

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7596373号
(P7596373)

(45)発行日 令和6年12月9日(2024.12.9)

(24)登録日 令和6年11月29日(2024.11.29)

(51)国際特許分類		F I		
H O 4 L	9/32 (2006.01)	H O 4 L	9/32	2 0 0 Z
		H O 4 L	9/32	2 0 0 B
		H O 4 L	9/32	2 0 0 E
請求項の数 35 (全56頁)				
(21)出願番号	特願2022-525178(P2022-525178)	(73)特許権者	318001991	
(86)(22)出願日	令和2年10月5日(2020.10.5)		エヌチェーン ライセンシング アーゲー	
(65)公表番号	特表2023-500258(P2023-500258 A)		スイス・6 3 0 0・ツーク・グラーフェ	
(43)公表日	令和5年1月5日(2023.1.5)	(74)代理人	100107766	
(86)国際出願番号	PCT/IB2020/059323		弁理士 伊東 忠重	
(87)国際公開番号	WO2021/084348	(74)代理人	100070150	
(87)国際公開日	令和3年5月6日(2021.5.6)		弁理士 伊東 忠彦	
審査請求日	令和5年9月12日(2023.9.12)	(74)代理人	100135079	
(31)優先権主張番号	1915842.7		弁理士 宮崎 修	
(32)優先日	令和1年10月31日(2019.10.31)	(72)発明者	マッケイ, アレグザンダー	
(33)優先権主張国・地域又は機関	英国(GB)		イギリス ダブリュー1ダブリュー 8エ	
			ービー ロンドン マーケット プレイス	
			3 0 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する要求および応答プロトコル

(57)【特許請求の範囲】

【請求項1】

第1のネットワークの1つまたは複数のデバイスを制御す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ノードでもあり、各デバイスはそれぞれのデバイス識別子を有し、前記コンピュータ実装方法は、前記ブリッジノードのうちの第1のノードによって実行され、

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を含み、ここで、前記第1のブロックチェーントランザクションは、前記第1のノードの第1の公開鍵にリンクされた署名を含む第1の入力と、コマンドデータを含む第1の出力とを含み、前記コマンドデータは、前記ブリッジノードのうちの第2のノードによって制御される前記デバイスのうちの第1のデバイスのそれぞれの識別子と、前記第1のデバイスを制御するためのコマンドメッセージとを含む、

方法。

【請求項2】

前記第1のブロックチェーントランザクションは、前記第2のノードの第1の公開鍵にロックされた第2の出力を含む、請求項1に記載の方法。

【請求項3】

前記第1のブロックチェーントランザクションを前記第2のノードに送信することを含

む、請求項 1 または 2 に記載の方法。

【請求項 4】

ブロックチェーンは前のブロックチェーントランザクションを含み、前記前のブロックチェーントランザクションは、前記ブリッジノードのうちの第 3 のノードの第 1 の公開鍵にリンクされた署名を含む第 1 の入力と、前記第 1 のデバイスの前記それぞれの識別子および前記第 1 のデバイス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第 1 の出力と、少なくとも前記第 1 のノードの前記第 1 の公開鍵および前記ブリッジノードのうちの第 4 のノードの第 1 の公開鍵にロックされた第 2 の出力とを含み、前記第 1 のブロックチェーントランザクションの前記第 1 の入力は、前記第 4 のノードの前記第 1 の公開鍵にリンクされた署名を含む、請求項 2 に記載の方法。

10

【請求項 5】

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ことを含む、請求項 4 に記載の方法。

【請求項 6】

前記コマンドメッセージは、前記第 2 のノードに、制御命令を前記第 1 のデバイスに送信させるように構成され、前記制御命令は前記コマンドメッセージに基づく、請求項 1 から 5 のいずれか一項に記載の方法。

【請求項 7】

前記コマンドデータは、暗号化されたコマンドデータ、または他の方法で符号化されたコマンドデータである、請求項 1 から 6 のいずれか一項に記載の方法。

20

【請求項 8】

前記コマンドデータは、前記第 1 のノードの第 2 の公開鍵および前記第 2 のノードの第 2 の公開鍵に基づいて生成された第 1 の暗号鍵を使用して暗号化される、請求項 7 に記載の方法。

【請求項 9】

前記ブリッジノードのセットはマスタノードを含み、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 1 の出力は暗号化データを含み、前記暗号化データは、前記第 1 のノードの第 3 の公開鍵および前記マスタノードの第 1 の公開鍵に基づいて生成された第 2 の暗号鍵を使用して暗号化された前記第 1 の暗号鍵を含む、請求項 8 に記載の方法。

【請求項 10】

30

前記第 1 のネットワークは、マスタノードを含むマスタ層と、各々が複数の中間ノードを含む 1 つまたは複数の中間層と、前記 1 つまたは複数のデバイスを含むデバイス層とを含み、前記第 1 のノードは前記マスタノードである、請求項 1 から 9 のいずれか一項に記載の方法。

【請求項 11】

前記マスタノードは主マスタ秘密鍵を有し、前記方法は、
シード秘密鍵を取得することと、
前記シード秘密鍵に基づいてジョイント秘密鍵のセットを生成することと、
前記ジョイント秘密鍵のセットを前記複数の中間ノードに送信することと
を含む、請求項 10 に記載の方法。

40

【請求項 12】

二次マスタ秘密鍵のセットを生成することを含み、ここで、前記二次マスタ秘密鍵のセット内の各秘密鍵は、前記主マスタ秘密鍵を前記ジョイント秘密鍵のセット内の各秘密鍵に追加することによって生成され、署名にリンクされた前記第 1 のノードの各公開鍵は、前記二次マスタ秘密鍵のセットからのそれぞれの秘密鍵に基づいて生成される、

請求項 11 に記載の方法。

【請求項 13】

前記主マスタ秘密鍵は、前記マスタノードによって証明された主マスタ公開鍵に対応する、請求項 11 または 12 に記載の方法。

【請求項 14】

50

前記第 1 のブロックチェーンランザクションは、少なくとも第 3 のノードの公開鍵および第 4 のノードの公開鍵にロックされた第 2 の出力を含む、請求項 1 に記載の方法。

【請求項 1 5】

前記第 1 のネットワークは、マスタノードを含むマスタ層と、各々が複数の中間ノードを含む 1 つまたは複数の中間層と、前記 1 つまたは複数のデバイスを含むデバイス層とを含み、前記第 1 のノードは中間ノードである、請求項 1 から 9 または請求項 1 4 のいずれか一項に記載の方法。

【請求項 1 6】

前記第 1 のノードは主秘密鍵を有し、前記方法は、
ジョイント秘密鍵のセットを取得することと、
前記主秘密鍵を前記ジョイント秘密鍵のセット内の各秘密鍵に追加することによって二次秘密鍵のセットを生成することを含み、ここで、署名にリンクされた前記第 1 のノードの各公開鍵は、前記二次秘密鍵のセットからのそれぞれの秘密鍵に基づいて生成される、
請求項 1 5 に記載の方法。

【請求項 1 7】

前記主秘密鍵は、前記マスタノードによって証明された主公開鍵に対応する、請求項 1 6 に記載の方法。

【請求項 1 8】

前記コマンドデータは、
プロトコル識別子、
ソフトウェアバージョン番号、
データタイプインジケータ、
前記第 1 のデバイスに関するデバイス証明書ロケーション、
前記第 1 のデバイスの現在のステータス、および
前記第 1 のデバイスについての前のステータス
のうちの 1 つまたは複数をさらに含む、請求項 1 から 1 7 のいずれか一項に記載の方法。

【請求項 1 9】

第 1 のネットワークの 1 つまたは複数のデバイスを制御す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、複数のブリッジノードと、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ノードでもあり、各デバイスはそれぞれのデバイス識別子を有し、前記コンピュータ実装方法は、前記ブリッジノードのうちの第 2 のノードによって実行され、

第 1 のブロックチェーンランザクションを取得することと、ここで、前記第 1 のブロックチェーンランザクションは、前記ブリッジノードのうちの第 1 のノードの第 1 の公開鍵にリンクされた署名を含む第 1 の入力と、コマンドデータを含む第 1 の出力とを含み、前記コマンドデータは、前記第 2 のノードによって制御される前記デバイスのうちの第 1 のデバイスのそれぞれの識別子と、前記第 1 のデバイスを制御するためのコマンドメッセージとを含み、

前記第 1 のデバイスにコマンドを送信することと、ここで、前記コマンドは前記コマンドメッセージに基づく、
を含む方法。

【請求項 2 0】

前記送信することは、前記署名が有効な署名であることを条件とする、請求項 1 9 に記載の方法。

【請求項 2 1】

前記コマンドは、オフチェーン接続を介して前記第 1 のデバイスに送信される、請求項 1 9 または 2 0 に記載の方法。

【請求項 2 2】

前記第 2 のノードの第 1 の公開鍵にリンクされた署名を含む第 2 の入力を含めることに

10

20

30

40

50

よって前記第 1 のブロックチェーントランザクションを更新することと、

前記更新された第 1 のブロックチェーントランザクションが、ブロックチェーンに含まれるために前記ブロックチェーンネットワークに送信されることを生じさせることとを含む、請求項 19 から 21 のいずれか一項に記載の方法。

【請求項 23】

前記生じさせることは、前記更新された第 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ことを含む、請求項 22 に記載の方法。

【請求項 24】

前記第 1 のデバイスからコマンド完了メッセージを受信することを含み、
前記更新することは、前記コマンド完了メッセージの受信を条件とする、
請求項 22 または 23 に記載の方法。

10

【請求項 25】

前記更新することは、前記第 2 のノードの第 2 の公開鍵にロックされた第 2 の出力を含めることを含む、請求項 22 から 24 のいずれか一項に記載の方法。

【請求項 26】

前記取得することは、前記第 1 のノードから前記第 1 のブロックチェーントランザクションを受信することを含む、請求項 19 から 25 のいずれか一項に記載の方法。

【請求項 27】

前記取得することは、ブロックチェーンから前記第 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を含む、請求項 19 から 26 のいずれか一項に記載の方法。

20

【請求項 28】

前記コマンドデータは、前記第 1 のノードの第 2 の公開鍵および前記第 2 のノードの第 2 の公開鍵に基づいて生成された第 1 の暗号鍵を使用して暗号化され、前記方法は、前記第 1 の暗号鍵を使用して前記コマンドデータを復号することを含む、請求項 19 から 27 のいずれか一項に記載の方法。

【請求項 29】

前記第 1 のネットワークは、マスタノードを含むマスタ層と、各々が複数の中間ノードを含む 1 つまたは複数の中間層と、前記 1 つまたは複数のデバイスを含むデバイス層とを含み、前記第 2 のノードは中間ノードである、請求項 19 から 28 のいずれか一項に記載の方法。

30

【請求項 30】

前記第 2 のノードは主秘密鍵を有し、前記方法は、
ジョイント秘密鍵のセットを取得することと、
前記主秘密鍵を前記ジョイント秘密鍵のセット内の各秘密鍵に追加することによって二次秘密鍵のセットを生成することと
を含む、請求項 29 に記載の方法。

【請求項 31】

前記第 2 のノードの前記第 1 の公開鍵は、対応する二次秘密鍵を有する公開鍵である、請求項 22 に従属する場合の請求項 29 に記載の方法。

【請求項 32】

コンピュータ機器であって、
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 18 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。

40

【請求項 33】

コンピュータ可読ストレージ上に具現化され、請求項 32 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 18 のいずれか一項に記載の方法を実行するように構成さ

50

れたコンピュータプログラム。

【請求項 3 4】

コンピュータ機器であって、
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 9 から 3 1 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 3 5】

コンピュータ可読ストレージ上に具現化され、請求項 3 4 に記載のコンピュータ機器上で実行されると、請求項 1 9 から 3 1 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】

【0 0 0 1】

本開示は、ブロックチェーントランザクションを使用してネットワークのデバイスを制御するための方法に関する。

【背景技術】

【0 0 0 2】

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（P 2 P）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。

【0 0 0 3】

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。

【0 0 0 4】

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（v a l i d a t e）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

10

20

30

40

50

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。

10

【0006】

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。

【0007】

本明細書で開示される一態様によれば、第1のネットワークの1つまたは複数のデバイスを制御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ノードでもあり、各デバイスはそれぞれのデバイス識別子を有し、方法は、ブリッジノードのうちの第1のブリッジノードによって実行され、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を含み、第1のブロックチェーントランザクションは、第1のノードの第1の公開鍵にリンクされた署名を含む第1の入力と、コマンドデータを含む第1の出力とを含み、コマンドデータは、ブリッジノードのうちの第2のブリッジノードによって制御されるデバイスのうちの第1のデバイスのそれぞれの識別子と、第1のデバイスを制御するためのコマンドメッセージとを含む方法が提供される。

20

30

【0008】

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つまたは複数または、ブロックチェーンネットワークのノードであり得る。

40

【0009】

要求ノード（第1のブリッジノード）は、ブロックチェーントランザクション（第1のトランザクション）を生成する。トランザクションは、要求ノードによって署名され、コマンドデータを含む出力を含む。コマンドデータは、制御されるべきデバイスの識別子と、コマンドを指定するコマンドメッセージとを含む。例えば、デバイスはスマート洗濯機

50

であり得、コマンドメッセージは洗濯サイクルを開始する命令であり得る。トランザクションは、ブロックチェーンに送信され得、および／または第2のノード（例えば、応答ノードまたは承認ノード）にオフチェーンで送信され得る。応答ノードは、指定された識別子を有するエンドデバイスを直接制御するブリッジノードである。承認ノードは、エンドデバイスを制御するコマンドの承認を与える権限を有するブリッジノードである。洗濯機の例を続けると、第2のノードは、洗濯機を制御するスマートホームハブであり得る。第1のノードは、例えば、ラップトップ、携帯電話などのユーザデバイスであり得る。

【0010】

第1のブロックチェーントランザクションは、部分的に完全なトランザクションであってもよい。この文脈における部分的に完全とは、トランザクションを完了するために追加の入力（例えば署名）が追加される必要があり得ることを意味する。いくつかの例では、部分的なトランザクションは、追加の入力がなければ無効であり得、これは、ブロックチェーンネットワークに送信されても、ネットワークによって拒否されることを意味する。

【0011】

本明細書に開示される別の態様によれば、第1のネットワークの1つまたは複数のデバイスを制御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、複数のブリッジノードと、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ノードでもあり、各デバイスはそれぞれのデバイス識別子を有し、方法は、ブリッジノードのうちの第2のブリッジノードによって実行され、第1のブロックチェーントランザクションを取得することと、ここで、第1のブロックチェーントランザクションは、ブリッジノードのうちの第1のブリッジノードの第1の公開鍵にリンクされた署名を含む第1の入力と、コマンドデータを含む第1の出力とを含み、コマンドデータは、第2のノードによって制御されるデバイスのうちの第1のデバイスのそれぞれの識別子と、第1のデバイスを制御するためのコマンドメッセージとを含み、第1のデバイスにコマンドを送信することと、ここで、コマンドはコマンドメッセージに基づく、を含む方法が提供される。

【0012】

応答ノードは、例えば、（第1のトランザクションがブロックチェーンに送信された場合）ブロックチェーン自体からまたは要求ノードから直接第1のトランザクションを取得する。次いで、応答ノードは、コマンドデータ内のデバイス識別子と、コマンドを指定するコマンドメッセージとを使用して、制御されるべきデバイスを識別する。次いで、応答ノードは、識別されたエンドデバイスにコマンドを送信する。言い換えれば、応答ノードは、第1のトランザクション内のコマンドデータに基づいてアクションを実行するようにエンドデバイスに命令する。

【0013】

好ましくは、応答ノードは、取得された第1のトランザクションに署名することによって第1のブロックチェーントランザクションを更新し、次いで、更新された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。更新されたトランザクションは、コマンドがエンドデバイスによって実行されていることの確認応答およびコマンド要求の記録として機能する。

【0014】

ブロックチェーントランザクションは、分け隔てなくグローバルネットワークにわたって伝播され得るデジタル署名されたメッセージのためのテンプレートである。ノードは、コマンド、ステータス更新、および他の関連アクションをトランザクションの出力（例えば、利用不可能な出力）に符号化することによって、ブロックチェーンをデータキャリアネットワークとして使用する。通信待ち時間を最小化するために、コマンドデータを符号化するトランザクションは、ブロックチェーンネットワークにブロードキャストされる前に、ピアツーピアで（すなわち、ノード間で直接）送信される。

【0015】

一緒に、第1のネットワークのブリッジノードは、ブロックチェーントランザクション

10

20

30

40

50

を使用して分散型ＩｏＴ通信プロトコルを動作させる。ブロックチェーンプロトコルは、大容量および低低数量のマイクロトランザクションスループットならびにスケーラブルなネットワークインフラストラクチャを可能にするため、デバイスが、最小限のコストで通信しながら、確実にかつグローバルスケールで接続されることを可能にする。マルチレベル制御階層とブロックチェーンベースの通信プロトコルとを組み合わせることで、要求およびＩｏＴ通信プロトコルは、低料金マイクロトランザクションを使用した大規模Ｐ２Ｐ通信、１つのプラットフォームへの価値転送および制御の統合、ＩｏＴネットワークデバイスへの低い参入障壁、ＩｏＴ通信データの安全なタイムスタンプ付きストレージ、ならびに監査および性能監視のためにアクセス可能なＩｏＴメタデータを提供する。

【図面の簡単な説明】

【００１６】

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】

10

20

30

40

50

【0017】

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。

【0018】

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。

【0019】

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104F、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。

【0020】

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トを含み、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

10

20

30

40

50

要はない。したがって、本明細書における「先行する (preceding)」は、ポイント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152 i、152 j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152 i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。

【0021】

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

10

20

【0022】

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。

【0023】

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。

30

40

【0024】

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152 j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152 j内の暗号

50

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。

10

【0025】

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。

20

【0026】

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、「ブルーフオブワーク」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。

30

40

【0027】

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

50

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151 $n-1$ を指し示す新しいブロック151 n 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。

10

【0028】

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151 n 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ブ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。

20

【0029】

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151 n 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151 n 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。

30

【0030】

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。

40

【0031】

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。

50

【0032】

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。純粹に例示として、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。

10

【0033】

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。

20

30

【0034】

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。

40

【0035】

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。

50

【0036】

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）。

【0037】

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、前述したようなブ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、ブ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。

【0038】

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

10

20

30

40

50

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。

【0039】

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106全体にわたって伝搬されることを意味する。

【0040】

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。

【0041】

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。

【0042】

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。

【0043】

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。

10

20

30

40

50

【0044】

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「Tx₁」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「Tx₀」とラベル付けされている。Tx₀およびTx₁は、単なる任意のラベルである。それらは、Tx₀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、Tx₁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。Tx₁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。

10

【0045】

先行するトランザクションTx₀は、アリスが新しいトランザクションTx₁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、Tx₀およびTx₁を作成してネットワーク102と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、Tx₀をTx₁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ト（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。

20

30

【0046】

先行するトランザクションTx₀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UTXO₀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。

40

【0047】

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

50

(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。

【0048】

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 PA` を必要とするロックスクリプト [`Checksig PA`] を含む。

【0049】

[`Checksig PA`] は、アリスの公開鍵－秘密鍵ペアからの公開鍵 PA 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 PA>` 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。

【0050】

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 PA><PA> || [Checksig PA]`

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 PA を使用して、 Tx_1 の入力内のロックスクリプトが、データの予想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)。

【0051】

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。

【0052】

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 104 M 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

10

20

30

40

50

154のプールにそれを追加することを意味する。それがフォーディングノード104Fである場合、トランザクション T_{x1} 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 T_{x1} がネットワーク全体に伝搬されるようにする。 T_{x1} 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、 T_{x0} からのUTXO₀を使用済みと定義する。 T_{x1} 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、 T_{x1} 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 T_{x0} 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ることが重要である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。

10

【0053】

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、 T_{x0} 内のUTXO₀において定義された額は、 T_{x1} 内の複数のUTXO間で分割され得る。したがって、アリスが、UTXO₀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、 T_{x1} 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。

20

【0054】

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、 T_{x0} 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 Protokol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、UTXO₀へのポインタが T_{x1} への唯一の入力であり、 T_{x1} は唯一の出力UTXO₁を有するとする。UTXO₀で指定されているデジタル資産の額がUTXO₁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い。

30

40

【0055】

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。

【0056】

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

50

。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すべての値と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。

【0057】

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 PA]`と書いて`[Checksig PA] = 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。

【0058】

署名PA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）。

【0059】

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。

【0060】

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、102b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

10

20

30

40

50

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、サイドチャンネル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。

【0061】

サイドチャンネル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、サイドチャンネル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ンネル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、サイドチャンネル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ンネル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。

【0062】

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 152_{m-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 152_m (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 の入力内のポイントに基づいて、 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。

【0063】

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、Sc r i

10

20

30

40

50

p t) にしたがって、スタック 4 0 3 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。

【0064】

スクリプトと一緒に実行することによって、スクリプトエンジン 4 0 2 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 4 0 2 は、この決定の結果をプロトコルエンジン 4 0 1 に返す。スクリプトエンジン 4 0 2 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。

【0065】

出力ベースのモデルでは、スクリプトエンジン 4 0 2 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、 $T x_m$ 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 $T x_{m-1}$ 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 4 0 1 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 4 0 1 は、スクリプトエンジン 4 0 2 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 $T x_m$ を妥当性確認する。プロトコルエンジン 4 0 1 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 4 0 4 に出力する。 $T x_m$ 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 4 0 4 は、 $T x_m$ 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 4 0 5 M およびフォワーディングモジュール 4 0 5 F 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 4 0 5 M が、ブロック 1 5 1 にマイニングするために $T x_m$ をノードのそれぞれのプール 1 5 4 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 4 0 5 F が $T x_m$ を P 2 P ネットワーク 1 0 6 内の別のノード 1 0 4 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 4 0 4 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 4 0 4 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 1 0 4 M 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。

【0066】

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 1 0 4 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）。

【0067】

モノのインターネット

I o T 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、I o T 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

10

20

30

40

50

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。

【0068】

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

10

【0069】

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。

20

【0070】

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。

30

【0071】

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

40

【0072】

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ビギン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ー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

50

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。

【0073】

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105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。

【0074】

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。

【0075】

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105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。

【0076】

I o T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。

【0077】

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、I o T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。

【0078】

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 I o T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。

【0079】

図6は、例示的なI o Tネットワーク501トポロジーを示す。I o T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。I o T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。

【0080】

図6の例示的なI o T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。

【0081】

I o T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

10

20

30

40

50

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。

【0082】

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105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の直接コマンド下であり、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。

【0083】

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105を動作させ得る。

【0084】

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、クライアントー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、サーバノードおよびスレーブノードは、SPVウォレット105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とプロ

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。

【0085】

要求および応答プロトコル

本開示の実施形態は、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。

10

【0086】

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のランザクションTx₁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。

【0087】

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のランザクションTx₁を送信し得る。第1のランザクションTx₁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のランザクションTx₁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のランザクションTx₁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のランザクションTx₁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。

20

【0088】

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のランザクションTx₁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のランザクションTx₁が有効な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のランザクションTx₁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。

30

【0089】

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、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）を使用してランザクションを交換し得る。

40

【0090】

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のランザクションTx₁を取得し得、例えば、第1のランザクションTx₁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

50

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。

【0091】

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ランザクションTxを介して送信されてもよい。

【0092】

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、リクエスト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、ランザクションの展性（malleability）は、この通信方法を可能にする。

【0093】

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。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）。

【0094】

第1のノードから第2のノードに送信された第1のランザクションTx₁は、第2の出力なしで送信され得る。すなわち、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）。部分的なランザクションを完了するために、第2は、第1の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。完了したランザクション（これらの実施形態ではコマンド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

10

20

30

40

50

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。

【0095】

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Tx₁（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Tx₁（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。S I G H A S H__S I N G L E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PK₀を有するノードは、公開鍵PK₁を有するノードに命令を送信する。命令は、S I G H A S H__S I N G L E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、O P__R E T U R N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、PK₁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、PK₁を有する第2のノードは、S I G H A S H__A L L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。

10

【0096】

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Tx₁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。

20

【0097】

第1のトランザクションTx₁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Tx₁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。

30

【0098】

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Tx₁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Tx₁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Tx₁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。

40

【0099】

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Tx₁（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Tx₁（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。PK₀を有する第1のノードが、PK₁

50

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】

$\langle \text{Sig}_{PK_0} \rangle$

のためのS I G H A S Hフラグは、S I G H A S H _ J A N U S A L C A N P A Yに設定され、コマンドデータと共にO P _ R E T U R N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、S I G H A S H _ A L Lを使用してトランザクション出力に署名する。

【0100】

S I G H A S H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S I G H A S H _ A L L（S c r i p t S i g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。

【0101】

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d e v i c e _ I D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。

【0102】

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T x 1（図10a参照）を送信する。トランザクションに含まれるI o Tメッセージは、コマンドおよびd e v i c e _ I Dを有するターゲットデバイスを指定する。

【0103】

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I o T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。

【0104】

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、B l u e t o o t h、I P - t o - I P）を介してコマンドメッセージ（「M s g」）をデバイス（d e v i c e _ I D）に送信する。

【0105】

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（d e v i c e _ I D）は、コマンド完了または確認応答メッセージ（「a c k」）を第2の制御デバイス（ PK_{slave} ）に送り返す。

【0106】

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。

【0107】

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

10

20

30

40

50

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。
【0108】

いくつかの実施形態では、第1のトランザクション T_{x1} 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 T_{x1} （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 503 a 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 503 b）からの承認を要求するサーバノード 503 b であり得る。第1のトランザクション T_{x1} 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 T_{x1} 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 T_{x1} 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 T_{x2} 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 T_{x1} 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 T_{x2} 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 T_{x2} （承認）の入力に署名し、そのトランザクション T_{x2} をブロックチェーンネットワーク 160 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 T_{x2} は、コマンド要求トランザクション T_{x1} 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。

【0109】

図 11 a および図 11 b 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 T_{x1} 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 T_{x2} 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 T_{x1} は、第1のノードによって署名された入力 1101 a と、コマンドデータを含む第1の出力 1102 a 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 1102 b とを含む。第2のトランザクション T_{x2} は、第3および第4のノードによって署名された入力 1101 b と、第1の出力 1102 a と、第2のノードの公開鍵にロックされた第2の出力 1102 c 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 multisig) 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。

10

20

30

40

50

【0110】

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。

10

【0111】

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。

20

【0112】

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。

30

【0113】

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。

40

【0114】

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

50

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。

【0115】

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。

【0116】

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HTTP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ト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。

【0117】

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。

【0118】

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。

【0119】

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503a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。

【0120】

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。

【0121】

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。

【0122】

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。

【0123】

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】

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

【0124】

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 ECI ES）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。

【0125】

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。

【0126】

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】

$$s_{ij}^M$$

を生成し、ここで、以下である：

【数4】

$$sk_{ij}^M = sk_{ij}^{joint} + sk_{Master}$$

【0127】

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

10

20

30

40

50

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】

$$PK_{ij}^M$$

を導出することができ、ここで以下である：

【数6】

$$PK_{ij}^M = PK_{ij}^{joint} + PK_{Master}.$$

10

【0128】

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。

【0129】

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。

20

【0130】

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。

【0131】

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ない、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。

30

【0132】

40

【表 1】

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	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	証明書TX内の失効UTXOロケーション
		VOUT	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

10

20

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、公開鍵暗号およびブルーフオブワークを使用して保護される。

【0133】

30

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。

【0134】

要求および応答メッセージングシステムー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。

40

【0135】

オフライントランザクション伝搬ー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。

【0136】

ノードとエンドデバイスとの間の直接通信ートランザクションコマンドペイロードに符

50

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。

【0137】

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一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。

【0138】

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。

【0139】

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。

【0140】

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。

【0141】

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。

【0142】

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）。

【0143】

許可プロトコル

本開示はまた、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。

【0144】

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

10

20

30

40

50

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。

【0145】

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ト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、リクエスト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内のリクエストに固有である。リクエスト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、リクエスト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。

10

20

【0146】

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、リクエストは、証明書を使用して、リクエストがネットワーク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、リクエストは、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。

30

【0147】

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。

【0148】

リクエストがネットワーク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。

【0149】

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。

40

【0150】

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

50

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）。

【0151】

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。

【0152】

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。

【0153】

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。

【0154】

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。

【0155】

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。

【0156】

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

10

20

30

40

50

ジョン識別子 $T \times I D$ 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） $S H A 2 5 6$ ハッシュを計算することによって生成され得る。 $S H A 2 5 6$ 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トに送信し得る。これにより、リクエストは、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、リクエストは、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。

【0157】

リクエストがノード（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、リクエストは、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、リクエストは、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、 $P 2 P K H$ 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、リクエストは、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。

【0158】

リクエスト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 $B l u e t o o t h$ 、 $W i - F i$ 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。

【0159】

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、 $I P$ アドレス）を含み得る。リクエストは、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。

【0160】

登録機関は、リクエスト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トと通信し、リクエストがネットワーク 501 に参加する許可を与えられたかどうかを検証し得る。

【0161】

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。

【0162】

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

10

20

30

40

50

となく、証明書をリクエストに発行し得る。他の例では、リクエストは最初に登録機関に要求を送信し得る。要求は、リクエストの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、オープン、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。

【0163】

場合によっては、リクエスト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、リクエストが、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）を参照する入力を含む。入力、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。

【0164】

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。

【0165】

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。

上述したように、証明書は、リクエストの一意の識別子を含む。証明書は、リクエスト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：

10

20

30

40

50

【表 2】

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、オープン、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)。

10

20

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。

30

【0166】

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。リクエストはノード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、リクエストはマスタノード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トの両方である場合、マスタノード503aは、自身の証明書に自己署名する。

【0167】

40

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。

【0168】

50

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。

【0169】

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。

【0170】

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。

【0171】

ステップ4：新しいノードは、トランザクションを発見し、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。

【0172】

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。

【0173】

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。

【0174】

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。

【0175】

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暗号化される。

【0176】

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。

【0177】

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。

【0178】

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

10

20

30

40

50

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S I G H A S H _ S I N G L E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、S I G H A S H _ A L L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にブロードキャストされる例示的なコマンドー確認応答トランザクションを示す。

【0179】

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。

【0180】

ステートメント1．第1のネットワークの1つまたは複数のデバイスを制御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ノードでもあり、各デバイスはそれぞれのデバイス識別子を有し、方法は、ブリッジノードのうちの第1のブリッジノードによって実行され、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、第1のノードの第1の公開鍵にリンクされた署名を含む第1の入力と、コマンドデータを含む第1の出力とを含み、コマンドデータは、ブリッジノードのうちの第2のブリッジノードによって制御されるデバイスのうちの第1のデバイスのそれぞれの識別子と、第1のデバイスを制御するためのコマンドメッセージとを含む。

【0181】

第1のブロックチェーントランザクションは、部分的に完全なブロックチェーントランザクションであり得る。この文脈における部分的に完全とは、それを完了するために追加の入力／署名が追加される必要があることを意味する。第1のブロックチェーントランザクションは、第2のノードにオフチェーンで送信される。実施形態では、第1のブロックチェーントランザクションは、第1のノードからブロックチェーンネットワークに送信されない。

【0182】

ステートメント2．第1のブロックチェーントランザクションは、第2のノードの第1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。

10

【0183】

第2の出力は、ロック解除されるために、その公開鍵にリンクされた署名が、第2の出力を参照する後のトランザクションの入力において提供されなければならないという意味で、第2のノードの第1の公開鍵にロックされる。

【0184】

ステートメント3．第1のブロックチェーントランザクションを第2のノードに送信することを含む、ステートメント1またはステートメント2に記載の方法。

【0185】

第1のブロックチェーントランザクションは、第2のノードに直接送信され得るか、またはネットワーク501の1つまたは複数の中間ノードを介して送信され得る。部分的なトランザクションは、より広いネットワークコンセンサスルールにしたがって有効でない場合がある。

20

【0186】

ステートメント4．ブロックチェーンは前のブロックチェーントランザクションを含み、前のトランザクションは、ブリッジノードのうちの第3のブリッジノードの第1の公開鍵にリンクされた署名を含む第1の入力と、第1のデバイスのそれぞれの識別子および第1のデバイス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第1の出力と、少なくとも第1のノードの第1の公開鍵およびブリッジノードのうちの第4のブリッジノードの第1の公開鍵にロックされた第2の出力とを含み、第1のブロックチェーントランザクションの第1の入力は、第4のノードの第1の公開鍵にリンクされた署名を含む、ステートメント2に記載の方法。

30

【0187】

第1のネットワークは階層構造で構成され得、それによって、第1のノードは、異なるノード（第3のノード）からのコマンドを承認するように求められている。前のトランザクションは、ブロックチェーンのUTXOセットにおいて見つけることができる。

【0188】

ステートメント5．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を含む、ステートメント4に記載の方法。

【0189】

ステートメント6．コマンドメッセージは、第2のノードに、制御命令を第1のデバイスに送信させるように構成され、制御命令はコマンドメッセージに基づく、ステートメント1から5のいずれかに記載の方法。

40

【0190】

ステートメント7．コマンドデータは、暗号化されたコマンドデータである、ステートメント1から6のいずれかに記載の方法。

【0191】

ステートメント8．コマンドデータは、第1のノードの第2の公開鍵および第2のノードの第2の公開鍵に基づいて生成された第1の暗号鍵を使用して暗号化される、ステートメント7に記載の方法。

50

【0192】

第1および第2のノードの第2の公開鍵は、それぞれ第1および第2の当事者の第1の公開鍵と同じであり得る。代替的に、異なる公開鍵が使用し得る。

【0193】

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を含み、第1のブロックチェーンランザクションの第1の出力は暗号化データを含み、暗号化データは、第1のノードの第3の公開鍵およびマスタノードの第1の公開鍵に基づいて生成された第2の暗号鍵を使用して暗号化された第1の暗号鍵を含む、ステートメント8に記載の方法。

【0194】

マスタノードは、階層の最上部に位置し、他のすべてのノードを制御する。ここで、制御とは、他のノードがマスタノードから送信された命令を実行するように構成されることを意味する。制御のメカニズムはブロックチェーンランザクションである。

【0195】

第1のノードの第3の公開鍵は、第1のノードの第1および／または第2の公開鍵と同じであってもよい。代替的に、第3の公開鍵は、異なる公開鍵であってもよい。

【0196】

ステートメント10．第1のネットワークは、マスタノードを含むマスタ層と、各々が複数の中間ノードを含む1つまたは複数の中間層と、1つまたは複数のデバイスを含むデバイス層とを含み、第1のノードはマスタノードである、ステートメント1から9のいずれかに記載の方法。

【0197】

ステートメント11．マスタノードは主マスタ秘密鍵を有し、方法は以下を含む：

シード秘密鍵を取得すること、

シード秘密鍵に基づいてジョイント秘密鍵のセットを生成すること、および

ジョイント秘密鍵のセットを複数の中間ノードに送信すること。

【0198】

ステートメント12．ステートメント11に記載の方法であって、以下を含む：

二次マスタ秘密鍵のセットを生成すること、ここで、二次マスタ秘密鍵のセット内の各秘密鍵は、主マスタ秘密鍵をジョイント秘密鍵のセット内の各秘密鍵に追加することによって生成され、署名にリンクされた第1のノードの各公開鍵は、二次マスタ秘密鍵のセットからのそれぞれの秘密鍵に基づいて生成される。

【0199】

ステートメント13．主マスタ秘密鍵は、マスタノードによって証明された主マスタ公開鍵に対応する、ステートメント11またはステートメント12に記載の方法。

【0200】

ステートメント14．第1のブロックチェーンランザクションは、少なくとも第3のノードの公開鍵および第4のノード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。

【0201】

第1のネットワークは階層構造で構成され得、それによって、第1のノードは、第1のランザクションを第2のノードに送信するために、第3および第4のノードから承認を得なければならない。

【0202】

ステートメント15．第1のネットワークは、マスタノードを含むマスタ層と、各々が複数の中間ノードを含む1つまたは複数の中間層と、1つまたは複数のデバイスを含むデバイス層とを含み、第1のノードは中間ノードである、ステートメント1から9またはステートメント14のいずれかに記載の方法。

【0203】

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のノードは主秘密鍵を有し、方法は以下を含む：

ジョイント秘密鍵のセットを取得すること、および

主秘密鍵をジョイント秘密鍵のセット内の各秘密鍵に追加することによって二次秘密鍵のセットを生成すること、ここで、署名にリンクされた第1のノードの各公開鍵は、二次秘密鍵のセットからのそれぞれの秘密鍵に基づいて生成される。

【0204】

ステートメント17. 主秘密鍵は、マスタノードによって証明された主公開鍵に対応する、ステートメント16に記載の方法。

【0205】

ステートメント18. ステートメント1から17のいずれかに記載の方法であって、コマンドデータは、以下のうちの1つまたは複数をさらに含む：

プロトコル識別子、
ソフトウェアバージョン番号、
データタイプインジケータ、
第1のデバイスに関するデバイス証明書ロケーション、
第1のデバイスの現在のステータス、および
第1のデバイスについての前のステータス。

【0206】

ステートメント19. 第1のネットワークの1つまたは複数のデバイスを制御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、複数のブリッジノードと、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ノードでもあり、各デバイスはそれぞれのデバイス識別子を有し、方法は、ブリッジノードのうちの第2のブリッジノードによって実行され、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取得すること、ここで、第1のブロックチェーントランザクションは、ブリッジノードのうちの第1のブリッジノードの第1の公開鍵にリンクされた署名を含む第1の入力と、コマンドデータを含む第1の出力とを含み、コマンドデータは、第2のノードによって制御されるデバイスのうちの第1のデバイスのそれぞれの識別子と、第1のデバイスを制御するためのコマンドメッセージとを含む、および

第1のデバイスにコマンドを送信すること、ここで、コマンドはコマンドメッセージに基づく。

【0207】

ステートメント20. 上記送信することは、署名が有効な署名であることを条件とする、ステートメント19に記載の方法。

【0208】

ステートメント21. コマンドは、オフチェーン接続を介して第1のデバイスに送信される、ステートメント19またはステートメント20に記載の方法。

【0209】

例えば、オフチェーン接続は、有線接続、ワイヤレス接続（例えば、BluetoothまたはWi-Fi）などのうちの1つまたは複数であり得る。

【0210】

ステートメント22. ステートメント19から21のいずれかに記載の方法であって、以下を含む：

第2のノードの第1の公開鍵にリンクされた署名を含む第2の入力を含めることによって第1のブロックチェーントランザクションを更新すること、および

更新された第1のブロックチェーントランザクションが、ブロックチェーンに含まれるためにブロックチェーンネットワークに送信されることを生じさせること。

【0211】

ステートメント23. 上記生じさせることは、更新された第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を含む、ステートメント2

10

20

30

40

50

2に記載の方法。

【0212】

ステートメント24．ステートメント22またはステートメント23に記載の方法であって、以下を含む：

第1のデバイスからコマンド完了メッセージを受信すること、および
上記更新することは、コマンド完了メッセージの受信を条件とする。

【0213】

ステートメント25．上記更新することは、第2のノードの第2の公開鍵にロックされた第2の出力を含めることを含む、ステートメント22から24のいずれかに記載の方法。

【0214】

ステートメント26．上記取得することは、第1のノードから第1のブロックチェーントランザクションを受信することを含む、ステートメント19から25のいずれかに記載の方法。

【0215】

ステートメント27．上記取得することは、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を含む、ステートメント19から26のいずれかに記載の方法。

【0216】

ステートメント28．コマンドデータは、第1のノードの第2の公開鍵および第2のノードの第2の公開鍵に基づいて生成された第1の暗号鍵を使用して暗号化され、方法は、第1の暗号鍵を使用してコマンドデータを復号することを含む、ステートメント19から27のいずれかに記載の方法。

【0217】

ステートメント29．第1のネットワークは、マスタノードを含むマスタ層と、各々が複数の中間ノードを含む1つまたは複数の中間層と、1つまたは複数のデバイスを含むデバイス層とを含み、第2のノードは中間ノードである、ステートメント19から28のいずれかに記載の方法。

【0218】

ステートメント30．ステートメント29に記載の方法であって、第2のノードは主秘密鍵を有し、方法は以下を含む：

ジョイント秘密鍵のセットを取得すること、および

主秘密鍵をジョイント秘密鍵のセット内の各秘密鍵に追加することによって二次秘密鍵のセットを生成すること。

【0219】

ステートメント31．第2のノードの第1の公開鍵は、対応する二次秘密鍵を有する公開鍵である、ステートメント22に従属する場合のステートメント29に記載の方法。

【0220】

ステートメント32．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18のいずれかに記載の方法を実行するように構成される。

【0221】

ステートメント33．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32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18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。

【0222】

ステートメント34．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

10

20

30

40

50

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 19 から 31 のいずれかに記載の方法を実行するように構成される。

【0223】

ステートメント 35 . 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 34 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 19 から 31 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。

【0224】

ステートメント 36 . ブロックチェーン内に含めるための第 1 のトランザクションであって、第 1 のトランザクションは以下を含む：

第 1 のノードの公開鍵にリンクされた署名を含む第 1 の入力、および

コマンドデータを含む第 1 の出力、ここで、コマンドデータは、第 2 のノードによって制御されるデバイスのそれぞれの識別子と、デバイスを制御するためのコマンドメッセージとを含む。

【0225】

ステートメント 37 . ブロックチェーン内に含めるための更新された第 1 のトランザクションであって、更新された第 1 のトランザクションは以下を含む：

第 1 のノードの第 1 の公開鍵にリンクされた署名を含む第 1 の入力、

第 2 のノードの第 1 の公開鍵にリンクされた署名を含む第 2 の入力、

コマンドデータを含む第 1 の出力、ここで、コマンドデータは、第 2 のノードによって制御される第 1 のデバイスのそれぞれの識別子と、第 1 のデバイスを制御するためのコマンドメッセージとを含み、および

第 2 のノードの第 1 の公開鍵にロックされた第 2 の出力。

【0226】

ステートメント 38 . ブロックチェーン内に含めるためのコマンド要求トランザクションであって、コマンド要求トランザクションは以下を含む：

第 1 のノードの公開鍵にリンクされた署名を含む第 1 の入力、

コマンドデータを含む第 1 の出力、ここで、コマンドデータは、第 2 のノードによって制御されるデバイスのそれぞれの識別子と、デバイスを制御するためのコマンドメッセージとを含み、および

少なくとも第 3 のノードの公開鍵および第 4 のノードの公開鍵にロックされた第 2 の出力。

【0227】

ステートメント 39 . ブロックチェーン内に含めるためのコマンド承認トランザクションであって、コマンド承認トランザクションは以下を含む：

第 3 のノードの公開鍵にリンクされた署名および第 4 のノードの公開鍵にリンクされた署名を含む第 1 の入力、ここで、第 1 の入力は、ステートメント 37 に記載のコマンド要求トランザクションの第 2 の出力を参照し、

コマンドデータを含む第 1 の出力、ここで、コマンドデータは、第 2 のノードによって制御されるデバイスのそれぞれの識別子と、デバイスを制御するためのコマンドメッセージとを含み、および

第 2 のノードの公開鍵にロックされた第 2 の出力。

【0228】

ステートメント 40 . ステートメント 36 に記載の第 1 のトランザクションを記憶したコンピュータ可読記憶媒体。

【0229】

ステートメント 41 . ステートメント 37 に記載された更新された第 1 のトランザクションを記憶したコンピュータ可読記憶媒体。

【0230】

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トのアクションを含む方法が提供され得る。

10

20

30

40

50

【 0 2 3 1 】

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トのコンピュータ機器を備えるシステムが提供され得る。

【 0 2 3 2 】

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ランザクションを含むランザクションのセットが提供され得る。

【 0 2 3 3 】

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。

10

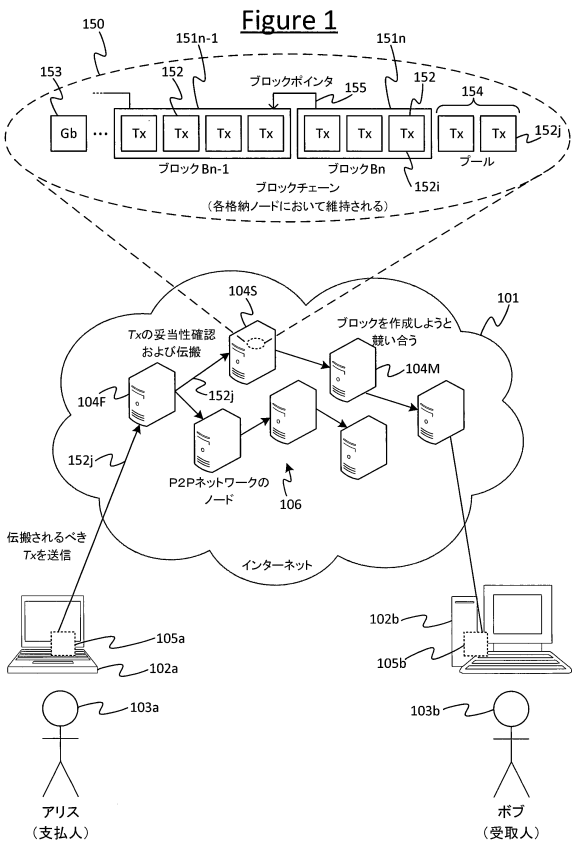
20

30

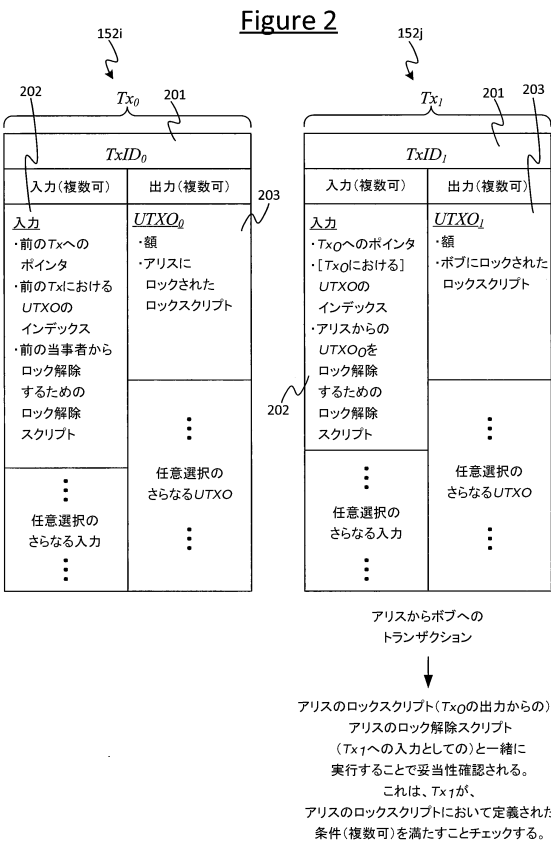
40

50

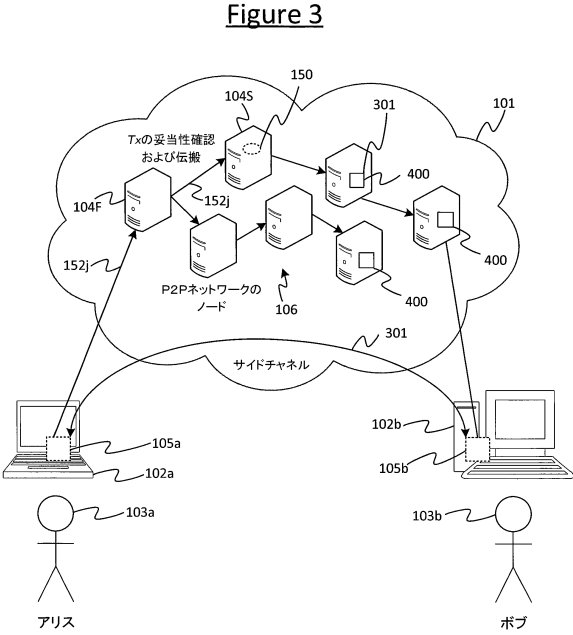
【図面】
【図 1】



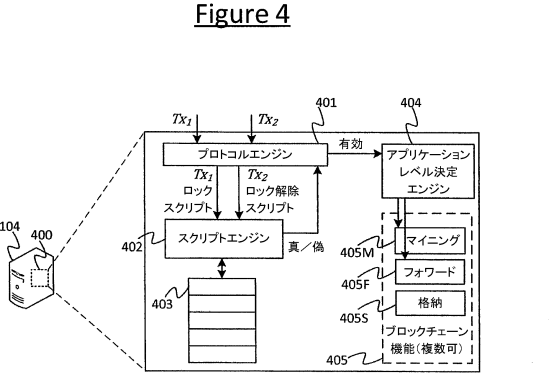
【図 2】



【図 3】



【図 4】



10

20

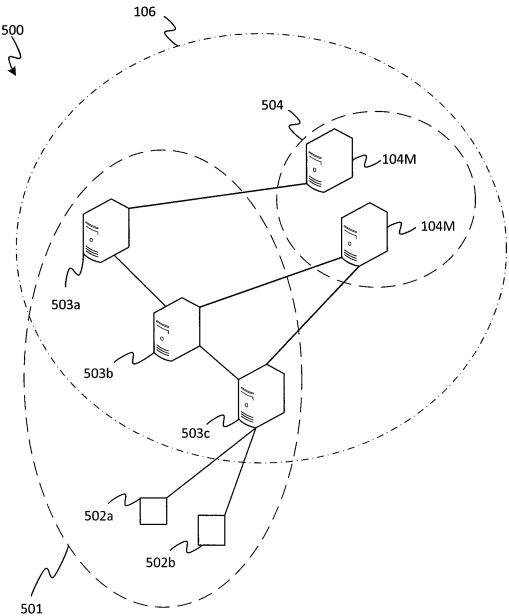
30

40

50

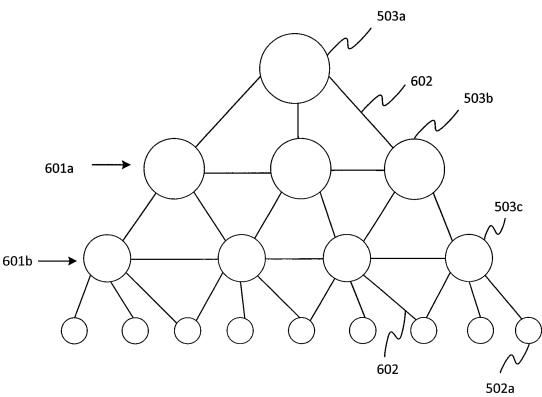
【図 5】

Figure 5



【図 6】

Figure 6



10

20

【図 7 a】

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 7 b】

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

30

40

50

【図 8 a】

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 + δ	OP_DUP OP_HASH160 <H ₁₆₀ (PK ₁)> OP_EQUALVERIFY OP_CHECKSIG

801a802b802a

【図 8 b】

Figure 8b

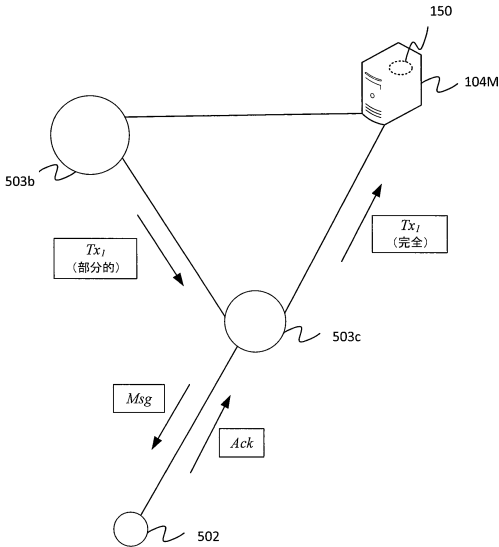
Tx ₁ (完全)			
入力		出力	
値		値	
x	<Sig _{PK0} > <PK ₀ >	0	OP_FALSE OP_RETURN 0x4d494f54 <Command data>
> δ	<Sig _{PK1} > <PK ₁ >	x	OP_DUP OP_HASH160 <H ₁₆₀ (PK ₁)> OP_EQUALVERIFY OP_CHECKSIG

801b

10

【図 9】

Figure 9



【図 1 0 a】

Figure 10a

Tx ₁ (部分的)			
入力		出力	
値		値	
x	<Sig _{PKserv} > <PK _{serv} >	0	OP_FALSE OP_RETURN 0x4d494f54 <Command data>
		x + δ	OP_DUP OP_HASH160 <H ₁₆₀ (PK _{stave})> OP_EQUALVERIFY OP_CHECKSIG

20

30

40

50

【図 1 0 b】

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>
> δ	<Sig _{PKslave} > <PK _{slave} >	x	OP_DUP OP_HASH160 <H ₁₆₀ (PK _{slave})> OP_EQUALVERIFY OP_CHECKSIG

【図 1 1 a】

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

10

【図 1 1 b】

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 1 2 a】

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 + δ	OP_DUP OP_HASH160 <H ₁₆₀ (PK _{slave_0})> OP_EQUALVERIFY OP_CHECKSIG

20

30

40

50

【図 1 2 b】

Figure 12b

Tx _i (完全)			
入力		出力	
値		値	
x	< Sig _{PK_{serv}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 >
> δ	< Sig _{PK_{slave}1} > < PK _{slave} 1 >	x + δ	OP_DUP OP_HASH160 < H ₁₆₀ (PK _{slave} 0) > OP_EQUALVERIFY OP_CHECKSIG

【図 1 3】

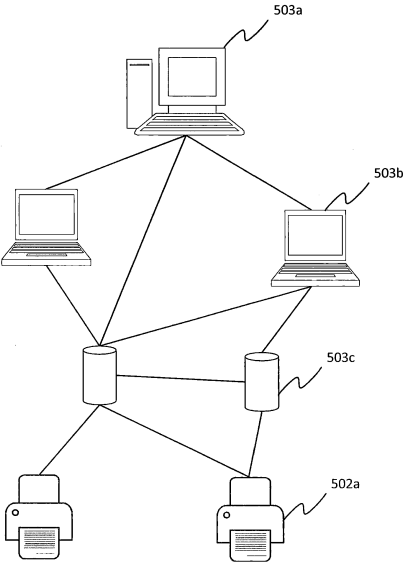
Figure 13

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

10

【図 1 4】

Figure 14



【図 1 5 a】

Figure 15a

Tx _i (証明書)			
入力		出力	
値		値	
x _i	< Sig _{PK_{admin}} > < PK _{admin} >	0	OP_FALSE OP_RETURN 0x4d494f54 < Certificate data >
		y _i	OP_DUP OP_HASH160 < H ₁₆₀ (PK _{admin}) > OP_EQUALVERIFY OP_CHECKSIG

20

30

40

50

【図 1 5 b】

Figure 15b

Tx _j (部分的)			
入力		出力	
値		値	
x ₂	< SigPK _{laptop_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 1 5 c】

Figure 15c

Tx _j (コントローラ)			
入力		出力	
値		値	
x ₂	< SigPK _{laptop_0} > < PK _{laptop_0} >	0	OP_FALSE OP_RETURN 0x4d494f54 < Encrypted Command data >
x ₃	< SigPK _{controller_0} > < PK _{controller_0} >	y ₂	OP_DUP OP_HASH160 < H ₁₆₀ (PK _{controller_0}) > OP_EQUALVERIFY OP_CHECKSIG

10

【図 1 6 a】

Figure 16a

Tx _j (証明書)			
入力		出力	
値		値	
x _j	< SigPK _{reg} > < 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 1 6 b】

Figure 16b

出力	
値	
0	OP_FALSE OP_RETURN OP_PUSHDATA1 < payload length > 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

20

30

40

50

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 塩澤 如正
- (56)参考文献 国際公開第2019/045589 (WO, A1)
 国際公開第2018/189657 (WO, A1)
- (58)調査した分野 (Int.Cl., DB名)
 H04L 9/00- 9/40