(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7346537号
(P7346537)

(45)発行日　令和5年9月19日(2023.9.19)

(24)登録日　令和5年9月8日(2023.9.8)

(51)国際特許分類

Ｈ０４Ｌ　9/32　(2006.01)		Ｈ０４Ｌ	9/32　２００Ｚ
Ｇ０６Ｑ　20/04　(2012.01)		Ｇ０６Ｑ	20/04

ＦＩ

請求項の数　15　（全36頁）

(21)出願番号　特願2021-500561(P2021-500561)
(86)(22)出願日　令和1年7月15日(2019.7.15)
(65)公表番号　特表2021-530172(P2021-530172A)
(43)公表日　令和3年11月4日(2021.11.4)
(86)国際出願番号　PCT/IB2019/056014
(87)国際公開番号　WO2020/016739
(87)国際公開日　令和2年1月23日(2020.1.23)
審査請求日　令和4年6月17日(2022.6.17)
(31)優先権主張番号　1811773.9
(32)優先日　平成30年7月19日(2018.7.19)
(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】　分散型システムの処理ステップを制御するための、コンピュータにより実施されるシステム及び方法

(57)【特許請求の範囲】

【請求項１】

コンピュータにより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する第１ノードセットのうちの第１イニシエータノードにより、秘密鍵を生成するステップであって、前記第１ノードセットは循環順序付きである、ステップと、

前記第１イニシエータノードにより、前記第１ノードセットのノードの各々について、前記秘密鍵の暗号シェアを生成し、前記暗号シェアのそれぞれ１つを前記第１ノードセットの他のノードに分配するステップと、

前記第１イニシエータノードにより、前記暗号シェアの各々に対応する公開鍵に基づき、第１ロック値を決定するステップと、

前記第１ロック値に対応する第１アンロック値の供給を含む実行条件の充足に応答して、前記第１イニシエータノードにより、前記第１イニシエータノードに関連するソースアドレスから、前記第１ノードセットの中で前記第１イニシエータノードの直後のノードの受信側アドレスへ、リソースの制御を渡すよう構成されるトランザクションを準備するステップと、

前記第１イニシエータノードにより、前記第１ノードセットの中の隣接ノードの各ペアの間のトランザクション回路を形成するために、更なるトランザクションの準備を開始するステップであって、前記更なるトランザクションの各々は、それぞれのロック値に対応するそれぞれのアンロック値の供給を含む実行条件の充足に応答して、各ペアの第１ノー

ドに関連付けられたアドレスから、該ペアの第２ノードに関連付けられたアドレスへ、それぞれのリソースの制御を渡すよう構成され、各ペアの前記第２ノードは、前記第１ノードセットの中で該ペアの前記第１ノードの直後にあり、前記それぞれのロック値は、所与のペアの第１ノードに分配された暗号シェア及び前記第１ノードセットの中の該ノードの直前にあるノードから受信した値に基づき決定される、ステップと、
を含む方法。
【請求項２】
前記第１イニシエータノードにより、循環順序付きイニシエータノードセットのマスタイニシエータノードへ、前記秘密鍵に対応する公開鍵を送信するステップであって、前記イニシエータノードセットのイニシエータノードの各々は、複数の循環順序付きノードセットのうちのそれぞれ１つのイニシエータノードであり、前記複数の循環順序付きノードセットは前記第１ノードセットを含む、ステップと、
前記第１イニシエータノードにより、第２イニシエータノードから、前記第１イニシエータノードから前記マスタイニシエータノードまで、セットの中の各ノードに関連付けられた公開鍵に基づく第１値を受信するステップであって、前記第２イニシエータノードは、前記イニシエータノードセットの中で前記第１イニシエータノードの直前にある、ステップと、
を更に含む請求項１に記載の方法。
【請求項３】
前記第１イニシエータノードにより、前記第１値及び前記秘密鍵に対応する前記公開鍵に基づき、第２ロック値を決定するステップと、
前記第１イニシエータノードにより、前記第２ロック値に対応する第２アンロック値の供給を含む実行条件の充足に応答して、前記第１イニシエータノードに関連付けられたソースアドレスから、第３イニシエータノードに関連付けられた受信側アドレスへ、リソースの制御を渡すよう構成されるトランザクションを準備するステップであって、前記第３イニシエータノードは、前記イニシエータノードセットの中で前記第１イニシエータノードの直後にある、ステップと、
を更に含む請求項２に記載の方法。
【請求項４】
前記第１イニシエータノードにより、前記第３イニシエータノードから前記マスタイニシエータノードまで、前記イニシエータノードセットの中の各ノードに関連付けられた公開鍵に対応する秘密鍵に基づき、第２値を取得するステップ、
を更に含む請求項３に記載の方法。
【請求項５】
前記第２値は、ブロックチェーンから、前記第１イニシエータノードにより取得される、請求項４に記載の方法。
【請求項６】
前記第２値は、前記第３イニシエータノードから、前記第１イニシエータノードにより取得される、請求項４に記載の方法。
【請求項７】
前記第１イニシエータノードにより、前記トランザクション回路の実行が開始されるべきではないと決定するステップと、
前記トランザクション回路の実行が開始されるべきではないと決定することに応答して、
前記第１イニシエータノードにより、前記第２値及び前記秘密鍵に基づき、第３アンロック値を決定するステップと、
前記第１イニシエータノードにより、前記第３アンロック値を用いて、前記第２イニシエータノードにより準備された、第２イニシエータノードに関連付けられたソースアドレスから前記第１イニシエータノードの受信側アドレスへリソースの制御を渡すよう構成されたトランザクションを実行するステップであって、前記リソースの制御は、前記第３アンロック値の供給を含む実行条件の充足に応答して渡される、ステップと、

を更に含む請求項４～６のいずれか一項に記載の方法。
【請求項８】
前記第１ロック値は、前記第２値に更に基づく、請求項４～７のいずれか一項に記載の方法。
【請求項９】
前記第１イニシエータノードにより、前記トランザクション回路の実行が開始されるべきであると決定するステップと、
前記トランザクション回路の実行が開始されるべきであると決定することに応答して、
前記第１イニシエータノードにより、前記第２値および前記秘密鍵の前記暗号シェアのそれぞれ１つに基づき、第４アンロック値を決定するステップと、
前記第１イニシエータノードにより、前記第４アンロック値を用いて、前記第１ノードセットの中の前記イニシエータノードの直前にあるノードにより準備された、前記直前のノードに関連付けられたソースアドレスから前記イニシエータノードの受信側アドレスへリソースの制御を渡すよう構成されたトランザクションを実行するステップであって、前記リソースの制御は、前記第４アンロック値の供給を含む実行条件の充足に応答して渡される、ステップと、
を更に含む請求項４～８のいずれか一項に記載の方法。
【請求項１０】
前記第１ノードセットの中の隣接ノードの各ペアの間のトランザクション回路を形成するための更なるトランザクションの準備を開始する前記ステップは、前記第１イニシエータノードにより、前記第１ノードセットの中の前記第１イニシエータノードの直後にあるノードへ、前記第１ロック値を送信するステップを含む、請求項１～９のいずれか一項に記載の方法。
【請求項１１】
前記リソースの各々は、前記リソースの他のものと同じ量である、請求項１～１０のいずれか一項に記載の方法。
【請求項１２】
各公開鍵及びその対応する秘密鍵は、楕円曲線暗号公開－秘密鍵ペアを形成する、請求項１～１１のいずれか一項に記載の方法。
【請求項１３】
前記秘密鍵の前記暗号シェアは、公に検証可能な暗号シークレット共有アルゴリズムを用いて生成される、請求項１～１２のいずれか一項に記載の方法。
【請求項１４】
請求項１～１３のいずれか一項に記載の方法を実行する、コンピュータにより実装されたシステム。
【請求項１５】
コンピュータシステムを請求項１～１３のいずれか一項に記載の方法を実行するようにする命令を格納した非一時的コンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、概して、分散型コンピューティングシステムに関し、より具体的には、ブロックチェーンネットワークを用いる順序付き処理ステップの指示及び制御を含む、分散型システムのノードに関連付けられた処理ステップの制御及び調整に関する。
【背景技術】
【０００２】
分散型コンピューティングシステムでは、分散型システムの中のノードとして動作する様々なコンピューティング装置が、ネットワークを介して通信してよい。メッセージは、これらのノードの間で交換されてよい。このようなメッセージの交換は、例えば、コンピューティングタスクを実行するためにノードが協力することを可能にする。このようなタ

スクは、様々なノードに渡り分散された処理を含んでよい。このような分散された処理は、様々なノードに関連付けられたステップの制御及び調整を必要とすることがある。例えば、特定の処理順序が強いられることがある。

【０００３】

分散型システムの一例は、航空機情報システムの状況で生じ得る。例えば、航空機情報システムは、種々のサブシステムを含んでよく、各サブシステムはサブコンポーネントで構成される。これらのサブシステムの各々は、例えば、飛行前検査を実行するときのような、特定の処理を実行することを担ってよい。これらのステップを実行するために、サブシステムの各々は、サブコンポーネントに、特定の処理ステップを実行することを要求し得る。

【０００４】

このような分散型システムでは、処理の中のステップの順序及び／又は特定の処理を形成するステップの順序を強制可能であることが必須であり又は望ましいことがある。例えば、サブシステムは、それらそれぞれの処理を順番に実行する必要があってよい。更に、サブコンポーネントに関連付けられたノードは、それらそれぞれの処理ステップを順番に実行する必要があってよい。

【０００５】

更に、処理ステップ又は特定の処理の改ざん防止又は改ざん防止監査証跡記録アウトプットを提供することが望ましいことがある。このような監査証跡は、様々なシナリオにおいて用途がある。例えば、航空機のコンテキストでは、これは例えば、例えば事故後に事故を法的に調査可能にし得る。特定の例では、このようなシステムは、例えば、根本原因決定のために、障害を航空機の特定のサブシステムに帰属させることを可能にし得る。

【０００６】

本願明細書では、私たちは、全ての形式の電子的な、コンピュータに基づく、分散型台帳を包含するために用語「ブロックチェーン」を使用する。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、並びにこれらの変形を含む。他のブロックチェーン実装が提案され開発されているが、ブロックチェーン技術の最も広く知られているアプリケーションは、Ｂｉｔｃｏｉｎ台帳である。Ｂｉｔｃｏｉｎは、ここでは、便宜上及び説明の目的で参照されることがあるが、本開示はＢｉｔｃｏｉｎ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実装及びプロトコルが本開示の範囲に包含されることに留意すべきである。

【０００７】

ブロックチェーンは、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の電子台帳であり、ブロックにより構成され、ブロックはまた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の中の参加者間で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、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、これらのブロックは一緒に繋げられて、起源以来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記録を生成する。トランザクションは、スクリプトとして知られている小さなプログラムを含む。スクリプトは、それらのインプット及びアウトプットを埋め込まれ、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であるかを指定する。Ｂｉｔｃｏｉｎプラットフォームでは、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。

【０００８】

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は、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、無効なトランザクションがネットワークから拒否され、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する。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、未使用トランザクション（unspent transaction, UTXO）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、UTXOに対してこの検証作業を実行する。ロック及びアンロックスクリプトの実行が真（TR

UE）と評価する場合、トランザクションは有効であり、トランザクションはブロックチェーンに書き込まれる。したがって、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は、（ｉ）トランザクションを受信した第１ノードにより検証され、トランザクションが有効な場合には、ノードが該トランザクションをネットワーク内の他のノードに中継する、（ｉｉ）マイナーにより構築された新しいブロックに追加される、（ｉｉｉ）マイニングされる、つまり過去のトランザクションのパブリック台帳に追加される、ことが必要である。

【０００９】

ブロックチェーン技術は、暗号通貨の実装の使用のために最も広く知られているが、デジタル事業家が、Ｂｉｔｃｏｉｎの基づく暗号セキュリティシステム及び新しいシステムを実装するためにブロックチェーンに格納できるデータの両方の使用を開発し始めている。ブロックチェーンが、暗号通貨の分野に限定されない自動化タスク及びプロセスのために使用できれば、非常に有利になる。このようなソリューションは、ブロックチェーンの利益（例えば、永久性、イベントの記録の耐タンパ性、分散型処理、等）を利用しながら、それらの用途をより多様化し得る。

【００１０】

ブロックチェーンネットワークに参加している循環順序付きノードセットの中のノードにより実施され得る分散型システムにおける処理ステップの制御及び調整のための方法、システム、及びコンピュータ可読記憶媒体は、共同所有の英国特許出願番号第１８０６４４８.５号、名称「Computer－Implemented Methods and Systems」、２０１８年４月２０日出願、に開示されている。該出願の主題は、多因子依存性決定（Multifactor Dependent Decisions (MDD)）プロトコルと呼ばれるプロトコルの説明を含む。該出願の主題は、分散型システムのノードの間のステップの順序づけを提供するために利用され得る。例えば、それは、システム内の処理ステップの順序づけを提供するために使用されてよい。

【００１１】

ブロックチェーンネットワークに参加している循環順序付きノードセットにより形成される分散型システム内の機能を提供するための他の方法、システム、及びコンピュータ可読記憶媒体は、共同所有の英国特許出願番号第１７０６０７１.６号、第１８０２０６３.６号、それぞれ名称「Computer－Implemented Methods and Systems」、それぞれ２０１７年４月１８日及び２０１８年２月８日出願に開示されている。前者は、エンティティに関連付けられたアドレス間のリンクを不明瞭にすると共に、該エンティティがそのリソースを着服する可能性を除去しながら、資金をあるアドレスから別のアドレスへと移動させ得るグループランダム交換（Group Random Exchange (GRE)）という名称のプロトコルに関連する開示を含む。概略では、ＧＲＥは、ノードのグループの各メンバがコンピューティングリソースのｘ個の単位を他のメンバに移転することに合意するというグループの概念の上に構築される。その結果、全員が、コンピューティングリソースのｘ個の単位を受け取る。該出願は、ＧＲＥプロトコルの変形である、各支払いチャネルに対して異なるシークレットを使用しながら同一の機能を提供するようＧＲＥプロトコルを変更したグループランダム交換変更シークレット（Group Random Exchange Change Secret (GRECS)）に関連する開示も含む。特に、ＧＲＥＣＳプロトコルに参加している各ノードは、それぞれのシークレット値をイニシエータノードへ送信し、イニシエータノードは次にこれらの値を用いて移転を開始する。後者は、ＧＲＥプロトコルに対する改良、より具体的にはＧＲＥＣＳプロトコルに関する開示を含む。この改良されたプロトコルグループランダム交換遅れた開示（Group Random Exchange Late Reveal (GRELR)）は、共通シークレットを使用する必要を回避することにより、ＧＲＥＣＳを改良する。

【００１２】

しかしながら、特に、上述のように、システムのコンポーネントの間、例えばサブシステムの間で処理順序を制御できるだけでなく、特定のサブシステムを構成するサブコンポ

ーネントの間でも処理順序を制御できることが望ましいことがある。更に、特定のサブシステムによる処理が任意的にスキップされ又は省略可能であることが望ましいことがある。

【００１３】

このような改良されたソリューションがここで考案される。

【発明の概要】

【００１４】

したがって、本開示によると、添付の請求項において定められる方法が提供される。

【００１５】

本開示により、コンピュータにより実施される方法が提供され得る。当該方法は、ブロックチェーンネットワークに参加している第１循環順序付きノードセットのうちの第１イニシエータノードにより、秘密鍵を生成するステップを含んでよい。当該方法は、第１イニシエータノードにより、第１セットのうちのノードの各々の秘密鍵の暗号シェアを生成するステップと、暗号シェアのうちのそれぞれを第１セットの他のノードに分配するステップと、を含んでよい。当該方法は、第１イニシエータノードにより、暗号シェアのそれぞれに対応する公開鍵に基づき、第１ロック値を決定するステップを含んでよい。当該方法は、第１イニシエータノードにより、第１ロック値に対応する第１アンロック値の供給を含む実行条件の充足に応答して、リソースの制御を第１イニシエータノードに関連付けられたソースアドレスから第１セットの中の第１イニシエータノードの直後にあるノードの受信側アドレスへ渡すよう構成されるトランザクションを準備するステップを含んでよい。当該方法は、第１イニシエータノードにより、第１セットの中の隣接ノードの各ペアの間のトランザクション回路を形成するために、更なるトランザクションの準備を開始するステップを含んでよい。更なるトランザクションの各々は、それぞれのロック値に対応するそれぞれのアンロック値の供給を含む実行条件の充足に応答して、それぞれのリソースの制御を各ペアの第１ノードに関連付けられたアドレスから該ペアの第２ノードに関連付けられたアドレスへ渡すよう構成される。各ペアの第２ノードは、第１セットの中で、該ペアの第１ノードの直後にあってよい。それぞれのロック値は、所与のペアの第１ノードに分配された暗号シェア、及び第１セットの中の害ノードの直前にあるノードから受信した値に基づき決定されてよい。

【００１６】

幾つかの実装では、当該方法は、第１イニシエータノードにより、循環順序付きイニシエータノードセットのマスタイニシエータノードへ、秘密鍵に対応する公開鍵を送信するステップを含んでよい。イニシエータノードセットの各々のイニシエータノードは、服すの循環順序付きノードセットのうちのそれぞれ１つのイニシエータノードであってよい。複数の循環順序付きノードセットは、第１セットを含んでよい。当該方法は、第１イニシエータノードにより、第２イニシエータノードから、第１イニシエータノードからマスタイニシエータノードまで、該セットの中の各ノードに関連付けられた公開鍵に基づき第１値を受信するステップを含んでよい。第２イニシエータノードは、イニシエータノードセットの中で第１イニシエータノードの直前にあってよい。

【００１７】

幾つかの実装では、当該方法は、第１イニシエータノードにより、第１値および秘密鍵に対応する公開鍵に基づき第２ロック値を決定するステップを含んでよい。当該方法は、第１イニシエータノードにより、第２ロック値に対応する第２アンロック値の供給を含む実行条件の充足に応答して、リソースの制御を第１イニシエータノードに関連付けられたソースアドレスから第３イニシエータノードの受信側アドレスへ渡すよう構成されるトランザクションを準備するステップを含んでよい。第３イニシエータノードは、イニシエータノードセットの中で第１イニシエータノードの直後にあってよい。

【００１８】

幾つかの実装では、当該方法は、第１イニシエータノードにより、第３イニシエータノードからマスタイニシエータノードまで、イニシエータノードセットの中の各ノードに関連付けられた公開鍵に対応する秘密鍵に基づき、第２値を取得するステップを含んでよい。

【００１９】
幾つかの実装では、第２値は、第１イニシエータノードによりブロックチェーンから取得されてよい。
【００２０】
幾つかの実装では、第２値は、第１イニシエータノードにより第３イニシエータノードから取得されてよい。
【００２１】
幾つかの実装では、当該方法は、第１イニシエータノードにより、トランザクション回路の実行が開始されるべきでないと決定するステップを含んでよい。当該方法は、トランザクション回路の実行が開始されるべきではないと決定することに応答して、第１イニシエータノードにより、第２値及び秘密鍵に基づき、第３アンロック値を決定するステップと、第１イニシエータノードにより、第３アンロック値を用いて、第２イニシエータノードにより準備された、第２イニシエータノードに関連付けられたソースアドレスから第１イニシエータノードの受信側アドレスへリソースの制御を渡すよう構成されたトランザクションを実行するステップと、を含んでよい。リソースの制御は、第３アンロック値の供給を含む実行条件の充足に応答して渡されてよい。
【００２２】
幾つかの実装では、第１ロック値は、第２値に更に基づいてよい。
【００２３】
幾つかの実装では、当該方法は、第１イニシエータノードにより、トランザクション回路の実行が開始されるべきであると決定するステップと、トランザクション回路の実行が開始されるべきであると決定することに応答して、第１イニシエータノードにより、第２値及び秘密鍵の暗号シェアのそれぞれ１つに基づき、第４アンロック値を決定するステップと、第１イニシエータノードにより、第４アンロック値を用いて、第１循環順序付きノードセットの中のイニシエータノードの直前にあるノードにより準備された、直前のノードに関連付けられたソースアドレスからイニシエータノードの受信側アドレスへリソースの制御を渡すよう構成されたトランザクションを実行するステップと、を含んでよい。リソースの制御は、第４アンロック値の供給を含む実行条件の充足に応答して渡されてよい。
【００２４】
幾つかの実装では、第１セットの中の隣接ノードの各ペアの間のトランザクション回路を形成するための更なるトランザクションの準備を開始するステップは、第１イニシエータノードにより、第１セットの中の第１イニシエータノードの直後にあるノードへ、第１ロック値を送信するステップを含んでよい。
【００２５】
幾つかの実装では、リソースの各々は、他のリソースと同じ量であってよい。
【００２６】
幾つかの実装では、各公開鍵及びその対応する秘密鍵は、楕円曲線暗号公開－秘密鍵ペアを形成してよい。
【００２７】
幾つかの実装では、秘密鍵の暗号シェアは、公に検証可能な暗号シークレット共有アルゴリズムを用いて生成されてよい。
【００２８】
本開示は、プロセッサとメモリとを含むシステムも提供する。メモリは、プロセッサによる実行の結果として、システムに本願明細書に記載のコンピュータにより実施される方法のいずれかの実施形態を実行させる実行可能命令を含む。
【００２９】
本開示は、実行可能命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記実行可能命令は、コンピュータシステムのプロセッサにより実行された結果として、少なくとも、前記コンピュータシステムに、本願明細書に記載のコンピュータにより実施される方法を実行させる、非一時的コンピュータ可読記憶媒体も提供する。

【００３０】
本開示のこれらの及び他の態様は、本願明細書に記載の実施形態から明らかであり及びそれらを参照して教示される。本開示の実施形態は、単なる例を用いて及び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】
【００３１】
【図１】支払いチャネルで使用されるトランザクションを示す図である。
【００３２】
【図２】図１に従う支払いチャネルがどのように生成されるかを示すフローチャートである。
【００３３】
【図３】本願の例示的な動作環境を示す簡略概略図である。
【００３４】
【図４】本願の別の例示的な動作環境を示す簡略概略図である。
【００３５】
【図５】例示的なコンピューティング装置を示す。
【００３６】
【図６】ノードにより構成される循環順序付きセットを示す図である。
【００３７】
【図７】複数の循環順序付きセットに概念的に対応し得る２次元構造を示す簡略概略図である。
【００３８】
【図８】図７に示したものと類似する２次元構造のより詳細な表現を提供する図である。
【００３９】
【図９】図８の循環順序付きセットのうちの標準的な１つを示す図である。
【００４０】
【図１０】図８のイニシエータノードにより構成される循環順序付きセットを示す図である。
【００４１】
【図１１】イニシエータノードの間の回路を形成する支払いチャネルの構成のための動作を示すフローチャートである。
【００４２】
【図１２】所与の循環順序付きノードセットのノードの間の回路を形成する支払いチャネルの構成のための動作を示すフローチャートである。
【００４３】
【図１３】ノード間の支払いチャネルに関連付けられ得るアンロック値を示すための図８の注釈されたバージョンを提供する。
【００４４】
【図１４】図１１及び１２のフローチャートの中の動作がどのように結合され得るかを示すフローチャートである。
【００４５】
【図１５】支払いチャネルのコミットメントチャネルバージョンの表現を提供する。
【００４６】
【図１６】支払いチャネルのコミットメントトランザクションとして機能し得る例示的なＢｉｔｃｏｉｎトランザクションを示す。
【００４７】
【図１７】支払いチャネルの返金トランザクションとして機能し得る例示的なＢｉｔｃｏｉｎトランザクションを示す。
【００４８】
【図１８】支払いチャネルの支払いトランザクションとして機能し得る例示的なＢｉｔｃ

ｏｉｎトランザクションを示す。
【００４９】
【図１９】図１９のコミットメントトランザクションの態様が図１８の支払いトランザクションの態様とどのように結合され得るかを示す。
【００５０】
【図２０】順序付き処理を制御し及び実行するときに、種々の順序付きセットのノードにより実行される動作を示すフローチャートである。
【発明を実施するための形態】
【００５１】
図中の同様の参照符号は同様の要素及び特徴を示すために使用される。
【００５２】
本開示を実現するプロトコルは、本願明細書では、それぞれが順序付き方法で決定を行うために、協力するノードのグループについて説明される。該プロトコルは、処理タスクの中で前のステップのアウトプットから導出された累積シークレット値を利用することによってのみ復元できるコンピューティングリソースの単位をコミットするようノードに要求するという概念の上に構築される。
【００５３】
幾つかの概念は、背景として以下に記載される。
【００５４】
本願明細書に記載されるプロトコルは、既存のＢｉｔｃｏｉｎに関連する技術、支払いチャネル（Payment Channels）（例えば、「Introduction to Micropayment Channels」、http://super３.org/introduction－to－micropayment－channels/を参照）の上に構築される。これは、参加者のペアの間のオフブロックＢｉｔｃｏｉｎトランザクションのために設計された技術であり、特に返金トランザクションの使用を組み入れる。
【００５５】
＜支払いチャネル＞
上述のように、支払いチャネルは以下の説明で言及され、支払いチャネルの概要は読者の便宜のために以下に示される。
【００５６】
支払いチャネルは、全部のトランザクションをブロックチェーンにコミットすることなく、パーティが複数の暗号通貨トランザクションを生成するために設計された技術である。標準的な支払いチャネルの実装では、ほぼ無限の量の支払いを行うことができるが、ブロックチェーンに２つのトランザクションを追加する必要があるだけである。
【００５７】
ブロックチェーンに追加されるトランザクション数が削減され、及び関連するコストが削減されることに加えて、支払いチャネルは、速度の利点も提供し、重要なことに、物事が計画通りに進まなかった場合に、又はいずれかの参加者が特定の払いセットを超えて進めないと決定した場合に、パーティが単位を返金される能力も有する。支払いチャネルの実装の説明は以下に概略を示す。
【００５８】
ＡｌｉｃｅがＢｏｂへブロックチェーンリソースを移転する必要のあるシナリオを検討する。これは、ＡｌｉｃｅからＢｏｂへ、状況の要求に応じて、時間期間に渡り、複数の支払いを要求してよい。Ａｌｉｃｅは、可能な交換セットの中で、最大で例えば（全部で）１５ＢＴＣを、Ｂｏｂへ送金することを期待している。これを進めるために、支払いチャネルがＡｌｉｃｅとＢｏｂとの間に確立され、以下のように動作する。
【００５９】
先ず、Ａｌｉｃｅは、Ａｌｉｃｅに由来する１５ＢＴＣをコミットする、Ａｌｉｃｅ及びＢｏｂの両者により制御される２－ｏｆ－２マルチシグネチャＰ２ＳＨ（pay－to－script hash）トランザクションＴｃを生成する。このとき、トランザクションは、Ｂｉｔｃ

ｏｉｎネットワークに提出されない（このようなマルチシグネチャアドレスは、２人の個人（Ａｌｉｃｅ及びＢｏｂ）が、このアドレスからの金銭を使う任意のトランザクションに暗号的に署名することを要求する）。

【００６０】

次に、Ａｌｉｃｅは、マルチシグネチャにより制御される資金の全部の単位をＡｌｉｃｅに返金する別個の返金トランザクション$T_{r,0}$を生成する。このトランザクションは、１００ブロックのｎＬｏｃｋＴｉｍｅ値を含む（ｎＬｏｃｋＴｉｍｅは、指定された時間が経過した後にのみ、Ｂｉｔｃｏｉｎトランザクションが実行可能になることを可能にするＢｉｔｃｏｉｎトランザクションパラメータである）。Ｂｏｂはトランザクションに署名する。この返金トランザクションは、ｎＬｏｃｋＴｉｍｅが終了した後に、Ａｌｉｃｅ及びＢｏｂの間の交換が食い違った場合に、Ａｌｉｃｅが返金されることを可能にする。

【００６１】

次に、Ａｌｉｃｅは元のトランザクションT_Cに署名する。

【００６２】

このとき、Ａｌｉｃｅ及びＢｏｂは、（ブロックチェーン外の）移転がＡｌｉｃｅからＢｏｂへと行われることを反映するために、新しい返金トランザクションを生成することに進んでよい。これらの返金トランザクションは、その時点でＡｌｉｃｅがＢｏｂへと移転することを要求されるリソースの総量を反映し得る。一例として、ＡｌｉｃｅがＢｏｂに５ＢＴＣを送金する場合、５ＢＴＣをＢｏｂへ送信し１０ＢＴＣをＡｌｉｃｅへと戻すアウトプットを有する新しい返金トランザクション$T_{r,i}$が生成される。ＡｌｉｃｅがＢｏｂに別の５ＢＴＣを送金する必要がある場合、１０ＢＴＣをＢｏｂへ及び５ＢＴＣをＡｌｉｃｅへ送信するアウトプットを有する新しい返金トランザクション$T_{r,i+1}$が生成される。新しい返金トランザクション毎に、彼らは、両方のパーティがトランザクションに署名するがネットワークにトランザクションを提出する必要がないことを詳細に合意していると仮定する。

【００６３】

生成された各々の連続する返金トランザクションは、前の返金トランザクションより少ないｎＬｏｃｋＴｉｍｅを有することに留意する。つまり、$nLockTime(T_{r,i+1}) < nLockTime(T_{r,i})$である。

【００６４】

参加者が任意の$T_{r,i}$に署名することを拒否した場合、不当な扱いを受けた参加者は単に$T_{r,i-1}$を提出してよい。最悪のシナリオでは、（ｎＬｏｃｋＴｉｍｅが終了した後に、）Ａｌｉｃｅは$T_{r,0}$に署名し、それをネットワークに提出して彼女の単位の全部を取り戻す。

【００６５】

最終的に構成される返金トランザクションは、ＡｌｉｃｅからＢｏｂへ移転される単位の総和を表す。このトランザクションは、ネットワークに提出される。

【００６６】

図１は、支払いチャネルの中で使用されるトランザクションT_C１００Ａ、及び$T_{r,n}$１００Ｄを示す。Ｍは、ＡｌｉｃｅからＢｏｂへ送金され得る最大金額を表す。x_iは、ＡｌｉｃｅがＢｏｂに支払う必要のある単位の現在の総量である。S_{stop}は、初期返金トランザクションにおけるｎＬｏｃｋＴｉｍｅである。ｎは、ＡｌｉｃｅとＢｏｂの間で行われる進行中の（オフブロック）移転の中で生成された返金トランザクションの数である（これは、初期返金トランザクションを含まない）。ｓは、他のパーティが前の返金トランザクションを提出し実質的にＡｌｉｃｅとＢｏｂとの間の交換を終了するリスクをパーティが負う前に、返金トランザクションに合意するために両方の参加者（Ａｌｉｃｅ及びＢｏｂ）に割り当てられた時間である。

【００６７】

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

【００６８】
図１のトランザクションＴ$_{C}$１００Ａ、Ｔ$_{r,0}$１００Ｂ、Ｔ$_{r,1}$１００Ｃ、及びＴ$_{r,n}$１００Ｄは、ブロックチェーン上に現れ得るトランザクションである。
【００６９】
ＡｌｉｃｅとＢｏｂのあいだの支払いチャネルを構成する動作は、図２のフローチャート２００に示される。動作２１０以降は、コンピュータ可読記憶媒体に格納され得るようなコンピュータ実行可能命令を含むソフトウェアを実行する１つ以上のコンピューティング装置の１つ以上のプロセッサにより実行される。
【００７０】
動作２１０で、Ａｌｉｃｅに関連付けられたコンピューティング装置のプロセッサは、上述の方法で２－ｏｆ－２マルチシグネチャＰ２ＳＨ（pay－to－script hash）トランザクションＴ$_{C}$を生成する。
【００７１】
動作２１０から、制御フローは動作２１２へ進む。動作２１２で、Ａｌｉｃｅに関連付けられたコンピューティング装置のプロセッサは、上述の方法で、マルチシグネチャにより制御される資金の全部の単位をＡｌｉｃｅに関連付けられたアカウントに返金する別個の返金トランザクションＴ$_{r,0}$を生成する。
【００７２】
動作２１２から、制御フローは動作２１４へ進む。動作２１４で、Ａｌｉｃｅに関連付けられたコンピューティング装置のプロセッサは、上述の返金トランザクションに署名する。
【００７３】
動作２１４から、制御フローは動作２１６へ進む。動作２１４で、Ｂｏｂに関連付けられたコンピューティング装置のプロセッサも、上述の返金トランザクションに署名してよい。トランザクションがこのように署名された場合、制御フローは動作２１８へ進む。代替として、トランザクションが署名されない場合、支払いチャネルの生成は中止される。
【００７４】
動作２１８で、Ａｌｉｃｅに関連付けられたコンピューティング装置のプロセッサは、Ｔ$_{C}$に署名し、それをブロックチェーンに提出する。制御フローは、次に、動作２２０へ進む。
【００７５】
動作２２０で、上述の返金トランザクションは、第１返金トランザクションとして認識され、ＡｌｉｃｅからＢｏｂへの更なる送金が後に交渉できるようにする。
【００７６】
動作２２０から、制御フローは動作２２２へ進む。動作２２２で、更なる送金を交渉するために十分な時間が残っていることが決定される。十分な時間が残っていない場合、制御フローは、動作２２４へと進み、最後の返金トランザクションがブロックチェーンに提出される。代替として、十分な時間が残っている場合、制御フローは動作２２６へ進む。
【００７７】
動作２２６で、ＡｌｉｃｅとＢｏｂの間で更なる送金が交渉される。動作２２６から、制御フローは動作２２８へと進み、交渉が成功したかどうかが決定される。交渉が成功しなかった場合、制御は上述の動作２２４へと進む。代替として、交渉が成功すると、結果として制御フローは動作２３０へ進む。
【００７８】
動作２３０で、成功した交渉に起因する新しい返金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、動作２４０へと進み、新しい返金トランザクションが現在返金トランザクションとして認識される。
【００７９】
動作２４０から、制御フローは動作２４２へと進み、Ａｌｉｃｅ及びＢｏｂに関連付けられたコンピューティング装置のプロセッサは、現在返金トランザクションに署名してよ

い。署名した場合、制御フローは、上述の動作２２２へと進む。代替として、トランザクションが署名されなかった場合、動作２４４で、現在返金トランザクションの認識は、前の返金トランザクションへと復帰する。動作２４４から、制御フローは、上述の動作２２２に戻る。

【００８０】

＜シークレット共有＞

【００８１】

以下に更に説明するように、本願の主題では、閾値シークレット共有方式が利用される。

【００８２】

幾つかの例では、閾値シークレット共有プロトコルは、（ｔ；ｎ）閾値により定められてよい。ここで、ｎは参加ノードの数であってよく、ｔ＋１はシークレットを再構成することを要求されるノードの最小数であってよい。シークレット共有方式は、閾値暗号システムの例であり、シークレットは、ｎ個のノードの間で部分に分けられてよく、少なくともｔ＋１個のノードがシークレットを再構成するために参加する必要がある。任意のｔ個の部分が分かると、シークレットが明らかにされてよい。

【００８３】

＜シャミアのシークレット共有＞

【００８４】

閾値シークレット共有ソリューションの一例は、文献「How to share a secret」、Shamir, A.（１９７９）、Communications of the ACM, ２２(１１), ６１２－６１３（“Shamir method”)に記載されている。シャミア（Shamir）法は、多項式補間に基づいてよく、一般性を失うことなく、シークレットは有限フィールドの要素であると想定される。シャミア法は、ディーラ又はディーラ不要の方法を含んでよく、ｎ個のノードＵ$_1$、Ｕ$_2$、．．．、Ｕ$_n$のセット及びアクセス構造Ａを含んでよい。参加者のグループは、シークレットを再構成可能であってよい。シャミア法により、任意のランダムシークレットは、ｔ次多項式ｆ（ｘ）の中のｆ（０）として格納され、参加者ｉのみが自身のシェアｆ（ｘ$_i$）を計算できる。ｎ人のうちのｔ＋１人の参加者が協力する場合、彼らは、ラグランジュ多項式補間を用いて、（鍵ｋの）彼らのシェアｋ$_1$、ｋ$_2$、．．．、ｋ$_n$によりｆ（ｘ）上の任意の点を再構成できる。ラグランジュ多項式補間を用いると、次数ｔを有する関数ｆ（ｘ）は、ｔ＋１個の点により再構成できる。

【００８５】

【数１】

$$p = \left\{\left(x_1, f\left(x_1\right)\right), \left(x_2, f\left(x_2\right)\right), \ldots, \left(x_{t+1}, f\left(x_{t+1}\right)\right)\right\}$$

【００８６】

【数２】

$$f(x) = \sum_{i\in p} f(x_i) \prod_{j\in p, j\neq i} \frac{x-x_j}{x_i-x_j} = \sum_{i\in p} f(x_i) b_{i,p}(x)$$

【００８７】

ここで、

【００８８】

【数３】

$$b_{i,p}(x) = \prod_{j\in p, j\neq i} \frac{x-x_j}{x_i-x_j}$$

【００８９】
【数４】

以下に留意する：$b_{i,p}(x_i) = 1$、及び、$b_{i,p}(x_j) = 0$

【００９０】
ディーラノードが存在するとき、ディーラノードは、サイズｐ（ｐは素数（prime））の有限フィールドＦの要素であると想定されるシークレット$a_0 = k$を選択してよく、ランダムにｔ－１個の正の整数$a_1, \ldots, a_{t-1}$を選び取ってよい。これらは、多項式$f(x) = a_1 + a_1 x + a_2 x^2 + \ldots$の係数を表す。ディーラノードは、次に、多項式に属するｎ個の点$(x_i, f(x_i))$を計算し、それらを参加者に分配してよい。
【００９１】
シャミアのディーラ不要シェア分配段階では、
【００９２】
１）各参加者U_iは、全員に知られているx_iを割り当てられる。各x_iはユニークでなければならない。
【００９３】
２）各プレイヤU_iは、次数ｔを有するランダム多項式$f_i(x)$を生成する。
【００９４】
３）各プレイヤU_iは、全ての他のプレイヤに、多項式上の彼らそれぞれの点$f_i(x_i) \mod n$を秘密裏に（参加者の公開鍵により暗号化される）送信する。
【００９５】
４）各プレイヤU_iは、全ての彼らの受信した$f_1(x_i)$、$f_2(x_i)$、．．．、$f_p(x_i)$（all mod n、ここでｎは楕円曲線上の点Ｇにより生成されるグループの順序である）を加算して、$k_i = f(x_i) \mod n$を形成し、これは多項式$f(x) \mod n$上のシェアである。
【００９６】
シャミア法の上述の説明は、閾値シークレット共有ソリューションの一例であるが、他の方法又は敷地シークレット共有ソリューションが考えられてよい。
【００９７】
幾つかの実装では、敷地署名計算は、ｋ×Ｇの決定に基づき、ここでｋはシークレット鍵であり、Ｇは楕円曲線上の点である。
【００９８】
ｆ（ｘ）がｔ次の多項式である場合、シークレットｋは、$k = \Sigma_{i \in \pi} b_{i,\pi} k_i$により補間できる。ここで、πは、シェア$k_a$、$k_b$、．．．、$k_t$、$k_{t+1}$のサイズｔ＋１の部分集合であってよく、ｂは補間係数であってよい。πは、自身のシェアk_iを開示することなく、ｋ×Ｇを計算するために協力する、ｔ＋１個のノードのグループであってよい。ｋは、ｔ次多項式の点ｘ＝０であってよい。
【００９９】
－各ノードU_iは、部分$b_{i,\pi} k_i \times G$を計算してよい。
【０１００】
－Πの中の全部のノードは、それらの部分を一緒に加算して（ラグランジュ補間によりシークレットｋを再構成し）、次式を与える。
【０１０１】
【数５】

$$b_{a,\pi}k_a \times G + b_{b,\pi}k_b \times G + \cdots + b_{t+1,\pi}k_{t+1} \times G = k \times G$$

【０１０２】
上述のＱ＝ｋＧを計算する処理は、シークレット共有結合（Secret Share Joining）

と呼ばれてよい。

【０１０３】

＜公に検証可能なシークレット共有＞

【０１０４】

本願の主題において参加するノードは、それらのシークレットシェアの正しさを検証することができることが望ましい。鍵シェア検証のために、公に検証可能なシークレット共有（Publicly Verifiable Secret Sharing (PVSS)）プロトコル又はアルゴリズムが利用されてよい。鍵シェア検証は、ノードが、彼ら自身のシェア及び他のノードのシェアを検証することを可能にし得る。

【０１０５】

例示的なＰＶＳＳプロトコルは、「Publicly verifiable secret sharing」、International Conference on the Theory and Applications of Cryptographic Techniques (Springer Berlin Heidelberg), pages １９０－１９９, Stadler, M.,published May １９９６ [Stadler]に記載されている。

【０１０６】

例示的なＰＶＳＳプロトコルでは、各ノードU_iは、解読関数D_iを有してよい。これは、対応する公開暗号関数E_iにより暗号化された情報にアクセスできる。このように、ディーラノードは、公開暗号関数を使用して、鍵シェアを分配し、及びそれらを以下の形式で発行してよい。

【０１０７】

$K_i = E_i(k_i)$、$i = 1, \ldots, n$

【０１０８】

暗号化シェアは、関心のあるノードにより公に検証可能である。ノードは、自身の鍵シェアを検証でき、他のノードが正しい鍵シェアを受信したこと、つまりディーラノードが正直か否かを検証できる。

【０１０９】

ＰＶＳＳプロトコルの主な（上位）コンポーネントは以下を含む。

【０１１０】

（ｉ）シークレット共有：ディーラノードは、アルゴリズム$Share(k) = (k_1, \ldots, k_n)$を走らせ、シェアを計算し、それらを参加ノードの間に分配する。

【０１１１】

（ｉｉ）再構成：参加ノードは、アルゴリズムＲｅｃｏｖｅｒを走らせることにより、シークレットを再構成できる。従って、$Recover(\{D_i(K_i) \mid i \in A\}) = k$。

【０１１２】

（ｉｉｉ）検証：アルゴリズムＰｕｂＶｅｒｉｆｙが、暗号化シェアを検証するために使用されてよい。ノードがアルゴリズム$PubVerify(\{K_i \mid i \in A\}) = 1 \rightarrow Recover(\{D_i(K_i) \mid i \in A\}) = u$、$u = k$、を作動する場合、ディーラノードは、正直であると決定されてよく、鍵シェアは一貫している。

【０１１３】

幾つかの実装では、ＰＶＳＳ方式は、復元段階における必要性に依存して、対話的又は非対話的であり得る。

【０１１４】

幾つかの暗号システムに基づくＰＶＳＳプロトコルの種々の実装が考えられてよい。説明のために、以下は、文献「A simple publicly verifiable secret sharing scheme and its application to electronic voting」、Annual International Cryptology Conference (Springer Berlin Heidelberg), pages １４８－１６４, by Schoenmakers, B., published August １９９９に記載のプロトコルを強調する。

【０１１５】

初期化段階では、公開手順を用いて、グループG_q及び２つの独立に選択された生成元

（generator）Ｇ及びｇが選択されてよい。各ノードは、それ自体の秘密鍵$x_i \in Z_q^*$を与えられ、$y_i = G^{x_i}$を公開鍵として設定してよい。

【０１１６】

ディーラノードは、次に、Z_qにおける係数で、最大でｔ－１の次数を有するランダム多項式$f(x) = \Sigma_{j=0}^{t-1} a_j x^j$を選択してよく、$a_0 = k$に設定する。コミットメント$C_j = g^{a_j}$、$j = 0, \ldots, t$、及び暗号化シェア$f(i)$（参加者の公開鍵を用いる）$Y_i = y_i^{f(x_i)}$、$i = 1, \ldots, n$が発行されてよい。

【０１１７】

$X_i = \Pi_{j=0}^{t-1} (C_j)^{i^j}$を計算することにより、ディーラノードは、暗号化シェアが一貫していることを示すことができる。特に、ディーラノードは、$X_i = g^{f(x_i)}$、$Y_i = y_i^{f(x_i)}$を示すことにより、$f(i)$の知識の証明を生成できる。

【０１１８】

鍵シェアの検証は、上述のＳｔａｄｌｅｒのフィアット－シャミア（Fiat－Shamir）暗号技術を用いて実行できる。非対話型プロトコルの主要なステップは以下を含む。

【０１１９】

証明者（全てのノード）は、ランダム（random）$w_i \in Z_q$を選び出し、$a_{1i} = g^{w_i}$、$a_{2i} = y^{w_i}$を計算してよく、値をブロードキャストしてよい。

【０１２０】

$c = H(X_i, Y_i, a_{1i}, a_{2i})$、Ｈ（）は暗号ハッシュ関数である、を用いて、証明者（例えば、ノード）は、$r_i = w_i - f(x_i)\,c$を計算しブロードキャストしてよい。

【０１２１】

r_i，ｃが与えられると、検証者ノードは、$a_{1i} = g^{r_i} X^{c_i}$、$a_{2i} = y^{r_i} Y^{c_i}$を計算してよく、

【０１２２】

X_i，Y_i，a_{1i}，a_{2i}、$1 \leqq i \leqq n$、のハッシュがｃに一致することを証明してよい。

【０１２３】

必要なとき、参加者ノードは、他のノードのシェア$f(x_i)$に関して何ら知る必要無しに、シークレットｓを再構成できる。必要な情報は、$S_i = G^{f(x_i)}$、$i = 1, \ldots, t$を含んでよい。シークレットは、ラグランジュ補間により計算される。

【０１２４】

【数６】

$$\prod_{i=1}^{t} S_i^{\lambda_i} = \prod_{i=1}^{t} (G^{f(x_i)})^{\lambda_i} = G^{\sum_{i=1}^{t} f(x_i)\lambda_i} = G^{f(0)} = G^k$$

【０１２５】

【数７】

ここで、$b_{i,p}(x) = \prod_{j \in p, j \neq i} \frac{x - x_j}{x_i - x_j}$ はラグランジュ係数である。

【０１２６】

上述の例及び本願の主題の以下の説明は、鍵シェアを計算し及び分配するためにディーラノードを利用する。しかしながら、シークレット共有プロトコルのディーラ不要の実装は、フィアット－シャミアヒューリスティック（Fiat－Shamir heuristics）技術により記載されるように実装されてよい（例えば、「Fiat－Shamir heuristic」、https://en.wikipedia.org/wiki/Fiat%E２%８０%９３Shamir_heuristic, accessed June １３, ２０１８を参照する）。

【０１２７】

ここで、図３を参照すると、本願の例示的な動作環境が示される。図示のように、複数のノード３００が、コンピュータネットワーク３１０を介して通信する。ノード３００の各々は、コンピューティング装置であり、ブロックチェーンに参加しており、ブロックチェーンの関連付けられた、例えば計算リソースの単位のような単位の量を反映する１つ以上の関連アドレスを有する。

【０１２８】

ノード３００の様々なノードは、分散型処理の中の処理ステップを実行してよく、そのアウトプットは結合されて結果を形成する。

【０１２９】

本願の主題は、様々な環境において適用され得る。例えば、図４は、システム、より具体的には航空機情報システム４００のコンテキストにおける本願の特定の例示的な動作環境を示す。

【０１３０】

航空機情報システム４００による分散型処理は制御コンピュータ装置４１０により制御され、複数のサブシステムに依存してよい。例えば、分散型処理は、燃料サブシステム４２０、ナビゲーションサブシステム４３０、除氷サブシステム４４０、及び着陸ギアサブシステム４５０のような、種々のノードを含んでよい。これらのサブシステムの各々は、また、サブコンポーネントを含んでよい。例えば、燃料サブシステム４２０は、油検査サブコンポーネント４２２、温度サブコンポーネント４２４、油圧サブコンポーネント４２６、及び燃料容積サブコンポーネント４２８を含んでよい。サブシステムは、飛行前検査の部分として、特定の順序で検査されてよく、また、所与のサブシステムのサブコンポーネントは特定の順序で検査されてよい。例えば、サブシステムは、燃料サブシステム４２０から開始して、着陸ギアサブシステム４５０まで、図中の矢印により示される順序で検査されてよい。別の例では、燃料サブシステム４２０のサブコンポーネントは、油検査サブコンポーネント４２２から開始して燃料容積サブコンポーネント４２８まで時計回りに検査されてよい。更に後述するように、本願の主題は、順序付き方法でこのような処理を制御するために利用されてよい。更に、例えば除氷サブシステム４４０が熱帯気候ではスキップされてよいというように、環境が正当だと理由づける場合には、サブシステムはスキップされてよい。追加又は代替として、サブシステムが故障した場合でも、他のサブシステムはそれらの処理を順番に完了するよう制御されてよい。

【０１３１】

図３に戻り、ノード３００の各々はコンピューティング装置であることに留意する。図５は、例示的なコンピューティング装置の高レベル動作図である。例示的なコンピューティング装置５００は、例えばノード３００の１つ以上を含む本願明細書に記載されるコンピュータシステムの１つ以上の例であってよい。例示的なコンピューティング装置５００は、自身を特定の機能を実行するよう適応するソフトウェアを含む。

【０１３２】

例示的なコンピューティング装置５００は、種々のモジュールを含む。例えば、図示のように、例示的なコンピューティング装置５００は、プロセッサ５１０、メモリ５２０、及びネットワークインタフェース５３０を含んでよい。図示のように、例示的なコンピューティング装置５００の前述のコンポーネントは、バス５４０を介して通信する。

【０１３３】

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つ以上のＡＲＭ、Ｉｎｔｅｌ　ｘ８６、ＰｏｗｅＰＣプロセッサ、等であってよい。

【０１３４】

メモリ５２０は、データを格納し読み出し可能にする。メモリ５２０は、例えば、ランダムアクセスメモリ、読み出し専用メモリ、永久記憶を含んでよい。永久記憶は、例えば、フラッシュメモリ、固体ドライブ、等であってよい。読み出し専用メモリ及び永久記憶は、非一時的コンピュータ可読記憶媒体である。コンピュータ可読媒体は、例示的なコンピューティング装置５００の全体動作を制御するオペレーティングシステムにより管理さ

れ得るようなファイルシステムを用いて構成されてよい。

【０１３５】

ネットワークインタフェース５３０は、例示的なコンピューティング装置５００を、他のコンピューティング装置及び／又は種々の通信ネットワーク、例えばコンピュータネットワーク３１０（図３）と通信可能にする。

【０１３６】

命令を含むソフトウェアは、コンピュータ可読媒体から、プロセッサ５１０により実行される。例えば、ソフトウェアは、メモリ５２０の永久記憶から、ランダムアクセスメモリへとロードされてよい。追加又は代替として、命令は、メモリ５２０の読み出し専用メモリから直接、プロセッサ５１０により実行されてよい。

【０１３７】

更に後述するように、ソフトウェアは、例えば、ノード３００の１つ以上及び／又は制御コンピューティング装置４１０のうちの１つ以上、及び／又はサブシステム４２０～４５０及び／又はそのサブコンポーネントを含む本願明細書で言及する種々のコンピュータシステムのうちの１つ以上のとして機能するよう、例示的なコンピューティング装置５００のインスタンスを適応してよい。

【０１３８】

図６を参照すると、更に後述するように、本願明細書に記載するプロトコルで、全体の分散型コンプーティングタスクの処理ステップを実行することを担うノードは、タスクの中の処理ステップの順序に対応する方式で順序付けられた、循環順序付きセット６００を形成してよい。循環順序付きセットは、ノード６１０の各々がノードのうちの２つの他のノードを該ノードの直前ノード及び直後のノードとして有する、リングと考えられてよい。例えば、有向リングは、図６に示されるように、ノード６１０の循環順序付きセットにより形成されてよい。

【０１３９】

更に後述するように、本願の主題では、１つより多くのこのような循環順序付きノードセットが利用される。

【０１４０】

例えば、循環順序付きノードセットのうちのノードは、（ＭＤＤプロトコルにおけるものと類似する）所与のサブシステムにより実行される処理ステップに対応してよい。ここで、複数のこのような循環順序付きセットが利用され、各セットは特定のサブシステムに対応する。更に、サブシステム間の関係、特にサブシステムに跨がる処理の順序は、別の循環順序付きノードセット、各ノードが所与のサブシステムに対応する循環順序付きセットのうちの特定の１つから導出される特に別の循環順序付きノードセットを用いて表現されてもよい。

【０１４１】

背景として、同様の循環順序付きセット、特に「回路（circuit）」又は「グループ交換回路（group exchange circuits）」を形成する支払いチャネルのセットによりリンクされるセットは、上述の共同所有の英国特許出願番号第１７０６０７１.６号、第１８０２０６３.６号、及び第１８０６４４８.５号で議論されており、上述のＭＤＤ、ＧＲＥＣＳ、及びＧＲＥＬＲプロトコルにより特徴付けられる。

【０１４２】

別の例では、循環順序付きノードセットのうちのノードは、処理セットを制御するのに加えて又は代替として、他の目的のために、例えばＧＲＥＣＳ又はＧＲＥＬＲプロトコルにおけるものと同様の目的で、利用されてよく、複数のこのようなセットは、各ノードがそれらのセットのうちの１つから導出される別の循環順序付きノードセットにリンクされる。

【０１４３】

任意のイベントで、循環順序付きノードセットのこのような２次元構造は、図７に示すような全体リング７１０に加わったリング７００のセットとして視覚化されてよい。図７

の簡易概略表現では、個々のノードは、リング７００の各々を形成し、全体リング７１０は示されない。しかしながら、矢印は、リング７００に対応する循環順序付きノードセットの各々の中の、及び全体リング７１０に対応する循環順序付きノードセットの中の、順序の可能な方向を示す。

【０１４４】

このような２次元構造のより詳細な概略表現は図８に示される。

【０１４５】

図示のように、２次元構造は、循環順序付きセットω個のC_a、C_b、．．．C_i．．．C_ωを含む。これらは例えばリング７００に対応してよい。

【０１４６】

循環順序付きセットの各々は、図示のように、n_j個のノード：イニシエータノードI_jと、ｎ－１個の他のノード$U_1{}^j$、$U_2{}^j$、．．．$U_i{}^j$、．．．$U^j{}_{nx-1}$を含んでよい。n_jは、種々の循環順序付きセットC_jについて異なり又は同じである。各イニシエータノードI_jは、自身のそれぞれの循環順序付きセットの管理者として機能する。

【０１４７】

別の循環順序付きセット、つまりイニシエータノード８００の循環順序付きセットは、循環順序付きセットC_a、C_b、．．．C_i．．．C_ωの各々のイニシエータノードI_a、I_b、．．．I_i．．．I_ω、及びマスタイニシエータノードI_Mで構成される。更に後述するように、マスタイニシエータノードI_Mは、プロトコル全体の管理者として機能する。

【０１４８】

更に後述するように、本願の主題は、非対称暗号技術に依存する。種々の公開鍵暗号システムが利用されてよい。例えば、楕円曲線暗号法（elliptic curve cryptography (ECC)）が利用されてよい。特定の例では、楕円曲線暗号法が利用されてよく、種々の秘密鍵及び対応する公開鍵が利用され、それにより、所与の秘密鍵ｋ及びその対応する公開鍵Ｐは、楕円曲線暗号公開－秘密鍵ペアであってよい。その結果、Ｐ＝ｋＧであり、Ｇは次数ｑの楕円曲線上の基点（base point）であり、ｑ×Ｇ＝０、ここでｑは大きな素数であり、０＜ｋ＜ｑである。特定の例では、文献「Standards for Efficient Cryptography 2 (SEC2)」、Certicom Corp, January 27, 2010で定められる楕円曲線ｓｅｃｐ２５６ｋ１が利用されてよい。言い換えると、各公開鍵及びその対応する秘密鍵は、楕円曲線暗号公開－秘密鍵ペアを形成してよい。

【０１４９】

循環順序付きノードセットの各々のノードをリンクする支払いチャネルの構造が以下に説明される。

【０１５０】

例えば、循環順序付きセットC_a、C_b、．．．C_i．．．C_ωの所与の１つを検討する。単に例として、以下の議論はC_jを参照するが、これは回路C_a、C_b、．．．C_i．．．C_ωのいずれにも等しく適用可能である。例示的な循環順序付きセットC_jは、図９の概略図９００に示される。

【０１５１】

概略図９００を参照すると、ノードC_j、つまり$U_1{}^j$、$U_2{}^j$、．．．$U^j{}_{nx-1}$、及びI_jは、順序付きセット$\{I_j, U_1{}^j, U_2{}^j, \ldots U^j{}_{nx-1}\}$を形成する。より具体的には、図示のように、ノードは、ノード間の矢印により示されるように、円又はリングを形成するよう順序付けられる。

【０１５２】

図示のように、鍵シェア$k_\varepsilon{}^i$、$k_1{}^j$、$k_2{}^j$、．．．．$k^j{}_{nj-2}$、$k^j{}_{nj-1}$は、それぞれノードI_j，$U_1{}^j$，$U_2{}^j$，．．．$U^j{}_{nx-1}$の各々に関連付けられる。特に、更に後述するように、イニシエータI_jは、前述の鍵シェアを生成し、次に、それらを順序付きセットの他のノードに分配する。概略として、イニシエータI_jは、秘密鍵v_iを生成してよく、次にシークレット共有プロトコルを用いて、ｎ個の鍵シェア$k_\varepsilon{}^i$、$k_1{}^j$、$k_2{}^j$、．．．．$k^j{}_{nj-2}$、$k^j{}_{nj-1}$を生成し、次に、シェア$k_1{}^j$、$k_2{}^j$、

．．．．k^{j}_{nj-2}、k^{j}_{nj-1}を（何らかの秘密及びセキュアなチャネルを介して）他のノード、つまりU_{1}^{j}，U_{2}^{j}，．．．U^{j}_{nx-1}へそれぞれ分配してよく、一方で鍵シェアの残りの１つ、つまり$k_{\varepsilon}{}^{i}$を自身のために保持する。幾つかの実装では、公に検証可能な暗号シークレット共有（publicly erifiable cryptographic secret sharing（PVSS））アルゴリズムが使用されてよい。

【０１５３】

特に、楕円曲線暗号法（ＥＣＣ）が利用される場合、秘密鍵v_{i}は、$0<v_{j}<q$となるよう選択された乱数であってよい。同様に、シェアは、それらがＥＣＣ秘密鍵としての使用にも適するように生成されてよく、$0<k_{i}{}^{j}<q$を含む。

【０１５４】

更に後述するように、鍵シェアは、循環順序付きセットC_{j}のノードの間の支払いチャネルの回路を確立するときに使用される。特に、支払いチャネルは、ノードのうちの隣接ノード間に確立され、これらのチャネルは、鍵シェアに様々に基づく値を用いてロック及びアンロックされる。支払いチャネルは、矢印により結合されたノードのペアの間で生成されてよい。更に後述するように、これらの支払いチャネルは、図９の矢印により示されるのと反対方向に、リソースの移転を提供することを意図してよい。

【０１５５】

図１０は、図８のイニシエータノード８００の循環順序付きセットを示す図である。上述のように、イニシエータノード８００の循環順序付きセットは、全体リング７１０（図７）に対応する第２の次元を提供する。

【０１５６】

図示のように及び上述のように、イニシエータノード８００の循環順序付きセットのうちのイニシエータノードは、イニシエータノード間の矢印により示されるように、円又はリングを形成するよう順序付けられる。

【０１５７】

上述のようｊに、イニシエータノードI_{a}，I_{b}，．．．I_{i}．．．I_{ω}の各々は、関連する秘密鍵v_{j}を有する。更に、図示のように、マスタイニシエータI_{M}は、関連する秘密鍵v_{x}を有する。更に後述するように、関連する秘密鍵は、イニシエータノード８００の循環順序付きセットのノードの間の支払いチャネルの回路を確立するときに使用される。特に、支払いチャネルは、ノードのうちの隣接ノード間に確立され、これらのチャネルは、イニシエータノードに関連付けられた秘密鍵に様々に基づく値を用いてロック及びアンロックされる。支払いチャネルは、矢印により結合されたノードのペアの間で生成されてよい。更に後述するように、これらの支払いチャネルは、図１０の矢印により示されるのと反対方向に、リソースの移転を提供することを意図してよい。

【０１５８】

種々の支払いチャネルの生成が以下に説明される。

【０１５９】

事前知識として、支払いチャネルの使用は、コンピューティングリソースの単位がチャネルにコミットされることを要求することに留意する。例えば、支払いチャネルの所与の回路では、コンピューティングリソースは、各支払いチャネルにより、ノードからその支払いチャネルに直ちに移転されてよい。

【０１６０】

ノード間の値の移転は、本願明細書に記載のプロトコルの何らかの適用の要件でなくてよいので、この量は額面価額（nominal amount.）であり得る。しかしながら、明らかなように、各ノードは、それが支払いチャネルにコミットする値の喪失を条件として、結果を選択するときに参加することを要求される。従って、ノードは、支払いチャネルへの十分に多くのコンピューティングリソースの貢献を要求され、本願明細書に記載のプロトコルの軽率な参加及び／又は違反を思いとどまらせてもよい。支払いチャネルの各々にコミットされるコンピューティングリソースの量は、同じ量であってよく又は異なってよい。支払いチャネルの各々にコミットされたリソースは、リソースが代替可能な場合には同

一であってよい。代替として、これらの値は変化してよい。例えば、幾つかの異なる回路のうちの所与の１つにおいて、同じ量のコンピューティングリソースが、支払いチャネルの各々にコミットされてよいが、これらの値は、該異なる回路の間で同一ではない。

【０１６１】

更に、説明の明確化のために、循環順序付きイニシエータノードセットのうちのイニシエータノードの間の回路を形成する支払いチャネルの生成、及び循環順序付きセットC_a，C_b，．．．C_i．．．C_ωの各々のノードの間の回路を形成する支払いチャネルの生成の議論のために、以下の表記が採用される。

【０１６２】

先ず、文字ｉは、循環順序付きイニシエータノードセットの要素（例えば、イニシエータノード、及び／又はそれらそれぞれの循環順序付きセットの各々）に関連するインデックスとして使用される。

【０１６３】

第２に、文字ｊは、循環順序付きセットC_a，C_b，．．．C_i．．．C_ωのうちの所与の１つC_iの内部の要素（例えば、参加者）に関連するインデックスとして使用される。

【０１６４】

次に、下付き及び上付き文字を用いて記載される循環順序付きセットC_iの要素について、上付きインデックスはｉの値（セット識別子）を表し、下付きインデックスはセットの内部のｊの値（循環順序付きセットの内部の順序のコンポーネント識別子）を表す。一例として、$k_4{}^2$は、循環順序付きセットC_2のうちのインデックス４の秘密鍵である。

【０１６５】

次に、n_iは、回路C_iの中の参加者の番号を示す。特に、n_iは、イニシエータ$I_i = U_0{}^i$を含む。

【０１６６】

更に、所与の循環順序付きセットC_iのインデックスｊへの任意の参照は、モジュロn_iと考えられ、$U_j{}^i$が$U^i{}_{j \bmod n_i}$として扱われるべきであることを意味する。例えば、$n_i = 4$ならば、$U_4{}^i$が$U^i{}_{4 \bmod 4} = U_0{}^i$である。

【０１６７】

更に、以上で使用された$k_\varepsilon{}^i$の表記は、循環順序付きセットC_iのイニシエータの鍵シェアを表し、$k_0{}^i$とも表されてよい。これは、イニシエータI_iに関連付けられた（v_iの）鍵シェアである。

【０１６８】

次に、（循環順序付きイニシエータノードセット以外に）ω個の循環順序付きセットがあり、マスタイニシエータI_M（I_0と呼ばれてもよい）の存在により、ω＋１個のノードが循環順序付きイニシエータノードセットの中にあることに、留意する。

【０１６９】

更に、循環順序付きイニシエータノードセットのインデックスｉへの任意の参照は、モジュロ（ω＋１）と考えられ、例えば$I_i = I_{i \bmod (\omega+1)}$を意味する。従って、ω＝４ならば、$I_{4+1} = I_{5 \bmod 5} = I_0 = I_M$である。

【０１７０】

次に、循環順序付きイニシエータノードセットへの任意の参照は、例えば図８及び１０のように添え字（ａ，ｂ，ｃ，ｄ，．．．）を使用し得ること、或いは代替として、数値の対応するインデックス（０，１，２，３，．．．）を等価的に使用し得ることに留意する。例えば、後者は、循環順序付きイニシエータノードセットに渡る繰り返しを表すときに使用され得る。

【０１７１】

最後に、以下の説明では、直前及び直後という用語が使用され、図８、９、１０に示される矢印の方向に対応することに留意する。例えば、図８でI_aとラベル付けされたノードは、ノードI_bの直前のノードと考えられてよく、I_bとラベル付けされたノードは、ノードI_aの直後のノードと考えられてよい。

【０１７２】
本願明細書に記載されるプロトコルに参加するノードにより実行される例示的な動作は、図１１、１２、１４に示された一連のフローチャートを参照して以下に議論される。
【０１７３】
先ず、イニシエータノードの間の回路を形成する支払いチャネルの構成は、図１１を参照して議論される。図１１は、支払いチャネルの回路を生成するための準備において動作を実行する循環順序付きイニシエータノードセットのイニシエータノードにより実行される動作を示すフローチャート１１００を示す。動作１１１０以降は、種々のイニシエータノードの各々の１つ以上のプロセッサにより実行される。このように、動作１１１０以降は、例えば、メモリ５２０の記憶のようなコンピュータ可読記憶媒体に格納され得るようなコンピュータ実行可能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装置５００の適切に構成されたインスタンスの例えばプロセッサ５１０（図５）のような、種々のコンピューティング装置の１つ以上のプロセッサにより実行される。
【０１７４】
動作１１１０で、マスタイニシエータノードI_Mは、上述の秘密鍵v_xを生成する。
【０１７５】
特に、この値は、プロトコルのセキュリティ、特に、マスタイニシエータノードI_Mにのみ知られているので、循環順序付きイニシエータノードセットのセキュリティの核である。値v_xは、少なくとも循環順序付きセットの各々の間の回路を形成する支払いチャネルの全部が形成されるまで、秘密として維持される。
【０１７６】
更に、動作１１１０で、マスタイニシエータノードI_Mは、秘密鍵v_xに対応する公開鍵P_x*を生成する。例えば、楕円鍵暗号法が利用される場合、マスタイニシエータノードI_Mは、$P_x* = v_x G$を計算してよい。
【０１７７】
次に、動作１１２０で、他のイニシエータノード（I_a～I_ω）の各々は、それらのそれぞれの秘密鍵v_iを生成する。更に、各々が、対応する公開鍵P_i*を生成し、その値をマスタイニシエータノードI_Mへ送信する。例えば、楕円鍵暗号法が利用される場合、イニシエータノードの各々は、自身のそれぞれのv_iについて$P_i* = v_i G$を計算してよい。
【０１７８】
次に、動作１１３０で、マスタイニシエータノードI_Mは、受信したP_i*値を自身の関連付けられたP_x*と結合して、値Q_A*を生成する。例えば、ＥＣＣが利用される場合、それは、値を加算してよい：$Q_A* = P_a* + P_b* + \ldots P_{\omega-1}* + P_\omega* + P_x*$。
【０１７９】
後述するように、支払いチャネルは、I_aのために、I_MとI_aとの間で生成される。つまり、$I_M \rightarrow I_a$であり、$Tpay$トランザクションが値Q_A*により「ロック」され、これは、対応する秘密鍵S_{VA}からQ_A*（該鍵を用いる署名ではない）が、$Tpay$をアンロックするために供給されなければならないことを意味する。特に、Q_A*の定義により、$Tpayy$トランザクションをアンロックするアンロック値S_{VA}は、全部のイニシエータノードに関連付けられた秘密鍵に基づいてよい。例えば、ＥＣＣが利用される場合、Q_A*に対応するシークレット値S_{VA}は、以下のようにQ_A*に関連付けられる。先ず、楕円曲線加算の準同形特性に従い、Ｅ（ｍ＋ｎ）＝Ｅ（ｍ）＋Ｅ（ｎ）を呼び出す。ここでＥは関数Ｅ（ｘ）＝ｘＧである。従って、Q_A*に対応する秘密鍵S_{VA}は、秘密鍵の$v_a + v_b + \ldots + v_{\omega-1} + v_\omega + v_x$である。
【０１８０】
支払いチャネルを確立するために、繰り返しが必要である。繰り返しは、ｉ＝０、従って、マスタイニシエータノード（$I_M = I_0$を思い出す）で開始する。
【０１８１】
次に、動作１１４０で、支払いチャネルの設定を完了するのに十分な時間が残されてい

るか否かが決定される。支払いチャネルの設定を完了するのに十分な時間が残されているか否かを決定する方法は、以下に更に説明される。十分な時間が残されてい無いと決定された場合、設定は終了する（確立された任意の支払いチャネルの返金トランザクションが、タイムアウトにより生じる）。

【０１８２】

十分な時間が残されている場合、動作１１４０に続き、支払いチャネルの全部がイニシエータノードの間で生成されたか否かが決定される。生成された場合、設定は完了したとして終了する。代替として、循環順序付きイニシエータノードセットのうちのノードの間で回路を形成するために生成される必要のある支払いチャネルが依然として存在する場合、次の動作は動作１１５０である。

【０１８３】

動作１１５０で、ノードI_iは、ロック値Q_{i+1}^*を用いてリソースをロックする、I_iとI_{i+1}との間の支払いチャネルを生成する。例えば、ｉ＝０の場合、上述の支払いチャネル$I_M \rightarrow I_a$が生成され、Q_A^*よりロックされる。

【０１８４】

動作１１５０から、支払いチャネルの生成に成功した場合、概念的なイニシエータｉは、インクリメントされ、次に動作１１６０である。代替として、失敗した場合、次の動作は動作１１４０であり、支払いチャネルの生成は十分な時間が残っている限り再試行される。

【０１８５】

動作１１６０で、ノードI_iは、最後の支払いチャネルをロックするために使用されたロック値、及び自身の秘密鍵SV_iに対応する公開鍵P_i^*に基づき、次のロック値を決定する。例えば、ＥＣＣが利用される場合、ノードI_iは、$Q_{i+1}^* = Q_i^* - P_i^*$を計算してよい（ここで、$Q_i^*$はイニシエータｉのインクリメントにより前のロック値を表す）。

【０１８６】

動作１１６０から、次の動作は動作１１４０であり、十分な時間が残っている（生成に成功できる）限り、別の支払いチャネルが（必要に応じて）生成される。

【０１８７】

特に、上述の処理では、マスタイニシエータノードI_M以外の各イニシエータノードは、自身がリソースを提供する支払いチャネルが生成される前に、有利な支払いチャネルの生成により利益を得る。便利なことに、この方法では、イニシエータノードは別のイニシエータノードがリソースを横領するリスクがない、（マスタイニシエータノードを信頼する必要がない）。特に、I_Mは、全てのロック値の基づくv_xの処理において単独のノードであるので、自身の利益のために、このような順序が提供されることを要求しない。

【０１８８】

先ず、所与の循環順序付きノードセットC_iのノードの間の回路を形成する支払いチャネルの構成は、図１２を参照して議論される。図１２は、セットの中の隣接ノードの各ペアの間の支払いチャネルの回路を生成するための準備において動作を実行する標準的な循環順序付きイニシエータノードセットC_iのノードにより実行される動作を示すフローチャート１２００を示す。動作１２１０以降は、種々のイニシエータノードの各々の１つ以上のプロセッサにより実行される。このように、動作１２１０以降は、例えば、メモリ５２０の記憶のようなコンピュータ可読記憶媒体に格納され得るようなコンピュータ実行可能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装置５００の適切に構成されたインスタンスの例えばプロセッサ５１０（図５）のような、種々のコンピューティング装置の１つ以上のプロセッサにより実行される。

【０１８９】

先ず、動作１２１０で、循環順序付きノードセットC_iのイニシエータノードI_iは、上述の方法で、秘密鍵v_iを生成し、その鍵シェアk_ε^i、k_1^i、k_2^i、．．．、$k_{n_i-2}^i$、$k_{n_i-1}^i$を分配する。

【０１９０】
次に、動作１２２０で、イニシエータノードＩ$_i$は、ロック値Ｑ$_{i+1}$*を取得する。Ｑ$_{i+1}$*は、循環順序付きイニシエータノードセットの中の次のイニシエータノードから取得されてよい。例えば、値Ｑ$_{i+1}$*は、次の循環順序付きノードセットＣ$_{i+1}$のイニシエータノードＩ$_{i+1}$から受信されてよい。
【０１９１】
次に、動作１２３０で、イニシエータノードＩ$_i$は、鍵シェア及びＱ$_{i+1}$*に基づき、ロック値を決定する。特に、それは、秘密鍵として扱われるとき、各鍵シェアに対応する公開鍵を決定する。例えば、ｋ$_\varepsilon^i$の場合に、ＥＣＣが利用される場合、ＩｉはＰ$_\varepsilon^i$＝ｋ$_\varepsilon^i$Ｇを計算してよい。特に、循環順序付きノードセットＣ$_i$の他のノードの各々も、同様の方法で、自身の鍵シェアに対応する公開鍵を決定できる。
【０１９２】
イニシエータノードＩ$_i$は、次に、鍵シェアに対応する公開鍵に基づき及びＱ$_{i+1}$*に基づき、ロック値Ｑ$_1^i$を決定する。例えば、ＥＣＣが利用される場合、それは、値を加算してよい：Ｑ$_1^i$＝Ｐ$_1^i$＋Ｐ$_2^i$＋．．．＋Ｐ$_{ni-2}^i$＋Ｐ$_{ni-1}^i$＋Ｐ$_\varepsilon^i$＋Ｑ$_{i+1}$*。
【０１９３】
後述するように、支払いチャネルは、Ｕ$_1^i$のために、Ｉ$_j$とＵ$_1^i$との間で生成される。つまり、Ｕ$_0^i$→Ｕ$_1^i$であり、ＴＰａｙトランザクションが値Ｑ$_1^i$により「ロック」され、これは、対応する秘密鍵ｓｖ$_1$からＱ$_1^i$に（該鍵を用いる署名ではない）が、ＴＰａｙをアンロックするために供給されなければならないことを意味する。特に、Ｑ$_1^i$の定義により、ＴＰａｙｙトランザクションをアンロックするアンロック値ｓｖ$_1$は、Ｃ$_i$のノードの各々に関連付けられた鍵シェアに基づいてよい（及びＱ$_{i+1}$*にも基づく）。例えば、ＥＣＣが利用される場合、Ｑ$_1^i$に対応するシークレット値ｓｖ$_1^i$は、上述の準同形特性によりＱ$_1^i$に関連付けられる。このように、Ｑ$_1^i$に対応する秘密鍵ｓｖ$_1^i$は、前述の鍵シェア及びＳＶ$_{i+1}$の和であり、つまりＱ$_{i+1}$*、つまりｋ$_1^i$＋ｋ$_2^i$＋．．．＋ｋ$_{ni-2}^i$＋ｋ$_{ni-1}^i$＋ｋ$_\varepsilon^i$＋ＳＶ$_{i+1}$である。
【０１９４】
支払いチャネルを確立するために、繰り返しが必要である。繰り返しは、ｊ＝０、従って、イニシエータノード（Ｉ$_i$＝Ｕ$_j^i$を思い出す）で開始する。
【０１９５】
次に、動作１２４０で、支払いチャネルの設定を完了するのに十分な時間が残されているか否かが決定される。支払いチャネルの設定を完了するのに十分な時間が残されているか否かを決定する方法は、以下に更に説明される。
【０１９６】
十分な時間が残されてい無いと決定された場合、設定は終了する（確立された任意の支払いチャネルの返金トランザクションが、タイムアウトにより生じる）。
【０１９７】
代替として、十分な時間が残されている場合、動作１２４０に続き、支払いチャネルの全部がＣ$_i$のノードの間で生成されたか否かが決定される。生成された場合、該循環順序付きセットについて支払いチャネルの設定が完了したとして終了する。代替として、該循環順序付きセットのうちのノードの間で回路を形成するために生成される必要のある支払いチャネルが依然として存在する場合、次の動作は動作１２５０である。
【０１９８】
動作１２５０で、ノードＵ$_j^i$は、ロック値Ｑ$_{j+1}^i$を用いてリソースをロックする、Ｕ$_i$とＵ$_{j+1}^i$との間の支払いチャネルを生成する。例えば、ｊ＝０の場合、上述の支払いチャネルＩ$_j$→Ｕ$_1^i$が生成され、Ｑ$_1^j$よりロックされる。
【０１９９】
動作１２５０から、支払いチャネルの生成に成功した場合、概念的なイニシエータｊは、インクリメントされ、次に動作１２６０である。代替として、失敗した場合、次の動作

は動作１２４０であり、支払いチャネルの生成は十分な時間が残っている限り再試行される。

【０２００】

動作１２６０で、ノードU_j^iは、最後の支払いチャネルをロックするために使用されたロック値、及び自身の鍵シェアk_j^iに対応する公開鍵P_j^iに基づき、次のロック値を決定する。例えば、ＥＣＣが利用される場合、ノードU_j^iは、$Q_{j+1}^i = Q_j^i - P_j^i$を計算してよい（ここで、$Q_j^i$はイニシエータｊのインクリメントにより前のロック値を表す）。

【０２０１】

動作１２６０から、次の動作は動作１２４０であり、十分な時間が残っている（生成に成功できる）限り、別の支払いチャネルが（必要に応じて）生成される。

【０２０２】

図１３は、上記に従い確立されたノード間の支払いチャネルに関連付けられ得るアンロック値を示すための図８の注釈されたバージョンを提供する。

【０２０３】

最後に、フローチャート１１００及びフローチャート１２００に示された動作は、種々の動作を生成し利用される共通の値により必ずしも相互依存する必要がないことに留意する。図１４は、フローチャート１１００及びフローチャート１２００の中の動作がどのように結合され得るかを示すフローチャート１４００を提供する。特に、破線で囲まれて示されたフローチャート１４００の中の動作は、並列に進行してよい。更に、影付きで示された動作は、フローチャート１２００の種々の動作に対応することに留意する。

【０２０４】

＜コミットメントチャネル＞

【０２０５】

更に後述するように、本願の主題は、１回より多くの支払いの繰り返しのための支払いチャネルの使用を必要としない。実際に、チャネルの主な焦点は支払い自体ではない。むしろ、支払いチャネルは、本願明細書に記載されるプロトコルに従い実施されるエスクロー（escrow）メカニズムとして機能する。従って、上述の既存の支払いチャネルと異なり得る支払いチャネルのバージョンを利用することが可能である。この異なるバージョンの支払いチャネルでは、返金支払いトランザクションの複数の繰り返しを提供するのではなく、リソースのエスクローと時間制限された返金トランザクションに重点が置かれ得るからである。支払いチャネルのこのような変形は、コミットメントチャネルと呼ばれてよい。

【０２０６】

コミットメントチャネルは、上述の支払いチャネルの例示的な実装と同様に、リソースの移転が一方向のみである点で、「一方向」チャネルである。ノードのペアでは、U_AがU_Bを支払うコミットメントチャネルは、表記$U_A \rightarrow U_B$により表されてよい。

【０２０７】

図１５は、$U_A \rightarrow U_B$コミットメントチャネルの表現１５００を提供する。

【０２０８】

図示のように、コミットメントチャネルは、３つのトランザクションT_C、T_{Pay}、及びT_rを含む。表記σ（U_x）は、U_xの暗号署名を表す。

【０２０９】

T_Cトランザクションは、支払いチャネルのコミットメントコンポーネントを表す。ここで、トランザクションを通じて、U_bは、２－ｏｆ－２マルチシグネチャ（U_j^i及びU_{j+1}^i）、又はシークレット値sv_{j+1}^iとU_{j+1}^iの暗号署名との知識のいずれかにより制御されるべき、指定された数の単位を送信／コミットする。

【０２１０】

T_rトランザクションは、指定された時間が経過した後にブロックチェーンへの提出の資格を得た、指定された数の単位の（コミットメントトランザクションから）U_bへの返金を表す。従って、T_rは、時間満了の返却条件が充足されると、リソースの制御を、特

定のノードU_bに返すよう構成される。このトランザクションの実行に成功するためには、U_A及びU_Bの暗号署名が必要である。

【０２１１】

T_{Pay}トランザクションは、U_bのコミットされた資金からのU_aへの（つまり、U_bに関連付けられたソースアドレスから、U_aに関連付けられた宛先アドレスへの）、指定された数の単位の移転である。このトランザクションの実行に成功するためには、シークレット値sv_{j+1}^{i}およびU_Bの暗号署名が必要である。言い換えると、トランザクションは、アンロック値を含む必要な値の供給を含む実行条件の充足に応答してのみ実行する。

【０２１２】

コミットメントチャネルの実装は、Ｂｉｔｃｏｉｎブロックチェーンを用いて提供され得る。このような実装では、T_C、T_r、及びT_{Pay}の各々は、Ｂｉｔｃｏｉｎトランザクションであってよい。対応するＢｉｔｃｏｉｎトランザクション１６００、１７００、及び１８００は、それぞれ図１６、１７、１８に示される。特に、トランザクション１６００、１７００、及び１８００のうちの１つ以上は、非標準オペコードＯＰ＿ＥＣＰＭＵＬＴを利用してよい。ＯＰ＿ＥＣＰＭＵＬＴは、符号化楕円曲線点及び数値を取り込み、楕円曲線乗算をスカラー（Scalar）により実行して、符号化楕円曲線点として結果を出力する。ＯＰ＿ＥＣＰＭＵＬＴオペコードの詳細は、共同所有の英国特許出願番号第１７０５７４９.８号、「Computer－Implemented System and Method」、２０１７年４月１０日出願、において提供される。

【０２１３】

図１９は、表１９００を提供し、コミットメントトランザクション１６００の＜ｓｃｒｉｐｔＰｕｂＫｅｙ＞が、コミットされたリソース（Ｂｉｔｃｏｉｎ）をアンロックするために、オペコードＯＰ＿ＥＣＰＭＵＬＴを用いて支払いトランザクション１８００の＜ｓｃｒｉｐｔＳｉｇ＞とどのように結合されるかを示す。

【０２１４】

＜十分な残り時間＞

【０２１５】

支払いチャネルを生成するために十分な時間が残っているか否かを決定する方法は、支払いチャネルに関して以下に議論される。

【０２１６】

支払いチャネルがコミットメントチャネルであるシナリオを検討する。この十分な又は「必要なだけの時間」という考えの中核となるのは、参加者のペアが、彼らが以下を検討することを保証しなければならない、支払いチャネル$U_A \rightarrow U_B$を生成することである。

【０２１７】

１）各コミットメントチャネルは、ｎＴｉｍｅＬｏｃｋｅｄ返金トランザクションT_rを有する。T_rは、参加者U_Aがブロックチェーンに提出してよく、U_Aが特定の条件が満たされたことに満足しない場合に、コミットメントチャネルのエスクロー資金をU_Aに返す。

【０２１８】

２）各参加者はU_A、チャネル$U_A \rightarrow U_B$を構築し及び彼らの資金を回収するために、「公平な」又は「合意された」時間期間を与えられなければならない。

【０２１９】

３）$U_A \rightarrow U_B$自体が生成された後に生成される必要のある他のコミットメントチャネルが存在してよい。これは、以下を含んでよい：

【０２２０】

ａ）同じ循環順序付きセットC_iの回路の中のコミットメントチャネル。

【０２２１】

ｂ）他の循環順序付きセットに関連付けられたの回路の中のコミットメントチャネル。ここで、C_k：$k > i$である。

【０２２２】

ｃ）循環順序付きイニシエータノードセットに関連付けられたの回路の中のコミットメントチャネル$U_e \rightarrow U_f$。ここで、C_k：ｅ＞Ａである。

【０２２３】

４）（$SVx = v_x$を開示する処理の中で）マスタイニシエータI_Mが、第１T_{Pay}トランザクションをブロックチェーンに提出する期待され合意された時間T_{2D}が存在する。

【０２２４】

また留意すべき事に、開始時間T_{2D}の前に何らかが上手くいかなかった場合（全部の必要なチャネルが生成されないことを含む）、参加者U_bは、彼らが以下のための十分な時間を確実に有することを望む。

【０２２５】

ａ）コミットメントチャネル$U_b \rightarrow U_c$からの返金T_rを提出する、又は

【０２２６】

ｂ）U_cがコミットメントチャネル$U_b \rightarrow U_c$からの支払いT_{Pay}を提出していると想定して、コミットメントチャネル$U_a \rightarrow U_b$からの支払いT_{Pay}を提出する。

【０２２７】

開始時間T_{2D}により、これらのシナリオを検討する場合、「十分な時間」が残っているか否かの決定は、循環順序付きセットC_i（又は循環順序付きイニシエータノードセット）の各ノードにとって、彼らのチャネルを生成し返金を受け取るために十分な時間が残されているか否かを考慮しなければならない。

【０２２８】

この目的のために、値s_i及び$S_{i \rightarrow i+1}{}^*$は、以下のように定義されてよい。

【０２２９】

s_iは、各$U_j{}^i$が、以下を行うために与えられる時間量を表す。

【０２３０】

１．コミットメントチャネル$U_j{}^i \rightarrow U_{j+1}{}^i$を構築する。

【０２３１】

２．ブロックチェーン内で見付かる必要なシークレット値を決定する

【０２３２】

３．$U_{j+1}{}^i$を支払うトランザクションT_{Pay}をブロックチェーンネットワークに提出する。

【０２３３】

s_iは、秒、又はブロック数のような単位で表されてよい。

【０２３４】

$S_{i \rightarrow i+1}{}^*$は、所与のイニシエータがコミットメントチャネルを生成するために与えられる時間量を表す。$S_{i \rightarrow i+1}{}^*$値は、以下のために十分なサイズである。

【０２３５】

１．循環順序付きセットC_{i+1}に関連付けられた回路のコミットメントチャネル$U_j{}^{i+1} \rightarrow U_{j+1}{}^{i+1}$の全部が構築できる。

【０２３６】

２．循環順序付きセットC_{i+1}に関連付けられた回路のコミットメントチャネルの全部の必要なシークレット値がブロックチェーン内で見付かる。

【０２３７】

３．回路C_{i+1}のT_{Pay}トランザクションの全部がブロックチェーンに提出され得る。

【０２３８】

４．循環順序付きイニシエータノードセットに関連付けられた回路のコミットメントチャネル$I_i \rightarrow I_{i+1}$が構築できる。

【０２３９】

５．コミットメントチャネル$I_i \rightarrow I_{i+1}$の必要なシークレット値がブロックチェーン内で見付かる。

【０２４０】
６．$I_i \rightarrow I_{i+1}$のＴ$_{Pay}$トランザクションがＢｉｔｃｏｉｎネットワークに提出され得る。
【０２４１】
特に、$S_{i \rightarrow i+1}{}^*$の値の選択は、上述の動作のうちの１つ以上が並列に生じ得ることを考慮に入れてよい。
【０２４２】
$S_{i \rightarrow i+1}{}^*$は、秒、又はブロック数のような単位で表されてよい。
【０２４３】
上述のように定義された値s_i及び$S_{i \rightarrow i+1}{}^*$により、十分な時間が残っているか否かを評価するための条件は、適切な場合、T_{2D}－現在時間＞ｆ（｛s_i：ｉ∈［ａ，ω］｝）、又はT_{2D}－現在時間＞ｆ（｛$S_{i \rightarrow i+1}{}^*$：ｉ∈［ａ，ω］｝）に対応してよい。
【０２４４】
以上は、支払いチャネルの回路が、種々の循環順序付きセットについてどのように確立され得るかを説明した。システムのコンポーネントの（例えば、ω個の循環順序付きノードセットC_a，C_b，．．．，C_i，．．．C_ωの各々に対応するサブシステム間の）、及びこれらのコンポーネントのサブコンポーネント内の（例えば、ω個の循環順序付きセットC_a，C_b，．．．，C_i，．．．C_ωのうちのノードの各々に対応するサブコンポーネント間の）順序付き処理を制御する際のこれらの回路の使用は、図２０を参照して以下に説明される。
【０２４５】
図２０は、種々の循環順序付きノードセットの間の及び各セットを生成するノードの間の処理の順序を制御する動作を実行するときに、種々の循環順序付きセットのノードにより実行される動作を示すフローチャート２０００を示す。動作２０１０以降は、種々のノードの各々の１つ以上のプロセッサにより実行される。このように、動作２０１０以降は、例えば、メモリ５２０の記憶のようなコンピュータ可読記憶媒体に格納され得るようなコンピュータ実行可能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装置５００の適切に構成されたインスタンスの例えばプロセッサ５１０（図５）のような、種々のコンピューティング装置の１つ以上のプロセッサにより実行される。
【０２４６】
図２０に示される動作は、循環順序付きノードセット（循環順序付きイニシエータノードセットを含む）の各々のノードの間の回路を形成するために、支払いチャネルが確立されていると推定する。
【０２４７】
図２０を検討するとき、図８又は図１３を参照して、これらの図に示された例では、循環順序付きイニシエータノードセット以外に４個の循環順序付きセットが存在すること、つまりω＝ｄ＝４であること、及びこれらの４個の循環順序付きセットの各々が正確に４個のノードを含むこと、つまり$n_i = 4 \; \forall$ ｉ∈［１，ｄ］であること、に留意することは、読者にとって助けとなり得る。図８及び１３の種々の循環順序付きノードセットの表現に従い、これらのノードの間の支払いチャネルの実行、従って結果として生じるこれらのノードによる処理のトリガは、反時計回り方向に進むことに留意する。
【０２４８】
動作２０１０で、マスタイニシエータI_Mは、自身の利益のために（$I_\omega \rightarrow I_M$）、支払いチャネルのトランザクションＴ$_{Pay}$を提出する。特に、支払いチャネルが構築される方法のために、該チャネルのＴ$_{Pay}$は、ロック値$Q_x{}^*$によりロックされ、対応するアンロック値SV_xによりアンロックされ（ロック値は、アンロック値の公開鍵に対応する秘密鍵である）、$Q_x{}^*$及びSV_xの両者はv_xに基づく。（v_xは、この時点までＩ$_M$だけに知られていることを思い出す）。特に、これは、Ｔ$_{Pay}$が提出されるとき、ブロックチェーン上でv_xを開示する効果を有する。
【０２４９】

SV_xの値は、I_ωが自身の利益のためにT_{Pay}トランザクションをアンロックするために必要とされる。幾つかの実装では、I_MはSV_xをI_ωへ送信してよい。追加又は代替として、I_ωはSV_xをブロックチェーンから読み出してよい。

【０２５０】

処理がサブシステムに渡り（従って、該サブシステムに対応する循環順序付きセットのイニシエータによりトリガされると、該循環順序付きセットのノードを通じて）進行し得るために、繰り返しが必要である。繰り返しはｉ＝ωで、従って循環順序付きノードセットC_ωのイニシエータノードI_ωで開始する。

【０２５１】

次に、動作２０２０で、処理を未だ完了していない（循環順序付きイニシエータノードセット以外の）循環順序付きノードセットに関連付けられた回路の支払いチャネルの提出を完了するために十分な時間が残されているか否か、及び循環順序付きイニシエータノードセットに関連付けられた回路の残りの支払いチャネルの提出を完了するために十分な時間が残されているか否か、が決定される。

【０２５２】

十分な時間が残されていないと決定された場合、処理は動作２０３０へ進み、イニシエータノードI_{i-1}は、自身の生成した支払いチャネルの返金トランザクション（つまり、$T_r：I_{i-1}\rightarrow I_i$）をブロックチェーンに提出する。特に、これは、チェーン反応をトリガする効果を有し、それにより、循環順序付きイニシエータノードセットの中の他のイニシエータノードの全部が、彼らの利益のために返金トランザクションを同様に提出する。これは、未だ実行していない循環順序付きセットのノードに、返金トランザクションを彼らの利益のために提出させる効果も有する。

【０２５３】

代替として、動作２０２０で、十分な時間が残されていると決定された場合、処理は動作２０４０へ進む。

【０２５４】

動作２０４０で、イニシエータノードI_iは、自身の属する循環順序付き（非イニシエータ）ノードセットC_iに関連付けられた処理が進行すべきか否かを決定する。このような決定は、例えば、特定の目的及び分散型処理の適用が、全体で又は特定のサブシステムの中でノードにより行われている状況に基づいてよい。例えば、図４に関して議論したように、特定の循環順序付きノードセットにより実行され得るような処理が不要である又は保証されない状況が存在し得る。

【０２５５】

循環順序付きノードセットC_iに関連付けられた処理がスキップされるべきであると決定された場合、処理は動作２０５０に進む。代替として、循環順序付きノードセットC_iに関連付けられた処理が進行すべきであると決定された場合、処理は動作２０７００に進む。

【０２５６】

なぜなら、更に後述するように、所与の循環順序付きノードセットC_iの各ノードによる処理の結果として、トランザクションがブロックチェーンにコミットされるので、ブロックチェーンは、どんな処理が生じたかに関して監査証跡を提供する。更に、ブロックチェーンの特性により、このような監査証跡は、改ざん防止又は耐改ざん性であると考えられる。

【０２５７】

動作２０５０で、イニシエータノードI_iは、ブロックチェーンからsv_{i+1}を読み出す。追加又は代替として、イニシエータノードI_iは、I_{i+1}から値を受信してよい。sv_{i+1}が取得されると、この値はv_iに基づき及びSV_{i+1}の基づくコンポーネントに基づくので、値は、sv_iを決定するために使用できる。つまり、秘密鍵v_jは、循環順序付きイニシエータノードセットのI_iの直後のイニシエータノードからマスタイニシエータノードまでの、循環順序付きイニシエータノードセットの中の各ノードに対応する。

例えば、ＥＣＣが利用される場合、v_i及びsv_{i+1}は加算されてよい。

【０２５８】

動作２０５０から、処理は動作２０６０へ進む。動作２０６０で、I_iは、支払いチャネル$I_{i-1} \rightarrow I_i$のための支払いトランザクションT_{Pay}を、ブロックチェーンに提出する。特に、これは、ブロックチェーン上でsv_iを開示する効果を有し、それにより、シーケンスの中の次のイニシエータノード（つまりI_{i-1}）が自身の処理を進めることを可能にする。

【０２５９】

動作２０６０に続き、概念上のイニシエータｉは、デクリメントされ、イニシエータノードが残っていると想定して（つまり、ｉ＞０）、処理は動作２０２０において継続する。代替として、全部のイニシエータノードが実行する機会を有する場合、処理は終了する。

【０２６０】

I_iは、循環順序付きセットC_iのノードに関連付けられた処理をスキップしないと決定することもできたことを思い出す。このシナリオでは、上述のように、処理は動作２０７０へ進み、循環順序付きセットC_iの処理が開始される。

【０２６１】

動作２０７０で、イニシエータノードは、値sv_{ε}^{i}を識別する。（$sv_{\varepsilon}^{i} = k_{\varepsilon}^{i} + AV_{i+1}$を思い出す）。

【０２６２】

動作２０７０に続き、動作２０８０で、イニシエータノードI_iは、自身の利益のためにT_{Pay}トランザクションをブロックチェーンに提出し、それにより値sv_{ε}^{i}を開示する。特に、更に後述するように、これは、ノード$U_{n_i-1}^{i}$が自身の利益のためにブロックチェーンにT_{Pay}トランザクションを提出するために必要なアンロック値を計算するために必要な値を開示する効果を有する。従って、動作２０８０は、ノード$U_{n_i-1}^{i}$による処理をトリガする効果を有する。

【０２６３】

処理が循環順序付きセットC_iに渡り、つまり対応するサブシステムのサブコンポーネントに渡り進行し得るように、繰り返しが必要である。特に、処理は各々のこのようなノードに到達するので、該ノードに関連付けられた任意の処理を実行するようトリガされる。

【０２６４】

繰り返しはｊ＝n_i－１で、従って循環順序付きノードセットC_iのノード$U_{n_i-1}^{i}$で開始する。

【０２６５】

動作２０８０から、処理は動作２０９０へ進む。動作２０９０で、循環順序付きノードセットC_iに関連付けられた回路の支払いチャネルの提出を完了するために十分な時間が残されているか否かが決定される。

【０２６６】

十分な時間が残されていないと決定された場合、処理は動作２１００へ進み、ノードU_{j-1}^{i}は、自身の生成した支払いチャネルの返金トランザクション（つまり、$T_r : U_{j-1}^{i} \rightarrow U_j^{i}$）をブロックチェーンに提出し、それにより、順に、$C_i$の他のノードが対応する返金トランザクション$T_r : U_k^{i} \rightarrow U_{k+1}^{i}$）を提出するのを開始する。

【０２６７】

代替として、動作２０９０で、十分な時間が残されていると決定された場合、処理は動作２１１０へ進む。

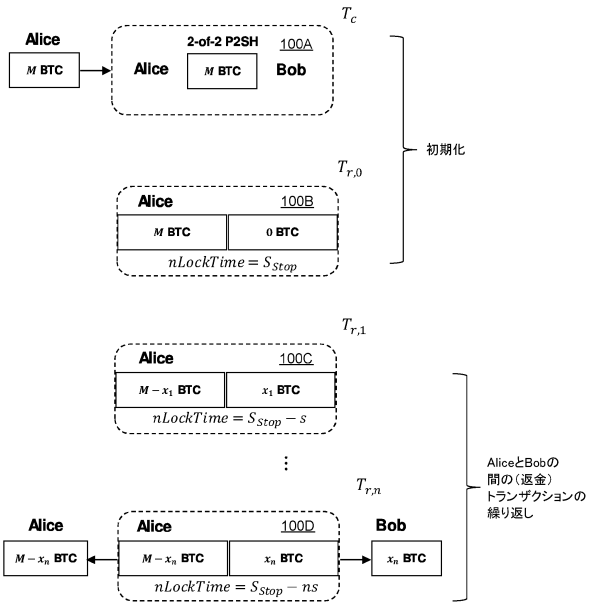
【０２６８】

動作２１１０で、循環順序付きノードセットC_iに関連付けられた回路の中の次のノードU_j^{i}は、シーケンスの中の前のノードがブロックチェーンに提出した支払いトランザクション（つまり、$T_{Pay} : U_j^{i} \rightarrow U_{j+1}^{i}$）から$sv_{j+1}^{i}$を読み出し、該値に基づき及び自身の鍵シェア$k_j^{i}$に基づき$sv_j^{i}$を決定する。例えば、ＥＣＣが利用される場合、それは$sv_j^{i} = k_j^{i} + sv_{j+1}^{i}$を計算してよい。

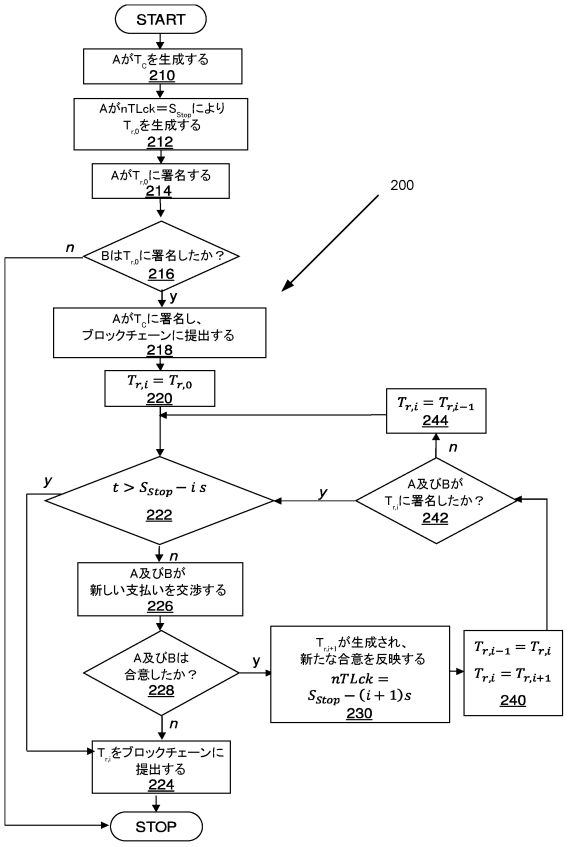
【０２６９】
動作２１１０から、処理は動作２１２０へ進む。動作２１２０で、ノードU_j^iは、自身の利益のために、支払いチャネルの支払いトランザクション（つまり、$T_{Pay}:U_{j-1}^i \rightarrow U_j^i$）をブロックチェーンに提出する。特に、これは、ブロックチェーン上でsv_j^iを公に開示する効果を有する。
【０２７０】
動作２１２０から、処理が循環順序付きノードセットC_iの中の次のノードへと進むとき、概念的なイニシエータｊはデクリメントされる。該セットの中に未だ処理を完了していないノードが残っている場合、処理は動作２０９０に進む。代替として、処理がC_iの中の全部のノードについて完了した場合、制御フローは動作２０５０に戻る。
【０２７１】
動作２０５０で、上述のように、次のイニシエータノード（I_i）は、ブロックチェーンからSV_{i+1}を読み出して、SV_iを計算できるようになり、次に、動作２０６０で、自身の利益のために、SV_iを用いてアンロックされる支払いチャネルの、つまりイニシエータノードI_iの直前のノードにより準備され得るような支払いチャネルの、T_{Pay}トランザクションを提出する。しかしながら、特に、循環順序付きセットC_iのノードの各々に関連付けられたシークレット値はv_iの鍵シェアであり、これらの値は全部、循環順序付きセットC_iのノードの利益のために、支払いトランザクションの提出によりブロックチェーン上で開示されるので、I_iが支払いチャネル$I_{i-1} \rightarrow I_i$のT_{Pay}トランザクションのブロックチェーンへの提出に失敗したとしても、循環順序付きセットC_iのノードによる処理は、依然として、開示された鍵シェアを用いてv_iを再構成することにより、次のイニシエータノードI_{i-1}により開始されてよい。
【０２７２】
上述の実施形態は、事実上、循環順序付きセットのうちのある循環順序付きセットに関連し、前者は循環順序付きセットC_iを参照し、後者は循環順序付きイニシエータノードセットにより定義されるそれらの間の関係を参照する（該セットの各メンバは、マスタイニシエータノードを除き、循環順序付きセットC_iのうちの１つのメンバでもある）。言い換えると、上述の実施形態は、事実上、循環順序付きセットに２次元に関連する。幾つかの実施形態では、上述の実施形態の主題は、更に多くの次元について可能になるよう更に一般化されてよい。例えば、循環順序付きイニシエータノードセットが定義されてよく、それらのノードの各々は、それぞれの更なる循環順序付きイニシエータノードセットのマスタイニシエータノードであり、従って、それらのイニシエータノードの各々は、それぞれの循環順序付きセットのイニシエータであり、それにより３次元を提供する。
【０２７３】
上述の実施形態は、本開示を限定するのではなく、説明すること、及び当業者は添付の特許請求の範囲により定められる本開示の範囲から逸脱することなく多くの代替的実施形態を考案できることに留意すべきである。特許請求の範囲において、括弧内の任意の参照符号は、請求項を限定することを意図しない。用語「有する」及び「含む」（comprising、comprises）等は、任意の請求項又は明細書全体に列挙されたもの以外の要素またはステップの存在を排除しない。本願明細書では、「有する」は「有する又は構成される」を意味し、「含む」は「含む又は構成される」を意味する。要素の単数の参照は、該要素の複数の参照を排除しない。逆も同様である。本開示は、幾つかの別個の要素を含むハードウェアにより、及び適切にプログラムされたコンピュータにより、実装できる。幾つかの手段を列挙する装置クレームでは、これらの手段のうちの幾つかは、１つの同じハードウェアアイテムにより具現化されてよい。単に特定の手段が相互に異なる従属請求項に記載されるという事実は、これらの手段の組み合わせが有利に使用されないことを示さない。

【図面】

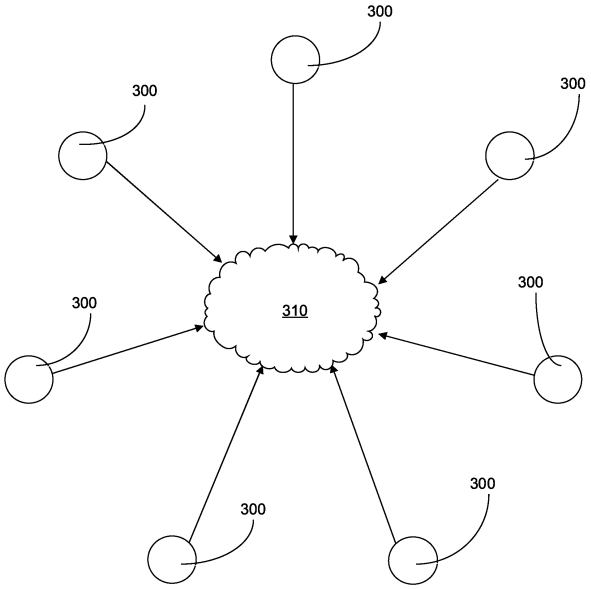
【図１】


T_c
Alice
M BTC
2-of-2 P2SH
100A
Alice
M BTC
Bob
初期化
$T_{r,0}$
Alice
100B
M BTC
0 BTC
$nLockTime = S_{Stop}$
$T_{r,1}$
Alice
100C
$M - x_1$ BTC
x_1 BTC
$nLockTime = S_{Stop} - s$
$T_{r,n}$
Alice
Alice
100D
Bob
$M - x_n$ BTC
$M - x_n$ BTC
x_n BTC
x_n BTC
$nLockTime = S_{Stop} - ns$
AliceとBobの
間の（返金）
トランザクションの
繰り返し

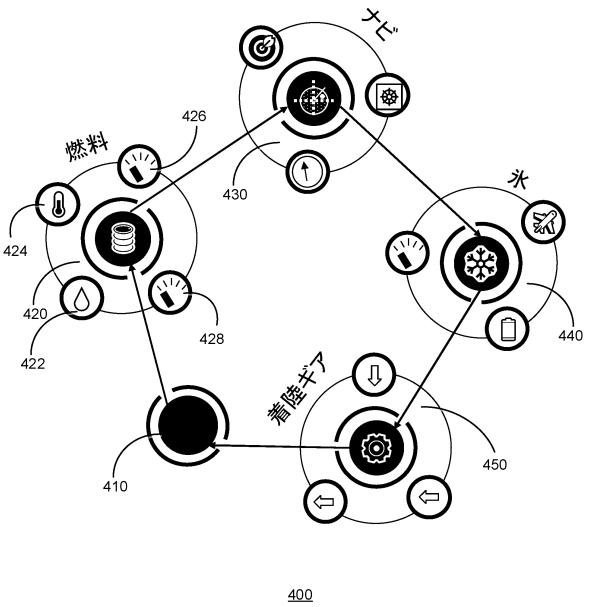
【図２】


START
AがT_cを生成する
210
Aが$nTLck = S_{Stop}$により
$T_{r,0}$を生成する
212
Aが$T_{r,0}$に署名する
214
200
n
BはT$_{r,0}$に署名したか？
216
y
AがT$_c$に署名し、
ブロックチェーンに提出する
218
$T_{r,i} = T_{r,0}$
220
$T_{r,i} = T_{r,i-1}$
244
n
y
$t > S_{Stop} - i\,s$
222
y
A及びBが
T$_{r,i}$に署名したか？
242
n
A及びBが
新しい支払いを交渉する
226
A及びBは
合意したか？
228
y
T$_{r,i+1}$が生成され、
新たな合意を反映する
$nTLck = S_{Stop} - (i+1)s$
230
$T_{r,i-1} = T_{r,i}$
$T_{r,i} = T_{r,i+1}$
240
n
T$_{r,i}$をブロックチェーンに
提出する
224
STOP

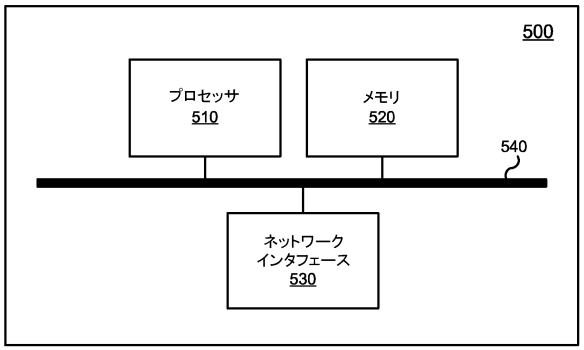
【図３】


300
300
300
300
300
300
300
300
310

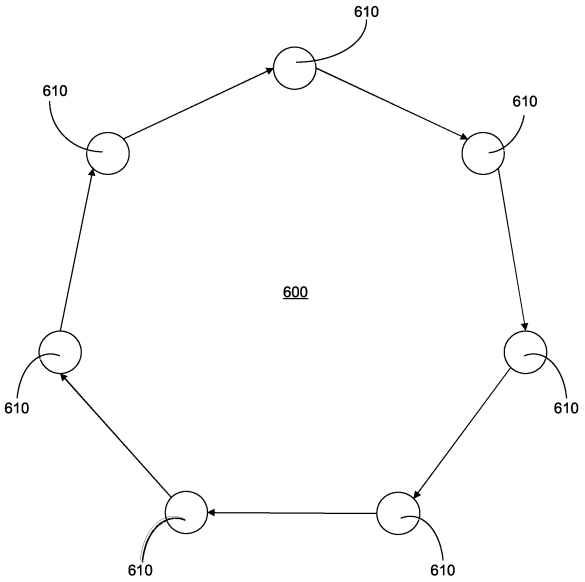
【図４】


ナビ
燃料
氷
着陸ギア
426
430
424
420
422
428
440
410
450
400

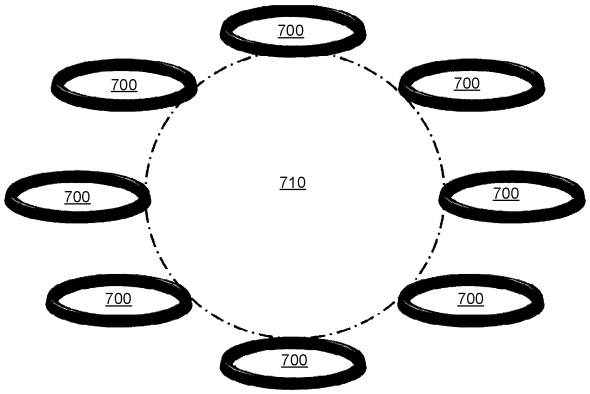
【図５】


500
プロセッサ
510
メモリ
520
540
ネットワーク
インタフェース
530

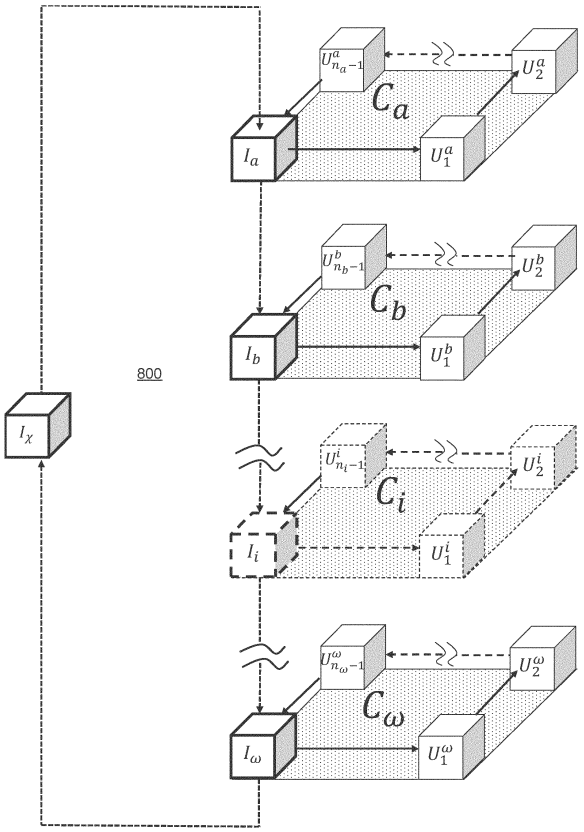
【図６】


610
610
610
600
610
610
610
610

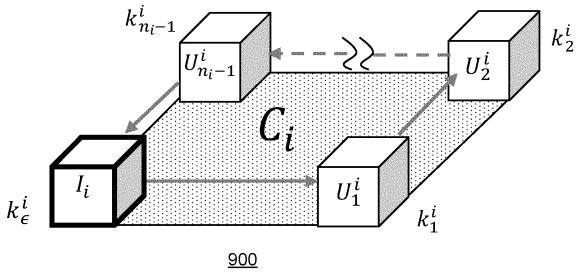
【図７】


700
700
700
700
710
700
700
700
700

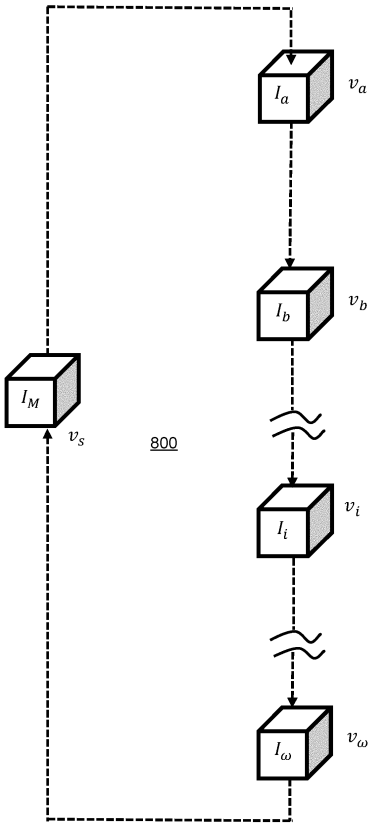
【図８】


800
I_χ
I_a
C_a
$U^a_{n_a-1}$
U^a_2
U^a_1
I_b
C_b
$U^b_{n_b-1}$
U^b_2
U^b_1
I_i
C_i
$U^i_{n_i-1}$
U^i_2
U^i_1
I_ω
C_ω
$U^\omega_{n_\omega-1}$
U^ω_2
U^ω_1

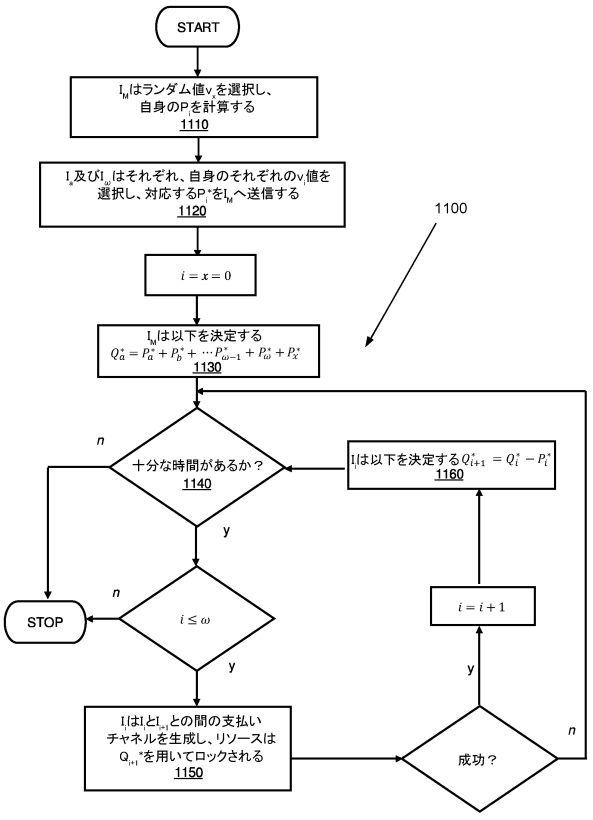
【図９】



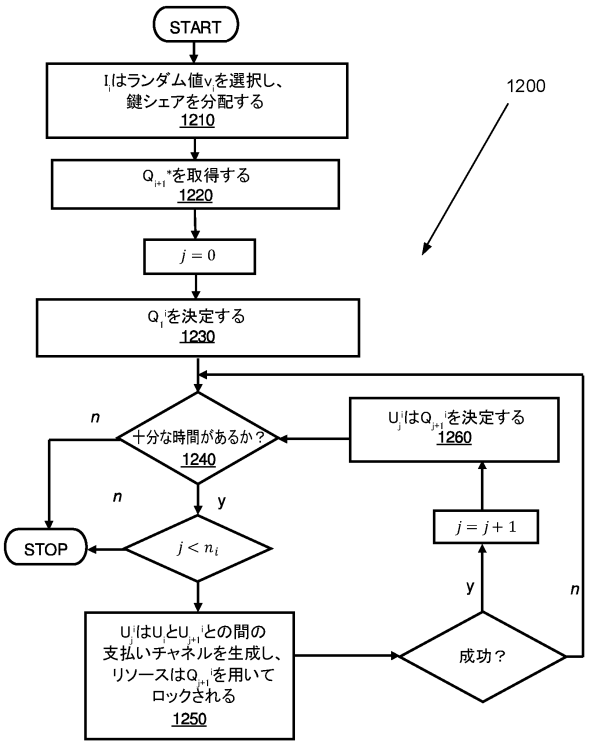
【図１０】



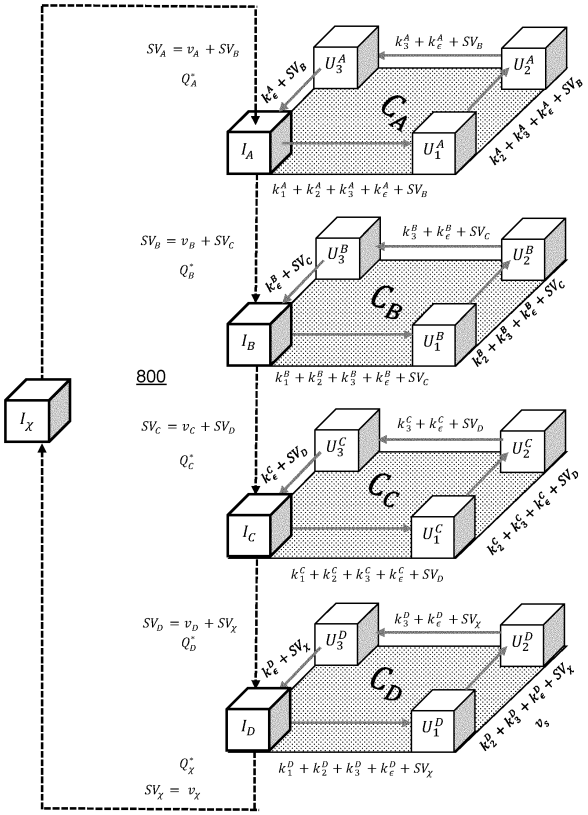
【図１１】



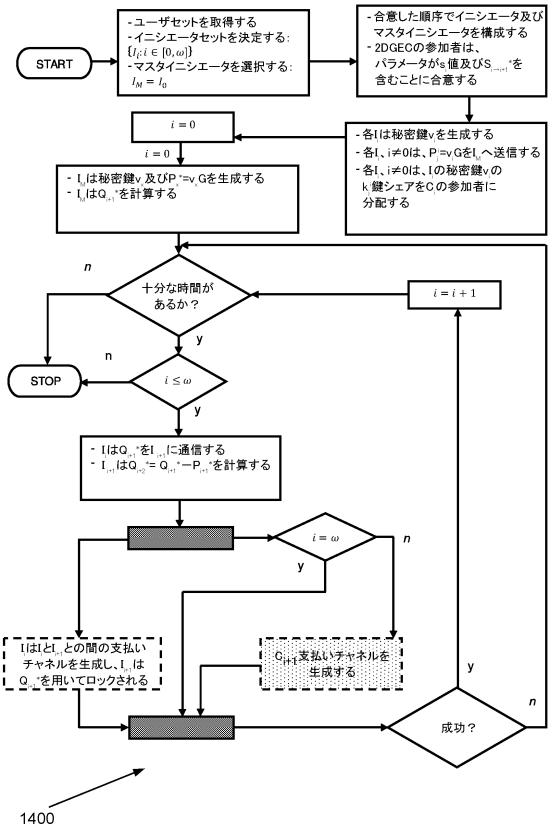
【図１２】



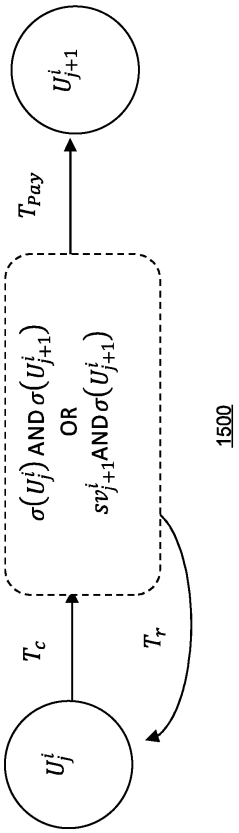
【図１３】



【図１４】



【図１５】



【図１６】

1600	
100	Transaction-ID
Version Number	Version number
<#>	Number of inputs
<Ref>	Prev Trans Output
<#>	Prev Trans Output index
Script length	Script length
<Script>	ScriptSig
<ScriptSignatureFlags>	ScriptSig
Sequence number	Sequence number
<#>	Number of Outputs
xBTC	Output value
Output Script Length	Output script length
redeemScript: OP_IF OP_2 <pubkey U_A> <pubkey U_B> OP_2 OP_CHECKMULTISIG OP_ELSE <*basepoint G*> OP_ECPMULT <Q_B> OP_EQUALVERIFY <*pubkey* U_B> OP_CHECKSIG OP_ENDIF **scriptPupkey:** OP_HASH160 <$H(redeemScript)$> OP_EQUAL	Output script
LockTime	LockTime

【図１７】

1700	
<ID>	Transaction-ID
Version Number	Version number
<#>	Number of inputs
100	Prev Trans ID
0	Prev Trans Output index
Script length	Script length
OP_0 <sig U_A> <sig U_B> OP_TRUE <T_c redeemScript>	ScriptSig
<ScriptSignatureFlags>	ScriptSig
Sequence number	Sequence number
<#>	Number of Outputs
xBTC	Output value
Output Script Length	Output script length
OP_DUP OP_HASH160 $<H(\text{pubkey } U_A)>$ OP_EQUALVERIFY OP_CHECKSIG	Output script
$f(\{s_i : i \in [a, \omega]\})$	LockTime

【図１８】

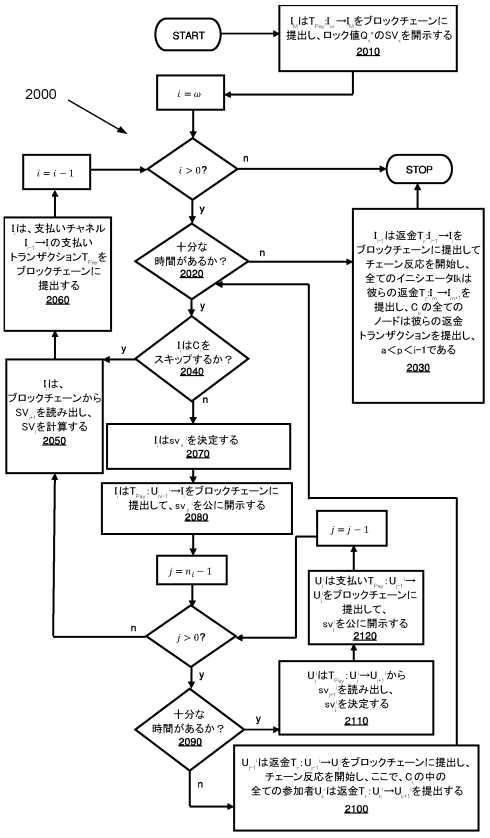
1800	
<ID>	Transaction-ID
Version Number	Version number
<#>	Number of inputs
100	Prev Trans ID
0	Prev Trans Output index
Script length	Script length
<sig U_B> <sv_B> <T_c redeemScript>	ScriptSig
<ScriptSignatureFlags>	ScriptSig
Sequence number	Sequence number
<#>	Number of Outputs
xBTC	Output value
Output Script Length	Output script length
OP_DUP OP_HASH160 $<H(\text{pubkey } U_B)>$ OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

【図１９】

<scriptSig> T_{Pay}	<scriptPubkey> T_c	Combined (r_hash) **STACKED**	Combined (r_pure) **STACKED**
<sig U_B> <sv_B> <T_c redeemScript>	Redeem Pure <*basepoint G*> OP_ECPMULT <Q_B> *OP_EQUALVERIFY* <*pubkey* U_B> OP_CHECKSIG Redeem Hash *OP_HASH160* <$H(redeemScript)$> *OP_EQUAL*	*OP_EQUAL* <$H(redeemScript)$> *OP_HASH160* <T_c redeemScript> <sv_B> <sig U_B>	OP_CHECKSIG <*pubkey* U_B> *OP_EQUALVERIFY* <Q_B> OP_ECPMULT <*basepoint G*> <sv_B> <sig U_B>

1900

【図２０】



フロントページの続き

内
審査官　行田 悦資
(56)参考文献　特表２０１５－５０５２３０（ＪＰ，Ａ）
特表２０１４－５１３３５１（ＪＰ，Ａ）
特開２０００－１５１５７３（ＪＰ，Ａ）
特開２００９－２２５３５６（ＪＰ，Ａ）
米国特許出願公開第２００４／０２６０９２６（ＵＳ，Ａ１）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／３２
Ｇ０６Ｑ　２０／０４