

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7128111号  
(P7128111)

(45)発行日 令和4年8月30日(2022.8.30)

(24)登録日 令和4年8月22日(2022.8.22)

|            |                |         |      |         |  |
|------------|----------------|---------|------|---------|--|
| (51)国際特許分類 |                | F I     |      |         |  |
| H 0 4 L    | 9/32 (2006.01) | H 0 4 L | 9/32 | 2 0 0 Z |  |
| G 0 9 C    | 1/00 (2006.01) | H 0 4 L | 9/32 | 2 0 0 B |  |
|            |                | G 0 9 C | 1/00 | 6 4 0 D |  |

請求項の数 17 (全44頁)

|                   |                               |          |                                                                          |
|-------------------|-------------------------------|----------|--------------------------------------------------------------------------|
| (21)出願番号          | 特願2018-539889(P2018-539889)   | (73)特許権者 | 318001991                                                                |
| (86)(22)出願日       | 平成29年2月14日(2017.2.14)         |          | エヌチェーン ホールディングス リミテッド                                                    |
| (65)公表番号          | 特表2019-511150(P2019-511150 A) |          | NCHAIN HOLDINGS LIMITED                                                  |
| (43)公表日           | 平成31年4月18日(2019.4.18)         |          | アンティグア・バーブーダ、セントジョンズ、44 チャーチ ストリート、フィッツジェラルド ハウス                         |
| (86)国際出願番号        | PCT/IB2017/050824             |          | Fitzgerald House, 44 Church Street, St. John's, Antigua and Barbuda (AG) |
| (87)国際公開番号        | WO2017/145007                 | (74)代理人  | 100107766                                                                |
| (87)国際公開日         | 平成29年8月31日(2017.8.31)         |          | 弁理士 伊東 忠重                                                                |
| 審査請求日             | 令和2年1月15日(2020.1.15)          | (74)代理人  | 100070150                                                                |
| (31)優先権主張番号       | 1603117.1                     |          |                                                                          |
| (32)優先日           | 平成28年2月23日(2016.2.23)         |          |                                                                          |
| (33)優先権主張国・地域又は機関 | 英国(GB)                        |          |                                                                          |
| (31)優先権主張番号       | 1603123.9                     |          |                                                                          |
| (32)優先日           | 平成28年2月23日(2016.2.23)         |          |                                                                          |
|                   | 最終頁に続く                        |          | 最終頁に続く                                                                   |

(54)【発明の名称】 ブロックチェーンを介して資産関連活動を制御するシステム及び方法

## (57)【特許請求の範囲】

## 【請求項1】

ブロックチェーンを介して行われる転送を制御するよう構成されたコンピュータにより実装されるシステムであって、前記システムは、

資産に関連付けられた第1電子装置及び該資産の所有者に関連付けられた第2電子装置であって、前記第1電子装置が第1又は第2エンティティの一方であり、前記第2電子装置が第1又は第2エンティティの他方であり、前記第1エンティティ及び前記第2エンティティは電子装置であり、前記第1電子装置及び前記第2電子装置のそれぞれは、既存暗号鍵ペアのサブキーとして生成された少なくとも1つの秘密鍵を有し、以下：

前記第1エンティティにおいて、第1エンティティマスタ秘密鍵と生成器値とに基づき、第1エンティティ第2秘密鍵を決定し、

前記第2エンティティにおいて、第2エンティティマスタ秘密鍵と前記生成器値とに基づき、第2エンティティ第2秘密鍵を決定し、

前記第1エンティティ第2秘密鍵と第2エンティティ第2公開鍵とに基づき、前記第1エンティティにおいて共通シークレット(CS)を決定し、前記第2エンティティ第2秘密鍵と第1エンティティ第2公開鍵とに基づき、前記第2エンティティにおいて前記共通シークレット(CS)を決定し、

前記第1エンティティ及び前記第2エンティティのそれぞれにおいて、前記共通シークレット(CS)に基づき新たな秘密鍵を生成する、

ことにより複数の秘密暗号鍵を生成し、前記第1エンティティ第2公開鍵及び前記第

10

20

2 エンティティ第2公開鍵は、それぞれ、前記第1／第2エンティティマスタ鍵及び前記生成器値に基づく、資産に関連付けられた第1電子装置及び該資産の所有者に関連付けられた第2電子装置と、

複数の公開暗号鍵を有するレジストリであって、各公開鍵は、

前記資産の複数の秘密鍵のうちの1つの秘密鍵に対応し、

未使用ブロックチェーンランザクションの中で識別可能なRedeemScriptハッシュに関連付けられる、レジストリと、

前記資産に関して転送が行われるための1又は複数のブロックチェーンランザクションの自動生成に関連する少なくとも1つのルールを有するスマートコントラクトと、

前記スマートコントラクトの前記少なくとも1つのルールを評価し及び／又は実行するよう構成された計算エージェントと、

を含むシステム。

【請求項2】

前記転送は、前記資産に関して行われるべき通貨の転送であり、前記転送は、前記資産に関連付けられた収入又は費用に関連する、請求項1に記載のシステム。

【請求項3】

前記通貨はビットコイン、又は任意のデジタル通貨、又はトークン化された値である、請求項2に記載のシステム。

【請求項4】

前記スマートコントラクトの中で提供される前記少なくとも1つのルールは、前記資産に関連付けられる支払スケジュールを定める、請求項1乃至3のいずれか一項に記載のシステム。

【請求項5】

前記計算エージェントは、別の計算エージェントと通信するよう構成され、前記別の計算エージェントの各々は、前記資産に関連付けられたタスクを実行するよう構成される、請求項1乃至4のいずれか一項に記載のシステム。

【請求項6】

前記計算エージェントは、前記別の計算エージェントの各々の活動を調整し及び／又は認可するよう構成される、請求項5に記載のシステム。

【請求項7】

i) 前記資産に関連付けられた収入に関するブロックチェーンランザクションを生成するよう構成される少なくとも1つの別の計算エージェントと、

ii) 前記資産に関連付けられた費用に関するブロックチェーンランザクションを生成するよう構成される少なくとも1つの別の計算エージェントと、

iii) 前記資産に関連付けられた支払に対する収入の相殺決済を処理するよう構成される少なくとも1つの別の計算エージェントと、

iv) 警報を生成し及び／又は前記資産の所有者に送信するよう構成される少なくとも1つの別の計算エージェントであって、前記警報は前記資産に関するイベントの通知を提供する、少なくとも1つの別の計算エージェントと、

を更に含む請求項1乃至6のいずれか一項に記載のシステム。

【請求項8】

計算エージェントは、前記スマートコントラクトの中で指定されたイベント又はトリガを検出すると、ブロックチェーンランザクションを生成するよう構成される、請求項1乃至7のいずれか一項に記載のシステム。

【請求項9】

前記スマートコントラクトは、分散ハッシュテーブルに格納され、前記スマートコントラクトのハッシュは、ブロックチェーンランザクションに関連付けられるメタデータ内で提供される、請求項1乃至8のいずれか一項に記載のシステム。

【請求項10】

ブロックチェーンを介して行われる転送を制御するコンピュータにより実施される方法

10

20

30

40

50

であって、前記方法は、

資産に関連付けられた第1電子装置及び該資産の所有者に関連付けられた第2電子装置を定めるステップであって、前記第1電子装置が第1又は第2エンティティの一方であり、前記第2電子装置が第1又は第2エンティティの他方であり、前記第1エンティティ及び前記第2エンティティは電子装置であり、前記第1電子装置及び前記第2電子装置のそれぞれは、既存暗号鍵ペアのサブキーとして生成された少なくとも1つの秘密鍵を有し、  
以下：

前記第1エンティティにおいて、第1エンティティマスタ秘密鍵と生成器値とに基づき、第1エンティティ第2秘密鍵を決定し、

前記第2エンティティにおいて、第2エンティティマスタ秘密鍵と前記生成器値とに基づき、第2エンティティ第2秘密鍵を決定し、

前記第1エンティティ第2秘密鍵と第2エンティティ第2公開鍵とに基づき、前記第1エンティティにおいて共通シークレット（CS）を決定し、前記第2エンティティ第2秘密鍵と第1エンティティ第2公開鍵とに基づき、前記第2エンティティにおいて前記共通シークレット（CS）を決定し、

前記第1エンティティ及び前記第2エンティティのそれぞれにおいて、前記共通シークレット（CS）に基づき新たな秘密鍵を生成する、

ことにより複数の秘密暗号鍵を生成し、前記第1エンティティ第2公開鍵及び前記第2エンティティ第2公開鍵は、それぞれ、前記第1／第2エンティティマスタ鍵及び前記生成器値に基づく、ステップと、

複数の公開暗号鍵を有するレジストリを生成するステップであって、各公開鍵は、

前記資産の複数の秘密鍵のうちの1つの秘密鍵に対応し、

未使用ブロックチェーントランザクションの中で識別可能なRedeemScriptハッシュに関連付けられる、ステップと、

前記資産に関して転送が行われるための1又は複数のブロックチェーントランザクションの自動生成に関連する少なくとも1つのルールを有するスマートコントラクトを生成するステップと、

前記スマートコントラクトの前記少なくとも1つのルールを評価し及び／又は実行するために、計算エージェントを使用するステップと、

を含む方法。

【請求項11】

前記転送は、前記資産に関して行われるべき通貨の転送であり、前記転送は、前記資産に関連付けられた収入又は費用に関連し、

前記通貨はビットコイン、又はデジタル通貨、又はトークン化された値である、請求項10に記載の方法。

【請求項12】

前記スマートコントラクトの中で提供される前記少なくとも1つのルールは、前記資産に関連付けられる支払スケジュールを定める、請求項10又は11に記載の方法。

【請求項13】

前記計算エージェントを別の計算エージェントと通信するよう構成し、前記別の計算エージェントの各々を、前記資産に関連付けられたタスクを実行するよう構成するステップ、を更に含む請求項10乃至12のいずれか一項に記載の方法。

【請求項14】

前記計算エージェントは、前記別の計算エージェントの各々の活動を調整し及び／又は認可するよう構成される、請求項13に記載の方法。

【請求項15】

i) 前記資産に関連付けられた収入に関するブロックチェーントランザクションを生成するよう構成される少なくとも1つの別の計算エージェントと、

ii) 前記資産に関連付けられた費用に関するブロックチェーントランザクションを生成するよう構成される少なくとも1つの別の計算エージェントと、

10

20

30

40

50

i i i) 前記資産に関連付けられた支払に対する収入の相殺決済を処理するよう構成される少なくとも1つの別の計算エージェントと、

i v) 警報を生成し及び／又は前記資産の所有者に送信するよう構成される少なくとも1つの別の計算エージェントであって、前記警報は前記資産に関するイベントの通知を提供する、少なくとも1つの別の計算エージェントと、

を提供するステップ、を更に含む請求項10乃至14のいずれか一項に記載の方法。

【請求項16】

計算エージェントは、前記スマートコントラクトの中で指定されたイベント又はトリガを検出すると、ブロックチェーンランザクションを生成するよう構成される、請求項10乃至15のいずれか一項に記載の方法。

【請求項17】

前記スマートコントラクトは、分散ハッシュテーブルに格納され、前記スマートコントラクトのハッシュは、ブロックチェーンランザクションに関連付けられるメタデータ内で提供される、請求項10乃至16のいずれか一項に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、概して、コンピュータにより実施される資産の制御、交換、及び処理に関する。特に、本発明は、スマートコントラクトの分野に関し、ブロックチェーンのようなピアツーピア分散ネットワークと共に使用することに適する。これは、ビットコインブロックチェーン又は代替のブロックチェーンプラットフォームであって良い。本発明は、特に、資産の所有権が複数のエンティティの間で分割され、及び／又は該資産から価値が生じ又は生成される状況で有利である。本発明は、このような状況における活動のためのトリガとして機能するイベントを決定し及びそれに応答するメカニズムを提供する。本発明は、電子資産の制御及び転送に関連して向上したセキュリティ及び匿名性を提供する暗号技術も含む。

【背景技術】

【0002】

ブロックチェーンは、ブロックにより構成される、コンピュータにより実装される非集中型の、合意に基づく分散型システムである。また、ブロックはランザクションにより構成される。各ブロックは前のブロックのハッシュを含み、ブロックは共にチェーンになって、その発端からブロックチェーンに書き込まれている全てのランザクションの永久的な変更不可能なレコードを生成する。ランザクションは、そのインプット及びアウトプットに組み込まれるスクリプトとして知られる小さなプログラムを含む。スクリプトは、ランザクションのアウトプットがどのように及び誰によりアクセス可能かを指定する。

【0003】

最も広く知られているブロックチェーン技術の用途はビットコイン台帳であるが、他のブロックチェーンの実装が提案され開発されている。ビットコインは便宜上及び説明を目的として本願明細書において言及されるが、本発明はビットコインのブロックチェーンと共に使用することに限定されず、代替のブロックチェーンの実装が本発明の範囲に含まれることに留意すべきである。

【0004】

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。しかしながら、更に近年は、デジタル起業家が、新しいシステムを実装するために、ビットコインの基づく暗号通貨セキュリティシステムの使用、及びブロックチェーンに格納可能なデータの両者を探索し始めている。これらは、限定ではないが、以下を含む：

- ・メタデータを格納する。情報をランザクション内のメタデータとして埋め込むことにより、該情報がブロックチェーンを介して転送可能である。これは、以下のために使用可能である。

- ・デジタルトークンを実施する。

10

20

30

40

50

・デジタル署名により署名された取引を確立する。スマートコントラクトは、どの動作が実行されるべきか、例えば支払が特定パーティに対して行われるべきか否かを決定するために評価され得る条件及びテストを含み得る、自己実行型の機械可読ソリューションを提供するよう設計できる。

#### 【0005】

自動で、セキュアな且つ効果的な方法で、収入及び／又は費用を計算し及びブロックチェーンに基づく資産の所有者に割り当て可能であることは有利である。これを行うために、このような資産の所有者に又はそれにより支払可能な収入又は費用のような動作を生成するイベント（トリガ）が検出される必要があり得る。次に、収入又は費用支払を実行するためにブロックチェーントランザクションを自動的に（つまり機械により）生成し署名するための技術が必要である。

10

#### 【0006】

このような改良されたソリューションが考案されている。本発明は、ブロックチェーントランザクションの自動生成を可能にするブロックチェーンプロトコルに基づき構築される技術ソリューションを含み得る。トランザクションは、収入又は費用の支払のような、ブロックチェーンを介するエンティティ間の転送を行うために使用されて良い。ブロックチェーンの使用は、所与の取引に関して行われる任意の支払のセキュアな、公の記録が存在することを保証する。これは、争いのあるシナリオで、取引に対する滞納の証拠を提供するために独自システム及びデータソースへのアクセスを得る必要を除去し又は少なくとも軽減する。本発明は、暗号鍵の生成のための新規技術も含み及びそれを利用し得る。本発明から生じる他の利点のうちの幾つかは、以下に議論される。

20

#### 【発明の概要】

#### 【0007】

本発明は、ブロックチェーン上での交換を制御し、計算し、生成し、及び／又は実行するコンピュータにより実装されるシステム又はメカニズムを提供できる。交換（転送）は、任意の種類の物理的、仮想的、又は論理的資産に関連して良い。

#### 【0008】

本発明は、資産に関して、ブロックチェーンを介して行われる転送を制御するよう構成されて良く、資産又は資産の表現を含んで良い。資産又は表現は、1又は複数の秘密暗号鍵を含んで良く、各秘密鍵は資産の所有者に関連付けられる。秘密鍵は、暗号鍵ペアの部分であって良い。秘密鍵のうちの1又は複数のは、「サブキー生成の方法」と題される章で後述される技術に従い生成されて良い。秘密鍵のうちの1又は複数のは、共通シークレットを用いて生成されて良い。それは、マスタ暗号鍵から生成されるサブキーであって良い。

30

#### 【0009】

本発明は、第1及び第2エンティティに共通のシークレットを決定する技術を含み得る。第1又は第2エンティティは資産であって良く、他方のエンティティは制御部又は資産に関連付けられた他のエンティティであって良い。本方法は、以下のステップ：

- ・少なくとも1つの第1エンティティマスタ秘密鍵及び生成器値に基づき、第1エンティティ第2秘密鍵を決定するステップと、
- ・少なくとも1つの第2エンティティマスタ秘密鍵及び生成器値に基づき、第2エンティティ第2秘密鍵を決定するステップと、

40

第1エンティティ第2秘密鍵及び第2エンティティ第2公開鍵に基づき、第1エンティティにおいて共通シークレット（CS）を決定し、第2エンティティ第2秘密鍵及び第1エンティティ第2公開鍵に基づき、第2エンティティにおいて共通シークレット（CS）を決定するステップと、を含み、

- ・第1エンティティ第2公開鍵及び第2エンティティ第2公開鍵は、それぞれ、第1／第2エンティティマスタキー及び生成器値に基づく。

#### 【0010】

本発明は、レジストリを更に含んで良い。レジストリは、1又は複数の公開暗号鍵を格納して良い。各公開鍵は、資産の秘密鍵のうちの1つに対応して良い。（1又は複数の）

50

公開鍵は、未使用ブロックチェーントランザクション（UTXO）の中で識別可能なRedeemScriptハッシュに関連付けられて良い。

【0011】

本発明は、転送が資産に関して行われることを可能にする、1又は複数のブロックチェーントランザクションの自動生成に関する少なくとも1つのルールを含むスマートコントラクトを含んで良い。スマートコントラクトは、機械可読及び／又は実行可能であって良い。

【0012】

本発明は、スマートコントラクトの少なくとも1つのルールを評価し及び／又は実行するよう構成された計算エージェントを含んで良い。計算エージェントは、実質的に後述される通りであって良い。

10

【0013】

転送は、例えば公債又は暗号通貨の一部のような何らかの通貨の転送であって良い。これは、資産に関して行われるべき支払であって良い。これは、資産に関連付けられる収入又は費用に関連して良い。転送は、スマートコントラクトの中で指定されるパーティ間で行われて良い。通貨はビットコイン、又は任意のデジタル（暗号）通貨、又はトークン化された値であって良い。

【0014】

スマートコントラクトの中で提供される少なくとも1つのルールは、資産に関連付けられる支払スケジュールを定めて良い。

20

【0015】

計算エージェントは、別の計算エージェントと通信するよう構成されて良い。前記別の計算エージェントの各々は、資産に関連付けられたタスクを実行するよう構成されて良い。計算エージェントは、別の計算エージェントの各々の活動を調整し及び／又は認可するよう構成されて良い。したがって、「マスタ」計算エージェント及び1又は複数の従属（別の）計算エージェントが存在して良い。マスタエージェントは、1又は複数の従属計算エージェントの活動を制御し、指示し、又は監視するよう構成されて良い。

【0016】

本発明は、資産に関連付けられた収入に関するブロックチェーントランザクションを生成するよう構成される少なくとも1つの別の計算エージェント（後述するような「収入エージェント」）を含んで良い。

30

【0017】

本発明は、資産に関連付けられた費用に関するブロックチェーントランザクションを生成するよう構成される少なくとも1つの別の計算エージェント（後述するような「費用エージェント」）を含んで良い。

【0018】

本発明は、資産に関連付けられた支払に対する収入の相殺決済を処理するよう構成される少なくとも1つの別の計算エージェント（後述するような「相殺決済エージェント」）を含んで良い。

【0019】

本発明は、警告を生成し及び／又は資産の所有者に送信するよう構成される少なくとも1つの別の計算エージェント（後述するような「通知エージェント」）を含んで良い。該警告は、資産に関するイベントについての通知を提供する。

40

【0020】

計算エージェントは、スマートコントラクトの中で指定されたイベント又はトリガを検出すると、ブロックチェーントランザクションを生成するよう構成されて良い。

【0021】

スマートコントラクトは、分散ハッシュテーブルに格納されて良い。取引のハッシュは、ブロックチェーントランザクションに関連付けられたメタデータの中で提供されて良い。

【0022】

50

本発明は、資産に関してブロックチェーンを介して行われる転送を制御する、コンピュータにより実施される方法であって、前記方法は、以下のステップ：

少なくとも1つの秘密暗号鍵を含む資産を定めるステップであって、各秘密鍵は資産の所有者に関連付けられる、ステップと、

少なくとも1つの公開暗号鍵を含むレジストリを生成するステップであって、各公開鍵は、

資産の1又は複数の秘密鍵のうちの1つの秘密鍵に対応し、

未使用ブロックチェーントランザクションの中で識別可能なRedeemScriptハッシュに関連付けられる、ステップと、

転送が資産に関して行われることを可能にする、1又は複数のブロックチェーントランザクションの自動生成に関する少なくとも1つのルールを含むスマートコントラクトを生成するステップと、

スマートコントラクトの少なくとも1つのルールを評価し及び／又は実行するよう構成された計算エージェントを使用するステップと、を含む方法を更に含む。

【0023】

前記転送は、前記資産に関して行われるべき通貨の転送であって良く、前記転送は、前記資産に関連付けられた収入又は費用に関連する。通貨はビットコイン、又は任意のデジタル通貨、又はトークン化された値であって良い。

【0024】

スマートコントラクトの中で提供される少なくとも1つのルールは、資産に関連付けられる支払スケジュールを定めて良い。

【0025】

前記方法は、前記計算エージェントを別の計算エージェントと通信するよう構成し、前記別の計算エージェントの各々を、前記資産に関連付けられたタスクを実行するよう構成するステップ、を更に含んで良い。

【0026】

計算エージェントは、別の計算エージェントの各々の活動を調整し及び／又は認可するよう構成されて良い。

【0027】

前記方法は、

i) 資産に関連付けられた収入に関するブロックチェーントランザクションを生成するよう構成される少なくとも1つの別の計算エージェントと、

ii) 資産に関連付けられた費用に関するブロックチェーントランザクションを生成するよう構成される少なくとも1つの別の計算エージェントと、

iii) 資産に関連付けられた支払に対する収入の相殺決済を処理するよう構成される少なくとも1つの別の計算エージェントと、

iv) 警告を生成し及び／又は資産の所有者に送信するよう構成される少なくとも1つの別の計算エージェントであって、該警告は、資産に関するイベントについての通知を提供する、少なくとも1つの別の計算エージェントと、を提供するステップを更に含んで良い。

【0028】

計算エージェントは、スマートコントラクトの中で指定されたイベント又はトリガを検出すると、ブロックチェーントランザクションを生成するよう構成されて良い。

【0029】

前記スマートコントラクトは、分散ハッシュテーブルに格納されて良く、前記取引のハッシュは、ブロックチェーントランザクションに関連付けられるメタデータ内で提供される。

【0030】

本発明の1又は複数の実施形態は、「サブキー生成の方法」と題される章で後述される技術を含み得る。これは、第1及び第2エンティティに共通のシークレットを決定する技

10

20

30

40

50

術を含み得る。以後、用語「ノード」は、参照を容易にするために、「エンティティ」の代わりに使用されることがある。本方法は、以下のステップ：

- ・少なくとも1つの第1ノードマスタ秘密鍵及び生成器値に基づき、第1ノード第2秘密鍵を決定するステップと、
- ・少なくとも1つの第2ノードマスタ秘密鍵及び生成器値に基づき、第2ノード第2秘密鍵を決定するステップと、

第1ノード第2秘密鍵及び第2ノード第2公開鍵に基づき、第1ノードにおいて共通シークレット（CS）を決定し、第2ノード第2秘密鍵及び第1ノード第2公開鍵に基づき、第2ノードにおいて共通シークレット（CS）を決定するステップと、を含んで良く、  
 ・第1ノード第2公開鍵及び第2ノード第2公開鍵は、それぞれ、第1ノードマスタキー及び生成器値に基づく。

#### 【0031】

追加又は代替で、本発明の1又は複数の実施形態は、以下のようなコンピュータにより実施される方法：

第1ノード（C）において、第1ノード（C）及び第2ノード（S）に共通な共通シークレット（CS）を決定するステップであって、第1ノード（C）は第1ノードマスタ秘密鍵（V<sub>1C</sub>）及び第1ノードマスタ公開鍵（P<sub>1C</sub>）を有する第1非対称暗号ペアに関連付けられ、第2ノードは第2ノードマスタ秘密鍵（V<sub>1S</sub>）及び第2ノードマスタ公開鍵（P<sub>1S</sub>）を有する第2非対称暗号ペアに関連付けられる、ステップ、を含んで良く、前記方法は、

少なくとも1つの第1ノードマスタ秘密鍵（V<sub>1C</sub>）及び決定キー（DK）に基づき、第1ノード第2秘密鍵（V<sub>2C</sub>）を決定するステップと、

少なくとも1つの第2ノードマスタ公開鍵（P<sub>1S</sub>）及び決定キー（DK）に基づき、第2ノード第2公開鍵（P<sub>2S</sub>）を決定するステップと、

第1ノード第2秘密鍵（V<sub>2C</sub>）及び第2ノード第2公開鍵（P<sub>2S</sub>）に基づき、共通シークレット（CS）を決定するステップと、を含み、

第2ノード（S）は、第1ノード第2公開鍵（P<sub>2C</sub>）及び第2ノード第2秘密鍵（V<sub>2S</sub>）に基づく同じ共通シークレット（S）を有し、

第1ノード第2公開鍵（P<sub>2S</sub>）は、少なくとも1つの第1ノードマスタ公開鍵（P<sub>1C</sub>）及び決定キー（DK）に基づき、

第2ノード第2秘密鍵（V<sub>2S</sub>）は、少なくとも1つの第2ノードマスタ秘密鍵（V<sub>1S</sub>）及び決定キー（DK）に基づく。

#### 【0032】

決定キー（DK）は、メッセージ（M）に基づいて良い。前記方法は、以下のステップ：

メッセージ（M）及び第1ノード第2秘密鍵（V<sub>2C</sub>）に基づき、第1署名メッセージ（SM<sub>1</sub>）を生成するステップと、

通信ネットワークを介して、第1署名メッセージ（SM<sub>1</sub>）を第2ノード（S）へ送信するステップと、を含んで良く、

第1署名メッセージ（SM<sub>1</sub>）は、第1ノードを認証するために、第1ノード第2公開鍵（P<sub>2C</sub>）により検証できる。

#### 【0033】

望ましくは、前記方法は、以下のステップ：

通信ネットワークを介して、第2署名メッセージ（SM<sub>2</sub>）を第2ノード（S）から受信するステップと、

第2ノード第2公開鍵（P<sub>2S</sub>）により第2署名メッセージ（SM<sub>2</sub>）を検証するステップと、

第2署名メッセージ（SM<sub>2</sub>）を検証するステップの結果に基づき、第2ノード（S）を認証するステップと、を含んで良く、

第2署名メッセージ（SM<sub>2</sub>）は、メッセージ（M）又は第2メッセージ（M<sub>2</sub>）、及び第2ノード第2秘密鍵（V<sub>2S</sub>）に基づき生成される。

## 【0034】

望ましくは、前記方法は、以下のステップ：

メッセージ（M）を生成するステップと、

通信ネットワークを介して、メッセージ（M）を第2ノード（S）へ送信するステップと、を含んで良い。

## 【0035】

望ましくは、前記方法は、以下のステップ：

通信ネットワークを介して、メッセージ（M）を第2ノード（S）から受信するステップ、を含んで良い。

## 【0036】

望ましくは、前記方法は、以下のステップ：

通信ネットワークを介して、メッセージ（M）を別のノードから受信するステップ、を含んで良い。

## 【0037】

望ましくは、前記方法は、以下のステップ：

メッセージ（M）をデータストア、及び/又は第1ノード（C）に関連付けられた入力インタフェースから受信するステップ、を含んで良い。

## 【0038】

望ましくは、第1ノードマスタ公開鍵（ $P_{1C}$ ）及び第2ノードマスタ公開鍵（ $P_{1S}$ ）は、それぞれの第1ノードマスタ秘密鍵（ $V_{1C}$ ）及び第2ノードマスタ秘密鍵（ $V_{1S}$ ）及び生成器（G）の楕円曲線点乗算に基づいて良い。

## 【0039】

望ましくは、前記方法は、以下のステップ：

通信ネットワークを介して、第2ノードマスタ公開鍵（ $P_{1S}$ ）を受信するステップと、第1ノード（C）に関連付けられたデータストアに、第2ノードマスタ公開鍵（ $P_{1S}$ ）を格納するステップと、を含んで良い。

## 【0040】

望ましくは、前記方法は、以下のステップ：

第1ノード（C）において、第1ノード秘密鍵（ $V_{1C}$ ）及び第1ノードマスタ公開鍵（ $P_{1C}$ ）を生成するステップと、

通信ネットワークを介して、第1ノードマスタ公開鍵（ $P_{1C}$ ）を第2ノード（S）及び/又は他のノードを送信するステップと、

第1ノード（C）に関連付けられた第1データストアに、第1ノードマスタ秘密鍵（ $V_{1C}$ ）を格納するステップと、を含んで良い。

## 【0041】

望ましくは、前記方法は、以下のステップ：

通信ネットワークを介して、第2ノードへ、共通シークレット（CS）を決定する方法のために、共通生成器（G）と共に共通楕円曲線暗号（ECC）システムを使用することを示す通知を送信するステップ、を含んで良く、

第1ノードマスタ秘密鍵（ $V_{1C}$ ）及び第1ノードマスタ公開鍵（ $P_{1C}$ ）を生成するステップは、

共通ECCシステムの中で指定された許容範囲の中のランダム整数に基づき、第1ノードマスタ秘密鍵（ $V_{1C}$ ）を生成するステップと、

次式： $P_{1C} = V_{1C} \times G$ に従い、第1ノードマスタ秘密鍵（ $V_{1C}$ ）及び共通生成器（G）の楕円曲線点乗算に基づき、第1ノードマスタ公開鍵（ $P_{1C}$ ）を決定するステップと、を含む。

## 【0042】

望ましくは、前記方法は、以下のステップ：

メッセージ（M）のハッシュの決定に基づき、決定キー（DK）を決定するステップ、を含んで良く、

10

20

30

40

50

第1ノード第2秘密鍵 ( $V_2C$ ) を決定するステップは、次式:  $V_2C = V_1C + DK$  に従い、第1ノードマスタ秘密鍵 ( $V_1C$ ) 及び決定キー ( $DK$ ) のスカラ和に基づき、第2ノード第2公開鍵 ( $P_2S$ ) を決定するステップは、次式  $P_2S = P_1S + DK \times G$  に従い、決定キー ( $DK$ ) 及び共通生成器 ( $G$ ) の楕円曲線点乗算への第2ノードマスタ公開鍵 ( $P_1S$ ) の楕円曲線点加算に基づく。

【0043】

望ましくは、決定キー ( $DK$ ) は、前の決定キーのハッシュを決定することに基づいて良い。

【0044】

第1非対称暗号ペア及び第2非対称暗号ペアは、それぞれ、前の第1非対称暗号ペア及び前の第2非対称暗号ペアの関数に基づいて良い。

【0045】

ある実施形態又は態様に關連して上述した任意の特徴は、任意の他の実施形態又は態様に關連して使用されても良い。例えば、方法に關連して記載されて任意の特徴は、システムに關連して使用されて良く、逆も同様である。

【0046】

本発明の上述の及び他の態様は、本願明細書に記載される実施形態から明らかであり、それらの実施形態を参照して教示される。

【図面の簡単な説明】

【0047】

本発明の実施形態は、単なる例として添付の図面を参照して以下に説明される。

【図1】本発明の実施形態により実装されたシステムの概観を示す。

【図2】本発明に關連して利用され得る幾つかの説明のための使用例を示す。

【図3】本願で使用される競走馬 (Racehorse) 例について、図5～11のトランザクションで使用されるメタデータのフォーマットを示す。

【図4】Racehorse資産の定義を示す。

【図5】以下で使用される「競走馬」シナリオのシーケンスを開始するトランザクションを示し、該トランザクションでは、stableは以下に記載されるサブキーを生成する方法を用いて競走馬のための秘密鍵を生成し、次にseedが株式発行のための競走馬口座を見付ける。

【図6】図5から続くブロックチェーントランザクションを示し、該トランザクションでは、次に競走馬が自身で株式 (share、持分) を発行する。留意すべきことに、技術的にstableは競走馬の秘密鍵を所有するが、論理的にこれが起こっていることである。

【図7】投資家からのstable料金セットを請求するトランザクションを示す。

【図8】競走馬がstable料金をstableに渡すトランザクションを示す。

【図9】競走馬は自身の初めてのレースに勝つと生成されるトランザクションを示し、stableは£20,400を稼ぐ。標準的なBTCブローカトランザクション (図示せず) を用いて、GBP20,400と引き換えに、ExchangeがstableにBTC等価額を支払う (トランザクションID=EXCHANGE-S3-T1)。EXCHANGE-S3-T1からのアウトプットをインプットとして用いて、stableは、競走馬キーにBTCとして賞金を支払うトランザクションを生成する。

【図10】競走馬が投資家に自身の賞金を支払うトランザクションを示す。

【図11】AliceがEveに1株を売るトランザクションを示す。

【図12】後述するような親キーからサブキーを導出する説明のための技術の種々の態様を示す。

【図13】後述するような親キーからサブキーを導出する説明のための技術の種々の態様を示す。

【図14】後述するような親キーからサブキーを導出する説明のための技術の種々の態様を示す。

【図15】後述するような親キーからサブキーを導出する説明のための技術の種々の態様

10

20

30

40

50

を示す。

【図 1 6】後述するような親キーからサブキーを導出する説明のための技術の種々の態様を示す。

【図 1 7】後述するような親キーからサブキーを導出する説明のための技術の種々の態様を示す。

【図 1 8】後述するような親キーからサブキーを導出する説明のための技術の種々の態様を示す。

【 0 0 4 8 】

本技術は、本発明の態様と関連して使用することに適する。

【発明を実施するための形態】

10

【 0 0 4 9 】

<本願明細書で使用される主な用語>

以下の用語は、鍵の概念及びコンポーネントを定めるために本願明細書で使用される。

20

30

40

50

【表 1】

| 名称                                                 | 種類                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 主体                                                 | 主体は、本発明の一実施形態の中のエンティティ又はコンポーネントである。                                                                                                                                                                                                                                                                                      |
| Agent (エージェント)                                     | ここで、用語「エージェント」は、計算リソースを表すために使用されることがある。計算リソースは、ブロックチェーンと並列に処理機能を提供する、プログラムを実行するサーバのような適切に構成された計算コンポーネントである。ブロックチェーンの状態を監視し、及びブロック外ソースからインプットを受信できる。受信するインプット及び／又はテストの結果に依存して、様々に応答し及び振る舞うことができる。図又は説明の中で、用語「o r a c l e」は、「エージェント」の代わりとして使用されることがある。A g e n t は、実質的に以下に「本発明と共に使用するための説明のための計算エージェント」と題される章に記載される通りであり得る。 |
| 相殺                                                 | 「相殺」又は「相殺決済」は、単一の値を生成するために2以上のポジション又はトランザクションの値が合併整理される（差し引き勘定される）処理を記載するために、金融の文脈で使用される用語である。複数パーティの合意の中でどのパーティが報酬を得るかを決定するために使用され得る。                                                                                                                                                                                   |
| A s s e t (資産)                                     | 資産は、所有権を複数の A s s e t H o l d e r のうちの1人に可変量で配分できるものを表す主体である。これらの量は、単に百分率（例えば5%）又は単位（例えば100単位）であり得る。A s s e t は、秘密暗号鍵（最終所有者又は制御部の鍵から決定論的に導出され得る）と A g e n t（上述した）の両方を含む又は少なくとも関連付けられる。秘密鍵は、実質的に「サブキー生成の方法」と題される章で後述される技術を用いて最終所有者又は制御部の鍵から決定論的に導出されて良い。                                                                  |
| Asset Holder<br>(資産保有者)                            | 資産保有者は、A s s e t のある割合を現在持っている、つまり所有するものを表す主体である。                                                                                                                                                                                                                                                                        |
| レポジトリ                                              | レポジトリは、資産に関連する支払スケジュールを定める C o n t r a c t 文書の記憶（及び潜在的に進行中の維持）を担うエンティティを表す主体である。                                                                                                                                                                                                                                         |
| C o n t r a c t<br>(取引)                            | C o n t r a c t は、どのように支払を決定するかを正式な定義、及びこの資産のための支払に関連付けられたスケジュールを含む文書である。機械可読及び実行可能であり得るスマートコントラクトである。（使用される実装方法に依存して）資産の現在の所有権の詳細も保持して良い。                                                                                                                                                                               |
| O w n e r s h i p<br>R e g i s t r y<br>(所有権レジストリ) | O w n e r s h i p R e g i s t r y は、A s s e t の現在の所有権（及び所有権の履歴）を定める文書又はデータベースである。幾つかの実施形態では、O w n e r s h i p R e g i s t r y は C o n t r a c t 自体の中に含まれ得ることに留意する。                                                                                                                                                       |

1つの観点からは、ここに記載される発明は、ブロックチェーン上の交換を制御する一般的技術メカニズムを提供するとして考えられる。これらの交換は、支払に関連する金融上の交換であって良い。本発明は、この交換を可能にする技術的メカニズム又は箱を対象とする。

## 【0050】

1つの観点によると、本発明は、資産の現在所有権に比例して費用を計算し、登録し、及び／又は分配し、及び／又は収入を生成する技術的構成を提供する。したがって、本発明により提供される1つの利点は、ブロックチェーン上に登録され維持される資産の費用及び収入を表すデータの分配又は転送を可能にすることである。また、これは、資産自体

及び投資団体のような様々なエンティティの現実世界の必要を満たす、このような資産の能力を増大する。

【0051】

ブロックチェーンアプリケーションの最近の研究は、「スマートコントラクト」として知られるものを生成し及び使用するメカニズム、及びこれらがブロックチェーンに対してどのように監視され管理され得るかの探索を含む。1つの可能性は、資産の所有権が多数の他のエンティティの間で分割できる資産を表すために、スマートコントラクトを使用することである。この所有権は、次に、転送の箱 (vehicle) のようなブロックチェーンを用いるエンティティの間でセキュアに転送され得る。

【0052】

幾つかの例では、資産は、生成される収入に権利を授与する。追加又は代替で、資産は、保有者を特定の費用債務に結合できる。本発明は、資産に対して未払いの費用又は資産から生成される収入のような値が、ブロックチェーンを介して交換され得るようにする、一般的メカニズムを提供する。

【0053】

＜本発明の主な要素＞

本発明の幾つかの要素は次の通りである。

- ・ブロックチェーンに基づく資産の所有者に支払可能な収入を生成するイベント (トリガ) を決定する技術的ソリューション及び技術。
- ・ブロックチェーンに基づく資産の所有者により支払可能な費用を生成するイベント (トリガ) を決定する技術的ソリューション及び技術。
- ・所定ルールセットに基づき、ブロックチェーンに基づく資産の収入／費用を計算し所有者に割り当てる技術的ソリューション及び技術。
- ・収入／費用支払を実行するためのビットコインランザクションを生成し署名する技術的ソリューション及び技術。

【0054】

図1に示すように、処理は階層構造のエージェントにより制御される。これらのエージェントは、ブロックチェーンと並列に実行するコンピュータに基づくリソースであって良い。これらは、ブロックチェーンの状態を監視し、テストの結果及び条件に応答するよう適切にプログラムでき、及び／又はブロック外リソースからインプットを受信できる。このようなエージェントがどのように実装できるかの例については、以下の「本発明と共に使用するための説明のための計算エージェント」と題された章を参照する。

【0055】

本願明細書に記載の例では、以下が仮定される。しかしながら、本例は基礎にある方法において可能な変形を限定しないことが理解される。

- ・取引毎に (つまり資産毎に) 1つのマスタエージェント (Master Agent) がある。離散機能を実行する幾つかの従属エージェント (Agent) があって良い。
- ・マスタエージェントは全ての従属エージェントを管理し「認可する」。
- ・本発明の従属エージェントは、以下の種類のエージェントのうちの1又は複数の組み合わせを含む。
  - － 資産から生成される収入を処理することを担う1又は複数の収入エージェント (Income Agent) 。
  - － 資産から生成される費用を処理することを担う1又は複数の費用エージェント (Cost Agent) 。
  - ・任意で、適切な場合には支払に対して収入を相殺することを担う1又は複数の相殺エージェント (Netting Agent) 。
  - － 行われた収入支払、費用未払金、資産に影響する又は関連するイベント、のような種々のイベント、又は任意の他の通知を株 (share、持分) 所有者に警告することを担う1又は複数の通知エージェント (Notification Agent) 。
  - － 必要な又は任意の離散機能を実行する任意の他の種類のエージェント。

## 【0056】

エージェントの動作は、ルールに基づき、任意の所定ルールセットに従って良い。ルールは、実行されるべきタスクの要件に従い決定される。本発明によると、マスタエージェントは全ての他のエージェントを管理する。制御及び処理の階層的及び可能なフローの一例は、以下の通りであって良い。

- ・マスタエージェントは、収入／費用エージェントがそれらの例えば以下のタスクを実行することを認可する。

- － マスタエージェントは全てのトリガを監視する、又は、

- － 従属エージェントは、処理するために、トリガを監視し、マスタエージェントから権限を求める、又は、

- － 従属エージェントは、それらのタスクを監視し及び実行することを予め認可される、又は、

- － 上述の任意の組み合わせ。

- ・収入／費用エージェントからのアウトプット及び制御は、マスタエージェントに渡される。

- ・マスタエージェントは、（収入／費用エージェントからのデータを渡すことを）相殺エージェントに保証する。

- ・相殺エージェントはマスタエージェントにアウトプットを渡す。

- ・マスタエージェントは、トランザクションブロードキャストを認可する（自身がブロードキャストを行う、又は従属エージェントがブロードキャストすることを認可する、又は組み合わせ、等）。

## 【0057】

以下は、本発明の一実施形態の鍵要件の概要である。

## 【0058】

＜現在の所有権を決定する＞

資産に対して費用／収入を配分するために、該資産の現在の所有権を決定可能である必要がある。本発明の一実施形態によると、現在の所有権に関する詳細事項は、レジストリ（公衆又は私設、分散又は集中型）の中に維持される。レジストリは、未使用ブロックチェーントランザクション（UTXO）の中で識別可能なRedeemScriptハッシュに関連付けられた個々の所有者の公開鍵を含んで良い。UTXOは、各公開鍵により所有される株数を示す。さらに、電子メールアドレス等のような他の情報が格納されて良い。

## 【0059】

＜費用を計算し支払う＞

資産の所有権に費用を生じさせるために、

- ・資産は、その現在の所有権を決定する能力を有しなければならない、
- ・資産は、所与の時間期間の間の合計費用／収入を決定可能でなければならない、
- ・資産は、現在の所有権の間で費用／収入を分割可能でなければならない、
- ・資産は、現在の資産保有者から／へのこれらの費用／収入の支払をトリガ可能でなければならない、支払は、パーティ間で何らかの値を交換するためのブロックチェーントランザクションを生成することにより、実行される。

## 【0060】

支払ルールは、資産の種類、及び収入の種類に依存して、簡易又は複雑であって良く、必要なインプットデータ及び／又はパラメータの提供が可能である。例えば、収入ルールは、それらの現在の所有権比率に従い、合計収入のうちの割り当てを支払うことであり得る。別の例では、各保有者は、収入が生じた時間期間に渡る彼らの所有権比率に基づき、収入の割り当てを与えられるべきであって良い。別の例では、「ボーナス」が、所与の時間長の間の株所有権の保持に対して、支払われて良い。これらの例は、単に説明のためであり、設定可能な異なる支払ルールの数又は種類を限定しない。

## 【0061】

取引執行（例えば、未払費用の回収）は、幾つかの方法で達成可能である。これらのう

10

20

30

40

50

ちの幾つかは、自動的に（ここでも計算エージェントを使用することにより）又は非自動的手段により制御できる。例えば、次の通りである。

- ・ 法的補償
- ・ 資産の所有権の再割り当て（発行人が資産再割り当てに副署することを要求される場合のみ可能であるが）
- ・ 費用が回収されるまで、収入の支払を阻止する。

【0062】

＜他の機能＞

他の（任意的）機能は、エージェントを設計すること及び相応してルールを設定することにより、構築できる。説明のための、非包括的な例は、以下を含む：

- ・ 資産は、収入から費用を相殺することが可能であって良い、
- ・ 資産は、前の費用が支払われていない場合、収入を保持することが可能であって良い、
- ・ この収入／費用／相殺は、時間期間の終わりに（例えば、6ヶ月毎に）又は生成されるときに（例えば、直ちに）到来して良い。

【0063】

＜技術的モデル＞

本願明細書に提示される技術的ソリューションは、計算エージェント及びスマートコントラクトの使用を介して資産の制御のためのブロックチェーンに基づくメカニズムを提供する。本発明は、ブロックチェーントランザクションセットの自動生成を可能にする。これらのトランザクションは、スマートコントラクトの実行に関連する費用の取り戻し又は収入の支払のいずれかへの支払のために使用できる。図1は、本発明の一実施形態によるシステムの概観を示す。該システムは、該システムのアーキテクチャを実装するよう構成され得るコンポーネントを含む。

【0064】

＜Cost Agent（費用エージェント）＞

費用エージェントは、トリガ条件が満たされると、

- ・ 全体として資産の合計未払い金を計算し、
- ・ 取引内で定められた支払ルールに基づき、資産の現在の所有権分割に渡り、例えば比例して、個々の支払を計算し、
- ・ 個々の資産保有者の各々について支払トランザクションを生成する。

【0065】

必要な場合には、所与の取引に対して定められた1つより多くの、例えば定期的な費用について1つ、臨時費用について1つの、費用エージェントが存在し得る。

【0066】

＜Income Agent（収入エージェント）＞

収入エージェントは、トリガ条件が外れると、

- ・ 全体として資産の合計未払い金を計算し、
- ・ 取引内で定められた支払ルールに基づき、資産の現在の所有権分割に渡り、例えば比例して、個々の支払を計算し、
- ・ 個々の資産保有者の各々について支払トランザクションを生成する。

【0067】

必要な場合には、所与の取引に対して定められた1つより多くの（例えば定期的な収入について1つ、臨時収入について1つの）収入エージェントが存在し得る。

【0068】

＜Netting Agent（相殺エージェント）＞

相殺エージェントは、トリガ条件が満たされると、

- ・ 支払アドレスの一致する複数の個々のトランザクションから、支払トランザクションを単一の相殺トランザクションに合併整理する。

【0069】

本願で提供される例では、単一の相殺取引のみが存在する。しかしながら、重要なこと

10

20

30

40

50

に、本発明はこれに関して限定されないことに留意する。1又は複数の実施形態では、複数の相殺エージェントが存在して良い。このような場合には、各インスタンスは、明確に定められた別個のルールセットと共に提供される。

【0070】

＜現在の所有権の分割を決定する＞

本願明細書に記載される実施形態は、資産の現在の所有権がレジストリ（又はデータベース）及びUTXOリストの組み合わせにより決定できると仮定する。用語「UTXOリスト」は、従来知られており、未だ使用されていないブロックチェーン上のトランザクションアウトプットのリストを表す。

【0071】

図2は、本願明細書に記載の技術的モデルに関連する幾つかの主要な使用例を示す。これらは、以下に更に詳細に説明される。

【0072】

＜所有権の決定【100】＞

留意すべきことに、以下の例は、資産自体により動作が直接実行されると仮定する。しかしながら、他の実施形態では、この主体は、同じ処理を使用する何らかの他の資産制御部により置換されて良い。本使用例では、資産（Asset）は、支払を実現するために自身の現在の所有権を決定する必要がある。

【表1】

【表2】

| ステップ   | 説明                                                                                                                                                              |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 100.10 | <p>Contractに関連付けられたOwnership Registryから、Agentプロセスが以下を決定できる：</p> <ul style="list-style-type: none"> <li>・所与の期間内の資産の所有権、及び、</li> <li>・各所有者について、資産の割り当て</li> </ul> |

＜支払計算【200】＞

再び、本例における主要主体は資産である。しかしながら、他の実施形態では、資産は別の主要主体により置き換えられて良い。支払計算の本使用例では、資産は、自身が（上述の使用例100で決定したような）自身の現在の所有者に支払う又はそれら収集すべき収入及び／又は費用の額を計算したいと望む。

【表2】

10

20

30

40

50

【表 3】

| ステップ   | 説明                                                                                                                                                                                                                           |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200.10 | <p>A g e n t に依存して、A s s e t は、相応の収入を計算すべき指定期間についての全ての情報を引き込み、相応の費用を計算すべき期間についての全ての情報を引き込む。</p> <p>注：所与の A g e n t は、費用又は収入のみを計算し、又は両方とも計算しない。</p> <p>注：使用されるメカニズムは、資産自体の特性に特有であり、関連取引文書の中で正式に定められる。</p>                     |
| 200.20 | <p>200.10からのアウトプットは、次に、取引に従い割り当てに配分される。</p> <p>例えば、アウトプット値を存在する割り当て数で除算し、次に期間内で保持された時間に基づき比率を定めることによる（しかし、C o n t r a c t が割り当てルール上で更なる複雑性を提供することを妨げない）。</p> <p>丸め込みルールは、関連取引文書の中で定められる傾向があるが、通常、相応の収入を切り捨て、相応の費用を切り上げる。</p> |
| 200.30 | <p>ステップ200.40で導出された単位当たり配当は、使用例100で導出された保有権割り当てを用いて単なる乗算を通じて保有権毎に割り当てられる。</p>                                                                                                                                                |
| 200.40 | <p>費用が回収される必要がある場合、使用例300は、個々の保有権毎に実行される。</p>                                                                                                                                                                                |
| 200.50 | <p>収入が支払われる必要がある場合、使用例400は、個々の保有権毎に実行される。</p>                                                                                                                                                                                |

#### <費用を支払う [300]>

再び、本例における主要主体は資産である。資産保持者（Asset Holder）は、資産の所有権の結果として発生した費用を支払うことを望む。

【表 3】

10

20

30

40

50

【表 4】

| ステップ   | 説明                                                                                                                                                                                                                                                                                                                                                                                                |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 300.10 | この状況では、Assetは、資産保有者についてAsset Holderに渡すトランザクションを生成して、必要なトランザクションインプットを追加し、次にトランザクションを完成させるために署名する。                                                                                                                                                                                                                                                                                                 |
| 300.20 | Assetは、以下を有する部分的に形成されたトランザクションを生成する： <ul style="list-style-type: none"> <li>・トランザクションインプットがない、</li> <li>・1つのトランザクションアウトプット。</li> </ul> トランザクションアウトプットは、 <ul style="list-style-type: none"> <li>・Assetの公開鍵ハッシュアドレスに支払われる費用トランザクションである。</li> </ul> しかしながら、Assetにより適用される署名は、S I G H A S H _ A L L   S I G H A S H _ A N Y O N E C A N P A Yとしてマークされる。これは、アウトプットが変更されるのを防ぐが、追加のトランザクションインプットの適用を許容する。 |
| 300.30 | Assetは、このトランザクションを完了するためにAsset Holderに送信する。                                                                                                                                                                                                                                                                                                                                                       |
| 300.40 | Asset Holderは、次に、本トランザクション及びマイニング料金の和について正しいサイズのトランザクションアウトプットを生成するために、トランザクションを生成する。                                                                                                                                                                                                                                                                                                             |
| 300.50 | Asset Holderは、次に、ステップ300.30からのトランザクションアウトプットを、300.20で生成されたトランザクションへの追加インプットとして使用する。それらは次に本トランザクションに署名し提出する。                                                                                                                                                                                                                                                                                       |

## ＜考察＞

留意すべきことに、資産保持者が彼らの比率の費用を支払う特定の動機がないことである。上述の処理では、2つの経路を通じて可能な補償が存在する。

・費用が支払われない場合、後に生じる収入は保留され得る。これは、取引の中で定められる必要がある（資産に対して生成される収入が存在すると仮定する）。しかし、これは、ステップ100.10の中で検出することが容易である。

・関連法的チャネルを通じる法的補償。

## 【0073】

資産保持者が支払わない場合、n人のうちのm人のマルチシグネチャトランザクションを用いて、彼らの資産所有権が取り消され、資産自体に戻されるように、発行トランザクションについて公証人が存在するように、取引を生成することが可能である。このシナリオでは、以下の通りである。

・ステップ300.20が実行された後に、費用を支払うために許可された時間に基づき設定されたロック時間と共に、第2トランザクションが生成される。これは、最後のトランザクション発行のトランザクションインプットを取り入れるが、資産を資産に割り当て戻す。このトランザクションは、関連公証人により署名される。

・資産保持者が彼らの費用を支払う場合、このトランザクションは無効であるが（トランザクションインプットが使用済みなので）、支払わない場合、資産所有権は取り消される。支払トランザクションは効果的に資産と共に株を再発行するので、本トランザクションは、次の期間にロールオーバーすることができず、むしろ、費用トランザクションが生成されるときはいつも、2つのトランザクションが生成されなければならない。

## 【0074】

### <変形 1>

上述の例は、費用がビットコインで支払われると仮定した。しかしながら、任意のトークン化された値が使用可能なので、これについて要件が存在しない。この状況では、以下の変更が必要である。

・ステップ 300.50 は、単に 1 つではなく、2 つのトランザクションインプットを追加する。1 番目は BTC でのマイニング料金であり、2 番目は（例えばフィアット通貨での）トークン化された費用トランザクションである。

【0075】

### <変形 2>

上述の例は、マイニング料金が、（他の費用と一緒に）資産保持者により支払われる。これについて要件は存在しない。資産自体によりマイニング料金が支払われる場合、以下の変更が必要である。

・ステップ 300.20 は、2 つのトランザクションインプットを生成し、資産発行トランザクションインプット、及びマイニング料金のための BTC トランザクションインプットである。

【0076】

### <変形 3>

資産保有者毎に費用トランザクションを生成するのではなく、単一の費用トランザクションが生成され、各個人資産保有者が彼らのインプットを追加する（資産保有者毎に別個のトランザクションではない）。トランザクションアウトプットは、代替経路を通じて通知される、資産保有者毎の額の詳細による相応の費用の合計値である。

【0077】

このモデルは、単純な分割、及び比較的少数の資産保有者に有効であるが、全ての所有者が支払に合意するときのみトランザクションが有効なので、多くの場合に実用的ではない。資産の観点から、支払が遅延される又は多くの資産保有者によって行われない場合のあることが想定される状況では、最短可能時間内に受信される収入額を最大化するという点で、変形 3 が好適である。

【0078】

支払のために使用される実際の変形は、マスタエージェントに埋め込まれ（ハードコードされ）て良い又は別個のルックアップテーブルに置かれて良い所定ルールセットに基づき、マスタエージェントにより制御できる。例えば、ルールは、単純なブールフラグに基づき、どの変形が使用されるべきかを管理して良い、又は、閾所有者数又は任意の他の適切な試験可能な条件を含んで良い。

【0079】

### <収入を支払う [400]>

再び、本例における主要主体は資産である。資産は、それらの所有権に比例して、自身の所有者に収入を支払いたいと望む。

[表 4]

10

20

30

40

50

【表 5】

| ステップ   | 説明                                                                                                                                                                                                                                                                                                                       |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 400.10 | <p>Assetは、資産保有者毎に、1つのトランザクションインプット及び1つのトランザクションアウトプットを有するトランザクションを生成する。</p> <p>トランザクションインプットは、</p> <ul style="list-style-type: none"> <li>・合計収入分配額、及びマイニング料金</li> </ul> <p>現在の投資家毎に1つのトランザクションアウトプットがあり、以下のように構造化される。</p> <ul style="list-style-type: none"> <li>・Asset Holderの公開鍵ハッシュに支払われる収入分配インプットトランザクション</li> </ul> |

10

#### <変形>

収入が、固有のBTCではなく、トークン化された形式で支払われる場合を検討する。この状況では、トランザクション収入トランザクションは、以下を含むRedeemScriptアドレス指定される。

- ・資産保有者の公開鍵、
- ・収入分配に関連付けられたメタデータ

20

#### <例示的シナリオー競走馬>

本発明の実行可能な実装を説明する目的で、例示的シナリオが以下に提供される。シナリオのための個々のトランザクションシーケンスは、添付の図面の中で提供される。

#### 【0080】

この説明のためのシナリオでは、stableは、競走馬の中で、関連する賞金の権利及びstable料金のような支払義務を維持しながら、株を発行することを選択する。本例では、stableは50%の所有権を保持し、一方でAliceは30%、Bobは20%を購入する。

#### 【0081】

契約上の義務として、所有者は、彼らの詳細を発行人(Issuer、Stable)に登録しなければならない。これは、ビットコイン支払(つまり、BTC Address)の公開鍵を簡略化し得る。しかしながら、これは、通知のための電子メールアドレス、正式名称、誕生日、住居/職場所在地、納税番号、等も含んで良い。これらのうちの任意のものは、発行人自身の処理要件及び/又は発行人が満たさなければならないと感じ得る又はそう感じない任意の規定に依存して契約上の要件又は選択肢であって良い。

30

#### 【0082】

発行人の要望に依存して、取引トークン(つまり、メタデータを含むビットコイントランザクション)は、所有権の全ての転送のために(つまり、「web-of-trust」の状況で)、発行人の署名を要求して良い、又は(本例におけるように)発行人の署名無しで転送されることを許可されて良い。後者の場合、取引の契約条件により、新しい所有者は、彼らの所有権を登録することを、発行人と契約しなければならない(1.1章を参照)。

40

#### 【0083】

これらの登録された詳細は、取引自体の中に格納され、(所有権情報のような)変更可能なフィールドを除く取引のハッシュに依存する。事実上、取引に対するハッシュは、取引自体の中で定められる。これは、取引の必ずしも全ての属性がハッシュに含まれる必要がないことを意味する。これは、所有権リストが取引文書の中に保持されることを可能にするが、所有者はハッシュに含まれない。したがって、文書の署名は、文書の所有者が変更されても、不変のままである。

#### 【0084】

#### <取引の概要>

50

以下は、これらの条項を実装する２つのエージェントに関する詳細を含む取引の人間により可読なバージョンの選択された強調部分を提示する。

【表 6】

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <p>Naka' s Revenge Title Contract<br/>         Issuer: Blockchain Stable Ltd.; hereafter 'the Stable'<br/>         Public Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<br/>         This contract covers ... &lt;some content&gt; ... Part share of ownership of racehorse known as Naka' s Revenge ... &lt;more content&gt; ...<br/>         Contract specifics<br/>         NumShares 1 0<br/>         ShareVal 1 0 %<br/>         Peggingrate 1 0 0 0 Satoshis/share<br/>         ...</p>                                                                               | 10 |
| <p>条項 1 : 所有権登録<br/>         1. 1 株の所有権は、トークンを所有する公開鍵に関する最近の検証可能情報として決定される。I s s u e r は、以下のうちのいずれも満たされない場合に、収入支払を誤る責任を受け入れない。<br/>         1. 1. 1 T o k e n は、I s s u e r により又は認可された署名者により署名されている。<br/>         1. 1. 2 T o k e n 所有者は、彼らの署名と一緒に関連する R e d e e m S c r i p t を提示することにより、所有権を証明している。<br/>         1. 1. 3 従属条項 1. 1. 1 及び／又は 1. 1. 2 に従い、所有者情報は、I s s u e r のレジストリに追加される。<br/>         1. 2 1. 1 に従い、所有者が彼らの公開鍵を I s s u e r に対して非公開のままにしたい、したがって I s s u e r のレジストリから無くしたいと望む場合、大部分の最近の登録者は「O w n e r」であると考えられ、全ての義務及び権利を残す。<br/>         ...</p> | 20 |
| <p>条項 3 : 収入の分配<br/>         発行人は、株所有者に、彼らの所有百分率に正比例して、レース日の 3 0 日以内に、全ての賞金の分け前を支払う。株所有者は、ここで、レース日におけるこの取引の保有者として定められる。<br/>         ...</p>                                                                                                                                                                                                                                                                                                                                                                                                                        | 30 |
| <p>条項 7 : 維持費<br/>         維持費（包括的であり、s t a b l e 料金、訓練、飼料、等）は、この取引の寿命の間、1 0 0 0 サトシ／日である。株所有者は、彼らの所有権に比例して、実際に 6 月 3 0 日に s t a b l e 料金を支払うことに合意する。料金は、日毎に発生し、株転送が生じると、新しい株所有者へ転送される。<br/>         ...</p>                                                                                                                                                                                                                                                                                                                                                        | 40 |
| <p>条項 2 8 : 終了<br/>         この合意は、以下の条件のうちのいずれかにより、直ちに害されることなく、終了する。<br/>         ...<br/>         2 8. 4 指名された競走馬の死亡</p>                                                                                                                                                                                                                                                                                                                                                                                                                                           | 40 |

#### <発行 (I s s u a n c e) >

S t a b l e は、まず、自身の秘密鍵と共に競走馬を登録する。これは、完全に新しい秘密鍵であって良く、又は以下に実質的に「サブキー生成の方法」と題された章で後述するようなメカニズムを使用して良い。

【0 0 8 5】

S t a b l e は、次に、S T A B L E - S 3 - T 1（図 5 を参照）の中にある発行トランザクションの競走馬口座にシード資金を供給する必要がある。

【0 0 8 6】

競走馬は、次に、R A C E H O R S E - S 3 - T 1（図 6 を参照）で、自身の株を、様

々な資産保有者に発行する。取引が、費用が突出する場合に（所有権が直接無効にされないことを意味する）賞金が支払われないことを明言するので、発行トランザクションは、所有者による直接の資産の自由な転送を可能にする。このトランザクションのアウトプット4は、競走馬自体が取引の条項28を実施可能にすることをカバーするスマートコントラクトを生成する。

【0087】

＜費用エージェント（Cost Agent）（Stable料金）＞

2年毎に、stable料金の請求を担う費用エージェントが日にちによりトリガされる。費用は、1000サトシ／株／日（簡単のため、本例は6ヶ月＝180日とする）として計算される。

10

【0088】

キャッシュフローを向上するために、エージェントは、公開鍵所有者毎に、別個のトランザクションを生成する。ここで、所有者直詳細は現行取引に関連付けられた資産レジストリ（又は発行人自身の私設内部データベース）から取り入れられ、及び現在の株割り当ては（UTXO上の基本BTC及び取引で指定されたペギングレートに基づき）ブロックチェーンから取り入れられる。ペギングレートに関する詳細については、以下の「コード化スキーム」に関する章を参照のこと。

【0089】

エージェントの高レベル処理は、次の通りである。

・Ownership Registry（所有権レジストリ）及び関連UTXOから所有権を決定する。

20

－ 本例では、これはStable-5（PubK-Stable）、Alice-3（PubK-Alice）、Bob-2（PubK-Bob）のリストを提供する。

・期間の間の、相応する合計費用を計算する。

－ 本例では、これは合計180,000サトシを提供する。

・取引（Contract）から割り当てルールを決定する。

－ 本例では、Alice（54,000サトシ）及びBob（36,000サトシ）及びStable（90,000サトシ）のみに課金する。

・支払トランザクションを生成し、（トランザクションインプットが続いて追加されることを可能にするために）いかなるトランザクションインプットもS I G H A S H \_ A N Y O N E C A N P A Y マーカを用いて署名されない。

30

－ 本例では、これは、R A C E H O R S E - S 3 - T 2 及び R A C E H O R S E - S 3 - T 3 である。

・支払がブロックチェーン上に公開されるのを待つ。取引自体は、エージェントに、支払が受信されないとき、どのように振る舞うべきかを指示する。例えば、支払が14日以内に受信されない場合に、リマインダが送信されて良く、60日後に口座が滞納とマークされる。

・支払が後に受信された場合、更なる収集活動を止めるために、監視エージェントを更新する。

【0090】

40

図7は、本例の本態様に関連するトランザクションを示す。ここで、stable料金の第1セットは、投資家から請求される必要がある（6ヶ月の価値、180000サトシ／株）。金銭がstableに返済されるので、費用トランザクションは、隠れ所有者に生成されもとされない（stableが自身の株を「隠そう」としていた場合、ブロックチェーン上の公開記録は所有者の間を区別しない）。Alice及びBobに変化が与えられないので、正しいサイズのトランザクションインプットを生成すると想定されたトランザクションがある（A L I C E - S 3 - T 1 及び B O B - S 3 - T 1 であると仮定する）。

【0091】

＜費用の投資家による支払＞

50

不完全なトランザクションを受信すると、投資家（A l i c e 及び B o b）は、料金を満たし及び彼らのビットコインウォレットを用いてトランザクションを完成させるために、正しいトランザクションインプットを割り当てる。これは、スタブ（stubbed）トランザクション R A C E H O R S E - S 3 - T 2（A l i c e）及び R A C E H O R S E - S 3 - T 3（B o b）にトランザクションインプットを追加する（図7を参照）。

【0092】

＜収入エージェント（Income Agent）（賞金）＞

競走馬により生じた賞金は、s t a b l e が競馬場からの支払を受信すると、支払われる。状況に依存して、賞金は、B T C で受信されて良く（例えば、競馬産業がビットコイントランザクションが可能である場合、又は、特定の競馬場がビットコインでの賞金を有効にする、等）、又はフィアット通貨で受信されて良い。後者の場合、s t a b l e は、収入分配ステップを実行する前に、賞金を B T C に変換するステップを実行して良い。

【0093】

このエージェントの高レベル処理は、次の通りである。

- ・取引の使用は、この投資についてどのように収入を生じるかを決定する。
- － 本例では、レース結果の詳細について、外部ニュースフィードが調べられる。
- ・資金供給要件について、（取引の中で定められるような）関連パーティと通信する。
- － 本例では、彼らが競走馬を代表して競馬場から受信すべき賞金に基づき支払うための請求書を含む電子メールが、S t a b l e へ送信される。
- ・O w n e r s h i p R e p o s i t o r y から所有権を決定する。
- － 本例では、これは S t a b l e - 5（P u b K - S t a b l e）、A l i c e - 3（P u b K - A l i c e）、B o b - 2（P u b K - B o b）のリストを提供する。
- ・収入が関連口座に支払われるのを待つ。
- － 本例では、80 B T C になる £ 20,400 の賞金が、競走馬の公開鍵ハッシュ（P u b l i c K e y H a s h）アドレスに支払われる。
- ・期間の間の、相応する賞金をどのように分けるかを計算する。
- － 本例では、これは合計 8 B T C / 株を提供する。
- ・取引（C o n t r a c t）から割り当てルールを決定する。
- － 本例では、S t a b l e 40 B T C、A l i c e 24 B T C、及び B o b 16 B T C である。
- ・報酬を支払うために支払トランザクションを生成する。

【0094】

図8は、競走馬が s t a b l e 料金を s t a b l e に渡すトランザクションを示す。s t a b l e が自身の所有権を秘密に保っていなかった場合、S 3 - T 2 ~ S 3 - T 5 における競走馬の公開鍵ではなく、単に s t a b l e の公開鍵を用いることにより、s t a b l e 料金を s t a b l e に直接支払わせることが可能であることに留意する。

【0095】

本発明と関連して使用され得る他の技術及びメカニズムは、以下に更に詳述される。

【0096】

＜コード化スキーム＞

ブロックチェーントランザクションの中のメタデータは、種々の方法でフォーマット化できる。しかしながら、1つの可能なコード化スキームがここに記載される。

【0097】

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

【0098】

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

10

20

30

40

50

## 【0099】

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

i) 取引の下で利用可能な株の量

(これは、本願明細書で「NumShares」と呼ばれることがある)、

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

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

## 【0100】

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

## 【0101】

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

## 【0102】

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

## 【0103】

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

## 【0104】

限定トークンは、NumSharesと呼ばれる量により定められる固定非ゼロの株数により合計発行値が固定された (又は「限定された」) トークンである。したがって、限定取引の下では更なる株は発行されない。例えば、競走馬の共同所有権のための取引は、

10

20

30

40

50

競走馬の100%に限定される（例えば、それぞれ1%で100個の株、又はそれぞれ10%で10個の株、等）。非限定取引は、例えば要求額のフィアット通貨を彼らの引当金（Reserve Account）に追加することにより、発行人が株の更なる発行を引き受け可能であることを意味する。NumSharesは、全ての取引に明示的に記載されなければならない。限定取引は、NumShares>0を有しなければならない。非限定取引は、NumShares=0を設定することにより示される。

#### 【0105】

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

#### 【0106】

ペギングレートは、分割可能取引にのみ適用される。ここで、(ShareValと呼ばれる量により表される)株の値は、基礎BTC額に固定される。例えば、取引は、発行人が基礎1BTC毎に\$10,000のレートでトークンを償還することを約束すると指定して良い。これは、(例えば)15,400サトシのトークン化された基礎アウトプット値を有するトランザクションが\$1.54で償還可能であることを意味し得る。ペギングレートについて0の値は、取引が非分割可能であることを示す（つまり、無記名債券のように、全体のみが転送可能である）。ペギングレートが0に設定されると（非分割可能トークンを意味する）、基礎BTC値は、取引値と無関係であり、任意の額に設定可能である。通常、この場合には、運用コストを最小化するために、基礎BTC額を可能な限り小さく保つ（つまり、ダストに設定する）ことが望ましい。

#### 【0107】

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

#### 【0108】

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

#### 【0109】

TotalIssuanceは、発行された株の合計値を表す。この値は限定取引にのみ関連する。非限定取引については、発行は固定されず、より多くの株が発行されて良い。株がパーセンテージとして表現される場合、定義によりTotalIssuance=0を設定100%である。

#### 【0110】

限定取引では、NumShares、ShareVal、及びTotalIssuanceは、以下のように関連する。

【0111】

$$\text{NumShares} \times \text{ShareVal} = \text{TotalIssuance}$$

TotalIssuanceの0の値は、非限定取引であることを意味する。非限定取引の一例は、フィアット通貨である（したがって、TotalIssuanceは0に設定される）。限定取引の例は、（i）限定版記念硬貨（1000個鑄造され、1株＝1硬貨である）、TotalIssuance＝1000×1＝1000個の硬貨、（ii）入場券のある会場の席、TotalIssuance＝利用可能な合計席数、である。

【0112】

流通は、未使用トークンの合計値として定められる（つまり、UTXO（未使用トランザクションアウトプット）の中のトランザクションにより決定される）。全部の未使用トランザクションの完全な集合は、全てのビットコインノードに利用可能なリストの中に保持される。例えば、発行人が最初に\$10,000をフィアット通貨型のトークンとして発行し、時間を経て\$5500の価値のトークンが償還された場合、流通＝\$4500である（未償還トークンの値である）。この値は、関連する準備口座の差引残高に調整されるべきである。

【0113】

<サブキー生成の方法>

以上では、元の（マスタ）キーからサブキーを生成することが有利である状況を参照した。これが実行され得る1つの方法を説明するために、これを達成する方法が以下に提供される。以下の説明は、図12～18を参照して与えられる。

【0114】

図12は、通信ネットワーク5を介して第2ノード7と通信する第1ノード3を含むシステム1を示す。第1ノード3は関連する第1処理装置23を有し、及び第2ノード5は関連する第2処理装置27を有する。第1及び第2ノード3、7は、コンピュータ、電話機、タブレットコンピュータ、モバイル通信装置、コンピュータサーバ、等のような電子装置を含んで良い。一例では、第1ノード3はクライアント（ユーザ）装置であって良く、第2ノード7はサーバであって良い。サーバは、デジタルウォレットプロバイダのサーバであって良い。

【0115】

第1ノード3は、第1ノードマスタ秘密鍵（V<sub>1C</sub>）及び第1ノードマスタ公開鍵（P<sub>1C</sub>）を有する第1非対称暗号対に関連付けられる。第2ノード7は、第2ノードマスタ秘密鍵（V<sub>1S</sub>）及び第2ノードマスタ公開鍵（P<sub>1S</sub>）を有する第2非対称暗号対に関連付けられる。言い換えると、第1及び第2ノードは、それぞれ、個々の公開－秘密鍵対を保有する。

【0116】

個々の第1及び第2ノード3、7の第1及び第2非対称暗号対は、ウォレットの登録のような登録処理中に生成されて良い。各ノードの公開鍵は、通信ネットワーク5に渡るように、公に共有されて良い。

【0117】

第1ノード3及び第2ノード7の両者において共通シークレット（common secret：SC）を決定するために、ノード3、7は、通信ネットワーク5を介して秘密鍵を通信することなく、それぞれ方法300、400のステップを実行する。

【0118】

第1ノード3により実行される方法300は、少なくとも第1ノードマスタ秘密鍵（V<sub>1C</sub>）及び生成器値（Generator Value：GV）に基づき、第1ノード第2秘密鍵（V<sub>2C</sub>）を決定するステップ330を含む。生成器値は、第1ノードと第2ノードとの間で共有されるメッセージ（M）に基づいて良い。これは、以下に詳述するように、通信ネットワーク5を介してメッセージを共有するステップを含んで良い。方法300は、少なく

10

20

30

40

50

とも第2ノードマスタ公開鍵 ( $P_{1S}$ ) 及び生成器値 (Generator Value:  $GV$ ) に基づき、第2ノード第2公開鍵 ( $P_{2S}$ ) を決定するステップ370を更に含む。方法300は、第1ノード第2秘密鍵 ( $V_{2C}$ ) 及び第2ノード第2公開鍵 ( $P_{2S}$ ) に基づき、共通シークレット (common secret:  $CS$ ) を決定するステップ380を含む。

#### 【0119】

重要なことに、同じ共通シークレット ( $CS$ ) が、方法400により第2ノード7においても決定できる。方法400は、第1ノードマスタ公開鍵 ( $P_{1C}$ ) 及び生成器値 (Generator Value:  $GV$ ) に基づき、第1ノード第2公開鍵 ( $P_{2C}$ ) を決定するステップ430を含む。方法400は、第2ノードマスタ秘密鍵 ( $V_{1S}$ ) 及び生成器値 (Generator Value:  $GV$ ) に基づき、第2ノード第2秘密鍵 ( $V_{2S}$ ) を決定するステップ470を更に含む。方法400は、第2ノード第2秘密鍵 ( $V_{2S}$ ) 及び第1ノード第2公開鍵 ( $P_{2C}$ ) に基づき、共通シークレット (common secret:  $CS$ ) を決定するステップ480を含む。

#### 【0120】

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

#### 【0121】

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

#### 【0122】

方法300、400は、追加ステップを含んで良い。図13を参照する。方法300は、第1ノード3において、メッセージ ( $M$ ) 及び第1ノード第2秘密鍵 ( $V_{2C}$ ) に基づき、署名メッセージ ( $SM1$ ) を生成するステップを含んで良い。方法300は、第1署名メッセージ ( $SM1$ ) を通信ネットワーク5を介して第2ノード7へ送信するステップ360を更に含む。一方、第2ノード7は、第1署名メッセージ ( $SM1$ ) を受信するステップ440を実行して良い。方法400は、第1署名メッセージ ( $SM2$ ) を第1ノード第2公開鍵 ( $P_{2C}$ ) により検証するステップ450、及び第1署名メッセージ ( $SM1$ ) を検証するステップの結果に基づき第1ノード3を認証するステップ460を更に含む。有利なことに、これは、第2ノード7が、(第1署名メッセージが生成された場所である) 意図された第1ノードが第1ノード3であることを認証することを可能にする。これは、第1ノード3だけが、第1ノードマスタ秘密鍵 ( $V_{1C}$ ) へのアクセスを有する、したがって、第1ノード3だけが、第1署名メッセージ ( $SM1$ ) を生成するための第1ノード第2秘密鍵 ( $V_{2C}$ ) を決定できるという仮定に基づく。理解されるべきことに、同様に、第2署名メッセージ ( $SM2$ ) は、第2ノード7において生成され、第1ノード3へ送信され得る。したがって、ピアツーピアシナリオにおけるように、第1ノード3は、第2ノード7を認証できる。

#### 【0123】

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

とを理解する。

【0124】

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

【0125】

＜登録の方法100、200＞

登録の方法100、200の一例は、図14を参照して記載される。図14では、方法100は第1ノード3により実行され、方法200は第2ノード7により実行される。これは、それぞれ第1ノード3及び第2ノード7のために第1及び第2非対称暗号対を確立するステップを含む。

【0126】

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

【0127】

方法100、200では、これは、共通ECCシステムに合意している110、210、且つ基点（G）を用いる、第1ノード及び第2ノードを含む（注：基点は、共通生成器として参照され得るが、用語「基点」は、生成器値GVとの混同を避けるために使用される）。一例では、共通ECCシステムは、ビットコインにより使用されるECCシステムであるsecp256k1に基づいて良い。基点（G）は、選択され、ランダムに生成され、又は割り当てられて良い。

【0128】

ここで第1ノード3について考えると、方法100は、共通ECCシステム及び基点（G）を解決するステップ110を含む。これは、第2ノード7又は第3ノード9から、共通ECCシステム及び基点を受信するステップを含んで良い。代替として、ユーザインタフェース15は、第1ノード3に関連付けられる。これにより、ユーザは、共通ECCシステム及び／又は基点（G）を選択的に提供できる。更に別の代替案では、共通ECCシステム及び／又は基点（G）の一方又は両方が、第1ノード3によりランダムに選択されて良い。第1ノード3は、通信ネットワーク5を介して、基点（G）と共に共通ECCシステムを使用することを示す通知を、第2ノード7へ送信して良い。また、第2ノード7は、共通ECCシステム及び基点（G）の使用に対する肯定応答を示す通知を送信することにより、解決して良い210。

【0129】

方法100は、第1ノード3が、第1ノードマスタ秘密鍵（V1C）及び第1ノードマスタ公開鍵（P1C）を有する第1非対称暗号対を生成するステップ120を更に含む。これは、共通ECCシステムの中で指定された許容範囲の中のランダム整数に少なくとも部分的に基づき、第1マスタ秘密鍵（V1C）を生成するステップを含む。これは、次式に従い第1ノードマスタ秘密鍵（V1C）及び基点（G）の楕円曲線点乗算に基づき、第1ノードマスタ公開鍵（P1C）を決定するステップを更に含む。

$$P1C = V1C \times G \quad (\text{式1})$$

したがって、第1非対称暗号対は、以下を含む：

V1C：第1ノードにより秘密に保持される第1ノードマスタ秘密鍵。

P1C：公に知られる第1ノードマスタ公開鍵。

【0130】

第1ノード3は、第1ノードマスタ秘密鍵（V1C）及び第1ノードマスタ公開鍵（P1C）を、第1ノード3に関連付けられた第1データストア13に格納して良い。セキュリティのために、第1ノードマスタ秘密鍵（V1C）は、鍵が秘密のままであることを保証するために、第1データストア13のセキュアな部分に格納されて良い。

【0131】

方法100は、図14に示すように、第1ノードマスタ公開鍵（P1C）を通信ネット

10

20

30

40

50

ワーク5を介して第2ノード7へ送信するステップ130を更に含む。第2ノード7は、第1ノードマスタ公開鍵 ( $P_{1C}$ ) を受信すると220、第1ノードマスタ公開鍵 ( $P_{1C}$ ) を第2ノード7に関連付けられた第2データストア17に格納して良い230。

【0132】

第1ノード3と同様に、第2ノード7の方法200は、第2ノードマスタ秘密鍵 ( $V_{1S}$ ) 及び第2ノードマスタ公開鍵 ( $P_{1S}$ ) を有する第2非対称暗号対を生成するステップ240を含む。第2ノードマスタ秘密鍵 ( $V_{1S}$ ) も、許容範囲内のランダム整数である。また、第2ノードマスタ公開鍵 ( $P_{1S}$ ) は、次式により決定される。

$P_{1S} = V_{1S} \times G$  (式2)

したがって、第2非対称暗号対は、以下を含む：

$V_{1S}$ ：第2ノードにより秘密に保持される第2ノードマスタ秘密鍵。

$P_{1S}$ ：公に知らされる第2ノードマスタ公開鍵。

【0133】

第2ノード7は、第2非対称暗号対を第2データストア17に格納して良い。方法200は、第2ノードマスタ公開鍵 ( $P_{1S}$ ) を第1ノード3へ送信するステップ250を更に含む。また、第1ノード3は、第2ノードマスタ公開鍵 ( $P_{1S}$ ) を受信し140、格納して良い150。

【0134】

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

【0135】

登録ステップは、初期設定として1度生じるだけで良い。

【0136】

<セッション開始及び第1ノード3による共通シークレットの決定>

共通シークレット ( $CS$ ) を決定する一例は、図15を参照してここに記載される。共通シークレット ( $CS$ ) は、第1ノード3と第2ノード7との間の特定のセッション、時間、トランザクション、又は他の目的のために使用されて良く、同じ共通シークレット ( $CS$ ) を使用することが望ましい又はセキュアでなくて良い。したがって、共通シークレット ( $CS$ ) は、異なるセッション、時間、トランザクション、等の間で変更されて良い。

【0137】

以下は、上述したセキュアな送信技術の説明のために提供される。

【0138】

[メッセージ ( $M$ ) を生成する310]

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

【0139】

メッセージ ( $M$ ) = Unix時間 + ノンス (式3)

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

【0140】

方法300は、メッセージ ( $M$ ) を通信ネットワーク5を介して第2ノード7へ送信するステップ315を含む。メッセージ ( $M$ ) は秘密鍵についての情報を含まないで、メッセージ ( $M$ ) は、セキュアでないネットワークを介して送信されて良い。

10

20

30

40

50

## 【0141】

〔生成器値 (GV) を決定する320〕

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

$$GV = \text{SHA-256}(M) \quad (\text{式4})$$

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

## 【0142】

〔第1ノード第2秘密鍵を決定する330〕

方法300は、次に、第2ノードマスタ秘密鍵 ( $V_{1C}$ ) 及び生成器値 (GV) に基づき、第1ノード第2秘密鍵 ( $V_{2C}$ ) を決定するステップ330を含む。これは、次式に従い第1ノードマスタ秘密鍵 ( $V_{1C}$ ) 及び生成器値 (GV) のスカラ加算に基づき得る。

$$V_{2C} = V_{1C} + GV \quad (\text{式5})$$

したがって、第1ノード第2秘密鍵 ( $V_{2C}$ ) は、ランダム値ではないが、代わりに第1ノードマスタ秘密鍵から確定的に導出される。暗号対の中の対応する公開鍵、つまり第1ノード第2公開鍵 ( $P_{2C}$ ) は、以下の関係を有する。

$$P_{2C} = V_{2C} \times G \quad (\text{式6})$$

式5から式6に $V_{2C}$ を代入すると、次式を得る。

$$P_{2C} = (V_{1C} + GV) \times G \quad (\text{式7})$$

ここで、「+」演算子は楕円曲線点加算を表す。楕円曲線暗号代数は、分配的であり、式7は次式のように表すことができる。

$$P_{2C} = V_{1C} \times G + GV \times G \quad (\text{式8})$$

最後に、(式1) は(式7) に代入され、次式を得る。

$$P_{2C} = P_{1C} + GV \times G \quad (\text{式9.1})$$

$$P_{2C} = P_{1C} + \text{SHA-256}(M) \times G \quad (\text{式9.2})$$

したがって、対応する第1ノード第2公開鍵 ( $P_{2C}$ ) は、第1ノードマスタ公開鍵 ( $P_{1C}$ ) 及びメッセージ (M) の導出可能な所与の知識であり得る。方法400に関して以下に更に詳述するように、第2ノード7は、第1ノード第2公開鍵 ( $P_{2C}$ ) を独立に決定するために、このような知識を有して良い。

## 【0143】

〔メッセージ及び第1ノード第2秘密鍵に基づき、第1署名メッセージ (SM1) を生成する350〕

方法300は、メッセージ (M) 及び決定した第1ノード第2秘密鍵 ( $V_{2C}$ ) に基づき、第1署名メッセージ (SM1) を生成するステップ350を更に含む。署名メッセージを生成するステップは、メッセージ (M) にデジタル方式で署名するために、デジタル署名アルゴリズムを適用するステップを含む。一例では、これは、第1署名メッセージ (SM1) を得るために、楕円曲線デジタル署名アルゴリズム (Elliptic Curve Digital Signature Algorithm: ECDSA) の中でメッセージに第1ノード第2秘密鍵 ( $V_{2C}$ ) を適用するステップを含む。ECDSAの例は、secp256k1、secp256r1、secp384r1、secp521r1を有するECCシステムに基づくものを含む。

## 【0144】

第1署名メッセージ（SM1）は、第2ノード7において対応する第1ノード第2公開鍵（P2C）により検証できる。第1署名メッセージ（SM1）のこの検証は、第1ノード3を認証するために第2ノード7により使用されて良い。これは、方法400において以下に議論される。

## 【0145】

〔第2ノード第2公開鍵を決定する370'〕

第1ノード3は、次に、第2ノード第2公開鍵（P2S）を決定して良い370。上述のように、第2ノード第2公開鍵（P2S）は、少なくとも第2ノードマスタ公開鍵（P1S）及び生成器値（GV）に基づいて良い。本例では、公開鍵は、基点（G）との楕円曲線点乗算により秘密鍵として決定されるので370'、第2ノード第2公開鍵（P2S）は、式6と同様に次のように表すことができる。

$$P2S = V2S \times G \quad (\text{式10.1})$$

$$P2S = P1S + GV \times G \quad (\text{式10.2})$$

式10.2の数学的証明は、第1ノード第2公開鍵（P2C）について式9.1を導出するために上述したものと同一である。理解されるべきことに、第1ノード3は、第2ノード7と独立に第2ノード第2公開鍵を決定できる370。

## 【0146】

〔第1ノード3において共通シークレットを決定する380〕

第1ノード3は、次に、第1ノード第2秘密鍵（V2C）及び決定した第2ノード第2公開鍵（P2S）に基づき、共通シークレット（CS）を決定して良い380。共通シークレット（CS）は、第1ノード3により次式により決定されて良い。

$$S = V2C \times P2S \quad (\text{式11})$$

<第2ノード7において実行される方法400>

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

## 【0147】

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

## 【0148】

次のステップは、第1ノードマスタ公開鍵（P1C）及び生成器値（GV）に基づき、第1ノード第2公開鍵（P2C）を決定するステップ430を含む。本例では、公開鍵は、基点（G）との楕円曲線点乗算により秘密鍵として決定されるので430'、第1ノード第2公開鍵（P2C）は、式9と同様に次のように表すことができる。

$$P2C = V2C \times G \quad (\text{式12.1})$$

$$P2C = P1C + GV \times G \quad (\text{式12.2})$$

式12.1及び12.2の数学的証明は、式10.1及び10.2について上述したものと同一である。

## 【0149】

〔第2ノード7が第1ノード3を認証する〕

方法400は、未確認第1ノード3が第1ノード3であることを認証するために、第2ノード7により実行されるステップを含んで良い。上述のように、これは、第1ノード3から第1署名メッセージ（SM1）を受信するステップ440を含む。第2ノード7は、次に、ステップ430で決定された第1ノード第2公開鍵（P2C）により第1署名メッセージ（SM1）の署名を検証して良い450。

10

20

30

40

50

## 【0150】

デジタル署名の検証は、上述の楕円曲線デジタル署名アルゴリズム (Elliptic Curve Digital Signature Algorithm: ECDSA) に従い行われて良い。重要なことに、第1ノード第2秘密鍵 ( $V_{2C}$ ) により署名された第1署名メッセージ ( $SM1$ ) は、 $V_{2C}$  及び  $P_{2C}$  が暗号対を形成するので、対応する第1ノード第2公開鍵 ( $P_{2C}$ ) によってのみ正しく検証されるべきである。これらの鍵は、第1ノード3の登録で生成された第1ノードマスタ秘密鍵 ( $V_{1C}$ ) 及び第1ノードマスタ公開鍵 ( $P_{1C}$ ) において確定するので、第1署名メッセージ ( $SM1$ ) の検証は、第1署名メッセージ ( $SM1$ ) を送信する未確認第1ノードが登録中と同じ第1ノード3であることを認証する基礎として使用できる。したがって、第2ノード7は、第1署名メッセージを検証するステップ (450) の結果に基づき、第1ノード3を認証するステップ (460) を更に実行して良い (図16を参照)。

10

## 【0151】

〔第2ノード7が共通シークレットを決定する〕

方法400は、第2ノード7が、第2ノードマスタ秘密鍵 ( $V_{1S}$ ) 及び生成器値 ( $GV$ ) に基づき、第2ノード第2秘密鍵 ( $V_{2S}$ ) を決定するステップ470を更に含んで良い。第1ノード3により実行されるステップ330と同様に、第2ノード第2秘密鍵 ( $V_{2S}$ ) は、次式に従い、第2ノードマスタ秘密鍵 ( $V_{1S}$ ) 及び生成器値 ( $GV$ ) のスカラー加算に基づき得る。

20

$$V_{2S} = V_{1S} + GV \quad (\text{式13.1})$$

$$V_{2S} = V_{1S} + SHA-256(M) \quad (\text{式13.2})$$

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

$$S = V_{2S} \times P_{2C} \quad (\text{式14})$$

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

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

## 【0152】

30

第1ノード3により決定された共通シークレット ( $CS$ ) を考えると、次のように式10.1は式11に代入できる。

$$S = V_{2C} \times P_{2S} \quad (\text{式11})$$

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

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

第2ノード7により決定された共通シークレット ( $CS$ ) を考えると、次のように式12.1は式14に代入できる。

$$S = V_{2S} \times P_{2C} \quad (\text{式14})$$

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

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

40

EC代数学は可換性なので、次の通り式15及び式16は等価である。

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

〔共通シークレット ( $CS$ ) 及び秘密鍵〕

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

## 【0153】

共通シークレット ( $CS$ ) は、楕円曲線点 ( $x_S, y_S$ ) の形式であって良い。これは、ノード3、7により合意された標準的な公に知られた演算を用いて、標準的な鍵フォーマットに変換されて良い。例えば、 $x_S$  値は、AES256暗号鍵として使用され得る2

50

56ビットの整数であって良い。これは、更に、160ビットの長さの鍵を必要とする任意のアプリケーションのために、RIPEMD160を用いて160ビットの整数に変換され得る。

【0154】

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

10

【0155】

しかしながら、幾つかのアプリケーションに依存して、共通シークレット (CS) が第1ノードマスタ秘密鍵 (V1C) と同じくらいセキュアに保たれるならば、共通シークレット (CS) は、第1ノードに関連付けられた第1データストア (X) に格納され得る。

【0156】

有利なことに、この技術は、単一のマスタキー暗号対に基づき、複数のセキュアな秘密鍵に対応し得る複数の共通シークレットを決定するために使用できる。

【0157】

＜生成器値 (鍵) の階層構造＞

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

20

【0158】

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

30

$$M' = \text{SHA-256}(M) \quad (\text{式18})$$

$$GV' = \text{SHA-256}(M') \quad (\text{式19.1})$$

$$GV' = \text{SHA-256}(\text{SHA-256}(M)) \quad (\text{式19.2})$$

第1ノード3は、次に、次世代の第2ノード第2公開鍵 (P2S') 及び第1ノード第2秘密鍵 (V2C') を上述のステップ370及び330と同様であるが次式により適応され、決定して良い。

$$P2S' = P1S + GV' \times G \quad (\text{式20.1})$$

$$V2C' = V1C + GV' \quad (\text{式20.2})$$

第2ノード7は、次に、次世代の第1ノード第2公開鍵 (P2C') 及び第2ノード第2秘密鍵 (V2S') を上述のステップ430及び470と同様であるが次式により適応され、決定して良い。

40

$$P2C' = P1C + GV' \times G \quad (\text{式21.1})$$

$$V2S' = V1S + GV' \quad (\text{式21.2})$$

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

$$CS' = V2C' \times P2S' \quad (\text{式22})$$

第2ノード7は、次式により次世代共通シークレット (CS') を決定する。

$$CS' = V2S' \times P2C' \quad (\text{式23})$$

50

更なる世代（ $CS''$ 、 $CS'''$ 、等）は、チェーン階層構造を生成するために同じ方法で計算できる。この技術は、第1ノード3及び第2ノード7の両者が元のメッセージ（ $M$ ）又は最初に計算された生成器値（ $GV$ ）、及びそれがどのノードに関連するか、を見失わないことを要求する。これは公に知られた情報なので、この情報の保有に関してセキュリティ問題は存在しない。したがって、この情報は、（ハッシュ値を公開鍵にリンクする）「ハッシュテーブル」に保持され、（例えばTorrentを用いて）ネットワーク5に渡り自由に配布されて良い。さらに、階層構造内の任意の個々の共通シークレット（ $CS$ ）が今までに解決されていない場合、これは、秘密鍵 $V_1C$ 、 $V_1S$ がセキュアなままであるならば、階層構造の中の任意の他の共通シークレットのセキュリティに影響しない。

【0159】

<鍵の木構造>

上述のようなチェーン（線形）階層構造と同様に、木構造の形式の階層構造が生成できる。木構造によると、認証鍵、暗号鍵、署名鍵、支払鍵、等のような異なる目的の様々な鍵が決定されて良い。それにより、これらの鍵は、全て単一のセキュアに維持されるマスタキーにリンクされる。これは、種々の異なる鍵を有する木構造901を示す図17に図示される。これらの各々は、別のパーティと共有されるシークレットを生成するために使用できる。枝分かれする木は、幾つかの方法で達成でき、それらのうちの3つが以下に記載される。

【0160】

(i) マスタキー・スポーニング

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

$$V_2C = V_1C + \text{SHA-256}(M) \quad (\text{式24})$$

$$V_2C' = V_1C + \text{SHA-256}(\text{SHA-256}(M)) \quad (\text{式25})$$

$$V_2C'' = V_1C + \text{SHA-256}(\text{SHA-256}(\text{SHA-256}(M))) \quad (\text{式26})$$

等である。

【0161】

枝を生成するために、任意の鍵がサブマスタキーとして使用できる。例えば、 $V_2C'$ は、正規のマスタキーに対して行われるように、ハッシュを加算することにより、サブマスタキー（ $V_3C$ ）として使用できる。

$$V_3C = V_2C' + \text{SHA-256}(M) \quad (\text{式27})$$

サブマスタキー（ $V_3C$ ）は、それ自体が、次世代鍵（ $V_3C'$ ）を有して良い。例えば次式の通りである。

$$V_3C' = V_2C' + \text{SHA-256}(\text{SHA-256}(M)) \quad (\text{式28})$$

これは、図18に示すマスタキー・スポーニング（spawning）方法を用いて、木構造903を提供する。

【0162】

(ii) 論理的関連付け

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

【0163】

(iii) メッセージ多様性

新しい公開／秘密鍵ペアは、チェーン又は木の任意のポイントに新しいメッセージを導入することにより、生成できる。メッセージ自体は、任意であって良く、又は何らかの意味若しくは関数を伝達して良い（例えば、それは、「現実の」銀行口座番号に関連して良い

10

20

30

40

50

、等である)。新しい公開／秘密鍵ペアを形成するためのこのような新しいメッセージがセキュアに保持されることが望ましい場合がある。

【0164】

＜本発明と共に使用するための説明のための計算エージェント＞

本発明は、取引処理の自動化された態様を実行するために、計算リソース又はエージェントを利用できる。適切なエージェントの一例が以下に提供されるが、他の実装が使用されて良い。

【0165】

エージェントは、チューリング (Turing) 機械の実装において非消去可能テープとしてブロックチェーンを使用して、ブロックチェーンと連携して動作して良い。このエージェントは、ブロックチェーンネットワークと並列に実行し、(ループ) 処理の実行を監督し及び扱う。ループ処理は、例えば装置又はシステムの処理又は制御の自動化のような所与のタスクを実行するよう設計される。この並列リソースは、ブロックチェーンの状態を監視し、トランザクションをブロックチェーンに書き込ませることができる。ある意味で、これは、ブロックチェーンをチューリング機械の非消去可能テープとして利用し、以下の定義及び特徴を有する。

1. ブロックチェーンは、チューリング機械のテープとして作用する。ブロックチェーンの中の各トランザクションは、テープ上のセルを表す。このセルは、有限なアルファベットからの記号を含み得る。
2. テープヘッドは、ブロックチェーン上に既に書き込まれているブロックから情報を読み出すことができる。
3. テープヘッドは、多くのトランザクションを含む新しいブロックを、ブロックチェーンの終わりに書き込むことができる。しかしながら、それらは、既に存在するブロックに書き込むことができない、このように、ブロックチェーンテープは非消去可能である。
4. 各トランザクションのマルチシグネチャの P2SH (pay-to-script-hash) トランザクションの部分として格納され得る。

【0166】

エージェントに重要な機能は、ブロックチェーンの現在状態を監視する自動エンティティとして作用することである。これは更に、任意のオフブロックソースから、信号又は入力を受信できる。ブロックチェーン状態及び／又は受信した入力に依存して、エージェントは特定動作を実行して良い。エージェントは、どの動作が実行されるべきかを決定する。これらは、「現実世界」(つまり、オフブロック) の中の作用、及び／又(新しいトランザクションを生成する及びブロードキャストするような) はブロックチェーンに対する作用を含んで良く又は含まなくて良い。エージェントが取る作用は、ブロックチェーン状態によりトリガされて良い。エージェントは、更に、ビットコインネットワークにブロードキャストされるべき次のトランザクションセットについて決定し、後にブロックチェーンに書き込んで良い。

【0167】

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

【0168】

チューリング機械は、以下の2つのスタックを含む。

- ・データスタック：これは、上述のようにブロックチェーンにより表される。
- ・制御スタック：これは、エージェント機能により表される。これは、繰り返し制御フロー機能に関連する情報を格納する。

【0169】

制御スタックのデータスタックからの分離は、無限ループがビットコイン中核で生じる

ことを防ぎ、サービス拒否攻撃を軽減するという利点を提供する。

#### 【0170】

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

#### 【0171】

各反復について、

1. インデックスが1だけ増大する。終了条件については、インデックスが限界に達すると、反復は停止する。
2. 「if condition then action（[条件] ならば [動作] する）」（ICTA）文を含むコードブロックが実行される。動作は、ブロックチェーン上の又は外の任意の動作であって良い。
3. このサブルーチンの暗号ハッシュが計算される。これは、トランザクションの部分としてブロックチェーンに格納できる。ハッシュは各コードにユニークなので、どのコードが使用されているかの検証を可能にする。

#### 【0172】

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

- ・開始又はトリガ条件（例えば、特定日に達したとき）、
- ・繰り返し条件（つまり、前の反復に関連付けられたメタデータ又はハッシュ）、
- ・停止条件（つまり、ループの最後の反復）。

#### 【0173】

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

#### 【0174】

各トランザクションは、再利用可能な方法で、ブロックチェーンに保存される。ビットコイン実装では、トランザクションの中の各署名は、S I G H A S Hフラグを付加される。このフラグは、異なる値を取ることができる。これらの値の各々は、この署名の所有者の関与無しに、トランザクションの他の部分が変更され得るか否かを示す。再利用可能トランザクションは、トランザクションインプットのうちの1つの中にS I G H A S Hフラグ「S i g H a s h \_ A n y o n e C a n P a y」を有する。これは、誰もがトランザクションのインプットに貢献することを許容する。このパラメータは、エージェントのICTA関数が、複数回、異なるインプットで、実行され且つ繰り返されることを可能にする。この関数の使用は、例えば再利用トランザクションの複製により、認可パーティに制限できる。

#### 【0175】

ICTAコードブロックの「If condition」部分は、任意の種類の条件を監視できる。これは、他のプログラミング言語（例えば、C、C++、J a v e）と同様であり、ブロックチェーンに格納された情報に限定されない。例えば、これは、日付及び時間（つまり、特定の日時に達したとき）を監視し、又は天気（つまり、気温が10℃より低く且つ雨

10

20

30

40

50

が降っているとき)を監視し、取引又は信用の条件(つまり、企業Aが企業Bを買うとき)を監視し得る。

【0176】

ICTAコードブロックの「Then action」部分は、多数の動作を実行できる。本発明は、取り得る動作の数又は種類に関して限定されない。動作は、ブロックチェーン上のトランザクションに限定されないが、動作に関連するメタデータを含むトランザクションは、ブロックチェーンに書き込まれて良い。

【0177】

メタデータは、任意の形式であり得る。しかしながら、一実施形態では、メタデータは、動作に関連するより多くのデータ又は指示を含むファイルへのハイパーリンクを格納して良い。メタデータは、動作に関連するより多くのデータ又は指示を含むハッシュテーブルへのハイパーリンク、及びハッシュテーブルに対する検索キーとして作用する動作のハッシュの両方を格納して良い。

10

【0178】

エージェントの制御スタックは、各ユーザの必要に特化した多数の方法で実装できる。例えば、制御スタックの繰り返しループは、任意のチューリング完全言語に基づき得る。言語の1つの可能な選択は、Forth型スタックに基づく言語である。この言語を使用する利点は、制御スタックを、既知であり且つ広く使用されているビットコインスクリプトとプログラミング型において一貫性を保つことである。

【0179】

<ビットコインスクリプトの交互形式スタック(Alternate Stack)をデータ記憶空間として使用する>

20

ビットコインスクリプトは、コマンド、更に呼び出されるオペコード、を含む。これらは、ユーザが、データを「alt stack」として知られる交互形式スタックに移動させることを可能にする。

【0180】

オペコードは、次の通りである。

- ・OP\_TOALTSTACK。これは、データを主スタックの最上部からalt stackの最上部に移動させる。
- ・OP\_FROMALTSTACK。これは、データをalt stackの最上部から主スタックの最上部に移動させる。

30

【0181】

これは、計算機にデータを格納させる「記憶(memory)」機能と同様に、中間計算ステップからのデータをalt stackに格納させる。一実施形態では、alt stackは、小さな計算タスクを解決するようビットコインスクリプトを構成するために、及び結果を計算に返すために、使用される。

【0182】

<エージェントを管理するためにコードレジスタを使用する>

エージェントは、また、自身の所有し且つ実行する全てのコードのレジストリを管理する。このレジストリは、特定キーを特定値にマッピングするルックアップテーブル又は辞書のように構造化される。キー及び値ペアは、それぞれ、コードブロックのハッシュ(H<sub>1</sub>)及びコードが格納された場所のIPv6アドレスにより表される。キーH<sub>1</sub>を用いてコードブロックを読み出すために、ルックアップテーブルが使用されて、関連する値(これは、コードの格納されている場所である)を読み出し、及びそれに応じてソースコードを読み出す。コードレジストリの実装は変化し得る。

40

【0183】

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

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

【0184】

50

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

【0185】

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

【0186】

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

【0187】

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

1. エージェントは、コードレジストリ内のエントリに一致するコードブロックのハッシュを含むトランザクションについて、ブロックチェーンを監視する。
2. エージェントは、対応するハッシュ ( $H_1$ ) を含むトランザクションを見付ける。
3. エージェントは、「メタデータ-CodeHash」を読み出し、 $H_1$  を得るために CodeHash フィールドを得て、 $H_1$  を用いてコード ( $C_1$ ) を読み出す。RIPEMD-160 (SHA256 ( $C_1$ )) が  $H_1$  に等しい場合、コードは変化しておらず、安全に次のステップに進める。
4. エージェントは、インデックス  $i$  を格納する「メタデータ-CodeHash」を読み出し、 $i$  番目の反復でコードを再スポーニングする。言い換えると、ループが、適切な反復で「リロード」される。
5. メタデータの発生元を検証するために、ユーザの署名が P2SH コマンドに含まれる。
6. これらのデータがループのこの反復のために必要な場合、エージェントは、「メタデータ-OutputHash」及び「メタデータ-OutputPointer」を読み出し、前のステップのアウトプットを検索する。

【0188】

留意すべきことに、上述の実施形態は、本発明を限定するのではなく、当業者は添付の請求項により定められる本発明の範囲から逸脱することなく多数の代替の実施形態を考案できる。請求項中、括弧内に記載された如何なる参照符号も、請求項を制限すると見なされるべきではない。用語「有する (comprising又はcomprises)」等は、全体としていかなる請求項中に及び明細書に列挙された以外の要素又はステップの存在を排除するものではない。本願明細書において、「有する (comprises)」は「含む (includes) 又は構成される (consists of)」を意味し、「有する (comprising)」は「含む (including)

10

20

30

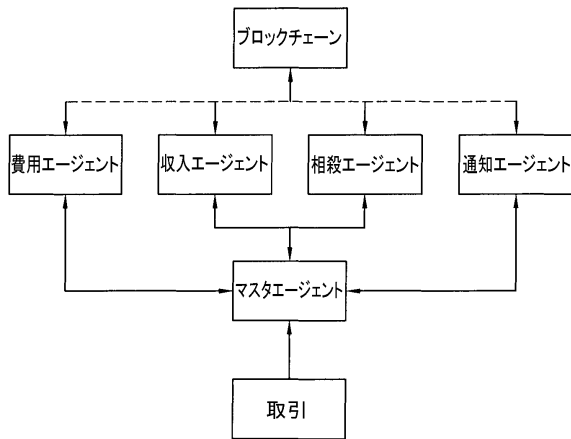
40

50

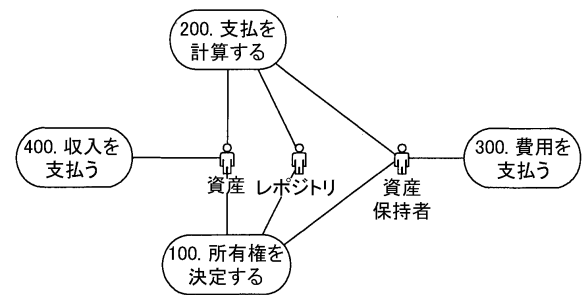
）又は構成される（including of）」を意味する。要素の単数の参照は、該要素の複数の存在を排除するものではなく、逆も同様である。本発明は、複数の別個の要素を有するハードウェアにより又は適切にプログラムされたコンピュータにより、実施され得る。複数の手段を列挙している装置の請求項では、これらの複数の手段は、1つの同一のハードウェア要素により実装することができる。特定の量が相互に異なる従属請求項に記載されるという事実は、これらの量の組合せが有利に用いることが出来ないことを示すものではない。

【図面】

【図 1】



【図 2】



【図 3】

| 比較用：一般取引メタデータフォーマット |                   |     |   |                                                          |
|---------------------|-------------------|-----|---|----------------------------------------------------------|
| フィールド               | サブフィールド           | バイト | 値 | コメント                                                     |
| Metadata1           | ContractType      | 4   |   | 符号化値は取引の種類を示す                                            |
|                     | ContractPointer   | 16  |   | 実際の取引ファイル位置のIPv6アドレス                                     |
|                     | ContractTypeData1 | 12  |   | フォーマットはContractTypeに依存する。ゼロでパディングされる。                    |
| Metadata2           | ContractHash      | 20  |   | RIPEMD-160(SHA256(ContractPointerによりアドレス指定された実際の取引ファイル)) |
|                     | Jurisdiction      | 2   |   | メタデータをカバーする法域                                            |
|                     | ContractTypeData2 | 10  |   | フォーマットはContractTypeに依存する。ゼロでパディングされる。                    |

【図 4】

| シナリオ定義     |                 |     |                                 |                                           |
|------------|-----------------|-----|---------------------------------|-------------------------------------------|
|            |                 |     |                                 | Racehorse資産を生成し、賞金を支払い、費用を集める             |
| 企業メタデータ    |                 |     |                                 |                                           |
| フィールド      | サブフィールド         | バイト | 値                               | コメント                                      |
| Asset      | ContractType    | 4   | 0x0000FF04                      | 単位を示す                                     |
|            | ContractPointer | 16  | xxxx.xxxx.xxxx.xxxx [...] .xxxx | 資産定義(AssetDefinition)ファイルのアドレス            |
|            | Padding         | 12  |                                 | 予備                                        |
| Asset      | ContractHash    | 20  | #####                           | 資産定義(AssetDefinition)ファイルのハッシュ(トークン化ではない) |
|            | Jurisdiction    | 2   | EN                              | 資産が英法によりカバーされることを指定する                     |
|            | Options         | 2   | 0x0000                          | オプション指定無し                                 |
| Metadata B | Options         | 2   | 0x0000                          | オプション指定無し                                 |
|            | Padding         | 8   |                                 | 予備                                        |

10

20

30

40

50

【図 5】

| Stableのシード資金供給                                                     |                  |
|--------------------------------------------------------------------|------------------|
| STABLE-S3-T1                                                       | トランザクションID       |
| Version number                                                     | バージョン番号          |
| 1                                                                  | インプット数           |
| <Stable's current Satoshi pool - assume 75000 Satoshi              | 前のトランザクションアウトプット |
| IDX-00                                                             | 第01スラッシュアウトプット   |
| Script length                                                      | スクリプト長           |
| Sig-Stable PubK-Stable                                             | スクリプト署名          |
| Sequence number                                                    | シーケンス番号          |
| 2                                                                  | アウトプット数          |
| 13,000                                                             | アウトプット値          |
| Output script length                                               | アウトプットスクリプト長     |
| OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG | アウトプットスクリプト      |
| 61,000                                                             | アウトプット値          |
| Output script length                                               | アウトプットスクリプト長     |
| OP_DUP OP_HASH160 <PubK-Stable Hash> OP_EQUALVERIFY OP_CHECKSIG    | アウトプットスクリプト      |
| LockTime                                                           | ロック時間            |

【図 6】

| Racehorse格発行                                |              |
|---------------------------------------------|--------------|
| RAHORSES311                                 | トランザクションID   |
| Version number                              | バージョン番号      |
| 1                                           | インプット数       |
| Stable311                                   | 競馬場のアドレス     |
| 0x-00                                       | 競馬場のアドレス     |
| Script length                               | スクリプト長       |
| Big-Racehorse PK-Racehorse                  | スクリプト署名      |
| Sequence number                             | シーケンス番号      |
| 4                                           | アウトプット数      |
| 5,000                                       | アウトプット値      |
| Output script length                        | アウトプットスクリプト長 |
| >>>OP_HUSHED<<< redeem script hash OP_EQUAL | アウトプットスクリプト値 |
| 3,000                                       | アウトプット値      |
| Output script length                        | アウトプットスクリプト長 |
| >>>OP_HUSHED<<< redeem script hash OP_EQUAL | アウトプットスクリプト値 |
| 2,000                                       | アウトプット値      |
| Output script length                        | アウトプットスクリプト長 |
| >>>OP_HUSHED<<< redeem script hash OP_EQUAL | アウトプットスクリプト値 |
| 2,000                                       | アウトプット値      |
| Output script length                        | アウトプットスクリプト長 |
| >>>OP_HUSHED<<< redeem script hash OP_EQUAL | アウトプットスクリプト値 |
| 2,000                                       | アウトプット値      |
| Output script length                        | アウトプットスクリプト長 |
| >>>OP_HUSHED<<< redeem script hash OP_EQUAL | アウトプットスクリプト値 |
| Use time                                    | ロック時間        |

【圖 7】

|                                                                    |                       |
|--------------------------------------------------------------------|-----------------------|
| AliceからのStable料金の支払い                                               |                       |
| RACEHORSE-S3-T2                                                    | トランザクションID            |
| Version number                                                     | バージョン番号               |
| 1                                                                  | インプット数                |
| ALICE-S3-T1                                                        | 前のトランザクションアウトプット      |
| IDX-00                                                             | 前のランザクションアウトプットインデックス |
| Script length                                                      | スクリプト長                |
| Sig-Alice PubK-Alice                                               | スクリプト署名               |
| Sequence number                                                    | シーケンス番号               |
| 1                                                                  | アウトプット数               |
| 540,000                                                            | アウトプット値               |
| Output script length                                               | アウトプットスクリプト長          |
| OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG | アウトプットスクリプト           |
| SIGHASH_ANYONECANPAY                                               |                       |
| LockTime                                                           | ロック時間                 |
|                                                                    |                       |
| BobからのStable料金の支払い                                                 |                       |
| RACEHORSE-S3-T3                                                    | トランザクションID            |
| Version number                                                     | バージョン番号               |
| 1                                                                  | インプット数                |
| BOB-S3-T1                                                          | 前のトランザクションアウトプット      |
| IDX-00                                                             | 前のランザクションアウトプットインデックス |
| Script length                                                      | スクリプト長                |
| Sig-Bob PubK-Bob                                                   | スクリプト署名               |
| Sequence number                                                    | シーケンス番号               |
| 1                                                                  | アウトプット数               |
| 360,000                                                            | アウトプット値               |
| Output script length                                               | アウトプットスクリプト長          |
| OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG | アウトプットスクリプト           |
| SIGHASH_ANYONECANPAY                                               |                       |
| LockTime                                                           | ロック時間                 |

【圖 8】

| RacehorseのStable料金                                              |                       |
|-----------------------------------------------------------------|-----------------------|
| RACEHORSE-S3-T6                                                 | トランザクションID            |
| Version number                                                  | バージョン番号               |
| 2                                                               | インプット数                |
| RACEHORSE-S3-T2                                                 | 最初のトランザクションアウトポイント    |
| IDX-00                                                          | 最初のトランザクションアウトポイントのパス |
| Script length                                                   | スクリプト長                |
| Sig-Racehorse PubK-Racehorse                                    | スクリプト署名               |
| RACEHORSE-S2-T4                                                 | 最初のトランザクションアウトポイント    |
| IDX-01                                                          | 最初のトランザクションアウトポイントのパス |
| Script length                                                   | スクリプト長                |
| Sig-Racehorse PubK-Racehorse                                    | スクリプト署名               |
| Sequence number                                                 | シーケンス番号               |
| 1                                                               | アウトプット数               |
| 8,999,000                                                       | アウトプット値               |
| Output script length                                            | アウトプットスクリプト長          |
| OP_DUP OP_HASH160 <PubK-Stable Hash> OP_EQUALVERIFY OP_CHECKSIG | アウトプットスクリプト           |
| LockTime                                                        | ロック時間                 |

【図 9】

| Racehorseの賞金                                                       |                        |
|--------------------------------------------------------------------|------------------------|
| STABLE-S3-T2                                                       | トランザクションID             |
| Version number                                                     | バージョン番号                |
| 1                                                                  | インプット数                 |
| EXCHANGE-S3-T1                                                     | 前のトランザクションアウトプット       |
| IDX-00                                                             | 前のトランザクションアウトプットインデックス |
| Script length                                                      | スクリプト長                 |
| Sig-Stable PubK-Stable                                             | スクリプト署名                |
| Sequence number                                                    | シーケンス番号                |
| 1                                                                  | アウトプット数                |
| 8,000,001,000                                                      | アウトプット値                |
| Output script length                                               | アウトプットスクリプト長           |
| OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG | アウトプットスクリプト            |
| LockTime                                                           | ロック時間                  |

【図 10】

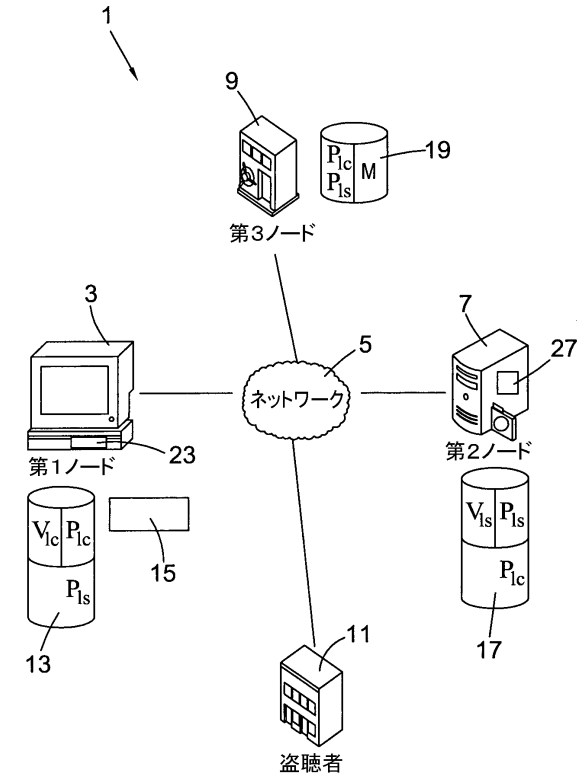
| 投資家へのRacehorseの賞金                                               |                        |
|-----------------------------------------------------------------|------------------------|
| RACEHORSE-S3-T8                                                 | トランザクションID             |
| Version number                                                  | バージョン番号                |
| 1                                                               | インプット数                 |
| STABLE-S3-T2                                                    | 前のトランザクションアウトプット       |
| IDX-00                                                          | 前のトランザクションアウトプットインデックス |
| Script length                                                   | スクリプト長                 |
| Sig-Racehorse PubK-Racehorse                                    | スクリプト署名                |
| Sequence number                                                 | シーケンス番号                |
| 3                                                               | アウトプット数                |
| 4,000,000,000                                                   | アウトプット値                |
| Output script length                                            | アウトプットスクリプト長           |
| OP_DUP OP_HASH160 <PubK-Stable Hash> OP_EQUALVERIFY OP_CHECKSIG | アウトプットスクリプト            |
| 2,400,000,000                                                   | アウトプット値                |
| Output script length                                            | アウトプットスクリプト長           |
| OP_DUP OP_HASH160 <PubK-Alice Hash> OP_EQUALVERIFY OP_CHECKSIG  | アウトプットスクリプト            |
| 1,600,000,000                                                   | アウトプット値                |
| Output script length                                            | アウトプットスクリプト長           |
| OP_DUP OP_HASH160 <PubK-Bob Hash> OP_EQUALVERIFY OP_CHECKSIG    | アウトプットスクリプト            |
| LockTime                                                        | ロック時間                  |

10

【図 11】

| AliceはEveに1株を売る                                                                 |                        |
|---------------------------------------------------------------------------------|------------------------|
| Alice-S3-T2                                                                     | トランザクションID             |
| Version number                                                                  | バージョン番号                |
| 2                                                                               | インプット数                 |
| RACEHORSE-S3-T1                                                                 | 前のトランザクションアウトプット       |
| IDX-01                                                                          | 前のトランザクションアウトプットインデックス |
| Script length                                                                   | スクリプト長                 |
| Sig-Alice OP_1 InvestMetadataA InvestMetadataB PubK-Alice OP_3 OP_CHECKMULTISIG | スクリプト署名                |
| マイニング料金を支払うための第2インプット                                                           |                        |
| <Alice's standard BTC account - assume 1000 satoshis>                           |                        |
| IDX-00                                                                          | 前のトランザクションアウトプットインデックス |
| Script length                                                                   | スクリプト長                 |
| Sig-Alice PubK-Alice                                                            | スクリプト署名                |
| Sequence number                                                                 | シーケンス番号                |
| 2                                                                               | アウトプット数                |
| 2,000                                                                           | アウトプット値                |
| Output script length                                                            | アウトプットスクリプト長           |
| OP_1 InvestMetadataA InvestMetadataB PubK-Alice OP_3 OP_CHECKMULTISIG           | アウトプットスクリプト            |
| 1,000                                                                           | アウトプット値                |
| Output script length                                                            | アウトプットスクリプト長           |
| OP_HASH160 redeem script hash OP_EQUAL                                          | アウトプットスクリプト            |
| Redeem スクリプトはAliceが売り出すことを可能にする                                                 |                        |
| OP_1 InvestMetadataA InvestMetadataB PubK-Alice OP_3 OP_CHECKMULTISIG           |                        |
| Redeem スクリプトはEveが売り出すことを可能にする                                                   |                        |
| OP_1 InvestMetadataA InvestMetadataB PubK-Eve OP_3 OP_CHECKMULTISIG             |                        |
| LockTime                                                                        |                        |

【図 12】



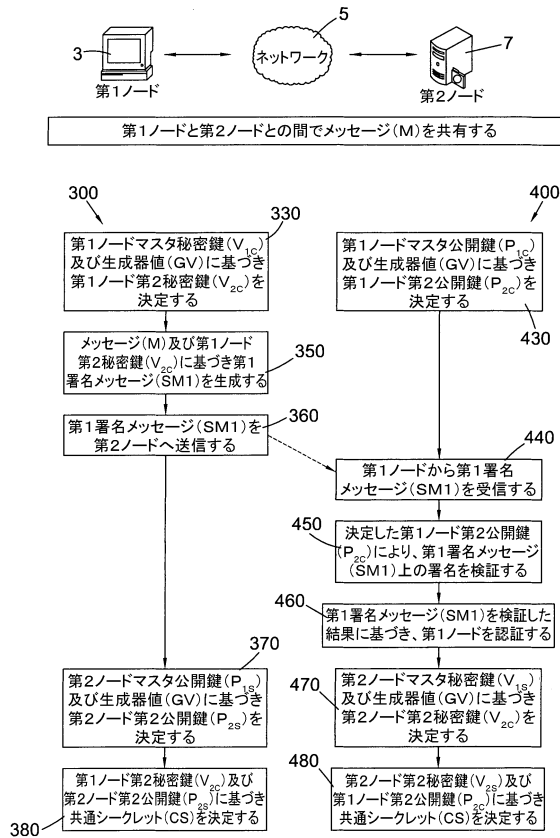
20

30

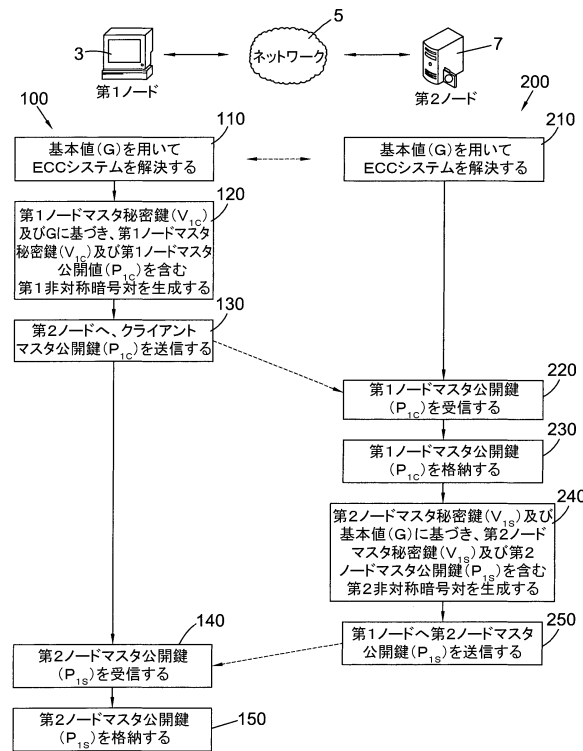
40

50

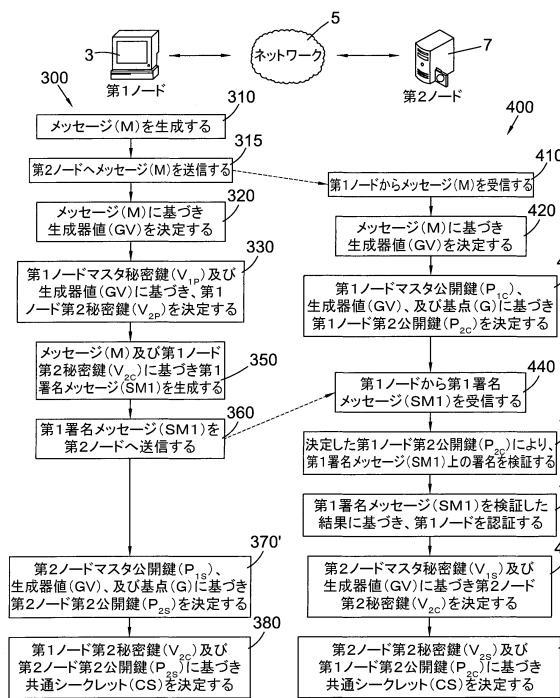
【図 13】



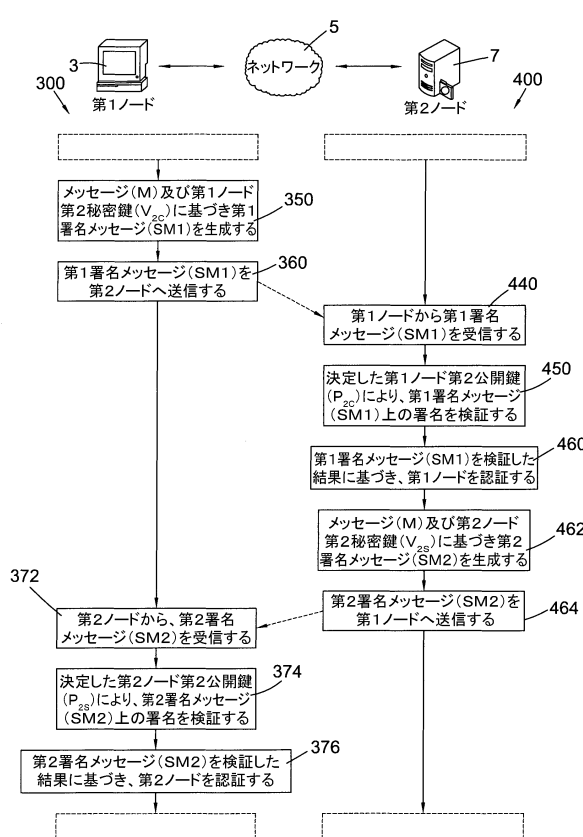
【図 14】



【図 15】



【図 16】



10

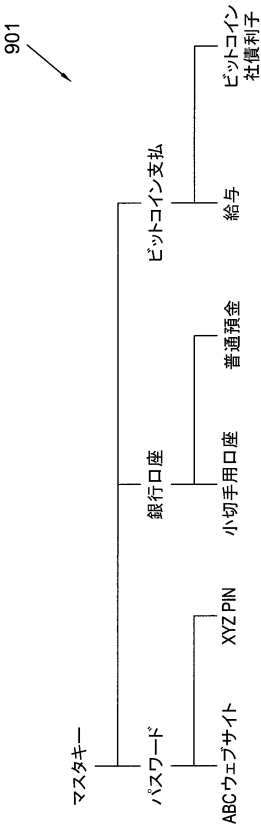
20

30

40

50

【図 1 7】



【図 1 8】

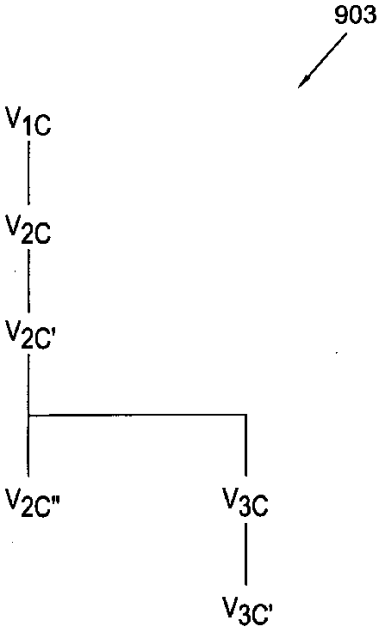


Fig. 18

10

20

30

40

50

## フロントページの続き

(33)優先権主張国・地域又は機関

英国(GB)

(31)優先権主張番号 1603114.8

(32)優先日 平成28年2月23日(2016.2.23)

(33)優先権主張国・地域又は機関

英国(GB)

(31)優先権主張番号 1604225.1

(32)優先日 平成28年3月11日(2016.3.11)

(33)優先権主張国・地域又は機関

英国(GB)

(31)優先権主張番号 1606630.0

(32)優先日 平成28年4月15日(2016.4.15)

(33)優先権主張国・地域又は機関

英国(GB)

(31)優先権主張番号 1619301.3

(32)優先日 平成28年11月15日(2016.11.15)

(33)優先権主張国・地域又は機関

英国(GB)

弁理士 伊東 忠彦

(74)代理人 100091214

弁理士 大貫 進介

(72)発明者 ライト, クレイグ スティーヴン

イギリス国 シーエフ10 2エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス  
7ス フロア アーカートーダイクス アンド ロード エルエルピー 内

(72)発明者 アレン, ガヴィン

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 アーカートーダイクス アンド ロード エルエルピー 内

審査官 青木 重徳

(56)参考文献 特開2013-017197(JP, A)

米国特許出願公開第2015/0262137(US, A1)

米国特許出願公開第2015/0379510(US, A1)

Joseph Poon et al., The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments, [online], 2016年01月14日, <https://lightning.network/lightning-network-paper.pdf>, [2021年3月5日検索]、インターネットAlan Reiner et al., Bitcoin Wallet Identity Verification Specification, [online], 2015年02月27日, <http://diyhp1.us/~bryan/papers2/bitcoin/armory-verisign-bitcoin-wallet-identity-specification.pdf>, [2021年3月5日検索]、インターネット

フォン ヤオカイ ほか, 次世代暗号通貨プラットフォームEthereumの実験的評価, CSS2015 コンピュータセキュリティシンポジウム2015 論文集, 日本, 一般社団法人情報処理学会, 2015年10月14日, Vol. 2015, No. 3, p. 1151-1158

淵田 康之, 特集: イノベーションと金融 ブロックチェーンと金融取引の革新, 野村資本市場クォーターリー, 日本, 株式会社野村資本市場研究所, 2015年11月01日, 第19巻第2号(通巻74号), p. 11-35

(58)調査した分野 (Int.Cl., DB名)

H04L 9/32

G09C 1/00