(19)日本国特許庁(JP)　　(12)公表特許公報(A)　　(11)公表番号
特表2024-536076
(P2024-536076A)
(43)公表日　令和6年10月4日(2024.10.4)

(51)国際特許分類　　FI
H04L 9/32 (2006.01)　　H04L 9/32 200Z

審査請求　未請求　予備審査請求　未請求　（全58頁）

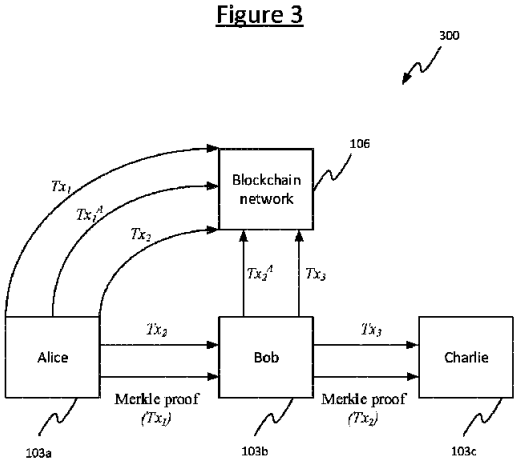
(21)出願番号　特願2024-518593(P2024-518593)
(86)(22)出願日　令和4年8月30日(2022.8.30)
(85)翻訳文提出日　令和6年3月25日(2024.3.25)
(86)国際出願番号　PCT/EP2022/074013
(87)国際公開番号　WO2023/052019
(87)国際公開日　令和5年4月6日(2023.4.6)
(31)優先権主張番号　2113977.9
(32)優先日　令和3年9月30日(2021.9.30)
(33)優先権主張国・地域又は機関
英国(GB)
(81)指定国・地域　AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,
最終頁に続く

(71)出願人　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　100229448
弁理士　中槇 利明
(72)発明者　ペティット，ミカエラ
イギリス ダブリュー１ダブリュー ８エーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内
(72)発明者　コグラン，スティーヴン，パトリック
イギリス ダブリュー１ダブリュー ８エ
最終頁に続く

(54)【発明の名称】　ロックスクリプトの伝搬

(57)【要約】
ブロックチェーントランザクションのチェーン全体にわたって伝搬されるようにロックスクリプトを強制する方法。この方法は、補助トランザクションのロック解除スクリプトによって実行されると、ロック解除スクリプトが補助トランザクションに基づいて生成されたメッセージを含むことを検証し、メッセージに基づいて、補助トランザクションの出力が補助ロックスクリプトを含むことを検証するように構成された一次ロックスクリプトを含む一次トランザクションを生成するステップを含む。補助ロックスクリプトは、第２の一次トランザクションのロック解除スクリプトが第２の一次トランザクションに基づいて生成されたメッセージを含むことを検証し、第２の一次トランザクションのロック解除スクリプトが一次トランザクションの１つまたは複数のロックスクリプトを含むデータ項目を含むことを検証し、メッセージおよびデータ項目に基づいて、第２の一次トランザクションの出力が一次ロックスクリプトを含むことを検証するように構成される。

Figure 3



【特許請求の範囲】
【請求項１】
ブロックチェーントランザクションのチェーン全体にわたって伝搬されるようにロックスクリプトを強制するコンピュータ実装方法であって、前記方法は、第１の当事者によって実行され、
第１の出力を含む第１の一次トランザクションを生成するステップであって、前記第１の出力は、一次ロックスクリプトを含み、前記一次ロックスクリプトは、第１の補助トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）前記第１のロック解除スクリプトが、前記第１の補助トランザクションに基づいて生成された第１のメッセージを含むことを検証し、ｉｉ）前記第１のメッセージに基づいて、前記第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを検証するように構成され、前記補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）前記第２の一次トランザクションの前記第１のロック解除スクリプトが、前記第２の一次トランザクションに基づいて生成された第２のメッセージを含むことを検証し、ｉｉ）前記第２の一次トランザクションの前記第１のロック解除スクリプトが、前記第１の一次トランザクションの１つまたは複数のロックスクリプトを含む第１のデータ項目を含むことを検証し、ｉｉｉ）前記第２のメッセージおよび前記第１のデータ項目に基づいて、前記第２の一次トランザクションの第１の出力が前記一次ロックスクリプトを含むことを検証するように構成される、ステップを含む方法。
【請求項２】
前記第１の補助トランザクションを生成するステップであって、前記第１の補助トランザクションは、前記第１の一次トランザクションの前記第１の出力を参照し、前記第１のメッセージを含む第１の入力と、前記補助ロックスクリプトを含む第１の出力とを含む、ステップ
を含む、請求項１に記載の方法。
【請求項３】
前記一次ロックスクリプトは、前記補助ロックスクリプトのハッシュを含み、前記第１のメッセージは、候補補助ロックスクリプトを含み、前記第１の補助トランザクションの前記第１の出力が前記補助ロックスクリプトを含むことを前記検証することは、前記一次ロックスクリプトが前記候補補助ロックスクリプトをハッシュし、前記候補補助ロックスクリプトのハッシュが前記補助ロックスクリプトの前記ハッシュと一致することを検証することを含む、請求項１に記載の方法。
【請求項４】
前記第１の一次トランザクションは第２の出力を含み、前記補助ロックスクリプトは、前記第２の一次トランザクションの前記第１のロック解除スクリプトによって実行されると、ｉｉｉ）前記第１の一次トランザクションの前記第２の出力が前記第２の一次トランザクションの第２の入力によってロック解除されることを検証するように構成される、請求項１に記載の方法。
【請求項５】
前記補助ロックスクリプトは、前記第２の一次トランザクションの前記第１のロック解除スクリプトによって実行されると、ｉｖ）前記第２の一次トランザクションの前記第１のロック解除スクリプトが、前記第１の補助トランザクションを含む第２のデータ項目を含むことを検証するように構成され、前記第２の出力が前記第２の一次トランザクションの前記第２の入力によってロック解除されることを前記検証することは、ａ）前記第２のメッセージから、前記第２の一次トランザクションの前記第２の入力によってロック解除された前記第２の出力のそれぞれのトランザクション識別子を抽出することと、ｂ）前記第２のデータ項目から、前記第１の補助トランザクションの第１の入力によってロック解除された前記第１の出力のそれぞれのトランザクション識別子を抽出することと、ｃ）ステップａ）の間に抽出された前記それぞれのトランザクション識別子が、ステップｂ）の間に抽出された前記それぞれのトランザクション識別子と一致することを検証することと

を含む、請求項４に記載の方法。
【請求項６】
前記第１の一次トランザクションは、ハッシュ関数を前記第１の一次トランザクションに適用することによって生成するそれぞれのトランザクション識別子を有し、前記第２の一次トランザクションの前記第１のロック解除スクリプトは、第３のデータ項目を含み、前記第３のデータ項目は、前記第１の一次トランザクションに適用された前記ハッシュ関数のミッドステートを含み、前記補助ロックスクリプトは、前記第２の一次トランザクションの前記第１のロック解除スクリプトによって実行されると、ｉｖ）前記第１のデータ項目および前記第３のデータ項目に基づいて候補トランザクション識別子を生成し、前記候補トランザクション識別子が前記第１の一次トランザクションの前記それぞれのトランザクション識別子と一致することを検証するように構成される、請求項１に記載の方法。
【請求項７】
前記補助ロックスクリプトは、前記第２の一次トランザクションの前記第１のロック解除スクリプトによって実行されると、ｖ）前記第２の一次トランザクションが特定のトランザクション識別子を含むことを検証するように構成される、請求項１に記載の方法。
【請求項８】
前記第２の一次トランザクションを生成するステップであって、前記第２の一次トランザクションは、前記第１の補助トランザクションの前記第１の出力を参照し、前記第２のメッセージおよび前記第１のデータ項目を含む第１の入力を含み、前記第２の一次トランザクションは、前記一次ロックスクリプトを含む第１の出力を含む、ステップ
を含む、請求項１に記載の方法。
【請求項９】
前記第１の補助トランザクションおよび前記第２の一次トランザクションを第２の当事者に送信するステップを含む、請求項１に記載の方法。
【請求項１０】
前記一次ロックスクリプトおよび／または前記補助ロックスクリプトは、トークンの所有権に関するデータを含む、請求項１に記載の方法。
【請求項１１】
前記第１の一次トランザクションは、前記ブロックチェーントランザクションのチェーンにおける最初の一次トランザクションである、請求項１に記載の方法。
【請求項１２】
前記特定のトランザクション識別子は、前記第１の一次トランザクションのそれぞれのトランザクション識別子である、請求項７に記載の方法。
【請求項１３】
前記第１の一次トランザクションは第１のカウンタ値を含み、前記補助ロックスクリプトは、前記第２の一次トランザクションの前記第１のロック解除スクリプトによって実行されると、前記第２の一次トランザクションが第２のカウンタ値を含むことを検証するように構成され、前記第２のカウンタ値は前記カウンタ値の反復である、請求項１に記載の方法。
【請求項１４】
前記第１の一次トランザクションは、前記ブロックチェーントランザクションのチェーンにおける初期トランザクションではなく、前記第１の一次トランザクションは、前記補助ロックスクリプトを含む前の補助トランザクションの第１の出力を参照する第１の入力を含み、前記前の補助トランザクションは、前記一次ロックスクリプトを含む前の一次トランザクションの第１の出力を参照する第１の入力を含み、前記補助ロックスクリプトは、前記第２の一次トランザクションの前記第１のロック解除スクリプトによって実行されると、ｖｉ）前記第２の一次トランザクションの前記第１のロック解除スクリプトが、前記前の一次トランザクションの１つまたは複数のロックスクリプトを含む第４のデータ項目を含むことを検証し、ｖｉｉ）前記第２のメッセージおよび前記第４のデータ項目に基づいて、前記第２の一次トランザクションの前記第１の出力が前記一次ロックスクリプト

を含むことを検証するように構成される、請求項１に記載の方法。
【請求項１５】
前記前の一次トランザクションは、前記前の一次トランザクションにハッシュ関数を適用することによって生成するそれぞれのトランザクション識別子を有し、前記第２の一次トランザクションの前記第１のロック解除スクリプトは、第５のデータ項目を含み、前記第５のデータ項目は、前記前の一次トランザクションに適用された前記ハッシュ関数のミッドステートを含み、前記補助ロックスクリプトは、前記第２の一次トランザクションの前記第１のロック解除スクリプトによって実行されると、ｖｉｉｉ）前記第４のデータ項目および前記第５のデータ項目に基づいて候補トランザクション識別子を生成し、前記候補トランザクション識別子が前記前の一次トランザクションの前記それぞれのトランザクション識別子と一致することを検証するように構成される、請求項１４に記載の方法。
【請求項１６】
前記前の一次トランザクションは、前記ブロックチェーントランザクションのチェーンにおける最初の一次トランザクションであり、前記方法は、前記最初の一次トランザクションが所定の公開鍵に対応する署名で署名されたことを検証するステップを含む、請求項１４に記載の方法。
【請求項１７】
ロックスクリプトがブロックチェーントランザクションのチェーン全体にわたって強制されたことを検証するコンピュータ実装方法であって、前記方法は、第２の当事者によって実行され、
第２の一次トランザクションを取得するステップであって、前記第２の一次トランザクションは、第１の入力および第１の出力を含み、前記第１の入力は、前記第２の一次トランザクションに基づいて生成された第２のメッセージを含み、第１の補助トランザクションは、第１の一次トランザクションの第１の出力をロック解除する第１の入力を含み、第１のデータ項目は、前記第１の一次トランザクションの１つまたは複数のロックスクリプトを含み、前記第１の一次トランザクションの前記第１の出力は、一次ロックスクリプトを含み、前記第１の補助トランザクションの第１の出力は、補助ロックスクリプトを含み、前記第２の一次トランザクションの前記第１の出力は、前記一次ロックスクリプトを含む、ステップと、
第１の一次トランザクションがブロックチェーンに記録されたことを検証するステップと、
前記第１の一次トランザクションの前記一次ロックスクリプトが、前記第１の補助トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）前記第１のロック解除スクリプトが、前記第１の補助トランザクションに基づいて生成された第１のメッセージを含むことを検証し、ｉｉ）前記第１のメッセージに基づいて、前記第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを検証するように構成されることを検証するステップであって、前記補助ロックスクリプトは、前記第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）前記第１のロック解除スクリプトが前記第２のメッセージを含むことを検証し、ｉｉ）前記第１のロック解除スクリプトが前記第１のデータ項目を含むことを検証し、ｉｉｉ）前記第２のメッセージおよび前記第１のデータ項目に基づいて、前記第２の一次トランザクションの前記第１の出力が前記一次ロックスクリプトを含むことを検証するように構成され、それによって、前記一次ロックスクリプトが前記第１の一次トランザクションおよび前記第２の一次トランザクション全体にわたって強制されたことを検証する、ステップと
を含む方法。
【請求項１８】
前記一次ロックスクリプトは、前記補助ロックスクリプトのハッシュを含み、前記第１のメッセージは、候補補助ロックスクリプトを含み、前記一次ロックスクリプトは、前記第１の補助トランザクションの前記第１のロック解除スクリプトによって実行されると、前記候補補助ロックスクリプトをハッシュし、前記候補補助ロックスクリプトのハッシュ

が前記補助ロックスクリプトのハッシュと一致することを検証することによって、前記第１の補助トランザクションの前記第１の出力が前記補助ロックスクリプトを含むことを検証するように構成され、前記一次ロックスクリプトがステップｉ）およびｉｉ）を実行するように構成されることを検証する前記ステップは、ハッシュパズルの一部として前記一次ロックスクリプトが前記補助ロックスクリプトの前記ハッシュを含むことを検証するステップを含む、請求項１７に記載の方法。

【請求項１９】

コンピュータ機器であって、

１つまたは複数のメモリユニットを備えるメモリと、

１つまたは複数の処理ユニットを備える処理装置であって、前記メモリは、前記処理装置上で実行されるように構成されたコードを記憶し、前記コードは、前記処理装置上にあるときに、請求項１から１０のいずれか一項に記載の方法を実行するように構成される、処理装置と

を備えるコンピュータ機器。

【請求項２０】

コンピュータ可読ストレージ上に具現化され、１つまたは複数のプロセッサ上で実行されると、請求項１から１８のいずれか一項に記載の方法を実行するように構成されたコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本開示は、ブロックチェーントランザクションのチェーン全体に伝搬されるようにロックスクリプトを強制（enforce）する方法、およびブロックチェーントランザクションのチェーン全体にロックスクリプトが強制されたことを検証する方法に関する。

【背景技術】

【０００２】

ブロックチェーンは、分散型データ構造の形態を指し、ブロックチェーンの複製コピーが、分散型ピアツーピア（Ｐ２Ｐ）ネットワーク（以下、「ブロックチェーンネットワーク」と呼ぶ）内の複数のノードの各々において維持され、広く公開されている。ブロックチェーンはデータブロックのチェーンを含み、各ブロックは１つまたは複数のトランザクションを含む。いわゆる「コインベーストランザクション」以外の各トランザクションは、１つまたは複数のコインベーストランザクションまで遡る１つまたは複数のブロックにまたがり得るシーケンス内の先行するトランザクションを指し示す。コインベーストランザクションについては以下でさらに説明する。ブロックチェーンネットワークにサブミットされたトランザクションは、新しいブロックに含まれる。新しいブロックは、「マイニング（mining）」と呼ばれることが多いプロセスによって作成され、このプロセスは、複数のノードの各々が「プルーフオブワーク（proof-of-work）」を実行しようと競うこと、すなわち、ブロックチェーンの新しいブロックに含まれるのを待っている、順序付けられ妥当性確認（validate）された保留中のトランザクションのプールに基づいて暗号パズルを解くことを伴う。ブロックチェーンはいくつかのノードでプルーニング（prune）され得、ブロックの公開は、単なるブロックヘッダの公開を通して達成され得ることに留意されたい。

【０００３】

ブロックチェーン内のトランザクションは、デジタル資産（すなわち、いくつかのデジタルトークン）を伝達すること、仮想台帳またはレジストリにおけるエントリのセットを順序付けること、タイムスタンプエントリを受信および処理すること、および／またはインデックスポインタを時間順にすることという目的のうちの１つまたは複数のために使用され得る。ブロックチェーンは、ブロックチェーンの上に追加の機能を重ねるために活用することもできる。例えば、ブロックチェーンプロトコルは、トランザクションにおける追加のユーザデータまたはデータへのインデックスの記憶を可能にし得る。単一のトラン

ザクション内に記憶することができる最大データ容量に対してあらかじめ指定された制限はなく、したがって、より複雑なデータを組み込むことができる。例えば、これを使用して、ブロックチェーンに電子文書を記憶したり、オーディオまたはビデオデータを記憶したりすることができる。

【０００４】

ブロックチェーンネットワークのノード（「マイナー（miner）」と呼ばれることが多い）は、以下でより詳細に説明する分散型のトランザクション登録および検証プロセスを実行する。要するに、このプロセスの間に、ノードは、トランザクションを妥当性確認し、有効なプルーフオブワークの解の識別を試みるブロックテンプレートにそれらを挿入する。有効な解が見つかると、ネットワークの他のノードに新しいブロックが伝搬され、これにより、各ノードが、新しいブロックをブロックチェーンに記録することができる。トランザクションをブロックチェーンに記録させるために、ユーザ（例えば、ブロックチェーンクライアントアプリケーション）は、ネットワークのノードのうちの１つにトランザクションを送信して、伝搬させる。トランザクションを受信したノードは、妥当性確認済みトランザクションを新しいブロックに組み込むプルーフオブワークの解を見つけようと競い合い得る。各ノードは、同じノードプロトコルを強制するように構成され、そのノードプロトコルには、トランザクションが有効であるための１つまたは複数の条件が含まれる。無効なトランザクションは、伝搬もブロックへの組込みもされない。トランザクションが妥当性確認され、それによってブロックチェーン上に受け入れられたと仮定すると、トランザクション（任意のユーザデータを含む）は、不変の公開記録としてブロックチェーンネットワーク内のノードの各々において登録およびインデックス付けされたままになる。

【０００５】

プルーフオブワークパズルを解くのに成功して最新のブロックを作成したノードは、典型的には、ある額のデジタル資産、すなわちいくつかのトークンを配布する「コインベーストランザクション」と呼ばれる新しいトランザクションで報酬が与えられる。無効なトランザクションの検出および拒否は、ネットワークのエージェントとして働き、不正を報告および阻止するようにインセンティブが与えられる競合ノードのアクションによって強制される。情報の広範な公開により、ユーザは、ノードの性能を連続的に監査することができる。単なるブロックヘッダの公開により、参加者は、ブロックチェーンの継続的な完全性を確実にすることができる。

【０００６】

「出力ベースの」モデル（ＵＴＸＯベースモデルと呼ばれることもある）では、所与のトランザクションのデータ構造は、１つまたは複数の入力と１つまたは複数の出力とを含む。任意の使用可能な出力は、トランザクションの先行するシーケンスから導出可能なデジタル資産の額を指定する要素を含む。使用可能な出力は、ＵＴＸＯ（「未使用トランザクション出力」）と呼ばれることがある。出力は、出力を将来償還するための条件を指定するロックスクリプトをさらに含み得る。ロックスクリプトは、デジタルトークンまたは資産を妥当性確認および転送するために必要な条件を定義する述語（predicate）である。トランザクション（コインベーストランザクション以外）の各入力は、先行するトランザクションにおけるそのような出力へのポインタ（すなわち、参照）を含み、指し示された出力のロックスクリプトをロック解除するためのロック解除スクリプトをさらに含み得る。そのため、トランザクションのペアを考慮して、それらを第１のトランザクションおよび第２のトランザクション（または「ターゲット」トランザクション）と呼ぶ。第１のトランザクションは、デジタル資産の額を指定する少なくとも１つの出力を含み、出力をロック解除する１つまたは複数の条件を定義するロックスクリプトを含む。第２のターゲットトランザクションは、第１のトランザクションの出力へのポインタを含む少なくとも１つの入力と、第１のトランザクションの出力をロック解除するためのロック解除スクリプトとを含む。

【０００７】

そのようなモデルでは、第２のターゲットトランザクションがブロックチェーンネットワークに送信されて伝搬されブロックチェーンに記録されるとき、各ノードで適用される有効性の基準のうちの１つは、ロック解除スクリプトが第１のトランザクションのロックスクリプトで定義された１つまたは複数の条件のすべてを満たすことである。もう１つは、第１のトランザクションの出力が別の先の有効なトランザクションによってまだ償還されていないということである。これらの条件のいずれかにしたがってターゲットトランザクションが無効であることを発見したノードは、（無効なトランザクションを登録するために伝搬する可能性はあるが、有効なトランザクションとしては）それを伝搬することも、ブロックチェーンに記録されるように新しいブロックにそれを含めることもしない。

【０００８】

トランザクションモデルの別のタイプは、アカウントベースのモデルである。このケースでは、各トランザクションは、過去のトランザクションのシーケンスにおける先行するトランザクションのＵＴＸＯを参照することによってではなく、絶対アカウント残高を参照することによって転送されるべき額を定義する。すべてのアカウントの現在の状態は、ブロックチェーンとは別個にノードによって記憶され、絶えず更新される。

【発明の概要】

【０００９】

支出トランザクション、すなわち、前のトランザクションの出力の使用（ロック解除、割り当て、転送など）を行うトランザクションのフィールドに対して条件を適用（ｅｎｆｏｒｃｅ）できることが一般に知られている。例えば、前のトランザクションは、支出トランザクションの１つまたは複数の出力に条件を課すロックスクリプトを含み得る。「支出トランザクション」という用語は、少なくとも１つの前のトランザクションの少なくとも１つの出力をロック解除する現在のトランザクションを意味するために当技術分野で使用され、現在のトランザクションが金融トランザクションに関連することを必ずしも意味しないことに留意されたい。

【００１０】

支出トランザクションの分野に条件を適用することを望む理由の１つは、支出トランザクションが前のトランザクションと同じロックスクリプトを含む出力を有することを確実にすることである。こうすることで、支出トランザクションが次の支出トランザクションに対して同じ条件を適用することを確実にすることができる。すなわち、ｎ－１番目のトランザクションは、ｎ番目のトランザクションに強制的に同じロックスクリプトを含ませるロックスクリプトを含み、したがって、ｎ＋１番目のトランザクションに強制的に同じロックスクリプトを含ませる。このようにして、トランザクションのチェーンが作成され、それによって、各トランザクションに同じロックスクリプトが含まれる。これは、例えば、現実世界オブジェクト、または仮想世界内のオブジェクトの所有権を表すデジタルトークンの文脈で使用され得る。これは、トークンの各転送が同じ規則に従うことを意味するので、有利である。

【００１１】

トランザクションのチェーンに強制的に同じロックスクリプトを含ませる以前の試みは、以下の問題のうちの少なくとも１つに悩まされる。第１に、試みの中には、第１のトランザクションが（例えば、トークン発行者などの特定の権限者によって）正しく作成されたこと、または直近のトランザクションに含まれるロックスクリプトが、チェーン内の第１のトランザクションおよび他のすべてのトランザクションに含まれていたロックスクリプトと同じであることを確実にするために、検証者がチェーン内の第１のトランザクション（例えば、「発行トランザクション」）まで遡ることを必要とするものがある。これは、チェーン内の最新のトランザクションを妥当性確認する際に検証者のリソース（計算リソース、時間、および労力の両方に関して）を消費する。第２に、試みの中には、次のトランザクションが正しく作成されることを確実にするために、第三者がチェーン内の次のトランザクション（例えば、トークンの転送）の作成および／または妥当性確認に関与することを必要とするものがある。これは、第三者に依存することによってセキュリティ上

の脆弱性をもたらし、互いに対話しなければならない当事者の数を増加させ、効率性を低下させる。第三に、トランザクションのチェーンに強制的に同じロックスクリプトを含ませる試みの中には、チェーン内の最新のトランザクションがチェーン内の前の各トランザクションを含むことを必要とするものがあり、これは、トランザクション肥大化問題を発生させる。これは、トランザクションの送信および記憶の両方にとって問題である。第四に、試みの中には、以下に説明されるリプレイ攻撃の影響を受けやすいものもある。

【００１２】

したがって、識別された問題のうちの１つ、いくつか、またはすべてを解決する同じロックスクリプトを含むようにトランザクションのチェーンを強制する機構を実装することが望ましい。

【００１３】

本明細書に開示される一態様によれば、ブロックチェーントランザクションのチェーン全体にわたって伝搬されるようにロックスクリプトを強制するコンピュータ実装方法が提供される。この方法は、第１の当事者によって実行され、第１の出力を含む第１の一次トランザクションを生成するステップであって、第１の出力は、一次ロックスクリプトを含み、一次ロックスクリプトは、第１の補助トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第１のロック解除スクリプトが、第１の補助トランザクションに基づいて生成された第１のメッセージを含むことを検証し、ｉｉ）第１のメッセージに基づいて、第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを検証するように構成され、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第２の一次トランザクションの第１のロック解除スクリプトが、第２の一次トランザクションに基づいて生成された第２のメッセージを含むことを検証し、ｉｉ）第２の一次トランザクションの第１のロック解除スクリプトが、第１の一次トランザクションの１つまたは複数のロックスクリプトを含む第１のデータ項目を含むことを検証し、ｉｉｉ）第２のメッセージおよび第１のデータ項目に基づいて、第２の一次トランザクションの第１の出力が一次ロックスクリプトを含むことを検証するように構成される、ステップを含む。

【００１４】

「一次トランザクション」という用語は、本明細書では、「一次ロックスクリプト」と呼ばれる同じロックスクリプトを含むように強制されたトランザクションのチェーン内のトランザクションを指すために使用される。例えば、一次ロックスクリプトは、所有物の権利書など、トークンに関連するデータを含み得る。「補助トランザクション」という用語は、一次トランザクションに対する条件、すなわち一次トランザクションが一次ロックスクリプトを含まなければならないという条件を適用するために使用されるトランザクションを指すために使用される。これらの条件は、少なくとも部分的に、各補助トランザクションに含まれる「補助ロックスクリプト」によって適用される。

【００１５】

補助トランザクションは、連続する一次トランザクション間のブリッジとして働く。すなわち、第１の補助トランザクションは、第１の一次トランザクションの出力を使用（すなわち、ロック解除）し、第２の一次トランザクションは、第１の補助トランザクションの出力を使用（すなわち、ロック解除）する。一次ロックスクリプトは、次の補助トランザクションに条件を課す。例えば、一次ロックスクリプトは、次の補助トランザクションに強制的に補助ロックスクリプトを含ませる。補助ロックスクリプトは、次の一次トランザクションに条件を課す。例えば、補助ロックスクリプトは、次の一次トランザクションに強制的に一次ロックスクリプトを含ませる。

【００１６】

いくつかの例では、第１の一次トランザクションは、チェーン内の最初の一次トランザクション、すなわち、一次ロックスクリプトを含む最初のトランザクションである。これは、例えば、トークンを発行するトークン発行者によって作成されてもよい。他の例では、第１の一次トランザクションは、最初の一次トランザクションではなく、すなわち、一

次ロックスクリプトを含むトランザクションのチェーン内に少なくとも１つの他の前の一次トランザクションが存在する。例えば、この例では、第１の一次トランザクションは、トークンの転送者によって作成されてもよい。

【００１７】

本明細書に開示される別の態様によれば、ロックスクリプトがブロックチェーントランザクションのチェーン全体にわたって強制されたことを検証するコンピュータ実装方法が提供される。この方法は、第２の当事者によって実行され、第２の一次トランザクションを取得するステップであって、第２の一次トランザクションは、第１の入力および第１の出力を含み、第１の入力は、第２の一次トランザクションに基づいて生成された第２のメッセージを含み、第１の補助トランザクションは、第１の一次トランザクションの第１の出力をロック解除する第１の入力を含み、第１のデータ項目は、第１の一次トランザクションの１つまたは複数のロックスクリプトを含み、第１の一次トランザクションの第１の出力は、一次ロックスクリプトを含み、第１の補助トランザクションの第１の出力は、補助ロックスクリプトを含み、第２の一次トランザクションの第１の出力は、一次ロックスクリプトを含む、ステップを含む。この方法は、第１の一次トランザクションがブロックチェーンに記録されたことを検証するステップと、第１の一次トランザクションの一次ロックスクリプトが、第１の補助トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第１のロック解除スクリプトが、第１の補助トランザクションに基づいて生成された第１のメッセージを含むことを検証し、ｉｉ）第１のメッセージに基づいて、第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを検証するように構成されることを検証するステップであって、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第１のロック解除スクリプトが第２のメッセージを含むことを検証し、ｉｉ）第１のロック解除スクリプトが第１のデータ項目を含むことを検証し、ｉｉｉ）第２のメッセージおよび第１のデータ項目に基づいて、第２の一次トランザクションの第１の出力が一次ロックスクリプトを含むことを検証するように構成され、それによって、一次ロックスクリプトが第１の一次トランザクションおよび第２の一次トランザクション全体にわたって強制されたことを検証する、ステップをさらに含む。

【００１８】

第２の当事者は、第２の一次トランザクションを取得する（例えば、第１の当事者から受信する）。第２の一次トランザクションは、トランザクションのチェーンにおいて２番目に現れる一次トランザクションであってもよいし、後のトランザクションであってもよい。すなわち、第２のトランザクションは、一般に、トランザクションのチェーンにおけるｎ番目のトランザクションであり得る。第２の一次トランザクションは、一次ロックスクリプトを出力に含む。第２の一次トランザクションは、第１の補助トランザクションの出力を使用する入力を有し、したがって、補助ロックスクリプトの要件を満たす。第１の補助トランザクションは、第１の一次トランザクションの出力を使用し、したがって、一次ロックスクリプトの要件を満たす。したがって、第１の補助トランザクションが第１の一次トランザクションの一次ロックスクリプトのロック解除に成功し、第２の一次トランザクションが第１の補助トランザクションの補助ロックスクリプトのロック解除に成功した場合、第２の一次トランザクションは一次ロックスクリプトを必ず含む。これは、正しい一次ロックスクリプトがトランザクションのチェーンに含まれていることに依存する。

【００１９】

第２の当事者は、第１の一次トランザクションがブロックチェーンに記録されており、したがって、有効なトランザクションであることを検証する。第２の当事者はまた、一次ロックスクリプトが正しいこと、例えば、それが補助トランザクションに補助ロックスクリプトを強制的に含ませることを検証し、次いで、次の一次トランザクションに同じ一次ロックスクリプトを強制的に含ませる。例えば、これは、一次ロックスクリプトが補助ロックスクリプトのハッシュに基づくハッシュパズルを含むことを確実にすることによって達成され得る。このケースでは、ハッシュパズルは、補助トランザクションが補助ロック

スクリプトを含む場合にのみ解くことができる（したがって、一次ロックスクリプトはロック解除される）。

【００２０】

次いで、第２の当事者は、第２の補助トランザクションおよび第３の一次トランザクションを生成し得る。第２の補助トランザクションは、第２の一次トランザクションの一次ロックスクリプトをロック解除するために必要である。第２の補助トランザクションはまた、第３の一次トランザクションに一次ロックスクリプトを強制的に含ませる。チェーン内の次の各トランザクションは、同じ方法で一次ロックスクリプトを含むように強制される。

【００２１】

実際には、２つのロックスクリプトが伝搬されるように強制されることに留意されたい。一次ロックスクリプトは、各一次トランザクションの出力に含まれるように強制され、補助ロックスクリプトは、各補助トランザクションの出力に含まれように強制される。

【図面の簡単な説明】

【００２２】

本開示の実施形態の理解を助け、そのような実施形態がどのように実施され得るかを示すために、単なる例として添付の図面を参照する。

【図１】ブロックチェーンを実装するためのシステムの概略ブロック図である。

【図２】ブロックチェーンに記録され得るトランザクションのいくつかの例を概略的に示す。

【図３】ロックスクリプトを伝搬するための例示的なシステムを概略的に示す。

【図４】例示的なトークンプロトコルのトランザクション構造の概略図である。

【図５】１つの入力および１つの出力を有するトランザクションのチェーンの概略図である。

【図６】トランザクション肥大化問題を概略的に示す。

【図７】２つの入力および２つの出力を有するトランザクションのチェーンの概略図である。

【図８】例示的なトークンプロトコルによってトランザクションの肥大化がどのように防止されるかを示す概略図である。

【図９】例示的な発行トランザクションを概略的に示す。

【図１０】例示的な補助トランザクションを概略的に示す。

【図１１】アリスからボブにトークンを転送する例示的な一次トランザクションを概略的に示す。

【図１２】別の補助トランザクションの例を概略的に示す。

【図１３】ボブからチャーリーにトークンを転送する一次トランザクションの例を概略的に示す。

【図１４】ｎ番目の一次トランザクションの例示的なフォーマットを概略的に示す。

【図１５】ｎ番目の補助トランザクションの例示的なフォーマットを概略的に示す。

【発明を実施するための形態】

【００２３】

１．例示的なシステムの概要

図１は、ブロックチェーン１５０を実装するための例示的なシステム１００を示す。システム１００は、典型的にはインターネットなどの広域インターネットワークであるパケット交換ネットワーク１０１で構成され得る。パケット交換ネットワーク１０１は、パケット交換ネットワーク１０１内にピアツーピア（Ｐ２Ｐ）ネットワーク１０６を形成するように構成され得る複数のブロックチェーンノード１０４を含む。図示されていないが、ブロックチェーンノード１０４は、ほぼ完全なグラフとして構成され得る。したがって、各ブロックチェーンノード１０４は、他のブロックチェーンノード１０４に高度に接続される。

【００２４】

各ブロックチェーンノード１０４は、ピアのコンピュータ機器を含み、ノード１０４の異なるものは、異なるピアに属する。各ブロックチェーンノード１０４は、１つまたは複数のプロセッサ、例えば、１つまたは複数の中央処理装置（ＣＰＵ）、アクセラレータプロセッサ、特定用途向けプロセッサおよび／またはフィールドプログラマブルゲートアレイ（ＦＰＧＡ）、ならびに特定用途向け集積回路（ＡＳＩＣ）などの他の機器を含む処理装置を備える。各ノードはまた、メモリ、すなわち、１つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備える。メモリは、１つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ソリッドステートドライブ（ＳＳＤ）、フラッシュメモリもしくはＥＥＰＲＯＭなどの電子媒体、および／または光ディスクドライブなどの光学媒体を採用する１つまたは複数のメモリユニットを備え得る。

【００２５】

ブロックチェーン１５０は、データブロック１５１のチェーンを含み、ブロックチェーン１５０のそれぞれのコピーは、分散型またはブロックチェーンネットワーク１０６内の複数のブロックチェーンノード１０４の各々で維持される。上述したように、ブロックチェーン１５０のコピーを維持することは、ブロックチェーン１５０を完全に記憶することを必ずしも意味しない。代わりに、ブロックチェーン１５０は、各ブロックチェーンノード１５０が各ブロック１５１のブロックヘッダ（後述する）を記憶している限り、データがプルーニングされ得る。チェーン内の各ブロック１５１は、１つまたは複数のトランザクション１５２を含み、この文脈におけるトランザクションは、データ構造の一種を指す。データ構造の性質は、トランザクションモデルまたは方式の一部として使用されるトランザクションプロトコルのタイプに依存する。所与のブロックチェーンは、全体を通して１つの特定のトランザクションプロトコルを使用する。１つの一般的なタイプのトランザクションプロトコルでは、各トランザクション１５２のデータ構造は、少なくとも１つの入力および少なくとも１つの出力を含む。各出力は、プロパティとしてデジタル資産の量を表す額を指定し、その例は、出力が暗号的にロックされている（ロック解除され、それによって償還または使用されるためにはそのユーザの署名または他の解決策を必要とする）ユーザ１０３である。各入力は、先行するトランザクション１５２の出力を指し示し、それによってトランザクションをリンクする。

【００２６】

各ブロック１５１はまた、ブロック１５１への順番を定義するために、チェーン内の前に作成されたブロック１５１を指し示すブロックポインタ１５５を含む。各トランザクション１５２（コインベーストランザクション以外）は、トランザクションのシーケンスへの順序を定義するために、前のトランザクションへ戻るポインタを含む（注意：トランザクション１５２のシーケンスは分岐することが可能である）。ブロック１５１のチェーンは、チェーン内の最初のブロックであったジェネシスブロック（Ｇｂ：genesis block）１５３まで戻る。チェーン１５０内の早期にある１つまたは複数の元のトランザクション１５２は、先行するトランザクションではなくジェネシスブロック１５３を指し示していた。

【００２７】

ブロックチェーンノード１０４の各々は、トランザクション１５２を他のブロックチェーンノード１０４にフォワードし、それによってトランザクション１５２をネットワーク１０６全体に伝搬させるように構成される。各ブロックチェーンノード１０４は、ブロック１５１を作成し、同じブロックチェーン１５０のそれぞれのコピーをそれらのそれぞれのメモリに記憶するように構成される。各ブロックチェーンノード１０４はまた、ブロック１５１に組み込まれるのを待っているトランザクション１５２の順序付きセット（またはプール）１５４を維持する。順序付きプール１５４は、「メムプール（mempool）」と呼ばれることが多い。本明細書におけるこの用語は、任意の特定のブロックチェーン、プロトコル、またはモデルに限定することを意図するものではない。これは、ノード１０４が有効であるとして受け入れたトランザクションの順序付きセットを指し、それに対して、ノード１０４は、同じ出力を使用しようとする他のトランザクションを受け入れない

ように義務付けられている。

【００２８】

所与の現在のトランザクション１５２ｊにおいて、その入力（または各入力）は、トランザクションのシーケンスにおける先行するトランザクション１５２ｉの出力を参照するポインタを含み、この出力が現在のトランザクション１５２ｊにおいて償還または「使用」されるべきであることを指定する。使用することまたは償還することは、必ずしも金融資産の移転を意味するものではないが、それは確かに１つの一般的な用途である。より一般的には、使用することは、出力を消費すること、または別の前方のトランザクションにおける１つまたは複数の出力にそれを割り当てることとして説明され得る。一般に、先行するトランザクションは、順序付きセット１５４または任意のブロック１５１内の任意のトランザクションであり得る。先行するトランザクション１５２ｉは、現在のトランザクションが有効となるためには存在および妥当性確認される必要があるが、先行するトランザクション１５２ｉは、現在のトランザクション１５２ｊが作成されるときまたはネットワーク１０６に送信されるときに必ずしも存在する必要はない。したがって、本明細書における「先行する（preceding）」は、ポインタによってリンクされた論理シーケンスにおける先行するものを指し、必ずしも時間シーケンスにおける作成または送信の時間を指すものではなく、したがって、トランザクション１５２ｉ、１５２ｊが順不同に作成または送信されることを必ずしも除外するものではない（オーファントランザクションに関する以下の説明を参照）。先行するトランザクション１５２ｉは、同様に、先のトランザクション（antecedent transaction）または先行したトランザクション（predecessor transaction）とも呼ばれる。

【００２９】

現在のトランザクション１５２ｊの入力はまた、入力認可、例えば、先行するトランザクション１５２ｉの出力がロックされている対象のユーザ１０３ａの署名を含む。次に、現在のトランザクション１５２ｊの出力は、新しいユーザまたはエンティティ１０３ｂに暗号的にロックされ得る。したがって、現在のトランザクション１５２ｊは、先行するトランザクション１５２ｉの入力において定義された額を、現在のトランザクション１５２ｊの出力において定義されたように、新しいユーザまたはエンティティ１０３ｂに転送することができる。いくつかのケースでは、トランザクション１５２は、複数のユーザまたはエンティティ（そのうちの１つは残り（change）を与えるために元のユーザまたはエンティティ１０３ａであり得る）間で入力額を分割するために複数の出力を有し得る。いくつかのケースでは、トランザクションはまた、１つまたは複数の先行するトランザクションの複数の出力からの額をまとめ、現在のトランザクションの１つまたは複数の出力に再分配するために複数の入力を有することができる。

【００３０】

ビットコインなどの出力ベースのトランザクションプロトコルによれば、個々のユーザまたは組織などの当事者１０３が（手動でまたは当事者によって採用される自動プロセスによって）新しいトランザクション１５２ｊを制定することを望むとき、制定を行う当事者は、新しいトランザクションをそのコンピュータ端末１０２から受信者に送信する。制定を行う当事者または受信者は、最終的に、このトランザクションをネットワーク１０６のブロックチェーンノード１０４の１つまたは複数（これは、現在では、典型的にはサーバまたはデータセンタであるが、原則として他のユーザ端末であってもよい）に送信する。新しいトランザクション１５２ｊを制定する当事者１０３がトランザクションをブロックチェーンノード１０４の１つまたは複数に直接送信し、いくつかの例では、受信者に送信しないことも除外されない。トランザクションを受信するブロックチェーンノード１０４は、ブロックチェーンノード１０４の各々で適用されるブロックチェーンノードプロトコルにしたがって、トランザクションが有効であるかどうかをチェックする。ブロックチェーンノードプロトコルは、典型的には、新しいトランザクション１５２ｊ内の暗号署名が、トランザクション１５２の順序付きシーケンス内で前のトランザクション１５２ｉに依存する予想される署名と一致することをチェックするようにブロックチェーンノード１

０４に要求する。そのような出力ベースのトランザクションプロトコルでは、これは、新しいトランザクション１５２ｊの入力に含まれる当事者１０３の暗号署名または他の認可が、新しいトランザクションが使用する（または「割り当てる」）先行するトランザクション１５２ｉの出力において定義される条件と一致することをチェックすることを含み得、この条件は、典型的には、新しいトランザクション１５２ｊの入力における暗号署名または他の認可が、新しいトランザクションの入力がリンクされている前のトランザクション１５２ｉの出力をロック解除することを少なくともチェックすることを含む。条件は、先行するトランザクション１５２ｉの出力に含まれるスクリプトによって少なくとも部分的に定義され得る。代替的に、それは、単にブロックチェーンノードプロトコルのみによって固定されてもよいし、これらの組合せによるものであってもよい。いずれにしても、新しいトランザクション１５２ｊが有効である場合、ブロックチェーンノード１０４は、それをブロックチェーンネットワーク１０６内の１つまたは複数の他のブロックチェーンノード１０４にフォワードする。これらの他のブロックチェーンノード１０４は、同じブロックチェーンノードプロトコルにしたがって同じテストを適用し、そして、新しいトランザクション１５２ｊを１つまたは複数のさらなるノード１０４にフォワードし、以下同様である。このようにして、新しいトランザクションはブロックチェーンノード１０４のネットワーク全体に伝搬される。

【００３１】

出力ベースのモデルでは、所与の出力（例えば、ＵＴＸＯ）が割り当てられる（または、「使用される」）かどうかの定義は、それがブロックチェーンノードプロトコルにしたがって別の前方のトランザクション１５２ｊの入力によって有効に償還されたかどうかである。トランザクションが有効であるための別の条件は、それが償還しようとする先行するトランザクション１５２ｉの出力が、別のトランザクションによってまだ償還されていないことである。この場合も同様に、有効ではない場合、トランザクション１５２ｊは、（無効としてフラグ付けされ、警告のために伝搬されない限り）伝搬されることも、ブロックチェーン１５０内に記録されることもない。これは、トランザクタ（transactor）が同じトランザクションの出力を複数回割り当てようとする二重支出を防止する。一方、アカウントベースのモデルは、アカウント残高を維持することによって二重支出を防止する。ここでも、トランザクション順序が定義されているので、アカウント残高は常に単一の定義された状態にある。

【００３２】

トランザクションを妥当性確認することに加えて、ブロックチェーンノード１０４はまた、「プルーフオブワーク」によって支持される、一般にマイニングと呼ばれるプロセスにおいてトランザクションのブロックを最初に作成しようと競い合う。ブロックチェーンノード１０４において、新しいトランザクションが、ブロックチェーン１５０上に記録されたブロック１５１内にまだ現れていない有効なトランザクションの順序付きプール１５４に追加される。次いで、ブロックチェーンノードは、暗号パズルを解こうとすることで、トランザクションの順序付きセット１５４からトランザクション１５２の新しい有効なブロック１５１を組み立てようと競い合う。典型的には、これは、ナンスが保留中のトランザクションの順序付きプール１５４の表現と連結されハッシュされたときにハッシュの出力が所定の条件を満たすような「ナンス」値を探索することを含む。例えば、所定の条件とは、ハッシュの出力が特定の所定の数の先行ゼロを有することであり得る。これは、プルーフオブワークパズルの１つの特定のタイプにすぎず、他のタイプが除外されないことに留意されたい。ハッシュ関数のプロパティは、その入力に対して予測不可能な出力を持つことである。したがって、この探索は、総当たりでしか実行することができないので、パズルを解こうとしている各ブロックチェーンノード１０４でかなりの量の処理リソースを消費する。

【００３３】

最初にパズルを解いたブロックチェーンノード１０４は、これをネットワーク１０６に公表し、後にネットワーク内の他のブロックチェーンノード１０４によって容易にチェッ

クすることができるその解をプルーフとして提供する（ハッシュに対する解が与えられると、ハッシュの出力が条件を満たすことをチェックすることは簡単である）。この最初のブロックチェーンノード１０４は、ブロックを、このブロックを受け入れる他のノードのしきい値コンセンサスに伝搬し、プロトコルルールを強制する。次いで、トランザクションの順序付きセット１５４は、ブロックチェーンノード１０４の各々によってブロックチェーン１５０内に新しいブロック１５１として記録されるようになる。ブロックポインタ１５５はまた、チェーン内の前に作成されたブロック１５１ｎ－１を指し示す新しいブロック１５１ｎに割り当てられる。プルーフオブワークの解を作成するために必要とされる、例えばハッシュの形態の、かなりの量の労力は、ブロックチェーンプロトコルのルールに従うという最初のノード１０４の意図を示す。そのようなルールは、前に妥当性確認されたトランザクションと同じ出力の使用または割当てを行った場合にトランザクションを有効として受け入れること（別名二重支出としても知られている）を行わないことを含む。ブロック１５１は、一旦作成されると、ブロックチェーンネットワーク１０６内のブロックチェーンノード１０４の各々において認識および維持されるので、修正することができない。ブロックポインタ１５５はまた、ブロック１５１に順番を付与する。トランザクション１５２は、ネットワーク１０６内の各ブロックチェーンノード１０４において順序付きブロックに記録されるので、これは、トランザクションの不変の公開台帳を提供する。

【００３４】

任意の所与の時間にパズルを解こうと競い合う異なるブロックチェーンノード１０４は、それらがいつ解を探索し始めたかまたはトランザクションが受信された順序に応じて、任意の所与の時間に、まだ公開されていないトランザクションのプール１５４の異なるスナップショットに基づいてそれを行っていてもよいことに留意されたい。誰がそれぞれのパズルを最初に解いても、どのトランザクション１５２が次の新しいブロック１５１ｎにどの順序で含まれるかを定義し、公開されていないトランザクションの現在のプール１５４が更新される。次いで、ブロックチェーンノード１０４は、新たに定義された、公開されていないトランザクションの順序付きプール１５４からブロックを作成しようと競い合い続け、以下同様である。２つのブロックチェーンノード１０４が互いに非常に短い時間内にパズルを解いて、ブロックチェーンの相反する見解がノード１０４間で伝搬される場合に発生し得る任意の「フォーク」を解決するためのプロトコルも存在する。要するに、フォークのどちらのプロングでも最も長く成長した方が、確定的なブロックチェーン１５０となる。同じトランザクションが両方のフォークに現れるので、これがネットワークのユーザまたはエージェントに影響を与えないことに留意されたい。

【００３５】

ビットコインブロックチェーン（およびほとんどの他のブロックチェーン）によれば、新しいブロック１０４の構築に成功したノードには、（あるエージェントまたはユーザから別のエージェントまたはユーザにある額のデジタル資産を転送するエージェント間またはユーザ間のトランザクションとは対照的に）追加の定義された量のデジタル資産を分配する新しい特別な種類のトランザクションにおいて、受け入れられた追加の額のデジタル資産を新たに割り当てる能力が与えられる。この特別なタイプのトランザクションは、通常、「コインベーストランザクション」と呼ばれるが、「開始トランザクション（initiation transaction）」または「生成トランザクジョン（generation transaction）」と称されることもある。これは典型的に、新しいブロック１５１ｎの最初のトランザクションを形成する。プルーフオブワークは、新しいブロックを構築するノードが、この特別なトランザクションが後に償還されることを可能にするプロトコルルールに従うという意図を示す。ブロックチェーンプロトコルルールは、この特別なトランザクションが償還され得る前に、満期期間、例えば１００個のブロックを必要とし得る。多くの場合、通常の（非生成）トランザクション１５２はまた、そのトランザクションが公開されたブロック１５１ｎを作成したブロックチェーンノード１０４にさらに報酬を与えるために、その出力のうちの１つにおいて追加のトランザクション手数料を指定する。この手数料は通

常「トランザクション手数料」と呼ばれ、以下で説明される。

【００３６】

トランザクション妥当性確認および公開に関与するリソースに起因して、典型的には、ブロックチェーンノード１０４の少なくとも各々は、１つまたは複数の物理サーバユニットで構成されるサーバの形態をとるか、さらにはデータセンタ全体の形態をとる。しかしながら、原則として、任意の所与のブロックチェーンノード１０４は、ユーザ端末または互いにネットワーク化されたユーザ端末のグループの形態をとることができる。

【００３７】

各ブロックチェーンノード１０４のメモリは、そのそれぞれの１つまたは複数の役割を実行し、ブロックチェーンノードプロトコルにしたがってトランザクション１５２を処理するために、ブロックチェーンノード１０４の処理装置上で実行されるように構成されたソフトウェアを記憶する。本明細書においてブロックチェーンノード１０４に帰する任意のアクションは、それぞれのコンピュータ機器の処理装置上で実行されるソフトウェアによって実行され得ることが理解されよう。ノードソフトウェアは、アプリケーション層、またはオペレーティングシステム層もしくはプロトコル層などの下位層、またはこれらの任意の組合せにおける１つまたは複数のアプリケーションにおいて実装され得る。

【００３８】

消費ユーザの役割を担う複数の当事者１０３の各々のコンピュータ機器１０２もネットワーク１０１に接続されている。これらのユーザは、ブロックチェーンネットワーク１０６と対話し得るが、トランザクションの妥当性確認にもブロックの構築にも参加しない。これらのユーザまたはエージェント１０３のうちのいくつかは、トランザクションの送信者および受信者として動作し得る。他のユーザは、必ずしも送信者または受信者として動作することなくブロックチェーン１５０と対話し得る。例えば、いくつかの当事者は、ブロックチェーン１５０のコピーを記憶する（例えば、ブロックチェーンノード１０４からブロックチェーンのコピーを取得した）ストレージエンティティとして動作し得る。

【００３９】

当事者１０３のうちのいくつかまたはすべては、異なるネットワーク、例えば、ブロックチェーンネットワーク１０６の上にオーバーレイされたネットワークの一部として接続され得る。ブロックチェーンネットワークのユーザ（「クライアント」と呼ばれることが多い）は、ブロックチェーンネットワーク１０６を含むシステムの一部であるといえるが、これらのユーザは、ブロックチェーンノードに要求される役割を果たさないので、ブロックチェーンノード１０４ではない。代わりに、各当事者１０３はブロックチェーンネットワーク１０６と対話してもよく、それによって、ブロックチェーンノード１０６に接続する（すなわち通信する）ことでブロックチェーン１５０を利用し得る。２つの当事者１０３およびそれらのそれぞれの機器１０２、すなわち、第１の当事者１０３ａおよびそのそれぞれのコンピュータ機器１０２ａ、ならびに第２の当事者１０３ｂおよびそのそれぞれのコンピュータ機器１０２ｂは、例示の目的で示されている。そのような当事者１０３およびそれらのそれぞれのコンピュータ機器１０２ははるかに多く存在し、システム１００に参加し得るが、便宜上、それらは図示されていないことが理解されよう。各当事者１０３は、個人または組織であり得る。純粋に例示として、第１の当事者１０３ａは、本明細書ではアリスと呼ばれ、第２の当事者１０３ｂはボブと呼ばれるが、これは限定的なものではなく、本明細書におけるアリスまたはボブへのいかなる言及も、それぞれ「第１の当事者」および「第２の当事者」と置き換えられ得ることが理解されよう。

【００４０】

各当事者１０３のコンピュータ機器１０２は、１つまたは複数のプロセッサ、例えば、１つまたは複数のＣＰＵ、ＧＰＵ、他のアクセラレータプロセッサ、特定用途向けプロセッサ、および／またはＦＰＧＡを含むそれぞれの処理装置を備える。各当事者１０３のコンピュータ機器１０２は、メモリ、すなわち、１つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージをさらに備える。このメモリは、１つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ＳＳＤ、フラッシュメモリ

もしくはＥＥＰＲＯＭなどの電子媒体、および／または光ディスクドライブなどの光学媒体を採用する１つまたは複数のメモリユニットを備え得る。各当事者１０３のコンピュータ機器１０２上のメモリは、処理装置上で実行されるように構成された少なくとも１つのクライアントアプリケーション１０５のそれぞれのインスタンスを含むソフトウェアを記憶する。本明細書において所与の当事者１０３に帰する任意のアクションは、それぞれのコンピュータ機器１０２の処理装置上で実行されるソフトウェアを使用して実行され得ることが理解されよう。各当事者１０３のコンピュータ機器１０２は、少なくとも１つのユーザ端末、例えば、デスクトップもしくはラップトップコンピュータ、タブレット、スマートフォン、またはスマートウォッチなどのウェアラブルデバイスを含む。所与の当事者１０３のコンピュータ機器１０２はまた、ユーザ端末を介してアクセスされるクラウドコンピューティングリソースなどの１つまたは複数の他のネットワーク化されたリソースを含み得る。

【００４１】

クライアントアプリケーション１０５は、最初に、１つまたは複数の適切なコンピュータ可読ストレージ上で任意の所与の当事者１０３のコンピュータ機器１０２に提供され得、例えば、サーバからダウンロードされ得るか、またはリムーバブルＳＳＤ、フラッシュメモリキー、リムーバブルＥＥＰＲＯＭ、リムーバブル磁気ディスクドライブ、磁気フロッピーディスクもしくはテープ、ＣＤもしくはＤＶＤ　ＲＯＭなどの光ディスク、またはリムーバブル光学ドライブなどのリムーバブル記憶デバイス上で提供され得る。

【００４２】

クライアントアプリケーション１０５は、少なくとも「ウォレット」機能を備える。これは２つの主要な機能を有する。これらのうちの１つは、それぞれの当事者１０３が、トランザクション１５２を作成し、認可し（例えば署名し）、１つまたは複数のビットコインノード１０４に送信することを可能にして、トランザクション１５２をブロックチェーンノード１０４のネットワーク全体に伝搬させ、それによってブロックチェーン１５０に含まれるようにすることである。もう１つは、それぞれの当事者に、その当事者が現在所有しているデジタル資産の額を報告することである。出力ベースのシステムでは、この第２の機能は、当該当事者に属するブロックチェーン１５０全体に散在している様々なトランザクション１５２の出力において定義された額を照合することを含む。

【００４３】

様々なクライアント機能が、所与のクライアントアプリケーション１０５に統合されるものとして説明され得るが、必ずしもこれに限定されるものではなく、代わりに、本明細書で説明される任意のクライアント機能は、例えば、ＡＰＩを介してインターフェースする、または一方が他方へのプラグインである２つ以上の別個のアプリケーション一式において実装され得ることに留意されたい。より一般的には、クライアント機能は、アプリケーション層もしくはオペレーティングシステムなどの下位層、またはこれらの任意の組合せにおいて実装され得る。以下では、クライアントアプリケーション１０５に関して説明するが、これに限定されないことが理解されよう。

【００４４】

各コンピュータ機器１０２上のクライアントアプリケーションまたはソフトウェア１０５のインスタンスは、ネットワーク１０６のブロックチェーンノード１０４のうちの少なくとも１つに動作可能に結合される。これにより、クライアント１０５のウォレット機能はトランザクション１５２をネットワーク１０６に送信することができる。クライアント１０５はまた、それぞれの当事者１０３が受信者である任意のトランザクションについてブロックチェーン１５０にクエリを行うためにブロックチェーンノード１０４にコンタクトすることができる（または、実施形態では、ブロックチェーン１５０は、部分的にその公開性（public visibility）を通じてトランザクションにおける信頼を提供する公共施設であるので、実際にブロックチェーン１５０における他の当事者のトランザクションを検査する）。各コンピュータ機器１０２上のウォレット機能は、トランザクションプロトコルにしたがってトランザクション１５２を定式化し、送信するように構成される。上

述したように、各ブロックチェーンノード１０４は、ブロックチェーンノードプロトコルにしたがってトランザクション１５２を妥当性確認し、トランザクション１５２をフォワードして、それらをブロックチェーンネットワーク１０６全体に伝搬するように構成されたソフトウェアを実行する。トランザクションプロトコルおよびノードプロトコルは互いに対応し、所与のトランザクションプロトコルは所与のノードプロトコルに従い（ｇｏ ｗｉｔｈ）、一緒に所与のトランザクションモデルを実装する。ブロックチェーン１５０内のすべてのトランザクション１５２に対して同じトランザクションプロトコルが使用される。ネットワーク１０６内のすべてのノード１０４によって同じノードプロトコルが使用される。

【００４５】

所与の当事者１０３、例えばアリスが、ブロックチェーン１５０に含まれるべき新しいトランザクション１５２ｊを送信することを望むとき、アリスは、関連トランザクションプロトコルにしたがって（アリスのクライアントアプリケーション１０５内のウォレット機能を使用して）新しいトランザクションを定式化する。次いで、アリスは、クライアントアプリケーション１０５から、アリスが接続されている１つまたは複数のブロックチェーンノード１０４にトランザクション１５２を送信する。例えば、これは、アリスのコンピュータ１０２に最良に接続されたブロックチェーンノード１０４であり得る。任意の所与のブロックチェーンノード１０４は、新しいトランザクション１５２ｊを受信すると、ブロックチェーンノードプロトコルおよびそのそれぞれの役割にしたがってそれを処理する。これには、新たに受信されたトランザクション１５２ｊが「有効」であるための特定の条件を満たすかを最初にチェックすることが含まれ、その例については、以下でより詳細に説明する。いくつかのトランザクションプロトコルでは、妥当性確認のための条件は、トランザクション１５２に含まれるスクリプトによってトランザクションごとに構成可能であり得る。代替的に、条件は、単にノードプロトコルの組込み特徴であってもよいし、スクリプトとノードプロトコルとの組合せによって定義されてもよい。

【００４６】

新たに受信されたトランザクション１５２ｊが、有効であるとみなされるためのテストにパスすることを条件として（すなわち、それが「妥当性確認される」ことを条件として）、トランザクション１５２ｊを受信する任意のブロックチェーンノード１０４は、そのブロックチェーンノード１０４において維持されるトランザクションの順序付きセット１５４に新たな妥当性確認済みトランザクション１５２を追加する。さらに、トランザクション１５２ｊを受信する任意のブロックチェーンノード１０４は、妥当性確認済みトランザクション１５２をネットワーク１０６内の１つまたは複数の他のブロックチェーンノード１０４に伝搬する。各ブロックチェーンノード１０４は同じプロトコルを適用するので、トランザクション１５２ｊが有効であると仮定すると、これは、ネットワーク１０６全体にわたってすぐに伝搬されることを意味する。

【００４７】

所与のブロックチェーンノード１０４において維持される保留中のトランザクションの順序付きプール１５４に承認されると、そのブロックチェーンノード１０４は、新しいトランザクション１５２を含むそれぞれのプール１５４の最新バージョンに対してプルーフオブワークパズルを解こうと競い始める（他のブロックチェーンノード１０４が、トランザクションの異なるプール１５４に基づいてパズルを解こうと試みている可能性があるが、どのノードでも最初に解いたものが、最新のブロック１５１に含まれるトランザクションのセットを定義することを想起されたい。最終的に、ブロックチェーンノード１０４は、アリスのトランザクション１５２ｊを含む順序付きプール１５４の一部についてパズルを解くことになる。）新しいトランザクション１５２ｊを含むプール１５４に対してプルーフオブワークが行われると、それは普遍的にブロックチェーン１５０内のブロック１５１のうちの１つの一部となる。各トランザクション１５２は、先のトランザクションへ戻るポインタを含むので、トランザクションの順序も不変的に記録される。

【００４８】

異なるブロックチェーンノード１０４は、最初、所与のトランザクションの異なるインスタンスを受信し得るので、１つのインスタンスが新しいブロック１５１において公開される（この時点で、公開されたインスタンスが唯一の有効なインスタンスであることにすべてのブロックチェーンノード１０４が同意している）までは、どのインスタンスが「有効」であるかについて相反する見解を有する。ブロックチェーンノード１０４が１つのインスタンスを有効として受け入れ、次いで、別のインスタンスがブロックチェーン１５０に記録されていることを発見した場合、そのブロックチェーンノード１０４は、これを受け入れなければならず、最初に受け入れたインスタンス（すなわち、ブロック１５１で公開されていないもの）を破棄する（すなわち、無効として扱う）。

【００４９】

いくつかのブロックチェーンネットワークによって運用される代替タイプのトランザクションプロトコルは、アカウントベースのトランザクションモデルの一部として、「アカウントベース」プロトコルと呼ばれ得る。アカウントベースのケースでは、各トランザクションは、過去のトランザクションのシーケンスにおける先行するトランザクションのＵＴＸＯを参照することによってではなく、絶対アカウント残高を参照することによって転送されるべき額を定義する。すべてのアカウントの現在の状態は、ブロックチェーンとは別個にそのネットワークのノードによって記憶され、絶えず更新される。そのようなシステムでは、トランザクションは、アカウントの実行中のトランザクションタリー（「ポジション」とも呼ばれる）を使用して順序付けられる。この値は、送信者によってその暗号署名の一部として署名され、トランザクション参照計算の一部としてハッシュされる。加えて、トランザクションにおけるオプションのデータフィールドも署名され得る。このデータフィールドは、例えば、前のトランザクションＩＤがデータフィールドに含まれている場合、前のトランザクションを指し示し得る。

【００５０】

２．ＵＴＸＯベースのモデル

図２は、例示的なトランザクションプロトコルを示す。これは、ＵＴＸＯベースのプロトコルの一例である。トランザクション１５２（「Ｔｘ」と略記される）は、ブロックチェーン１５０の基本的なデータ構造である（各ブロック１５１は１つまたは複数のトランザクション１５２を含む）。以下では、出力ベースまたは「ＵＴＸＯ」ベースのプロトコルを参照して説明する。しかしながら、これはすべての可能な実施形態に限定されるものではない。例示的なＵＴＸＯベースのプロトコルは、ビットコインを参照して説明されるが、他の例示的なブロックチェーンネットワーク上でも等しく実装され得ることに留意されたい。

【００５１】

ＵＴＸＯベースのモデルでは、各トランザクション（「Ｔｘ」）１５２は、１つまたは複数の入力２０２および１つまたは複数の出力２０３を含むデータ構造を含む。各出力２０３は、未使用トランザクション出力（ＵＴＸＯ）を含み得、これは、（ＵＴＸＯがまだ償還されていない場合）別の新しいトランザクションの入力２０２のソースとして使用され得る。ＵＴＸＯは、デジタル資産の額を指定する値を含む。これは、分散型台帳上のトークンの設定数を表す。ＵＴＸＯはまた、他の情報の中でも、元となるトランザクションのトランザクションＩＤを含み得る。トランザクションデータ構造は、入力フィールド（複数可）２０２および出力フィールド（複数可）２０３のサイズを示すインジケータを含み得るヘッダ２０１も含み得る。ヘッダ２０１はまた、トランザクションのＩＤを含み得る。実施形態では、トランザクションＩＤは、（トランザクションＩＤ自体を除く）トランザクションデータのハッシュであり、ノード１０４にサブミットされる生のトランザクション１５２のヘッダ２０１に記憶される。

【００５２】

アリス１０３ａが、当該デジタル資産の額をボブ１０３ｂに転送するトランザクション１５２ｊを作成することを望むとする。図２では、アリスの新しいトランザクション１５２ｊは「Tx_1」とラベル付けされている。これは、シーケンス内の先行するトランザク

ション１５２ｉの出力２０３においてアリスにロックされたデジタル資産の額をとり、これのうちの少なくとも一部をボブに転送する。先行するトランザクション１５２ｉは、図２では「Tx_0」とラベル付けされている。Tx_0およびTx_1は、単なる任意のラベルである。それらは、Tx_0がブロックチェーン１５１内の最初のトランザクションであること、またはTx_1がプール１５４内のすぐ次のトランザクションであることを必ずしも意味するものではない。Tx_1は、アリスにロックされた未使用の出力２０３を依然として有する任意の先行する（すなわち先の）トランザクションを指し示すことができる。

【００５３】

先行するトランザクションTx_0は、アリスが新しいトランザクションTx_1を作成した時点では、または少なくともアリスがそれをネットワーク１０６に送信する時点までには、すでに妥当性確認されブロックチェーン１５０のブロック１５１に含まれている可能性がある。それは、その時点でブロック１５１のうちの１つにすでに含まれていてもよいし、順序付きセット１５４で依然として待機していてもよく、このケースでは、すぐに新しいブロック１５１に含まれることになる。代替的に、Tx_0およびTx_1を作成してネットワーク１０６に一緒に送信することもできるし、ノードプロトコルが「オーファン」トランザクションのバッファリングを可能にする場合には、Tx_0をTx_1の後に送信することさえもできる。トランザクションのシーケンスの文脈において本明細書で使用される「先行する」および「後続する」という用語は、トランザクション内で指定されているトランザクションポインタ（どのトランザクションがどの他のトランザクションを指し示すかなど）によって定義されるシーケンス内のトランザクションの順序を指す。それらは、同様に、「先行するもの（predecessor）」および「後続するもの（successor）」、または「先の（antecedent）」および「後の（descendant）」、「親（parent）」および「子（child）」などと置き換えられ得る。これは、それらの作成、ネットワーク１０６への送信、または任意の所与のブロックチェーンノード１０４への到着の順序を必ずしも意味するものではない。それにもかかわらず、先行するトランザクション（先のトランザクションまたは「親」）を指し示す後続するトランザクション（後のトランザクションまたは「子」）は、親トランザクションが妥当性確認されない限り、妥当性確認されない。親より前にブロックチェーンノード１０４に到着する子は、オーファンとみなされる。それは、ノードプロトコルおよび／またはノード挙動に応じて、親を待つために特定の時間バッファされるかまたは破棄され得る。

【００５４】

先行するトランザクションTx_0の１つまたは複数の出力２０３のうちの１つは、本明細書では$UTXO_0$とラベル付けされた特定のＵＴＸＯを含む。各ＵＴＸＯは、ＵＴＸＯによって表されるデジタル資産の額を指定する値と、ロックスクリプトとを含み、ロックスクリプトは、後続するトランザクションが妥当性確認され、したがってＵＴＸＯが正常に償還されるために、後続するトランザクションの入力２０２内のロック解除スクリプトが満たさなければならない条件を定義する。典型的には、ロックスクリプトは、その額を特定の当事者（それが含まれるトランザクションの受益者）にロックする。すなわち、ロックスクリプトは、典型的には、後続するトランザクションの入力内のロック解除スクリプトに、先行するトランザクションがロックされる当事者の暗号署名が含まれるという条件を含むロック解除条件を定義する。

【００５５】

ロックスクリプト（通称scriptPubKey）は、ノードプロトコルによって認識されるドメイン固有言語で書かれたコードの一部分である。そのような言語の特定の例は、ブロックチェーンネットワークによって使用される「スクリプト（Script）」（大文字Ｓ）と呼ばれる。ロックスクリプトは、トランザクション出力２０３を使用するためにどの情報が必要とされるか、例えばアリスの署名の必要性、を指定する。ロック解除スクリプトはトランザクションの出力に現れる。ロック解除スクリプト（通称scriptSig）は、ロックスクリプト基準を満たすのに必要な情報を提供するドメイン固有言語で書かれたコードの一部分である。例えば、それはボブの署名を含み得る。ロック解除スクリプトは、トラ

ンザクションの入力２０２に現れる。

【００５６】

つまり、図示の例では、Tx_0の出力２０３内の$UTXO_0$は、$UTXO_0$が償還されるために（厳密には、$UTXO_0$を償還しようとする後続するトランザクションが有効となるために）アリスの署名Sig P_Aを必要とするロックスクリプト[Checksig P_A]を含む。[Checksig P_A]は、アリスの公開鍵－秘密鍵ペアの公開鍵P_Aの表現（すなわち、ハッシュ）を含む。Tx_1の入力２０２は、（例えば、実施形態ではトランザクションTx_0全体のハッシュであるそのトランザクションＩＤ、$TxID_0$によって）Tx_1を指し示すポインタを含む。Tx_1の入力２０２は、Tx_0の任意の他の可能な出力の中から$UTXO_0$を識別するために、Tx_0内の$UTXO_0$を識別するインデックスを含む。Tx_1の入力２０２は、アリスが鍵ペアのアリスの秘密鍵をデータの所定の部分（暗号では「メッセージ」と呼ばれることもある）に適用することによって作成された、アリスの暗号署名を含むロック解除スクリプト〈Sig P_A〉をさらに含む。有効な署名を提供するためにアリスによって署名される必要があるデータ（または「メッセージ」）は、ロックスクリプトによって、またはノードプロトコルによって、またはこれらの組合せによって定義され得る。

【００５７】

新しいトランザクションTx_1がブロックチェーンノード１０４に到着すると、ノードはノードプロトコルを適用する。これは、ロックスクリプトおよびロック解除スクリプトを一緒に実行して、ロック解除スクリプトがロックスクリプトで定義されている条件（この条件は１つまたは複数の基準を含み得る）を満たすかどうかをチェックすることを含む。実施形態では、これは２つのスクリプトを連結することを含む：

〈Sig P_A〉 〈P_A〉 ¦¦ [Checksig A_P]

ここで、「¦¦」は連結を表し、「〈…〉」はデータをスタックに置くことを意味し、「[…]」はロックスクリプト（この例ではスタックベースの言語）で構成される関数である。同等に、スクリプトは、スクリプトを連結するのではなく、共通スタックを用いて次々に実行され得る。いずれにしても、一緒に実行されると、スクリプトは、Tx_0の出力内のロックスクリプトに含まれるようなアリスの公開鍵P_Aを使用して、Tx_1の入力内のロック解除スクリプトが、データの予想される部分に署名するアリスの署名を含むことを認証する。この認証を実行するためには、データの予想される部分自体（「メッセージ」）も含まれる必要がある。実施形態では、署名されるデータはTx_1の全体を含む（つまり、平文のデータの署名された部分を指定する別個の要素は、すでに本質的に存在するので、含まれる必要がない）。

【００５８】

公開－秘密暗号法による認証の詳細は、当業者によく知られている。基本的に、アリスが自身の秘密鍵を使用してメッセージに署名した場合、アリスの公開鍵および平文のメッセージが与えられると、ノード１０４などの別のエンティティは、メッセージがアリスによって署名されたものに違いないことを認証することができる。署名は、典型的には、メッセージをハッシュし、ハッシュに署名し、これを署名としてメッセージにタグ付けすることを含み、これにより、公開鍵の任意の保持者が署名を認証することができるようになる。したがって、データの特定の部分またはトランザクションの一部などに署名することへの本明細書におけるいかなる参照も、実施形態では、データのその部分またはトランザクションの一部のハッシュに署名することを意味し得ることに留意されたい。

【００５９】

Tx_1内のロック解除スクリプトが、Tx_0のロックスクリプト内で指定されている１つまたは複数の条件を満たす場合（つまり、図示の例では、アリスの署名がTx_1内で提供され、認証された場合）、ブロックチェーンノード１０４は、Tx_1が有効であるとみなす。これは、ブロックチェーンノード１０４が、保留中のトランザクションの順序付きプール１５４にTx_1を追加することとなることを意味する。ブロックチェーンノード１０４はまた、トランザクションTx_1をネットワーク１０６内の１つまたは複数の他のブ

ロックチェーンノード１０４にフォワードして、トランザクションTx_1がネットワーク１０６全体に伝搬されるようにする。Tx_1が妥当性確認されてブロックチェーン１５０に含まれると、これは、Tx_0からの$UTXO_0$を使用済みとして定義する。Tx_1は、未使用トランザクション出力２０３を使用する場合にのみ有効になり得ることに留意されたい。別のトランザクション１５２によってすでに使用された出力を使用しようとする場合、Tx_1は、他のすべての条件が満たされたとしても無効になる。したがって、ブロックチェーンノード１０４はまた、先行するトランザクションTx_0内の参照されたＵＴＸＯがすでに使用済みであるかどうか（すなわち、それが別の有効なトランザクションへの有効な入力をすでに形成したかどうか）をチェックする必要がある。これは、ブロックチェーン１５０がトランザクション１５２に定義された順序を課すことが重要である理由の１つである。実際には、所与のブロックチェーンノード１０４は、どのトランザクション１５２内のどのＵＴＸＯ２０３が使用されたかをマーキングする別個のデータベースを維持し得るが、最終的には、ＵＴＸＯが使用されたかどうかを定義するものは、ブロックチェーン１５０内の別の有効なトランザクションへの有効な入力をすでに形成しているかどうかである。

【００６０】

所与のトランザクション１５２のすべての出力２０３において指定された総額が、そのすべての入力２０２によって指し示された総額よりも大きい場合、これは、ほとんどのトランザクションモデルにおいて無効性の別の根拠となる。そのため、そのようなトランザクションは、伝搬されることも、ブロック１５１に含まれることもない。

【００６１】

ＵＴＸＯベースのトランザクションモデルでは、所与のＵＴＸＯが全体として使用される必要があることに留意されたい。ＵＴＸＯにおいて使用済みとして定義された額の一部は、別の一部が使用されている間、「残す」ことはできない。しかしながら、次のトランザクションの複数の出力間でＵＴＸＯからの額を分割することはできる。例えば、Tx_0内の$UTXO_0$において定義された額を、Tx_1内の複数のＵＴＸＯ間で分割することができる。したがって、アリスが、$UTXO_0$において定義された額のすべてをボブに与えたくない場合、アリスは、リマインダを使用して、Tx_1の第２の出力において自分自身に残りを与えるか、または別の当事者に支払うことができる。

【００６２】

実際には、アリスはまた、通常、アリスのトランザクション１０４をブロック１５１に成功裏に含めるビットコインノード１０４に対する手数料を含める必要がある。アリスがそのような手数料を含めない場合、Tx_0は、ブロックチェーンノード１０４によって拒否され得、したがって、技術的に有効であっても、伝搬されず、ブロックチェーン１５０に含まれない可能性がある（ノードプロトコルは、ブロックチェーンノード１０４が望まない場合にトランザクション１５２を受け入れることを強制しない）。いくつかのプロトコルでは、トランザクション手数料は、それ自体の別個の出力２０３を必要としない（すなわち、別個のＵＴＸＯを必要としない）。代わりに、所与のトランザクション１５２の入力（複数可）２０２によって指し示される総額と出力（複数可）２０３で指定されている総額との間の任意の差が、トランザクションを公開するブロックチェーンノード１０４に自動的に与えられる。例えば、$UTXO_0$へのポインタがTx_1への唯一の入力であり、Tx_1は唯一の出力$UTXO_1$を有するとする。$UTXO_0$において指定されたデジタル資産の額が$UTXO_1$において指定された額よりも大きい場合、その差分は、$UTXO_1$を含むブロックを作成するためのプルーフオブワーク競争に勝つノード１０４によって割り当てられ得る（または、使用され得る）。しかしながら、代替的にまたは追加的に、トランザクション手数料がトランザクション１５２のＵＴＸＯ２０３のうちのそれ自体の１つにおいて明示的に指定され得ることは必ずしも除外されない。

【００６３】

アリスおよびボブのデジタル資産は、ブロックチェーン１５０内のどこかで任意のトランザクション１５２においてそれらにロックされたＵＴＸＯから構成される。したがって

、典型的には、所与の当事者１０３の資産は、ブロックチェーン１５０全体にわたる様々なトランザクション１５２のＵＴＸＯ全体に散在している。ブロックチェーン１５０内のどこにも、所与の当事者１０３の総残高を定義する数字は記憶されない。クライアントアプリケーション１０５におけるウォレット機能の役割は、それぞれの当事者にロックされ、別の前方のトランザクションでまだ使用されていない様々なＵＴＸＯのすべての値を一緒に照合することである。これは、ビットコインノード１０４のいずれかに記憶されたブロックチェーン１５０のコピーにクエリを行うことによって行うことができる。

【００６４】

スクリプトコードは、多くの場合、概略的に（すなわち、正確な言語を使用せずに）表されることに留意されたい。例えば、特定の機能を表すためにオペレーションコード（オペコード）が使用され得る。「OP_...」は、スクリプト言語の特定のオペコードを指す。例として、ＯＰ＿ＲＥＴＵＲＮは、ロックスクリプトの最初にＯＰ＿ＦＡＬＳＥが先行するときに、トランザクション内にデータを記憶することができ、それによってデータをブロックチェーン１５０内に不変的に記録することができる、トランザクションの使用不可能な出力を作成するスクリプト言語のオペコードである。例えば、データは、ブロックチェーンに記憶することが望まれる文書を含み得る。

【００６５】

典型的には、トランザクションの入力は、公開鍵P_Aに対応するデジタル署名を含む。実施形態において、これは、楕円曲線ｓｅｃｐ２５６ｋ１を使用するＥＣＤＳＡに基づく。デジタル署名は、データの特定の一部分に署名する。いくつかの実施形態では、所与のトランザクションについて、署名は、トランザクション入力の一部、およびトランザクション出力の一部または全部に署名する。署名された出力の特定の部分は、ＳＩＧＨＡＳＨフラグに依存する。ＳＩＧＨＡＳＨフラグは、通常、どの出力が署名されるかを選択するために署名の最後に含まれる４バイトコードである（したがって、署名時に固定される）。

【００６６】

ロックスクリプトは、典型的には、それぞれのトランザクションがロックされる当事者の公開鍵を含むという事実を指して、「scriptPubKey」と呼ばれることがある。ロック解除スクリプトは、典型的には、それが対応する署名を供給するという事実を指して「scriptSig」と呼ばれることがある。しかしながら、より一般的には、ＵＴＸＯが償還されるための条件が署名を認証することを含むことは、ブロックチェーン１５０のすべてのアプリケーションにおいて必須ではない。より一般的には、スクリプト言語を使用して、任意の１つまたは複数の条件を定義することができる。したがって、より一般的な用語「ロックスクリプト」および「ロック解除スクリプト」が好まれ得る。

【００６７】

３．サイドチャネル

図１に示すように、アリスおよびボブのコンピュータ機器１０２ａ、１２０ｂの各々上のクライアントアプリケーションは、それぞれ、追加の通信機能を含み得る。この追加の機能により、（いずれかの当事者または第三者の扇動で）アリス１０３ａは、ボブ１０３ｂと別個のサイドチャネル１０７を確立することができる。サイドチャネル１０７は、ブロックチェーンネットワークとは別でのデータの交換を可能にする。そのような通信は、「オフチェーン」通信と呼ばれることがある。例えば、これは、当事者の一方がトランザクションをネットワーク１０６にブロードキャストすることを選択するまで、トランザクションが（まだ）ブロックチェーンネットワーク１０６に登録されることなく、またはチェーン１５０上に進むことなく、アリスとボブとの間でトランザクション１５２を交換するために使用され得る。このようにトランザクションを共有することは、「トランザクションテンプレート」の共有と呼ばれることがある。トランザクションテンプレートは、完全なトランザクションを形成するために必要とされる１つまたは複数の入力および／または出力を欠いていてもよい。代替的にまたは追加的に、サイドチャネル１０７は、鍵、交渉された額または条件、データコンテンツなどの任意の他のトランザクション関連データ

を交換するために使用され得る。

【００６８】

サイドチャネル１０７は、ブロックチェーンネットワーク１０６と同じパケット交換ネットワーク１０１を介して確立され得る。代替的にまたは追加的に、サイドチャネル３０１は、モバイルセルラーネットワークなどの異なるネットワーク、またはローカルワイヤレスネットワークなどのローカルエリアネットワーク、またはアリスのデバイス１０２ａとボブのデバイス１０２ｂとの間の直接のワイヤードまたはワイヤレスリンクを介して確立され得る。一般に、本明細書のどこでも、参照されるサイドチャネル１０７は、「オフチェーン」すなわちブロックチェーンネットワーク１０６とは別でデータを交換するための１つまたは複数のネットワーキング技術または通信媒体を介した任意の１つまたは複数のリンクを含み得る。２つ以上のリンクが使用される場合、全体としてのオフチェーンリンクの束または集合は、サイドチャネル１０７と呼ばれることがある。したがって、アリスおよびボブがサイドチャネル１０７上で情報またはデータの特定の部分などを交換すると言われている場合、これは、これらのデータの部分のすべてが全く同じリンクまたは同じタイプのネットワーク上で送信されなければならないことを必ずしも意味するものではないことに留意されたい。

【００６９】

４．ロックスクリプトの伝搬の強制

図３は、ロックスクリプトの伝搬を強制する、すなわち、トランザクションのチェーン内の各トランザクションに強制的に同じロックスクリプトを含ませるための例示的なシステム３００を示す。システム３００は、第１の当事者（例えば、アリス１０３ａ）および第２の当事者（例えば、ボブ１０３ｂ）を含む。システム３００は、１つまたは複数の追加の当事者、例えばチャーリー１０３ｃも含み得る。システム３００は、ブロックチェーンネットワーク１０６の１つまたは複数のブロックチェーンノード１０４も含む。第１の当事者および第２の当事者は、以下の説明を通してアリス１０３ａおよびボブ１０３ｂと呼ばれるが、これは単に便宜上のものであることに留意されたい。すなわち、第１の当事者および第２の当事者は、必ずしも、図１および２に関連してアリス１０３ａおよびボブ１０３ｂによって実行されるものとして上記で説明されたアクションの一部または全部を実行するように構成される必要はないが、それは、当然ながらオプションである。より一般的には、第１の当事者と第２の当事者の両方が、図１および図２に関連してアリス１０３ａおよび／またはボブ１０３ｂによって実行されるものとして上記で説明されたアクションの一部または全部を実行するように構成されてもよい。アリス１０３ａ、ボブ１０３ｂ、またはチャーリー１０３ｃによって実行されるものとして説明される任意の動作は、それらのそれぞれのコンピューティング機器によって実行されることが理解されよう。

【００７０】

本発明の実施形態は、ロックスクリプトがブロックチェーントランザクションのチェーン全体に伝搬されることを確実にするために使用され得る。一般に、プロトコルは、一次トランザクションおよび補助トランザクションという２つのタイプのトランザクションに依存する。一次トランザクションはそれぞれ、それぞれの一次ロックスクリプトをロック解除しようとするそれぞれの補助トランザクションに対して条件を適用するそれぞれの一次ロックスクリプトを含む。各一次ロックスクリプトは同じであり、すなわち、それは同じフォーマットをとり、同じデータを含み、同じ機能を実装する。補助トランザクションはそれぞれ、それぞれの補助ロックスクリプトをロック解除しようとするそれぞれの一次トランザクションに対して条件を適用するそれぞれの補助ロックスクリプトを含む。各補助ロックスクリプトは同じである。

【００７１】

第１の当事者アリス１０３ａは、発行トランザクションの生成を担い得る。発行トランザクションは、トランザクションのチェーン内の最初の一次トランザクションである。発行トランザクションは、第１の一次トランザクションと呼ばれることもある。一次トランザクションがトークン関連データを含む例では、アリス１０３ａがトークンの発行者であ

り得る。他の例では、異なる当事者がトークンチェーンを開始してもよい。

【００７２】

第１の一次トランザクションは、１つまたは複数の入力および１つまたは複数の出力を含む。例えば、一次トランザクションの入力は、アリスの前のトランザクションのうちの１つの出力をロック解除するために使用され得、したがって、それを行うために必要とされるデータ、例えば、アリスの署名および公開鍵を含み得る。第１の一次トランザクションの出力（例えば、第１の出力）は、一次ロックスクリプト、すなわち、将来のトランザクション全体にわたって伝搬されることになる同じ一次ロックスクリプトのインスタンスを含む。第１の一次トランザクションの別の出力（例えば、第２の出力）は、使用条件（spending condition）を指定し得、例えば、それは、アリス１０３ａまたは異なる当事者によって所有される公開鍵にロックされ得る。例えば、第２の出力は、Ｐ２ＰＫＨ（pay-to-public-key-hash）出力を含み得る。

【００７３】

一次ロックスクリプトは、一次ロックスクリプトをロック解除しようとするトランザクション（このケースでは第１の補助トランザクション）に対していくつかの条件を適用するように構成される。補助トランザクションに対して適用される条件の１つは、第１の補助トランザクションのロック解除スクリプトが、第１の補助トランザクション自体に基づく第１のメッセージを含まなければならないことである。すなわち、第１のメッセージは、第１の補助トランザクションのフィールドのうちの１つ、いくつか、またはすべてに基づく。フィールドのいくつかは生の形式であってもよく、いくつかは難読化、例えばハッシュ化されていてもよい。ロック解除スクリプトにそのようなメッセージを強制的に含ませるための１つの技法は、ＰＵＳＨＴＸとして当技術分野で知られており、これは、擬似オペコード、すなわち、特定の機能を実行するように構成されたオペコードの組合せである。ＰＵＳＨＴＸについては、https://xiaohuiliu.medium.com/op-push-tx-3d3d279174c1に記載されている。別の例は、英国特許出願ＧＢ２１１２９３０．９に記載されている。ロック解除スクリプトに第１のメッセージを強制的に含ませることにより、第１の補助トランザクションに対して特定のチェックを行うことができる。このメッセージは、ブロックチェーン署名プロトコルの一部として署名の前にハッシュされるプリイメージなので、ｓｉｇｈａｓｈプリイメージと呼ばれることがある。別の言い方をすれば、ロック解除スクリプトに第１のメッセージを強制的に含ませることにより、第１の補助トランザクションに対して特定の条件を適用することができる。適用される１つの条件は、第１の補助トランザクションの出力（例えば、第１の出力）が補助ロックスクリプト（以下で説明）を含むことである。

【００７４】

この条件を適用する方法の例として、一次ロックスクリプトは、補助ロックスクリプトのハッシュに基づいてハッシュパズルを実装するように構成されたサブスクリプトを含み得る。すなわち、ハッシュパズルは、補助ロックスクリプトのハッシュを含み、ハッシュに対するプリイメージが提供された場合にのみ解くことができ、したがってサブスクリプトがロック解除されるように構成され、プリイメージは補助ロックスクリプトである。したがって、一次ロックスクリプトは、第１のメッセージからロックスクリプト（例えば、第１のロックスクリプト）を抽出し、抽出されたロックスクリプトをハッシュし、それが一次ロックスクリプトの一部として含まれる補助ロックスクリプトのハッシュにハッシュすることを検証するように構成され得る。

【００７５】

アリス１０３ａが第１の補助トランザクションを作成してもいし、それは異なる当事者によって生成されてもよい。第１の補助トランザクションは、上述したように、第１のメッセージを含む入力と、補助ロックスクリプトを含む出力とを含む。補助ロックスクリプトは、補助ロックスクリプトをロック解除しようとするトランザクション（このケースでは第２の一次トランザクション）に対していくつかの条件を適用するように構成される。第２の一次トランザクションに対して適用される条件のうちの１つは、第２の一次トラン

ザクションのロック解除スクリプト（例えば、第１のロック解除スクリプト）が、第２の一次トランザクションに基づく第２のメッセージを含まなければならないことである。すなわち、第２のメッセージは、第２の一次トランザクションのフィールドのうちの１つ、いくつか、またはすべてに基づく。フィールドのいくつかは生の形式であってもよく、いくつかは難読化、例えばハッシュ化されていてもよい。第２の一次トランザクションに対して適用される別の条件は、ロック解除スクリプトが、第１の一次トランザクションの１つ、いくつか、または各ロックスクリプトを含むデータ項目も含まなければならず、したがって、第１のデータ項目が一次ロックスクリプトを含まなければならないことである。すなわち、第１のデータ項目は、少なくとも一次ロックスクリプトを含まなければならない。いくつかの例では、第１のデータ項目は、以下に説明する第１の一次トランザクションの第２のロックスクリプトも含まなければならない。第２のメッセージおよび第１のデータ項目を使用して、補助ロックスクリプトは、第２の一次トランザクションに一次ロックスクリプトを強制的に出力（例えば、第１の出力）に含ませる。したがって、一次ロックスクリプトが一次トランザクション間で伝搬される。一次ロックスクリプトは、補助トランザクションに補助ロックスクリプトを強制的に含ませるので、補助ロックスクリプトもまた、補助トランザクション間で伝搬される。したがって、一次ロックスクリプトまたは補助ロックスクリプトの一部として含まれる任意のデータ（例えば、トークン関連データ）は、それぞれ一次トランザクションまたは補助トランザクションの間で伝搬される。

【００７６】

上述したように、第１の一次トランザクションは第２の出力を含み得る。補助ロックスクリプトは、第１の一次トランザクションの第２の出力が第２の一次トランザクションの入力によって強制的に使用されるように構成され得る。すなわち、これらの例では、第１の一次ロックスクリプトの第２の出力および第１の補助トランザクションの第１の出力は、別個のトランザクションではなく、第２の一次トランザクションのそれぞれの入力によって使用されなければならない。

【００７７】

この条件を適用するための１つのオプションは、第２の一次トランザクションの第１のロック解除スクリプトに第１の補助トランザクションを強制的に含ませることである。すなわち、第２のデータ項目は、第２の一次トランザクションのロック解除スクリプトに含まれ、その第２のデータ項目は、第１の補助トランザクションを、例えばシリアル化された形式で含む。補助ロックスクリプトは、第２のメッセージから、第２の一次トランザクションの第１の入力によって使用される出力のそれぞれのトランザクション識別子を抽出するように構成される。同様に、補助ロックスクリプトは、第２のデータ項目から、第１の補助トランザクションの第１の入力によって使用される出力のそれぞれのトランザクション識別子を抽出するように構成される。補助トランザクションは、抽出されたトランザクション識別子を強制的に同じにするように構成される。

【００７８】

これらの例では、第２の一次トランザクションは、第１の一次トランザクションの第２の出力をロック解除するように構成された第２の入力を含み、したがって、トランザクションのリンクされたチェーンを作成する。トランザクションは、第１の一次トランザクションの出力および第１の補助トランザクションの出力が同じ第２の一次トランザクションによって使用されるという上記で説明した強制なしに、このようにリンクされてもよいことに留意されたい。しかしながら、この条件をスクリプト内で適用させることには（後述する）利点がある。

【００７９】

いくつかの例では、第１のデータ項目（すなわち、第１の一次トランザクションのロックスクリプトを含むデータ項目）が正しいことを検証するために、補助ロックスクリプトによって追加の検証が実行され得る。第１の一次トランザクションは、ハッシュ関数、例えばＳＨＡ２５６またはＳＨＡ５１２を用いてトランザクションをハッシュすることによって生成されるトランザクション識別子を有するか、または少なくともそれに関連付けら

れる。任意の適切なハッシュ関数が使用され得る。第２の一次トランザクションの第１のロック解除スクリプトは、ハッシュ関数を第１の一次トランザクションに適用するミッドステート（midstate）を含む第３のデータ項目を含み得る。当業者は、ミッドステート自体に精通しているであろう。一般に、データが（例えば、ＳＨＡベースのハッシュ関数を用いて）ハッシュされると、データはブロックに分割される。ブロックの第１のセットを使用して一連の計算を実行して第１のミッドステートを生成する。第１のミッドステートとブロックの第２のセットとを使用して一連の計算を実行して第２のミッドステートを生成し、以下同様に、すべてのブロックが演算されてハッシュダイジェストを得られるまで続けられる。言い換えると、ハッシュ関数は単一の演算を含まない。むしろ、ハッシュ関数への入力はブロック（例えば、ＳＨＡ２５６の場合は６４バイト）に分割され、これらのブロックは順番に処理される。ミッドステートは、これらのブロックのうちの１つまたは複数を処理した後の内部状態である。ミッドステートは、各ブロックが処理された後に生成される。ｎ番目のブロックを処理した後に計算されたミッドステートは、すべての前のブロックに依存するが、後続するブロックには依存しない。ＳＨＡ２５６ハッシュ関数の内部動作の一例は、https://qvault.io/cryptography/how-sha-2-works-step-by-step-sha-256/において見ることができる。

【００８０】

本開示は、ロックスクリプトを含まない第１の一次トランザクションの第１の部分にハッシュ関数を適用することによって生成されたミッドステートを提供することによって、第１のデータ項目（すなわち、ロックスクリプト）を使用してスクリプト内で計算を継続して、第１の一次トランザクションの候補トランザクション識別子を生成することができることを認識する。候補トランザクション識別子が第１の一次トランザクションの実際のトランザクション識別子と一致する場合、第１のデータ項目は、第１の一次トランザクションの実際のロックスクリプト（したがって、一次ロックスクリプト）を含まなければならない。

【００８１】

補助ロックスクリプトはまた、第２の一次トランザクションに、チェーン内の最初の一次トランザクションのトランザクション識別子を強制的に含ませるように構成され得る。例えば、第２の一次トランザクションは、トランザクション識別子を第２の出力に含めるように要求され得る。第１の一次トランザクションがチェーン内の最初のトランザクションである場合、第２の一次トランザクションを含む後続する各トランザクションは、第１の一次トランザクションのトランザクション識別子を含むように強制される。これらの例では、最初の一次トランザクション（例えば、第１の一次トランザクション）は、最初の一次トランザクションの全体に基づくので、それ自体のトランザクション識別子を含まない。後続する各一次トランザクションは、最初の一次トランザクションのトランザクション識別子を含むように強制され得る。

【００８２】

第２の一次トランザクションは、アリス１０３ａまたは異なる当事者によって生成され得る。上述したように、第２の一次トランザクションは、第１の補助トランザクションの第１の出力を使用する第１の入力を含み、第１の一次トランザクションの第２の出力を使用する第２の入力を含み得る。第２の一次トランザクションはまた、一次ロックスクリプトを含む第１の入力を含む。第２の一次トランザクションは、例えば、次の一次トランザクションの第２の出力によってロック解除される第２の出力を含み得る。第２の出力は、第２の当事者であるボブ１０３ｂの公開鍵にロックされ得る。第２の出力は、最初の一次トランザクションのトランザクション識別子を含み得る。

【００８３】

アリス１０３ａは、第１の一次トランザクションを１つまたは複数のブロックチェーンノード１０４にサブミットさせるように構成される。例えば、アリス１０３ａ自身が第１の一次トランザクションをブロックチェーンネットワーク１０６に送信してもよいし、アリス１０３ａが第１の一次トランザクションをブロックチェーンネットワークに中間者、

例えばボブ１０３ｂを介して送信してもよい。同様に、アリス１０３ａは、第１の補助トランザクションおよび／または第２の一次トランザクションを１つまたは複数のブロックチェーンノード１０４にサブミットさせるように構成され得る。

【００８４】

いくつかの例では、アリス１０３ａは、第２の一次トランザクションをボブ１０３ｂに送信し得る。アリス１３０ａはまた、マークルプルーフをボブ１０３ｂに送信し得、マークルプルーフは、第１の一次トランザクションがブロックチェーン１５０に記録されたことを検証するために使用され得る。

【００８５】

ここまで、第１の一次トランザクションは、最初の一次トランザクションとして主に説明してきた。しかしながら、第１の一次トランザクションは、一般に、ｎ番目の一次トランザクションであり得、このケースでは、上記で説明される第２の一次トランザクションは、ｎ＋１番目の一次トランザクションとなる。同様に、この例では、ｎ－１番目の一次トランザクションが存在する。このケースでは、アリス１３０ａは、発行トランザクションの生成を担っても担わなくてもよい。

【００８６】

この例を続けると、第１の一次トランザクションは、前の補助トランザクションの第１の出力を参照する第１の入力を含み得る。第１の補助トランザクションと同様に、前の補助トランザクションの第１の出力は、補助ロックスクリプト（すなわち、同じ正確な補助スクリプトのインスタンス）を含む。前の補助トランザクションは、前の一次トランザクションの第１の出力を参照する入力も有する。前の一次トランザクションの第１の出力は、一次ロックスクリプトを含む。この例では、前の一次トランザクションは、最初の一次トランザクションであり得るか、一次トランザクションのチェーン内の異なる一次トランザクションであり得る。

【００８７】

前の補助トランザクションの補助ロックスクリプトは、第１の補助トランザクションの補助ロックスクリプトが第２の一次トランザクションに対して行うのと同じように、第１の一次トランザクションに対して上記で説明したチェックを行う（すなわち、同じ条件を適用する）ように構成される。例えば、第１の一次トランザクションに一次ロックスクリプトを強制的に含ませる。

【００８８】

これらの例では、第２の一次トランザクションが親一次トランザクション（すなわち、第１の一次トランザクション）および祖父母一次トランザクション（すなわち、前の一次トランザクション）を有するとき、第１の一次トランザクションの補助ロックスクリプトは、第２の一次トランザクションのロック解除スクリプトに、前の一次トランザクションの１つ、いくつか、または各ロックスクリプトを含む第４のデータ項目を強制的に含ませるように構成される。これは、一次ロックスクリプトを含む。第２の一次トランザクションのロック解除スクリプトに含まれる第２のメッセージ（すなわち、第２の一次トランザクションに基づいて生成されたメッセージ）および第４のデータ項目を使用して、補助ロックスクリプトは、第２の一次トランザクションの第１の出力に含まれる一次ロックスクリプトが前の一次トランザクションに含まれる一次ロックスクリプトと同じであることを検証する。これは、同じ一次ロックスクリプトがある一次トランザクションから次の一次トランザクションへ、そして次の一次トランザクションへ渡されることを検証する。言い換えると、前の一次トランザクションの一次ロックスクリプトが実行されたことを証明する。

【００８９】

前の一次トランザクションにハッシュ関数（例えば、ＳＨＡ２５６）を適用することによって生成されたミッドステートを使用して、第４のデータ項目が正しいことを検証し得る。すなわち、第４のデータ項目（すなわち、前の一次トランザクションのロックスクリプトを含むデータ項目）が正しいことを検証するために、補助ロックスクリプトによって

追加の検証が実行され得る。前の一次トランザクションは、ハッシュ関数、例えば、ＳＨＡ２５６を用いてトランザクションをハッシュすることによって生成されるトランザクション識別子を有するか、または少なくともそれに関連付けられる。第２の一次トランザクションの第１のロック解除スクリプトは、ハッシュ関数を前の一次トランザクションに適用するミッドステートを含む第５のデータ項目を含み得る。補助ロックスクリプトは、第４のデータ項目および第５のデータ項目を使用して前の一次トランザクションの候補トランザクション識別子を生成し得る。候補トランザクション識別子が前の一次トランザクションの実際のトランザクション識別子と一致する場合、第４のデータ項目は、前の一次トランザクションの実際のロックスクリプト（したがって、一次ロックスクリプト）を含まなければならない。

【００９０】

第１の一次トランザクションは、前の一次トランザクションの第２の出力を使用する第２の入力を含み得る。条件は、前の補助ロックスクリプトに含まれる補助ロックスクリプトによって適用され得る。

【００９１】

前の一次トランザクションおよび／または前の補助トランザクションは、アリス１０３ａによって生成されてもよいし、異なる当事者、例えばボブ１０３ｂもしくはチャーリー１０３ｃによって生成されてもよい。同様に、上記の説明は概して、アリス１０３ａが第１の一次トランザクション、第１の補助トランザクション、および第２の一次トランザクションを生成することに関して説明したが、これらのトランザクションのいずれか１つまたは複数は、代わりに、ボブ１０３ｂまたはチャーリー１０３ｃによって生成されてもよい。

【００９２】

当事者、例えばボブ１０３ｂは、一次トランザクションが正しく生成されたことを検証する意図をもって、一次トランザクションを取得し得る。例えば、ボブ１０３ｂは、アリス１０３ａから第２の一次トランザクションを取得し得る。以上、第２の一次トランザクションについて説明した。簡単に要約すると、第２の一次トランザクションは、一次ロックスクリプトを含む第１の出力と、少なくとも（第２の一次トランザクションに基づいて生成された）第２のメッセージおよび第１の一次トランザクションの１つ、いくつか、または各ロックスクリプトを含む第１のデータ項目とを含む第１の入力（すなわち、第１の入力のロック解除スクリプト）とを有する。ロックスクリプトの１つは、一次ロックスクリプトを含む。以下の例では、第２の一次トランザクションの第１の入力は、第１の補助トランザクションも含む。

【００９３】

ボブ１０３ｂは、第１の一次トランザクションがブロックチェーン１５０上に記録されており、したがって、有効なトランザクションであることを検証する。そうするための１つのオプションは、候補ブロックヘッダを計算するために、第１の一次トランザクションのトランザクション識別子とともに使用することができるマークルプルーフ（すなわち、ハッシュのセット）を取得することである。候補ブロックヘッダがブロックチェーンブロック１５１のブロックヘッダと一致する場合、ボブ１０３ｂは、ブロックが第１の一次トランザクションを含むことを確信することができる。別のオプションは、ブロックチェーンノード１０４に確認を要求することである。

【００９４】

上述したように、第２の一次トランザクションのロック解除スクリプトは、第３のデータ項目（第１の一次トランザクションの一部に適用されたハッシュ関数のミッドステート）を含み得る。ボブ１０３ｂは、第１のデータ項目および第３のデータ項目を使用して、第１の一次トランザクションの候補トランザクション識別子を生成し、候補トランザクション識別子を使用してマークルプルーフを実行し得る。

【００９５】

ボブ１０３ｂはまた、一次ロックスクリプトが、第１の補助トランザクションが補助ロ

ックスクリプトを含むことを要求するように構成されているという意味で、第１の一次トランザクションに含まれる一次ロックスクリプトが正しいことを検証する。ボブ１０３ｂは、一次ロックスクリプトを明示的に検証し、すなわち、それが正しい機能などを含むことをチェックし得る。より効率的なオプションとして、上述したように、一次ロックスクリプトは、ハッシュパズルの一部として補助ロックスクリプトのハッシュを含み得る。したがって、ハッシュパズルを解くためには、補助トランザクションに補助ロックスクリプトが含まれていなければならない。したがって、ボブ１０３ｂは、一次ロックスクリプトが補助ロックスクリプトのハッシュを含むことを検証し得る。例えば、ハッシュは、例えば、アリス１０３ａまたは最初の一次トランザクションを生成した当事者によって送信され得るなどして、ボブ１０３ｂが事前に知っていてもよい。

【００９６】

オプションのチェックとして、ボブ１０３ａは、第２の一次トランザクションが、第１の一次トランザクションであり得る、一次トランザクションのチェーン内の最初のトランザクションのトランザクション識別子を（例えば、第２の出力内に）含むことを検証し得る。

【００９７】

検証を実行すると、ボブ１０３ｂは、第２の補助トランザクションおよび第３の一次トランザクションを生成し得る。例えば、これは、トークンをチャーリー１０３ｃに転送するために使用され得る。第２の補助トランザクションは、第１の補助トランザクションと同じフォーマットをとる。第２の補助トランザクションは、第２の一次トランザクションの第１の出力、すなわち一次ロックスクリプトを含む出力を使用する第１の入力を有する。そうするために、それは、第２の補助トランザクションのいくつかまたはすべてのフィールドに基づいて生成された第３のメッセージを含む。第２の補助トランザクションは、補助ロックスクリプトを含む第１の入力を有する。ボブ１０３ｂは、第２の補助トランザクションをブロックチェーンネットワーク１０６に、直接または異なる当事者を介して送信する。

【００９８】

ボブ１０３ｂはまた、第３の一次トランザクションを生成する。第３の一次トランザクションは、第２の一次トランザクションと同様である。それは、第２の一次トランザクションの第１の出力を使用する第１の入力を含む。第３の一次トランザクションの第１の入力は、第３の一次トランザクションのいくつかまたはすべてのフィールドに基づいて生成された第３のメッセージを含む。この例では、第１の入力は、第２の補助トランザクションと、第１のデータ項目（すなわち、第１の一次トランザクションの各ロックスクリプト）と、第２の一次トランザクションの１つ、いくつか、または各ロックスクリプトとを含む第６のデータ項目を含む。第３の一次トランザクションは、チャーリー１０３ｃの公開鍵にロックされた第２の出力を含み得る。第３の一次トランザクションは、第２の一次トランザクションの第２の出力を使用する第２の入力を含み得る。ボブ１０３ｂは、第２の一次トランザクションをブロックチェーンネットワーク１０６に、直接または異なる当事者を介して送信する。

【００９９】

５．ブロックチェーンノード妥当性確認済みトークン

このセクションでは、上述の実施形態の詳細な例を説明した。このセクションで提示される特徴のいくつかはオプションであることを理解されたい。この例は、トランザクションがトークントランザクションである、すなわち、それらがトークン関連データを含むユースケースに焦点を当てる。これは、例示的なユースケースにすぎず、一般に、発行まで戻ってすべてのトランザクションを検証する必要なしに、証明可能なトランザクションチェーンを必要とする他のアプリケーションに使用され得る。

【０１００】

このセクションでは、４つの要件を満たすトークンプロトコルについて説明する。

１．トークントランザクションの検証者はトークンの発行トランザクションまで遡る必

要がなく、よって、効率が向上する。
２．トークンが発行された後は、信頼できる第三者が２人のユーザ間のトークンの転送に関与せず、よって、セキュリティおよびユーザビリティが向上する。
３．トークントランザクションのサイズは一定である。履歴トランザクションの数とは無関係であり、よって、ストレージが向上する。
４．この解決策は、リプレイ攻撃の影響を受けにくく、すなわち、トークンをコピーすることができなく、よって、セキュリティが向上する。

【０１０１】
検証者または第三者がトークン規則を実行する必要なしにスクリプトで強制されるトークンは、「マイナー妥当性確認済み」トークン、または「ブロックチェーンノード妥当性確認済み」トークンとして知られている。本出願は、出願人の知る限りにおいて、ブロックチェーン１５０上の第１のマイナー妥当性確認済みトークンプロトコルを開示する。

【０１０２】
プロトコルは、以下のビルディングブロックを使用する：
・ トランザクションチェーン内の祖父母の２つのロックスクリプトの各々に同じオペコードが現れたこと、およびこれらのロックスクリプトが実行されたことを検証するスクリプトスニペット、
・ トランザクションチェーン内の次の子ロックスクリプトの両方が同じオペコードを含むことを強制するスクリプトスニペット、
・ 部分的ＳＨＡ２５６を実装するスクリプトスニペット、
・ （自己参照を回避するために）１つのトークントランザクションからその支出トランザクションに検証データを渡す補助トランザクション、および
・ 再生不可能な一意のトークンＩＤ（ＵＴＩＤ）
簡単にするために、これは発行トランザクションＩＤであると仮定する。

【０１０３】
プロトコルは以下にステップで示され、一度に解決策の一部を説明する。解決策の反復のたびに問題が作り出され、その問題は解決策の次の反復によって解決され、最終的に完全な解決策に終わる。支出トランザクションＴｘ$_{n+1}$を視点とし、発行トランザクションまで遡ることなく、どのようにしてＴｘ$_{n+1}$を有効なトークントランザクションとして決定することができるかを説明する。Ｔｘ$_{n+1}$に関連するトランザクションを参照するために、以下の用語を使用する：

【数１】
- 親トランザクションTx_n:Tx_{n+1}の第2の入力によって参照されるトランザクション、
- 祖父母トランザクションTx_{n-1}:Tx_nの第2の入力によって参照されるトランザクション、および
- 補助トランザクションTx_n^A（以下$Tx^A{}_n$と記す）:その入力においてTx_nの第1の出力を参照し、Tx_{n+1}の第1の入力によって参照されるトランザクション。

【０１０４】
これらのトランザクションおよびそれらの関係を図４に示す。祖父母トランザクション、親トランザクション、および補助トランザクションはすべてマイナーによって受け入れられたと仮定する。支出トランザクションＴｘ$_{n+1}$は、最後に見られたトランザクションである。

【０１０５】
５．１．トランザクション構造
このプロトコルは、トークンチェーン内の最後の３つのトランザクションを考慮する。トランザクションごとに、補助トランザクションがある。補助トランザクションは、トランザクションの肥大化および自己参照を回避するために必要である。図４にトランザクション構造の概略図が示されており、トランザクションチェーンは、第２の入力および出力のチェーンとして定義されている。これについて、以下でより詳細に説明する。

【０１０６】
トランザクションのシリアル化では、すべての入力が最初に現れ、次いですべての出力が現れる。入力および出力は、それらのインデックスの順序で現れる。以下の説明は、トランザクション符号化を考慮から外して、ロック解除スクリプトおよびロックスクリプトのみに焦点を当てる。ロック解除スクリプトに言及するとき、そのロック解除スクリプトに対応して参照されているアウトポイントにも言及し得ることに留意されたい。これは、関数への「入力」との混同を避けるためである。
【０１０７】
非代替トークン（ＮＦＴ）の表現のようなトランザクションの単一のチェーンを考える。トークンの分割および結合はなく、トランザクション有向非巡回グラフ（ＤＡＧ）が作成されると仮定する。また、追加可能な他の入力および出力はないと仮定する。
【０１０８】
５．２．表記
ＯＰプレフィックスを有する事前定義されたオペコードをＯＰ＿ＣＯＤＥと表記し、プレフィックスのない事前定義されたオペコードの組合せを、例えばＣＯＤＥと表記する。公開鍵などの変数に依存するオペコードまたはオペコードの組合せは、ＣＯＤＥ（ＰＫ）のように括弧で表記される。オペコードの組合せの構成をＣＯＤＥ１＋ＣＯＤＥ２と表記する。
【０１０９】
５．３．プッシュトランザクションおよびその派生物
トランザクションＴｘ$_{n+1}$のｓｉｇｈａｓｈプリイメージｍ$_{n+1}$が入力として与えられると、Ｔｘ$_{n}$のロックスクリプト内のＰＵＳＨＴＸとラベル付けされたオペコードの組合せはＴＲＵＥと評価される。内部で（ｕｎｄｅｒ ｔｈｅ ｈｏｏｄ）、オペコードは、ｓｉｇｈａｓｈプリイメージｍ$_{n+1}$の署名を構築し、検証する。最後のオペコードＯＰ＿ＣＨＥＣＫＳＩＧは、対応する公開鍵を有する署名が入力された場合、ＴＲＵＥと評価され、署名は、トランザクション自体から抽出されたトランザクションメッセージｍ$_{n+1}$に対するものである。
【０１１０】
トランザクションメッセージｍ$_{n+1}$は、Ｔｘ$_{n+1}$およびＴｘ$_{n}$内の様々なフィールドから構築される。特に関心のある分野は次の通りである：
【数２】
1. Tx_{n+1}のロックスクリプト（複数可）
2. アウトポイントTxIDn、$TxID_n^A$（以下$TxID^A_n$と記す）；および
3. Tx_nのロックスクリプト。

【０１１１】
上付き文字Ａは、トランザクションが補助トランザクションであることを示す。
【０１１２】
ＰＵＳＨＴＸを含めることの意味は、現在のトランザクションＴｘ$_{n+1}$のｓｉｇｈａｓｈプリイメージがスタック上にあることが証明可能であるということである。これは、これらのフィールドのいずれかが、オペコードまたはシリアル番号の特定の組合せを含むなど、それらに対して適用される条件を有することができることを意味する。これらの条件が満たされない場合、支出トランザクションは有効ではない。
【０１１３】
ｄＳＨＡ２５６（Ｔｘ$_{n}$）＝ＴｘＩＤ$_{n}$である場合、Ｔｘ$_{n}$およびＴｘＩＤ$_{n}$（ｍ$_{n+1}$からとられる）が入力として与えられると、Ｔｘ$_{n}$内のロックスクリプト内のオペコードＰＵＳＨＴＸＰＡＲＥＮＴの組合せはＴＲＵＥと評価され、ここで、ｄＳＨＡ２５６はダブルＳＨＡ２５６ハッシュ関数である。内部で、オペコードは、シリアル化されたトランザクションＴｘ$_{n}$のハッシュを計算し、その結果をＴｘＩＤ$_{n}$と比較する。ここで、トラ

ンザクションＩＤは、ＰＵＳＨＴＸに入力されたｓｉｇｈａｓｈプリイメージ内のアウトポイントから抽出されている。これは、２つの値が同じである場合にＴＲＵＥと評価される。スタックにプッシュされるのが実際に現在のトランザクションの親トランザクションであることを確実にするために、ＰＵＳＨＴＸＰＡＲＥＮＴは、これからアウトポイントを抽出するためのＰＵＳＨＴＸを含む。ＰＵＳＨＴＸはすべてのアウトポイントを含むので、親トランザクションは、それが現れる入力のアウトポイントに対応しない場合であっても、プッシュされ得る。

【０１１４】

シリアル化されたトランザクションメッセージTx_nは、様々なフィールドから構築される。特に関心のある分野は次の通りである：

１．Tx_nのロックスクリプト（複数可）；および

２．アウトポイント$TxID_{n-1}$。

【０１１５】

ＰＵＳＨＴＸＰＡＲＥＮＴをＰＵＳＨＴＸと組み合わせて含めることの意味は、シリアル化された親トランザクション内の任意のフィールドが、上記のように、それらに対して適用される条件を有することができるということである。

【０１１６】

同様に、$dSHA256(Tx_{n-1}) = TxID_{n-1}$である場合、$Tx_{n-1}$および$TxID_{n-1}$（$Tx_n$から）が入力として与えられると、$Tx_n$内のロックスクリプト内のオペコードＰＵＳＨＴＸＧＲＡＮＤＰＡＲＥＮＴの組合せはＴＲＵＥと評価される。ＰＵＳＨＴＸＰＡＲＥＮＴおよびＰＵＳＨＴＸと同様に、Tx_nは、これが実際に祖父母トランザクションであることを確実にするためにＰＵＳＨＴＸＰＡＲＥＮＴに与えられる入力である。これは、リンクがＰＵＳＨＴＸＰＡＲＥＮＴ、ＰＵＳＨＴＸＧＲＡＮＤＰＡＲＥＮＴなどを介して反復的に行われる限り、任意の祖先トランザクションをプッシュするために繰り返すことができる。

【０１１７】

５．４．ＳＨＡ２５６ハッシュ関数

ハッシュ関数は、任意のサイズｍのデータを固定サイズのデータにマッピングする。入力データが４４７ビットよりも長い場合、ＳＨＡ２５６ハッシュ関数の実行は、

【数３】

Merkle-Damgård構造

を使用して、圧縮関数ＣＦ（■，■）によってデータを反復的に処理する。圧縮関数は、以下の方法でＳＨＡ２５６（ｍ）を計算するために使用される。

１．パディングされたプリイメージＰＩ＝ｍ┆┆ｐａｄを計算する。ここで、ｐａｄは、長さが５１２ビットの倍数であることを保証するメッセージ上のパディングである。

２．ＰＩを、サイズ５１２ビットのｌ個のブロックに分割し、ＰＩ＝m_1┆┆m_2┆┆…┆┆m_lとする。

３．$mid_1 = CF(m_1, IV)$を計算する。ここで、ＩＶは所定の初期化ベクトルである。

４．$mid_2 = CF(m_2, mid_1)$を計算する。

５．プリイメージ全体が処理されるまで、５１２ビットの各ｊ番目のブロック$mid_j = CF(m_j, mid_{j-1})$を続ける。

６．$CF(m_l, mid_{l-1}) = SHA256(m)$を出力する。

【０１１８】

上記のステップの計算は、ｊ番目の反復まで実行され、停止され得る。残りのプリイメージＰＰＩ＝m_{j+1}┆┆…┆┆m_lおよびミッドステート$mid = CF(m_j, mid_{j-1})$を記憶することができ、計算は後で続けることができる。さらなる「部分的ＳＨＡ」計算は、$h = partialSHA256(PPI, mid)$と表記され、ＳＨＡ２５６（

ｍ）と同じものを出力する。
【０１１９】
５．５．オペコードにおける部分的ＳＨＡ２５６
最初のｊ回の反復の計算は、オフチェーンで実行される。最後の（ｌ－ｊ）回の反復は、スクリプトで実行され、オペコードでの計算の実装を必要とする。これらのオペコードＰＡＲＴＩＡＬＳＨＡ２５６は、ｈ＝ｐａｒｔｉａｌＳＨＡ２５６（ＰＰＩ，ｍｉｄ）である場合、ｈならびに残りのプリイメージＰＰＩおよびミッドステートｍｉｄが入力として与えられるとＴＲＵＥを返す。内部で、オペコードは、ＰＰＩおよびｍｉｄが入力として与えられると、ＳＨＡ２５６ハッシュ関数の残りの反復を計算する。この結果をｈと比較する。それらが同じである場合、このオペコードのセットはＴＲＵＥを返す。
【０１２０】
このケースでは、ハッシュされ、所与のＴｘＩＤと比較されるシリアル化されたトランザクションの部分的ＳＨＡ２５６を計算することとなる。上述したように、入力は、シリアル化されたトランザクションにおいて最初に現れる。これは、部分的ＳＨＡ２５６計算を使用して、ＰＵＳＨＴＸＰＡＲＥＮＴまたはＰＵＳＨＴＸＧＲＡＮＤＰＡＲＥＮＴへの入力として必要とされるロック解除スクリプトにおけるシリアル化されたトランザクションの第１の入力を編集することができることを意味する。
【０１２１】
部分的ＳＨＡ２５６を利用して、シリアル化されたトランザクションの第１のロック解除スクリプトを編集する。上記の表記を乱用して、第１の入力の終わりまでのＳＨＡ２５６ハッシュ関数の最初のｊ回の反復の計算を意味するようにＣＦ（$us_{n,0}$，ＩＶ）と書く。言い換えると、このように書き込むことは、$us_{n,0}$がわずか５１２ビット（および、ＳＨＡ２５６ハッシュ計算の最初の反復）であることを意味しない。もっと長くなる可能性が高い。
【０１２２】
５．６．課題１：実行と伝搬の証明
最初に考察する問題は、いわゆる「バック・トゥ・ジェネシス」問題である。これは、トークンのバリデータが、受け取ったトークントランザクションが発行トランザクションに戻るトランザクションのチェーンの一部であることを証明したい場合に生じる。
【０１２３】
ＰＵＳＨＴＸを単独で使用することにより、発行者は、任意の子トランザクションを、それらの子、孫などのいずれかを含む特定のフォーマットに強制することができる。例えば、それらは、シリアル番号が将来の各トランザクションに含まれることを必要としてもよい。しかしながら、バリデータは、発行トランザクションまで遡ってすべてのトランザクションを明示的に妥当性確認しなければならない。これにより、転送のたびにプルーフ長および妥当性確認時間が増加する。
【０１２４】
各トランザクションをジェネシスまで遡って妥当性確認する必要性を回避するために、信頼できるトークンプロセッサが各転送の証人（witness）となる可能性がある。これは、バック・トゥ・ジェネシス問題を解決するが、性能のボトルネックおよびセキュリティリスクが生じるためユーザビリティが低下する。また、通信およびインフラストラクチャのコストも発生する。追加的に、これは、マイナー妥当性確認済みトークンを目指すという目的を損なう。信頼できるトークンプロセッサは、スクリプトの複雑さおよびＰＵＳＨＴＸのコストを除去して、転送全体を処理することもできる。
【０１２５】
図５に示すように、１つの入力および１つの出力を含むトランザクションの単純なチェーンを考える。
【０１２６】
Tx_{n-1}およびTx_nのすべて（このケースでは１つ）のロックスクリプトが同一である場合、すなわち、Tx_n内の第１のロックスクリプトがTx_{n-1}内の第１のロック

スクリプトと同じである場合、ｓｉｇｈａｓｈプリイメージm_{n+1}が入力として与えられるとＴＲＵＥと評価されるＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮを定義する。オペコード内の同じ関数がls_nおよびls_{n-1}で実行されることを保証するためにスクリプトスニペットが必要であるので、この実行証明（proof-of-execution）にラベル付けする。ロックスクリプトが公開鍵などの変数を含む場合、それは同じままであるとは期待されず、無視される。オペコードＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮは、以下を含む：

１．m_{n+1}がトランザクションTx_{n+1}のｓｉｇｈａｓｈプリイメージである場合、m_{n+1}を入力としてとり、ＴＲＵＥを出力するＰＵＳＨＴＸ。

２．Tx_nが$TxID_n$のハッシュプリイメージである場合、Tx_nおよびm_{n+1}からのアウトポイント$TxID_n$をとり、ＴＲＵＥを出力するＰＵＳＨＴＸＰＡＲＥＮＴ。

３．Tx_{n-1}が$TxID_{n-1}$のハッシュプリイメージである場合、Tx_nからTx_{n-1}および$TxID_{n-1}$をとり、ＴＲＵＥを出力するＰＵＳＨＴＸＧＲＡＮＤＰＡＲＥＮＴ。

４．Tx_{n-1}およびT_n内のロックスクリプトが同じオペコードを含む場合、Tx_{n-1}からのls_{n-1}内のオペコードをTx_nからのls_n内のオペコードと比較し、ＴＲＵＥを出力するオペコード。

５．親トランザクションおよび祖父母トランザクションに１つの入力および１つの出力のみが存在することを保証するオペコード。

【０１２７】

したがって、ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮを含むスクリプトは、（m_{n+1}，Tx_n，Tx_{n-1}）によってロック解除される。Tx_nおよびTx_{n+1}のすべて（このケースでは１つ）のロックスクリプトが同一である場合、すなわちTx_n内の第１のロックスクリプトがTx_{n+1}内の第１のロックスクリプトと同じである場合などに、ｓｉｇｈａｓｈプリイメージm_{n+1}が入力として与えられるとＴＲＵＥと評価されるＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮを定義する。オペコード内の同じ関数がls_nで実行され、ls_{n+1}で実行されることとなることを保証するためにスクリプトスニペットが必要であるので、伝搬証明（proof-of-propagation）にラベル付けする。ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮと同様に、ロックスクリプトが公開鍵のような変数を含む場合、それは無視される。オペコードＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮは、以下を含む：

１．m_{n+1}がトランザクションTx_{n+1}のｓｉｇｈａｓｈプリイメージである場合、m_{n+1}を入力としてとり、ＴＲＵＥを出力するＰＵＳＨＴＸ。

２．Tx_nが$TxID_n$のハッシュプリイメージである場合、Tx_nおよびm_{n+1}からのアウトポイント$TxID_n$をとり、ＴＲＵＥを出力するＰＵＳＨＴＸＰＡＲＥＮＴ。

３．Tx_nからのls_n内のオペコードをm_{n+1}からのls_{n+1}内のオペコードと比較し、第１のロックスクリプト内のオペコードが同じである場合にＴＲＵＥを出力するオペコード。

４．子トランザクションに１つの入力および１つの出力のみが存在することを保証するオペコード。

【０１２８】

したがって、ＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮを含むスクリプトは、（m_{n+1}，Tx_n）によってロック解除される。

【０１２９】

【数４】

オペコードのセットs_1（以下s_1と記す）を次のように定義する：

```
PROOFOFPROPAGATION
OP_IF
      PROOFOFEXECUTION
OP_ENDIF
```

ここで、ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮはｎ≧３の場合に実行される。どのようにこのケースをキャプチャ（ｃａｐｔｕｒｅ）するかについては後で詳細に説明する。

【０１３０】

すべてのｎ≧３に対するＴｘ$_{n}$内のロックスクリプトｌｓ$_{n}$がｓ$_{1}$である場合、ロック解除スクリプトは、ｕｓ$_{n}$＝（ｍ$_{n+1}$，Ｔｘ$_{n}$，Ｔｘ$_{n-1}$）であり、ここで、以下の通りである：

ｍ$_{n+1}$＝Ｔｘ$_{n+1}$のｓｉｇｈａｓｈプリイメージ、

Ｔｘ$_{n}$＝シリアル化された親トランザクション、および

Ｔｘ$_{n-1}$＝シリアル化された祖父母トランザクション。

【０１３１】

ｎ＝２の場合、ロック解除スクリプトは、ｕｓ$_{n}$＝（ｍ$_{2}$，Ｔｘ$_{1}$）である。Ｔｘ$_{1}$が発行であるので、Ｔｘ$_{0}$には関連するロックスクリプトはない。Ｔｘ$_{1}$は、このロックスクリプトが現れる最初のトランザクションである。

【０１３２】

ｌｓ$_{n}$内のオペコードｓ$_{1}$の組合せは、ロックスクリプトｌｓ$_{n}$内のオペコードｓ$_{1}$がｌｓ$_{n-1}$およびｌｓ$_{n+1}$にも含まれることを証明することに留意されたい。

【０１３３】

ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮおよびＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮは両方とも、それら自身も含むｓ$_{1}$の存在を探す。言い換えると、オペコードのこれらの両方のセットを含むロックスクリプトは、オペコードの両方のセットへの入力の一部である。

【０１３４】

Ｔｘ$_{n+1}$が上記のオペコードを含む場合、Ｔｘ$_{n+1}$は発行トランザクションＴｘ$_{1}$にリンクされなければならないことが帰納法によって証明される。

【０１３５】

命題：

それぞれが少なくとも１つの入力および１つの出力を有するＴｘ$_{1}$からＴｘ$_{n}$までのトランザクションのセットを考える。Ｔｘ$_{n}$は以下の条件を満たすとする：

・　Ｔｘ$_{n}$の入力のうちの１つは、上記で定義されたように、ロックスクリプトｌｓ$_{n}$にｓ$_{1}$を含むＴｘ$_{n-1}$の出力を使用する、

・　Ｔｘ$_{n}$はマイナーによって受け入れられる、

すると、Ｔｘ$_{n}$が、発行トランザクションＴｘ$_{1}$にリンクされる。

【０１３６】

証明：

この主張は帰納法によって証明される。ｎ＝２のとき、Ｔｘ$_{n}$がマイナーによって受け入れられると、それがＴｘ$_{2}$のアウトポイントによってＴｘ$_{1}$にリンクされることは明らかである。基本のケースｎ＝３：

Ｔｘ$_{2}$にはロックスクリプトｌｓ$_{2}$＝ｓ$_{1}$が含まれており、ロック解除スクリプトｕｓ$_{2}$＝（ｍ$_{3}$，Ｔｘ$_{2}$，Ｔｘ$_{1}$）はＴｘ$_{3}$内にある。Ｔｘ$_{3}$がマイナーによって受け入れられる場合、それは、ｓ$_{1}$の使用条件により、Ｔｘ$_{1}$の孫でなければならない。したがって、発行トランザクションにリンクされる。

ｎ番目のケース：

ｎ番目のケースが真であると仮定する。

ｎ＋１番目のケース：

ロックスクリプトｌｓ$_{n}$＝ｓ$_{1}$を含む出力を有するトランザクションＴｘ$_{n}$が与えられると、Ｔｘ$_{n+1}$がマイナーによって受け入れられる場合、ｌｓ$_{n}$を有する出力を使用するトランザクションＴｘ$_{n+1}$が、トランザクションのチェーンを介して発行トランザクションＴｘ$_{1}$にリンクされることが証明される。以下の３つの条件がある場合：

・　ｌｓ$_{n}$＝ｓ$_{1}$であること、

・　Ｔｘ$_{n+1}$がＴｘ$_{n}$のその出力を使用すること、および

・　Ｔｘ$_{n+1}$がマイナーによって受け入れられること、

出力をロック解除するためには、us_n＝（m_{n+1}，Tx_n，Tx_{n-1}）でなければならない。$ls_n = s_1$であるので、ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮにより$ls_{n-1} = s_1$であり、ＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮにより$ls_{n+1} = s_1$でなければならない。したがって、Tx_{n-1}は：

・ ロックスクリプト$ls_{n-1} = s_1$を含み、

・ これは、Tx_nによって使用され（us_nがls_nを使用するためのリンクが必要である）、そして、

・ Tx_nがマイナーによって受け入れられる（Tx_{n+1}がそうである場合）。

【０１３７】

したがって、Tx_nはTx_1にリンクされる。また、Tx_{n+1}はTx_nの子であるので、Tx_nにリンクされる。したがって、Tx_{n+1}はTx_1にリンクされる。

【０１３８】

図６に示すように、チェーン内の各トランザクションがすべての前のトランザクションを含み、トランザクション肥大化問題が発生することを証明する。

【０１３９】

命題：

上記の主張を満たすトランザクションTx_1からTx_nのセットを考える。ｎ番目のトランザクションTx_nは、発行Tx_1まで遡ってすべてのトランザクションを含む。

【０１４０】

証明：

トランザクションTx_nのロック解除スクリプトは、Tx_{n-1}のロックスクリプト内のオペコードs_1によって必要とされるように、その親トランザクションおよび祖父母トランザクションを含む。これは、すべてのｎ≧３に当てはまる。したがって、所与のTx_nについて、その親および祖父母もまた、それぞれの親および祖父母を含む。これは、これらのトランザクションの各々に当てはまり、したがって、Tx_nは、発行まで遡ってすべてのトランザクションを含む。

【０１４１】

５．７．課題２：部分的ＳＨＡ２５６ハッシュ関数

問題１を解く際、ＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮおよびＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮを実行するためには、Tx_nのls_n、Tx_nのアウトポイントのトランザクションＩＤ　$TxID_{n-1}$、および親トランザクションおよび祖父母トランザクションからのus_nへの入力におけるTx_{n-1}のls_{n-1}のみが必要である。親トランザクションおよび祖父母トランザクションのロック解除スクリプトは不要である。トランザクションのこれらの部分をどのように編集するかについて考える。

【０１４２】

図７に示すような、２つの入力および２つの出力を有するトランザクションのチェーンを考える。追加の入力および出力は、Ｐ２ＰＫＨ（ＰＫ）スクリプトに対応する。第１のロック解除スクリプトが編集され、ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮが第２のアウトポイントを使用して祖父母トランザクションを依然として利用できるようにするために、第２の入力および出力を導入する。以下の解決策を説明した後、ｎ番目のトランザクションが発行までのすべてのトランザクションは含まなくなる理由を明示的に示す。大まかに言えば、これは、前のトランザクションに関する情報を搬送するトランザクションの一部であるロック解除スクリプトデータが伝搬されないためである。（ＴｘＩＤと比較するために）トランザクションのハッシュをオフチェーンで計算することによって、データが伝搬されることを防止する。

【０１４３】

$TxID_n$＝ＳＨＡ２５６（ｐａｒｔｉａｌＳＨＡ２５６（PPI_n，mid_n））である場合、Tx_{n+1}における第２のアウトポイントからの$TxID_n$ならびにPPI_nおよびmid_nが入力として与えられるとＴＲＵＥと評価されるＰＵＳＨＴＸＰＡＲＴＩＡＬＰＡＲＥＮＴを定義する。

【０１４４】
ＴｘＩＤ$_{n-1}$＝ＳＨＡ２５６（ｐａｒｔｉａｌＳＨＡ２５６（ＰＰＩ$_{n-1}$，ｍｉｄ$_{n-1}$））である場合、Ｔｘ$_{n}$における第２のアウトポイントからのＴｘＩＤ$_{n-1}$ならびにＰＰＩ$_{n-1}$およびｍｉｄ$_{n-1}$が入力として与えられるとＴＲＵＥと評価されるＰＵＳＨＴＸＰＡＲＴＩＡＬＧＲＡＮＤＰＡＲＥＮＴを定義する。
【０１４５】
上述したように、部分的ＳＨＡ２５６計算は、それぞれこれらの２つのオペコードにおける親トランザクションおよび祖父母トランザクションの第１の入力の編集を可能にする。したがって、入力ｕｓ$_{n-1,0}$およびｕｓ$_{n-2,0}$は、ｕｓ$_{n,0}$に現れない。次いで、ＰＰＩ$_{n}$は、第２のロック解除スクリプトと、親トランザクションの両方のロックスクリプトとを含む。ＰＰＩ$_{n-1}$は、祖父母トランザクションの２つのロックスクリプトを含む。親トランザクションの第２のロック解除スクリプトは、祖父母トランザクションＩＤの抽出を可能にする。ロックスクリプトは、ＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮおよびＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮの実行への入力として抽出される。
【０１４６】
それぞれＰＵＳＨＴＸＰＡＲＥＮＴおよびＰＵＳＨＴＸＧＲＡＮＤＰＡＲＥＮＴの代わりに、ＰＵＳＨＴＸＰＡＲＴＩＡＬＰＡＲＥＮＴおよびＰＵＳＨＴＸＰＡＲＴＩＡＬＧＲＡＮＤＰＡＲＥＮＴを含むようにＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮおよびＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮを再定義する。再定義とは、スクリプトが、前述のように、Ｔｘ$_{n}$およびＴｘ$_{n-1}$ではなくＰＰＩ$_{n}$またはＰＰＩ$_{n-1}$から必要なロックスクリプト内のオペコードを抽出することを意味する。したがって、ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮを含むスクリプトは、（ｍ$_{n+1}$，ＰＰＩ$_{n}$，ｍｉｄ$_{n}$，ＰＰＩ$_{n-1}$，ｍｉｄ$_{n-1}$）によってロック解除され、ＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮを含むスクリプトは、（ｍ$_{n+1}$，ＰＰＩ$_{n}$，ｍｉｄ$_{n}$）によってロック解除される。
【０１４７】
追加的に、ＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮおよびＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮは、ロックスクリプトの１つのセットだけではなく、ロックスクリプトの複数のセット内のオペコードを比較するオペコードを含むように再定義される。前述のように、それは、Ｔｘ$_{n}$内の第１のロックスクリプト内のオペコードを、それぞれ、Ｔｘ$_{n+1}$またはＴｘ$_{n-1}$の第１のロックスクリプト内のオペコードとそれぞれ比較する。ここで、これらのトランザクションにおける第２のロックスクリプトのセットも比較する。これらのオペコードは、第１のロックスクリプトのセットが同じオペコードを含み、第２のロックスクリプトのセットが同じオペコードを含む場合、ＴＲＵＥを返す。これはまた、各トランザクションにおいて１つの入力および出力をチェックするのではなく、各トランザクションに２つの入力および出力のみがあることを保証することを意味する。これらのオペコードのさらなる参照は、これらの新しい定義を参照する。再定義されたオペコードでオペコードのセットをｓ$_{2}$となるように定義する：
【数５】

```
PROOFOFEXECUTION +
OP_IF
      PROOFOFPROPAGATION
OP_ENDIF
```

【０１４８】
ロックスクリプトｌｓ$_{n,0}$＝ｓ$_{2}$を、すべてのｎについて、ｕｓ$_{n,0}$＝（ｍ$_{n+1}$，ｍｉｄ$_{n}$，ＰＰＩ$_{n}$，ｍｉｄ$_{n-1}$，ＰＰＩ$_{n-1}$）でロック解除されるように定義し、ここで、以下の通りである：
ｍ$_{n+1}$＝Ｔｘ$_{n+1}$のｓｉｇｈａｓｈプリイメージ、

$mid_n = CF(us_{n-1,0}, IV)$、
$PPI_n = us_{n-1,1} \| ls_{n,0} \| ls_{n,1} \| pad_n$、
$mid_{n-1} = CF(us_{n-2,0} \| us_{n-2,1}, IV)$、
$PPI_{n-1} = ls_{n-1,0} \| ls_{n-1,1} \| pad_{n-1}$。

【０１４９】

各ロック解除スクリプトは、これらの５つの構成要素のみを含み、これらは、固定長を有することが証明される。肥大化につながるロック解除スクリプト$us_{n-1,0}$、$us_{n-2,0}$は、部分的ＳＨＡ２５６計算で編集されている。

【０１５０】

命題：

Tx_nを、２つの入力および２つの出力を有する上記で定義されたトランザクションチェーンのトランザクションとすると、Tx_nのサイズは、すべてのｎに対して定数未満であり、すなわち、ｌｅｎ（Tx_n）≦ｃｏｎｓｔである。

【０１５１】

証明：

ｌｅｎ（Tx_n）がｎに依存しないことを簡単に示す。シリアル化されたトランザクションおよびｓｉｇｈａｓｈプリイメージでは、ほとんどのフィールドが、そのサイズに関して明確な境界を有する。

【０１５２】

Ｓｉｇｈａｓｈプリイメージ：

ロックスクリプトおよびロック解除スクリプトを除くと、上限は以下となる：

４＋３２＋３２＋３２＋４＋９＋８＋４＋３２＋４＋４＝１６５バイト。

【０１５３】

本発明の解決策ではロックスクリプトを検査する必要があるので、最後の３２バイト（ハッシュ関数の出力である）をそのプリイメージで置き換える必要があることに留意されたい。これは、ロックスクリプトを除き、上限が以下であることを意味する：

４＋３２＋３２＋３２＋４＋９＋８＋４＋８ｖｏｕｔ＋４＋４＝１４９バイト。ここで、本願のケースでは、ｖｏｕｔは値２を有する。９は、ＶａｒＩｎｔであり、したがって、１～９バイトのいずれかであり得る。使用されているロックスクリプト（親ロックスクリプトls_{n-1}）および現在のトランザクション内のロックスクリプトls_n，ｖｏｕｔを除外した。

【０１５４】

明示的に、ロック解除スクリプトおよびロックスクリプトは、以下の表に与えられるサイズを有する。

【表１】

スクリプトセグメント	サブスクリプトセグメント	サブスクリプトのサイズ（バイト）	合計スクリプトサイズ（上限）
$us_{n-1,0} = \begin{pmatrix} m_n, mid_{n-1}, \\ PPI_{n-1}, mid_{n-2}, \\ PPI_{n-2} \end{pmatrix}$	m_n	$149 + len(ls_{n-1,0}) + len(ls_{n,0}) + len(ls_{n,1})$	$392 + 4len(s_2) + len(mid_{n-1}) + len(mid_{n-1})$
	mid_{n-1}	$len(mid_{n-1})$	
	PPI_{n-1}	$32 + 4 + 9 + 4 + len(us_{n-1,1}) + 2(8+9) + len(ls_{n-1,0}) + len(ls_{n-1,1})$	
	mid_{n-2}	$len(mid_{n-2})$	
	PPI_{n-2}	$2(8+9) + len(ls_{n-2,0}) + len(ls_{n-2,1})$	
$us_{n-1,1} = (sig_{n-1}, PK_{n-1})$	sig_{n-1}	71 or 72	137
	PK_{n-1}	33 or 65	
$ls_{n,0} = s_2$		$len(s_2)$	$len(s_2)$
$ls_{n,1} = P2PKH(PK_n)$		24	24

【０１５５】

ロックスクリプトおよびロック解除スクリプトを除いて、シリアル化されたトランザクションの上限は、（４＋９＋９＋４＋ｖｉｎ（３２＋４＋９＋４）＋ｖｏｕｔ（８＋９））＝１５８バイトとなる。

ここで、ｖｉｎおよびｖｏｕｔは、それぞれ入力および出力の数であり、それぞれ値２を有する。９バイトの各々がＶａｒＩｎｔフィールドを表すことに留意されたい。これらは、１バイトから９バイトの間で指定でき、入力または出力ごとに異なり得る。

【０１５６】

ロックスクリプトおよびロック解除スクリプトを含めて、上記の表を使用すると、シリアル化されたトランザクションのサイズは、ｌｅｎ（Ｔｘ$_n$）＝７１１＋５ｌｅｎ（ｓ$_2$）＋ｌｅｎ（ｍｉｄ$_{n-1}$）＋ｌｅｎ（ｍｉｄ$_{n-2}$）バイトになる。

【０１５７】

トランザクションＴｘ$_n$内のすべてのフィールドは、スクリプトｓ$_2$の長さと、親トランザクションおよび祖父母トランザクションのミッドステートの長さとに依存し、これらは両方とも一定長である。したがって、ｎに依存しないトランザクションのサイズの上限が得られる。

【０１５８】

備考：ロック解除スクリプトには重複があり、ｓ$_2$が４回出現する。これは、それを入力として一度とり、データを複製することによって軽減され得る。その結果、サイズ（７１１＋２ｌｅｎ（ｓ$_2$）＋ｌｅｎ（ｍｉｄ$_{n-1}$）＋ｌｅｎ（ｍｉｄ$_{n-2}$））バイトを

有するシリアル化されたトランザクションが得られる。

【０１５９】

現在のトランザクションから親トランザクションおよび祖父母トランザクションの第１のロック解除スクリプトを編集することによって、ｎ番目のトランザクションTx_nは、発行Tx_1までのすべてのトランザクションは含まなくなる。図８に示されるように、第１の入力には、第２のロック解除スクリプトおよび２つの前のトランザクションの２つの出力のみが含まれ、前のトランザクションに関するいかなる情報も含まれていない。

【０１６０】

トランザクション肥大化問題を再導入することになるので、$us_{n-1,0}$および$us_{n-1,1}$に対応するアウトポイントをスワップすることはできないことに留意されたい。$us_{n,0}$で編集されるためには、前のトランザクションを含むロック解除スクリプト$us_{n-1,0}$が最初に現れなければならない。

【０１６１】

肥大化問題を解決した一方で、次のような自己参照問題が発生した。圧縮関数の各ラウンドは６４バイトを処理し、オペコードで書かれなければならない。関数ＣＦ（）は、既存のオペコードに基づいて６４バイトより多くを必要とする。

【０１６２】

ロックスクリプト$ls_{n,0}$は、mid_nおよびPPI_nを入力としてとる、ｐａｒｔｉａｌＳＨＡ２５６（）として定義される複数の圧縮関数ラウンドを含む。これは、オペコード内のｐａｒｔｉａｌＳＨＡ２５６（）が、自分自身を（オペコード内に）含むPPI_nを処理しなければならないことを意味する。自己参照問題を説明するために、６５バイトで実装することができる最良のシナリオを考える。PPI_nは、６５バイトの長さを有するように、最初はＳＨＡ２５６（）を１ラウンドのみを含むと仮定する。

【０１６３】

PPI_nの６５バイトを処理するためには、PPI_n内のオペコードにＳＨＡ２５６（）が２ラウンドなければならない。したがって、PPI_nの長さは１３０バイトに増加する。この時点で、PPI_nを処理するためにはオペコード内にＳＨＡ２５６（）が３ラウンドなければならず、すなわち、PPI_nが実際には１９５バイトである。自己参照問題があることはすぐに分かる。

【０１６４】

ＰＵＳＨＴＸＰＡＲＥＮＴの長さはプリイメージ長に依存して変化しないので、これは問題ではない。オペコードＯＰ＿ＨＡＳＨ２５６は、任意の長さのプリイメージを処理することができ、依然として単一のオペコードを使用して計算することができる。

【０１６５】

５．８．解決策：補助トランザクション

図４に示されるような、Tx_{n-1}の第１の出力を使用し、Tx_nの第１の入力によって参照される補助トランザクションを導入する。補助トランザクション内のロックスクリプトは、Tx_nではなくオペコード内の部分的ＳＨＡ２５６ハッシュ関数を含む。このトランザクションを含めた結果、部分的ＳＨＡ２５６ハッシュ関数がそれ自体への入力として使用されることはない。

【０１６６】

ｄＳＨＡ２５６（Tx^A_n）＝$TxID^A_n$の場合、m_{n+1}およびTx^A_nを入力として与えられると、オペコードＰＵＳＨＴＸＡＵＸＩＬＩＡＲＹのセットはＴＲＵＥと評価され、ここで、$TxID^A_n$は、m_{n+1}の第１のアウトポイントから抽出される。

【０１６７】

オペコードＰＵＳＨＴＸＡＵＸＩＬＩＡＲＹは、以下を含む：

１．m_{n+1}を入力としてとり、m_{n+1}がトランザクションTx_{n+1}のｓｉｇｈａｓｈプリイメージである場合、ＴＲＵＥを出力するＰＵＳＨＴＸ

２．m_{n+1}からTx^A_nおよび$TxID^A_n$をとり、Tx^A_nが$TxID^A_n$のハッシュプリイメージである場合、ＴＲＵＥを出力するＰＵＳＨＴＸＰＡＲＥＮＴ

【０１６８】
したがって、ＰＵＳＨＴＸＡＵＸＩＬＩＡＲＹを含むスクリプトは、（m_{n+1}，Tx^{A}_{n}）によってロック解除される。
【０１６９】
スクリプトs_3を以下のように定義する：
【数６】

```
PUSHTXAUXILIARY                    +
PROOFOFPROPAGATION
OP_IF
     PROOFOFEXECUTION
OP_ENDIF
```

【０１７０】
図１に示されるような$ls^{A}_{n}=s_3$を有するロックスクリプトは、$us_{n,0}$＝（m_{n+1}，Tx^{A}_{n}，mid_n，PPI_n，mid_{n-1}，PPI_{n-1}）によってロック解除され、ここで、以下の通りである：
$m_{n+1}=Tx_{n+1}$のｓｉｇｈａｓｈプリイメージ、
Tx^{A}_{n}＝シリアル化された補助トランザクション、
$mid_n=partialSHA256(us_{n-1,0}, IV)$、
$PPI_n=us_{n-1,1}||ls_{n,0}||ls_{n,1}||pad_n$、
$mid_{n-1}=partialSHA256(us_{n-2,0}||us_{n-2,1}, IV)$、および
$PPI_{n-1}=ls_{n-1,0}||ls_{n-1,1}||pad_{n-1}$。
【０１７１】
また、ｈａｓｈ（ls^{A}_{n}）＝ｈａｓｈ（s_3）である場合、ls^{A}_{n}が入力として与えられるとＴＲＵＥと評価されるオペコードＨＡＳＨＰＵＺＺＬＥのセットを定義し、ここで、ｈａｓｈ（s_3）は、ＨＡＳＨＰＵＺＺＬＥ内の固定定数である。
【０１７２】
スクリプトs_Aを以下のように定義する：
【数７】

$$s_A = \text{PUSHTX} + \text{HASHPUZZLE}$$

【０１７３】
すべてのｎに対するロックスクリプト$ls_{n,0}$をs_Aであると再定義すると、ロック解除スクリプトは、$us^{A}_{n}=m^{A}_{n}$であり、ここで、m^{A}_{n}は、Tx^{A}_{n}のｓｉｇｈａｓｈプリイメージである。
【０１７４】
このロックスクリプトは、ls^{A}_{n}が上記で定義されたオペコードを含むことを強制する。s_3が$ls_{n,0}$内のハッシュパズルをロック解除することを強制することによって、ＰＲＯＯＦＯＦＥＸＥＣＵＴＩＯＮおよびＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮの定義を変更する必要はなく、補助トランザクションは、明示的にチェックしなくても各転送において正しいフォーマットを有する。すべての補助トランザクションは自動的に強制される。
【０１７５】
最後に、ロックスクリプト$ls_{n,1}$は、前述のように、Ｐ２ＰＫＨ（PK_n）である。

【０１７６】
部分的ＳＨＡ２５６ハッシュ関数は、部分プリイメージＰＰＩに含まれることはない。これは、ＯＰ＿ＨＡＳＨ２５６に入力されるls^{A}_{n}に含まれる。新たな問題が発生することなく自己参照問題が解決された。
【０１７７】
５．９．プロトコルを開始
発行トランザクションは、祖父母トランザクションを有さない。これは、s_3への修正を必要とする特殊なケースを導入する。発行トランザクションは、少なくとも２つの方法で識別され得る：
１．発行者によって所有される公開鍵からトランザクションに関する署名を妥当性確認する。
２．各トークンに一意のトークンＩＤ（ＵＴＩＤ）を与える。
【０１７８】
特定の発行者の導入を避けるために、ＵＴＩＤが発行トランザクション$Tx\mathrm{ID}_1$のＴｘＩＤであると定義される第２の方法を選択する。Tx_{n+1}内のＵＴＩＤがTx_n内に存在するとき、m_{n+1}が入力として与えられるとＴＲＵＥと評価されるオペコードＵＴＩＤＰＲＥＳＥＮＴを定義する。また、Tx_n内のＵＴＩＤと$Tx\mathrm{ID}_n$への第２のアウトポイントからの$Tx\mathrm{ID}_{n-1}$とが等価である場合、m_{n+1}が入力として与えられるとＴＲＵＥと評価されるオペコードＤＥＦＩＮＥＵＴＩＤも定義する。これは、ＵＴＩＤ＝$Tx\mathrm{ID}_1$であるので、ｎ＝２のときのみ当てはまることに留意されたい。これは、ＵＴＩＤがTx_1に存在しないことを意味する。その代わりに、それは、すべてのｎ≧２に対してTx_nのロックスクリプトに存在することが強制される。
【０１７９】
また、$us_{n,1}$へのアウトポイントおよびus^{A}_{n}へのアウトポイントが与えられると、それらが同じＴｘＩＤを含む場合に、ＴＲＵＥと評価されるＳＡＭＥＯＵＴＰＯＩＮＴも定義する。
【０１８０】
スクリプトｓを以下のように定義する：
【数８】

```
PUSHTXAUXILIARY + PROOFOFPROPAGATION + SAMEOUTPOINT +
UTIDPRESENT
OP_IF
      PROOFOFEXECUTION
OP_ELSE
      DEFINEUTID
OP_ENDIF
```

【０１８１】
ＵＴＩＤＰＲＥＳＥＮＴが実行され、１を出力する場合（すなわち、ｎ＞２のとき）、ＰＲＯＯＦＯＦＰＲＯＰＡＧＡＴＩＯＮを含む「ｉｆ」ステートメントが実行される。ＵＴＩＤＰＲＥＳＥＮＴが実行され、０を出力する場合（すなわち、ｎ＝２のとき）、ＤＥＦＩＮＥＵＴＩＤを含む「ｅｌｓｅ」ステートメントが実行される。
【０１８２】
ロック解除スクリプト$us_{n,0}$＝（m_{n+1}，Tx^{A}_{n}，mid_n，PPI_n，mid_{n-1}，PPI_{n-1}）によってロック解除される図１に示されるようなls^{A}_{n}＝ｓを用いてロックスクリプトを精緻化する。これは、他のロック解除スクリプトとともに、前のロック解除スクリプトから変化しないままである。ロックスクリプトに対する唯一の変更は、以下の通りである：

・　$ls_{n,0}$は、ハッシュパズルの固定データとしてhash（s_3）ではなくhash（s）を含む；
・　OP＿RETURN＜UTID＞を終わりにさらに含む$ls_{n,1}$。
PROOFOFPROPAGATIONおよびPROOFOFEXECUTIONの両方におけるPUSHTXおよびPUSHTXPARTIALPARENTのように、オペコードの各セットに含まれるオペコードとの重複があることに留意されたい。これらは、同じ機能を利用し、効率を向上させるための任意の必要な情報を複製することができる。

【０１８３】
親トランザクションの２つの出力$TxID_{n-1}||0$、$TxID_{n-1}||1$がそれぞれ補助トランザクションおよび現在のトランザクションで使用されることを強制する。この方法で使用されるように出力が強制されない場合、トランザクションチェーンをフォークしても、プロトコルが正しく守れられているように見える可能性がある。これにより、攻撃者が第２の出力なしに第１の出力を容易に使用することができる攻撃ベクトルが導入される可能性がある。

【０１８４】
現状では、Tx_nの第２の出力とともに使用されなければならないTx^A_nの出力がロックされるので、第１の出力なしに第２の出力を使用することによってトークンをバーンする（burn）ことが可能である。これは、（オペコード内に部分的SHA256ハッシュ関数を含む）補助トランザクションが第２のロック解除スクリプトの一部となるので、自己参照問題を再導入しない限りスクリプト内で防止することはできない。したがって、部分的SHA256ハッシュ関数は、それ自身をオペコードにおいて処理しなければならず、これは自己参照問題である。Tx_nの第２の出力を使用するためには署名が必要とされるので、トークンがこのようにしてバーンされた場合は、署名がそれを証明することとなる。

【０１８５】
また、第２の出力で補助トランザクションを使用することによってトークンをバーンすることも可能であるが、これは、トランザクションサイズの増大および部分的SHA256の自己参照問題を再導入する。しかしながら、本発明の解決策は、PROOFOFEXECUTIONがm_{n+1}内の第２のアウトポイントのみからアウトポイントを抽出することを必要とし、第２のアウトポイントが補助トランザクションでない場合にのみトランザクションが有効になるようにすることで、これを防止する。

【０１８６】
前述したように、m_{n+1}における第２のアウトポイントからアウトポイントを抽出することをPROOFOFEXECUTIONに要求することによって、Tx^A_nのアウトポイントがTx_nの第２のアウトポイントとともに使用されなければならないことを強制する。これは、Tx_nおよびTx_{n-1}内のロックスクリプトが、同じオペコードのセットと発行トランザクションのTXIDとして定義された同じUTIDとを含む場合にのみ、Tx^A_nのロックスクリプト内のオペコードがTRUEを返すからである。これは、第２のアウトポイントがTx_nの第２の出力であるときのみ当てはまり、任意の他のアウトポイントは、同じオペコードとUTIDとを含まない。したがって、Tx^A_nの出力は、Tx_nの第２の出力とともに使用されなければならず、そうでなければ無効になる。

【０１８７】
スクリプトsは、上記で説明した補助ロックスクリプトの一例である。スクリプトs_Aは、上記で説明した一次ロックスクリプトの一例である。下付き文字Aは、一次ロックスクリプトが補助トランザクションに対して条件を適用することを意味する。

【０１８８】
<u>５．１０．トークン転送プロトコル</u>
発行者Iによるアリス１０３ａへのトークンの発行と、次いで、最初にボブ１０３ｂへ転送し、そのボブ１０３ｂがチャーリー１０３ｃへ転送する２回のトークンの転送とを考える。

１．発行者は、Tx_1を生成し、ここで、$ls_{1,0}$＝ｓおよび$ls_{1,1}$＝Ｐ２ＰＫＨ（PK_A）は、セクション３．３において定義され、トークンをアリスの公開鍵に支払う。
２．アリスは、セクション３．４で定義されるような補助トランザクションTx^A_1と、セクション３．３で定義されるようなTx_2とを生成し、これらをボブに送信する。
３．ボブは、以下のステップを使用してトランザクションを妥当性確認する。
４．ボブは、Tx^A_2およびTx_3を生成し、チャーリーに送信する。
５．チャーリーは、以下のステップを使用してトランザクションを妥当性確認する。

【０１８９】

さらなる転送はチャーリーへの転送と同じ方法で実行される。

【０１９０】

５．１１．ｎ回の転送のためのバリデータステップ

１．トランザクションTx_nおよびTx_{n-1}のマークルプルーフを受信する。Tx_n内のデータを使用して$TxID_{n-1}$を計算し、マークルプルーフが有効であることを確認する。
２．Tx_{n-1}内のハッシュパズルから補助ロックスクリプトのハッシュを抽出し、それがこれの周知のハッシュに対応することを検証する。
３．ＵＴＩＤを抽出し、発行者からの有効なＵＴＩＤのリストを用いて、または発行トランザクション上の署名を用いて、それが署名されていることを確認する。

【０１９１】

バリデータが実行するステップはいくつかあるが、これは、正しいマーキングを有する場合にのみ紙幣を受け入れること、または正しい公開鍵に対するＰ２ＰＰＫを含む場合にのみブロックチェーントランザクションを受け入れることなど、内部での現金の任意の交換と同様であることに留意されたい。これは、正しく発行されたトランザクションが与えられたときのマイナー妥当性確認済みトークンであり、いかなる転送も、バリデータからの努力なしに、常に別の有効なトークンを生成しなければならない。

【０１９２】

５．１２．プロトコルの詳細な例

発行者Ｉによるアリス１０３ａへのトークンの発行と、最初にボブ１０３ｂへ転送し、そのボブ１０３ｂがチャーリー１０３ｃへ転送する２回のトークンの転送とを考える。さらなる転送はチャーリー１０３ｃへの転送と同じ方法で実行される。

【０１９３】

発行

１．発行者は、図９のようなトークンを表す、署名を必要とする任意の出力を使用するトランザクションを生成し、これをアリスに送信する。このトークンのＵＴＩＤは$TxID_1$である。このトランザクションは、アリス１０３ａが出力をロック解除することを要求するＰ２ＰＫＨ出力を含む。他方の出力は、補助トランザクションが出力内に正しい関数を含むことを強制する。

【０１９４】

トークン転送

２．アリス１０３ａは、図１０に示されるように、Tx_1の第１の出力を使用する補助トランザクションを生成する。このトランザクションは、s_AおよびＵＴＩＤが次のトランザクションに存在し、Tx_1から出力されたＰ２ＰＫＨと同じトランザクションにおいてのみ使用されることを強制する。
３．このトークンをボブ１０３ｂに転送するために、アリス１０３ａは、図１１に示すトランザクションを生成して送信する。ボブ１０３ｂは、（上記で詳述した）検証に合格した場合、これを受け入れる。
４．ボブは、図１２に示されるように、Tx_2の第２の出力を使用する補助トランザクションを生成する。
５．このトークンをチャーリー１０３ｃに転送するために、ボブ１０３ｂは、図１３に

示すトランザクションを生成してチャーリー１０３ｃに送信する。チャーリー１０３ｃは、（上記で詳述した）検証に合格した場合、これを受け入れる。

【０１９５】

さらなる転送は、図１２および図１３のトランザクションと同じフォーマットで、同じ方法で実行される。

【０１９６】

ｎ番目のトークン転送の一般的なフォーマットを図１４に示す。補助トランザクションTx^{A}_{n-1}は図１５に示される。

【０１９７】

６．さらなる備考

開示される技法の他の変形形態またはユースケースは、本明細書の開示が与えられると、当業者には明らかになり得る。本開示の範囲は、説明された実施形態によって限定されず、添付の特許請求の範囲によってのみ限定される。

【０１９８】

例えば、上記のいくつかの実施形態は、ビットコインネットワーク１０６、ビットコインブロックチェーン１５０およびビットコインノード１０４に関して説明されている。しかしながら、ビットコインブロックチェーンはブロックチェーン１５０の１つの特定の例であり、上記の説明は一般に任意のブロックチェーンに適用されてもよいことが理解されるであろう。すなわち、本発明はビットコインブロックチェーンに限定されるものではない。より一般的には、ビットコインネットワーク１０６、ビットコインブロックチェーン１５０およびビットコインノード１０４への上記の任意の参照は、それぞれブロックチェーンネットワーク１０６、ブロックチェーン１５０およびブロックチェーンノード１０４への参照に置き換えられてもよい。ブロックチェーン、ブロックチェーンネットワークおよび／またはブロックチェーンノードは、上記で説明したようなビットコインブロックチェーン１５０、ビットコインネットワーク１０６およびビットコインノード１０４の説明された特性の一部またはすべてを共有してもよい。

【０１９９】

本発明の好ましい実施形態では、ブロックチェーンネットワーク１０６はビットコインネットワークであり、ビットコインノード１０４は、ブロックチェーン１５０のブロック１５１を作成、発行、伝搬、および記憶する説明した機能を少なくともすべて実行する。これらの機能のすべてではなく１つまたはいくつかのみを実行する他のネットワークエンティティ（またはネットワーク要素）が存在し得ることは除外されない。すなわち、ネットワークエンティティは、ブロックを作成および発行することなしに、ブロックを伝搬および／または記憶する機能を実行し得る（これらのエンティティが好ましいビットコインネットワーク１０６のノードとみなされないことを想起されたい）。

【０２００】

本発明の他の実施形態では、ブロックチェーンネットワーク１０６はビットコインネットワークでなくてもよい。これらの実施形態では、ノードが、ブロックチェーン１５０のブロック１５１を作成、発行、伝搬、および記憶する機能のうちの少なくとも１つまたはすべてではないがいくつかを実行してもよいことは除外されない。例えば、それらの他のブロックチェーンネットワーク上では、「ノード」は、ブロック１５１を作成および発行はするが、それらのブロック１５１を記憶および／または他のノードへの伝搬はしないように構成されたネットワークエンティティを指すために使用され得る。

【０２０１】

さらにより一般的には、上記の「ビットコインノード」１０４という用語へのいかなる言及も、「ネットワークエンティティ」または「ネットワーク要素」という用語と置き換えラレ絵、そのようなエンティティ／要素は、ブロックを作成、発行、伝搬、および記憶する役割の一部または全部を実行するように構成される。そのようなネットワークエンティティ／要素の機能は、ブロックチェーンノード１０４を参照して上記で説明したものと同じ方法でハードウェアに実装されてもよい。

【０２０２】
上記の実施形態は、単なる例として説明されていることが理解されるであろう。より一般的には、以下のステートメントのうちの任意の１つまたは複数による方法、装置、またはプログラムが提供され得る。
【０２０３】
＜ステートメント１＞ブロックチェーントランザクションのチェーン全体にわたって伝搬されるようにロックスクリプトを強制するコンピュータ実装方法であって、方法は、第１の当事者によって実行され、
第１の出力を含む第１の一次トランザクションを生成するステップであって、第１の出力は、一次ロックスクリプトを含み、一次ロックスクリプトは、第１の補助トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第１のロック解除スクリプトが、第１の補助トランザクションに基づいて生成された第１のメッセージを含むことを検証し、ｉｉ）第１のメッセージに基づいて、第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを検証するように構成され、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第２の一次トランザクションの第１のロック解除スクリプトが、第２の一次トランザクションに基づいて生成された第２のメッセージを含むことを検証し、ｉｉ）第２の一次トランザクションの第１のロック解除スクリプトが、第１の一次トランザクションの１つまたは複数のロックスクリプトを含む第１のデータ項目を含むことを検証し、ｉｉｉ）第２のメッセージおよび第１のデータ項目に基づいて、第２の一次トランザクションの第１の出力が一次ロックスクリプトを含むことを検証するように構成される、ステップを含む方法。
【０２０４】
＜ステートメント２＞第１の補助トランザクションを生成するステップであって、第１の補助トランザクションは、第１の一次トランザクションの第１の出力を参照し、第１のメッセージを含む第１の入力と、補助ロックスクリプトを含む第１の出力とを含む、ステップと
を含む、ステートメント１に記載の方法。
【０２０５】
＜ステートメント３＞一次ロックスクリプトは、補助ロックスクリプトのハッシュを含み、第１のメッセージは、候補補助ロックスクリプトを含み、第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを上記検証することは、一次ロックスクリプトが候補補助ロックスクリプトをハッシュし、候補補助ロックスクリプトのハッシュが補助ロックスクリプトのハッシュと一致することを検証することを含む、ステートメント１またはステートメント２に記載の方法。
【０２０６】
＜ステートメント４＞第１の一次トランザクションは第２の出力を含み、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉｉｉ）第１の一次トランザクションの第２の出力が第２の一次トランザクションの第２の入力によってロック解除されることを検証するように構成される、先行するステートメントのいずれかに記載の方法。
【０２０７】
＜ステートメント５＞補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉｖ）第２の一次トランザクションの第１のロック解除スクリプトが、第１の補助トランザクションを含む第２のデータ項目を含むことを検証するように構成され、第２の出力が第２の一次トランザクションの第２の入力によってロック解除されることを上記検証することは、ａ）第２のメッセージから、第２の一次トランザクションの第２の入力によってロック解除された第２の出力のそれぞれのトランザクション識別子を抽出することと、ｂ）第２のデータ項目から、第１の補助トランザクションの第１の入力によってロック解除された第１の出力のそれぞれのトランザク

ション識別子を抽出することと、ｃ）ステップａ）の間に抽出されたそれぞれのトランザクション識別子が、ステップｂ）の間に抽出されたそれぞれのトランザクション識別子と一致することを検証することとを含む、ステートメント４に記載の方法。

【０２０８】

＜ステートメント６＞第１の一次トランザクションは、ハッシュ関数を第１の一次トランザクションに適用することによって生成するそれぞれのトランザクション識別子を有し、第２の一次トランザクションの第１のロック解除スクリプトは、第３のデータ項目を含み、第３のデータ項目は、第１の一次トランザクションに適用されたハッシュ関数のミッドステートを含み、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉｖ）第１のデータ項目および第３のデータ項目に基づいて候補トランザクション識別子を生成し、候補トランザクション識別子が第１の一次トランザクション識別子のそれぞれのトランザクション識別子と一致することを検証するように構成される、先行するステートメントのいずれかに記載の方法。

【０２０９】

＜ステートメント７＞補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｖ）第２の一次トランザクションが特定のトランザクション識別子を含むことを検証するように構成される、先行するステートメントのいずれかに記載の方法。

【０２１０】

＜ステートメント８＞第２の一次トランザクションを生成するステップであって、第２の一次トランザクションは、第１の補助トランザクションの第１の出力を参照し、第２のメッセージおよび第１のデータ項目を含む第１の入力を含み、第２の一次トランザクションは、一次ロックスクリプトを含む第１の出力を含む、ステップ

を含む、先行するステートメントのいずれかに記載の方法。

【０２１１】

＜ステートメント９＞第１の補助トランザクションおよび第２の一次トランザクションを第２の当事者に送信するステップを含む、先行するステートメントのいずれかに記載の方法。

【０２１２】

＜ステートメント１０＞第１の一次トランザクションがブロックチェーンのブロックに記録されたことを証明するためのマークルプルーフを第２の当事者に送信するステップを含む、ステートメント９に記載の方法。

【０２１３】

＜ステートメント１１＞一次ロックスクリプトおよび／または補助ロックスクリプトは、トークンの所有権に関するデータを含む、先行するステートメントのいずれかに記載の方法。

【０２１４】

＜ステートメント１２＞第１の一次トランザクションは、ブロックチェーントランザクションのチェーンにおける最初の一次トランザクションである、先行するステートメントのいずれかに記載の方法。

【０２１５】

＜ステートメント１３＞第１の当事者は、トークンの発行者である、ステートメント１１およびステートメント１２に記載の方法。

【０２１６】

＜ステートメント１４＞特定のトランザクション識別子は、第１の一次トランザクションのそれぞれのトランザクション識別子である、ステートメント７に従属するときのステートメント１２またはステートメント１３に記載の方法。

【０２１７】

＜ステートメント１５＞第１の一次トランザクションは第１のカウンタ値を含み、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによっ

て実行されると、第２の一次トランザクションが第２のカウンタ値を含むことを検証するように構成され、第２のカウンタ値はカウンタ値の反復である、先行するステートメントのいずれかに記載の方法。

【０２１８】

＜ステートメント１６＞第１の一次トランザクションは、ブロックチェーントランザクションのチェーンにおける初期トランザクションではなく、第１の一次トランザクションは、補助ロックスクリプトを含む前の補助トランザクションの第１の出力を参照する第１の入力を含み、前の補助トランザクションは、一次ロックスクリプトを含む前の一次トランザクションの第１の出力を参照する第１の入力を含み、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｖｉ）第２の一次トランザクションの第１のロック解除スクリプトが、前の一次トランザクションの１つまたは複数のロックスクリプトを含む第４のデータ項目を含むことを検証し、ｖｉｉ）第２のメッセージおよび第４のデータ項目に基づいて、第２の一次トランザクションの第１の出力が一次ロックスクリプトを含むことを検証するように構成される、ステートメント１から１１のいずれかに記載の方法。

【０２１９】

＜ステートメント１７＞前の一次トランザクションは、前の一次トランザクションにハッシュ関数を適用することによって生成するそれぞれのトランザクション識別子を有し、第２の一次トランザクションの第１のロック解除スクリプトは、第５のデータ項目を含み、第５のデータ項目は、前の一次トランザクションに適用されたハッシュ関数のミッドステートを含み、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｖｉｉｉ）第４のデータ項目および第５のデータ項目に基づいて候補トランザクション識別子を生成し、候補トランザクション識別子が前の一次トランザクション識別子のそれぞれのトランザクション識別子と一致することを検証するように構成される、ステートメント１６に記載の方法。

【０２２０】

＜ステートメント１８＞前の一次トランザクションは、ブロックチェーントランザクションのチェーンにおける最初の一次トランザクションであり、方法は、最初の一次トランザクションが所定の公開鍵に対応する署名で署名されたことを検証するステップを含む、ステートメント１６またはステートメント１７に記載の方法。

【０２２１】

＜ステートメント１９＞第１の一次トランザクションをブロックチェーンネットワークにサブミットするステップを含む、先行するステートメントのいずれかに記載の方法。

【０２２２】

＜ステートメント２０＞第１の補助トランザクションおよび／または第２の一次トランザクションをブロックチェーンネットワークにサブミットするステップを含む、先行するステートメントのいずれかに記載の方法。

【０２２３】

＜ステートメント２１＞ロックスクリプトがブロックチェーントランザクションのチェーン全体にわたって強制されたことを検証するコンピュータ実装方法であって、方法は、第１の当事者によって実行され、

第２の一次トランザクションを取得するステップであって、第２の一次トランザクションは、第１の入力および第１の出力を含み、第１の入力は、第２の一次トランザクションに基づいて生成された第２のメッセージを含み、第１の補助トランザクションは、第１の一次トランザクションの第１の出力をロック解除する第１の入力を含み、第１のデータ項目は、第１の一次トランザクションの１つまたは複数のロックスクリプトを含み、第１の一次トランザクションの第１の出力は、一次ロックスクリプトを含み、第１の補助トランザクションの第１の出力は、補助ロックスクリプトを含み、第２の一次トランザクションの第１の出力は、一次ロックスクリプトを含む、ステップと、

第１の一次トランザクションがブロックチェーンに記録されたことを検証するステップ

と、
第１の一次トランザクションの一次ロックスクリプトが、第１の補助トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第１のロック解除スクリプトが、第１の補助トランザクションに基づいて生成された第１のメッセージを含むことを検証し、ｉｉ）第１のメッセージに基づいて、第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを検証するように構成されることを検証するステップであって、補助ロックスクリプトは、第２の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｉ）第１のロック解除スクリプトが第２のメッセージを含むことを検証し、ｉｉ）第１のロック解除スクリプトが第１のデータ項目を含むことを検証し、ｉｉｉ）第２のメッセージおよび第１のデータ項目に基づいて、第２の一次トランザクションの第１の出力が一次ロックスクリプトを含むことを検証するように構成され、それによって、一次ロックスクリプトが第１の一次トランザクションおよび第２の一次トランザクション全体にわたって強制されたことを検証する、ステップと
を含む方法。
【０２２４】
＜ステートメント２２＞一次ロックスクリプトは、補助ロックスクリプトのハッシュを含み、第１のメッセージは、候補補助ロックスクリプトを含み、一次ロックスクリプトは、第１の補助トランザクションの第１のロック解除スクリプトによって実行されると、候補補助ロックスクリプトをハッシュし、候補補助ロックスクリプトのハッシュが補助ロックスクリプトのハッシュと一致することを検証することによって、第１の補助トランザクションの第１の出力が補助ロックスクリプトを含むことを検証するように構成され、一次ロックスクリプトがステップｉ）およびｉｉ）を実行するように構成されることを検証する上記ステップは、ハッシュパズルの一部として一次ロックスクリプトが補助ロックスクリプトのハッシュを含むことを検証するステップを含む、ステートメント２１に記載の方法。
【０２２５】
＜ステートメント２３＞第１の一次トランザクションがブロックチェーンに記録されたことを検証する上記ステップは、第１の一次トランザクションがブロックチェーンのブロックに記録されたことを証明するためのマークルプルーフを取得するステップを含む、ステートメント２１またはステートメント２２に記載の方法。
【０２２６】
＜ステートメント２４＞第１の一次トランザクションは、ハッシュ関数を第１の一次トランザクションに適用することによって生成するそれぞれのトランザクション識別子を有し、第２の一次トランザクションの第１の入力は、第３のデータ項目を含み、第３のデータ項目は、第１の一次トランザクションに適用されたハッシュ関数のミッドステートを含み、第１の一次トランザクションがブロックチェーンに記録されたことを検証する上記ステップは、第１のデータ項目および第３のデータ項目に基づいて候補トランザクション識別子を生成するステップと、マークルプルーフが候補トランザクション識別子に対して有効であることを検証するステップとを含む、ステートメント２３に記載の方法。
【０２２７】
＜ステートメント２５＞第１の一次トランザクションを取得するステップを含み、第１の一次トランザクションがブロックチェーンに記録されたことの上記検証は、取得されたマークルプルーフおよび取得された第１の一次トランザクションに基づく、ステートメント２３またはステートメント２４に記載の方法。
【０２２８】
＜ステートメント２６＞第２の一次トランザクションが最初の一次トランザクションのそれぞれのトランザクション識別子を含むことを検証するステップを含む、ステートメント１９またはそれに従属する任意のステートメントに記載の方法。
【０２２９】
＜ステートメント２７＞第２の補助トランザクションを生成するステップであって、第

２の補助トランザクションは、第２の一次トランザクションの第１の出力を参照し、第２の補助トランザクションに基づいて生成された第３のメッセージを含む第１の入力と、補助ロックスクリプトを含む第１の出力とを含む、ステップと、
　第３の一次トランザクションを生成するステップであって、第３の一次トランザクションは、第２の補助トランザクションの第１の出力を参照し、第３の一次トランザクションに基づいて生成された第４のメッセージ、第１のデータ項目、第２の補助トランザクション、および第２の一次トランザクションの１つまたは複数のロックスクリプトを含む第６のデータ項目を含む第１の入力を含み、第３の一次トランザクションは、一次ロックスクリプトを含む第１の出力を含む、ステップと
　を含む、ステートメント１９またはそれに従属する任意のステートメントに記載の方法。
【０２３０】
　＜ステートメント２８＞補助ロックスクリプトは、第３の一次トランザクションの第１のロック解除スクリプトによって実行されると、ｖｉ）第２の一次トランザクションの第１のロック解除スクリプトが第３のデータ項目を含むことを検証し、ｖｉｉ）第３のメッセージおよび第６のデータ項目に基づいて、第３の一次トランザクションの第１の出力が一次ロックスクリプトを含むことを検証するように構成される、ステートメント２７に記載の方法。
【０２３１】
　＜ステートメント２９＞第２の補助トランザクションおよび第３の一次トランザクションを第三者に送信するステップを含む、ステートメント２７またはステートメント２８に記載の方法。
【０２３２】
　＜ステートメント３０＞第２の一次トランザクションがブロックチェーンのブロックに記録されたことを証明するためのマークルプルーフを第三者に送信するステップを含む、ステートメント２９に記載の方法。
【０２３３】
　＜ステートメント３１＞一次ロックスクリプトおよび／または補助ロックスクリプトは、トークンの所有権に関するデータを含み、第１の一次トランザクションおよび／または第１の補助ロックスクリプトは、トークン発行者によって生成されている、ステートメント２１またはそれに従属する任意のステートメントに記載の方法。
【０２３４】
　＜ステートメント３２＞第２の補助トランザクションおよび第３の一次トランザクションをブロックチェーンにサブミットすることを含む、ステートメント２７またはそれに従属する任意のステートメントに記載の方法。
【０２３５】
　＜ステートメント３３＞コンピュータ機器であって、
　１つまたは複数のメモリユニットを備えるメモリと、
　１つまたは複数の処理ユニットを備える処理装置であって、メモリは、処理装置上で実行されるように構成されたコードを記憶し、コードは、処理装置上にあるときに、ステートメント１から２９のいずれかに記載の方法を実行するように構成される、処理装置と
　を備えるコンピュータ機器。
【０２３６】
　＜ステートメント３４＞コンピュータ可読ストレージ上に具現化され、１つまたは複数のプロセッサ上で実行されると、ステートメント１から３２のいずれかに記載の方法を実行するように構成されたコンピュータプログラム。
【０２３７】
　本明細書で開示される別の態様によれば、第１の当事者および第２の当事者のアクションを含む方法が提供され得る。本明細書に開示される別の態様によれば、少なくとも第１の一次トランザクションおよび第１の補助トランザクションを処理するように構成された

ブロックチェーンノードのアクションを含む方法が提供され得る。

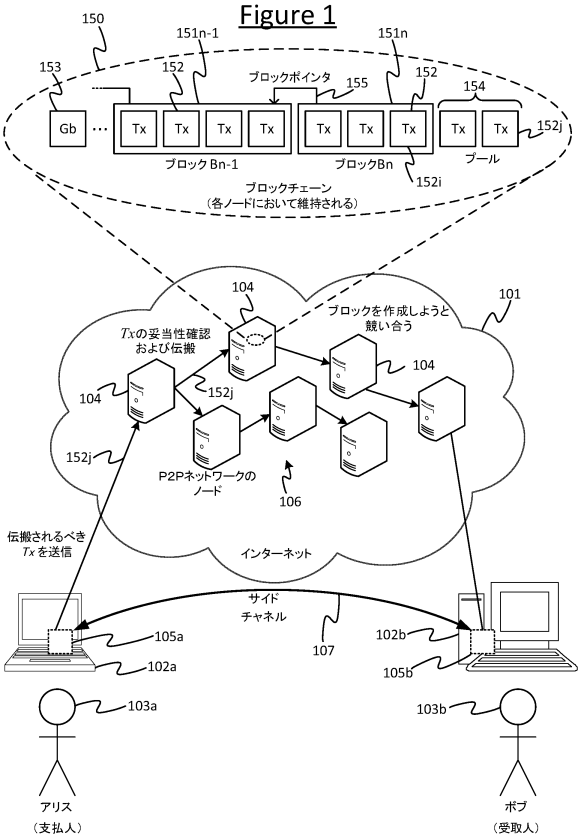
【０２３８】

本明細書で開示される別の態様によれば、第１の当事者および第２の当事者のコンピュータ機器を備えるシステムが提供され得る。本明細書に開示される別の態様によれば、少なくとも第１の一次トランザクションおよび第１の補助トランザクションを処理するように構成されたブロックチェーンノードが提供され得る。

【図面】

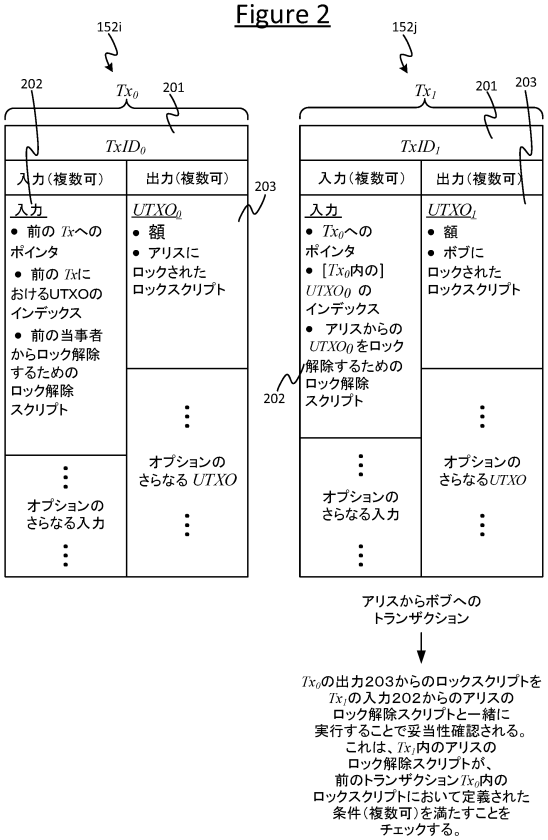
【図１】

Figure 1



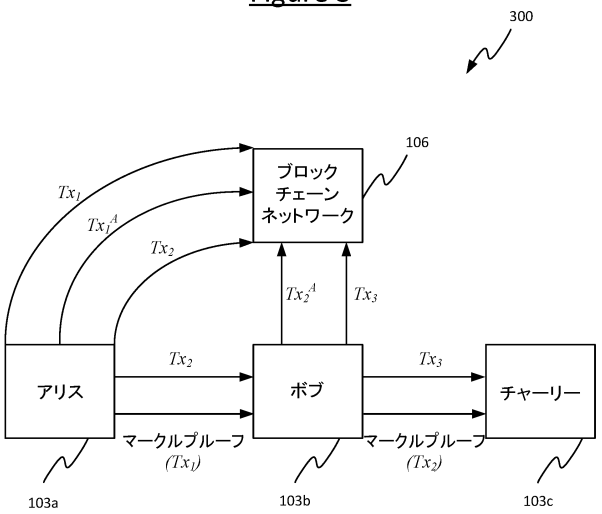
【図２】

Figure 2



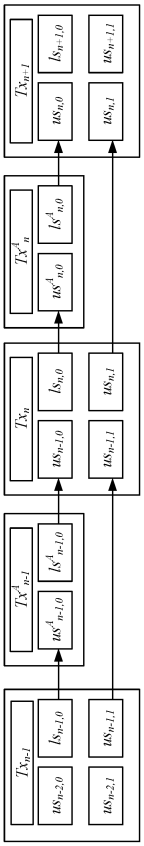
【図３】

Figure 3



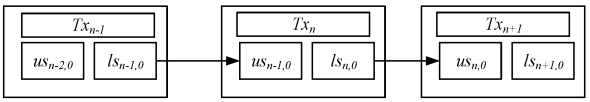
【図４】

Figure 4



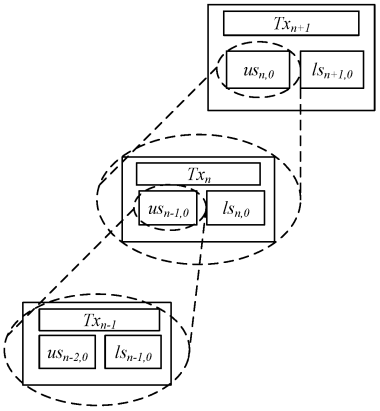
【図５】

Figure 5



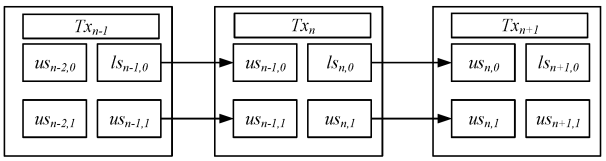
【図６】

Figure 6



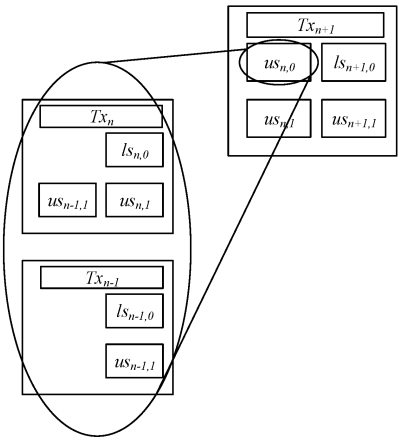
【図７】

Figure 7



【図８】

Figure 8



【図９】

Figure 9

Tx_1		
入力		出力
$TxID_0 \| \| i$	sig_{P_I}, P_I	s_A
		P2PKH(P_A)

【図１０】

Figure 10

Tx_1^A		
入力		出力
$TxID_1 \| \| 0$	m_1^A	s

【図１１】

Figure 11

Tx_2		
入力		出力
$TxID_1^A \|\|0$	mid_1+PPI_1, Tx_1^A, m_2	$\mathcal{S}_A$
$TxID_1 \|\|1$	sig_{P_A}, P_A	P2PKH(P_B) OP_RETURN <$UTID$>

【図１２】

Figure 12

Tx_2^A		
入力		出力
$TxID_2 \|\|0$	m_2^A	$\mathcal{S}$

【図１３】

Figure 13

Tx_3		
入力		出力
$TxID_2^A \|\|0$	mid_1+PPI_1, mid_2+PPI_2, Tx_2^A, m_3	$\mathcal{S}_A$
$TxID_2 \|\|1$	sig_{P_B}, P_B	P2PKH(P_C) OP_RETURN <$UTID$>

【図１４】

Figure 14

Tx_n		
入力		出力
$TxID_{n-1}^A \|\|0$	mid_{n-2}+PPI_{n-2}, mid_{n-1}+PPI_{n-1}, Tx_{n-1}^A, m_n	$\mathcal{S}_A$
$TxID_{n-1} \|\|1$	$sig_{P_{N-1}}, P_{N-1}$	P2PKH(P_N) OP_RETURN <$UTID$>

【図１５】

Figure 15

Tx_{n-1}^A		
入力		出力
$TxID_n \|\|0$	m_{n-1}^A	$\mathcal{S}$

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/074013

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2020/099529 A1 (CHAN YING [GB] ET AL) 26 March 2020 (2020-03-26) abstract paragraphs [0301] - [0305] ----- -/--	1-34

[x] Further documents are listed in the continuation of Box C.

[x] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 December 2022	15/12/2022

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Di Felice, M

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/074013

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DRAGOS IOAN ILIE ET AL: "Bitcoin CryptoBounties for Quantum Capable Adversaries", IACR, INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH , vol. 20200218:090452 14 February 2020 (2020-02-14), pages 1-16, XP061035138, Retrieved from the Internet: URL:http://eprint.iacr.org/2020/186.pdf [retrieved on 2020-02-14] abstract Chapter 1: "Introduction" Chapter 4: "Context & Modelling" -----	1-34
A	CN 112 102 091 A (NANJING TRUSTED BLOCKCHAIN AND ALGORITHMIC ECONOMIC RES INSTITUTE CO L) 18 December 2020 (2020-12-18) abstract "Summary of the invention" -----	1-34

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2022/074013

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2020099529 A1	26-03-2020	CN 110651289 A	03-01-2020
		CN 110663053 A	07-01-2020
		CN 110663054 A	07-01-2020
		CN 110709871 A	17-01-2020
		CN 110709872 A	17-01-2020
		CN 110709873 A	17-01-2020
		EP 3631723 A1	08-04-2020
		EP 3631724 A1	08-04-2020
		EP 3631725 A1	08-04-2020
		EP 3631726 A1	08-04-2020
		EP 3631727 A1	08-04-2020
		EP 3631728 A1	08-04-2020
		JP 7071403 B2	18-05-2022
		JP 7130673 B2	05-09-2022
		JP 2020521370 A	16-07-2020
		JP 2020521371 A	16-07-2020
		JP 2020521373 A	16-07-2020
		JP 2020522046 A	27-07-2020
		JP 2020522047 A	27-07-2020
		JP 2020525880 A	27-08-2020
		JP 2022169691 A	09-11-2022
		KR 20200010286 A	30-01-2020
		KR 20200010287 A	30-01-2020
		KR 20200010288 A	30-01-2020
		KR 20200010289 A	30-01-2020
		KR 20200011435 A	03-02-2020
		KR 20200011949 A	04-02-2020
		SG 10202112659X A	30-12-2021
		SG 10202112667U A	30-12-2021
		SG 10202112668X A	30-12-2021
		TW 201901546 A	01-01-2019
		US 2020099528 A1	26-03-2020
		US 2020099529 A1	26-03-2020
		US 2020143339 A1	07-05-2020
		US 2020160327 A1	21-05-2020
		US 2020175511 A1	04-06-2020
		US 2021184849 A1	17-06-2021
		WO 2018215871 A1	29-11-2018
		WO 2018215872 A1	29-11-2018
		WO 2018215873 A1	29-11-2018
		WO 2018215874 A1	29-11-2018
		WO 2018215875 A1	29-11-2018
		WO 2018215876 A1	29-11-2018
CN 112102091 A	18-12-2020	NONE	

Form PCT/ISA/210 (patent family annex) (April 2005)

フロントページの続き

MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CV,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IQ,IR,IS,IT,JM,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

ーピー ロンドン マーケット プレイス ３０ エヌチェーン ライセンシング アーゲー 内