(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7634046号
(P7634046)

(45)発行日　令和7年2月20日(2025.2.20)

(24)登録日　令和7年2月12日(2025.2.12)

(51)国際特許分類
H04L 9/32 (2006.01)

FI
H04L 9/32 200Z
H04L 9/32 200B

請求項の数　17　外国語出願　（全30頁）

(21)出願番号　特願2023-125939(P2023-125939)
(22)出願日　令和5年8月2日(2023.8.2)
(62)分割の表示　特願2020-555821(P2020-555821)の分割
原出願日　平成31年4月10日(2019.4.10)
(65)公開番号　特開2023-145662(P2023-145662A)
(43)公開日　令和5年10月11日(2023.10.11)
審査請求日　令和5年8月2日(2023.8.2)
(31)優先権主張番号　1806448.5
(32)優先日　平成30年4月20日(2018.4.20)
(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　ジョーゼフ，ダニエル
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・ネットワークに関与する周期的に順序付けられたノードの集合によって実施されるタスクを制御するためのコンピュータ実装方法およびシステム

(57)【特許請求の範囲】

【請求項１】

分散制御のためにコンピュータで実施される方法であって、

ブロックチェーン・ネットワークに参加している周期的に順序付けられたノードの集合における特定的なノードによって、前記特定的なノードの直後のノードから、前記直後のノードからスーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの可能な結果の組合せに対応する値を受信するステップであり、

前記値は、前記ノードと関連付けされており、それぞれが当該ノードと関連付けされた処理ステップの可能な結果に対応している公開鍵、および、前記スーパーバイザノードの公開鍵に基づくものである、

ステップと、

前記特定的なノードによって、前記受信した値、および、前記特定的なノードと関連付けされた処理ステップの可能な結果に対応している公開鍵に基づいて、状態値を生成するステップと、

前記特定的なノードによって、前記状態値を、前記スーパーバイザノード、および、前記特定的なノードの直前のノードと共有するステップと、

実行条件および戻り条件それぞれの満足に応答して、ノード間でリソースの制御を送信するステップと、

を含む、方法。

【請求項２】

前記リソースの制御は、戻り条件の満足に対して返される、
請求項１に記載の方法。
【請求項３】
前記戻り条件は、少なくとも２つのノードの署名を必要とする、
請求項２に記載の方法。
【請求項４】
前記ノードは、システムと関連付けされたモジュールに対応しており、かつ、
所与のノードと関連付けされた処理ステップの前記可能な結果は、前記システムの制御システムの関連するモジュールの出力に対応している、
請求項１に記載の方法。
【請求項５】
前記受信した値それぞれは、前記スーパーバイザノードと関連付けされた公開鍵を、前記直後のノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの可能な結果の組合せそれぞれに対応する前記公開鍵の値と、組合せることによって決定される、
請求項１乃至４いずれか一項に記載の方法。
【請求項６】
前記受信した値のうち所与の１つは、前記スーパーバイザノードと関連付けされた前記公開鍵と、可能な結果の組合せのうちそれぞれ１つの結果に対応している前記公開鍵のうち１つの値を、合計することによって決定される、
請求項５に記載の方法。
【請求項７】
前記状態値は、前記受信した値を、前記特定的なノードと関連付けされた処理ステップの可能な結果に対応する前記公開鍵と、組合せることによって形成される、
請求項１乃至６いずれか一項に記載の方法。
【請求項８】
前記状態値それぞれは、前記受信した値のうち１つと、前記特定的なノードと関連付けされた処理ステップの可能な結果に対応する前記公開鍵のうち１つを、合計することによって形成される、
請求項７に記載の方法。
【請求項９】
前記方法は、さらに、
前記状態値を使用して前記特定的なノードによって、前記受信した値に対応している複数のアンロッキング値のうち任意の１つの供給を含む実行条件の満足に応答して、前記特定的なノードと関連付けされた送信元アドレスから前記直後のノードの受信アドレスへ、リソースの制御を送信するように構成された、ブロックチェーントランザクションを準備するステップと、
実行すべき前記ブロックチェーントランザクションを待つステップ、および、前記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使用される第１のアンロッキング値を、前記ブロックチェーントランザクションから、獲得するステップであり、
前記第１のアンロッキング値は、直後のノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの出力を示しており、かつ、これらのノードと関連付けされた前記公開鍵に対応しており、かつ、前記出力に対応している秘密鍵に基づいている、
ステップと、
前記特定的なノードと関連付けされた前記処理ステップの可能な結果のうち１つを識別するステップと、
前記識別された結果および前記アンロッキング値に対応している前記公開鍵に対応する秘密鍵に基づいて、前記特定的なノードによって、第２のアンロッキング値を決定するステップであり、

前記第２のアンロッキング値は、前記状態値のうち１つに対応しており、かつ、前記特定的なノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの出力を示している、
ステップと、
前記第２のアンロッキング値を使用するステップであり、別のブロックチェーントランザクションを実行する、ステップと、
を含む、請求項１乃至８いずれか一項に記載の方法。
【請求項１０】
前記別のブロックチェーントランザクションは、前記直前のノードによって準備され、かつ、前記状態値のうち１つに対応している第２の複数のアンロッキング値のうち任意の１つの供給を含む第２の実行条件の満足に応答して、前記直前のノードと関連付けされた送信元アドレスから前記特定的なノードの受信アドレスに対して、第２のリソースの制御を送信するように構成されており、かつ、
前記別のブロックチェーントランザクションの準備の後で、前記特定的なノードと関連付けされた前記送信元アドレスから前記直後のノードの前記受信アドレスに対して、前記リソースの制御を送信するように構成された前記ブロックチェーントランザクションが準備される、
請求項９に記載の方法。
【請求項１１】
前記第２の複数のアンロッキング値のアンロッキング値それぞれは、前記状態値のそれぞれ１つに対応しており、かつ、前記状態値の基となる、前記公開鍵に対応している秘密鍵に基づくものである、
請求項１０に記載の方法。
【請求項１２】
前記アンロッキング値は、前記直後のノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの前記出力に対応している前記秘密鍵と、前記スーパーバイザノードと関連付けされた前記公開鍵に対応している秘密鍵との合計である、
請求項９乃至１１いずれか一項に記載の方法。
【請求項１３】
前記第２のアンロッキング値は、前記特定的なノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの前記出力に対応している前記秘密鍵と、前記スーパーバイザノードと関連付けされた前記公開鍵に対応している秘密鍵との合計である、
請求項９乃至１１いずれか一項に記載の方法。
【請求項１４】
前記リソースおよび前記第２のリソースは、同一である、
請求項１０または１１に記載の方法。
【請求項１５】
各公開鍵および対応する秘密鍵は、楕円曲線暗号化の公開鍵－秘密鍵ペアを形成する、
請求項１乃至１４いずれか一項に記載の方法。
【請求項１６】
請求項１乃至１５いずれか一項に記載の方法を実施するためのコンピュータ実施システム。
【請求項１７】
請求項１乃至１５いずれか一項に記載の方法を実行するようにコンピュータシステムを適合させるための命令を保管している、非一時的なコンピュータで読取り可能な記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】

本出願は、一般的に、分散化されたコンピューティングシステムに関する。より詳細には、分散コンピューティングシステムのノードに関連する処理ステップの制御および調整に関し、ブロックチェーン・ネットワークを使用して順序付けられた処理ステップを指示および制御することを含む。本発明は、また、分散コンピューティングシステムおよびネットワークの中のリソースの割り当て、および、そうしたリソースの制御および割り当てを実装し、かつ、実施するため、かつ、システムの中のエンティティ／ノードの階層的な組織化および配置のための、暗号技術およびメカニズムの使用、に関する。

【背景技術】

【０００２】

分散システムの利点は数多くあり、潜在的な攻撃に対する回復力（resilience）、リソースおよびエフォートの共有、等を含んでいる。分散コンピューティングシステムにおいて、分散システム内のノードとして動作する（acting）様々なコンピューティングデバイスは、これらのノード間で交換され得るネットワークメッセージを介して通信することができる。そうしたメッセージの交換により、例えば、ノードは、計算タスクを実行するように協調することができる。そうしたタスクは、様々なノードわたり分散された処理を含み得る。そうした分散処理は、様々なノードと関連付けされたステップの制御および調整を必要とし得る。例えば、特定の処理順序が実施され得る。

【０００３】

分散システムの一つの例は、ドライバー無し自動車（driverless car）のコンテキストにおいて発生し得る。例えば、ドライバー無し自動車は、各々がそれ自身のセンサデータを有する、様々なモジュールを含み得る。これらのモジュールそれぞれは、特定の処理ステップを実行することについて責任があり得る。追加的に、ドライバー無し自動車は、１つまたはそれ以上の内部ノード及び１つまたはそれ以上の外部ノードと相談して、例えば、車両の制御に関連し得る、といった、所定の処理を実行する、とういことであってよい。外部ノードは、例えば、ウェブまたはクラウドベースであり得る。

【０００４】

そうした分散システムでは、タスク処理（processing task）においてステップの順序付けを実施することができることが必要であり、または、望ましくあり得る。例えば、ノードは、処理ステップ（processing steps）を順番に実行する必要があり得る。処理ステップは、次いで、組合され、タスクのために複数の可能な結果のうち１つをもたらすことができる。分散システムに関してしばしば出くわす他の技術的課題は、ネットワークまたはシステムの中のリソースの割り当て、および、アクタ（actors）（すなわち、ノード)及びそれらの間のコミュニケーションをどのように効率的で堅牢な（robust）方法に組織化するか、から生じる困難を含んでいる。ノードの組織化および階層的な考察も、また、分散システムの中の課題を提起する。数学的および暗号技術を介して、実施され、かつ、決定される、分散システムの中に階層的な配置および制御フローを有することは、有利であろう。これは、セキュリティおよびプロセスの自動化に関連する利点を提供するだろう。

【０００５】

オンライン投票（voting）または意思決定プロトコルの提供での試みは、一般的に、悪用（abuse）を防止するように試みる方法で構築されるが、投票またはフィードバックを確証し、かつ、考慮に入れるために中央当局（central authority）に依拠する方法である。中央当局は、投票に秘密性と適格性の検証を加えるために、ブラインド署名（blind signatures）および同形特性（homomorphic property）といった、異なる暗号プリミティブを使用してよく、投票選挙（ballot vote）に対して秘密（secrecy）と適格性の検証を追加する。－例えば、R.RiemannとS.Grumbachの共著で、arXivの前刷り（preprint）であるarXiv:1705.04480を通じて入手可能な“Distributed Protocols at the Rescue for Trustworthy Online Voting”を参照のこと。例えば、集中投票プロトコルは、D.L.Chaum著で、Communications of ACM、24(2)、84-90の“Untraceable electronic mail, return addresses、and digital pseudonyms”(1981)に記

載されている。Chaumのプロトコルにおいて参加者は、投票者同士（vote-voters）のリンクを偽装するために、Mix Nets set-upを使用して、ブロードキャストの前に投票を再編成（reshuffle）し、かつ、暗号化する中央混合当局に対して投票を送信する。

【０００６】

他の以前の研究は、信頼できる中央当局への依存を取り除くが、参加者間の協力に依存している。例えば、Foundations of Computer Science、1982、SFCS ’08、23rd Annual Symposium （pp.160-164）における、A.C.Yao著の“Protocols for secure computations”(1982年11月)は、Secure Multi-Party Computations（MPC）プロトコルを説明しており、その目的は、信頼できる第三者を必要としないで、ユーザが、秘密を保持することを望む、彼らのジョイント私的入力の関数を計算できるようにすることである。MPCにより、当事者は、私的かつ安全に彼らのアベレージ投票を共同で計算することができるが、それに応じて、当事者間の協力に依存している。

【０００７】

より基本的には、前述の従来の研究それぞれは、独立した選択、および、非独立ではない選択のために意図されたものであり、言うまでもなく、結果に到達するために定められた順序で処理ステップを実行する。

【０００８】

上記に照らして、１つの技術的問題は、どのように分散システムにおけるノードによる処理ステップの順序付けを可能にし、かつ、実施するソリューションを提供するかであり得る。別の技術的問題は、分散システムにおいて、そうした順序付けられた処理から生じる結果の不正防止（tamperproof）または不正防止記録（tamper-resistant record）をどのように提供するかであり得る。別の技術的な問題は、分散システムの中で、制御、及び／又は、階層、および許可(例えば、アクセス許可)を実施するために暗号技術をどのように使用するかであり得る。

【０００９】

前述の技術的問題の１つまたはそれ以上に対してソリューションを提供することは、ブロックチェーンの使用を含み得る。この文書において、用語「ブロックチェーン（“blockchain”）」は、電子的、コンピュータベース、分散型の全ての形態の台帳（ledger）を含むように使用されている。これらは、合意に基づく（consensus-base）ブロックチェーンおよびトランザクションチェーン技術、許可及び非許可（un-permissioned）台帳、共有台帳、およびそれらの変形を含んでいる。ブロックチェーン技術の最も広く知られたアプリケーションはビットコイン台帳であるが、他のブロックチェーン実装が提案され、かつ、開発されている。ビットコイン（Bitcoin）は、便宜上および例示のためだけにここにおいて参照され得るが、本出願の技術的事項（subject matter）は、ビットコインブロックチェーンを伴う使用に限定されるものではなく、そして、代替的なブロックチェーンの実装およびプロトコルは本出願の範囲内にあることが留意されるべきである。

【００１０】

ブロックチェーンは、ピア・ツー・ピア（peer-to-peer）の電子台帳であり、これは、トランザクションで順番に構成される、ブロックから構成されるコンピュータベースの分散システムとして実装される。各トランザクションは、ブロックチェーンシステム内のアドレス間におけるデジタルアセットの制御の移転を符号化するデータ構造であり、そして、少なくとも１つの入力および少なくとも１つの出力を含んでいる。各ブロックは、そのブロックが一緒にチェーン化されるための以前のブロックのハッシュを含んでおり、その発端からブロックチェーンに対して書き込まれてきた全てのトランザクションについて永続的で、変更不可能なレコードを生成する。

【００１１】

ブロックチェーンは、多種多様なアプリケーションにおいて使用することができる。例えば、ブロックチェーンは、１つまたはそれ以上の商品（commodity）の所有権を反映する台帳を提供することに使用され得る。例えば、ビットコインブロックチェーンにおいて、台帳は、ビットコインおよびその一部（fraction）の所有権を反映している。そうした

商品のいくつかは、例えば、計算リソースのユニットといった、根底にある（underlying）ユニットを表すことができる。

【発明の概要】

【００１２】

本出願に従って、添付の請求項において定義されるような方法およびシステムが提供される。

【００１３】

本発明は、分散制御のためにコンピュータで実施される方法を提供することができる。本方法は、ブロックチェーン・ネットワークに参加している周期的に順序付けられたノードの集合における特定的なノードによって、前記特定的なノードの直後のノードから、前記直後のノードからスーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの可能な結果の組合せに対応する値を受信するステップ、を含み得る。前記値は、前記ノードと関連付けされており、それぞれが当該ノードと関連付けされた処理ステップの可能な結果に対応している公開鍵、および、前記スーパーバイザノードの公開鍵に基づくものであり得る。本方法は、前記特定的なノードによって、前記受信した値、および、前記特定的なノードと関連付けされた処理ステップの可能な結果に対応している公開鍵に基づいて、状態値を生成するステップ、を含み得る。本方法は、前記特定的なノードによって、前記状態値を、前記スーパーバイザノード、および、前記特定的なノードの直前のノードと共有するステップ、を含み得る。本方法は、前記状態値を使用して前記特定的なノードによって、前記受信した値に対応している複数のアンロッキング値のうち任意の１つの供給を含む実行条件の満足に応答して、前記特定的なノードと関連付けされた送信元アドレスから前記直後のノードの受信アドレスへ、リソースの制御を送信するように構成されたブロックチェーントランザクションを準備するステップ、を含み得る。

【００１４】

いくつかの実装において、ノードは、システムと関連付けされたモジュールに対応しており、かつ、所与のノードと関連付けされた処理ステップの前記可能な結果は、前記システムの制御システムの関連するモジュールの出力に対応し得る。

【００１５】

いくつかの実装において、前記受信した値それぞれは、前記スーパーバイザノードと関連付けされた公開鍵を、前記直後のノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの可能な結果の組合せそれぞれに対応する前記公開鍵の値と、組合せることによって決定され得る。

【００１６】

いくつかの実装において、前記受信した値のうち所与の１つは、前記スーパーバイザノードと関連付けされた前記公開鍵と、可能な結果の組合せのうちそれぞれ１つの結果に対応している前記公開鍵のうち１つの値を、合計することによって決定され得る。

【００１７】

いくつかの実装において、前記状態値は、前記受信した値を、前記特定的なノードと関連付けされた処理ステップの可能な結果に対応する前記公開鍵と、組合せることによって形成され得る。

【００１８】

いくつかの実装において、前記状態値は、前記受信した値のうち１つと、前記特定的なノードと関連付けされた処理ステップの可能な結果に対応する公前記開鍵のうち１つを、合計することによって形成され得る。

【００１９】

いくつかの実装においては、リターン条件（return condition）が満足されると、第２のリソースの制御を特定的なノードに返すように、別のブロックチェーントランザクションが構成され得る。いくつかの実装において、本方法は、さらに、実行すべき前記ブロックチェーントランザクションを待つステップ、および、前記トランザクションをアンロックするために使用される第１のアンロッキング値を、前記トランザクションから、獲得

するステップであり、前記第１のアンロッキング値は、直後のノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの出力を示しており、かつ、これらのノードと関連付けされた前記公開鍵に対応しており、かつ、前記出力に対応している秘密鍵に基づいている、ステップを含み得る。
そうした方法は、前記特定的なノードと関連付けされた前記処理ステップの可能な結果のうち１つを識別するステップ、を含み得る。そうした方法は、前記識別された結果および前記アンロッキング値に対応している前記公開鍵に対応する秘密鍵に基づいて、前記特定的なノードによって、第２のアンロッキング値を決定するステップ、を含み得る。前記第２のアンロッキング値は、前記状態値のうち１つに対応しており、かつ、前記特定的なノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの出力を示し得る。そうした方法は、前記第２のアンロッキング値を使用するステップであり、別のブロックチェーントランザクションを実行する、ステップ、を含み得る。

【００２０】

いくつかの実装において、前記別のブロックチェーントランザクションは、前記直前のノードによって準備され、かつ、前記状態値のうち１つに対応している第２の複数のアンロッキング値のうち任意の１つの供給を含む第２の実行条件の満足に応答して、前記直前のノードと関連付けされた送信元アドレスから前記特定的なノードの受信アドレスに対して、第２のリソースの制御を送信するように構成され得る。前記ブロックチェーントランザクションは、前記別のブロックチェーントランザクションの準備の後で、前記特定的なノードと関連付けされた前記送信元アドレスから前記直後のノードの前記受信アドレスに対して、前記リソースの制御を送信するように構成されている。

【００２１】

いくつかの実装において、前記第２の複数のアンロッキング値のアンロッキング値それぞれは、前記状態値のそれぞれ１つに対応しており、かつ、前記状態値の基となる、前記公開鍵に対応している秘密鍵に基づくものであり得る。

【００２２】

いくつかの実装において、前記アンロッキング値は、前記直後のノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの前記出力に対応している前記秘密鍵と、前記スーパーバイザノードと関連付けされた前記公開鍵に対応している秘密鍵との合計であり得る。

【００２３】

いくつかの実装において、前記第２のアンロッキング値は、前記特定的なノードから前記スーパーバイザノードの直前のノードに至るまでのノードと関連付けされた処理ステップの前記出力に対応している前記秘密鍵と、前記スーパーバイザノードと関連付けされた前記公開鍵に対応している秘密鍵との合計であり得る。

【００２４】

いくつかの実装において、前記リソースおよび前記第２のリソースは、同一であってよい。

【００２５】

いくつかの実装において、各公開鍵および対応する秘密鍵は、楕円曲線暗号化の公開鍵－秘密鍵ペアを形成し得る。

【００２６】

本発明は、また、上記に従った方法を実行するためのコンピュータ実施システムも提供し得る。

【００２７】

本発明は、また、上記に従った方法を実行するようにコンピュータシステムを適合させるための命令を保管している、非一時的なコンピュータで読取り可能な記憶媒体も提供し得る。

【図面の簡単な説明】

【００２８】
本発明の１つの態様または実施形態に関連して記載される任意の特徴は、また、１つまたはそれ以上の他の態様／実施形態に関しても使用され得る。本発明のこれら及び他の態様は、ここにおいて説明される実施形態から明らかであり、かつ、それを参照して解明されるだろう。本発明の実施形態が、これから、単なる例示として、そして、添付の図面を参照して説明される。
【図１】図1は、支払チャネル（payment channel）において使用されるトランザクションを示している。
【図２】図2は、図1に従った支払チャネルがどのようにして作り出され得るかを説明しているフローチャートである。
【図３】図3は、本出願の動作環境の一つの例を示している簡略化された概略図である。
【図４】図4は、本出願の動作環境の別の例を示している簡略化された概略図である。
【図５】図5は、一つの例示的なコンピューティングデバイスを示している。
【図６】図6は、ノードから構成される周期的に順序付けられた（cyclically-ordered）集合を示している図である。
【図７】図7は、一つの例示的なタスク処理対応しているディシジョンツリー（decision tree）を示している。
【図８】図8は、図7のディシジョンツリーの特定のエッジと関連付けされた秘密鍵（private key）を示している。
【図９】図9は、図8の秘密鍵に関連し、かつ、図7のディシジョンツリーの特定のエッジに関連する公開鍵（public key）の値を示している。
【図１０】図10は、図6の特定的なノードによって実行されるオペレーションを説明しているフローチャートである。
【図１１】図11は、本出願に従った、一つの例示的な支払チャネルを示している図である。
【図１２】図12は、図1に従った支払チャネルがどのようにして生成され得るかを説明しているフローチャートである。
【図１３】図13は、図6のノードによって実行されるオペレーションを説明しているフローチャートである。
【図１４】図14は、図6の特定的なノードによって実行されるオペレーションを説明しているフローチャートである。
【００２９】
図面においては、同様の参照番号が、同様のエレメント（element）および特徴（feature）を示すために使用されている。
【発明を実施するための形態】
【００３０】
本出願において、用語「及び／又は（“and/or”）」は、列挙されたエレメントの全ての可能な組合せ及び部分的組合せ（sU_B-combination）をカバーするように意図されており、列挙されたエレメント単独の任意の１つ、任意の部分的組合せ、または、全てのエレメントを含み、そして、必ずしも追加のエレメントを排除するものではない。
【００３１】
本出願において、フレーズ「少なくとも１つの・・・または・・・（“at least one of…or…”）」は、列挙されたエレメントのうち任意の１つまたはそれ以上をカバーするように意図されており、列挙されたエレメント単独の任意の１つ、任意の部分的組合せ、または、全てのエレメントを含み、そして、必ずしも追加のエレメントを排除せず、かつ、必ずしも全てのエレメントを必要とするものではない。
【００３２】
本発明を具現化するプロトコルは、順序付けられた方法で各々が決定を行うために協働するノードのグループについて、ここにおいて説明されている。これは、計算リソースのユニット（unit）をコミットするようにノードに対して要求することによる概念に基づい

て構築されており、ノードは、タスク処理における以前のステップの出力から導出された、蓄積されたシークレット値の利用によってのみ回復することができる。

【００３３】

ここにおいて説明されるプロトコルは、現存するビットコイン関連技術である、支払チャネル（Payment Channels）(例えば、“Introduction to Micropayment Channels”http://super3.org/introduction-to-micropayment-channels/を参照のこと)を基盤としている。これは、参加者のペア間でのオフブロック（off-block）のビットコイントランザクション（Bitcoin transaction）のために設計された技術であり、そして、特に、払戻し（refund）トランザクションの利用を組み込んでいる。

【００３４】

支払チャネル（Payment channels）

上述のように、支払チャネルは、以下の説明において参照され、そして、読者の便宜のために、支払チャネルの概要が続いている。

【００３５】

支払チャネルは、当事者（parties）のために、ブロックチェーンに対して全てのトランザクションをコミットすることなく、複数の暗号通貨トランザクションを行うようにデザインされた技術である。典型的な支払チャネルの実装においては、ほぼ無制限な金額（amount）の支払いを行うことができるが、ブロックチェーンに対して２つのトランザクションを追加することだけが必要である。

【００３６】

ブロックチェーンに対して追加されるトランザクションの数の減少、および、それに伴うコストの削減に加えて、支払チャネルは、また、速さの利点、および、重要なことには、事態が計画どおりに進まない場合、または、いずれかの参加者が支払いの所定の集合（set）を超えて進まないように決定した場合に、ユニットを返済させる当事者の能力も提供する。支払チャネル実装について以下に概説される。

【００３７】

アリス（Alice）がブロックチェーンリソースをボブ（Bob）に移転（transfer）する必要があるシナリオを考えてみる。これには、状況に応じて、一定期間にわたりアリスからボブへの複数の支払いが必要となり得る。アリスは、交換（exchange）に係る可能な集合において、ボブに対して、例えば、(合計で)15BTCを移転することを期待している。これを促進するために、アリスとボブとの間には支払チャネルが確立されており、そして、以下のように動作する。

【００３８】

最初に、アリスは、２オブ２（2-of-2）マルチシグネチャ（multi-signature）ペイ・ツー・スクリプト・ハッシュ（pay-to-script-hash、P2SH）トランザクション、T_c、を生成する。アリスとボブの両方によって管理され、アリスから発信された15BTCをコミットするものである。この時点で、トランザクションは、ビットコインネットワークに対してサブミットされない(そうしたマルチシグネチャアドレスは、このアドレスからお金を使うあらゆるトランザクションに２人の個人(アリスとボブ)が暗号的に署名することを必要とする)。

【００３９】

次に、アリスは、マルチシグネチャ管理ファンドの全てのユニットをアリスに返却する、別個の払戻しトランザクション、$T_{c,0}$、を生成する。このトランザクションは、100ブロックのｎＬｏｃｋＴｉｍｅ値を含んでいる(ｎＬｏｃｋＴｉｍｅは、ビットコイントランザクションのパラメータであり、これにより、ビットコイントランザクションは、指定された時間が経過した後でだけ実行可能になり得る)。ボブは、そのトランザクションに署名する。この払戻しトランザクションにより、アリスとボブとの間の交換がうまくいかない場合に、アリスは、ｎＬｏｃｋＴｉｍｅが経過（transpired）した後で、払戻しが成され得る。

【００４０】

次に、アリスは、元のトランザクションT_cに署名する。

【００４１】

この時点で、アリスとボブは、アリスからボブに対して行われる(ブロックチェーン外での)移転を反映するために、新たな払戻しトランザクションを生成するように進行し得る。これらの払戻しトランザクションは、その時点でアリスがボブに対して移転することが必要とされるリソースの正味数量を反映している。例として、アリスがボブへ5BTCを移転する場合には、ボブに対して5BTCを送り、かつ、アリスに対して10BTCを戻す出力を有する、新たな払戻しトランザクション$T_{r,i}$、が生成される。アリスが別の5BTCをボブに対して移転する必要がある場合、ボブに対して5BTC、および、アリスに対して5BTCを送る出力を伴う、新たな払戻しトランザクション$T_{r,i+1}$、が生成される。それぞれの新たな払戻しトランザクションについては、詳細について合意があると仮定して、両方の当事者がトランザクションに署名するが、必ずしもネットワークに対してトランザクションをサブミットするものではない。

【００４２】

生成された連続する払戻しトランザクションそれぞれは、前回の（previous）払戻しトランザクションよりも小さいｎＬｏｃｋＴｉｍｅを有することに注意する。－すなわち、ｎＬｏｃｋＴｉｍｅ（$T_{r,i+1}$）＜ｎＬｏｃｋＴｉｍｅ（$T_{r,i}$）である。

【００４３】

参加者が任意の$T_{r,i}$に署名することを拒否した場合には、被害の参加者（aggrieved participant）は、単に$T_{r,i-1}$を提出することができる。最悪のシナリオにおいて、アリスは、$T_{r,0}$に署名し、そして、ネットワークに対してサブミットして(ｎＬｏｃｋＴｉｍｅが満了した後で)彼女の全てのユニットの返還を要求する（reclaiming）。

【００４４】

構築された最終的な払戻しトランザクションは、アリスからボブに対して移転されるユニットの正味の合計を表している。このトランザクションは、ネットワークに対してサブミットされる。

【００４５】

図1は、支払チャネルにおいて使用されるトランザクションT_c 100Aおよび$T_{r,n}$ 100Dを示している。Ｍは、アリスからボブに対して送られ得る最大金額を表している。x_iは、アリスがボブに対して支払う必要のあるユニットの現在の正味金額である。S_{stop}は、最初の払戻しトランザクションにおけるｎＬｏｃｋＴｉｍｅである。ｎは、アリスとボブとの間で行われる進行中の(ブロック外の)移転で生成された払戻しトランザクションの数である(これは、最初の払戻しトランザクションを除いている)。ｓは、当事者が前回の払戻しトランザクションをサブミットしている他の当事者のリスクを負う前に、両方の当事者(アリスとボブ)が払戻しトランザクションについて同意するために割り当てられた時間であり、アリスとボブとの間の交換を効果的に終了させる。

【００４６】

$t+n*s<S_{stop}$ であることに留意する。ここで、ｔは、現在時刻であり、かつ、$(S_{stop}-n*s)\geqq s$である。

【００４７】

図1のトランザクションT_c 100A、$T_{r,0}$ 100B、$T_{r,1}$ 100C、および$T_{r,n}$ 100Dは、ブロックチェーン上に出現し得るトランザクションである。

【００４８】

アリスとボブとの間の支払チャネルを構築するためのオペレーションが、図2のフローチャート200に示されている。オペレーション210及びそれ以降は、コンピュータで読取り可能な記憶媒体において保管され得るようなコンピュータで実行可能な命令を含む、ソフトウェアを実行する１つまたはそれ以上のコンピューティングデバイスに係る１つまたはそれ以上のプロセッサによって実行される。

【００４９】

オペレーション210において、アリスと関連付けされたコンピューティングデバイスの

プロセッサは、上述の方法で、２オブ２・マルチシグネチャ・ペイ・ツー・スクリプト・ハッシュ（P2SH）トランザクション、T_cを生成する。

【００５０】

オペレーション210から、制御フローはオペレーション212へ進む。オペレーション212において、アリスと関連付けされたコンピューティングデバイスのプロセッサは、別個の払戻しトランザクション$T_{r,0}$を生成し、マルチシグネチャ制御ファンドの全てのユニットを、上記の方法で、アリスと関連付けされた口座（account）に対して戻す。

【００５１】

オペレーション212から、制御フローはオペレーション214へ進む。オペレーション214において、アリスと関連付けされたコンピューティングデバイスのプロセッサは、上述の払戻しトランザクションに署名する。

【００５２】

オペレーション214から、制御フローはオペレーション216へ進む。オペレーション214においては、ボブと関連付けされたコンピューティングデバイスのプロセッサも、また、上述の払戻しトランザクションに署名することができる。トランザクションがそのように署名された場合に、制御フローはオペレーション218へ進む。代替的に、トランザクションが署名されない場合には、支払チャネルの生成が中止される。

【００５３】

オペレーション218において、アリスと関連付けされたコンピューティングデバイスのプロセッサはT_cに署名し、そして、それをブロックチェーンに対してサブミットする。制御フローは、次いで、オペレーション220へ進む。

【００５４】

オペレーション220において、上記の払戻しトランザクションは、最初の払戻しトランザクションとして認識され、そうして、時間が許されるならば、アリスからボブに対するさらなる移転が交渉され得る。

【００５５】

オペレーション220から、制御フローはオペレーション222へ進む。オペレーション222では、さらなる移転を交渉するために十分な時間が残っているか判断される。十分な時間が残っていない場合、制御フローは、オペレーション224へ進み、そこでは、最後の払戻しトランザクションがブロックチェーンに対してサブミットされる。代替的に、十分な時間が残っている場合に、制御フローはオペレーション226へ進む。

【００５６】

オペレーション226では、アリスとボブとの間のさらなる移転が交渉される。オペレーション226から、制御フローはオペレーション228に進み、そこでは、交渉が成功したかどうかが判断される。その交渉が失敗した場合に、制御は、上述のオペレーション224へ進む。代替的に、交渉が成功すると、結果として、制御フローはオペレーション230へ進む。

【００５７】

オペレーション230では、成功した交渉に由来する合意を反映した新たな払戻しトランザクションが、上述の方法で生成される。次に、制御フローは、オペレーション240へ進み、そこでは、新たな払戻しトランザクションが現在の払戻しトランザクションとして認識される。

【００５８】

オペレーション240から、制御フローはオペレーション242へ進み、そこでは、アリスおよびボブと関連付けされたコンピューティングデバイスのプロセッサが現在の払戻しトランザクションに署名することができる。そうであれば、制御フローは、上述のオペレーション222へ戻る。代替的に、トランザクションがそのように署名されない場合、現在の払戻しトランザクションの認識は、オペレーション244において以前の払戻しトランザクションへ立ち戻る。オペレーション244から、制御フローは、上述のオペレーション222へ戻る。

【００５９】
図3は、本出願に係る一つの例示的な動作環境を示している。
【００６０】
図示されるように、複数のノード300が、コンピュータネットワーク310を介して通信している。ノード300それぞれは、コンピューティングデバイスであり、かつ、ブロックチェーンに参加しており、そして、例えば、コンピューティングリソースのユニットといった、ユニットの量をブロックチェーンが反映するようと関連付けされた１つまたはそれ以上の関連アドレスを有する。
【００６１】
ノード300に係る様々なものは、分散処理において処理ステップを実行することができ、その出力は組合されて結果（outcome）を形成する。
【００６２】
以下でさらに説明されるように、本出願の技術的事項（subject matter）は、様々な状況において適用され得る。例えば、図4は、システム、および、より具体的には、自動運転車両（self-driving vehicle）400のコンテキストにおける、本出願の特定の例示的な動作環境を示している。
【００６３】
自動運転車両400による分散処理は、複数のモジュールに依存し得る。例えば、分散処理は、マップモジュール410、ホスピタリティモジュール420、気象（weather）モジュール430、および交通（traffic）モジュール440といった、様々なノードを含むことができるだろう。図示されるように、モジュール410-440は、ネットワーク450を介して通信している。
【００６４】
ノードのうちいくつかは、自動運転車両400の中へ統合されてよく、そして、ノードのうち他のものは、ウェブまたはクラウドベースであってよい。例えば、図示されるように、マップモジュール410およびホスピタリティモジュール420は、自動運転車両400の中へ統合されており、一方で、気象モジュール430および交通モジュール440は、自動運転車両400から離れている。
【００６５】
マップモジュール410は、最短のルートを見つけるためのモジュールであり得る。例えば、マップモジュール410は、距離に基づいて最良のルートを選択することができる。
【００６６】
ホスピタリティモジュール420は、ホスピタリティの利用可能性（availability）に基づいてルートを選択することができる。例えば、ホスピタリティモジュール420は、レストランおよびホテルの利用可能性に基づいて、最良のルートを選択することができる。
【００６７】
気象モジュール430は、気象に基づいてルートを選択することができる。例えば、気象モジュール430は、過酷または厳しい天候を回避することに基づいて、最良のルートを選択することができる。
【００６８】
交通モジュール440は、交通状況に基づいてルートを選択することができる。例えば、交通モジュール440は、交通渋滞（traffic congestion）を回避することに基づいて、最良のルートを選択することができる。
【００６９】
都合のいいことに、本出願の技術的事項は、モジュール410-440が、自動運転車両400のルーティングに関するタスクに関連する処理ステップを集合的に実行できるように、適用されてよく、これらのステップの出力は、特定の結果、すなわち、選択されたルートを生じている。特に、以下でさらに説明されるように、ブロックチェーンは、様々な処理ステップの出力を記録し、かつ、それらをノード間で伝送するように使用されるため、モジュール410-440の出力は、また、例えば、自動運転車両400を巻き込んでいるインシデン

トの後に続く故障判定（fault determination）のため、といった、将来の監査およびレビューを可能にするためにも記録され得る。

【００７０】

上述のように、本出願の技術的事項は、また、他のコンテキストにも適用され得る。例えば、本出願の技術的事項がシステムの制御に適用されている場合に、制御される特定のタイプのシステムが、自動運転車両である必要はない。例えば、本出願の技術的事項は、例えば、工業プラントの運転または制御に責任を負い得るといったモジュールが、そうしたプラントに関連する処理ステップを集合的に実施することを可能にするように、適用され得る。特定的な例において、そうしたステップは、工業プロセスの決定または制御ポイントに対応し得る。別の例において、本出願の技術的事項は、そうしたシステムのモジュールが、交通制御システムのオペレーションに適用され、例えば、交通信号の制御といった、タスクに関連する処理ステップを集合的に実行することを可能にする。

【００７１】

図3に戻って、ノード300それぞれは、コンピューティングデバイスであることを思い出してみよ。図5は、コンピューティングデバイスのハイレベルな動作図である。例示的なコンピューティングデバイス500は、例えば、ノード300の１つまたはそれ以上を含む、ここにおいて説明されるコンピュータシステムの１つまたはそれ以上に係る典型例であってよい。例示的なコンピューティングデバイス500は、特定の機能を実行するように適合させるソフトウェアを含んでいる。

【００７２】

例示的なコンピューティングデバイス500は、様々なモジュールを含んでいる。例えば、図示されるように、例示的なコンピューティングデバイス500は、プロセッサ510、メモリ520、およびネットワークインターフェイス530を含み得る。図示されるように、例示的なコンピューティングデバイス500の前述のコンポーネントは、バス540上で通信している。

【００７３】

プロセッサ510は、ハードウェアプロセッサである。プロセッサ510は、例えば、１つまたはそれ以上のARM、Intel x86、PowerPCプロセッサ、等であり得る。

【００７４】

メモリ520は、データを保管し、かつ、取り出すことを可能にする。メモリ520は、例えば、ランダム・アクセス・メモリ、リード・オンリー・メモリ、および永続性ストレージを含み得る。永続性ストレージは、例えば、フラッシュメモリ、ソリッドステートドライブ、等であり得る。リード・オンリー・メモリおよび永続性ストレージは、非一時的なコンピュータで読取り可能な記憶媒体である。コンピュータで読取り可能な媒体は、例示的なコンピューティングデバイス500の全体的なオペレーションを支配するオペレーティングシステムによって管理され得るようなファイルシステムを使用して編成され得る。

【００７５】

ネットワークインターフェイス530により、例示的なコンピューティングデバイス500は、例えば、コンピュータネットワーク310(図3)といった他のコンピューティングデバイス及び／又は様々な通信ネットワークと通信することができる。

【００７６】

命令を含むソフトウェアは、コンピュータで読取り可能な媒体からプロセッサ510によって実行される。例えば、ソフトウェアは、メモリ520の永続性ストレージからランダム・アクセス・メモリの中へロードされ得る。追加的または代替的に、命令は、メモリ520のリード・オンリー・メモリから直接的にプロセッサ510によって実行され得る。

【００７７】

以下でさらに説明されるように、ソフトウェアは、例えば、ノード300のうち１つまたはそれ以上、及び／又は、モジュール410-440のうち１つまたはそれ以上を含む、ここにおいて言及される様々なコンピュータシステムのうち１つまたはそれ以上として動作するように、例示的なコンピューティングデバイス500のインスタンスを適合させることがで

きる。

【００７８】

以下でさらに説明されるように、ここにおいて説明されるプロトコルにおいて、ノードは、全体的な分散コンピューティングタスクの処理ステップを実行する責任を負う。図6を参照すると、特定的な実施例において、ノード600は、順序付けられた方法でタスク処理を実行する責任を負うことができる。ノードの１つは、スーパーバイザノードとして動作し、－例えば、ノード600のうち１つ、ノードU_0は、スーパーバイザノードであり得る－そして、他のノードは、オプションを選択する責任を負う。タスクにおけるステップ処理の順序は、ノードで形成される周期的に順序付けられた集合（cyclically-ordered set）を実現するために使用され得る。周期的に順序付けられた集合は、リングとして考えられてよく、ノードそれぞれは、そのノードに対する直後のノードおよび直前のノードとして２つの他のノードを有している。例えば、ノード600の周期的に順序付けられた集合は、図6に示されるように、方向付けされた（directed）リング610として描写され得る。

【００７９】

方向付けされたリング610においてノード600の非スーパーバイザノードそれぞれには、参照の便宜のためにラベルU_xが割り当てられ、xは、タスク処理におけるステップの意図された順序に従った、アルファベット順の文字に対応している。

【００８０】

スーパーバイザノードU_0は、プロトコルの実行を監督する責任を与えられたノードであることが期待される。スーパーバイザノードは、以下でさらに説明されるように、プロトコルのイニシエータ（initiator）およびコーディネータ（coordinator）として動作する。一つの例においては、スーパーバイザノードが、意思決定プロセスの結果に基づいて割り当てられるリソースを提供する当事者（party）と関連付けられている、ということであり得る。

【００８１】

ノード600、スーパーバイザノード及び／又は非スーパーバイザノード、は、ノード300及び／又はモジュール410-440のものであってよい。さらに、スーパーバイザノードU_0は、ノード300の中から選択されてよい。そうした選択は、様々な方法で行うことができる。例えば、ノード300それぞれがオペレーションを実行し、または、プロトコルに参加して、スーパーバイザノードとして選択される、ノードのうち特定のものを結果として生じる、とうことであり得る。代替的に、ノード300のうち１つが、ある中央当局（central authority）によってスーパーバイザとして選択され得る。

【００８２】

ここにおいて説明されるプロトコルは、ディシジョンツリーの構造に対応しているタスクにおけるステップ処理を用いてノード600に参加することによって、分散コンピューティングタスクを実行することを促進する。そうしたディシジョンツリーの一つの例は、ディシジョンツリー700を示す図7に見出だされる。ディシジョンツリー700において、ツリーの各レベルは、ここにおいて説明されるプロトコルに参加しているノード、そして、特には、それらのノード300のうち１つによる、処理の結果を表している。言い換えると、ディシジョンツリー700の各レベルは、ノード600によって形成される分散システムによる処理の最終的な結果における「ファクタ（“factor”）」に対応する。ディシジョンツリー700の所与のレベルにおけるノード600それぞれは、ツリーのより高いレベル(すなわち、ルートに対してより近いレベル)によって表されるステップ処理に係る特定の出力の組合せに対応する。ノードから発生するエッジそれぞれは、そのノードによって表される決定に関して当事者がなし得る、そのノードと関連付けされ／ノードよって実行される処理の可能な結果(「選択（“choice”）」)を表している。

【００８３】

ディシジョンツリー700では、処理に参加しているｎ＝３ステップ(および3ノード)が存在しており、処理に関連する各ノードは、それぞれｍ＝２のオプションのうち１つを結果として生じている。

【００８４】
一般的に、ディシジョンツリー700に係る$U_{A,i}$の各ノードは、ノード、処理ステップ、および、前の出力（previous outputs）(すなわち、ツリーのより高いレベルに対応している処理に係る出力の特定の組合せ)を表しており、ここで、aは、特定的なノードによって実行される処理ステップを表し、そして、iは、ツリーにおける前の処理ステップの出力を表している。特定的な例において、$U_{B,2}$は、当事者U_Aが処理ステップAについて特定的な出力を識別した後で処理ステップBを実行するノードを表している。
【００８５】
ディシジョンツリー700のリーフノード（leaf nodes）は、様々な処理ステップからの出力の特定の組合せから結果として生じる、可能な最終結果の集合を表している。言い換えると、ノードそれぞれに関連するタスクの出力は、分散処理の結果に向けたディシジョンツリーのパス（path）を形成する。注目すべきことに、多数決が勝者を決定する意思決定または投票プロトコルとは異なり、本プロトコルにおいて、どの結果が選択されるかを決定するのは、様々な処理ステップの出力の組合せである。より詳細には、特定的な処理ステップの出力は、特定的な結果に対する投票ではなく、むしろ、可能な結果の集合(すなわち、選択された出力に対応するブランチに沿ったディシジョンツリー700の対応するノードの子（child）である全ての結果)に対する分散処理の可能な結果を狭めている。かくして、事前に決定された（pre-determined）一連のステップにおける各処理ステップが減少するにつれて、可能な結果－分散処理の出力－の集合は、唯一の可能な結果になるまで残存する。
【００８６】
特定的な例では、ディシジョンツリー700において、可能な結果O_{212}が、より濃い矢印を用いて示される特定のパスを介して到達され得る。最初に、示されるように、U_Aがオプション2を選択した場合、このことは可能な結果の集合を｛O_{211}、O_{212}、O_{221}、O_{222}｝に制限する。次に、U_Bがオプション1を選択した場合、このことは可能な結果の集合｛O_{211}、O_{212}｝に制限する。次いで、U_Cがオプション2を選択した場合、このことは可能な結果の集合を｛O_{212}｝に制限する。
【００８７】
ノードによる処理には時系列的側面（chronological aspect）があり、ここにおいて説明されるプロトコルによって実行されているタスクにおけるステップ間の依存性と一貫している。特に、ノードはそれぞれ、事前設定されたシーケンスでそれぞれの処理ステップを実行する。例えば、ディシジョンツリー700において、ツリーのレベルに反映されている事前設定されたシーケンスは、図6において示された順序に対応している。U_Aが、その関連する処理ステップをU_Bの前に実行し、U_Bは、その関連する処理ステップをU_Cの前に順番に実行する。
【００８８】
もちろん、ディシジョンツリー700は、一つの例であり、そして、ここにおいて説明されるプロトコルの所与の展開に関連するディシジョンツリーは、変化し得る。例えば、処理ステップとノードとの間に１：１の対応が存在する必要はない。特定的な例において、当事者U_Aは、処理ステップB、並びに処理ステップAを実行するために、呼び出され（called）得る。追加的または代替的に、ステップの一部または全てが、より多くの（例えば、２より多い）可能な出力を有している、ということであり得る。追加的または代替的に、異なるステップは、異なる数の可能なオプションを有している、ということであり得る。さらに、より多くの、または、少ないステップまたはノードが存在する、ということであり得る。
【００８９】
ここにおいて説明されるプロトコルにおいて、特定的なノードに関連する各処理ステップについて、そのノードは、その処理ステップについて所与のノードが有し得る、各可能な結果（outcome）に対応している秘密鍵（private keys）の集合を生成する。特に、当事者は、既に行われた以前の処理ステップの出力の可能な組合せそれぞれのコンテキス

トにおいて選択され得る、各オプションに対応している秘密鍵を選択するように要求される。

【００９０】

図8に示されるこの一つの例は、ディシジョンツリー700を示しており、ディシジョンツリー700の各エッジと関連付けされた様々な秘密鍵によって補強され（augmented）ている。示されるように、秘密鍵は、より以前の処理ステップの出力、並びに、特定の秘密鍵と関連付けされた、そのレベル(すなわち、「現在の（“current”）」処理ステップのレベル)での処理の出力を符号化している（encoding）下付き文字（subscript）を用いて示され得る。例えば、図8において、U_Bによる処理は、４つの可能な結果に対応している、２つの可能な出力を有している。特定的な例において、１つの選択肢は、U_Bについて、U_Aがオプション2を選択した後で、オプション1を選択することである。その選択は、秘密鍵k_{21}に対応している。

【００９１】

この段階において、ノードはそれぞれの秘密鍵ｋを秘密として維持している。

【００９２】

秘密鍵ｋは、当業者にとって公知な技術を使用して生成され得る。さらに、様々な公開鍵暗号システムが採用され得る。例えば、楕円曲線暗号（ECC）が使用されてよい。特定的な例においては、楕円曲線暗号が採用され、そして、Ｐ＝ｋＧであるように、ｋおよび対応する公開鍵Ｐは、楕円曲線暗号の公開鍵－秘密鍵ペアであってよい。ここで、Ｇは、次数ｑ：ｑ×Ｇ＝０の楕円曲線におけるベースポイントであり、ここで、ｑは、大きな素数であり、かつ、０＜ｋ＜ｑである。特定的な例においては、楕円曲線secp256k1－“Standards for Efficient Cryptography2(SE2)”、Certicom Corp、2010年1月27日の文書において定義されたもの－が使用され得る。別の言葉で言えば、各公開鍵およびその対応する秘密鍵は、楕円曲線暗号の公開鍵－秘密鍵ペアを形成することができる。

【００９３】

秘密鍵の生成の後で、対応する公開鍵に基づいて値（value）が生成される。

【００９４】

最初に、スーパーバイザは、公開鍵／秘密鍵のペアを生成し、そして、ここにおいて説明されているプロトコルに参加している全てのノードと、そのペアの公開鍵、Q_Sと示されるもの、を共有する。

【００９５】

次に、参加ノードそれぞれは、順番に、その以前に生成された秘密鍵それぞれに対応している公開鍵の値を生成する。このことは、例えば、ディシジョンツリー900を示している図9において説明されており、ディシジョンツリー700に対応しているが、特定のエッジと関連付けされた公開鍵の値を示すように補強されている。

【００９６】

特に、公開鍵の値または状態値（state value）は、ルートからツリーの所与のエッジまでのエッジと関連付けされている以前に生成された秘密鍵のものに対応している公開鍵それぞれを組合せることによって生成される。別の言葉で言えば、状態値それぞれは、生成された秘密鍵(すなわち、ノードから外へ導く（leading）エッジ上のもの)のうち所与の１つ関連付けされた公開鍵を、ツリーにおいてより初期からの公開鍵値(すなわち、そのノードの中へ導くエッジ上のもの)と組合せることによって生成される。基本ケース（base case）(スーパーバイザの下のツリーの最初のレベル)において、状態値は、Q_Sを、そのレベルの秘密鍵と関連付けされた公開鍵それぞれと組合せることによって生成され得る。

【００９７】

本組合せは、楕円曲線暗号システムの場合、公開鍵を追加することによって生成され得る。例えば、所与の公開鍵が先行する選択値に対して追加され得る。ECCにおいては、秘密鍵ｋに対応する公開鍵Ｐが、Ｐ＝ｋ×Ｇによって定義されることを思い出してみよ。かくして、図9を参照すると、ECCが使用される場合には、$Q_2 = Q_S + k_2 G$であり得る。別の例では、$Q_{21} = Q_2 + k_{21} G$であり得る。

【００９８】
スーパーバイザは、その公開鍵Ｑを、ここにおいて説明されるプロトコルに参加している他のノードそれぞれと共有する。同様に、非スーパーバイザノードそれぞれは、その状態値を、周期的に順序付けられた集合における直前のノードと、および、スーパーバイザノードと共有する。加えて、非スーパーバイザノードそれぞれは、それらの値の(すなわち、それが対応する結果)セマティックな意味を、周期的に順序付けられた集合の直前のノードと、およびスーパーバイザノードと共有する。
【００９９】
注目すべきことに、投票シーケンス（voting sequence）における最終投票者（final voter）の状態値－例えば、図6のU_C－は、ここにおいて説明されるプロトコルに係る可能な結果（O_x）に対応している。スーパーバイザノードは、各結果と直接的に関連付けられる新しい公開鍵／アドレスS_xを、ここにおいて説明されるプロトコルに参加している他のノードに通信することができる。これは、集合のペア｛（O_x、S_x）｝になり、ここで集合｛O_x｝は、集合｛S_x｝の全単射（bijective）である。公開鍵S_xはスーパーバイザノード自体に属するものではなく(すなわち、スーパーバイザノードは対応する秘密鍵を知らなくてよい)が、その代わりに、秘密鍵は、特定の結果と関連付けされた義務（duties）を実行することが課された別個のエンティティに知られ得る。都合のいいことに、このようにして、フォローアップステップが、公開鍵S_xに対応する秘密鍵を所有し得る第三者によって実行され得る。
【０１００】
シークレット値sv_xは、状態値それぞれの(そして、従って、それぞれの可能な結果)に対応する。
【０１０１】
都合のいいことに、ECCが使用される場合、所与の選択値に対応するシークレット値sv_xは、以下のように、選択値に関連する。最初に、楕円曲線の加算に係る同形特性（homomorphic property）に従って、Ｅ（ｍ＋ｎ）＝Ｅ（ｍ）＋Ｅ（ｎ）であることを思い出してみよ。ここで、Ｅは、関数Ｅ（ｘ）＝ｘＧである。さらに、ECCが使用される場合、所与の選択値Q_xは、$Q_x = k_S G + k_A G + \ldots + k_x$によって定義されることを思い出してみよ。ここで、$k_S$は、その公開鍵$Q_S$（つまり、$Q_S = k_S G$）に対応しているスーパーバイザノードの秘密鍵である。別の言葉で言えば、選択値Q_xに対応する秘密鍵sv_xは、秘密鍵の合計 $k_i + k_{i+1} + \ldots + k_x$である。ディシジョンツリー700に関連する特定的な例において、$sv_{21} = k_i + k_{i+1} + \ldots + k_x$、かつ、$Q_{21} = G \times sv_{21}$である。
【０１０２】
ここにおいて説明されるプロトコルは、参加しているノード600間の支払チャネルの回線（circuit）を含んでいる。特に、以下でさらに説明されるように、支払チャネルは、ノード600それぞれと、その直前のノードとの間で確立され、ノードからその直前のノードに対して計算リソースを移転している。例えば、図6において、支払チャネルは、ノード間に示されるエッジの方向によって示されるように、ノード600それぞれと、その直前のノードとの間で確立されるだろう。支払チャネルは、直前のノードの状態値に対応しているシークレット値のうちいずれか１つによってアンロック可能（unlockable）であるように構成されている。支払チャネルによって移転される計算リソースは、ここにおいて説明されるプロトコルがビットコインブロックチェーンに関して動作している場合には、例えば、ビットコインまたはその一部であってよい。
【０１０３】
支払チャネルの使用は、計算リソースのユニットがチャネルについてコミットされていることを必要とする。例えば、支払チャネルの回線において、計算リソースは、各支払チャネルによって、ノードからその直後の支払チャネルへ移転される。
【０１０４】
値（value）の移転は、ここにおいて説明されるプロトコルの中心（central）ではない

ので、この量は、名目上の量（nominal amount）であってよい。明らかになるように、しかしながら、各ノードは、支払チャネルに対してコミットする値の喪失（forfeiture）というペナルティの下で、結果の選択に参加することが必要とされる。従って、また、ノードは、支払チャネルに対する十分に相当量な計算リソースに貢献することが必要とされ、ここにおいて説明されるプロトコルの軽率な（frivolous）参加及び／又は違反を阻止する、ということであってもよい。支払チャネルそれぞれに対してコミットされる計算リソースの量は、同一であってよく、または、別の言葉で言えば、リソースが代替可能（fungible）である場合に、支払チャネルそれぞれに対してコミットされるリソースは同一であってよい。

【０１０５】

回線の支払チャネルの構築の前に、ノード、および、特に、ここにおいて説明されるプロトコルに参加している他のノードとの組合せにおける、スーパーバイザノードは、２つの値、すなわち、時点Ｓおよび時間スパン（time span）ｓについて合意しなければならない。

【０１０６】

最初に、時間スパンｓが選択され、時間スパンは、ここにおいて説明されるプロトコルに参加している各ノードが、以下でさらに説明される、所定のオペレーションを完了するために必要とする時間量を表している。オペレーションは、すなわち、ノードと、その直前のノードとの間に支払チャネルを構築すること、回線において以前の支払チャネルをアンロックするために使用されるシークレット値を引き出す（retrieving）こと、ノードと関連付けされた処理ステップについてさらに可能な結果のうち１つを選択すること、および、そのノードに賛成する別の支払チャネルの支払トランザクションをブロックチェーンに対してサブミットすること、である。ノードのうち任意の１つは、他のユーザのコンセンサスを考慮して、時間ｓを選択することができる。いくつかの実装において、スーパーバイザノードは、プロトコルを単純化し得るｓを選ぶことができる。時間ｓは、例えば、秒またはブロック数といった、適切な単位で表され得る。

【０１０７】

次に、値Ｓは、ブロックチェーン・ネットワークへサブミットされているノードに対する最初の移転の開始時間として選択される。ノードのいずれも、他のノードのコンセンサスを考慮して、Ｓを選ぶことができる。いくつかの実装において、開始者（initiator）は、プロトコルの実装を単純化し得るＳを選ぶことができる。

【０１０８】

特に、Ｓは、例えば、ユニックス時間（unix time）またはブロック高さのいずれかにおいて、指定され得るポイントを表し、一方で、sは、時間スパンを表している。

【０１０９】

支払チャネルは、スーパーバイザノードと、その直後のノード(すなわち、最終的な選択を行う責任を負うノード)との間の支払チャネルで始まる、時計回り方法で構築される。例えば、図6を参照すると、構築されるべき最初の支払チャネルはU_0とU_Cとの間のものであり、かつ、構築されるべき最後の支払チャネルはU_AとU_0との間のものであろう。

【０１１０】

この順序付けされたシーケンスは、各ノードが、その直後のノードに賛成して（in favor of）支払チャネルにリソースをコミットする前に、各ノードに賛成する支払チャネルが存在することを保証している。しかしながら、スーパーバイザノードは、この保証を受け取らない。なぜなら、その秘密鍵k_sは、さまざまな支払いチャネルそれぞれをアンロックするために使用され得るものとして、全てのシークレット値のコンポーネントだからである。しかしながら、k_sは、この段階ではスーパーバイザノードによって秘密として保持されているので、スーパーバイザノードは、それに賛成する支払チャネルの保証がなくても、回線内で最初の支払いチャネルを安全に生成することができる。特に、スーパーバイザノードに賛成する支払チャネルが生成されない場合、生成した支払チャネルの払戻しトランザクションをサブミットするのに十分な期間だけ待つことができる。

【０１１１】
図10、図13および図14において示される一連のフローチャートを参照して、これから、ここにおいて説明されるプロトコルに参加しているノードによって実行される例示的なオペレーションが説明される。これらのフローチャート及びそれらの以下の説明において、プロトコルに参加している周期的に順序付けられた集合に係るｎノードは、$U_0 \ldots U_{n-1}$として識別される。それらは、スーパーバイザノードU_0から始まり、周期的に順序付けられた集合のノードを通って、様々な支払チャネルの構築の順序においてスーパーバイザノードに直接先行するノード(すなわち直前のノード)に至るまでのものである。－つまり、意図された処理におけるステップの順序とは逆の順序である。例えば、処理ステップの意図された順序が、図6に示されるように、反時計回りの方向であるとして描かれている場合、図10、図13、および図14において参照されるようなノード$U_0 \ldots U_{n-1}$の順序は、時計方向に従う。特定的な例においては、そうした番号付けに従って、図6においてU_Bとラベル付けされたノードがU_iとして識別される場合に、U_A(図6)は、U_{i+1}であり、U_C(図6)は、U_{i-1}である。言い換えると、図10、図13、および図14において示されるフローチャートの目的のために、図6においてU_Bとラベル付けされたノードは、ノードU_Aの直前のノードとみなされ、そして、U_Bとラベル付けされたノードは、ノードU_Cの直後のノードとみなされるだろう。
【０１１２】
図10について、これから説明される。図10は、ここにおいて説明されるプロトコルに参加している特定的なノードと関連付けされたオペレーションを説明するフローチャート1000を示しており、生成のために準備すること、および、ノードからその直前のノードへの支払チャネルを生成することにおけるオペレーションを実行している。様々な非スーパーバイザの参加ノードは、上述の支払チャネル回線の生成において、フローチャート1000に示されるオペレーションに対応する動作をそれぞれ実行することができる。オペレーション1010以降は、参加ノードのうち、特定的な、非スーパーバイザノードの１つに係る１つまたはそれ以上のプロセッサによって実行される。かくして、オペレーション1010以降は、コンピューティングデバイスの１つまたはそれ以上のプロセッサによって実行される。例えば、メモリ520のストレージといった、コンピュータで読取り可能な記憶媒体に保管され得るようなコンピュータで実行可能な命令を含むソフトウェアを実行する、例示的なコンピューティングデバイス500の適切に構成されたインスタンスのプロセッサ510(図5)、といったものである。
【０１１３】
オペレーション1010においては、特定的なノード（particular node）が、周期的に順序付けられた集合における特定的なノードの直後のノードから、直後のノードと関連付けされた状態値を受信する。かくして、状態値の定義および周期的に順序付けられた集合におけるノードの順序付けにより、特定的なノードは、直後のノードからスーパーバイザノードの直前のノードまで、周期的に順序付けられた集合におけるノード(例えば、方向付けされたリング610)と関連付けされた処理ステップの可能な結果の組合せに対応している値を受信する。上述のように、状態値は、ノードと関連付けされた公開鍵に基づいており、そして、これらの公開鍵それぞれは、そのノードと関連付けされた処理ステップの可能な出力に対応する。例えば、ECCが使用される場合、受信された状態値それぞれは、上述の、スーパーバイザノードと関連付けされた公開鍵と、特定の結果に対応している公開鍵それぞれを合計することによって決定され得る。
【０１１４】
オペレーション1010に続いて、制御フローは、オペレーション1020へ進む。
【０１１５】
オペレーション1020において、特定的なノードは、そのノードについて状態値を生成する。別の言葉で言えば、特定的なノードは、特定的なノードによって選択可能な出力に対応している状態値を生成する。状態値は、受信した状態値、および、そのノードと関連付けされた公開鍵(上記で説明されたように、そのノードと関連付けされた秘密鍵k_xに対

応するもの)に基づいて生成される。かくして、状態値は、受信された値、および、特定的なノードと関連付けされた処理ステップの可能な結果に対応している公開鍵に基づいて生成される。

【０１１６】

オペレーション1020に続いて、制御フローは、オペレーション1030へ進む。

【０１１７】

オペレーション1030において、状態値は、上述のように、スーパーバイザノード、および、周期的に順序付けられた集合における特定的なノードの直前のノードと共有される。

【０１１８】

オペレーション1030から、制御フローは、オペレーション1040へ進む。

【０１１９】

オペレーション1040において、特定的なノードは、直前のノードによる、それに賛成する支払チャネルの生成を待つことができる。上記の支払ルートそれぞれを生成するために、様々なトランザクションが、上記のように準備される。例えば、トランザクションT_{pay}は、その支払チャネルに対するロック値を使用して準備される。そのトランザクションは、直前のノードと関連付けされた送信元アドレスから特定的なノードの受信アドレスに対して、リソース－例えば、ｘユニットの計算リソース、または、ｘ個のBTC－を送信するように構成されている。そのトランザクションに従って、リソースの制御が、ロック値のいずれか１つ(すなわち、状態値のいずれか１つ)に対応しているアンロッキング値(sv_x)の供給を含む、実行条件の満足に応答して送信される。特に、アンロッキング値それぞれは、それぞれの状態値に対応しており、そして、その選択値が基づいている公開鍵に対応している秘密鍵に基づいている。例えば、ECCが使用される場合、各アンロッキング値は、それらの秘密鍵と、スーパーバイザノードの公開鍵Q_Sに対応している秘密鍵k_Sとの合計であってよい。かくして、特定的なノードは、直前のノードによって準備され、かつ、直前のノードと関連付けされた送信元アドレスから特定的なノードの受信アドレスへリソースの制御を送信するように構成された、ブロックチェーントランザクションの生成を待つことができる。

【０１２０】

次に、特定的なノードが、状態値を使用して、準備され、直後のノードに賛成する特定的なノードによって、第２の支払チャネルが生成される。例えば、特定的なノードは、それに賛成する支払チャネルが生成された後で、そうした第２の支払チャネルを準備することができる。そうした支払チャネルを準備することは、上記の第１の支払チャネルの説明に類似しており、特定的なノードと関連付けされた送信元アドレスから、直後のノードの受信アドレスへ、第２のリソースの制御を送信するように構成されたブロックチェーントランザクションを準備することを含む。支払チャネルは、実行条件の満足に応答するリソースの制御を送信するように構成されており、受信された値に対応している(第２の)複数のアンロッキング値のうち任意の１つの供給を含んでいる。

【０１２１】

図11は、前述の第１および第２の支払チャネルに対応し得るといった、$U_a \rightarrow U_b$支払チャネルの表示1100を提供している。

【０１２２】

図示されるように、チャネルは、３つのトランザクションT_c、T_{pay}、およびT_rを含んでいる。σ（U_x）は、U_xの暗号署名を表していることに留意する。

【０１２３】

T_cトランザクションは、支払チャネルのコミットメント・コンポーネントを表している。ここで、トランザクションを通して、U_bは、２オブ２マルチシグネチャ(U_b、U_a)、または、複数のシークレット値$\{sv\}_{(a)}$およびU_aの暗号署名のうち任意の１つの知識、のいずれかによって支配される、指定された数のユニットを送信／コミットする。

【０１２４】

T_rトランザクションは、指定された時間が経過した後でブロックチェーンに対するサ

ブミットが適格になる、指定された数のユニットを(コミットメント・トランザクションから)U_bに戻す払戻し（refund）を表している。かくして、T_rは、時間満了の戻り条件が満足されると、特定的なノード、U_bに対してソースの制御を戻すように構成されている。このトランザクションが成功裡に実行されるためには、暗号化された署名U_aおよび署名U_bを必要とする。

【０１２５】

T_{pay}トランザクションは、U_bのコミットメントされたファンドからU_aへの指定された数のユニットの移転である。このトランザクションが成功裡に実行されるためには、複数のシークレット値｛ｓｖ｝（ａ）(ディシジョンツリーのそのエレメント／レベルと関連付けされた状態値に対応しているもの)のうち任意の１つ、および、U_aの暗号署名を必要とする。

【０１２６】

ブロックチェーンがビットコインブロックチェーンである場合に、T_c、T_r、およびT_{pay}それぞれはビットコイントランザクションであってよい。

【０１２７】

図12は、上述のように、ノードから別のノードU_bへの支払いを生成することにおいて、ノードU_bによって実行され得るようなオペレーションを説明しているフローチャート1200を示している。オペレーション1210以降は、ノードU_bによって実行される。かくして、オペレーション1210以降は、コンピューティングデバイスの１つまたはそれ以上のプロセッサによって実行される。例えば、メモリ520のストレージといった、コンピュータで読取り可能な記憶媒体に保管され得るようなコンピュータで実行可能な命令を含むソフトウェアを実行する、例示的なコンピューティングデバイス500の適切に構成されたインスタンスのプロセッサ510(図5)、といったものである。

【０１２８】

オペレーション1210において、U_bは、コミットメント・トランザクション、T_cを生成する。トランザクションT_cは、上述のとおりである。

【０１２９】

オペレーション1210から、制御フローは、オペレーション1220へ進む。

【０１３０】

オペレーション1220において、払戻しトランザクション、T_rが生成される。トランザクションT_rは、上述のとおりである。T_rは、Ｓ＋（ａ＋１）ｓの満了（expiry）について指定された時間を用いて生成される。ここで、ａは、回線内の最初の支払いトランザクション(例えば、U_0からU_c)について0であり、そして、回線内で、その後ろの各支払いチャネルについて、１ずつ増分される(例えば、ａ＝０、１、２、３、...)。

【０１３１】

オペレーション1220から、制御フローは、オペレーション1230へ進む。

【０１３２】

オペレーション1230において、U_bは、払戻しトランザクションに署名する。

【０１３３】

オペレーション1230から、制御フローは、オペレーション1240へ進む。

【０１３４】

上述のように、支払チャネルにおける他の参加者、U_a、も、また、払戻しトランザクションT_rに署名することになる。オペレーション1240においては、U_aが払戻しトランザクションT_rに署名するか否かが決定される。もしそうでなければ、支払チャネルの生成は失敗し、そして、中止される。代替的に、U_aが払戻しトランザクションT_rに署名した場合、制御フローは、オペレーション1250へ進む。

【０１３５】

オペレーション1250において、U_bは、支払いトランザクションT_{pay}を生成する。トランザクションT_{pay}は、上述のとおりである。

【０１３６】

オペレーション1250から、制御フローは、オペレーション1260へ進む。

【０１３７】

オペレーション1260において、U_bは、支払いトランザクション（payment transaction）T_cに署名し、そして、それをブロックチェーンに対してサブミットする。

【０１３８】

オペレーション1260に続いて、制御フローは終了し、支払チャネルが生成されている。

【０１３９】

支払い回線が完了した後で、参加ノードそれぞれは、順番に、オプションを選択することができる。概要として、ここにおいて説明されるプロトコルに参加している各ノードは、順番に、オプションを選択し、そして、次いで、その選択に基づいて、そのノードに賛成するT_{pay}トランザクションを償還（redeem）する。

【０１４０】

図13は、オペレーション1310及びその先を説明するフローチャート1300を示しており、結果の選択において、ここにおいて説明されるプロトコルに参加しているノードによって実行され得るものである。かくして、オペレーション1310及びその先は、様々なコンピューティングデバイスの１つまたはそれ以上のプロセッサによって実行される。例えば、メモリ520のストレージ、といったコンピュータで読取り可能な記憶媒体に記憶され得る、コンピュータで実行可能な命令を含むソフトウェアを実行する例示的なコンピューティングデバイス500の適切に構成されたインスタンスに係る、例えば、プロセッサ510(図5)といったものである。

【０１４１】

上述のように、フローチャート1300及びその説明は、フローチャート1000の表記と一貫した表記を使用している。別の言葉で言えば、ノードは、スーパーバイザノードU_0から始まり、支払チャネルの構築の順番で番号付けされている。

【０１４２】

オペレーション1310においては、ここにおいて説明されるプロトコルにおける必要なステップを完了するのために十分な時間が残っていることを確保するためにチェックが行われる。特に、現在時刻が最初の移転の開始時間、Ｓ、と比較されてよく、支払チャネルのいずれもタイムアウトすることなく回線を完了することができるように、十分な時間が存在することを確保することができる。特に、$t < S + s$であるか否かが決定される。ここで、ｔは、現在時刻である。

【０１４３】

オペレーション1310でのチェックが失敗すると、ここにおいて説明されるプロトコルは中止される。そうでなければ、制御フローは、オペレーション1320へ進み、そこで、スーパーバイザノードは、賛成するT_{pay}ブロックチェーントランザクションをサブミットする。これは、その特定の支払チャネルをロックするために使用されるQ_Sに対応する値として、ブロックチェーン上で値k_Sを明らかにするという副作用（side effect）を有している。

【０１４４】

オペレーション1320に続いて、第１の処理ステップと関連付けされたノード、U_{n-1}、－すなわち、投票する第１のノード－に対応するように、カウンタｉが初期化される。制御フローは、次いで、オペレーション1330に進み、そこで、ループチェックは、プロトコルに参加している各ノードが関連する処理ステップを完了したか否かを判断し、そして、結果が決定される。もしそうであれば、制御フローは終了する。代替的に、全ての処理ステップが未だ完了していない場合(例えば、オペレーション1330を通る最初のパス上)、制御フローは、オペレーション1340へ進む。

【０１４５】

オペレーション1340においては、（オペレーション1320、または、以下で説明されるように、オペレーション1350でブロックチェーンにサブミットされたであろう）支払いチャネルの支払いトランザクションにおいて使用されるアンロッキング値、sv_{previ}

ｏｕｓ、が、ブロックチェーンから抽出される。－すなわち、U_iから$U_{(i+1) \bmod n}$への支払いチャネルに係る支払いトランザクションからである。都合のいいことに、$sv_{previous}$は、直後のノード(すなわち、前の処理ステップと関連付けされたノード)が出力を識別した後でだけサブミットされるブロックチェーントランザクションから抽出されるので、処理ステップの順序付けが実施される。

【０１４６】

オペレーション1340から、制御フローは、オペレーション1350へ進む。

【０１４７】

オペレーション1350において、ノードU_iは、さらに、そのノードと関連付けされた処理ステップに対して可能な結果のうち１つを識別する。そのノードは、次いで、値$sv_{previous}$を、識別された出力に対応する公開鍵に対応している秘密鍵(k_{ijk})と組合せることによって、対応するアンロッキング値を決定される。例えば、ECCが使用されている場合、U_iは、$sv_{ijk} = sv_{previous} + k_{ijk}$を計算することによってシークレット値$sv_{ijk}$を決定した。特に、このことは、様々な$T_{pay}$トランザクションをロックするために使用される様々な状態値の定義それぞれにより、かつ、同形特性により動作する。

【０１４８】

処理ステップの実行に関して、ノード30それぞれによって、様々な方法でステップが実行され得る。例えば、ノード300が自動運転車両のモジュールに対応する場合、処理ステップは、プログラムに従って出力を選択し得る。例えば、そのノードに対して提供された入力データに基づく、といったものである。入力データは、例えば、ユーザから、といった、１つまたはそれ以上の入力装置を介して受信されたデータを含む、ということであり得る。

【０１４９】

オペレーション1350から、制御フローはオペレーション1360へ進む。オペレーション1360においては、トランザクションを完了するのに十分な時間が残っているか否かが判断される。特には、$t < S + (n - i)s$であるか否かが判断される。ここで、ｔは、現在時刻であり、そして、ｉは、スーパーバイザノードから数えた現在のノードのインデックスである。スーパーバイザノードは0であり、スーパーバイザノードの直後のノード(すなわち、最後の処理ステップと関連付けされたノード)はインデックス1を有し、そして、上記に説明したように、インデックス（ｎ－ｉ）を有する第１の処理ステップと関連付けされたノードに至るまで同じように続く。

【０１５０】

オペレーション1360から、十分な時間が残っていると判断される場合に、制御フローは、オペレーション1380へ進む。代替的に、十分な時間が残っていないと判断される場合に、制御フローは、オペレーション1370へ進む。

【０１５１】

オペレーション1370において、十分な残りの時間の欠如は、払戻しトランザクションとして、ここにおいて説明されたプロトコルの中止をU_{n-1}にトリガーさせることによって、処理される。特に、U_{n-1}は、ブロックチェーンに対して、周期的に順序付けられたノードの集合における直後のノード、U_i、に賛成する払戻しトランザクションをサブミットし、それによって、プロセスを開始し、最終的な処理ステップを実行するように意図されたノード(すなわち、スーパーバイザノードの直後のノード)、U_1、に至るまでのノードに、それぞれの払戻しトランザクションをブロックチェーンに対してサブミットさせる。

【０１５２】

オペレーション1380において(十分な時間が残っている際)は、ノードU_iが、それに賛成するT_{pay}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に対してサブミットし、それによって、ブロックチェーン上で$sv_{current}$を明らかにしている。

【０１５３】

オペレーション1380に続いて、カウンタｉは、選択シーケンスにおける次のノードに対応するように１つ減らされ（decremented）、そして、次いで、制御フローは、(上述のように)オペレーション1330へ戻り、それによって、シーケンスにおける次のノードによる選択を処理するようにループしている。

【０１５４】

フローチャート1300は、全体としての分散コンピューティングタスクに参加しているノードにわたる処理ステップの実行を説明している。図14は、タスク処理に伴うオペレーションの実行において、ここにおいて説明されるプロトコルに参加している特定的なノードによって実行されるオペレーションを説明するフローチャート1400を示している。様々な非スーパーバイザな参加ノードは、上述の支払チャネル回線を生成する際に、それぞれ、フローチャート1400で説明されるオペレーションに対応する動作を実行することができる。オペレーション1410及びその先は、参加ノードのうち、特定的で、非スーパーバイザな１つのノードに係る１つまたはそれ以上のプロセッサによって実行される。かくして、オペレーション1410及びその先は、コンピューティングデバイスの１つまたはそれ以上のプロセッサによって実行される。例えば、メモリ520のストレージ、といったコンピュータで読取り可能な記憶媒体に記憶され得る、コンピュータで実行可能な命令を含むソフトウェアを実行する例示的なコンピューティングデバイス500の適切に構成されたインスタンスに係る、例えば、プロセッサ510(図5)といったものである。

【０１５５】

上述のように、フローチャート1400及びその説明は、フローチャート1000の表記と一貫した表記を使用している。別の言葉で言えば、ノードは、スーパーバイザノードU_0から始まり、支払チャネルの構築の順番で番号付けされている。

【０１５６】

最初に、特定的なノードは、直後のノードが選択を行うのを待つことを確保する必要がある。従って、オペレーション1410において、特定的なノードは、特定的なノードから直後のノードへの支払チャネルと関連付けされたブロックチェーントランザクションT_{pay}が実行されること、すなわち、アンロックされて、かつ、ブロックチェーンに対してサブミットされるのを待つ。ブロックチェーンに対してサブミットされたトランザクションに続いて、制御フローは、オペレーション1420へ進む。

【０１５７】

オペレーション1420において、特定的なノードは、ブロックチェーンから、オペレーション1410で検出されたブロックチェーントランザクションT_{pay}をアンロックするために使用される、アンロッキング値、$sv_{previous}$、を抽出する。上述のように、値$sv_{previous}$は、特定的なノードの前に選択しているノードと関連付けされた処理ステップの出力を示している。かくして、値$sv_{previous}$は、直後のノードからスーパーバイザノードの直前のノードに至るまでの周期的に順序付けられた集合におけるノードと関連付けされた処理ステップの出力を示している。さらに、上述のように、値$sv_{previous}$は、それらのノードと関連付けされた公開鍵に対応し、かつ、それらのノードによって以前に選ばれ、または選択された前述の選択されたオプションに対応している秘密鍵に基づくものである。別の言葉で言えば、値$sv_{previous}$は、直後のノードの選択値に対応している。例えば、上述のように、ECCが使用される場合、アンロッキング値は、直後のノードからスーパーバイザノードの直前のノードに至るまで周期的に順序付けられた集合におけるノードと関連付けされた処理ステップの出力に対応している秘密鍵と、スーパーバイザノードと関連付けされた公開鍵（Q_s）に対応している秘密鍵（k_s）との合計であり得る。

【０１５８】

次に、オペレーション1430において、特定的なノードは、さらに、そのノードと関連付けされた処理ステップに対して可能な結果のうち１つを上述の方法で識別する。上述のように、ノードは、例えば、そのノードと関連付けされた入力及び／又は状態に応じてプログラムに従って、様々な方法で出力を識別することができる。例えば、入力の一部は、

例えば、ユーザから、といった、入力インタフェースを介して受信される、ということであり得る。

【０１５９】

オペレーション1430から、制御フローは、オペレーション1440へ進む。

【０１６０】

上述のように、特定的なノードの公開鍵は、そのノードと関連付けされたタスク処理の可能な結果それぞれに対応している。オペレーション1440において、アンロッキング値は、オペレーション1420で抽出されたアンロッキング値$sv_{previous}$と、識別された結果に対応する公開鍵に対応している秘密鍵とに基づいて決定される。例えば、ECCが使用される場合、値は、上記の方法で加算することによって組合され得る。

【０１６１】

オペレーション1440から、制御フローは、オペレーション1450へ進む。

【０１６２】

オペレーション1450においては、直前のノードと特定的なノードとの間の支払チャネルと関連付けされたトランザクションを実行するために、オペレーション1440で決定されたアンロッキング値が使用される。特に、特定的なノードは、オペレーション1440で決定されたアンロッキング値を使用して対応するT_{pay}トランザクションをアンロックし、そして、次いで、アンロックされたトランザクションをブロックチェーンに対して送信し得る。例えば、T_{pay}トランザクションは、アンロッキング値をトランザクションのアンロッキングスクリプトの中へサブミットすることにより、アンロックされ得る。特に、上述のように、オペレーション1440で決定されたアンロッキング値を用いてトランザクションをアンロックすること、そして、それをブロックチェーンに対してサブミットすることは、ブロックチェーン上でアンロッキング値を明らかにするという副作用を有している。

【０１６３】

オペレーション1450に続いて、特定的なノードによる選択が完了する。

【０１６４】

ここにおいて説明されたプロトコルは、多種多様なアプリケーションに適用することができる。例えば、自動運転車両400といった、自動運転車両について適用され得る。いくつかのそうした実装においては、ここにおいて説明されるプロトコルに参加している所与のノードと関連付けされた処理ステップの可能な結果が、車両の制御システムの関連するモジュールの出力に対応し得る、ということであり得る。例えば、出力はモジュール410-440の出力であってよい。

【０１６５】

ここにおいて説明されるプロトコルを所与のブロックチェーン上で実装することにおいては、様々な実装の詳細が考慮されることを必要とする。例えば、ビットコインブロックチェーンにおいては、トランザクションが有効のままである一方で、トランザクションの関連するトランザクションIDが変化し得る、という可能性がある。これは、トランザクションの展性（malleability）問題として参照され得る。このことは、払戻しトランザクション（T_r）および支払トランザクション（T_{pay}）が以前のトランザクションの識別子を参照する方法で実行され得る場合の、支払チャネルに依存する、ここにおいて説明されるプロトコルといったプロトコルの実装における問題を提起し得る。ビットコインにおけるトランザクションの展性に関する追加情報は、例えば、Bitcoin Magazineに掲載された、Aaron van Wirdumによる、“The Who, What, Why and How of the Ongoing Transaction Malleability Attack”(2015年10月7日)において見出される。トランザクションの展性は、ビットコインのSeg Wit forkによって排除され得る。－例えば、https://bitcoincore.org/en/2016/01/26/segwit-benefits/(2016年10月19日更新)から利用可能な“Segregated Witness Benefits”を参照のこと。

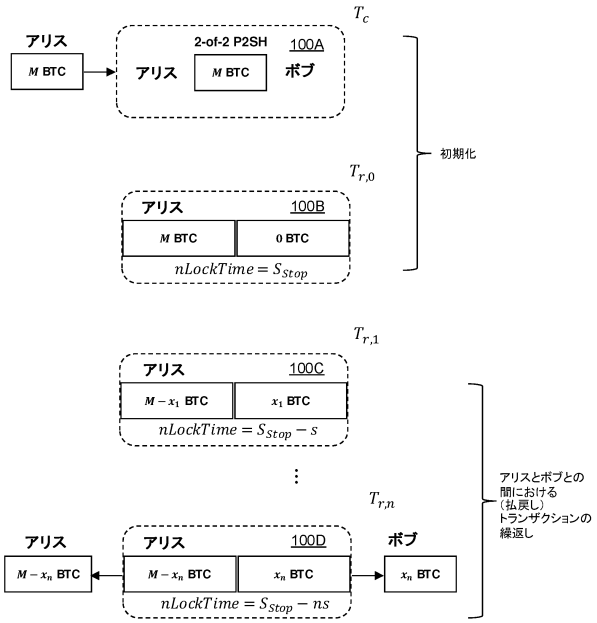
【０１６６】

上述の実施形態は、本発明の技術的事項（subject matter）を限定するものではなく

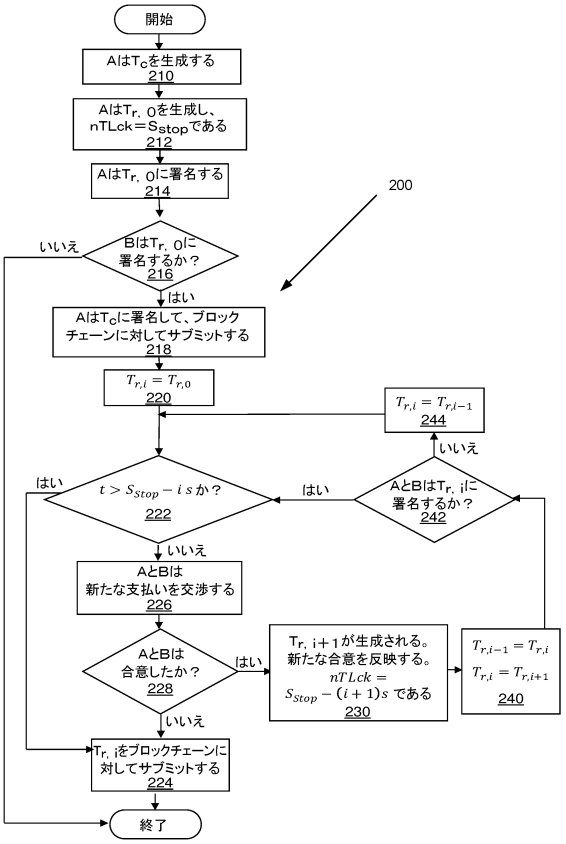
、むしろ例示するものであり、そして、当業者であれば、添付の請求項によって定義されるような本発明の範囲から逸脱することなく、多くの代替的な実施形態を設計することができることが留意されるべきである。請求項においては、括弧内に置かれたあらゆる参照符号は、クレームを限定するものとして解釈されない。用語「含む（“comprising”および“comprises”）」、等は、任意の請求項または明細書全体において列挙されたもの以外のエレメントまたはステップの存在を排除するものではない。本明細書において、「含む（“comprises”）」は、「有する又は構成される（“includes or consists of”）」を意味し、そして、「含んでいる（“comprising”）」は、「有している又は構成されている（“including or consisting of”）」を意味するものである。エレメントの単数形の参照は、そうしたエレメントの複数形の参照を排除するものではなく、そして、その逆もまた同様である。本発明の技術的事項は、いくつかの別個のエレメントを含むハードウェア手段によって、および、適切にプログラムされたコンピュータ手段によって実施され得る。いくつかの手段を列挙している装置クレームにおいて、これらの手段のうちいくつかは、１つの、そして、同一のハードウェアアイテムによって具現化され得る。所定の手段が相互に異なる従属クレームにおいて記載されているという単なる事実は、これらの措置の組合せが有利に利用できないことを示すものではない。

【図面】

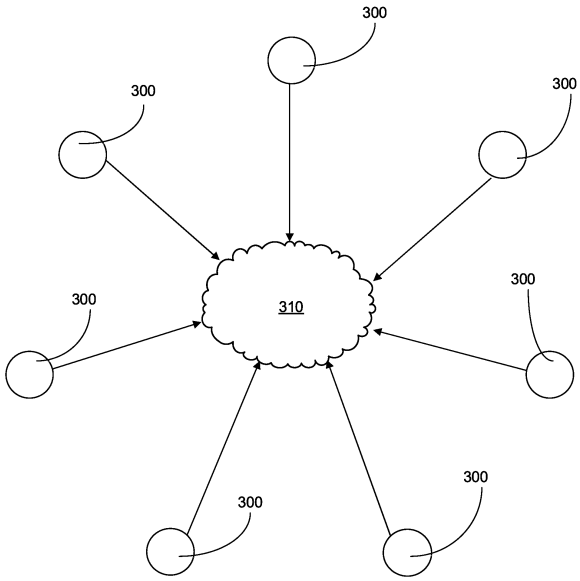
【図１】



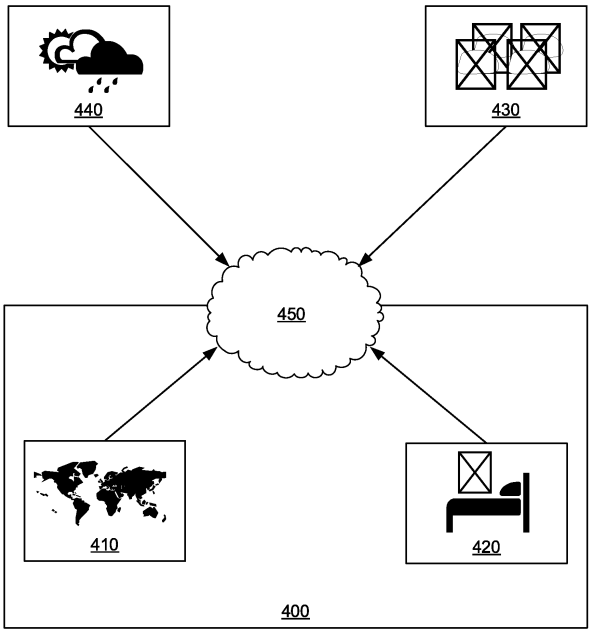
【図２】



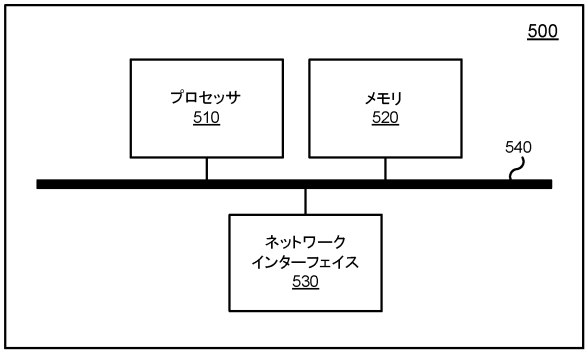
【図３】


300
300
300
300
310
300
300
300

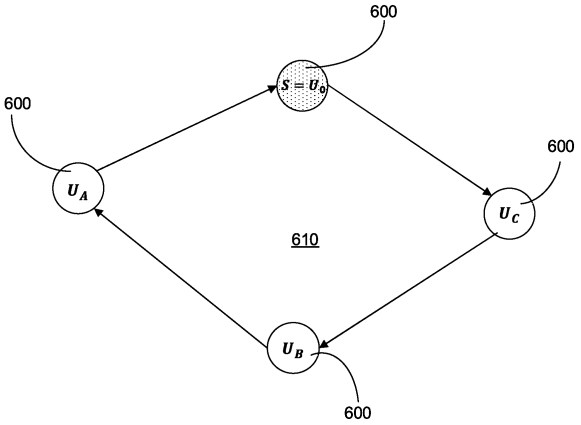
【図４】


440
430
450
410
420
400

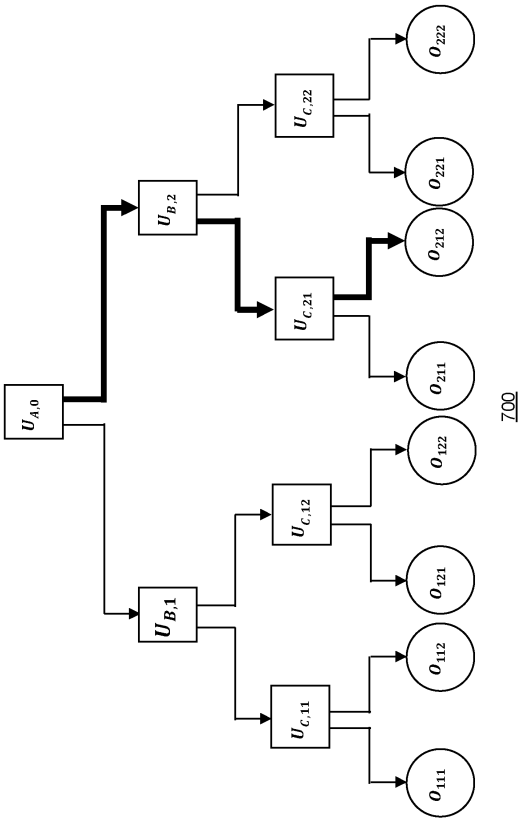
【図５】


500
プロセッサ
510
メモリ
520
540
ネットワーク
インターフェイス
530

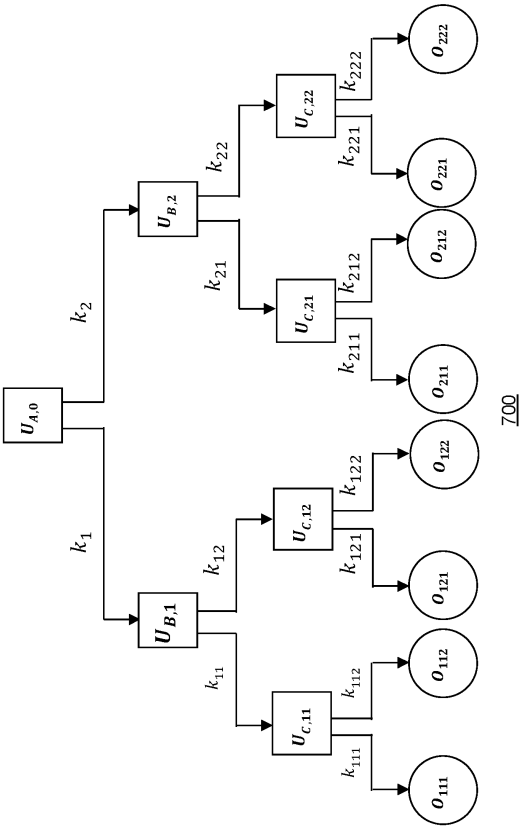
【図６】


600
$S = U_0$
600
U_A
600
U_C
610
U_B
600

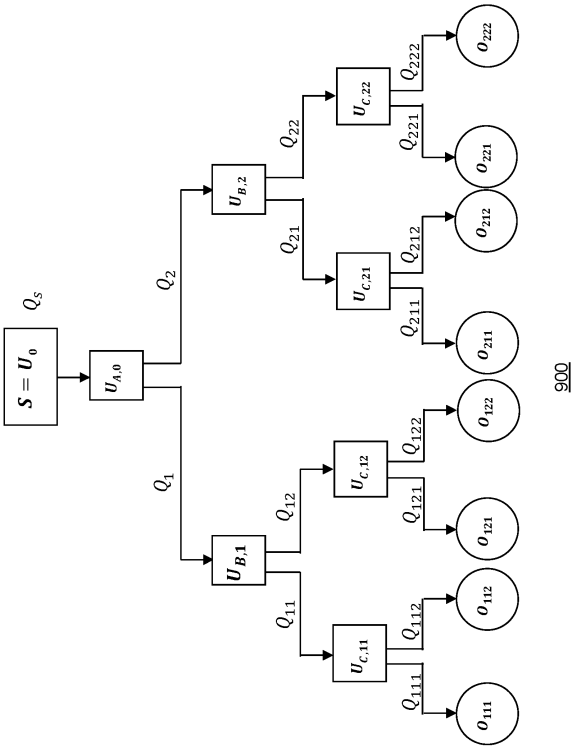
【図７】


$U_{A,0}$
$U_{B,2}$
$U_{B,1}$
$U_{C,22}$
$U_{C,21}$
$U_{C,12}$
$U_{C,11}$
O_{222}
O_{221}
O_{212}
O_{211}
O_{122}
O_{121}
O_{112}
O_{111}
700

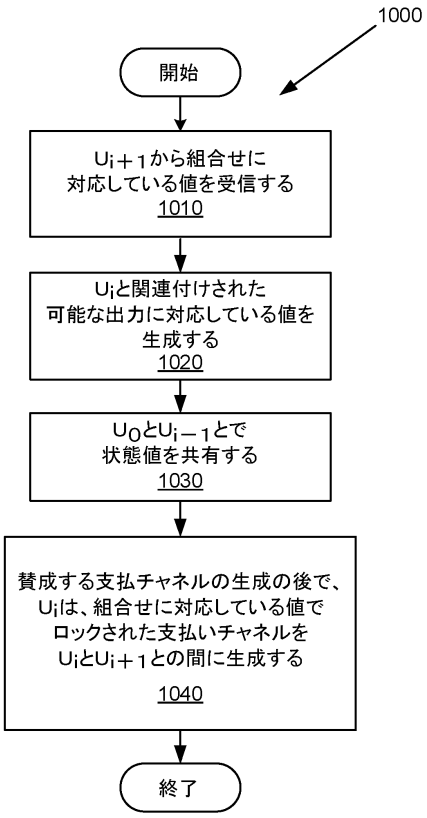
【図８】


$U_{A,0}$
$U_{B,2}$
$U_{B,1}$
$U_{C,22}$
$U_{C,21}$
$U_{C,12}$
$U_{C,11}$
k_2
k_1
k_{22}
k_{21}
k_{12}
k_{11}
k_{222}
k_{221}
k_{212}
k_{211}
k_{122}
k_{121}
k_{112}
k_{111}
O_{222}
O_{221}
O_{212}
O_{211}
O_{122}
O_{121}
O_{112}
O_{111}
700

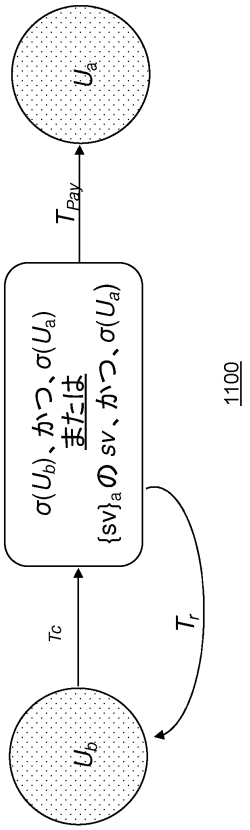
【図９】


$S = U_0$
Q_S
$U_{A,0}$
$U_{B,2}$
$U_{B,1}$
$U_{C,22}$
$U_{C,21}$
$U_{C,12}$
$U_{C,11}$
Q_2
Q_1
Q_{22}
Q_{21}
Q_{12}
Q_{11}
Q_{222}
Q_{221}
Q_{212}
Q_{211}
Q_{122}
Q_{121}
Q_{112}
Q_{111}
O_{222}
O_{221}
O_{212}
O_{211}
O_{122}
O_{121}
O_{112}
O_{111}
900

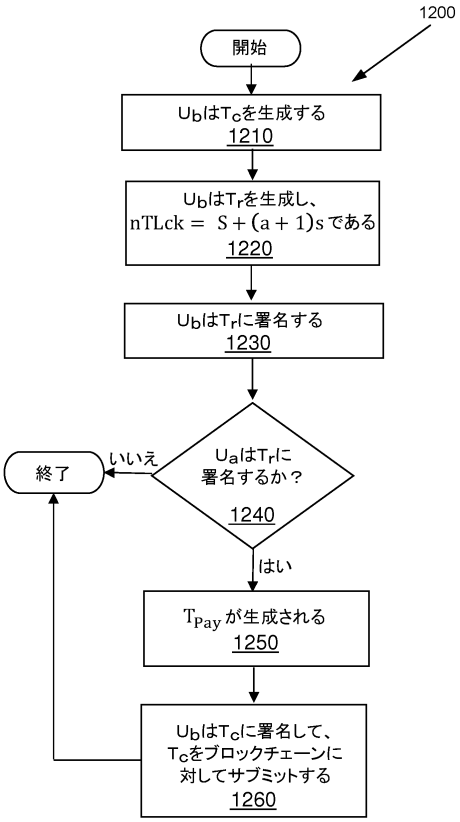
【図１０】


1000
開始
U_{i+1}から組合せに
対応している値を受信する
1010
U_iと関連付けされた
可能な出力に対応している値を
生成する
1020
U_0とU_{i-1}とで
状態値を共有する
1030
賛成する支払チャネルの生成の後で、
U_iは、組合せに対応している値で
ロックされた支払いチャネルを
U_iとU_{i+1}との間に生成する
1040
終了

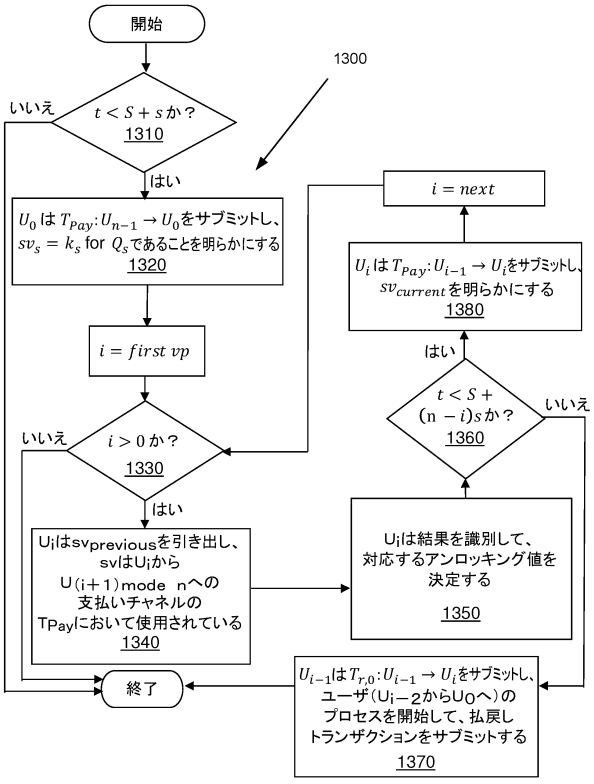
【図１１】



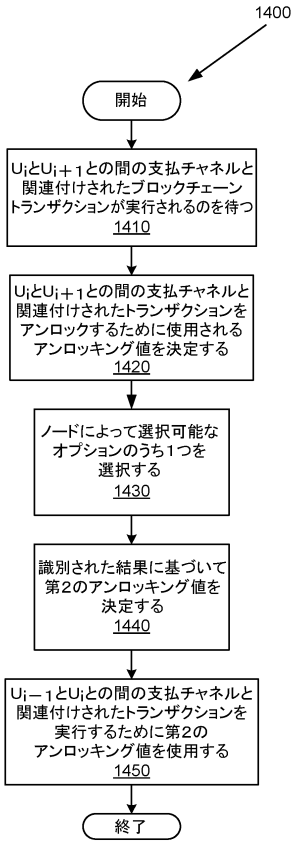
【図１２】



【図１３】



【図１４】



フロントページの続き

内
(72)発明者　ヴァルトルッチ，シルヴィア
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
(72)発明者　ベルナト，ポーリーン
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 内
審査官　行田 悦資
(56)参考文献　米国特許出願公開第２０１７／０２８６９５１（ＵＳ，Ａ１）
国際公開第２０１７／１４５００８（WO，Ａ１）
国際公開第２０１７／１４５０１６（WO，Ａ１）
国際公開第２０１７／１４５００３（WO，Ａ１）
国際公開第２０１７／１８７３９５（WO，Ａ１）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／３２