(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7656001号
(P7656001)

(45)発行日　令和7年4月2日(2025.4.2)

(24)登録日　令和7年3月25日(2025.3.25)

(51)国際特許分類

Ｇ０６Ｑ	*50/10*	*(2012.01)*	Ｇ０６Ｑ	50/10
Ｇ０６Ｑ	*20/36*	*(2012.01)*	Ｇ０６Ｑ	20/36
Ｈ０４Ｌ	*9/32*	*(2006.01)*	Ｈ０４Ｌ	9/32　２００Ｚ

請求項の数　15　外国語出願　（全34頁）

(21)出願番号　特願2023-173391(P2023-173391)
(22)出願日　令和5年10月5日(2023.10.5)
(62)分割の表示　特願2022-48598(P2022-48598)の分割
　原出願日　平成29年4月28日(2017.4.28)
(65)公開番号　特開2023-175927(P2023-175927A)
(43)公開日　令和5年12月12日(2023.12.12)
　審査請求日　令和5年10月5日(2023.10.5)
(31)優先権主張番号　1607476.7
(32)優先日　平成28年4月29日(2016.4.29)
(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　ライト，クレイグ スティーヴン
イギリス国 シーエフ１０ ２エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス ７ス フロア アーカート－ダイクス アンド ロード エルエルピー

最終頁に続く

(54)【発明の名称】　ブロックチェーンＩｏＴ装置のためのオペレーティングシステム

(57)【特許請求の範囲】

【請求項１】

コンピュータにより実装される制御システムであって、前記システムは、

ネットワークと通信するために構成された第１装置であって、前記第１装置は、前記第１装置に関連付けられたIPアドレスと公開－秘密鍵暗号鍵ペアを有する、第１装置と、

ブロックチェーンネットワークの状態を監視し、及び／又は前記ブロックチェーンネットワークへブロックチェーントランザクションを送信するよう構成されるソフトウェアにより実装される制御コンポーネントと、

前記制御コンポーネントに入力信号を送信するよう構成される１つ以上のクライアント装置であって、前記入力信号は、前記第１装置の制御に影響を与えるよう構成される、１つ以上のクライアント装置と、

前記第１装置の機能を制御するために前記制御コンポーネントによる実行のために構成された命令セットであって、前記命令セットは、前記第１装置と別個の場所に格納されたレポジトリに格納され、前記命令セットは、前記制御コンポーネントによる前記レポジトリからのダウンロード及びインストールのためにアクセスされ、前記レポジトリの場所は、ブロックチェーントランザクションの中で提供されるメタデータを用いて指示され又は提供される、命令セットと、

を含むシステム。

【請求項２】

前記１つ以上のクライアント装置は、無線周波数認識装置（ＲＦＩＤ）である、請求項

１に記載のシステム。
【請求項３】
前記第１装置は、前記１つ以上のクライアント装置が前記第１装置の近くにあることに基づき制御される、請求項１又は２に記載のシステム。
【請求項４】
ｉ）前記レポジトリは、分散ハッシュテーブル（ＤＨＴ）である、
ｉｉ）前記命令セットは、前記第１装置の動作を調整する、制御する、及び／又は影響を与えるために読み出される、及び／又は、
ｉｉｉ）前記第１装置はＩｏＴ装置である、請求項１～３のいずれかに記載のシステム。
【請求項５】
前記ソフトウェアにより実装される制御コンポーネントは、関連付けられたブロックチェーン関連アドレスと関連付けられた公開－秘密鍵暗号鍵ペアとを有し、
前記制御コンポーネントは、前記制御コンポーネントに関連付けられた前記公開－秘密鍵暗号鍵ペアに関連するルックアップキーを用いて、前記レポジトリからの前記命令セットにアクセスするよう構成される、請求項１～４のいずれかに記載のシステム。
【請求項６】
前記制御コンポーネントは、前記第１装置上に又は前記第１装置内で提供されるか、又は前記制御コンポーネントは前記第１装置と別個の場所に設けられ、前記第１装置との無線通信のために構成される、請求項１～５のいずれかに記載のシステム。
【請求項７】
前記制御コンポーネントは、有効なブロックチェーントランザクションの検出に基づき、前記第１装置の動作を制御するか又は影響を与えるよう構成される、請求項１～６のいずれかに記載のシステム。
【請求項８】
コンピュータにより実装される制御方法であって、
ネットワークと通信するために構成された第１装置を提供するステップであって、前記第１装置は、前記第１装置に関連付けられたIPアドレスと公開－秘密鍵暗号鍵ペアを有する、ステップと、
ブロックチェーンネットワークの状態を監視し、及び／又は前記ブロックチェーンネットワークへブロックチェーントランザクションを送信するよう構成されるソフトウェアにより実装される制御コンポーネントを提供するステップと、
前記制御コンポーネントに入力信号を送信するよう構成される１つ以上のクライアント装置を提供するステップであって、前記入力信号は、前記第１装置の制御に影響を与えるよう構成される、ステップと、
前記第１装置の機能を制御するために前記制御コンポーネントによる実行のために構成された命令セットを提供するステップと、
を含み、
ｉ）前記制御コンポーネントは、前記第１装置と別個の場所に格納されたレポジトリからの前記命令セットにアクセスするよう構成され、
ｉｉ）前記命令セットは、前記レポジトリに格納され、前記制御コンポーネントによる前記レポジトリからのダウンロード及びインストールのためにアクセスされ、
ｉｉｉ）前記レポジトリの場所は、ブロックチェーントランザクションの中で提供されるメタデータを用いて指示され又は提供される、方法。
【請求項９】
前記１つ以上のクライアント装置は、無線周波数認識装置（ＲＦＩＤ）である、請求項８に記載の方法。
【請求項１０】
前記第１装置は、前記１つ以上のクライアント装置が前記第１装置の近くにあることに基づき制御される、請求項８又は９に記載の方法。
【請求項１１】

ｉ）前記レポジトリは、分散ハッシュテーブル（ＤＨＴ）である、
ｉｉ）前記命令セットは、前記第１装置の動作を調整する、制御する、及び／又は影響を与えるために読み出される、及び／又は、
ｉｉｉ）前記第１装置はＩｏＴ装置である、請求項８～１０のいずれかに記載の方法。
【請求項１２】
前記ソフトウェアにより実装される制御コンポーネントは、関連付けられたブロックチェーン関連アドレスと関連付けられた公開－秘密鍵暗号鍵ペアとを有し、
前記制御コンポーネントは、前記制御コンポーネントに関連付けられた前記公開－秘密鍵暗号鍵ペアに関連するルックアップキーを用いて、前記レポジトリからの前記命令セットにアクセスするよう構成される、請求項９に記載の方法。
【請求項１３】
前記制御コンポーネントは、前記第１装置上に又は前記第１装置内で提供されるか、又は前記制御コンポーネントは前記第１装置と別個の場所に設けられ、前記第１装置との無線通信のために構成される、請求項８～１０のいずれかに記載の方法。
【請求項１４】
前記制御コンポーネントは、
暗号計算を実行し、
自身の関連付けられた秘密／公開鍵ペアにアクセスし、
関連付けられたBitcoin又は他のブロックチェーンに関連するアドレスを有し、
APIを介して前記第１装置に作用し、
シークレット共有プロトコル動作を実行する、
請求項８～１２のいずれかに記載の方法。
【請求項１５】
前記制御コンポーネントは、有効なブロックチェーントランザクションの検出に基づき、前記第１装置の動作を制御するか又は影響を与えるよう構成される、請求項８～１３のいずれかに記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、概して、分散台帳（ブロックチェーン）技術に関する。本発明は、（限定ではないが）ビットコインブロックチェーンを含む任意のブロックチェーン関連技術であり得る。本発明の態様は、ＩｏＴ（Internet of Things）にも関連する。本発明は、ＩｏＴ装置を制御するのに適し得る。
【背景技術】
【０００２】
本願明細書で、用語「ブロックチェーン」は、あらゆる形式の電子的な、コンピュータに基づく、分散台帳を含むよう使用される。これらは、総意に基づくブロックチェーン及びトランザクションチェーン技術、許可及び未許可台帳、共有台帳、サイドチェーン及びアルトチェーン、及びそれらの変形を含む。最も広く知られているブロックチェーン技術の用途はビットコイン台帳であるが、他のブロックチェーンの実装が提案され開発されている。ビットコインは便宜上及び説明を目的として本願明細書において言及されるが、本発明はビットコインのブロックチェーンと共に使用することに限定されず、代替のブロックチェーンの実装及びプロトコルが本発明の範囲に含まれることに留意すべきである。用語「ユーザ」は、人間又はコンピュータに基づくリソースを表すことがある。
【０００３】
ブロックチェーンは、ブロックにより構成される、コンピュータに基づく非集中型の分散型システムとして実装されるピアツーピア電子台帳である。また、ブロックはトランザクションにより構成される。各トランザクションは、ブロックチェーンシステム内で参加者間のデジタル資産の制御の転送を符号化するデータ構造であり、少なくとも１つのインプット及び少なくとも１つのアウトプットを含む。各ブロックは前のブロックのハッシュ

を含み、ブロックは共にチェーンになって、その発端からブロックチェーンに書き込まれている全てのトランザクションの永久的な変更不可能なレコードを生成する。トランザクションは、そのインプット及びアウトプットに埋め込まれたスクリプトとして知られる小さなプログラムを含む。スクリプトは、トランザクションのアウトプットがどのように及び誰によりアクセス可能かを指定する。ビットコインプラットフォーム上で、これらのスクリプトは、スタックに基づくスクリプト言語を用いて記述される。

【０００４】

トランザクションがブロックチェーンに書き込まれるために、「検証され」なければならない。ネットワークノード（マイナー）は、各トランザクションが有効であることを保証するために作業を実行し、無効なトランザクションはネットワークにより拒否される。ノードにインストールされたソフトウェアクライアントは、自身のロック及びアンロックスクリプトを実行することにより、この検証作業を未使用トランザクション（ＵＴＸＯ）に対して実行する。ロック及びアンロックスクリプトの実行が真と評価するならば、トランザクションは有効である。スクリプト言語の多くのコマンドは、ブール値（例えば、ＯＰ－ＥＱＵＡＬ）を返す。これは、ブロックチェーントランザクション内に条件付けを構築することを可能にする。

【０００５】

ブロックチェーン技術は、暗号通貨実装の使用のために最も広く知られている。しかしながら、更に近年は、デジタル起業家が、新しいシステムを実装するために、ビットコインの基づく暗号通貨セキュリティシステムの使用、及びブロックチェーンに格納可能なデータの両者を探索し始めている。本発明は、ブロックチェーン技術の１つのこのような新しい新規な使用に関する。

【０００６】

特に、本発明は、広範な且つ様々な範囲のコンピュータにより実装されるシステムを生成するための簡易だが効率的且つ強力なメカニズムを実装するための、ブロックチェーンの使用に関する。このようなシステムは、処理を自動化し及び制御する及び／又は装置の動作を指示する制御ユニット及び制御システムを含み得る。

【０００７】

このような装置は、ＩｏＴ装置を含み得る。ＩｏＴ装置は、電子回路、ソフトウェア、センサ、及びネットワーク接続能力、等を搭載されて、それらが他の装置及びシステムと多くの場合には無線手段を介して通信すること、及び所望のタスクを実行することを可能にする。幾つかの例では、ＩｏＴ装置は、極めて小さく、限られた処理及び記憶能力しか含まない場合がある。これは、装置のタスクのために必要なソフトウェアが大きく複雑である場合に、問題を呈する。さらに、ＩｏＴ接続性及び知性を提供するために必要なソフトウェア及びハードウェアが装置自体において又はその中でのみ提供されるので、そのインストール、維持、及び更新、等が一層困難且つ高価になる。さらに、ＩｏＴの可能性に関する近年の興奮は、セキュリティに関する関心により緩和されている。

【０００８】

従来の開示である２０１５年１月の「ADEPT:An IoT Practitioner Perspective」は、ブロックチェーン技術をＩｏＴ装置に統合する手法を開示する。出願時、これは［https://ia９０２６０１.us.archive.org/４/items/pdfy－esMcC００dKmdo５３－/IBM%２０ADEPT%２０Practictioner%２０Perspective%２０－%２０Pre%２０Publication%２０Draft%２０－%２０７%２０Jan%２０２０１５.pdf］から検索された。この開示（以後「ＡＤＥＰＴ」）は、洗濯機であって、コントラクトにより管理され、「洗剤の残量不足」の条件を満たすと、ブロックチェーンを介して、補給品を購入するために小売店と通信する、洗濯機を記載する。

【０００９】

しかしながら、ＡＤＥＰＴは、ＩｏＴシステムがブロックチェーンに対してどのように動作するかを開示しているが、このようなシステムがブロックチェーンを用いてどのように配置され、構成され、技術的に影響されるかという問題を解決しない。言い換えると、

ＡＤＥＰＴは、ＩｏＴ上のソフトウェアがブロックチェーンと自立的に相互作用する方法を説明するが、最初に装置上にソフトウェアを得る方法を、さらには、次に使用／動作／展開中に装置の動作を変更する方法を、議論又は開示していない。

【００１０】

したがって、強力なサイバーセキュリティを保持しながら、汎用であるが（つまり、装置固有ではない）、任意の装置にロードするのに十分に小型であるオペレーティングシステムを有することは有利である。望ましくは、このようなオペレーティングシステムは、固定的機能ではなく、装置の動的機能を提供し得る。言い換えると、簡易な効率的且つ動的な方法で、装置の構成、設定及び機能を変更できれば、有意に技術的に有利である。このような技術的ソリューションは、ＡＤＥＰＴのような従来技術では解決されない。既知の従来技術に優る更なる利点は、装置により提供されるサービスに対する支払を処理する可能性を含む、簡易、セキュア、且つロバストな制御機能を可能にする能力である。本発明は、ＩｏＴを、例えばビットコインプロトコルのようなブロックチェーンプロトコルとインタフェースすることにより、これらの目的及び他の目的を解決する。

【００１１】

したがって、本発明によると、添付の請求の範囲に定められるようなシステム及び方法が提供される。

【発明の概要】

【発明が解決しようとする課題】

【００１２】

本発明は、コンピュータにより実装されるシステム及び方法を提供し得る。本発明は、１又は複数の装置の活動を制御し、指示し、及び／又は影響を与えるので、制御方法として記載され得る。本発明は、オペレーティングシステムとして記載され得る。本発明は、ソフトウェアにより実装されて良い。

【００１３】

本発明は、少なくとも１つの装置の活動を調整し、制御し、及び／又は影響を与えるオペレーティングシステムを有し得る。オペレーティングシステムは、それが制御する装置と独立であるという意味で、汎用であって良い。

【００１４】

本発明は、オペレーティングシステム（「制御コンポーネント」）が相互作用するよう構成されるブロックチェーンプラットフォームを用いて実施されて良い。ブロックチェーンは、ビットコインブロックチェーンであって良い。好適な実施形態では、装置はＩｏＴ（Internet of Things）装置である。

【００１５】

本発明は、装置を制御する、コンピュータにより実装される制御システムを提供し得る。システムは、装置であって、ネットワークと無線通信するよう構成され、該装置に関連付けられたＩＰアドレス及び公開鍵－秘密鍵暗号鍵ペアを有する装置と、ブロックチェーンネットワークの状態を監視し及び／又はブロックチェーンネットワークへブロックチェーントランザクションを送信するよう構成されるソフトウェアにより実装される制御コンポーネントと、及び／又は、装置の機能を制御するために制御コンポーネントによる実行のために構成される命令セットと、を有する。

【００１６】

制御コンポーネントは、装置と別個の、つまり「装置外の」、記憶場所からの命令セットにアクセスするよう構成され得る。

【００１７】

制御コンポーネントは、入力ソースから入力信号を受信するよう構成されて良い。入力ソースは、更なる装置、及び／又は、コンピュータに基づくリソース若しくはエージェント、であって良い。コンピュータに基づくエージェント又はリソースは、実質的に後述される通りであって良い。

【００１８】

命令セットは、分散ハッシュテーブル（ＤＨＴ）に格納され、ＤＨＴから制御コンポーネントによりダウンロード及びインストールのためにアクセスされて良い。この利点は、命令（及び、したがって装置の機能）を変更する能力を提供することである。

【００１９】

ＤＨＴの場所は、ブロックチェーントランザクションの中で提供されるメタデータを用いて示され又は提供されて良い。この利点は、命令の場所がブロックチェーンの中に不変に記録されることである。したがって、永久的及び耐タンパー性記録が提供され、場所はブロックチェーンへのアクセスを有する任意のパーティにより検証可能である。したがって、セキュリティ及び検証が向上される。

【００２０】

命令セットは、暗号鍵ペアに関連するルックアップキーを用いて、制御コンポーネントによりアクセスされて良い。

【００２１】

制御コンポーネントは、装置上に又はその中に設けられて良い。制御コンポーネントは、装置外の場所に設けられ、装置と無線通信するよう構成されて良い。

【００２２】

制御コンポーネントは、暗号計算を実行し、自身の関連する秘密／公開鍵ペアにアクセスし、関連付けられたビットコイン又は他のブロックチェーンに関連するアドレスを有し、ＡＰＩを介して装置を作動させ、シークレット共有プロトコル演算を実行する、よう構成されて良い。これは、実質的に後述されるようなシークレット共有プロトコルに従って良い。

【００２３】

制御コンポーネントは、有効ブロックチェーントランザクションの検出に基づき、装置の活動に影響を与える又は制御するよう構成されて良い。

【００２４】

本発明は、実質的に後述するようなシステム及び／又は方法を提供して良い。

【００２５】

本発明は、装置又は複数の装置を制御するよう構成される、コンピュータにより実施される制御方法を提供し得る。方法は、装置であって、ネットワークと（無線）通信するよう構成され、該装置に関連付けられたＩＰアドレス及び公開鍵－秘密鍵暗号鍵ペアを有する装置を提供するステップと、ブロックチェーンネットワークの状態を監視し及び／又はブロックチェーンネットワークへブロックチェーントランザクションを送信するよう構成されるソフトウェアにより実装される制御コンポーネントを提供するステップと、装置の機能を制御するために制御コンポーネントによる実行のために構成される命令セットを提供するステップと、を有して良い。

【００２６】

制御コンポーネントは、装置と別個の記憶場所からの命令セットにアクセスするよう構成されて良い。制御コンポーネントは、入力ソースから入力信号を受信するよう構成されて良く、入力ソースは、更なる装置、及び／又は、コンピュータに基づくリソース若しくはエージェント、であって良い。

【００２７】

命令セットは、分散ハッシュテーブル（ＤＨＴ）に格納され、ＤＨＴから制御コンポーネントによりダウンロード及びインストールのためにアクセスされて良い。

【００２８】

ＤＨＴの場所は、ブロックチェーントランザクションの中で提供されるメタデータを用いて示され又は提供されて良い。命令セットは、暗号鍵ペアに関連するルックアップキーを用いて、制御コンポーネントによりアクセスされて良い。

【００２９】

制御コンポーネントは、装置上に又はその中に設けられて良い。制御コンポーネントは、装置外の場所に設けられ、装置と無線通信するよう構成されて良い。

【００３０】
制御コンポーネントは、暗号計算を実行し、自身の関連する秘密／公開鍵ペアにアクセスし、関連付けられたビットコイン又は他のブロックチェーンに関連するアドレスを有し、ＡＰＩを介して装置を作動させ、シークレット共有プロトコル演算を実行する、よう構成されて良い。
【００３１】
制御コンポーネントは、有効ブロックチェーントランザクションの検出に基づき、装置の活動に影響を与える又は制御するよう構成されて良い。
【００３２】
本発明の一実施形態又は態様に関して説明された任意の機能は、本発明の任意の他の特徴又は実施形態に適用されても良い。例えば、システムに関連して言及された任意の特徴は方法に適用されて良く、逆も同様である。
【００３３】
本発明の上述の及び他の態様は、本願明細書に記載される実施形態から明らかであり、それらの実施形態を参照して教示される。本発明の実施形態は、単なる例として添付の図面を参照して以下に説明される。
【図面の簡単な説明】
【００３４】
【図１】本発明の一実施形態により構成され、説明的使用例に関する、システムを示す。
【図２】図１の制御システムのための真理値表を示す。
【図３】図１の例のためのアンロックトランザクションの処理のステップを示す。
【図４】シークレットを共有する及び公開鍵又は秘密鍵を生成するために使用可能な技術を示す。
【図５】シークレットを共有する及び公開鍵又は秘密鍵を生成するために使用可能な技術を示す。
【図６】シークレットを共有する及び公開鍵又は秘密鍵を生成するために使用可能な技術を示す。
【図７】シークレットを共有する及び公開鍵又は秘密鍵を生成するために使用可能な技術を示す。
【図８】シークレットを共有する及び公開鍵又は秘密鍵を生成するために使用可能な技術を示す。
【図９】図９～１１はブロックチェーントランザクションのロックスクリプトが論理ゲートの機能を実装するために使用される一実装の態様を示す。図９は、ブール出力Ｘを生成するために、２つのブール入力Ａ及びＢが第１トランザクションのロックスクリプト内で評価される技術の概略を示す。
【図１０】第１及び第２ブロックチェーントランザクションを用いて論理ゲートを実装する技術の概略を示す。
【図１１】ブロックチェーントランザクションのロックスクリプトが論理ゲートの機能を実装するために使用される処理を示す。
【発明を実施するための形態】
【００３５】
本発明は、特に下の利点を提供する：
・本発明は、意図的に「薄い（thin）」（メモリ及び／又は処理要件に関して小型である）、したがって任意のＩｏＴ装置に実装可能な、オペレーティングシステムを提供する。
・装置固有機能が装置内でハードコードされないので、分散ハッシュテーブル（Distributed Hash Table：ＤＨＴ）のようなセキュアレポジトリからロードするのではなく、容易に「アップグレード」できる。これは、動的構成を実現しない従来技術に優る有意な技術的改良点である。
・自立コンピューティングエージェント（或いは、ＩｏＴ装置又はその外部に存在するソフトウェア）により制御及び管理可能である。

・ブロックチェーン、例えばビットコインプラットフォームとインタフェースするとき、支払処理機能の統合を可能にする。
・ビットコインＥＣＣのようなブロックチェーン暗号法に基づき、ロバストなセキュリティを提供する。
【００３６】
ブロックチェーンＩｏＴ装置（Blockchain IOT Device：ＢＩＴ）は、ＢＩＤ外に安全に格納され暗号鍵によりアクセスされる所定の命令を実行するよう設定されるコンピューティングエージェントである。「ＢＩＤ外（off－BID）」は、命令がＢＩＤ自体の中で提供されないが、他の場所に格納され必要に応じて及び必要なときにアクセスされることを意味する。これらの命令は、選択されたタスク又は複数のタスクを実行するために選択され及び構成される。実行されると、命令は、ＩｏＴ装置の動作を制御し及び影響を与え得る。好適な実施形態では、ＢＩＤは、ＩｏＴ自体に存在し、これはＢＩＤがＩｏＴ装置内に又はＩｏＴ装置上に設けられたメモリにインストールされることを意味する。しかしながら、他の実施形態では、ＢＩＤは、装置外にあり、装置へのインターネット接続を有して良い。
【００３７】
ＩｏＴ装置は、自身の暗号鍵（及びＩＰアドレス）を有し、したがって、他の装置又はＤＨＴ等と安全に通信し及び相互作用できる。その「オペレーティングシステム」は簡易な汎用システムであり、以下のための（少なくとも限定ではない）幾つかの埋め込まれた機能を備える。
・暗号計算、
・（ＤＨＴのような）外部ソースからの命令の読み出し、
・トグルスイッチのような（つまり、物理的ＩｏＴ装置上にあるような）単純動作の実行。
【００３８】
したがって、ＩｏＴ装置又はその関連付けられたＢＩＤは、それら自身の内蔵命令を含まず、自身が何を行うか、又はそれをどのように行うかを「知らない」。ＢＩＤは、他の場所から命令を安全に読み出すメカニズムを含むだけである。ＢＩＤは、単純動作セットを実行できるだけである（以下は、単なる説明のためであり、限定ではない）：
・自身のマスタ秘密鍵及び公開鍵ペアへのアクセス。自身の（導出可能な）ＢＴＣアドレスも有する。
・ＩＰアドレスへデータを送信する、及びＩＰアドレスからデータを受信する、能力。
・シークレット共有プロトコル計算（後述する）。好適な実施形態では、これらは機械コードで実装されて良い。
・ブロックチェーンイベントの検索及び解釈。
・（基本的に単なるスイッチセットである標準ＡＰＩによる）取り付けられる物理的装置の作動及び制御。
【００３９】
ＢＩＤの入って来る及び出て行く通信は、後述するようなセキュリティメカニズムを用いて暗号化される。これは、鍵が共有シークレットを用いて生成されることを可能にする。これは、以下を可能にする：
（ｉ）「ハッキング」からの大きなセキュリティ。
（ｉｉ）簡易な汎用ソフトウェアアップグレードプロトコル。
（ｉｉｉ）装置不可知主義。
【００４０】
本発明は、したがって、任意のＩｏＴ装置において使用可能な汎用オペレーティングシステムを提供する。装置自体はプログラミングされず、全てのプログラムは、別個に格納され、設定時に（或いは、幾つかの実施形態では実行時に）装置にロードされる。
【００４１】
＜本発明の使用例＞
以下の説明のための例は、オートフィーダＩｏＴ装置の制御のための、本発明の一実施

形態の使用に関する。これは、本発明の一実装がどのように使用され得るかの一例として、説明目的でのみ適用される。

【００４２】

図１を参照すると、システム１００は、それぞれ１０２ａ及び１０２ｂとして数字を付された第１及び第２クライアント装置、第１クライアント装置１０２ａ及び第２クライアント装置１０２ｂから入力を受信し並びに第１クライアント装置１０２ａ及び第２クライアント装置１０２ｂへ情報を送信するよう動作するＢＩＤ制御システム１０４、を有する。この例示的な使用例では、第１及び第２クライアント装置１０２ａ、１０２ｂは、ＢＩＤ制御システム１０４により検出可能な無線周波数認識装置（radio frequency identification device：ＲＦＩＤ）である。制御システム１０４は、ブロックチェーンを使用するよう動作し、及びブロックチェーンへ出力を送信するよう動作する。

【００４３】

制御システム１０４がどのように動作するかを、アルキメデス（Ａ、Archimedes）及びバートランド（Ｂ、Bertrand）という名のキャロル（Carol）の２匹の犬の例を用いて、説明する。２匹の犬は、一日中、裏庭に放っておかれ、彼らが同時に食べなければ２匹とも互いに仲が良い。同時の食事は何らかの理由で彼らを攻撃的にさせ互いに闘わせる。Ａ及びＢは、２匹とも、ＩｏＴ（Internet Of Things）装置１０１により検出可能な、識別できるＲＦＩＤ首輪、つまり第１ＲＦＩＤ首輪１０２ａ及び第２ＲＦＩＤ首輪１０２ｂを有する。このＩｏＴ装置は、１匹の犬により消費される餌の指定量を分配するオートフィーダである。つまり、ＢＩＤ制御システム１０４は、ＩｏＴ給餌装置の動作を制御する。

【００４４】

本例では、ＢＩＤ１０４は、ＩｏＴオートフィーダ上で提供されるソフトウェアリソース又はコンポーネントであり、フィーダの機能を制御するためにフィーダとインタフェースする。

【００４５】

ＢＩＤは、ＤＨＴから自身の命令をダウンロードしインストールすることにより、自身の寿命を開始する。これらの命令が変更されるまで、再びこれを行う必要はない。これは、例えば、ＢＩＤがアップグレードされる必要があるとき、又はＢＩＤの動作が完全に変更されるべきとき、例えばＢＩＤの命令セットが３以上のＲＦＩＤ信号を検出するよう変更され得るとき、であって良い。

【００４６】

制御エージェントは、ＢＩＤにより送信された値を用いて、ブロックチェーントランザクションを生成し、更に後述するように、各反復の後にＢＩＤと新しいシークレットを共有する。

【００４７】

ＢＩＤ制御システム１０４の機能は、以下のロックスクリプトを用いてロックされるブロックチェーントランザクションを用いて実装される。

【００４８】

OP_HASH１６０〈unlockingscripthash〉OP_EQUAL

トランザクションは、ＩｏＴオートフィーダ装置を制御するための命令セットを（分散ハッシュテーブル（distributed hash table：ＤＨＴ）にリンクするメタデータを介して）提供するために生成され、後述する内容に従い確立された計算リソースへの命令を含んで良い。トランザクション自体の中に命令を格納するのではなく、メタデータは、命令がアクセスされ得る場所へのポインタ又は参照を含み得る。したがって、命令は、「ブロック外（off－block）」に保持されて良い。

【００４９】

ブロックチェーンは、活動を制御するメカニズムを提供するだけでなく、行われたイベントに関する情報も記録する。例えば、ブロックチェーンは、給餌回数、何時に給餌が生じたか、どの犬が食べたか、最大餌割り当てが分配されたか、等を計数する能力を提供す

る。ブロックチェーンは、暗号セキュリティも提供する。

【００５０】

トランザクションの重要な機能は、一度に１匹の犬がフィーダに存在する場合にのみ、餌が分配されることを保証することである。したがって、何らかの条件が、トランザクションのスクリプト内に構築される必要がある。これは、図２に示される真理値表により、及び図９～１１を参照して、ＸＯＲ関数により達成される。

・ＡもＢもフィーダに存在しない場合、餌を分配しない。
・Ａがフィーダに存在するがＢが存在しない場合、餌を分配する。
・Ｂがフィーダに存在するがＡが存在しない場合、餌を分配する。
・Ａ及びＢの両方がフィーダに存在する場合、餌を分配しない。

【００５１】

Ａ又はＢがフィーダに存在するとき、ＲＦＩＤ信号が、それぞれのクライアント装置、つまり第１ＲＦＩＤ首輪１０２ａ又は第２ＲＦＩＤ首輪１０２ｂから、オートフィーダの制御システム１０４へ送信されて、当該犬のセキュアな現在のパズル解（puzzle solution）（これは、各反復の後に新しいパズル解によりセキュアに置換される）をアンロックする。代わりに、Ａ又はＢがフィーダに存在しない場合、それぞれのＲＦＩＤ首輪からフィーダへ、乱数が送信される。言い換えると、犬が「フィーダに存在する」ことは、自身のＲＦＩＤ首輪がフィーダの検出可能範囲内にあることを意味する。この場合、送信のために関連パズルがアンロックされる。存在しない場合、規定値は乱数である。

【００５２】

従来知られているように、パズル解は、ハッシュされると、（ビットコイン）スクリプト内の格納された値と比べたときに一致する値を結果として生じるデータである。つまり、処理は次の通りである：シークレット値（「解」）がハッシュされ、後に比較するためにロックスクリプト内に格納される。ロックスクリプトをアンロックするために、シークレットがアンロック（Ｒｅｄｅｅｍ）によりスクリプトに提示される。提示された値はハッシュされ、次に格納されたハッシュと比較される。比較が、それらが等しいことを決定した場合、比較の結果は「真」である。実際には、格納された値は、シークレットのダブルハッシュ（DOUBLE－hash）であり、提示された値は、シークレットのシングルハッシュ（SINGLE－hash）である。これは、任意の長さのシークレットが、スクリプトへの入力として標準的な管理可能なサイズ（つまり、常に１６０ビットの長さ）まで短縮されることを可能にする。

【００５３】

オートフィーダＢＩＤは、ＢＩＤのキー／ペアに関連するルックアップキーを用いてＤＨＴから読み出された自身の命令を実行する。制御エージェントは、ＢＩＤへ／からのデータフローを管理する（つまり、ＲＦＩＤ信号に関連するがＢＩＤの命令セットに関連しないデータ）。したがって、オートフィーダＢＩＤは自身の状態を監視する。オートフィーダＢＩＤは、別個の制御エージェント１０３から受信した２つのシークレット値（Ｓ１及びＳ２）を格納する。制御エージェント１０３は、給餌処理を監督するよう構成された適切にプログラムされた計算リソースであり得る。シークレット値Ｓ１及びＳ２は、犬のＲＦＩＤ首輪が範囲内で検出されるとき、条件付きで使用される。適切なＤＨＴから読み出された自身の命令に基づき、範囲内でＲＦＩＤを検出すると（１日のうちの時間、前の給餌回数、他の制限、等に関連する他の条件と一緒に）、自身の制御エージェント（後述する）として作用する汎用エージェントへ信号を送信する。信号は以下を含む：

・アルキメデスのＲＦＩＤが検出された場合、Ｓ１（＝パズルＡ解）；その他の場合、乱数。
・バートランドのＲＦＩＤが検出された場合、Ｓ２（＝パズルＢ解）；その他の場合、乱数。

【００５４】

オートフィーダＢＩＤは次に、以下を行う：

・オートフィーダは、ネットワーク上の有効なトランザクションについてチェックする（

ブロック上に発行されていて良く又は未発行であって良いが、有効なトランザクションでなければならない）。このトランザクションは、制御エージェントにより生成されブロードキャストされる。このトランザクションは、埋め込まれたＸＯＲテストに合格した場合、有効である。合格しない場合、このトランザクションは、無効であり、ネットワーク上の第１ホップを超えて伝搬されない。したがって、このトランザクションは、ＢＩＤにより検出されない。あるいは、ＢＩＤが第１ホップであり、したがってトランザクションが検出される場合、（任意の他のノードに関する）ＢＩＤの関数の部分がトランザクションを検証する。したがって、結果として生じる措置を取る前に、トランザクションが有効かどうかを検出できる。有効なトランザクションは、必要な、つまり給餌イベントに関する情報がブロックチェーンに格納され及び記録されていることも保証する。

・上述の答えが真である場合、ＢＩＤは、自身の条件付き命令を実行する。本例では、ＢＩＤは幾らかの餌を分配する。

・制御エージェント１０３から送信を受信し、２つのシークレット（後述のようにＳ１及びＳ２）を共有することを可能にし、次の反復に備えてこれらのシークレット値を内部で更新する。

【００５５】

ビットコイントランザクションのためのロックスクリプトは、以下により与えられる：

OP_HASH１６０〈Puzzle－A〉OP_EQUALOP_SWAP

OP_HASH１６０〈Puzzle－B〉OP_EQUAL

OP_NUMEQUALOP_NOTOP_VERIFY

OP_１metadata１PubK－CarolOP_２OP_CHECKMULTSIG

ここで、次の通りである。

パズルＡは、OP_HASH１６０の等価な結果である（パズルＡ解）。

パズルＢは、OP_HASH１６０の等価な結果である（パズルＢ解）。

metadata１は、ＤＨＴに格納された符号化された命令への参照を含む。

PubK－Carolはキャロルの公開鍵である。

【００５６】

エージェントのプログラミングはハードコードされて良く、又はＤＨＴから自身の命令を読み出しても良いことに留意する。

【００５７】

符号化された命令は、メタデータを用いるブロックチェーントランザクションからの取引を参照する後述の手順に従い格納され及びアクセスされて良い。

【００５８】

キャロルの公開鍵は、後述する処理を用いてセキュアに保持され又は再生成可能であって良い。

【００５９】

上述の説明のためのブロックチェーントランザクションをアンロックするために、以下のスクリプトが必要である：

Sig－CarolPuzzle－B－solutionPuzzle－A－Solution〈unlockingscript〉

以下のステップの説明のために図３を参照する。

【００６０】

ステップＳ３００で、制御システム１０４は、提示されたパズルＡ解をハッシュし、それを、記憶装置から読み出されたパズルＡの格納されたバージョン（このバージョンは解のハッシュである）と比較するよう動作する。パズルＡの格納されたバージョンは、制御システム１０４のローカルにある記憶装置に、又は任意の他の適切な記憶媒体に格納されて良い。それらが等しい場合、スタックの最上位＝１である。それらが異なる場合、スタックの最上位＝０である。

【００６１】

ステップＳ３０２で、スタックの最上位は、次に、パズルＢ解であるスタックの２番目のアイテムによりスワップされる。これは、ハッシュされ、記憶装置から読み出されたパ

ズルＢの格納されたバージョンと比較され、ここでも、Ｓ３００からの結果と同様に、スタックの最上位に１又は０をプッシュする。パズルＢの格納されたバージョンは、制御システム１０４のローカルにある記憶装置に、又は任意の他の適切な記憶媒体に格納されて良い。

【００６２】

ここで、上位２つのスタックアイテムは、それぞれ０又は１である。ステップＳ３０４で、OP_NUMEQUALは、数値が等しい場合に１を返し、その他の場合に０を返す。これは、ＸＯＲ真理値表の正反対である。

【００６３】

ステップＳ３０６で、OP_NOTは、必要なＸＯＲ結果を生成するために、スタック上の最上位アイテムをフリップする。

【００６４】

ステップＳ３０８で、OP_VERIFYは、スタックの最上位にあるアイテムが１であるかどうかを調べ、否である場合、つまり、ＸＯＲ演算が失敗した場合、第１及び第２クライアント装置からの１つより多くの入力が一致するパズル解を返しているので、トランザクションは直ちに無効としてマークされる。この結果、１より多くの犬がＩｏＴ分配器に居るので、ＩｏＴ分配器から餌が分配されない。つまり、制御システム１０４の出力は、基礎にあるビットコイントランザクションの実行により制御される。

【００６５】

OP_VERIFYが１を返す場合、制御システム１０４における処理は、スクリプトのマルチシグへ戻り、ステップＳ３１０でキャロルの署名の存在が調べられる。

【００６６】

アンロックスクリプトを分析する際に制御システム１０４により実行されるスタック演算は、以下に示される。先ず、制御システム１０４は、OP_EQUALを用いてハッシュをアンロックスクリプトのハッシュと比較するために、アンロックスクリプトをハッシュする。この後に、アンロックスクリプトが実行される。

【００６７】

【表１－１】

スタック	スクリプト	説明
Empty	Sig-Carol Puzzle-B-solution Puzzle-A-Solution OP_HASH160 <Puzzle-A> OP_EQUAL OP_SWAP OP_HASH160 <Puzzle-B> OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	
Sig-Carol Puzzle-B-solution Puzzle-A-Solution	OP_HASH160 <Puzzle-A> OP_EQUAL OP_SWAP OP_HASH160 <Puzzle-B> OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	スタックに追加されるデータ
Sig-Carol Puzzle-B-solution Puzzle-A-Solution-hashed	<Puzzle-A> OP_EQUAL OP_SWAP OP_HASH160 <Puzzle-B> OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	最上位スタックアイテムがハッシュされる
Sig-Carol Puzzle-B-solution Puzzle-A-Solution-hashed <Puzzle-A>	OP_EQUAL OP_SWAP OP_HASH160 <Puzzle-B> OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	所与のハッシュ（パズルＡ）がスタックの最上位にプッシュされる
Sig-Carol Puzzle-B-solution FALSE	OP_SWAP OP_HASH160 <Puzzle-B> OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	上位２つのアイテムが比較され、結果（偽）がスタックの最上位にプッシュされる
Sig-Carol FALSE	OP_HASH160	上位２つのスタックアイテ

【表１－２】

Puzzle-B-solution	<Puzzle-B> OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	ムがスワップされる
Sig-Carol FALSE Puzzle-B-solution-hashed	<Puzzle-B> OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	最上位のスタックアイテムがハッシュされる
Sig-Carol FALSE Puzzle-B-solution-hashed <Puzzle-B>	OP_EQUAL OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	所与のハッシュ（パズルＢ）がスタックの最上位にプッシュされる
Sig-Carol FALSE TRUE	OP_NUMEQUAL OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	上位２つのアイテムが比較され、結果（真）がスタックの最上位にプッシュされる
Sig-Carol FALSE	OP_NOT OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	上位２つの数値（０又は１）が比較され、結果（偽）がスタックの最上位にプッシュされる
Sig-Carol TRUE	OP_VERIFY OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	最上位のスタックアイテムはフリップされる（偽＝０から真＝１へ）
Sig-Carol	OP_1 metadata1 PubK-Carol OP_2 OP_CHECKMULTSIG	最上位のスタックアイテムが検証される。真ならば、トランザクションは（未だ）無効としてマークされず、スクリプトが続行する
TRUE	Empty	マルチシグがチェックされ合格する

＜共有シークレットを用いてキーを生成する＞

キーは、セキュアに保持され又は再生成されて良い。特に、公開鍵を導出するために使用され得る秘密鍵の場合には、秘密鍵は分解して格納されて良い。

【００６８】

ユーザ、つまりアリス又はボブは彼らの秘密鍵の一部を保持して良く、サービスプロバイダが第２の部分を保持して良く、第３の部分はリモートセキュアサイトに保持されて良い。秘密鍵は、３つの部分のうちの任意の２つを用いて再構成されて良い。あるいは、より一般的には、秘密鍵が、ｎ個の部分のうちの任意のｍ個を用いて再構成されて良い。

【００６９】

秘密鍵は、再構成可能な場合、使用点において公開鍵を再生成するために使用でき、秘密鍵及び公開鍵は使用後に再び廃棄できる。

【００７０】
秘密鍵の分離は、シャミアの秘密分散法（Shamir’s Secret Sharing Scheme）を用いて達成されて良い。秘密鍵－公開鍵ペアは、以下の方法を用いてマスタ鍵から確定的に導出できる。この方法は、シークレット値がそれらを送信することなく参加者により共有されることを可能にする。
【００７１】
システムは、以下に記載するサブキー生成の方法を用いて参加者の公開鍵を生成して良い。
【００７２】
図４は、通信ネットワーク５を介して第２ノード７と通信する第１ノード３を含むシステム１を示す。第１ノード３は関連する第１処理装置２３を有し、及び第２ノード５は関連する第２処理装置２７を有する。第１及び第２ノード３、７は、コンピュータ、電話機、タブレットコンピュータ、モバイル通信装置、コンピュータサーバ、等のような電子装置を含んで良い。一例では、第１ノード３はクライアント（ユーザ）装置であって良く、第２ノード７はサーバであって良い。サーバは、デジタルウォレットプロバイダのサーバであって良い。
【００７３】
第１ノード３は、第１ノードマスタ秘密鍵（V_{1C}）及び第１ノードマスタ公開鍵（P_{1C}）を有する第１非対称暗号対に関連付けられる。第２ノード（７）は、第２ノードマスタ秘密鍵（V_{1S}）及び第２ノードマスタ公開鍵（P_{1S}）を有する第２非対称暗号対に関連付けられる。言い換えると、第１及び第２ノードは、それぞれ、個々の公開鍵－秘密鍵ペアを保有する。
【００７４】
個々の第１及び第２ノード３、７の第１及び第２非対称暗号対は、ウォレットの登録のような登録処理中に生成されて良い。各ノードの公開鍵は、通信ネットワーク５に渡るように、公に共有されて良い。
【００７５】
第１ノード３及び第２ノード７の両者において共通シークレット（common secret：ＳＣ）を決定するために、ノード３、７は、通信ネットワーク５を介して秘密鍵を通信することなく、それぞれ方法３００、４００のステップを実行する。
【００７６】
第１ノード３により実行される方法３００は、少なくとも第１ノードマスタ秘密鍵（V_{1C}）及び生成器値（Generator Value：ＧＶ）に基づき、第１ノード第２秘密鍵（V_{2C}）を決定するステップ３３０を含む。生成器値は、第１ノードと第２ノードとの間で共有されるメッセージ（Ｍ）に基づいて良い。これは、以下に詳述するように、通信ネットワーク５を介してメッセージを共有するステップを含んで良い。方法３００は、少なくとも第２ノードマスタ公開鍵（P_{1S}）及び生成器値（Generator Value：ＧＶ）に基づき、第２ノード第２公開鍵（P_{2S}）を決定するステップ３７０を更に含む。方法３００は、第１ノード第２秘密鍵（V_{2C}）及び第２ノード第２公開鍵（P_{2S}）に基づき、共通シークレット（common secret：ＣＳ）を決定するステップ３８０を含む。
【００７７】
重要なことに、同じ共通シークレット（ＣＳ）が、方法４００により第２ノード７においても決定できる。方法４００は、第１ノードマスタ公開鍵（P_{1C}）及び生成器値（Generator Value：ＧＶ）に基づき、第１ノード第２公開鍵（P_{2C}）を決定するステップ４３０を含む。方法４００は、第２ノードマスタ秘密鍵（V_{1S}）及び生成器値（Generator Value：ＧＶ）に基づき、第２ノード第２秘密鍵（V_{2S}）を決定するステップ４７０を更に含む。方法４００は、第２ノード第２秘密鍵（V_{2S}）及び第１ノード第２公開鍵（P_{2C}）に基づき、共通シークレット（common secret：ＣＳ）を決定するステップ４８０を含む。
【００７８】

通信ネットワーク５は、ローカルエリアネットワーク、ワイドエリアネットワーク、セルラネットワーク、無線通信ネットワーク、インターネット、等を含んで良い。これらのネットワークでは、データは電気線、光ファイバ、又は無線のような通信媒体を介して送信されて良く、盗聴者１１による様な盗聴を受けやすい場合がある。方法３００、４００は、通信ネットワーク５を介して共通シークレットを送信することなく、第１ノード３及び第２ノード７が共通シークレットを両方とも独立して決定できるようにする。

【００７９】

したがって、１つの利点は、安全でない可能性のある通信ネットワーク５を介して秘密鍵を送信する必要を有しないで、共通シークレット（ＣＳ）が安全に且つ各ノードにより独立して決定できることである。また、共通シークレットは、秘密鍵として（又は秘密鍵の基礎として）使用されて良い。

【００８０】

方法３００、４００は、追加ステップを含んで良い。図８を参照する。方法３００は、第１ノード３において、メッセージ（Ｍ）及び第１ノード第２秘密鍵（V_{2C}）に基づき、署名メッセージ（ＳＭ１）を生成するステップを含んで良い。方法３００は、第１署名メッセージ（ＳＭ１）を通信ネットワークを介して第２ノード７へ送信するステップ３６０を更に含む。一方、第２ノード７は、第１署名メッセージ（ＳＭ１）を受信するステップ４４０を実行して良い。方法４００は、第１署名メッセージ（ＳＭ２）を第１ノード第２公開鍵（P_{2C}）により検証するステップ４５０、及び第１署名メッセージ（ＳＭ１）を検証するステップの結果に基づき第１ノード３を認証するステップ４６０を更に含む。有利なことに、これは、第２ノード７が、（第１署名メッセージが生成された場所である）意図された第１ノードが第１ノード３であることを認証することを可能にする。これは、第１ノード３だけが、第１ノードマスタ秘密鍵（V_{1C}）へのアクセスを有する、したがって、第１ノード３だけが、第１署名メッセージ（ＳＭ１）を生成するための第１ノード第２秘密鍵（V_{2C}）を決定できるという仮定に基づく。理解されるべきことに、同様に、第２署名メッセージ（ＳＭ２）は、第２ノード７において生成され、第１ノード３へ送信され得る。したがって、ピアツーピアシナリオにおけるように、第１ノード３は、第２ノード７を認証できる。

【００８１】

第１ノードと第２ノードの間のメッセージ（Ｍ）の共有は、様々な方法で達成されて良い。一例では、メッセージは、第１ノード３において生成されて良く、次に通信ネットワーク５を介して第２ノード７へ送信される。代替として、メッセージは、第２ノード７において生成されて良く、次に通信ネットワーク５を介して第１ノード３へ送信される。幾つかの例では、メッセージ（Ｍ）は公開されて良く、したがってセキュアでないネットワーク５を介して送信されて良い。１又は複数のメッセージ（Ｍ）は、データストア１３、１７、１９に格納されて良い。当業者は、メッセージの共有が様々な方法で達成できることを理解する。

【００８２】

有利なことに、共通シークレット（ＣＳ）の再生成を可能にするレコードは、そのレコード自体が秘密に格納され又はセキュアに送信される必要がなく、保持され得る。

【００８３】

＜登録の方法１００、２００＞

登録の方法１００、２００の一例は、図６を参照して記載される。図６では、方法１００は第１ノード３により実行され、方法２００は第２ノード７により実行される。これは、それぞれ第１ノード３及び第２ノード７のために第１及び第２非対称暗号対を確立するステップを含む。

【００８４】

非対称暗号対は、公開鍵暗号化で使用されるような、関連付けられた秘密鍵及び公開鍵を含む。本例では、非対称暗号対は、楕円曲線暗号システム（Elliptic Curve Cryptography：ＥＣＣ）及び楕円曲線演算の特性を用いて生成される。

【００８５】
方法１００、２００では、これは、共通ＥＣＣシステムに合意している１１０、２１０、且つ基点（Ｇ）を用いる、第１ノード及び第２ノードを含む（注：基点は、共通生成器として参照され得るが、用語「基点」は、生成器値ＧＶとの混同を避けるために使用される）。一例では、共通ＥＣＣシステムは、ビットコインにより使用されるＥＣＣシステムであるｓｅｃｐ２５６Ｋ１に基づいて良い。基点（Ｇ）は、選択され、ランダムに生成され、又は割り当てられて良い。
【００８６】
ここで第１ノード３について考えると、方法１００は、共通ＥＣＣシステム及び基点（Ｇ）を解決するステップ１１０を含む。これは、第２ノード７又は第３ノード９から、共通ＥＣＣシステム及び基点を受信するステップを含んで良い。代替として、ユーザインタフェース１５は、第１ノード３に関連付けられる。これにより、ユーザは、共通ＥＣＣシステム及び／又は基点（Ｇ）を選択的に提供できる。更に別の代替案では、共通ＥＣＣシステム及び／又は基点（Ｇ）の一方又は両方が、第１ノード３によりランダムに選択されて良い。第１ノード３は、通信ネットワーク５を介して、基点（Ｇ）と共に共通ＥＣＣシステムを使用することを示す通知を、第２ノード７へ送信して良い。また、第２ノード７は、共通ＥＣＣシステム及び基点（Ｇ）の使用に対する肯定応答を示す通知を送信することにより、解決して良い２１０。
【００８７】
方法１００は、第１ノード３が、第１ノードマスタ秘密鍵（V_{1C}）及び第１ノードマスタ公開鍵（P_{1C}）を有する第１非対称暗号対を生成するステップ１２０を更に含む。これは、共通ＥＣＣシステムの中で指定された許容範囲の中のランダム整数に少なくとも部分的に基づき、第１マスタ秘密鍵（V_{1C}）を生成するステップを含む。これは、次式に従い第１ノードマスタ秘密鍵（V_{1C}）及び基点（Ｇ）の楕円曲線点乗算に基づき、第１ノードマスタ公開鍵（P_{1C}）を決定するステップを更に含む。
$P_{1C}=V_{1C}\times G$（式１）
したがって、第１非対称暗号対は、以下を含む：
V_{1C}：第１ノードにより秘密に保持される第１ノードマスタ秘密鍵。
P_{1C}：公に知らされる第１ノードマスタ公開鍵。
【００８８】
第１ノード３は、第１ノードマスタ秘密鍵（V_{1C}）及び第１ノードマスタ公開鍵（P_{1C}）を、第１ノード３に関連付けられた第１データストア１３に格納して良い。セキュリティのために、第１ノードマスタ秘密鍵（V_{1C}）は、鍵が秘密のままであることを保証するために、第１データストア１３のセキュアな部分に格納されて良い。
【００８９】
方法１００は、図６に示すように、第１ノードマスタ公開鍵（P_{1C}）を通信ネットワーク５を介して第２ノード７へ送信するステップ１３０を更に含む。第２ノード７は、第１ノードマスタ公開鍵（P_{1C}）を受信すると２２０、第１ノードマスタ公開鍵（P_{1C}）を第２ノード７に関連付けられた第２データストア１７に格納して良い２３０。
【００９０】
第１ノード３と同様に、第２ノード７の方法２００は、第２ノードマスタ秘密鍵（V_{1S}）及び第２ノードマスタ公開鍵（P_{1S}）を有する第２非対称暗号対を生成するステップ２４０を含む。第２ノードマスタ秘密鍵（V_{1S}）も、許容範囲内のランダム整数である。また、第２ノードマスタ公開鍵（P_{1S}）は、次式により決定される。
$P_{1S}=V_{1S}\times G$（式２）
したがって、第２非対称暗号対は、以下を含む：
V_{1S}：第２ノードにより秘密に保持される第２ノードマスタ秘密鍵。
P_{1S}：公に知らされる第２ノードマスタ公開鍵。
【００９１】
第２ノード７は、第２非対称暗号対を第２データストア１７に格納して良い。方法２０

０は、第２ノードマスタ公開鍵（Ｐ１Ｓ）を第１ノード３へ送信するステップ２５０を更に含む。また、第１ノード３は、第２ノードマスタ公開鍵（Ｐ１Ｓ）を受信し１４０、格納して良い１５０。

【００９２】

理解されるべきことに、幾つかの代案では、それぞれの公開マスタ鍵は、受信され、（信頼できる第三者のような）第３ノード９に関連付けられた第３データストア１９に格納されて良い。これは、認証機関のような、公開ディレクトリとして動作する第三者を含んで良い。したがって、幾つかの例では、第１ノードマスタ公開鍵（Ｐ１Ｃ）は、共通シークレット（ＣＳ）が要求されるときだけ、第２ノード７により要求され受信されて良い（逆も同様である）。

【００９３】

登録ステップは、初期設定として１度生じるだけで良い。

【００９４】

＜セッション開始及び第１ノード３による共通シークレットの決定＞

共通シークレット（ＣＳ）を決定する一例は、図７を参照してここに記載される。共通シークレット（ＣＳ）は、第１ノード３と第２ノード７との間の特定のセッション、時間、トランザクション、又は他の目的のために使用されて良く、同じ共通シークレット（ＣＳ）を使用することが望ましい又はセキュアでなくて良い。したがって、共通シークレット（ＣＳ）は、異なるセッション、時間、トランザクション、等の間で変更されて良い。

【００９５】

以下は、上述したセキュアな送信技術の説明のために提供される。

【００９６】

［メッセージ（Ｍ）を生成する３１０］

本例では、第１ノード３により実行される方法３００は、メッセージ（Ｍ）を生成するステップ３１０を含む。メッセージ（Ｍ）は、ランダム、疑似ランダム、又はユーザ定義であって良い。一例では、メッセージ（Ｍ）は、Ｕｎｉｘ時間又はノンス（及び任意の値）に基づく。例えば、メッセージ（Ｍ）は次のように与えられ得る。

メッセージ（Ｍ）＝Ｕｎｉｘ時間＋ノンス（式３）

幾つかの例では、メッセージ（Ｍ）は任意である。しかしながら、理解されるべきことに、メッセージ（Ｍ）は、幾つかのアプリケーションで有用であり得る（Ｕｎｉｘ時間、等のような）選択的値を有して良い。

【００９７】

方法３００は、メッセージ（Ｍ）を通信ネットワーク３を介して第２ノード７へ送信するステップ３１５を含む。メッセージ（Ｍ）は秘密鍵についての情報を含まないので、メッセージ（Ｍ）は、セキュアでないネットワークを介して送信されて良い。

【００９８】

［生成器値（ＧＶ）を決定する３２０］

方法３００は、メッセージ（Ｍ）に基づき生成器値（Generator Value：ＧＶ）を決定するステップ３２０を更に含む。本例では、これは、メッセージの暗号ハッシュを決定するステップを含む。暗号ハッシュアルゴリズムの一例は、２５６ビット発生器値（ＧＶ）を生成するためにＳＨＡ－２５６を含む。つまり、

ＧＶ＝ＳＨＡ－２５６（Ｍ）（式４）

理解されるべきことに、他のハッシュアルゴリズムが使用されて良い。これは、セキュアなハッシュアルゴリズム（Secure Hash Algorithm：ＳＨＡ）ファミリの中の他のハッシュアルゴリズムを含んで良い。幾つかの特定の例は、ＳＨＡ３－２２４、ＳＨＡ３－２５６、ＳＨＡ３－３８４、ＳＨＡ３－５１２、ＳＨＡＫＥ１２８、ＳＨＡＫＥ２５６を含むＳＨＡ－３サブセットの中のインスタンスを含む。他のハッシュアルゴリズムは、ＲＩＰＥＭＤ（RACE Integrity Primitives Evaluation Message Digest）ファミリの中のアルゴリズムを含んで良い。特定の例は、ＲＩＰＥＭＤ－１６０を含んで良い。他のハッシュ関数は、Ｚｅｍｏｒ－Ｔｉｌｌｉｃｈハッシュ関数及びナップサック・ハッシ

ュ関数に基づくファミリを含んで良い。
【００９９】
［第１ノード第２秘密鍵を決定する３３０］
方法３００は、次に、第２ノードマスタ秘密鍵（V_{1C}）及び生成器値（ＧＶ）に基づき、第１ノード第２秘密鍵（V_{2C}）を決定するステップ３３０を含む。これは、次式に従い第１ノードマスタ秘密鍵（V_{1C}）及び生成器値（ＧＶ）のスカラ加算に基づき得る。
$V_{2C}=V_{1C}+GV$（式５）
したがって、第１ノード第２秘密鍵（V_{2C}）は、ランダム値ではないが、代わりに第１ノードマスタ秘密鍵から確定的に導出される。暗号対の中の対応する公開鍵、つまり第１ノード第２公開鍵（P_{2C}）は、以下の関係を有する。
$P_{2C}=V_{2C}\times G$（式６）
式５から式６にV_{2C}を代入すると、次式を得る。
$P_{2C}=(V_{1C}+GV)\times G$（式７）
ここで、「＋」演算子は楕円曲線点加算を表す。楕円曲線暗号代数は、分配的であり、式７は次式のように表すことができる。
$P_{2C}=V_{1C}\times G+GV\times G$（式８）
最後に、（式１）は（式７）に代入され、次式を得る。
$P_{2C}=P_{1C}+GV\times G$（式９．１）
$P_{2C}=P_{1C}+SHA-256(M)\times G$（式９．２）
したがって、対応する第１ノード第２公開鍵（P_{2C}）は、第１ノードマスタ公開鍵（P_{1C}）及びメッセージ（Ｍ）の導出可能な所与の知識であり得る。方法４００に関して以下に更に詳述するように、第２ノード７は、第１ノード第２公開鍵（P_{2C}）を独立に決定するために、このような知識を有して良い。
【０１００】
［メッセージ及び第１ノード第２秘密鍵に基づき、第１署名メッセージ（ＳＭ１）を生成する３５０］
方法３００は、メッセージ（Ｍ）及び決定した第１ノード第２秘密鍵（V_{2C}）に基づき、第１署名メッセージ（ＳＭ１）を生成するステップ３５０を更に含む。署名メッセージを生成するステップは、メッセージ（Ｍ）にデジタル方式で署名するために、デジタル署名アルゴリズムを適用するステップを含む。一例では、これは、第１署名メッセージ（ＳＭ１）を得るために、楕円曲線デジタル署名アルゴリズム（Elliptic Curve Digital Signature Algorithm：ＥＣＤＳＡ）の中でメッセージに第１ノード第２秘密鍵（V_{2C}）を適用するステップを含む。ＥＣＤＳＡの例は、ｓｅｃｐ２５６ｋ１、ｓｅｃｐ２５６ｒ１、ｓｅｃｐ３８４ｒ１、ｓｅ３ｃｐ５２１ｒ１を有するＥＣＣシステムに基づくものを含む。
【０１０１】
第１署名メッセージ（ＳＭ１）は、第２ノード７において対応する第１ノード第２公開鍵（P_{2C}）により検証できる。第１署名メッセージ（ＳＭ１）のこの検証は、第１ノード３を認証するために第２ノード７により使用されて良い。これは、方法４００において以下に議論される。
【０１０２】
［第２ノード第２公開鍵を決定する３７０’］
第１ノード３は、次に、第２ノード第２公開鍵（P_{2S}）を決定して良い３７０。上述のように、第２ノード第２公開鍵（P_{2S}）は、少なくとも第２ノードマスタ公開鍵（P_{1S}）及び生成器値（ＧＶ）に基づいて良い。本例では、公開鍵は、基点（Ｇ）との楕円曲線点乗算により秘密鍵として決定されるので３７０’、第２ノード第２公開鍵（P_{2S}）は、式６と同様に次のように表すことができる。
$P_{2S}=V_{2S}\times G$（式１０．１）
$P_{2S}=P_{1S}+GV\times G$（式１０．２）
式１０．２の数学的証明は、第１ノード第２公開鍵（P_{2C}）について式９．１を導出

するために上述したものと同じである。理解されるべきことに、第１ノード３は、第２ノード７と独立に第２ノード第２公開鍵を決定できる３７０。

【０１０３】

［第１ノード３において共通シークレットを決定する３８０］

第１ノード３は、次に、第１ノード第２秘密鍵（V_{2C}）及び決定した第２ノード第２公開鍵（P_{2S}）に基づき、共通シークレット（ＣＳ）を決定して良い３８０。共通シークレット（ＣＳ）は、第１ノード３により次式により決定されて良い。

$S = V_{2C} \times P_{2S}$（式１１）

＜第２ノード７において実行される方法４００＞

第２ノード７において実行される対応する方法４００が、ここで説明される。理解されるべきことに、これらのステップのうちの幾つかは、第１ノード３により実行された上述のステップと同様である。

【０１０４】

方法４００は、メッセージ（Ｍ）を通信ネットワーク５を介して第１ノード３から受信するステップ４１０を含む。これは、ステップ３１５において第１ノード３により送信されたメッセージ（Ｍ）を含んで良い。第２ノード７は、次に、メッセージ（Ｍ）に基づき生成器値（ＧＶ）を決定する４２０。第２ノード７により生成器値（ＧＶ）を決定するステップ４２０は、上述の第１ノードにより実行されるステップ３２０と同様である。本例では、第２ノード７は、第１ノード３と独立の、この決定するステップ４２０を実行する。

【０１０５】

次のステップは、第１ノードマスタ公開鍵（P_{1C}）及び生成器値（ＧＶ）に基づき、第１ノード第２公開鍵（P_{2C}）を決定するステップ４３０を含む。本例では、公開鍵は、基点（Ｇ）との楕円曲線点乗算により秘密鍵として決定されるので４３０’、第１ノード第２公開鍵（P_{2C}）は、式９と同様に次のように表すことができる。

$P_{2C} = V_{2C} \times G$（式１２．１）

$P_{2C} = P_{1C} + GV \times G$（式１２．２）

式１２．１及び１２．２の数学的証明は、式１０．１及び１０．２について上述したものと同じである。

【０１０６】

［第２ノード７が第１ノード３を認証する］

方法４００は、未確認第１ノード３が第１ノード３であることを認証するために、第２ノード７により実行されるステップを含んで良い。上述のように、これは、第１ノード３から第１署名メッセージ（ＳＭ１）を受信するステップ４４０を含む。第２ノード７は、次に、ステップ４３０で決定された第１ノード第２公開鍵（P_{2C}）により第１署名メッセージ（ＳＭ１）の署名を検証して良い４５０。

【０１０７】

デジタル署名の検証は、上述の楕円曲線デジタル署名アルゴリズム（Elliptic Curve Digital Signature Algorithm：ＥＣＤＳＡ）に従い行われて良い。重要なことに、第１ノード第２秘密鍵（V_{2C}）により署名された第１署名メッセージ（ＳＭ１）は、V_{2C}及びP_{2C}が暗号対を形成するので、対応する第１ノード第２公開鍵（P_{2C}）によってのみ正しく検証されるべきである。これらの鍵は、第１ノード３の登録で生成された第１ノードマスタ秘密鍵（V_{1C}）及び第１ノードマスタ公開鍵（P_{1C}）において確定するので、第１署名メッセージ（ＳＭ１）の検証は、第１署名メッセージ（ＳＭ１）を送信する未確認第１ノードが登録中と同じ第１ノード３であることを認証する基礎として使用できる。したがって、第２ノード７は、第１署名メッセージを検証するステップ（４５０）の結果に基づき、第１ノード３を認証するステップ（４６０）を更に実行して良い。

【０１０８】

［第２ノード７が共通シークレットを決定する］

方法４００は、第２ノード７が、第２ノードマスタ秘密鍵（V_{1S}）及び生成器値（ＧＶ）に基づき、第２ノード第２秘密鍵（V_{2S}）を決定するステップ４７０を更に含んで

良い。第１ノード３により実行されるステップ３３０と同様に、第２ノード第２秘密鍵（V_{2S}）は、次式に従い、第２ノードマスタ秘密鍵（V_{1S}）及び生成器値（ＧＶ）のスカラ加算に基づき得る。

$V_{2S}=V_{1S}+GV$（式１３．１）

$V_{2S}=V_{1S}+SHA-256(M)$（式１３．２）

第２ノード７は、次に、第１ノード３と独立して、次式に基づき、第２ノード第２秘密鍵（V_{2S}）及び第１ノード第２公開鍵（P_{2C}）に基づき、共通シークレット（ＣＳ）を決定して良い４８０。

$S=V_{2S}\times P_{2C}$（式１４）

［第１ノード３及び第２ノード７により決定された共通シークレット（ＣＳ）の証明］

第１ノード３により決定された共通シークレット（ＣＳ）は、第２ノード７において決定された共通シークレット（ＣＳ）と同じである。式１１及び式１４が同じ共通シークレット（ＣＳ）を提供することの数学的証明が、ここで記載される。

【０１０９】

第１ノード３により決定された共通シークレット（ＣＳ）を考えると、次のように式１０．１は式１１に代入できる。

$S=V_{2C}\times P_{2S}$（式１１）

$S=V_{2C}\times(V_{2S}\times G)$

$S=(V_{2C}\times V_{2S})\times G$（式１５）

第２ノード７により決定された共通シークレット（ＣＳ）を考えると、次のように式１２．１は式１４に代入できる。

$S=V_{2S}\times P_{2C}$（式１４）

$S=V_{2S}\times(V_{2C}\times G)$

$S=(V_{2S}\times V_{2C})\times G$（式１６）

ＥＣＣ代数学は可換性なので、次の通り式１５及び式１６は等価である。

$S=(V_{2C}\times V_{2S})\times G=(V_{2S}\times V_{2C})\times G$（式１７）

［共通シークレット（ＣＳ）及び秘密鍵］

共通シークレット（ＣＳ）は、ここで、第１ノード３と第２ノード７との間のセキュアな通信のために、対称鍵アルゴリズムにおいて、秘密鍵として又は秘密鍵の基礎として、使用できる。

【０１１０】

共通シークレット（ＣＳ）は、楕円曲線点（x_S，y_S）の形式であって良い。これは、ノード３、７により合意された標準的な公に知られた演算を用いて、標準的な鍵フォーマットに変換されて良い。例えば、x_S値は、AES_{256}暗号鍵として使用され得る２５６ビットの整数であって良い。これは、更に、１６０ビットの長さの鍵を必要とする任意のアプリケーションのために、ＲＩＰＥＭＤ１６０を用いて１６０ビットの整数に変換され得る。

【０１１１】

共通シークレット（ＣＳ）は、必要に応じて決定されて良い。重要なことに、共通シークレット（ＣＳ）はメッセージ（Ｍ）に基づき再決定できるので、第１ノード３は共通シークレット（ＣＳ）を格納する必要がない。幾つかの例では、使用されるメッセージ（Ｍ）は、マスタ秘密鍵のために要求されるのと同レベルのセキュリティを有しないデータストア１３、１７、１９（又は他のデータストア）に格納されて良い。幾つかの例では、メッセージ（Ｍ）は公に利用可能であって良い。

【０１１２】

しかしながら、幾つかのアプリケーションに依存して、共通シークレット（ＣＳ）が第１ノードマスタ秘密鍵（V_{1C}）と同じくらいセキュアに保たれるならば、共通シークレット（ＣＳ）は、第１ノードに関連付けられた第１データストア（Ｘ）に格納され得る。

【０１１３】

有利なことに、この技術は、単一のマスタキー暗号対に基づき、複数のセキュアな秘密

鍵に対応し得る複数の共通シークレットを決定するために使用できる。

【０１１４】

＜生成器値（鍵）の階層構造＞

例えば、一連の連続する生成器値（ＧＶ）が決定されて良い。ここで、各連続ＧＶは、前の生成器値（ＧＶ）に基づき決定されて良い。例えば、連続する専用鍵を生成するためにステップ３１０～３７０、４１０～４７０を繰り返す代わりに、ノード間の事前合意により、生成器値の階層構造を確立するために、前に使用された生成器値（ＧＶ）が両方のパーティにより繰り返し再ハッシュされ得る。実際に、メッセージ（Ｍ）のハッシュに基づく生成器値は、次世代生成器値（ＧＶ）のための次世代メッセージ（Ｍ鋳）になり得る。これを行うことは、計算されるべき共有シークレットの連続生成を可能にし、更なるプロトコルにより確立される送信、特に共通シークレットを生成する毎に複数のメッセージの送信の必要がない。次世代共通シークレット（ＣＳ’）は、以下のように計算できる。

【０１１５】

先ず、第１ノード３及び第２ノード７の両者が、次世代生成器値（ＧＶ’）を独立して決定する。これは、ステップ３２０及び４２０と同様であるが、次式により適応される。

Ｍ’＝ＳＨＡ－２５６（Ｍ）（式１８）

ＧＶ’＝ＳＨＡ－２５６（Ｍ’）（式１９．１）

ＧＶ’＝ＳＨＡ－２５６（ＳＨＡ－２５６（Ｍ））（式１９．２）

第１ノード３は、次に、次世代の第２ノード第２公開鍵（P_{2S}’）及び第１ノード第２秘密鍵（V_{2C}’）を上述のステップ３７０及び３３０と同様であるが次式により適応され、決定して良い。

$P_{2S}' = P_{1S} + GV' \times G$（式２０．１）

$V_{2C}' = V_{1C} + GV'$（式２０．２）

第２ノード７は、次に、次世代の第１ノード第２公開鍵（P_{2C}’）及び第２ノード第２秘密鍵（V_{2S}’）を上述のステップ４３０及び４７０と同様であるが次式により適応され、決定して良い。

$P_{2C}' = P_{1C} + GV' \times G$（式２１．１）

$V_{2S}' = V_{1S} + GV'$（式２１．２）

第１ノード３及び第２ノード７は、次に、それぞれ、次世代共通シークレット（ＣＳ’）を決定して良い。特に、第１ノード３は、次式により次世代共通シークレット（ＣＳ’）を決定する。

$CS' = V_{2C}' \times P_{2S}'$（式２２）

第２ノード７は、次式により次世代共通シークレット（ＣＳ’）を決定する。

$CS' = V_{2S}' \times P_{2C}'$（式２３）

更なる世代（ＣＳ’’、ＣＳ’’’、等）は、チェーン階層構造を生成するために同じ方法で計算できる。この技術は、第１ノード３及び第２ノード７の両者が元のメッセージ（Ｍ）又は最初に計算された生成器値（ＧＶ）、及びそれがどのノードに関連するか、を見失わないことを要求する。これは公に知られた情報なので、この情報の保有に関してセキュリティ問題は存在しない。したがって、この情報は、（ハッシュ値を公開鍵にリンクする）「ハッシュテーブル」に保持され、（例えばＴｏｒｒｅｎｔを用いて）ネットワーク５に渡り自由に配布されて良い。さらに、階層構造内の任意の個々の共通シークレット（ＣＳ）が今までに解決されていない場合、これは、秘密鍵V_{1C}、V_{1S}がセキュアなままであるならば、階層構造の中の任意の他の共通シークレットのセキュリティに影響しない。

【０１１６】

＜鍵の木構造＞

上述のようなチェーン（線形）階層構造と同様に、木構造の形式の階層構造が生成できる。木構造によると、認証鍵、暗号鍵、署名鍵、支払鍵、等のような異なる目的の様々な鍵が決定されて良い。それにより、これらの鍵は、全て単一のセキュアに維持されるマスタキーにリンクされる。これは、種々の異なる鍵を有する木構造９０１を示す図１２に図示される。これらの各々は、別のパーティと共有されるシークレットを生成するために使

用できる。枝分かれする木は、幾つかの方法で達成でき、それらのうちの３つが以下に記載される。

【０１１７】

（ｉ）マスタキー・スポーニング

チェーン階層構造では、乗算した再ハッシュしたメッセージを元のマスタキーに加算することにより、各々の新しい「リンク」（公開／秘密鍵ペア）が生成される。例えば（明確性のため、第１ノード３の秘密鍵のみを示す）、

$V_{2C}=V_{1C}+SHA-256(M)$（式２４）

$V_{2C}'=V_{1C}+SHA-256(SHA-256(M))$（式２５）

$V_{2C}''=V_{1C}+SHA-256(SHA-256(SHA-256(M)))$（式２６）

等である。

【０１１８】

枝を生成するために、任意の鍵がサブマスタキーとして使用できる。例えば、V_{2C}'は、正規のマスタキーに対して行われるように、ハッシュを加算することにより、サブマスタキー（V_{3C}）として使用できる。

【０１１９】

$V_{3C}=V_{2C}'+SHA-256(M)$（式２７）

サブマスタキー（V_{3C}）は、それ自体が、次世代鍵（V_{3C}'）を有して良い。例えば次式の通りである。

【０１２０】

$V_{3C}'=V_{2C}'+SHA-256(SHA-256(M))$（式２８）

これは、図１３に示すマスタキー・スポーニング（spawning）方法を用いて、木構造９０３を提供する。

【０１２１】

（ｉｉ）論理的関連付け

この方法では、木の中の全てのノード（公開／秘密鍵ペア）は、チェーンとして（又は任意の他の方法で）生成され、木の中のノード間の論理的関係は、ポインタを用いて木の中の各ノードが木の中の自身の親ノードに単純に関連付けられるテーブルにより維持される。したがって、ポインタは、セッションの共通シークレットキー（ＣＳ）を決定するために、関連する公開／秘密鍵ペアを決定するために使用できる。

【０１２２】

（ｉｉｉ）メッセージ多様性

新しい公開／秘密鍵ペアは、チェーン又は木の任意のポイントに新しいメッセージを導入することにより、生成できる。メッセージ自体は、任意であって良く、又は何らかの意味若しくは関数を伝達して良い（例えば、それは、「現実の」銀行口座番号に関連して良い、等である）。新しい公開／秘密鍵ペアを形成するためのこのような新しいメッセージがセキュアに保持されることが望ましい場合がある。

【０１２３】

＜コード化スキーム＞

トランザクションのメタデータは、ブロック外の文書に格納された命令にアクセスするために使用されて良い。この文書は、「取引（コントラクト、contract）」と呼ばれることがある。取引を参照するために使用されるメタデータは、様々な方法でフォーマット化できる。しかしながら、適切なコード化スキームがここに記載される。

【０１２４】

取引の定める権利が取引の保持者又は所有者に贈与される場合、取引は転送可能である。非転送可能取引の一例は、参加者が指名される取引である。つまり、権利が、取引の保持者ではなく特定の指名されたエンティティに贈与される。転送可能取引だけが、このコード化スキームで議論される。

【０１２５】

トークンは、取引により贈与される権利を詳述する又は定める特定取引を表す。本発明に従い、トークンは、ビットコイントランザクションの形式の取引の表現である。

【０１２６】

このコード化方法は、３つのパラメータ又はデータアイテムを有するメタデータを使用する。このデータは、以下を示し得る：

ｉ）取引の下で利用可能な持分（share、株）の量（これは、本願明細書で「ＮｕｍＳｈａｒｅｓ」と呼ばれることがある）、

ｉｉ）送り手から少なくとも１つの受け手へ転送されるべき転送単位の量（これは、「ＳｈａｒｅＶａｌ」と呼ばれることがある）、

ｉｉｉ）転送単位の量の値を計算するための因子（これは、「ペギングレート（pegging rate）」と呼ばれることがある）。

【０１２７】

このコード化スキームの利点は、上述の３つのパラメータだけを用いて、ブロックチェーン上のトークンとして取引をカプセル化又は表現するために使用できることである。実際に、取引は、これらの３つのデータアイテムのうちの最小限を用いて指定できる。このコード化スキームは任意の種類の転送可能取引のために使用可能なので、共通アルゴリズムが考案され適用され得る。これらのメタデータアイテムの更なる詳細は、以下に提供される。

【０１２８】

分割可能トークンは、トランザクションアウトプット上の値が、複数のトークンに渡り（つまり、複数のトランザクションに渡り）割り当てられるより小さな量に細分化され得るものである。典型は、トークン化されたフィアット通貨である。分割可能取引は、非ゼロペギングレート（Pegging Rate）を指定する取引として定められる。分割可能取引では、トランザクションアウトプットの中で転送されるトークン化された値は、ペギングレートにより基礎ビットコイン（bitcoin：ＢＴＣ）値に結び付けられる。つまり、取引は、ペギングレートの観点で、保持者の権利を指定する。非分割可能トークンでは、ペギングレートが存在せず、取引は、固定値の観点で、保持者の権利を指定する（例えば、「この取引は、正確に＄１０００で換金できる」という無記名債券、又は「この取引は１ヘアカットと交換可能である」という商品引換券のように）。非分割可能取引では、基礎トランザクションＢＴＣ値は取引値と無関係である。

【０１２９】

表現「基礎ＢＴＣ値」は、トランザクションアウトプットに付加されるビットコイン額（ＢＴＣ）を表す。ビットコインプロトコルでは、各トランザクションアウトプットは、有効と考えられる非ゼロＢＴＣ額を有しなければならない。実際には、ＢＴＣ額は、記述時に現在５４６サトシに設定される設定最小値（「ダスト（dust）」）より大きくなければならい。１ビットコインは、１００百万サトシに等しいと定められる。ビットコイントランザクションは本願明細書では所有権の交換を助ける手段としてのみ使用されるので、実際の基礎ＢＴＣ額は任意である。真の値は、取引仕様の中にある。理論上、全てのトークンは、ダストにより伝達され得る。

【０１３０】

本発明のコード化スキームに従い、具体的に、分割可能トークンでは、基礎ＢＴＣ値は次の意味：ペギングレートにより取引値との関係を運ぶ、を有する。ペギングレートは、それ自体任意であり、基礎ＢＴＣ額を小さく保つために選択される。単にダストを有する基礎となる各トークントランザクションではなく、ペギングレートを使用する理由は、本発明のプロトコルが可視性を実現するからである。トークンが幾つかの小さな額のトランザクションアウトプットに分けられるとき、元の取引を調整する必要がない。むしろ、各細分化トークンの取引値は、単に、ペギングレート及び基礎ＢＴＣ値の細分化された額に基づき計算される。

【０１３１】

限定トークンは、ＮｕｍＳｈａｒｅｓと呼ばれる量により定められる固定非ゼロの株数

により合計発行値が固定された（又は「限定された」）トークンである。したがって、限定取引の下では更なる株は発行されない。例えば、競走馬の共同所有権のための取引は、競走馬の１００％に限定される（例えば、それぞれ１％で１００個の株、又はそれぞれ１０％で１０個の株、等）。非限定取引は、例えば要求額のフィアット通貨を彼らの引当金（Ｒｅｓｅｒｖｅ Ａｃｃｏｕｎｔ）に追加することにより、発行人が株の更なる発行を引き受け可能であることを意味する。ＮｕｍＳｈａｒｅｓは、全ての取引に明示的に記載されなければならない。限定取引は、ＮｕｍＳｈａｒｅｓ＞０を有しなければならない。非限定取引は、ＮｕｍＳｈａｒｅｓ＝０を設定することにより示される。

【０１３２】

典型的な例は、準備預金口座に保持される合計値が存在する約束手形（つまり未償還トークン）の合計値に一致するようにする、通貨準備（金準備と類似する）である。この概念は、通貨準備を超えて、在庫ストックにまで拡大する。例えば、認可印刷Ｔシャツトークンの発行人は、１０，０００枚のＴシャツの在庫で開始して良く、これらの１０，０００枚のＴシャツを表す分割可能トークンを発行して良い（ここで、各株＝１枚のＴシャツを表す）。元のトークンは細分化でき、各細分化トークンは、ペギングレートにより定められるトランザクションアウトプットの基礎ＢＴＣ値に従い、Ｔシャツの枚数に償還可能である。しかしながら、需要が増大する場合、発行人は更なる株を発行することを決定して良い（つまり、更に１０，０００枚だけ流通株数を増大する）。このような場合には、更なる発行を引き受けるために、発行人の準備講座（つまり、在庫倉庫）に更に１０，０００枚のＴシャツを預けることが、発行人の義務である。したがって、在庫にあるＴシャツの合計枚数は、常に、合計未償還株数である（ここで、在庫は「準備口座」として作用する）。

【０１３３】

ペギングレートは、分割可能取引にのみ適用される。ここで、（ＳｈａｒｅＶａｌと呼ばれる量により表される）株の値は、基礎ＢＴＣ額に固定される。例えば、取引は、発行人が基礎１ＢＴＣ毎に＄１０，０００のレートでトークンを償還することを約束すると指定して良い。これは、（例えば）１５，４００サトシのトークン化された基礎アウトプット値を有するトランザクションが＄１．５４で償還可能であることを意味し得る。ペギングレートについて０の値は、取引が非分割可能であることを示す（つまり、無記名債権のように、全体のみが転送可能である）。ペギングレートが０に設定されると（非分割可能トークンを意味する）、基礎ＢＴＣ値は、取引値と無関係であり、任意の額に設定可能である。通常、この場合には、運用コストを最小化するために、基礎ＢＴＣ額を可能な限り小さく保つ（つまり、ダストに設定する）ことが望ましい。

【０１３４】

ＮｕｍＳｈａｒｅｓは、（限定）取引の下で利用可能な合計（固定）株数である。限定取引では、ＮｕｍＳｈａｒｅｓは、ゼロより大きい全体数でなければならない。非限定取引では、いつでも（引き受けられるならば）より多くの株が発行可能なので、ＮｕｍＳｈａｒｅｓは固定されない。これは、値を０に設定することにより示される。

【０１３５】

株は、転送単位として定められ、ＳｈａｒｅＶａｌはその単位の値である。例えば、フィアット通貨では、転送単位は１セントに設定されて良い。あるいは、例えば、転送単位は５０セントに設定されて良く、この場合には、転送は５０セントの「ロット」でのみ実行できる。ＳｈａｒｅＶａｌは、パーセンテージとしても表現できる。例えば、ブリーダが競走馬を１０個の等しい株で売りたいと望む場合、ＳｈａｒｅＶａｌ＝０を設定１０％である。ＳｈａｒｅＶａｌは０より大きくなければならず、且つ取引において定められなければならない。

【０１３６】

ＴｏｔａｌＩｓｓｕａｎｃｅは、発行された株の合計値を表す。この値は限定取引にのみ関連する。非限定取引については、発行は固定されず、より多くの株が発行されて良い。株がパーセンテージとして表現される場合、定義によりＴｏｔａｌＩｓｓｕａｎｃｅ＝

０を設定１００％である。
【０１３７】
限定取引では、ＮｕｍＳｈａｒｅｓ、ＳｈａｒｅＶａｌ、及びＴｏｔａｌＩｓｓｕａｎｃｅは、以下のように関連する。
ＮｕｍＳｈａｒｅｓ×ＳｈａｒｅＶａｌ＝ＴｏｔａｌＩｓｓｕａｎｃｅ
ＴｏｔａｌＩｓｓｕａｎｃｅの０の値は、非限定取引であることを意味する。非限定取引の一例は、フィアット通貨である（したがって、ＴｏｔａｌＩｓｓｕａｎｃｅは０に設定される）。限定取引の例は、（ｉ）限定版記念硬貨（１０００個鋳造され、１株＝１硬貨である）、ＴｏｔａｌＩｓｓｕａｎｃｅ＝１０００×１＝１０００個の硬貨、（ｉｉ）入場券のある会場の席、ＴｏｔａｌＩｓｓｕａｎｃｅ＝利用可能な合計席数、である。
【０１３８】
流通は、未使用トークンの合計値として定められる（つまり、ＵＴＸＯ（未使用トランザクションアウトプット）の中のトランザクションにより決定される）。全部の未使用トランザクションの完全な集合は、全てのビットコインノードに利用可能なリストの中に保持される。例えば、発行人が最初に＄１０，０００をフィアット通貨型のトークンとして発行し、時間を経て＄５５００の価値のトークンが償還された場合、流通＝＄４５００である（未償還トークンの値である）。この値は、関連する準備口座の差引残高に調整されるべきである。
【０１３９】
＜本発明の実施形態と共に使用するのに適する計算リソース（「エージェント」）の説明のための例＞
本発明は、所望の処理の自動化された態様を実行するために、適切に構成された計算リソース（ここでは「エージェント」）を利用できる。適切且つ好適なエージェントの一例が以下に提供されるが、他の実装が使用されて良い。
【０１４０】
エージェントは、チューリング（Turing）機械の実装において非消去可能テープとしてブロックチェーンを使用して、ブロックチェーンと連携して動作して良い。このエージェントは、ブロックチェーンネットワークと並列に実行し、（ループ）処理の実行を監督し及び扱う。ループ処理は、例えば装置又はシステムの処理又は制御の自動化のような所与のタスクを実行するよう設計される。この並列リソースは、ブロックチェーンの状態を監視し、トランザクションをブロックチェーンに書き込ませることができる。ある意味で、これは、ブロックチェーンをチューリング機械の非消去可能テープとして利用し、以下の定義及び特徴を有する。
１．ブロックチェーンは、チューリング機械のテープとして作用する。ブロックチェーンの中の各トランザクションは、テープ上のセルを表す。このセルは、有限なアルファベットからの記号を含み得る。
２．テープヘッドは、ブロックチェーン上に既に書き込まれているブロックから情報を読み出すことができる。
３．テープヘッドは、多くのトランザクションを含む新しいブロックを、ブロックチェーンの終わりに書き込むことができる。しかしながら、それらは、既に存在するブロックに書き込むことができない、このように、ブロックチェーンテープは非消去可能である。
４．各トランザクションのマルチシグネチャのＰ２ＳＨ（pay－to－script－hash）トランザクションの部分として格納され得る。
【０１４１】
エージェントに重要な機能は、ブロックチェーンの現在状態を監視する自動エンティティとして作用することである。これは更に、任意のオフブロックソースから、信号又は入力を受信できる。ブロックチェーン状態及び／又は受信した入力に依存して、エージェントは特定動作を実行して良い。エージェントは、どの動作が実行されるべきかを決定する。これらは、「現実世界」（つまり、オフブロック）の中の作用、及び／又（新しいトランザクションを生成する及びブロードキャストするような）はブロックチェーンに対する

作用を含んで良く又は含まなくて良い。エージェントが取る作用は、ブロックチェーン状態によりトリガされて良い。エージェントは、更に、ビットコインネットワークにブロードキャストされるべき次のトランザクションセットについて決定し、後にブロックチェーンに書き込んで良い。

【０１４２】

エージェントの作用は、並列に且つ同時にブロックチェーン（例えば、ビットコイン）ネットワークに対して実行する。ある意味で、これは、ブロックチェーン（例えば、ビットコイン）スクリプトの機能を格納する。この連続監視は、結合されたエージェント及びブロックチェーンシステムチューリング完全（Turing Complete）を作成する「ループ」制御フロー構成を実施する。

【０１４３】

チューリング機械は、以下の２つのスタックを含む。

・データスタック：これは、上述のようにブロックチェーンにより表される。

・制御スタック：これは、エージェント機能により表される。これは、繰り返し制御フロー機能に関連する情報を格納する。

【０１４４】

制御スタックのデータスタックからの分離は、無限ループがビットコイン中核で生じることを防ぎ、サービス拒否攻撃を軽減するという利点を提供する。

【０１４５】

エージェントは、任意の種類のループ構成（例えば、ＦＯＲ－ＮＥＸＴ、ＲＥＰＥＡＴ　ＵＮＴＩＬ、等）によりループ可能なサブルーチンを管理し及び実行する。本願明細書に記載の説明のための実施形態は、「繰り返し（ｒｅｐｅａｔ）」構成の一例を用いる処理を含む。ユーザは、インデックス（ｉ）及び限界（Ｊ）を指定して良い。これらはそれぞれ、現在の反復番号（標準的に０から開始してカウントされる）、及び繰り返しループの合計反復数、を表す。

【０１４６】

各反復について、

１．インデックスが１だけ増大する。終了条件については、インデックスが限界に達すると、反復は停止する。

２．「if condition then action（［条件］ならば［動作］する）」（ＩＣＴＡ）文を含むコードブロックが実行される。動作は、ブロックチェーン上の又は外の任意の動作であって良い。

３．このサブルーチンの暗号ハッシュが計算される。これは、トランザクションの部分としてブロックチェーンに格納できる。ハッシュは各コードにユニークなので、どのコードが使用されているかの検証を可能にする。

【０１４７】

ループ本体は、コードブロックを含む。各コードブロックは、「if condition then action（［条件］ならば［動作］する）」（ＩＣＴＡ）文を含む。これは、以下に一致するトランザクションについて、ブロックチェーンの現在状態を監視する。

・開始又はトリガ条件（例えば、特定日に達したとき）、

・繰り返し条件（つまり、前の反復に関連付けられたメタデータ又はハッシュ）、

・停止条件（つまり、ループの最後の反復）。

【０１４８】

ＩＣＴＡ文は、エージェントが、ブロックチェーンの現在状態に基づき、次のトランザクションについて決定することを可能にする。次のトランザクションを生成することは、トランザクションをビットコインネットワークにブロードキャストすること、及び新しいトランザクションをブロックチェーンに書き込むことを含む。これは、この反復が実行されたことの記録として作用する。トランザクションがブロックチェーンに書き込まれると、マネジャは、続いて、前の反復が実行されブロックチェーンに書き込まれたことを知り、次の反復を実行するだろう。後者は、インデックス（ｉ）がコードブロックの中で指定

された限度（Ｊ）に達するとき、繰り返しループが存在するまで継続する。
【０１４９】
　各トランザクションは、再利用可能な方法で、ブロックチェーンに保存される。ビットコイン実装では、トランザクションの中の各署名は、ＳＩＧＨＡＳＨフラグを付加される。このフラグは、異なる値を取ることができる。これらの値の各々は、この署名の所有者の関与無しに、トランザクションの他の部分が変更され得るか否かを示す。再利用可能トランザクションは、トランザクションインプットのうちの１つの中にＳＩＧＨＡＳＨフラグ「ＳｉｇＨａｓｈ＿ＡｎｙｏｎｅＣａｎＰａｙ」を有する。これは、誰もがトランザクションのインプットに貢献することを許容する。このパラメータは、エージェントのＩＣＴＡ関数が、複数回、異なるインプットで、実行され且つ繰り返されることを可能にする。この関数の使用は、例えば再利用トランザクションの複製により、認可パーティに制限できる。
【０１５０】
　ＩＣＴＡコードブロックの「If condition」部分は、任意の種類の条件を監視できる。これは、他のプログラミング言語（例えば、Ｃ、Ｃ＋＋、Ｊａｖｅ）と同様であり、ブロックチェーンに格納された情報に限定されない。例えば、これは、日付及び時間（つまり、特定の日時に達したとき）を監視し、又は天気（つまり、気温が１０℃より低く且つ雨が降っているとき）を監視し、取引又は信用の条件（つまり、企業Ａが企業Ｂを買うとき）を監視し得る。
【０１５１】
　ＩＣＴＡコードブロックの「Then action」部分は、多数の動作を実行できる。本発明は、取り得る動作の数又は種類に関して限定されない。動作は、ブロックチェーン上のトランザクションに限定されないが、動作に関連するメタデータを含むトランザクションは、ブロックチェーンに書き込まれて良い。
【０１５２】
　メタデータは、任意の形式であり得る。しかしながら、一実施形態では、メタデータは、動作に関連するより多くのデータ又は指示を含むファイルへのハイパーリンクを格納して良い。メタデータは、動作に関連するより多くのデータ又は指示を含むハッシュテーブルへのハイパーリンク、及びハッシュテーブルに対する検索キーとして作用する動作のハッシュの両方を格納して良い。
【０１５３】
　エージェントの制御スタックは、各ユーザの必要に特化した多数の方法で実装できる。例えば、制御スタックの繰り返しループは、任意のチューリング完全言語に基づき得る。言語の１つの可能な選択は、Ｆｏｒｔｈ型スタックに基づく言語である。この言語を使用する利点は、制御スタックを、既知であり且つ広く使用されているビットコインスクリプトとプログラミング型において一貫性を保つことである。
【０１５４】
　＜ビットコインスクリプトの交互形式スタック（Alternate Stack）をデータ記憶空間として使用する＞
　ビットコインスクリプトは、コマンド、更に呼び出されるオペコード、を含む。これらは、ユーザが、データを「ａｌｔｓｔａｃｋ」として知られる交互形式スタックに移動させることを可能にする。
【０１５５】
　オペコードは、次の通りである。
・ＯＰ＿ＴＯＡＬＴＳＴＡＣＫ。これは、データを主スタックの最上部からａｌｔｓｔａｃｋの最上部に移動させる。
・ＯＰ＿ＦＲＯＭＡＬＴＳＴＡＣＫ。これは、データをａｌｔｓｔａｃｋの最上部から主スタックの最上部に移動させる。
【０１５６】
　これは、計算機にデータを格納させる「記憶（memory）」機能と同様に、中間計算ス

テップからのデータをａｌｔｓｔａｃｋに格納させる。一実施形態では、ａｌｔｓｔａｃｋは、小さな計算タスクを解決するようビットコインスクリプトを構成するために、及び結果を計算に返すために、使用される。

【０１５７】

＜エージェントを管理するためにコードレジスタを使用する＞

エージェントは、また、自身の所有し且つ実行する全てのコードのレジストリを管理する。このレジストリは、特定キーを特定値にマッピングするルックアップテーブル又は辞書のように構造化される。キー及び値ペアは、それぞれ、コードブロックのハッシュ（H_1）及びコードが格納された場所のＩＰｖ６アドレスにより表される。キーH_1を用いてコードブロックを読み出すために、ルックアップテーブルが使用されて、関連する値（これは、コードの格納されている場所である）を読み出し、及びそれに応じてソースコードを読み出す。コードレジストリの実装は変化し得る。

【０１５８】

＜エージェントのコードのトランザクションメタデータ、及びループの再スポーニング＞

特定の反復でエージェントのループを再スポーニングするために必要な情報は、ブロックチェーンに記録されたトランザクションの中にメタデータとして格納される。

【０１５９】

このように、ブロックチェーン上のトランザクションは、エージェント上で実行されているループの所与の反復に関する情報を格納し、又はそれへのアクセスを提供する。この情報は、ループに関連付けられた任意の変数の値、例えばインデックスｉ、及び任意の他の必要な情報、例えばコードブロック内で使用されるパラメータの値又は更に要求される情報がアクセス可能な場所の位置関連データ、を含み得る。

【０１６０】

メタデータ自体は、トランザクションの中のマルチシグネチャのＰ２ＳＨ（pay－to－script－hash）スクリプトの部分として格納される。トランザクションと共に記録されたメタデータは、コードが過去にどのように実行されたかのオーディットトレイルを記録する能力も与える。

【０１６１】

エージェントが各反復で繰り返しループコードブロックを再スポーニングできる幾つかの方法が存在する。コードブロックは、エージェント自体にハードコードされて良く、又は秘密に若しくは公に利用可能なファイルに格納でき、又は、秘密若しくは公開ハッシュテーブルファイル上のエントリとして格納され得る、又はこれらの組み合わせである。コードブロックは、ハードコードされた変数に固定され、又は固定であるが、移植可能なパラメータを含み得る。パラメータは、任意のデータフォーマットの単一値であって良く、又は小さなコードチャンクであって良く、又はそれらの組み合わせであって良い。パラメータは、トランザクション（例えば、ビットコイントランザクション）の中のメタデータから、又は内部データベース又は秘密／公開ファイル又はハッシュテーブル又はこれらの任意の組み合わせのような外部ソースから、直接に該パラメータを読み出すことにより移植され得る。パラメータ値の外部ソースへのポインタは、トランザクションの中のメタデータに格納されて良い。

【０１６２】

以下のステップは、エージェントがｉ番目の反復で繰り返しループコードブロックをどのように再スポーニングできるかの一例を提供する。本例では、コードレジストリは、ハッシュテーブルである。これにより、ハッシュ値がテーブルの検索キーとして作用し、トランザクション上のメタデータに格納される。

１．エージェントは、コードレジストリ内のエントリに一致するコードブロックのハッシュを含むトランザクションについて、ブロックチェーンを監視する。

２．エージェントは、対応するハッシュ（H_1）を含むトランザクションを見付ける。

３．エージェントは、「メタデータ－ＣｏｄｅＨａｓｈ」を読み出し、H_1を得るためにＣｏｄｅＨａｓｈフィールドを得て、H_1を用いてコード（C_1）を読み出す。ＲＩＰＥ

ＭＤ－１６０（ＳＨＡ２５６（Ｃ$_1$））がＨ$_1$に等しい場合、コードは変化しておらず、安全に次のステップに進める。
４．エージェントは、インデックスＩを格納する「メタデータ－ＣｏｄｅＨａｓｈ」を読み出し、ｉ番目の反復でコードを再スポーニングする。言い換えると、ループが、適切な反復で「リロード」される。
５．メタデータの発生元を検証するために、ユーザの署名がＰ２ＳＨコマンドに含まれる。
６．これらのデータがループのこの反復のために必要な場合、エージェントは、「メタデータ－ＯｕｔｐｕｔＨａｓｈ」及び「メタデータ－ＯｕｔｐｕｔＰｏｉｎｔｅｒ」を読み出し、前のステップのアウトプットを検索する。

【０１６３】

留意すべきことに、上述の実施形態は、本発明を限定するのではなく、当業者は添付の請求項により定められる本発明の範囲から逸脱することなく多数の代替の実施形態を考案できる。請求項中、括弧内に記載された如何なる参照符号も、請求項を制限すると見なされるべきではない。用語「有する（comprising又はcomprises）」等は、全体としていかなる請求項中に及び明細書に列挙された以外の要素又はステップの存在を排除するものではない。本願明細書において、「有する（comprises）」は「含む（includes）又は構成される（consists of）」を意味し、「有する（comprising）」は「含む（including）又は構成される（including of）」を意味する。要素の単数の参照は、該要素の複数の存在を排除するものではなく、逆も同様である。本発明は、複数の別個の要素を有するハードウェアにより又は適切にプログラムされたコンピュータにより、実施され得る。複数の手段を列挙している装置の請求項では、これらの複数の手段は、１つの同一のハードウェア要素により実装することができる。特定の量が相互に異なる従属請求項に記載されるという事実は、これらの量の組合せが有利に用いることが出来ないことを示すものではない。

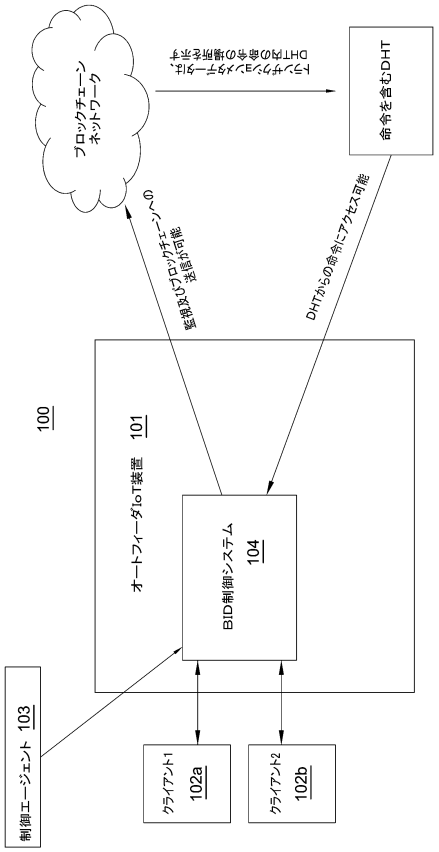
【符号の説明】

【０１６４】

１０１　オートフィーダIoT装置
１０２　クライアント
１０３　制御エージェント
１０４　BID制御システム

【図面】

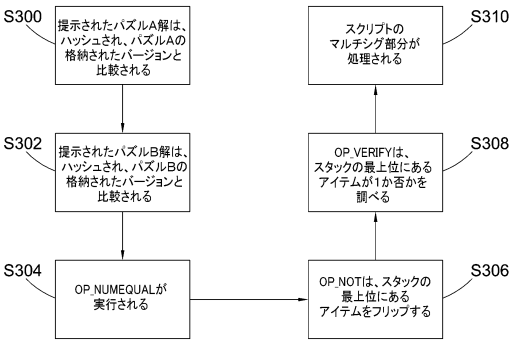
【図１】



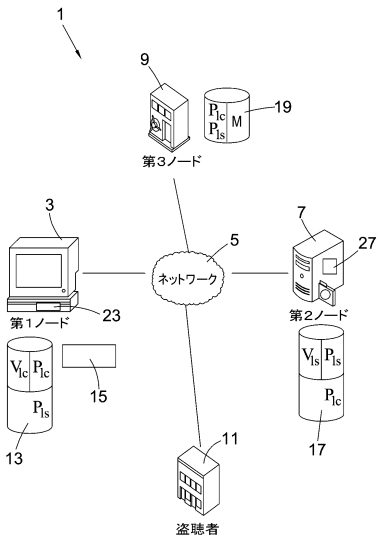
【図２】

A	B	X
0	0	0
1	0	1
0	1	1
1	1	0

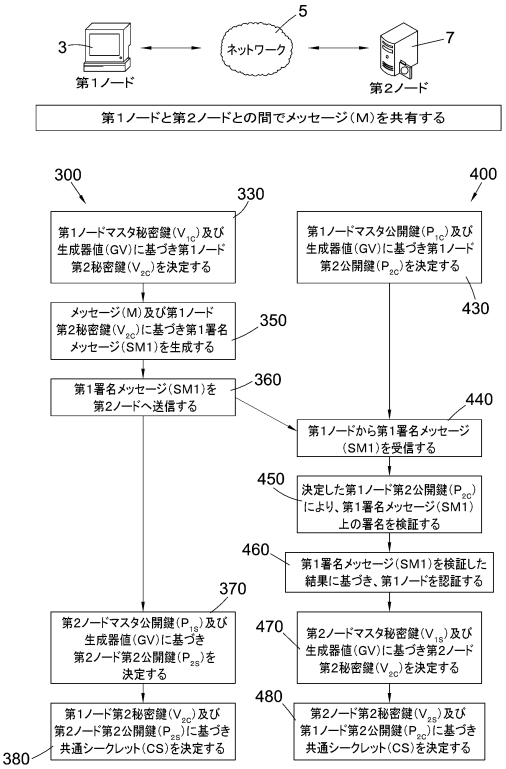
【図３】



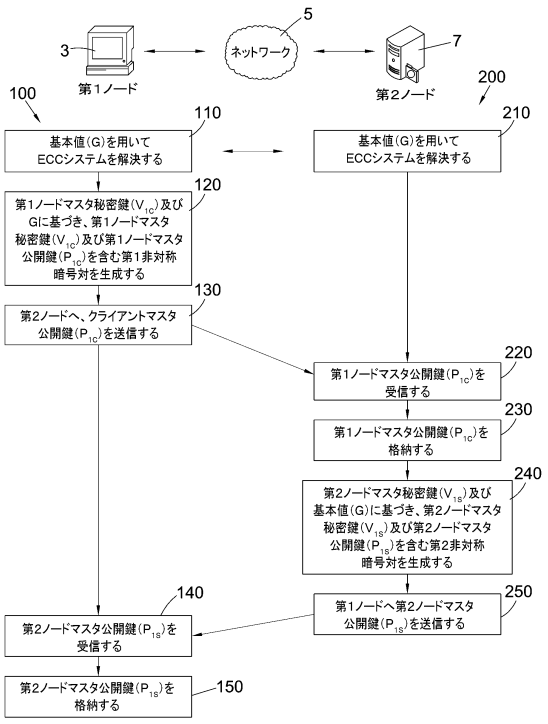
【図４】



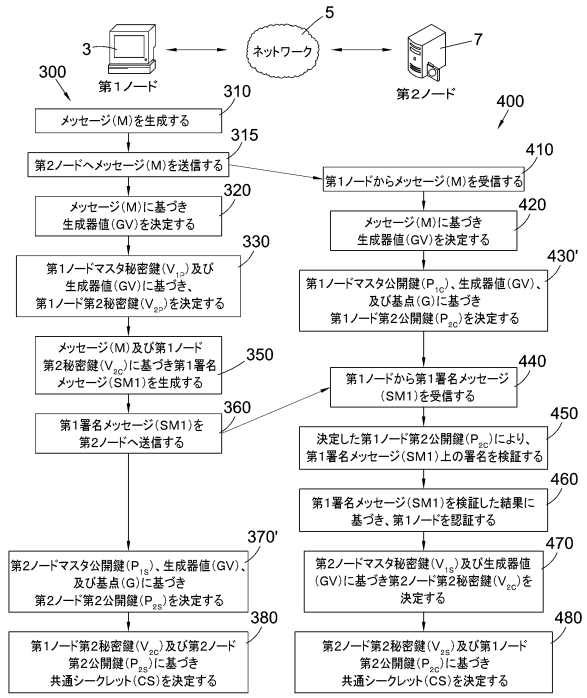
【図５】


3
第１ノード
5
ネットワーク
7
第２ノード
第１ノードと第２ノードとの間でメッセージ（M）を共有する
300
330
第１ノードマスタ秘密鍵（V_{1C}）及び生成器値（GV）に基づき第１ノード第２秘密鍵（V_{2C}）を決定する
400
第１ノードマスタ公開鍵（P_{1C}）及び生成器値（GV）に基づき第１ノード第２公開鍵（P_{2C}）を決定する
430
メッセージ（M）及び第１ノード第２秘密鍵（V_{2C}）に基づき第１署名メッセージ（SM1）を生成する
350
第１署名メッセージ（SM1）を第２ノードへ送信する
360
440
第１ノードから第１署名メッセージ（SM1）を受信する
450
決定した第１ノード第２公開鍵（P_{2C}）により、第１署名メッセージ（SM1）上の署名を検証する
460
第１署名メッセージ（SM1）を検証した結果に基づき、第１ノードを認証する
370
第２ノードマスタ公開鍵（P_{1S}）及び生成器値（GV）に基づき第２ノード第２公開鍵（P_{2S}）を決定する
470
第２ノードマスタ秘密鍵（V_{1S}）及び生成器値（GV）に基づき第２ノード第２秘密鍵（V_{2C}）を決定する
480
第１ノード第２秘密鍵（V_{2C}）及び第２ノード第２公開鍵（P_{2S}）に基づき共通シークレット（CS）を決定する
第２ノード第２秘密鍵（V_{2S}）及び第１ノード第２公開鍵（P_{2C}）に基づき共通シークレット（CS）を決定する
380

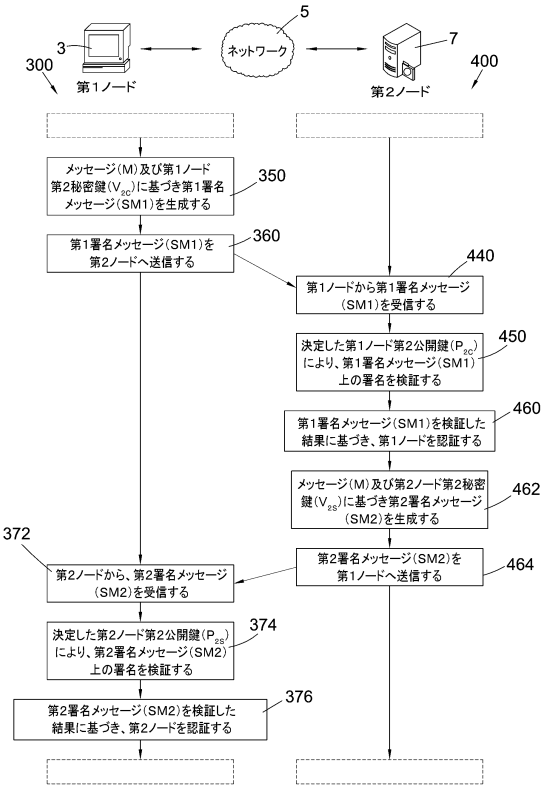
【図６】


3
第１ノード
5
ネットワーク
7
第２ノード
100
200
基本値（G）を用いてECCシステムを解決する
110
基本値（G）を用いてECCシステムを解決する
210
第１ノードマスタ秘密鍵（V_{1C}）及びGに基づき、第１ノードマスタ秘密鍵（V_{1C}）及び第１ノードマスタ公開鍵（P_{1C}）を含む第１非対称暗号対を生成する
120
第２ノードへ、クライアントマスタ公開鍵（P_{1C}）を送信する
130
第１ノードマスタ公開鍵（P_{1C}）を受信する
220
第１ノードマスタ公開鍵（P_{1C}）を格納する
230
第２ノードマスタ秘密鍵（V_{1S}）及び基本値（G）に基づき、第２ノードマスタ秘密鍵（V_{1S}）及び第２ノードマスタ公開鍵（P_{1S}）を含む第２非対称暗号対を生成する
240
第１ノードへ第２ノードマスタ公開鍵（P_{1S}）を送信する
250
第２ノードマスタ公開鍵（P_{1S}）を受信する
140
第２ノードマスタ公開鍵（P_{1S}）を格納する
150

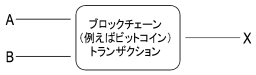
【図７】


3
第１ノード
5
ネットワーク
7
第２ノード
300
310
メッセージ（M）を生成する
315
第２ノードへメッセージ（M）を送信する
400
410
第１ノードからメッセージ（M）を受信する
320
メッセージ（M）に基づき生成器値（GV）を決定する
420
330
第１ノードマスタ秘密鍵（V_{1C}）及び生成器値（GV）に基づき、第１ノード第２秘密鍵（V_{2P}）を決定する
430'
第１ノードマスタ公開鍵（P_{1C}）、生成器値（GV）、及び基点（G）に基づき第１ノード第２公開鍵（P_{2C}）を決定する
350
メッセージ（M）及び第１ノード第２秘密鍵（V_{2C}）に基づき第１署名メッセージ（SM1）を生成する
440
第１ノードから第１署名メッセージ（SM1）を受信する
360
第１署名メッセージ（SM1）を第２ノードへ送信する
450
決定した第１ノード第２公開鍵（P_{2C}）により、第１署名メッセージ（SM1）上の署名を検証する
460
第１署名メッセージ（SM1）を検証した結果に基づき、第１ノードを認証する
370'
第２ノードマスタ公開鍵（P_{1S}）、生成器値（GV）、及び基点（G）に基づき第２ノード第２公開鍵（P_{2S}）を決定する
470
第２ノードマスタ秘密鍵（V_{1S}）及び生成器値（GV）に基づき第２ノード第２秘密鍵（V_{2C}）を決定する
380
第１ノード第２秘密鍵（V_{2C}）及び第２ノード第２公開鍵（P_{2S}）に基づき共通シークレット（CS）を決定する
480
第２ノード第２秘密鍵（V_{2S}）及び第１ノード第２公開鍵（P_{2C}）に基づき共通シークレット（CS）を決定する

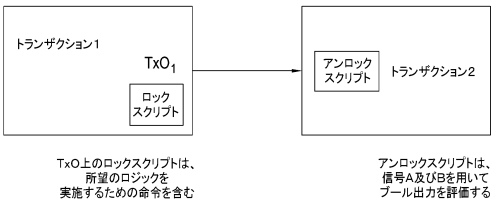
【図８】


3
第１ノード
5
ネットワーク
7
第２ノード
300
400
メッセージ（M）及び第１ノード第２秘密鍵（V_{2C}）に基づき第１署名メッセージ（SM1）を生成する
350
第１署名メッセージ（SM1）を第２ノードへ送信する
360
440
第１ノードから第１署名メッセージ（SM1）を受信する
450
決定した第１ノード第２公開鍵（P_{2C}）により、第１署名メッセージ（SM1）上の署名を検証する
460
第１署名メッセージ（SM1）を検証した結果に基づき、第１ノードを認証する
462
メッセージ（M）及び第２ノード第２秘密鍵（V_{2S}）に基づき第２署名メッセージ（SM2）を生成する
464
第２署名メッセージ（SM2）を第１ノードへ送信する
372
第２ノードから、第２署名メッセージ（SM2）を受信する
374
決定した第２ノード第２公開鍵（P_{2S}）により、第２署名メッセージ（SM2）上の署名を検証する
376
第２署名メッセージ（SM2）を検証した結果に基づき、第２ノードを認証する

【図９】



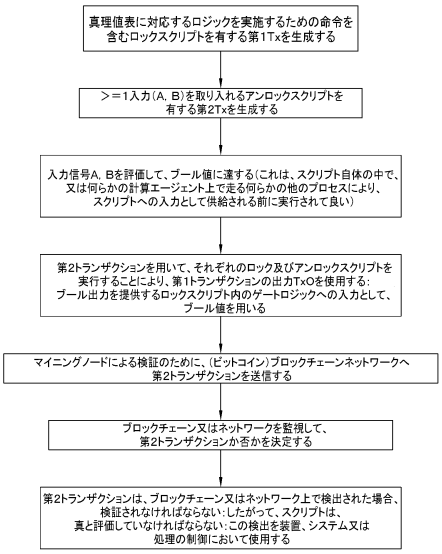
【図１０】



TxO上のロックスクリプトは、
所望のロジックを
実施するための命令を含む

アンロックスクリプトは、
信号A及びBを用いて
ブール出力を評価する

【図１１】



フロントページの続き

内
(72)発明者　サヴァナ，ステファヌ
イギリス国 シーエフ１０ ２エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス
７ス フロア アーカート－ダイクス アンド ロード エルエルピー 内

審査官　成瀬 博之

(56)参考文献　米国特許出願公開第２０１５／０３７９５１０（ＵＳ，Ａ１）
米国特許出願公開第２０１６／００９８７２３（ＵＳ，Ａ１）
米国特許出願公開第２０１６／０１６２８９７（ＵＳ，Ａ１）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｑ　１０／００－９９／００
Ｈ０４Ｌ　９／０８
Ｈ０４Ｌ　９／３２