、いつオンチェーンコマンド、アクション、またはステータス更新が発生したかを識別することを可能にする。
【０１６４】
図１３は、第１のトランザクションの例示的なコマンドデータ出力を示す。第１のトランザクションは、第１のノードの署名を含む入力（図示せず）と、コマンドデータを含む出力１３０１とを含む。第１のトランザクションはまた、第２の出力（図示せず）を含み得るが、これについては後述する。この例では、プロトコル識別子（４バイト）に続いて、ＩｏＴ通信情報を含む９３バイトのペイロードがある。通信情報は、コマンド命令の対象の受信者の３２バイトのデバイスＩＤ、デバイス証明書のロケーション、コマンド、およびデバイスステータスを含む。いくつかの例では、新しいコマンドまたはステータス更新を発行するすべてのトランザクションは、このフォーマットに従わなければならず、そうでなければ無効なコマンドと見なされる。フィールドがいずれのオンチェーンメッセージにも必要でない場合、そのバイトは０ｘ００００００００に設定され得る。好ましくは、後述するように、ペイロードデータ自体が暗号化される。ペイロードデータには、復号鍵を保持する当事者のみアクセスすることができる。以下の表は、例示的なＩＯＴメッセージペイロードのフィールドを説明する。
【０１６５】

【表２】

フィールドサイズ（バイト）	記述		データタイプ	コメント
4	IOTプロトコル識別子		uint32_t	IOTプロトコルを示すプレフィックス
1	ペイロードタイプ		uint16_t	Sメッセージが通常のIOTメッセージであるか証明書であるかを示す1バイト識別子。
4	ソフトウェアバージョン番号		uint32_t	IOTバージョン番号（プロトコル更新／アップグレードに必要）
32	デバイスID		char[32]	コマンド／メッセージを受けるデバイスの一意のデバイスID。
40 (32 + 4 + 4)	デバイス証明書ロケーション	TXID	char[32]	デバイス証明書を含むトランザクションのトランザクションID
		VOUT	uint16_t	証明書TX内の失効UTXOロケーション
		VOUT	uint16_t	証明書ペイロードの出力番号
4	コマンド／メッセージ		uint32_t	デバイスIDを有するデバイスに向けられたコマンドまたはメッセージを符号化する文字列。
4	ステータス		uint16_t	現在のデバイスステータス
4	前のステータス		uint16_t	デバイスの直近の前のステータス

デバイス状態レプリカは、デバイスの報告された状態または所望の状態の論理表現である。ＩｏＴメッセージ内で、デバイス状態情報は、デバイスＩＤ、ステータス、および前のステータスに符号化される。デバイスＩＤに関する最新のトランザクションは、現在のデバイスステータスを表す。デバイスのステータスに関連するコマンド、応答、およびデータを含むメッセージは、ブロックチェーン上のタイムスタンプ付きブロックに含まれ、公開鍵暗号およびプルーフオブワークを使用して保護される。

【０１６６】

要約すると、ＩｏＴネットワーク１０６上のノード５０３は、ＩｏＴコマンドデータを含むトランザクションを使用して直接通信するだけでなく、ブロックチェーンネットワーク５０１に接続してトランザクションをブロードキャストすることによっても通信する。ブロックチェーン１５０は、ＩｏＴネットワークコンポーネントからのコマンドおよびステータス更新を記録するため、ならびにＩｏＴデバイス５０２に関連する報告およびアラートを発行するための永続的データストアとして使用される。プロトコルは、以下の特徴のうちの１つまたは複数を利用し得る。

【０１６７】

要求および応答メッセージングシステム－コマンドを受信および確認応答するための要求および応答システムが使用される。要求は、エンドデバイスが解釈することができるＩｏＴ論理を符号化するオフライン（ピアツーピア）トランザクションである。応答または確認応答は、ブロックチェーンネットワーク１０６上のトランザクションの可視性から解釈される。

【０１６８】

オフライントランザクション伝搬－トランザクションに符号化された命令が直接送信される（ピアツーピア）。マスタノード、サーバノードおよびスレーブノードは、トランザクション署名を検証することによって、トランザクションのソースを独立して検証することができる。これは、コントローラの支払い方法としても機能する。

【０１６９】

ノードとエンドデバイスとの間の直接通信－トランザクションコマンドペイロードに符号化された命令がエンドデバイスを対象とする場合、サーバノードまたはスレーブノード

は、トランザクションから命令を抽出し、それをエンドデバイスに直接通信することができる。

【０１７０】

アクションの確認応答としてトランザクションをブロードキャストする－トランザクションをブロックチェーンネットワークにブロードキャストすることは、コマンド内に符号化されたアクションがデバイスＩＤを有するデバイスによって実行されたことを示す。

【０１７１】

デバイスステータスおよび履歴を符号化するマイニングされたトランザクション－ブロックチェーンは、完全なデバイスステータスおよび履歴を記憶する（論理的に）集中型のおよび物理的に分散されたデータベースとして機能する。

【０１７２】

実施形態は、以下の有利な特徴のうちの１つまたは複数を提供する。

【０１７３】

基礎となるブロックチェーンインフラストラクチャのセキュリティ－値の転送を符号化し、ＩｏＴ対話をログするすべてのトランザクションは、公開鍵暗号およびプルーフオブワークを使用して保護される。ｓｅｃｐ２５６ｋ１パラメータに基づく楕円曲線暗号（ＥＣＣ）は、ＩｏＴノードを識別するために使用される公開／秘密鍵を保護し、プルーフオブワークは、ＩｏＴネットワークステータスおよび履歴を記録するブロックチェーンを保護する。

【０１７４】

安全な鍵管理および難読化－鍵難読化技法は、機密公開鍵がそれらの対応する秘密鍵の過剰使用によって脆弱にされないことを保証するために使用される。鍵難読化はまた、ＩｏＴソリューションコンポーネントが、それらのパブリックアドレスをマスキングすることによってプライバシーを高めることを可能にする。

【０１７５】

暗号化－含まれるデバイス固有データは、復号鍵を有するＩｏＴノードのみがアクセスを得ることができるように、エンドツーエンドで暗号化される（例えば、ＢＩＥ１またはＡＥＳ）。

【０１７６】

例示的な使用事例

例示的な使用事例としては、公共図書館用のプリンタサービスがある。ほとんどの公的に資金提供された図書館または大学図書館では、１枚当たり最低額（例えば、６～１０ペンス）を請求することによって印刷コストが支払われる。現在の（集中型）モデル内では、ユーザは、図書館管理によって管理されるアカウントを開く。アカウントは事前に入金される必要があり、トランザクションは図書館で運用されるソフトウェアによって管理されなければならないため、図書館にとって管理上の負担が大きくなる。本開示は、許可プロトコルとピアツーピア制御プロトコルとを併用することによってこの問題を解決する。図１４は、Ｐ２Ｐ印刷のための例示的なＩｏＴネットワーク５０１を示す。１）図書館管理者は、（図書館管理者が制御する）マスタノード５０３ａを確立し、プリンタ（エンドデバイス）５０２を直接制御するスレーブノード５０３ｃを構成する。２）管理者は、スレーブノード５０３ｃおよびエンドデバイス５０２がメッセージを解釈するために使用することとなるルールエンジンを構成する。ルールエンジンは、１つまたは複数のルールを実行するシステムである。３）図書館管理者はスレーブノード５０３ｃを構成する。これは、物理的な動作を実行するようにプリンタに直接命令することができる支払い受信装置である。４）新規の図書館ユーザは、標準的な登録／ログイン方法（例えば、ユーザ名およびパスワードなどの１つまたは複数のクレデンシャル）を使用してマスタノード５０３ａによって許可され、検証される。ＩｏＴ許可アルゴリズムはバックエンドで実行される。図１５ａは、新規の図書館ユーザに証明書を発行するために図書館管理者によって使用される例示的なトランザクションを示す。この場合、図書館管理者はマスタノード５０３ａであるとともに、証明書失効権限を有する登録機関である。５）ユーザがアイテムの印

刷を望む場合、印刷された文書は図書館イントラネットシステム内で送信され得る。文書と共にコマンドトランザクションがある。このトランザクションは、プリンタを調整するスレーブノードへの支払い、プリンタデバイスＩＤ、およびＳＩＧＨＡＳＨ＿ＳＩＮＧＬＥトランザクション署名を含む。図１５ｂは、ラップトップ（サーバ）によって（スレーブ）コントローラに送信される例示的なコマンドトランザクションを示す。６）コントローラ（スレーブノード）は、トランザクションが有効なブロックチェーントランザクションであり、トランザクションソース（公開鍵）がシステム上で許可されており、トランザクション値が、コマンドに含まれる命令のコストを支払うのに十分であることを認証する。７）すべてのチェックにパスした場合、スレーブノードは、ユーザによって要求されたアクションを実行するようにプリンタに命令する。８）スレーブノードは、支払いをロックする出力をそれらのアドレスに追加し、ＳＩＧＨＡＳＨ＿ＡＬＬを有する署名を追加し、次いで、トランザクションをブロックチェーンネットワーク１０６にブロードキャストする。図１５ｃは、コントローラによってブロックチェーンネットワーク１０６にブロードキャストされる例示的なコマンド－確認応答トランザクションを示す。

【０１７７】

結論

上記の実施形態は、単なる例として説明されていることが理解されよう。より一般的には、以下のステートメントのいずれか１つまたは複数による方法、装置、またはプログラムを提供することができる。

【０１７８】

ステートメント１．第１のネットワークに参加する許可をリクエスタに与えるためのコンピュータ実装方法であって、第１のネットワークは、ブリッジノードのセットと、ブリッジノードのセットのうちの１つまたは複数によって制御可能なデバイスのセットとを含み、各ブリッジノードはブロックチェーンネットワークのそれぞれのノードでもあり、方法は、登録機関によって実行され、以下を含む：

－第１のブロックチェーントランザクションを生成すること、ここで、第１のブロックチェーントランザクションは、登録機関の第１の公開鍵にリンクされた署名を含む入力と、第１の証明書を含む第１の出力とを含み、第１の証明書は、リクエスタに割り当てられた識別子を含み、および

－ブロックチェーンに含めるために第１のブロックチェーントランザクションをブロックチェーンネットワークに送信すること。

【０１７９】

ステートメント２．第１のトランザクションは、登録機関の第２の公開鍵にロックされた第２の出力を含む、ステートメント１に記載の方法。

【０１８０】

ステートメント３．第１の出力は登録機関の第２の公開鍵にタイムロックされ、タイムロックは所定の期間後まで第１の出力がロック解除されるのを防ぐ、ステートメント２に記載の方法。

【０１８１】

ステートメント４．第１の出力は少なくとも登録機関の第２の公開鍵および異なる公開鍵にロックされる、ステートメント２またはステートメント３の方法。

【０１８２】

異なる公開鍵は、ネットワークのノードまたはネットワークの一部ではない第三者の公開鍵であり得る。これは、証明書失効には複数の署名が必要であることを意味する。

【０１８３】

ステートメント５．第１のブロックチェーントランザクションのトランザクション識別子を許可リクエスタに送信することを含む、ステートメント１から４のいずれかに記載の方法。

【０１８４】

ステートメント６．証明書は暗号鍵で暗号化され、暗号鍵は登録機関によって生成され

る、ステートメント１から５のいずれかに記載の方法。
【０１８５】
例えば、暗号鍵は、登録機関によって生成された乱数であり得る。
【０１８６】
ステートメント７．ステートメント１から６のいずれかに記載の方法であって、以下を含む：
－ネットワークに参加するための要求をリクエスタから受信すること、ここで、要求は１つまたは複数のクレデンシャルを含み、および
－１つまたは複数のクレデンシャルに基づいて要求を妥当性確認すること、ここで、第１のブロックチェーントランザクションを上記生成することは、要求が有効であることを条件とする。
【０１８７】
１つまたは複数のクレデンシャルは、例えば、リクエスタの製造情報および／またはＩＰアドレスを含み得る。
【０１８８】
ステートメント８．ブリッジノードのセットは、マスタノードと、マスタノードによって制御可能な中間ノードのセットとを含み、登録機関はマスタノードである、ステートメント１から７のいずれかに記載の方法。
【０１８９】
ステートメント９．ブリッジノードのセットは、マスタノードと、マスタノードによって制御可能な中間ノードのセットとを含み、リクエスタはマスタノードである、ステートメント１から８のいずれかに記載の方法。
【０１９０】
ステートメント１０．リクエスタはブロックチェーンネットワークのそれぞれのノードであり、証明書は許可リクエスタに割り当てられた公開鍵を含む、ステートメント１から９のいずれかに記載の方法。
【０１９１】
ステートメント１１．ステートメント１から８のいずれかに記載の方法であって、リクエスタは、第１のネットワークの１つまたは複数のブリッジノードによって制御可能なデバイスであり、方法が以下を含む：
－証明書のセットをリクエスタに送信すること、ここで、セット内の各証明書は、ノードのセットのうちのそれぞれの１つに送信されている。
【０１９２】
ステートメント１２．第１の証明書をブリッジノードのセットのうちの１つまたは複数に送信することを含む、ステートメント１から１１のいずれかに記載の方法。
【０１９３】
ステートメント１３．ステートメント２またはそれに従属する任意のステートメントに記載の方法であって、以下を含む：
－第２のブロックチェーントランザクションを生成すること、ここで、第２のブロックチェーントランザクションは、第１のトランザクションの第２の出力を参照する入力を含み、登録機関の第２の公開鍵にリンクされた署名を含み、および
－ブロックチェーンに含めるために第２のブロックチェーントランザクションをブロックチェーンネットワークに送信すること。
【０１９４】
ステートメント１４．第１のネットワークに参加する許可を要求するためのコンピュータ実装方法であって、第１のネットワークは、ブリッジノードのセットと、ブリッジノードのセットのうちの１つまたは複数によって制御可能なデバイスのセットとを含み、各ブリッジノードはブロックチェーンネットワークのそれぞれのノードでもあり、方法は、リクエスタによって実行され、以下を含む：
－第１のネットワークに参加する要求を登録機関に送信すること、および

－第１の証明書を取得すること、ここで、証明書は、登録機関によって発行され、リクエスタに割り当てられた識別子を含む。

【０１９５】

ステートメント１５．ステートメント１４に記載の方法であって、上記取得することが以下を含む：

－第１の証明書を含む第１のブロックチェーントランザクションのトランザクション識別子を受信すること、および

－トランザクション識別子を使用してブロックチェーンから第１のブロックチェーントランザクションを取得すること。

【０１９６】

ステートメント１６．ステートメント１５に記載の方法であって、第１のブロックチェーントランザクションは、証明書を含む第１の入力と、登録機関の公開鍵にリンクされた第２の出力とを含み、方法は以下を含む：

－登録機関の公開鍵を識別すること、および

－登録機関の公開鍵からブロックチェーンに送信された１つまたは複数のそれぞれのトランザクションに含まれる１つまたは複数の第２の証明書を識別すること、ここで、各第２の証明書は、それぞれのブリッジノードもしくはデバイスまたはネットワークに発行される。

【０１９７】

ステートメント１７．ステートメント１６に記載の方法であって、第１の証明書は、リクエスタの公開鍵を含み、第１のネットワークのブリッジノードのセットのそれぞれの１つに発行された各第２の証明書は、上記ノードのそれぞれの公開鍵を含み、方法は以下を含む：

－上記ブリッジノードのセットのうちの少なくとも１つに第３のブロックチェーントランザクションを送信すること、ここで、第３のブロックチェーントランザクションは、少なくとも１つのブリッジノードのそれぞれの公開鍵にロックされた出力を含む。

【０１９８】

ステートメント１８．第１の証明書を上記取得することは、登録機関から第１の証明書を受信することを含む、ステートメント１４から１７のいずれかに記載の方法。

【０１９９】

ステートメント１９．ステートメント１４から１８のいずれかに記載の方法であって、以下を含む：

－登録機関から１つまたは複数の第２の証明書を受信すること、ここで、各第２の証明書は、第１のネットワークのブリッジノードまたはデバイスのセットのうちのそれぞれの１つに発行される。

【０２００】

ステートメント２０．ステートメント１９の方法であって、第１の証明書は、リクエスタのネットワークアドレスを含み、第１のネットワークのそれぞれのブリッジノードに発行された各第２の証明書は、上記ノードのそれぞれのネットワークアドレスを含み、方法は以下を含む：

－ブリッジノードのセットのうちの１つまたは複数にメッセージを送信すること、ここで、メッセージは、リクエスタのネットワークアドレスから、メッセージが送信される１つまたは複数のブリッジノードのそれぞれのネットワークアドレスに送信される。

【０２０１】

ネットワークアドレスはＩＰアドレスであり得る。

【０２０２】

ステートメント２１．リクエスタは、第１のネットワークのデバイスのセットのうちの１つである、ステートメント１４またはステートメント１８から２０のいずれかに記載の方法。

【０２０３】

ステートメント２２．リクエスタは、第１のネットワークのノードのセットのうちの１つである、ステートメント１４から２１のいずれかに記載の方法。

【０２０４】

ステートメント２３．ブリッジノードのセットは、マスタノードと、マスタノードによって制御可能な１つまたは複数の中間ノードとを含み、リクエスタはマスタノードである、ステートメント２２に記載の方法。

【０２０５】

ステートメント２４．要求は、リクエスタの１つまたは複数のクレデンシャルを含む、ステートメント１４から２３のいずれかに記載の方法。

【０２０６】

ステートメント２５．１つまたは複数のクレデンシャルはリクエスタのＩＰアドレスを含む、ステートメント２４に記載の方法。

【０２０７】

ステートメント２６．コンピュータ機器であって、以下を備える：

－１つまたは複数のメモリユニットを備えるメモリ、および

－１つまたは複数の処理ユニットを備える処理装置、ここで、メモリは、処理装置上で実行されるように構成されたコードを記憶し、コードは、処理装置上にあるときに、ステートメント１から１３のいずれかに記載の方法を実行するように構成される。

【０２０８】

ステートメント２７．コンピュータ可読ストレージ上に具現化され、ステートメント２６に記載のコンピュータ機器上で実行されると、ステートメント１から１３のいずれかに記載の方法を実行するように構成されたコンピュータプログラム。

【０２０９】

ステートメント２８．コンピュータ機器であって、以下を備える：

－１つまたは複数のメモリユニットを備えるメモリ、および

－１つまたは複数の処理ユニットを備える処理装置、ここで、メモリは、処理装置上で実行されるように構成されたコードを記憶し、コードは、処理装置上にあるときに、ステートメント１４から２５のいずれかに記載の方法を実行するように構成される。

【０２１０】

ステートメント２９．コンピュータ可読ストレージ上に具現化され、ステートメント２８に記載のコンピュータ機器上で実行されると、ステートメント１４から２５のいずれかに記載の方法を実行するように構成されたコンピュータプログラム。

【０２１１】

本明細書に開示される教示の別の態様によれば、登録機関および許可リクエスタのアクションを含む方法が提供され得る。

【０２１２】

本明細書に開示される教示の別の態様によれば、登録機関および許可リクエスタのコンピュータ機器を備えるシステムが提供され得る。

【０２１３】

本明細書に開示される教示の別の態様によれば、第１および／または第２のブロックチェーントランザクションを含むトランザクションのセットが提供され得る。

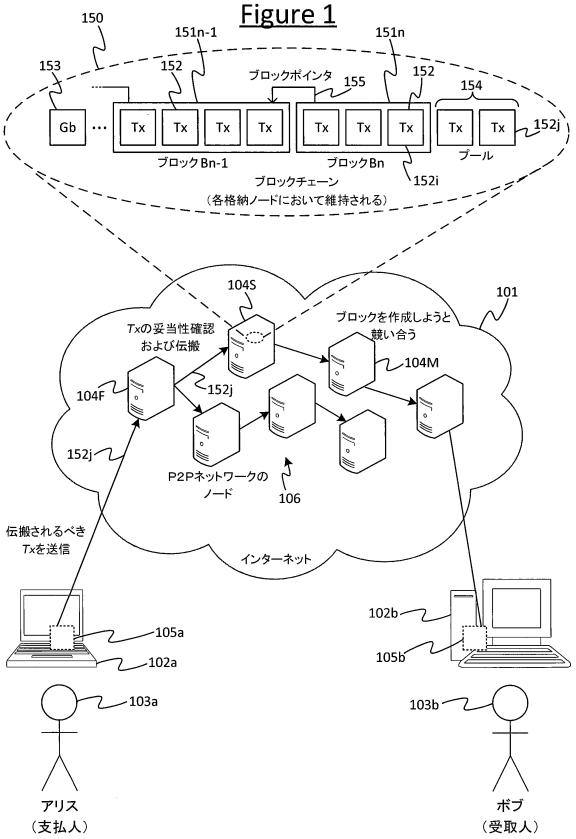
【０２１４】

開示された技法の他の変形または使用事例は、本明細書の開示が与えられれば、当業者に明らかになり得る。本開示の範囲は、説明した実施形態によって限定されず、添付のステートメントによってのみ限定される。

【図面】

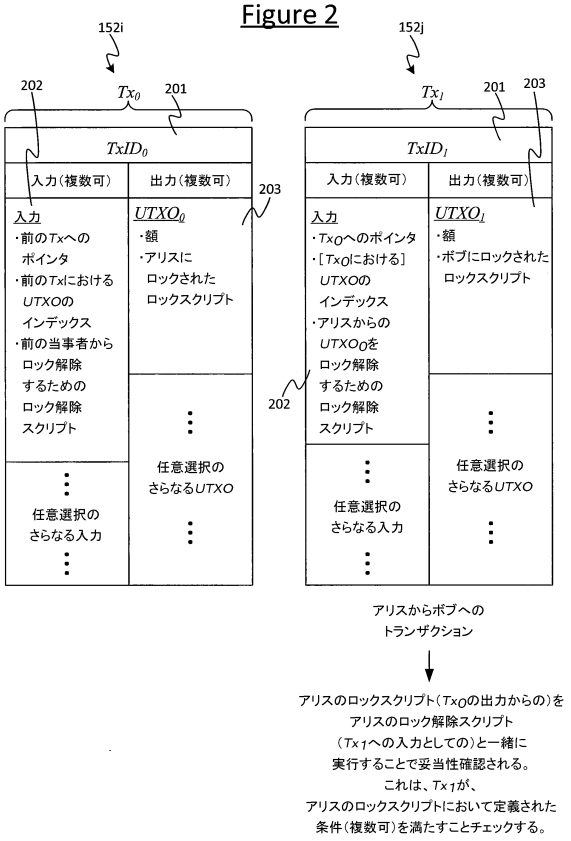
【図１】

Figure 1



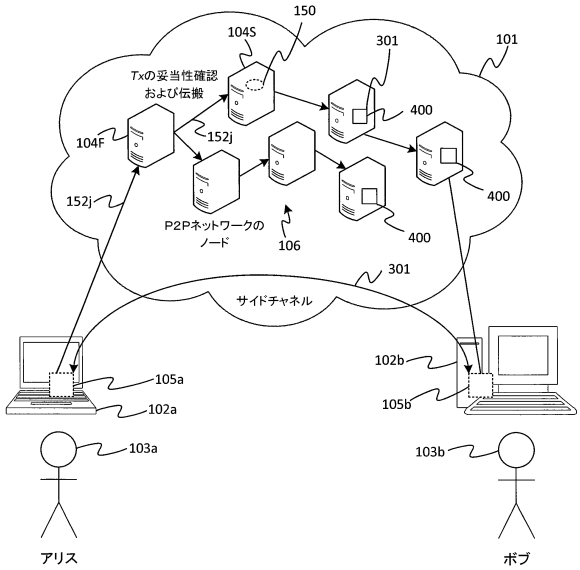
【図２】

Figure 2



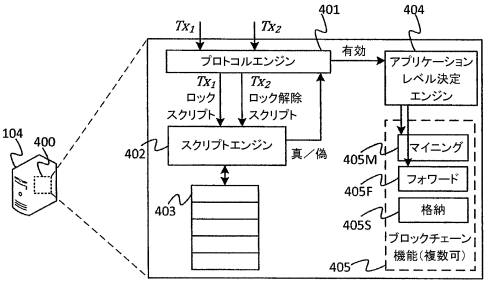
【図３】

Figure 3



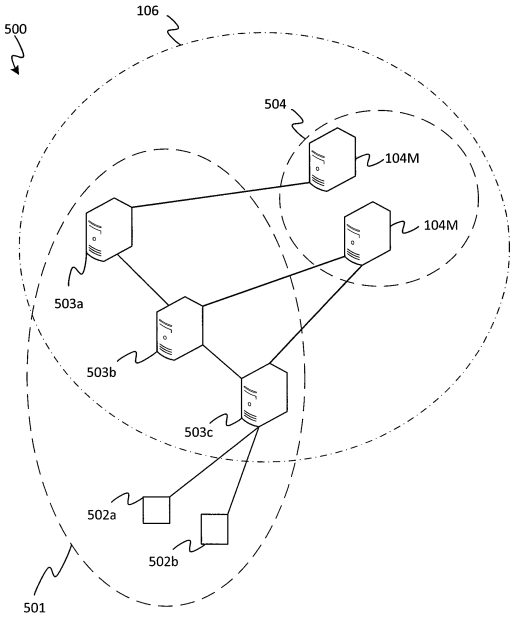
【図４】

Figure 4



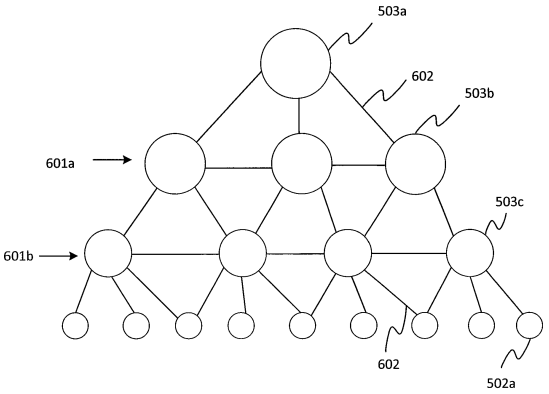
【図５】

Figure 5



【図６】

Figure 6



【図７ａ】

Figure 7a

Tx_1（部分的）			
入力		出力	
値		値	
x	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >

701a　702a

【図７ｂ】

Figure 7b

Tx_1（完全）			
入力		出力	
値		値	
x	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
> 0	$< Sig_{PK1} > < PK_1 >$	x	OP_DUP OP_HASH160 $< H_{160}(PK_1) >$ OP_EQUALVERIFY OP_CHECKSIG

701b　702b

【図８ａ】

Figure 8a

Tx_1（部分的）			
入力		出力	
値		値	
x	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_1) >$ OP_EQUALVERIFY OP_CHECKSIG

801a　802b　802a

【図８ｂ】

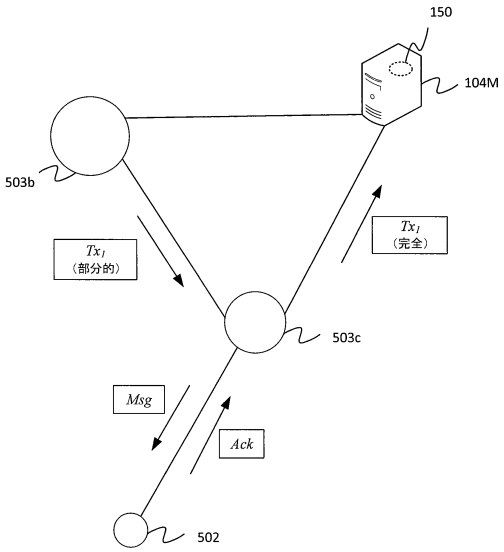
Figure 8b

Tx_1（完全）			
入力		出力	
値		値	
x	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
$> \delta$	$< Sig_{PK1} > < PK_1 >$	x	OP_DUP OP_HASH160 $< H_{160}(PK_1) >$ OP_EQUALVERIFY OP_CHECKSIG

801b

【図９】

Figure 9



【図１０ａ】

Figure 10a

Tx_1（部分的）			
入力			
値		値	出力
x	$< Sig_{PKserv} > < PK_{serv} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_{slave}) >$ OP_EQUALVERIFY OP_CHECKSIG

【図１０ｂ】

Figure 10b

Tx_1（完全）			
入力		出力	
値		値	
x	$< Sig_{PKserv} > < PK_{serv} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
$> \delta$	$< Sig_{PKslave} > < PK_{slave} >$	x	OP_DUP OP_HASH160 $< H_{160}(PK_{slave}) >$ OP_EQUALVERIFY OP_CHECKSIG

【図１１ａ】

Figure 11a

Tx_1（要求）			
入力		出力	
値		値	
x_1	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		y_1	OP_2 OP $< PK_1 > < PK_2 >$ OP_CHECKMULTISIG

1101a　1102b　1102a

【図１１ｂ】

Figure 11b

Tx_2（承認）			
入力		出力	
値		値	
x_2	$< Sig_{PK1} > < PK_1 >$ $< Sig_{PK2} > < PK_2 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		y_2	OP_DUP OP_HASH160 $< H_{160}(PK_3) >$ OP_EQUALVERIFY OP_CHECKSIG

1101b　1102c

【図１２ａ】

Figure 12a

Tx_1（部分的）			
入力		出力	
値		値	
x	$< Sig_{PKserv\_0} > < PK_{serv\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 OP_PUSHDATA1 [*Payload length*] <*BIE1 Encrypted command data*> OP_PUSHDATA1 [*Payload length*] < *BIE1 Encrypted Encryption key* >
		$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_{slave\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

【図１２ｂ】

Figure 12b

Tx_1（完全）			
入力		出力	
値		値	
x	$< Sig_{PKserv\_0} > < PK_{serv\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 OP_PUSHDATA1 [*Payload length*] <*BIE1 Encrypted command data*> OP_PUSHDATA1 [*Payload length*] < *BIE1 Encrypted Encryption key* >
$> \delta$	$< Sig_{PKslave\_1} > < PK_{slave\_1} >$	$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_{slave\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

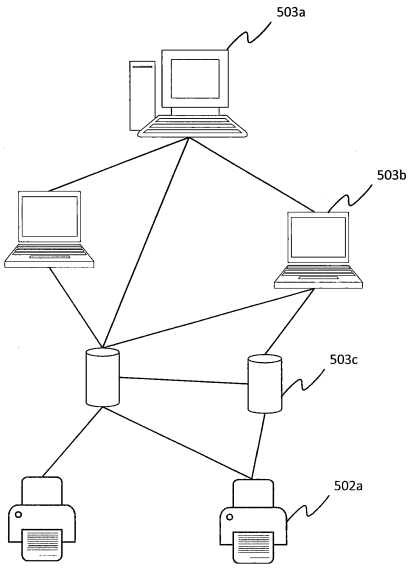
【図１３】

Figure 13

出力	
値	
0	OP_FALSE OP_RETURN OP_PUSHDATA1 0x4d494f54 – *IoT protocol identifier* 0x01 – *Payload type* 0x00000001 – *IoT software version number* 0x3dd5dfac...32 – *Device ID* 0x234a3789...22 – *Device Pubkey* 0x4d348912...87 – *Device certificate location data* 0x3ad21fac – *Command* 0x5665b456 – *Status* 0x5665b456 – *Previous status*

【図１４】

Figure 14



【図１５ａ】

Figure 15a

Tx_1（証明書）			
入力		出力	
値		値	
x_1	$< Sig_{PKadmin} > < PK_{admin} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Certificate data* >
		y_1	OP_DUP OP_HASH160 $< H_{160}(PK_{admin}) >$ OP_EQUALVERIFY OP_CHECKSIG

【図１５ｂ】

Figure 15b

Tx_1（部分的）			
入力		出力	
値		値	
x_2	$< Sig_{PKlaptop\_0} > < PK_{laptop\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Encrypted Command data* >
		y_2	OP_DUP OP_HASH160 $< H_{160}(PK_{controller\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

【図１５ｃ】

Figure 15c

Tx_1（コントローラ）			
入力		出力	
値		値	
x_2	$< Sig_{PKlaptop\_0} > < PK_{laptop\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Encrypted Command data* >
x_3	$< Sig_{PKcontroller\_0} >$ $< PK_{controller\_0} >$	y_2	OP_DUP OP_HASH160 $< H_{160}(PK_{controller\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

【図１６ａ】

Figure 16a

Tx_1（証明書）			
入力		出力	
値		値	
x_1	$< Sig_{PKreg} > < PK_{reg} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Certificate data* >
		y_1	OP_DUP OP_HASH160 $< H_{160}(PK_{reg}) >$ OP_EQUALVERIFY OP_CHECKSIG

1501a　1502b　1502a

【図１６ｂ】

Figure 16b

出力	
値	
0	OP_FALSE OP_RETURN OP_PUSHDATA1 < *payload lengih* > 0x4d494f54 – *IoT protocol identifier* 0x02 – *Payload type* 0x00000001 – *IoT software version number* 0x3dd5dfac...32 – *New device ID* 0x234a3789...22 – *New device Compressed Pubkey* 0x3ad21fac – *Device type* 0x00000004 – *Device IoT node type* 0x0000...06f – *IPv6 address and port number* 0x5664b456 – *UNIX time creation date* 0x5665b456 – *UNIX time certificate expiry date* 0x76d5335c…..45 – *Additional information*

フロントページの続き

(72)発明者　タータン，クロイー
イギリス ダブリュー１ダブリュー ８エーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内
(72)発明者　ワハブ，ジャド
イギリス ダブリュー１ダブリュー ８エーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内
(72)発明者　セルギエヴァ，アントアネータ
イギリス ダブリュー１ダブリュー ８エーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内
(72)発明者　ライト，クレイグ スティーヴン
イギリス ダブリュー１ダブリュー ８エーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内

審査官　金沢 史明
(56)参考文献　米国特許出願公開第２０１９／０１９９５３５（ＵＳ，Ａ１）
特表２０１９－５１５３７３（ＪＰ，Ａ）
国際公開第２０１９／１１５９３６（WO，Ａ１）
鈴木 明日香, 他，ビットコインにおけるユーザへの信頼性付与の手法，電子情報通信学会技術研究報告，日本，電子情報通信学会，2019年07月16日，Vol. 119, No. 140，pp. 29-34
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／３２
Ｈ０４Ｌ　９／０８