(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)公表番号
特表2023-539431
(P2023-539431A)
(43)公表日　令和5年9月14日(2023.9.14)

(51)国際特許分類　F I
H04L 9/32 (2006.01)　H04L 9/32 200B
H04L 9/32 200Z

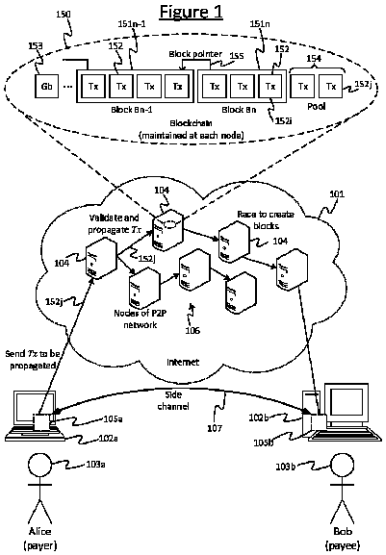
審査請求　未請求　予備審査請求　未請求　（全43頁）

(21)出願番号　特願2023-507540(P2023-507540)
(86)(22)出願日　令和3年7月19日(2021.7.19)
(85)翻訳文提出日　令和5年2月3日(2023.2.3)
(86)国際出願番号　PCT/EP2021/070105
(87)国際公開番号　WO2022/037868
(87)国際公開日　令和4年2月24日(2022.2.24)
(31)優先権主張番号　2012873.2
(32)優先日　令和2年8月18日(2020.8.18)
(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　100070150
弁理士　伊東 忠彦
(74)代理人　100135079
弁理士　宮崎 修
(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
最終頁に続く

(54)【発明の名称】　デジタル署名

(57)【要約】
デジタル署名を生成するコンピュータ実装方法であって、方法は、署名当事者によって実行され、第１のメッセージを取得することと、少なくとも外部データ項目のハッシュに基づいてエフェメラル秘密鍵を生成することと、第１の署名構成要素および第２の署名構成要素を含む第１の署名を生成することとを含み、第１の署名構成要素は、エフェメラル秘密鍵に対応するエフェメラル公開鍵に基づいて生成され、第２の署名構成要素は、第１のメッセージと、エフェメラル秘密鍵と、第１の署名構成要素と、第１の秘密鍵とに基づいて生成される、方法。



【特許請求の範囲】
【請求項１】
デジタル署名を生成するコンピュータ実装方法であって、前記方法は、署名当事者によって実行され、
第１のメッセージを取得することと、
少なくとも外部データ項目のハッシュに基づいてエフェメラル秘密鍵を生成することと、
第１の署名構成要素および第２の署名構成要素を含む第１の署名を生成することと
を含み、前記第１の署名構成要素は、前記エフェメラル秘密鍵に対応するエフェメラル公開鍵に基づいて生成され、前記第２の署名構成要素は、前記第１のメッセージと、前記エフェメラル秘密鍵と、前記第１の署名構成要素と、第１の秘密鍵とに基づいて生成される、方法。
【請求項２】
前記署名当事者が前記第１の署名を生成したことを証明するために、前記外部データ項目および前記第１の署名を検証当事者が利用できるようにすることを含む、請求項１に記載の方法。
【請求項３】
前記第１のメッセージを取得することは、前記第１のメッセージを生成することを含み、前記方法は、前記第１のメッセージを前記検証当事者が利用できるようにすることを含む、請求項２に記載の方法。
【請求項４】
第２のメッセージを取得することと、
少なくとも前記第２のメッセージと前記署名当事者の主秘密鍵とに基づいて第２の署名を生成することと
を含み、前記外部データ項目は前記第２の署名を含む、請求項１から３のいずれか一項に記載の方法。
【請求項５】
前記主秘密鍵に対応する主公開鍵を使用して検証されたときに、前記第２の署名が前記第２のメッセージの有効な署名であることを証明するために、前記第２の署名および前記第２のメッセージを利用できるようにすることを含む、請求項４に記載の方法。
【請求項６】
前記第２のメッセージは、前記第１のメッセージに基づいて生成される、請求項４または５に記載の方法。
【請求項７】
前記主秘密鍵に対応する前記主公開鍵は、前記署名当事者のアイデンティティにリンクされる、請求項４から６のいずれか一項に記載の方法。
【請求項８】
前記第１の秘密鍵は、少なくとも前記主秘密鍵に基づいて生成される、請求項４から７のいずれか一項に記載の方法。
【請求項９】
前記主秘密鍵は、階層的決定論的鍵構造のマスタ秘密鍵であり、前記ＨＤ鍵構造は、前記マスタ秘密鍵に基づいて生成された子秘密鍵のセットを含み、前記第１の秘密鍵は、前記子秘密鍵のセットのうちの１つである、請求項８に記載の方法。
【請求項１０】
前記第１の秘密鍵は、前記主秘密鍵と、前記署名当事者と第２の当事者の両方に知られている共通シークレットとに基づいて生成される、請求項８に記載の方法。
【請求項１１】
前記共通シークレットは、前記署名当事者の前記主秘密鍵と、前記第２の当事者の主秘密鍵に対応する主公開鍵とに基づいて生成される、請求項１０に記載の方法。
【請求項１２】

前記第２の当事者の前記主公開鍵は、前記第２の当事者のアイデンティティにリンクされる、請求項１１に記載の方法。
【請求項１３】
前記外部データ項目の前記ハッシュは、前記外部データ項目のダブルハッシュである、請求項１から１２のいずれか一項に記載の方法。
【請求項１４】
前記第２の署名構成要素は、前記第１のメッセージのハッシュまたはダブルハッシュに基づいて生成される、請求項１から１３のいずれか一項に記載の方法。
【請求項１５】
前記第１のエフェメラル秘密鍵は、ランダムソルト値に基づいて生成される、請求項１から１４のいずれか一項に記載の方法。
【請求項１６】
前記ランダムソルト値は秘密鍵であり、前記方法は、
第３のメッセージを取得することと、
少なくとも前記ランダムソルト値と前記第３のメッセージとに基づいて第３の署名を生成することと、
前記ランダムソルト値に対応する前記公開鍵を使用して検証されたときに、前記第３の署名が前記第３のメッセージの有効な署名であることを証明するために、前記第３の署名、前記第３のメッセージ、および前記ランダムソルト値に対応する公開鍵を前記検証当事者が利用できるようにすることと
を含む、請求項１５に記載の方法。
【請求項１７】
前記第３のメッセージは前記第２のメッセージを含む、請求項１６に記載の方法。
【請求項１８】
前記第１のメッセージは、ブロックチェーントランザクションの少なくとも一部を含む、請求項１から１７のいずれか一項に記載の方法。
【請求項１９】
前記第１のメッセージを前記検証当事者が利用できるようにすることは、前記ブロックチェーントランザクションを前記ブロックチェーンネットワークに送信することを含む、請求項１８に記載の方法。
【請求項２０】
前記第２のメッセージは、前記ブロックチェーンに関するデータを含む、請求項１８または１９に記載の方法。
【請求項２１】
デジタル署名が署名当事者によって生成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、検証当事者によって実行され、
第１の署名構成要素および第２の署名構成要素を含む第１の署名を取得することと、
前記署名当事者から候補の外部データ項目を取得することと、
前記候補の外部データ項目のハッシュに基づいて候補のエフェメラル秘密鍵を生成することと、
少なくとも前記候補のエフェメラル秘密鍵に対応する公開鍵に基づいて候補の第１の署名構成要素を生成することと、
前記候補の第１の署名構成要素が前記第１の署名構成要素に対応するかどうかに基づいて、前記第１の署名が前記署名当事者によって生成されたことを検証することと
を含む方法。
【請求項２２】
前記候補の外部データ項目は、第２の署名である、請求項２１に記載の方法。
【請求項２３】
第２のメッセージを取得することと、
前記署名当事者の主秘密鍵に対応する主公開鍵を取得することと、

前記主公開鍵を使用して検証されたときに、前記第２の署名が前記第２のメッセージの有効な署名であることを検証することと
を含む、請求項２２に記載の方法。
【請求項２４】
ランダムソルト値に対応する公開鍵を取得することを含み、前記候補の第１の署名構成要素は、前記ランダムソルト値に対応する前記公開鍵に基づいて生成される、請求項２１から２３のいずれか一項に記載の方法。
【請求項２５】
第３のメッセージを取得することと、
第３の署名を取得することと、
前記ランダムソルト値に対応する前記公開鍵を使用して検証されたときに、前記第３の署名が前記第３のメッセージの有効な署名であることを検証することと
を含む、請求項２４に記載の方法。
【請求項２６】
前記第１の署名は、第１のメッセージに署名し、前記方法は、
前記第１の署名を生成するために使用された秘密鍵に対応する第１の公開鍵を取得することと、
前記第１の公開鍵を使用して検証されたときに、前記第１の署名が前記第１のメッセージの有効な署名であることを検証することと
を含む、請求項２１から２５のいずれか一項に記載の方法。
【請求項２７】
前記第１の署名、前記第２の署名、および前記第３の署名のうちの１つ、いくつか、またはすべては、前記署名当事者から受信される、請求項２１から２６のいずれか一項に記載の方法。
【請求項２８】
前記第１のメッセージ、前記第２のメッセージ、および前記第３のメッセージのうちの１つ、いくつか、またはすべては、前記署名当事者から受信される、請求項２１から２７のいずれか一項に記載の方法。
【請求項２９】
前記第１のメッセージ、前記第２のメッセージ、および前記第３のメッセージのうちの１つ、いくつか、またはすべては、前記検証当事者によって生成される、請求項２１から２７のいずれか一項に記載の方法。
【請求項３０】
前記第１のメッセージは、ブロックチェーントランザクションの少なくとも一部を含む、請求項２１から２９のいずれか一項に記載の方法。
【請求項３１】
前記第１のメッセージを取得することは、前記ブロックチェーンから前記ブロックチェーントランザクションを取得することを含む、請求項３０に記載の方法。
【請求項３２】
前記第１の署名を取得することは、前記ブロックチェーントランザクションから前記第１の署名を抽出することを含む、請求項３０または３１に記載の方法。
【請求項３３】
前記第２のメッセージは、前記ブロックチェーンに関するデータを含み、前記方法は、前記ブロックチェーンに関する前記データを検証することを含む、請求項２３またはそれに従属する請求項のいずれか一項に記載の方法。
【請求項３４】
前記ブロックチェーントランザクションの入力は、前記第１の署名を含み、前記方法は、
前記ブロックチェーントランザクションの前記入力によって参照される前のブロックチェーントランザクションの出力が署名検証スクリプトを含むことを検証すること

を含む、請求項３０またはそれに従属する請求項のいずれか一項に記載の方法。
【請求項３５】
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と
を備え、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３４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、
コンピュータ機器。
【請求項３６】
コンピュータ可読ストレージ上に具現化されたコンピュータプログラムであって、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３４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、デジタル署名を生成する方法に関する。特に、本方法は、外部データがデジタル署名内に埋め込まれることを可能にする。
【背景技術】
【０００２】
公開鍵暗号法は、鍵のペア、すなわち、秘密鍵の所有者のみに知られている秘密鍵と、対応する秘密鍵に基づいて生成され、秘密鍵のセキュリティを損なうことなく配布され得る公開鍵とを使用する暗号システムの一種である。
【０００３】
公開鍵暗号法は、送信者が受信者の公開鍵（すなわち、受信者のみに知られている秘密鍵に対応する公開鍵）を使用してメッセージを暗号化することを可能にする。そのため、暗号化されたメッセージは、受信者の秘密鍵を使用してのみ復号することができる。
【０００４】
同様に、送信者は、例えば、メッセージが送信者によって送信されていることを証明するために、および／または、送信者がメッセージに同意することを示すために、自身の秘密鍵を使用してメッセージに署名することができる。署名者（すなわち、署名を生成する当事者）は、自身の秘密鍵を使用して、メッセージに基づいてデジタル署名を作成する。メッセージに基づいてデジタル署名を作成することは、メッセージおよび秘密鍵の両方に基づいて署名を生成する関数に、メッセージおよび秘密鍵を供給することを意味する。署名は、メッセージに追加される（例えば、タグ付けされる）か、または他の方法でメッセージに関連付けられる。署名者の対応する公開鍵を有する者であれば、同じメッセージおよびメッセージ上のデジタル署名を使用して、署名が有効に作成されたかどうか、すなわち、署名が実際に署名者の秘密鍵を使用して作成されたかどうかを検証することができる。デジタル署名は、メッセージの真正性を保証するだけでなく、メッセージの完全性および否認防止も保証する。すなわち、デジタル署名は、メッセージが署名で署名されてから変更されていないこと、および署名の作成者が署名を作成したことを将来否定することができないことを証明するために使用することができる。
【０００５】
デジタル署名方式は、典型的に、３つの手順、すなわちアルゴリズムを含む。鍵生成アルゴリズムは、ランダムな秘密鍵および対応する公開鍵を生成するために使用される。署名アルゴリズムは、メッセージと秘密鍵とに基づいて署名を生成するために使用される。検証アルゴリズムは、公開鍵およびメッセージが与えられた場合に、署名が、対応する秘密鍵を使用して、および、署名アルゴリズムにしたがって生成されたかどうかを検証するために使用される。
【発明の概要】

【０００６】
署名は、秘密鍵の所有者が所与のメッセージを証明することを可能にし、署名は、秘密鍵に対応する公開鍵を使用して検証される。例えば、ほとんどのブロックチェーンプロトコルによれば、特定の公開鍵に対応する秘密鍵を用いて作成された有効な署名を要求することによって、ビットコインまたは他のタイプのデジタル資産がロックされ得る。セキュリティ上の理由からブロックチェーントランザクションをリンクさせないために、同じ秘密鍵を複数回使用しないことが推奨される。これは、ユーザが、複数の異なる署名を作成するために使用する複数の異なる鍵を有する可能性が高いことを意味する。多くの異なる鍵を使用すると、ユーザが任意の１つの署名を生成したことを証明することが困難になる。したがって、ユーザが署名を生成したことを証明することができ、同じ秘密鍵を複数回使用する必要がない方法で署名を生成することができることが望ましい。

【０００７】
本明細書に開示される一態様によれば、デジタル署名を生成するコンピュータ実装方法であって、方法は、署名当事者によって実行され、第１のメッセージを取得することと、少なくとも外部データ項目のハッシュに基づいてエフェメラル秘密鍵（ephemeral private key）を生成することと、第１の署名構成要素（signature component）および第２の署名構成要素を含む第１の署名を生成することとを含み、第１の署名構成要素は、エフェメラル秘密鍵に対応するエフェメラル公開鍵（ephemeral public key）に基づいて生成され、第２の署名構成要素は、第１のメッセージと、エフェメラル秘密鍵と、第１の署名構成要素と、第１の秘密鍵とに基づいて生成される、方法が提供される。

【０００８】
本明細書に開示される別の態様によれば、デジタル署名が署名当事者によって生成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、検証当事者によって実行され、第１の署名構成要素および第２の署名構成要素を含む第１の署名を取得することと、署名当事者から候補の外部データ項目を取得することと、候補の外部データ項目のハッシュに基づいて候補のエフェメラル秘密鍵を生成することと、少なくとも候補のエフェメラル秘密鍵に対応する公開鍵に基づいて候補の第１の署名構成要素を生成することと、候補の第１の署名構成要素が第１の署名構成要素に対応するかどうかに基づいて、第１の署名が署名当事者によって生成されたことを検証することとを含む方法が提供される。

【０００９】
署名当事者は、メッセージに署名するためのデジタル署名を生成する。ブロックチェーンの文脈では、メッセージはトランザクションであり得、例えば、署名当事者は、前のトランザクションの出力をロック解除するための署名を生成する。一般に、メッセージは、任意の形態のメッセージ、例えば文書であり得、必ずしもブロックチェーンに関連する必要はない。署名は、エフェメラル秘密鍵、例えば、１回限り使用の秘密鍵に少なくとも部分的に基づいて生成される。エフェメラル秘密鍵は、外部情報、すなわち「外部データ項目」に少なくとも部分的に基づいて生成される。外部データ項目は、署名当事者の識別子、例えば、名前、住所、電話番号、国民保険番号、パスポート番号、公開鍵などを含み得、および／またはそれに基づいて生成され得る。いくつかの例では、外部データ項目は、別のデジタル署名である。

【００１０】
生成されると、署名は、公開鍵を使用して検証することができる。しかしながら、それは、署名当事者が署名を生成したことを証明するのに十分ではない。むしろ、署名当事者のみが外部データ項目を知っているので、署名当事者は、外部データ項目を明らかにし、検証当事者が署名の少なくとも一部（すなわち、「第１の署名構成要素」）を再構築することを可能にすることができる。すなわち、検証当事者は、外部データ項目に基づいて候補の第１の署名構成要素を生成する。再構築された第１の署名構成要素（すなわち、候補の第１の署名構成要素）が第１の署名構成要素と一致する場合、検証当事者は、署名当事者が実際に署名を生成したことを確信することができる。

【００１１】

本発明は、外部情報が署名に組み込まれること、すなわち、署名内に埋め込まれることを可能にする。外部情報は、署名当事者によって提供されない限り、検証当事者には知られない。具体的には、外部情報は、署名を導出するために使用される。例えば、署名当事者は、署名者のアイデンティティにリンクされた公開鍵を、異なる秘密鍵（すなわち、埋め込まれる公開鍵に対応しない秘密鍵）を使用して作成された署名に埋め込み得る。これにより、署名当事者は、アイデンティティにリンクされた公開鍵（identity-linked public key）に対応する秘密鍵を再使用する必要なしに、署名を生成したことを証明することができる。

【図面の簡単な説明】

【００１２】

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３Ａ】クライアントアプリケーションの概略ブロック図である。

【図３Ｂ】図３Ａのクライアントアプリケーションによって提示され得る例示的なユーザインターフェースの概略モックアップである。

【図４】本発明の実施形態を実装するための例示的なシステムの概略ブロック図である。

【図５】本発明のいくつかの実施形態による、デジタル署名を生成するための例示的な方法を示すフローチャートである。

【図６】本発明のいくつかの実施形態による、当事者がデジタル署名を生成したことを検証するための例示的な方法を示すフローチャートである。

【発明を実施するための形態】

【００１３】

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。

【００１４】

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。

【００１５】

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、特性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ソリューションを必要とする）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。

【００１６】

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発生ブロック（Ｇｂ：genesis block）１５３までずっと戻る。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発生ブロック１５３を指し示していた。

【００１７】

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していない。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。

【００１８】

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。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 transacti

on）または先行したトランザクション（predecessor transaction）とも呼ばれる。

【００１９】

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先のユーザ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。場合によっては、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。場合によっては、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。

【００２０】

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１０４に求める。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、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く、または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。

【００２１】

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（例えば、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。

【００２２】

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。ハッシュ関数の特性は、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。

【００２３】

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実施する。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、別名二重支出としても知られている、前に妥当性確認されたトランザクションと同じ出力の割当てを行う場合、トランザクションを有効として受け入れないことを含む。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。

【００２４】

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が最も長く成長しても、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。

【００２５】

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）」と呼ばれることもある。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、以下に説明する。

【００２６】

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を含む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。

【００２７】

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。

【００２８】

消費ユーザの役割を果たす複数の当事者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、（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）ブロックチェーン１５０のコピーを記憶するストレージエンティティとして動作し得る。

【００２９】

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求めら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、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。

【００３０】

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。

【００３１】

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。

【００３２】

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。

【００３３】

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ず、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。

【００３４】

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。

【００３５】

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を含み、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。

【００３６】

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へと前方に伝搬する。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。

【００３７】

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。

【００３８】

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）。

【００３９】

いくつかのブロックチェーンネットワークによって動作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場合、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することができる。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。

【００４０】

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。

【００４１】

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。

【００４２】

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取り、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。

【００４３】

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、この場合、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「後続の」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」および「後の」、「親」および「子」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。

【００４４】

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、後続の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、後続の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、後続の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。

【００４５】

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主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主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。

【００４６】

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の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。

【００４７】

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　P_A]

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き、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したアリスの署名を含むことを認証する。データの予想される部分自体（「メッセージ」）はまた、この認証を実行するために含まれる必要がある。実施形態では、署名されたデータは、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、含まれる必要はない）。

【００４８】

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。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。

【００４９】

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１つの理由である。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。

【００５０】

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。したがって、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であろう。

【００５１】

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ても、「後に残す」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。

【００５２】

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、$UTXO_1$を含むブロックを生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。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。

【００５３】

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。

【００５４】

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。

【００５５】

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）。

【００５６】

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。

【００５７】

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。そこのような通信は、「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。

【００５８】

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、またはさらに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と呼ばれ得る。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。

【００５９】

クライアントソフトウェア

図３Ａは、本開示の方式の実施形態を実装するためのクライアントアプリケーション１０５の例示的な実装を示す。クライアントアプリケーション１０５は、トランザクションエンジン４０１およびユーザインターフェース（ＵＩ）層４０２を含む。トランザクションエンジン４０１は、上記で説明した方式にしたがって、および、さらに詳細に簡潔に説明されるように、例えば、トランザクション１５２を定式化し、サイドチャネル３０１上でトランザクションおよび／または他のデータを受信および／または送信し、および／ま

たはトランザクションを１つまたは複数のノード１０４に送信してブロックチェーンネットワーク１０６全般に伝搬させるために、クライアント１０５の基本的なトランザクション関連機能を実装するように構成される。

【００６０】

ＵＩ層４０２は、機器１０２のユーザ出力手段を介してそれぞれのユーザ１０３に情報を出力すること、および機器１０２のユーザ入力手段を介してそれぞれのユーザ１０３から入力を受信することを含む、それぞれのユーザのコンピュータ機器１０２のユーザ入力／出力（Ｉ／Ｏ）手段を介してユーザインターフェースをレンダリングするように構成される。例えば、ユーザ出力手段は、視覚出力を提供するための１つまたは複数のディスプレイスクリーン（タッチスクリーンまたは非タッチスクリーン）、オーディオ出力を提供するための１つまたは複数のスピーカ、および／または触覚出力を提供するための１つまたは複数の触覚出力デバイスなどを含み得る。ユーザ入力手段は、例えば、１つまたは複数のタッチスクリーン（出力手段のために使用されるものと同一または異なる）の入力アレイ、マウス、トラックパッド、またはトラックボールなどの１つまたは複数のカーソルベースのデバイス、発話または音声入力を受信するための１つまたは複数のマイクロホンおよび発話または音声認識アルゴリズム、手動または身体ジェスチャの形態で入力を受信するための１つまたは複数のジェスチャベースの入力デバイス、あるいは１つまたは複数の機械式ボタン、スイッチ、またはジョイスティックなどを含み得る。

【００６１】

本明細書の様々な機能は、同じ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、これは、必ずしも限定的ではなく、代わりに、それらは、２つ以上の別個のアプリケーション一式で実装され得、例えば、一方が他方へのプラグインであるか、またはＡＰＩ（アプリケーションプログラミングインターフェース）を介してインターフェースしていることに留意されたい。例えば、トランザクションエンジン４０１の機能が、ＵＩ層４０２とは別個のアプリケーションにおいて実装されてもよいし、トランザクションエンジン４０１などの所与のモジュールの機能が、２つ以上のアプリケーション間で分割されてもよい。説明された機能のうちのいくつかまたはすべてが、例えばオペレーティングシステム層で実装され得ることも除外されない。本明細書のどこでも単一のまたは所与のアプリケーション１０５などを参照する場合、これは単なる例であり、より一般的には、説明される機能は任意の形態のソフトウェアで実装され得ることが理解されよう。

【００６２】

図３Ｂは、アリスの機器１０２ａ上のクライアントアプリケーション１０５ａのＵＩ層４０２によってレンダリングされ得るユーザインターフェース（ＵＩ）５００の例のモックアップを与える。同様のＵＩが、クライアント１０５ｂによってボブの機器１０２ｂ上に、または任意の他の当事者の機器上にレンダリングされ得ることが理解されよう。

【００６３】

例示として、図３Ｂは、アリスの視点からのＵＩ５００を示す。ＵＩ５００は、ユーザ出力手段を介して別個のＵＩ要素としてレンダリングされる１つまたは複数のＵＩ要素５０１、５０２、５０２を含み得る。

【００６４】

例えば、ＵＩ要素は、異なるオンスクリーンボタン、またはメニュー内の異なるオプションなどであり得る１つまたは複数のユーザ選択可能要素５０１を含み得る。ユーザ入力手段は、ユーザ１０３（この場合、アリス１０３ａ）が、スクリーン上のＵＩ要素をクリックもしくはタッチすることまたは所望のオプションの名前を言うことなどによって、オプションのうちの１つを選択または他の方法で動作させることを可能にするように構成される（注意：本明細書で使用される「手動」という用語は、自動との対比を意味し、必ずしも１つまたは複数の手の使用に限定されない）。オプションにより、ユーザ（アリス）は、トランザクション内に埋め込まれる署名を生成することができる。

【００６５】

代替的にまたは追加的に、ＵＩ要素は、１つまたは複数のデータ入力フィールド５０２を含み得、それを通して、ユーザは、署名内に埋め込まれるデータを入力することができる。これらのデータ入力フィールドは、ユーザ出力手段、例えばオンスクリーンを介してレンダリングされ、データは、ユーザ入力手段、例えば、キーボードまたはタッチスクリーンを通してフィールドに入力され得る。代替的に、データは、例えば、音声認識に基づいて、口頭で受信され得る。

【００６６】

代替的にまたは追加的に、ＵＩ要素は、ユーザに情報を出力するために出力される１つまたは複数の情報要素５０３を含むことができる。例えば、これ／これらは、スクリーン上にまたは音声でレンダリングされ得る。

【００６７】

様々なＵＩ要素をレンダリングし、オプションを選択し、データを入力する特定の手段は重要ではないことが理解されよう。これらのＵＩ要素の機能については、以下でより詳細に説明する。図３に示されるＵＩ５００は、単に図式化されたモックアップであり、実際には、簡潔にするために図示されていない１つまたは複数のさらなるＵＩ要素を含み得ることも理解されよう。

【００６８】

暗号プレリミナリ（ＣＲＹＰＴＯＧＲＡＰＨＹ　ＲＥＬＩＭＩＮＡＲＩＥＳ）

【００６９】

ＥＣＤＳＡ－楕円曲線群

【数１】

$$(E,\oplus)$$

は、有限体

【数２】

$$\mathbb{F}_p$$

上の巡回楕円曲線群であり、ｐは素数である。Ｅの要素の数はｎであり、ｎは素数である。Ｇ∈Ｅは、楕円曲線群の生成点であり、以下を意味する：

【数３】

$$\forall Y \in E\ \exists i \in \{1,\dots,n\} : Y = i \cdot G.$$

【００７０】

群演算

【数４】

「$\oplus$」

は、標準楕円曲線点加算であり、ｉ・Ｇは、Ｇに対する群演算のｉ回の反復を表す。

【数５】

$$i \cdot G = \underbrace{G \oplus G \oplus \dots \oplus G}_{i\text{回}}$$

【００７１】

以下において、文脈上他の意味に解すべき場合を除き、整数に対するすべての演算はモジュロｎである。

【００７２】

楕円曲線デジタル署名アルゴリズム
鍵生成は以下のように行われる：
１）署名用の秘密鍵ｊ∈｛１，…，ｎ－１｝を選択する
２）公開鍵は、Ｙ＝ｊ・Ｇであり、Ｇは、生成点である。
【００７３】
署名アルゴリズムは、秘密鍵ｊメッセージおよびエフェメラル鍵ｋを取り、署名を生成する：
３）ランダムなｋ∈｛１，…，ｎ－１｝（エフェメラル鍵）を選択する
４）Ｒ＝（r_x，r_y）＝ｋ・Ｇ－ＥＣ点を計算する
５）ｒ＝r_x　ｍｏｄ　ｎを計算する
　ａ．ｒ＝０である場合、ステップ３に進む
６）署名ｓ＝k^{-1}（ｅ＋ｊｒ）を生成する。ここで、ｅ＝ｈａｓｈ（ｍ）である
７）ｓ＝０である場合、ステップ３に進む
８）［ｒ，ｓ］をメッセージｍの署名として出力する
【００７４】
次いで、検証アルゴリズムは、署名およびメッセージを取り、署名者の公開鍵を使用してｒを再構築し、署名で与えられるｒ値を検証する。
１）ｅ＝ｈａｓｈ（ｍ）を計算する
２）k_1＝es^{-1}　ｍｏｄ　ｎおよびk_2＝rs^{-1}　ｍｏｄ　ｎを計算する
３）Ｑ＝（q_x，q_y）＝k_1・Ｇ＋k_2・Ｙを計算する
４）
【数６】

$$q_x \equiv r \bmod n$$

である場合、署名は有効である
そうでなければ無効である。
【００７５】
署名には以下の表記が使用される：
【数７】

$$Sig_Y = [r_Y, s_Y],$$

ここで、［r_Y，s_Y］は、公開鍵Ｙを使用して検証されたときに、有効な署名である。
【００７６】
ディフィーヘルマン（ＤＨ）鍵交換
２つの当事者は、以下の方法で対称シークレット鍵（secret　key）を作成することによって、セキュアな通信チャネルを確立し得る。アリスとボブが共有シークレット鍵（shared　secret　key）を作成したいと望み、アリスが、公開鍵PK_A＝sk_A・Ｇに対応する秘密鍵sk_Aを知っており、ボブが、ボブの公開鍵PK_B＝sk_B・Ｇに対応する秘密鍵sk_Bを知っていると仮定する。
【００７７】
共有シークレット鍵を見つけるために、以下のステップを行う。
１．アリスは、ディフィーヘルマン鍵sk_{AB}＝sk_A・PK_Bを計算する。
２．ボブは、ディフィーヘルマン鍵sk_{AB}＝sk_B・PK_Aを計算する。
【００７８】
共有シークレット鍵を確立するための別の方法は、国際公開第２０１７／１４５０１６号に記載されており、この方法では、事前合意されたメッセージがＤＨ鍵に追加され、新しい鍵が作成される。このメッセージは、送信される新しい通信ごとに変更することができ、決定論的鍵のセットを作成する。例えば、メッセージは、ｍ＝ｈａｓｈ（ｄａｔｅ｜｜ｔｉｍｅ）であり得る。次いで、アリスは、メッセージを使用して秘密鍵sk_{A1}＝　ｓ

k_A＋ｈａｓｈ（ｄａｔｅ｜｜ｔｉｍｅ）を生成し、同様に、ボブは、秘密鍵sk_{B1}＝sk_B＋ｈａｓｈ（ｄａｔｅ｜｜ｔｉｍｅ）を生成することができる。そのため、アリスとボブの両方が、共有秘密鍵sk_{AB1}＝sk_{A1}・PK_{B1}＝sk_{B1}・PK_{A1}を生成することができる。

【００７９】

ＨＤウォレット

階層的決定論的ウォレットは、ビットコイン改善提案３２（ＢＩＰ３２：Bitcoin Improvement Proposal 32）ウォレットがその特定のタイプであり、多くの鍵が単一の入力から導出され得る決定論的ウォレットである。入力は、シードと呼ばれる何らかのランダムエントロピーであり、そこからマスタ鍵が導出される。次いで、マスタ鍵は、図２に示されるように、複数の子鍵を導出するために使用される。

【００８０】

ＢＩＰ３２では、マスタ秘密鍵は、シードのＨＭＡＣ－ＳＨＡ５１２の結果の左の３２バイトであり、または明示的に、それは、

【数８】

$$sk_{master} = HMAC\text{-}SHA512_L('Bitcoin\ Seed', seed)$$

であり、チェーンコードは、

【数９】

$$c_{master} = HMAC\text{-}SHA512_R('Bitcoin\ Seed', seed)$$

であり、すべての子鍵はこれらから導出することができ、ここで、

【数１０】

$$HMAC\text{-}SHA512(c, K) = SHA512(c \oplus opad || SHA512((c \oplus ipad) || K))$$

は、ＳＨＡ５１２ハッシュ関数を使用するＨＭＡＣである。上記の式において、ｏｐａｄは、ブロックサイズの外側パディングであり、ｉｐａｄは、ブロックサイズの内部パディングである。

【００８１】

ＨＭＡＣは、２つの入力、すなわち、ｃおよびＫを必要とする。簡潔にするために、かつユーザが単一のシードを覚えるまたは記憶することのみを要求されるように、ＢＩＰ３２プロトコルは、第１の入力を文字列「Bitcoin Seed」として設定、すなわち、ｃ＝‘Bitcoin Seed’とする。これは、ＨＤウォレットを生成するための１つの例示的プロトコルであり、異なるプロトコルは、異なる入力、例えば、２つのランダムに生成されたシードを必要とし得ることが理解されよう。言い換えると、文字列「Bitcoin Seed」の使用は、ＨＤウォレットを生成するための必要要件ではない。

【００８２】

親秘密鍵sk_{parent}から、強化された子秘密鍵sk_{child}を計算するための式は、以下の通りである：

【数１１】

$$sk_{child} = sk_{parent} + HMAC\text{-}SHA512_L(c_{parent}, sk_{parent} || index)$$

ここで、c_{parent}は、親チェーンコードであり、０≦ｉｎｄｅｘ＜2^{31}は子インデックスであり、$HMAC\text{-}SHA512_L$は、ＳＨＡ－５１２ハッシュ関数を用いて計算されたＨＭＡＣ関数の結果の左の３２バイトである。子公開鍵のための対応する式は、この式にベースポイントＧを単純にポイント乗算（point multiply）することによって導出される。子チェーンコードc_{child}は、ＨＭＡＣ関数の結果の右の３２バイトであると定義される：

【数１２】

$$c_{child} = HMAC\text{-}SHA512_R(c_{parent}, sk_{parent}||index)$$

【００８３】

親公開鍵ｐｋ$_{parent}$および親秘密鍵ｓｋ$_{parent}$から、強化されていない子秘密鍵ｓｋ$_{child}$を計算するための式は以下である：

【数１３】

$$sk_{child} = sk_{parent} + HMAC\text{-}SHA512_L(c_{parent}, pk_{parent}||index)$$

ここで、ｃ$_{parent}$は、親チェーンコードであり、$2^{31} \leqq index < 2^{32}$は子インデックスであり、ＨＭＡＣ－ＳＨＡ５１２は、ＳＨＡ－５１２ハッシュ関数を用いて計算されたＨＭＡＣ関数である。強化された鍵と同様に、強化されていない鍵の子チェーンコードｃ$_{child}$は、ＨＭＡＣ関数の結果の右の３２バイトであると定義される：

【数１４】

$$c_{child} = HMAC\text{-}SHA512_R(c_{parent}, pk_{parent}||index)$$

【００８４】

この第２のタイプの子鍵により、親公開鍵およびチェーンコードの知識を有する者であれば、以下の式を使用して、子公開鍵を導出することができる：

【数１５】

$$pk_{child} = pk_{parent} + HMAC\text{-}SHA512_L(c_{parent}, pk_{parent}||index) \cdot G$$

【００８５】

これを使用して、外部当事者は、必要に応じて様々な支払いアドレスを導出し、通信および記憶の回数を減らすと同時に、鍵の再使用を回避することができる。

【００８６】

一般に、ＨＤウォレットは、秘密鍵－公開鍵ペアの何らかの階層ツリー状構造を生成する必要がある。これにより、１つのシードからすべて再生成することができる多数の鍵ペアが提供される。

【００８７】

デジタル署名

図４は、本発明のいくつかの実施形態による、デジタル署名を生成するための例示的なシステム４００を示す。一般に、システムは、少なくとも署名当事者４０１（すなわち、署名生成当事者）および検証当事者４０２（すなわち、署名検証当事者）を含む。いくつかの例では、システムは、１つまたは複数のブロックチェーンノード１０４および／または第２の当事者４０３を含む。

【００８８】

各当事者４０１、４０２、４０３は、それぞれのコンピューティング機器（図示せず）を操作する。それぞれの当事者４０１、４０２、４０３のそれぞれのコンピューティング機器の各々は、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（ＧＰＵ）、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）を含むそれぞれの処理装置を備える。それぞれのコンピューティング機器はまた、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得る。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。それぞれのコンピューティング機器は、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み得る。代替的にまたは追加的に、それぞれのコンピューティング機器は、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（クラウドコンピューティングリソースは、１つまたは複数のサイトで実装される１つまたは複数の物理サーバデバイスのリソースを含む）。システム４００の当事者によって実行されるものとして説明される任意の行為は、その当事者によって操作されるそれぞれのコンピューティング装置によって実行され得ることが理解されよう。

【００８９】

本発明は、ブロックチェーンの文脈での使用のみに限定されないが、以下では、署名当事者４０１を、図１～図３を参照して説明されたアリス１０３ａと同等として説明する。すなわち、いくつかの例では、アリス１０３ａは署名当事者４０１である。これらの例では、ボブ１０３ｂは第２の当事者４０３であり得る。検証当事者４０２は、以下ではキャロルと呼ばれる。

【００９０】

これらの実施形態では、アリス４０１は、署名を生成し、同じ秘密鍵を再使用することなく、例えば、同じ秘密鍵を使用して２つの署名を生成することなく、アリスがその署名を生成したことをキャロル４０２に証明したいと思う。

【００９１】

アリス４０１は、署名されるべきメッセージ、例えば、ブロックチェーントランザクション、文書、コントラクトなどのうちのいくつかまたは一部を取得する。アリス４０１は、自らメッセージを生成してもよいし、アリス４０１は、例えば、ボブ４０３からメッセージを受信してもよい。アリス４０１は、外部データ項目も取得する。アリス４０１は、例えば、名前、パスポート番号、公開鍵などの外部データ項目をすでに有していてもよいし、アリス４０１は、外部データ項目を生成してもよい。例えば、以下でより詳細に説明するように、外部データ項目は、アリス４０１によって生成された署名（「第２の署名」）であり得る。

【００９２】

アリス４０１は、外部データ項目のハッシュに基づいてエフェメラル秘密鍵を生成する。例えば、エフェメラル秘密鍵は、外部データ項目のハッシュから構成され得る（それを含み得る、またはそれから成り得る）。言い換えると、エフェメラル秘密鍵としてランダム値を割り当てる（上記のプレリミナリセクションを参照）のではなく、エフェメラル秘密鍵は、ここでは、外部データ項目をハッシュした結果の関数である。外部データ項目のハッシュを生成するために使用されるハッシュ関数は、任意の適切なハッシュ関数、例えば、ＳＨＡ２５６、ＳＨＡ５１２であり得、１つまたは複数のハッシュ関数を複数回適用することを含み得る。例えば、ハッシュ関数は、ダブルハッシュ関数、例えば、ＳＨＡ２５６ｄ（ｘ）＝ＳＨＡ２５６（ＳＨＡ２５６（ｘ））であり得る。

【００９３】

次いで、アリス４０１は、署名の第１の構成要素および第２の構成要素を生成する。第１の署名構成要素は、エフェメラル秘密鍵に対応する公開鍵に基づいて生成される。すなわち、アリス４０１は、エフェメラル秘密鍵に対応するエフェメラル公開鍵を生成する。公開鍵は、２つの構成要素（例えば、ｘ値およびｙ値）を有する。例において、第１の署名構成要素は、エフェメラル公開鍵の第１の構成要素、例えば、ｘ値に基づく。第２の署名構成要素は、署名されるべきメッセージと、エフェメラル秘密鍵と、第１の署名構成要素と、秘密鍵（「第１の秘密鍵」）とに基づく。第１の秘密鍵は、任意の秘密鍵、例えば、ランダム秘密鍵であってもよいし、以下で説明するように、アイデンティティにリンクされた公開鍵にリンクすることができる秘密鍵であってもよい。

【００９４】

したがって、アリス４０１は、この時点でアリス４０１のみに知られている外部データ

項目、例えば個人識別子を組み込んだ署名を生成している。外部データ項目自体が秘密（secret）である必要はないことに留意されたい。外部データ項目が署名内に埋め込まれていることは、最初は秘密に保たれることが好ましいが、これは必須ではない。例えば、外部データ項目は、それ自体が１つまたは複数の当事者に知られている証明された公開鍵であり得る。

【００９５】

アリス４０１は、署名をキャロル４０２が利用できるようにし得る。例えば、アリス４０１は、署名をキャロル４０２に送信してもよいし、アリスは、署名を公開するか、または他の方法でブロードキャストしてもよい。署名およびメッセージは、一緒に送信または公開されることが好ましい。アリス４０１はまた、署名と同時に、または異なる時間に、例えば後の時間に、外部データ項目をキャロル４０２が利用できるようにし得る。署名と同じ方法で、または異なる方法で外部データ項目を利用できるようにし得る。例えば、アリス４０１は、セキュアな通信チャネルを介して外部データ項目をキャロル４０２に送信し得る。

【００９６】

上述したように、外部データ項目は、別の署名であり得るか、または少なくとも別の署名を含み得る。その場合、アリス４０１は、第２のメッセージを取得し、少なくとも第２のメッセージと「主秘密鍵」とに基づいて第２の署名を生成する。最も広い例では、「主（main）」は、単に、第１の秘密鍵と区別するためのラベルとして使用される。すなわち、主秘密鍵は、第１の署名の第２の署名構成要素を生成するために使用される第１の秘密鍵以外の、アリス４０１によって所有される任意の秘密鍵であり得る。この例では、第２の署名を生成するために使用されるエフェメラル秘密鍵は、外部データ項目に基づく第１の署名を生成するために使用されるエフェメラル秘密鍵とは対照的に、ランダムに生成され得る。

【００９７】

アリス４０１は、第２のメッセージの少なくとも一部を自ら生成し得る。追加的にまたは代替的に、アリス４０１は、別の当事者、例えばキャロル４０２から、第２のメッセージの少なくとも一部を受信するか、または他の方法で取得し得る。すなわち、キャロル４０２が、第２のメッセージの一部または全部をアリス４０１に送信してもよいし、アリス４０１およびキャロル４０２が、メッセージの少なくとも一部に関して事前に合意していてもよい。例えば、アリス４０１およびキャロル４０２は、第２の署名が生成された時間および／またはデータの指示を含めることに合意していてもよい。いくつかの例では、第２のメッセージは、第１のメッセージを含むか、またはそれに基づいて生成され得る。例えば、第２のメッセージは、第１のメッセージの始まりまたは終わりに連結された追加データを有する第１のメッセージを含み得る。

【００９８】

アリス４０１は、少なくとも第２の署名をキャロル４０２に送信するか、またはアリス４０１は、第２の署名を公開し得る。キャロル４０２が第２のメッセージをまだ入手していない場合、アリス４０１は、第２のメッセージをキャロル４０２に送信するか、または第２のメッセージを公開し得る。アリス４０１はまた、主秘密鍵に対応する主公開鍵をキャロル４０２に送信し得るか、または少なくとも、キャロル４０２が主公開鍵を取得し得る場所、例えば、証明機関によって公開され、主公開鍵がアリス４０１にリンクされていることを証明する証明書を記憶している位置を示し得る。

【００９９】

いくつかの例では、主秘密鍵と第１の秘密鍵との間にリンクが存在せず、他の例では、第１の秘密鍵は主秘密鍵にリンクされ、すなわち主秘密鍵から導出される。例えば、主秘密鍵は、アリス４０１によって所有されるＨＤウォレットのマスタ秘密鍵であり得、第１の秘密鍵は、マスタ秘密鍵から導出された子鍵であり得、すなわち、第１の秘密鍵は、主秘密鍵に確定的にリンクされる。他の例では、第１の秘密鍵は、アリスの主秘密鍵およびボブの主秘密鍵に基づいて（または同等に、ボブの主秘密鍵に対応するボブの主公開鍵に

基づいて）生成され得る。例えば、第１の秘密鍵は、ＤＨ秘密鍵、または国際公開第２０１７／１４５０１６号に記載の技法を使用して導出された秘密鍵であり得、その例を以下に提供する。要約すると、国際公開第２０１７／１４５０１６号では、アリスの主秘密鍵とボブの主公開鍵とに基づいて共通シークレット（common secret）が生成される。次いで、アリス４０１は、アリスの主秘密鍵と共通シークレットとに基づいて第１の秘密鍵を生成し得る。ボブの主公開鍵は、ボブのアイデンティティにリンクされ得、例えば、それは、証明された公開鍵であり得る。

【０１００】

第１の署名を生成するために使用されるエフェメラル秘密鍵は、少なくとも外部データ項目のハッシュ、例えば、第２の署名のダブルハッシュに基づく。エフェメラル秘密鍵はまた、ランダムに生成されたソルト値、すなわち、外部データ項目のハッシュに追加された値に基づき得る。好ましくは、ソルト値は一度だけ使用され、すなわち、第１の署名の異なるインスタンスを生成するためには異なるソルト値が使用される。これらの例では、アリス４０１は、第３のメッセージとソルト値とに基づいて第３の署名を生成し得る。すなわち、ソルト値は、第３の署名を生成するための秘密鍵として使用される。第３のメッセージは、第１のメッセージおよび／または第２のメッセージに基づいて生成され得る。第３のメッセージは、第２のメッセージと同じであってもよい。アリス４０１は、第３の署名をキャロル４０２に送信し得る。アリスが第３のメッセージを生成した場合、アリスはそれもキャロル４０２に送信し得る。あるいは、キャロル４０２は、第３のメッセージをアリス４０１に送信している場合があり、その場合、アリス４０１は、それをキャロル４０２に送り返す必要はないが、送り返すことを選択してもよい。

【０１０１】

上述したように、本発明の実施形態は、ブロックチェーン１５０との併用に限定されない。しかしながら、そのようなものでは、第１のメッセージは、ブロックチェーントランザクションであり得る。例えば、アリス４０１は、ブロックチェーントランザクションの一部または全部、例えば、トランザクションの１つまたは複数の入力および／または１つまたは複数の出力に署名し得る。次いで、アリス４０１は、アリスが署名していないトランザクションの入力に第１の署名を含め得る。トランザクションは、ボブ４０３および／またはキャロル４０２にロックされた出力を含み得、例えば、出力は、ボブ４０３によって所有される公開鍵にロックされたpay-to-public-key（Ｐ２ＰＫ）またはpay-to-public-key-hash（Ｐ２ＰＫＨ）出力であり得る。第２のメッセージは、トランザクションを含み得る。第２のメッセージは、ブロックチェーン１５０に関するデータ、例えば、トランザクションが生成されたときのブロックチェーンの現在のブロック高さを含み得る。これらの例では、アリス４０１は、ブロックチェーン１５０へのトランザクションを送信することによって、第１のメッセージをキャロル４０２が利用できるようにし得、そこから、キャロル４０２がアクセスし得る。これは図４に示されている。

【０１０２】

図５は、本発明のいくつかの実施形態による、署名を生成するためにアリス４０１によって行われ得るステップの例示的なシーケンスを示す。ステップのいくつかが異なる順序で実行され得ることは理解されよう。ステップＳ５０１において、アリス４０１は、第１のメッセージ、例えば、ブロックチェーントランザクションを取得する。ステップＳ５０２において、アリス４０１は、外部データ項目、例えば、第２の署名を取得する。ステップＳ５０３において、アリス４０１は、外部データ項目に基づいて、例えば、第２の署名のハッシュに基づいて、エフェメラル秘密鍵を生成する。ステップＳ５０４において、アリス４０１は、エフェメラル秘密鍵に基づいて署名を生成し、ステップＳ５０５において、アリスは、少なくとも外部データ項目を検証当事者であるキャロル４０２に送信する。

【０１０３】

次に、検証当事者であるキャロル４０２が行うアクションについて説明する。キャロル４０２は、アリス４０１に、アリスが署名を生成したことを証明してもらいたい。キャロル４０２は、第１の署名を取得する。アリス４０２が第１の署名をキャロル４０２に送信

し得るか、または第１の署名は、公的に入手可能であり、例えば、ブロックチェーン１５０上に記録され得る。第１の署名がブロックチェーントランザクションの入力に含まれる場合、キャロル４０２は、トランザクションから第１の署名を抽出することによって第１の署名を取得する。キャロル４０２はまた、アリス４０１から候補の外部データ項目を取得する。ここで、「候補（の）（candidate）」は、アリス４０１が、第１の署名内に埋め込まれていると主張する外部データ項目を指すために使用される。実際にそうである場合、候補の外部データ項目は、上述した外部データ項目と同じである。しかしながら、この時点で、キャロル４０２は、そうであることを確認することができないので、「候補」としている。

【０１０４】

キャロル４０２は、アリス４０１がエフェメラル秘密鍵を生成した方法と同様に、候補の外部データ項目を使用して候補のエフェメラル秘密鍵を生成する。キャロル４０２は、アリス４０１と全く同じ方法を使用する必要はないことに留意されたい。例えば、アリス４０１は、アリスのエフェメラル秘密鍵を生成するために、キャロル４０２が入手することができないソルト値を使用してもよい。キャロル４０２は、候補のエフェメラル秘密鍵に対応する候補のエフェメラル公開鍵を生成し、それから、候補の第１の署名構成要素を生成する。例えば、候補の第１の署名構成要素は、候補のエフェメラル公開鍵の第１の構成要素（例えば、ｘ値）であり得る。

【０１０５】

キャロル４０１によって取得された第１の署名は、第１の署名構成要素および第２の署名構成要素を含む。アリス４０１が第１の署名を生成したことを検証するために、キャロル４０２は、候補の第１の署名構成要素を第１の署名構成要素と比較する。それらが一致する場合、キャロル４０２は、アリス４０１が実際に第１の署名を生成したことを確信することができる。すなわち、候補の第１の署名構成要素と第１の署名構成要素とが一致するためには、候補の外部データ項目は、第１の署名を生成するために使用される外部データ項目でなければならない。アリス４０１がキャロル４０２に候補の外部データ項目を提供したので、これは、アリス４０１が第１の署名を生成したことを証明する。このプロセスは、図６のステップＳ６０１～Ｓ６０５に示されている。

【０１０６】

キャロル４０２はまた、対応する公開鍵に対して妥当性確認されたとき、第１の署名が有効な署名であることを検証し得る。第１の署名がブロックチェーントランザクションに署名するために使用され、そのトランザクションがブロックチェーンに記録されている場合、キャロル４０１は、第１の署名が有効な署名であると仮定し得る（すなわち、署名が有効でなかった場合、トランザクションはブロックチェーンノードによって受け入れられていなかったであろう）。しかしながら、キャロル４０１は、使用されているロック解除スクリプトが署名チェックを含むことを依然として検証し得る（すなわち、ブロックチェーンノードがトランザクション妥当性確認中に署名に対して署名チェックを実行したことを確認するために）。これを行うために、キャロル４０１は、使用されたトランザクションのロック解除スクリプトがＯＰ＿ＣＨＥＣＫＳＩＧスクリプトを含むことをチェックし得る。

【０１０７】

アリスの観点から本発明の実施形態を説明する際に上述したように、外部データ項目は、それ自体が、署名、すなわち第２の署名であってもよい。この場合、キャロル４０２は、例えばアリス４０１から、第２のメッセージを取得し、例えば、証明された公開鍵などの、アリス４０１によって提供されるかまたは他の方法でアリス４０１にリンクされた公開鍵を使用して検証されたときに、第２の署名が有効な署名であることを検証し得る。

【０１０８】

アリス４０１が、第１の署名を生成するために使用されるエフェメラル秘密鍵を生成するためにソルト値を使用した場合、アリス４０１は、ソルト値に対応する公開鍵をキャロル４０２に提供し得る。次いで、キャロル４０２は、「ソルト公開鍵」に基づいて、例え

ば、候補のエフェメラル公開鍵とソルト公開鍵との組合せに基づいて、候補の第１の署名構成要素を生成し得る。上記組合せのｘ値は、候補の第１の署名構成要素を生成するために使用され得る。一例を以下にさらに提供する。これらの例では、アリス４０１はまた、キャロル４０２に第３の署名および第３のメッセージを提供し得る。キャロル４０２は、ソルト公開鍵を使用して検証されたときに、第３の署名が有効な署名であることを検証し得る。

【０１０９】

以下は、本発明の具体的な詳細な例を提供する。特に、以下では、秘密鍵の知識証明を可能にするために、ハッシュ関数を使用してトランザクション署名と秘密鍵とを暗号的にリンクするための方法を詳述する。この場合、誰かのアイデンティティに対応する公開鍵を所与の署名にリンクすることを伴うが、本方法は、任意の外部データを署名に含めるために使用されてもよい。これは１つの例示的な実装形態にすぎず、以下のうちの少なくともいくつかは、この特定の実装形態に固有のオプションの特徴であることが理解されよう。

【０１１０】

「安全な鍵交換証明方法」（ＳＫＥＡＭ：secure key exchange attest method）と呼ばれる方法は、証明者であるアリス４０１と検証者であるキャロル４０２との間の連続的な対話である。アリス４０１は、署名のペアを使用して、アリスのアイデンティティ鍵がブロックチェーン１５０自体に現れないことを保証しながら、支払トランザクションをアリスのアイデンティティ公開鍵に明確にリンクする。アルゴリズムの暗号セキュリティは、暗号ハッシュ関数を使用してエフェメラル鍵と署名とをリンクすることに依存する。

【０１１１】

セットアップ：アリス４０１およびボブ４０３は、公開鍵PK_AおよびPK_Bを有する。これらは、それらのアイデンティティにリンクされた固定の公開鍵である。アリス４０１およびボブ４０３は、共有鍵ペアPK_{AB1}，…，PK_{ABN}のセットを導出するために、プロトコルを使用する。代替的に、アリスは、ＢＩＰ３２プロトコルを使用することができ、マスタ鍵をアリスの「アイデンティティ鍵」として設定し、（強化されていない）子鍵導出方法を使用して、署名に使用される鍵を導出する。両方のプロトコルの主要な特徴は、固定のアイデンティティ鍵をトランザクション署名鍵のセットとリンクする鍵導出経路を利用することである。PK_{A1}，…，PK_{AN}は、トランザクション鍵（すなわち、トランザクションに署名するために使用される秘密鍵、および署名を妥当性確認するための公開鍵）であり、それぞれのアイデンティティ鍵に共通シークレットを追加することによって導出される（以下を参照）。

【０１１２】

ディフィーヘルマン基づく共有シークレット鍵導出の簡略化された例は、以下の計算を使用する：

【数１６】

$$A = sk_A \cdot PK_B = sk_B \cdot PK_A,$$
$$sk_{AB1} = H(A),$$
$$PK_{AB1} = sk_{AB1} \cdot G$$

ここで、Ｈ（・）は、ハッシュアルゴリズムであり、Ａは、アリス４０１およびボブ４０３のみが計算することができる共有ディフィーヘルマン鍵である。次いで、アリス４０１およびボブ４０３は、確立された共有シークレット整数sk_{AB1}を使用して、トランザクション署名鍵を計算することができる：

【数１７】

$$\text{アリス: } PK_{A1} = PK_A \oplus PK_{AB1}$$
$$sk_{A1} = sk_A + sk_{AB1}$$
$$\text{ボブ: } PK_{B1} = PK_B \oplus PK_{AB1}$$
$$sk_{B1} = sk_B + sk_{AB1}$$

【０１１３】

ここで、秘密鍵加算は、モジュロｎであり、公開鍵「加算」は、ｓｅｃｐ２５６ｋ１群演算

【数１８】

$$\oplus$$

を使用する。楕円曲線群の準同型特性により、秘密鍵と公開鍵との間の関係は加　算の下で成立することができる。

【０１１４】

証明（attestation）に使用することができる署名をどのようにセットアップするかについて以下で説明する。以下では、署名されるべきメッセージは、ビットコイントランザクションであるが、そうである必要はない。

【０１１５】

ステップ１：アリス４０１は、ボブのトランザクション公開鍵（または、ボブの公開鍵をハッシュすることによって生成されたアドレス）に支払う、署名されていないトランザクションメッセージＴｘ１’を作成する。例示的なトランザクションを以下に概略的に示す。

【表１】

$TxID1'$			
入力		出力	
値	ScriptSig	値	ScriptPubKey
1 BSV		0.99 BSV	OP_DUP OP_HASH160 < $H_{160}(PK_{B1})$> OP_EQUALVERIFY OP_CHECKSIG

【０１１６】

ステップ２：アリス４０１は、自身のアイデンティティ鍵およびトランザクションメッセージＭ（「第２のメッセージ」）を使用して署名（「第２の署名」）を作成する。この特定の例では、メッセージは、現在のチェーンチップブロック高さと連結されたシリアル化されたトランザクションバイトから構成される。ブロック高さは、メッセージのオプションの特徴として含まれる。

【数１９】

$$M = Tx1'||block\_hash,$$
$$Sig_{PK_A}(M) = [r_A, s_A]$$

【０１１７】

ｒ$_A$は、ランダムエフェメラル鍵を使用して導出された１回限りの値であることに留意されたい。メッセージＭおよび署名［ｒ$_A$，ｓ$_A$］は、この時点で、署名されていないトランザクションデータ、ブロックハッシュ（タイムスタンプ）、およびランダムエフェメラル鍵という３つのソースコンポーネントを有する。

【０１１８】

ステップ３ａ：アリス４０１は、r_A，s_Aおよび１回限りのランダムソルト値ｗ∈Ｚｎを使用して、エフェメラル鍵を作成し、

【数２０】

$$k_{A1} = SHA256d([r_A, s_A]) + w \bmod n,$$
$$r_{A1} = [k_{A1} \cdot G]_x \bmod n$$

ここで、ＳＨＡ２５６ｄ（ｘ）＝ＳＨＡ２５６（ＳＨＡ２５６（ｘ））である。

【０１１９】

ステップ３ｂ：アリス４０１は、ｗを使用して公開値Ｗおよび署名を計算する：

【数２１】

$$Sig_W(M), \qquad W = w \cdot G$$

【０１２０】

ステップ４：ステップ３ａで生成されたエフェメラル鍵を使用して、以下の署名を生成する：

【数２２】

$$s_{A1} = k_{A1}^{-1}(H(Tx1') + sk_{A1} \cdot r_{A1}),$$

ここで、PK_{A1}＝sk_{A1}・Ｇである。

【０１２１】

ステップ５：［r_{A1}，s_{A1}］は、PK_{A1}に対して検証されたときに、Ｔｘ１の有効な署名（「第１の署名」）を表す：

【数２３】

$$Sig_{PK_{A1}}(Tx1') = [r_{A1}, s_{A1}]$$

【０１２２】

以下の表は、トランザクションの最終状態を概略的に示す。r_{A1}は、アリスのアイデンティティ鍵によって生成された署名から導出されたので、アリスのアイデンティティ鍵は、Ｔｘ１のための署名に埋め込まれる。

【表２】

$TxID1$			
入力		出力	
値	ScriptSig	値	ScriptPubKey
1 BSV	$<[r_{A1}\ s_{A1}]>$ $<PK_{A1}>$	0.99 BSV	OP_DUP OP_HASH160 $<H_{160}(PK_{B1})>$ OP_EQUALVERIFY OP_CHECKSIG

【０１２３】

キャロル（第三者）４０２が、アリス４０１がＴｘ１の作成者であることを検証し、sk_{A1}およびsk_Aから署名を証明可能にリンクしたいと望むと仮定する。アリス４０１は、sk_{A1}を再使用する必要なく、自身がＴｘ１を生成して署名したという証拠を提供することができる。

【０１２４】

ステップ１：キャロル４０２は、Ｔｘ１を取得し、トランザクションが有効なビットコイントランザクションである（すなわち、ネットワークコンセンサスルールにしたがって

有効である）ことをチェックする。キャロルは、Ｔｘ１のScriptSigから［r_{A1}　s_{A1}］を抽出する。

【０１２５】

ステップ２ａ：アリス４０１は、上記ステップ２のＭおよび

【数２４】

$$Sig_{PK_A}(M)$$

を共有する。

【０１２６】

ステップ２ｂ：アリスは、ステップ３ｂのＷおよびＳｉｇｗ（Ｍ）を共有する。キャロル４０２は、アリス４０１がｗを使用して署名するための異なる（ランダム）メッセージを作成したいと望み得る。

【０１２７】

ステップ３：キャロル４０２は、以下の３つの条件を検証する。

１）

【数２５】

$$Sig_{PK_A}(M)$$

は、PK_Aを使用して検証されたときに、Ｍの有効なＥＣＤＳＡ署名である。

２）Ｍの最後の３２バイトは、以下の有効なブロックハッシュである：

【数２６】

$$M[len(M) - 32 : len(M)] = block\_hash$$

【０１２８】

キャロル４０２は、block_hashによって表されるブロックとＴｘ１が現れるブロックと間のブロック距離に追加の制約を課したいと望み得る。これは、アリス４０１が自身の署名のための正確なタイムスタンプを作成したことを保証する。

３）先行するバイトは、Ｔｘ１’に等しい（Ｔｘ１のための署名されていないトランザクションバイト）

【数２７】

$$M[0 : len(M) - 32] = Tx1'$$

【０１２９】

ステップ４ａ：キャロル４０２は以下を計算する：

【数２８】

$$k' = SHA256d\left(Sig_{PK_A}(M)\right),$$
$$r' = [k' \cdot G + W]_x \bmod n$$

【０１３０】

ステップ４ｂ：キャロル４０２は、公開値Ｗで検証されたときに、ＳｉｇＷ（Ｍ）がＭの有効なＥＣＤＳＡ署名であることをチェックする。

【０１３１】

ステップ５：キャロル４０２は、ｒ’＝r_{A1}であるかをチェックする。そうである場合、アリス４０１は、キャロル４０２に対して、自身のアイデンティティ鍵がＴｘ１のための署名生成で使用されたことを証明している。

【０１３２】

上記の方式では、署名鍵sk_{AB1}は、共通シークレットを使用してアイデンティティ鍵sk_Aから導出される。すなわち、以下である：

$$sk_{AB1} = sk_A + sk_{AB}$$

ここで、sk_{AB}は、アリス４０１とボブ４０３との間の共有シークレットである。アリス４０１は、２つの方法のうちの１つで、署名鍵がこのリンクを有することをキャロル４２に証明することができる。

【０１３３】

キャロル４０２は、知識のゼロ知識証明（zero-knowledge proof）を使用し得る。キャロル４０２は、上記の証明から、sk_Aおよびsk_{AB1}に対応する公開鍵を知っており、それぞれPK_AとPK_{AB1}と表記する。

１．アリス４０１は、共有シークレットsk_{AB}に対応する公開鍵と、ランダム値$y \in_R \{1, \cdots, k-1\}$とをキャロル４０２に送信する。すなわち、アリスは、$X = sk_{AB} \cdot G$および$Y = r \cdot G$を送信する。

２．キャロル４０２は、チャレンジ$c = hash(x)$をアリスに返す。

３．アリス４０１は、$u = c \cdot sk_{AB} + y$を計算し、これをキャロル４０２に送り返す。

４．キャロル４０２は、$u \cdot G = c \cdot X + Y$であることおよび$PK_{AB1} = P_A + X$であることをチェックする。

【０１３４】

これが成立する場合、キャロル４０２は、アリス４０１が実際にsk_{AB1}を知っていることを知る。このプロセスは、本質的に、離散対数に対するゼロ知識証明である。チャレンジは、何らかの事前合意されたチャレンジまたは標準チャレンジであり得るので、ステップ２はスキップすることができ、結果として、証明は非対話式になる。

【０１３５】

ソルトwの知識は、前のセクションで説明したような署名の代わりに、この方法を用いて証明され得ることに留意されたい。

【０１３６】

代替的に、アリス４０１は、$X = sk_{AB} \cdot G$で検証されるsk_{AB}を用いて別の署名を作成してもよい。このことを知っている他の参加者はボブ４０３だけであるが、アリス４０１も自身の対応する秘密鍵で署名を提供しているので、アリス４０１が証明していると仮定することができる。

【０１３７】

ＢＩＰ３２鍵がこの方式で使用され得ることは上述した。この場合、子鍵は、一般的な形式を有し、

【数２９】

$$sk_{child} = sk_{parent} + hash(K_{parent})$$

ここで、K_{parent}は、親鍵sk_{parent}の公開鍵または秘密鍵のいずれかであり、ハッシュ関数は、ＳＨＡ－５１２ハッシュ関数から、親に依存するチェーンコードと呼ばれる何らかの追加のランダム入力と、複数の子鍵が導出され得るためのカウンタとを用いて作成されたＨＭＡＣである。

【０１３８】

この場合、親鍵$PK_{parent} = sk_{parent} \cdot G$に対応する公開鍵は上記で説明したアイデンティと同じであり、sk_{child}は署名鍵に対応すると仮定すると、代わりに$X = hash(K_{parent})$である上記の２つの証明が成立する。第２項の明示的な形式は、検証者にとって重要ではなく、代わりに、アリスは、第２項が証明の目的であることを知っている。

【０１３９】

最後に、署名鍵は、必ずしもアイデンティティ鍵から導出される必要はないことに留意

されたい。エフェメラル鍵の導出にアイデンティティ鍵を含めるだけで、特定の署名者へのリンクを証明するのに十分である。

【０１４０】

結論

開示された技法の他の変形または使用事例は、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、記載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。

【０１４１】

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得ることが理解されよう。すなわち、本発明は、決して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への上記のいかなる言及も、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への言及に置き換えられ得る。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、上で説明した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いくつかまたはすべてを共有し得る。

【０１４２】

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を作成し、公開し、伝搬し、記憶するという説明した機能のすべてを少なくとも実行する。これらの機能のうちの１つまたはいくつかのみを実行し、すべては実行しない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公開することなく、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は、好ましいビットコインネットワーク１０６のノードとはみなされないことを想起されたい）。

【０１４３】

本発明の好ましくない実施形態では、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ない可能性がある。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を作成し、公開し、伝搬し、記憶するという機能のうちの少なくとも１つまたはいくつかを実行し、すべては実行しない可能性がある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、「ノード」は、ブロック１５１を作成して公開するが、それらのブロック１５１を記憶および／または他のノードに伝搬しないように構成されるネットワークエンティティを指すために使用され得る。

【０１４４】

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」という用語に置き換えられてもよく、そのようなエンティティ／要素は、ブロックの作成、公開、伝搬、および記憶という役割のうちのいくつかまたはすべて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を参照して上述したのと同じ方法でハードウェアに実装され得る。

【０１４５】

上記の実施形態は、単に例として説明されていることが理解されよう。より一般的には、下記ステートメントのうちのいずれか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。

【０１４６】
ステートメント１．デジタル署名を生成するコンピュータ実装方法であって、方法は、署名当事者によって実行され、
第１のメッセージを取得することと、
少なくとも外部データ項目のハッシュに基づいてエフェメラル秘密鍵を生成することと、
第１の署名構成要素および第２の署名構成要素を含む第１の署名を生成することと、
を含み、第１の署名構成要素は、エフェメラル秘密鍵に対応するエフェメラル公開鍵に基づいて生成され、第２の署名構成要素は、第１のメッセージと、エフェメラル秘密鍵と、第１の署名構成要素と、第１の秘密鍵とに基づいて生成される、方法。
【０１４７】
外部データ項目は、署名当事者の識別子、例えば、名前、住所、電話番号、国民保険番号、パスポート番号、公開鍵などを含み得る。
【０１４８】
好ましくは、第１の署名は、ＥＣＤＳＡ署名である。
【０１４９】
ステートメント２．署名当事者が第１の署名を生成したことを証明するために、外部データ項目および第１の署名を検証当事者が利用できるようにすることを含む、ステートメント１に記載の方法。
【０１５０】
ステートメント３．第１のメッセージを取得することは、第１のメッセージを生成することを含み、方法は、第１のメッセージを検証当事者が利用できるようにすることを含む、ステートメント２に記載の方法。
【０１５１】
例えば、署名当事者は、第１の署名および第１のメッセージを検証当事者に送信し得る。代替的に、署名当事者は、第１の署名および第１のメッセージを、例えば、インターネット上、ブロックチェーン上などで公開してもよい。
【０１５２】
ステートメント４．
第２のメッセージを取得することと、
少なくとも第２のメッセージと署名当事者の主秘密鍵とに基づいて第２の署名を生成することと
を含み、外部データ項目は第２の署名を含む、
先行ステートメントのいずれかに記載の方法。
【０１５３】
第２の署名は、それぞれの第１の署名構成要素およびそれぞれの第２の署名構成要素を含み、第１の署名構成要素および第２の署名構成要素の各々は、第２のエフェメラル秘密鍵、すなわち、ランダムエフェメラル秘密鍵に基づく。
【０１５４】
ステートメント５．主秘密鍵に対応する主公開鍵を使用して検証されたときに、第２の署名が第２のメッセージの有効な署名であることを証明するために、第２の署名および第２のメッセージを利用できるようにすることを含む、ステートメント４に記載の方法。
【０１５５】
例えば、署名当事者は、第２の署名および第２のメッセージを検証当事者に送信し得る。
【０１５６】
ステートメント６．第２のメッセージは、第１のメッセージに基づいて生成される、ステートメント４またはステートメント５に記載の方法。
【０１５７】
署名当事者が第２のメッセージを生成し得るか、または第２のメッセージは、別の当事

者、例えば検証当事者から取得され得る。
【０１５８】
ステートメント７．主秘密鍵に対応する主公開鍵は、署名当事者のアイデンティティにリンクされる、ステートメント４から６のいずれかに記載の方法。
【０１５９】
したがって、署名当事者のアイデンティティは、第１の署名内に埋め込まれる。
【０１６０】
ステートメント８．第１の秘密鍵は、少なくとも主秘密鍵に基づいて生成される、ステートメント４から７のいずれかに記載の方法。
【０１６１】
ステートメント９．主秘密鍵は、階層的決定論的鍵構造のマスタ秘密鍵であり、ＨＤ鍵構造は、マスタ秘密鍵に基づいて生成された子秘密鍵のセットを含み、第１の秘密鍵は、子秘密鍵のセットのうちの１つである、ステートメント８に記載の方法。
【０１６２】
ステートメント１０．第１の秘密鍵は、主秘密鍵と、署名当事者と第２の当事者の両方に知られている共通シークレットとに基づいて生成される、ステートメント８に記載の方法。
【０１６３】
ステートメント１１．共通シークレットは、署名当事者の主秘密鍵と、第２の当事者の主秘密鍵に対応する主公開鍵とに基づいて生成される、ステートメント１０に記載の方法。
【０１６４】
ステートメント１２．第２の当事者の主公開鍵は、第２の当事者のアイデンティティにリンクされる、ステートメント１１に記載の方法。
【０１６５】
好ましくは、第２の当事者は、検証当事者とは異なる。
【０１６６】
ステートメント１３．外部データ項目のハッシュは、外部データ項目のダブルハッシュである、先行ステートメントのいずれかに記載の方法。
【０１６７】
ステートメント１４．第２の署名構成要素は、第１のメッセージのハッシュまたはダブルハッシュに基づいて生成される、先行ステートメントのいずれかに記載の方法。
【０１６８】
ステートメント１５．第１のエフェメラル秘密鍵は、ランダムソルト値に基づいて生成される、先行ステートメントのいずれかに記載の方法。
【０１６９】
ステートメント１６．ランダムソルト値は秘密鍵であり、方法は、
第３のメッセージを取得することと、
少なくともランダムソルト値と第３のメッセージとに基づいて第３の署名を生成することと、
ランダムソルト値に対応する公開鍵を使用して検証されたときに、第３の署名が第３のメッセージの有効な署名であることを証明するために、第３の署名、第３のメッセージ、およびランダムソルト値に対応する公開鍵を検証当事者が利用できるようにすることと
を含む、ステートメント１５に記載の方法。
【０１７０】
ステートメント１７．第３のメッセージは第２のメッセージを含む、ステートメント１６に記載の方法。
【０１７１】
例えば、第３のメッセージは、第２のメッセージと同じであってもよい。
【０１７２】

ステートメント１８．第１のメッセージは、ブロックチェーントランザクションの少なくとも一部を含む、先行ステートメントのいずれかに記載の方法。

【０１７３】

第１の署名は、ブロックチェーントランザクションに、例えば、ブロックチェーントランザクションの入力に含まれ得る。

【０１７４】

ステートメント１９．第１のメッセージを検証当事者が利用できるようにすることは、ブロックチェーントランザクションをブロックチェーンネットワークに送信することを含む、ステートメント１８に記載の方法。

【０１７５】

ステートメント２０．第２のメッセージは、ブロックチェーンに関するデータを含む、ステートメント１８またはステートメント１９に記載の方法。

【０１７６】

例えば、ブロックチェーンに関するデータは、ブロックチェーンの現在のブロック高さを含み得る。

【０１７７】

ステートメント２１．デジタル署名が署名当事者によって生成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、検証当事者によって実行され、

第１の署名構成要素および第２の署名構成要素を含む第１の署名を取得することと、

署名当事者から候補の外部データ項目を取得することと、

候補の外部データ項目のハッシュに基づいて候補のエフェメラル秘密鍵を生成することと、

少なくとも候補のエフェメラル秘密鍵に対応する公開鍵に基づいて候補の第１の署名構成要素を生成することと、

候補の第１の署名構成要素が第１の署名構成要素に対応するかどうかに基づいて、第１の署名が署名当事者によって生成されたことを検証することと

を含む方法。

【０１７８】

ステートメント２２．候補の外部データ項目は、第２の署名である、ステートメント２１に記載の方法。

【０１７９】

ステートメント２３．

第２のメッセージを取得することと、

署名当事者の主秘密鍵に対応する主公開鍵を取得することと、

主公開鍵を使用して検証されたときに、第２の署名が第２のメッセージの有効な署名であることを検証することと

を含む、ステートメント２２に記載の方法。

【０１８０】

ステートメント２４．ランダムソルト値に対応する公開鍵を取得することを含み、候補の第１の署名構成要素は、ランダムソルト値に対応する公開鍵に基づいて生成される、ステートメント２１から２３のいずれかに記載の方法。

【０１８１】

ステートメント２５．

第３のメッセージを取得することと、

第３の署名を取得することと、

ランダムソルト値に対応する公開鍵を使用して検証されたときに、第３の署名が第３のメッセージの有効な署名であることを検証することと

を含む、ステートメント２４に記載の方法。

【０１８２】

ステートメント２６．第１の署名は、第１のメッセージに署名し、方法は、

第１の署名を生成するために使用された秘密鍵に対応する第１の公開鍵を取得することと、
第１の公開鍵を使用して検証されたときに、第１の署名が第１のメッセージの有効な署名であることを検証することと
を含む、ステートメント２１から２５のいずれかに記載の方法。
【０１８３】
ステートメント２７．第１の署名、第２の署名、および第３の署名のうちの１つ、いくつか、またはすべては、署名当事者から受信される、ステートメント２１から２６のいずれかに記載の方法。
【０１８４】
ステートメント２８．第１のメッセージ、第２のメッセージ、および第３のメッセージのうちの１つ、いくつか、またはすべては、署名当事者から受信される、ステートメント２１から２７のいずれかに記載の方法。
【０１８５】
ステートメント２９．第１のメッセージ、第２のメッセージ、および第３のメッセージのうちの１つ、いくつか、またはすべては、検証当事者によって生成される、ステートメント２１から２７のいずれかに記載の方法。
【０１８６】
ステートメント３０．第１のメッセージは、ブロックチェーントランザクションの少なくとも一部を含む、ステートメント２１から２９のいずれかに記載の方法。
【０１８７】
ステートメント３１．第１のメッセージを取得することは、ブロックチェーンからブロックチェーントランザクションを取得することを含む、ステートメント３０に記載の方法。
【０１８８】
ステートメント３２．第１の署名を取得することは、ブロックチェーントランザクションから第１の署名を抽出することを含む、ステートメント３０またはステートメント３１に記載の方法。
【０１８９】
ステートメント３３．第２のメッセージは、ブロックチェーンに関するデータを含み、方法は、ブロックチェーンに関するデータを検証することを含む、ステートメント２３またはそれに従属するステートメントのいずれかに記載の方法。
【０１９０】
ステートメント３４．ブロックチェーントランザクションの入力は、第１の署名を含み、方法は、
ブロックチェーントランザクションの入力によって参照される前のブロックチェーントランザクションの出力が署名検証スクリプトを含むことを検証すること
を含む、ステートメント３０またはそれに従属するステートメントのいずれかに記載の方法。
【０１９１】
ステートメント３５．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と
を備え、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３４のいずれかに記載の方法を実行するように構成される、
コンピュータ機器。
【０１９２】
ステートメント３６．コンピュータ可読ストレージ上に具現化されたコンピュータプログラムであって、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

３４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。

【０１９３】

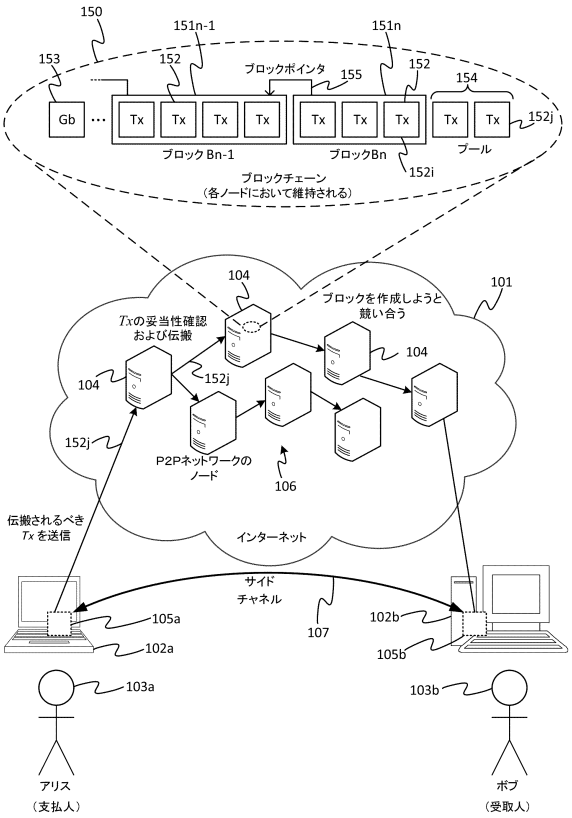
本明細書で開示される別の態様によれば、署名当事者および検証当事者のアクションを含む方法が提供され得る。

【０１９４】

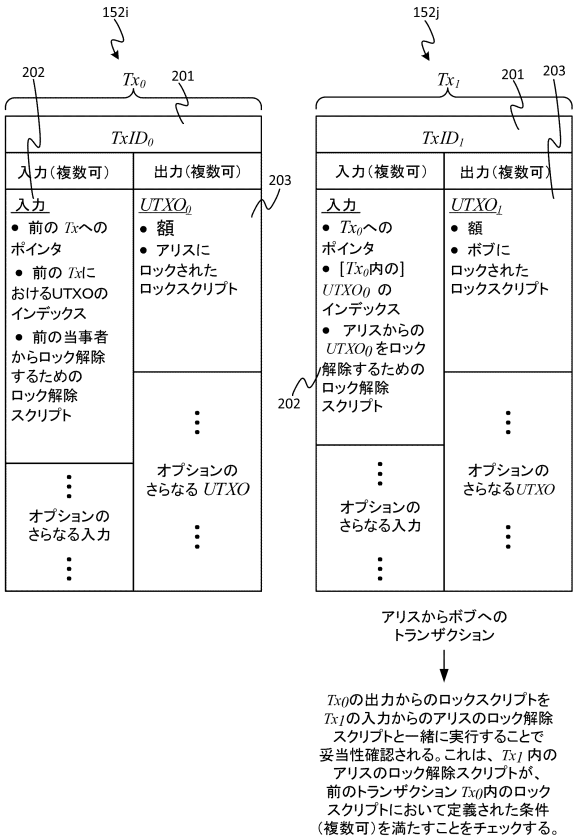
本明細書で開示される別の態様によれば、署名当事者および検証当事者のコンピュータ機器を備えるシステムが提供され得る。

【図面】

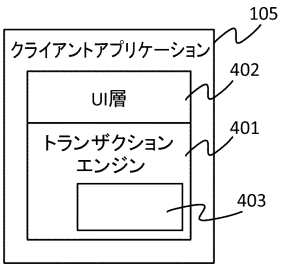
【図１】



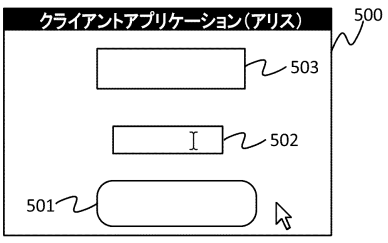
【図２】



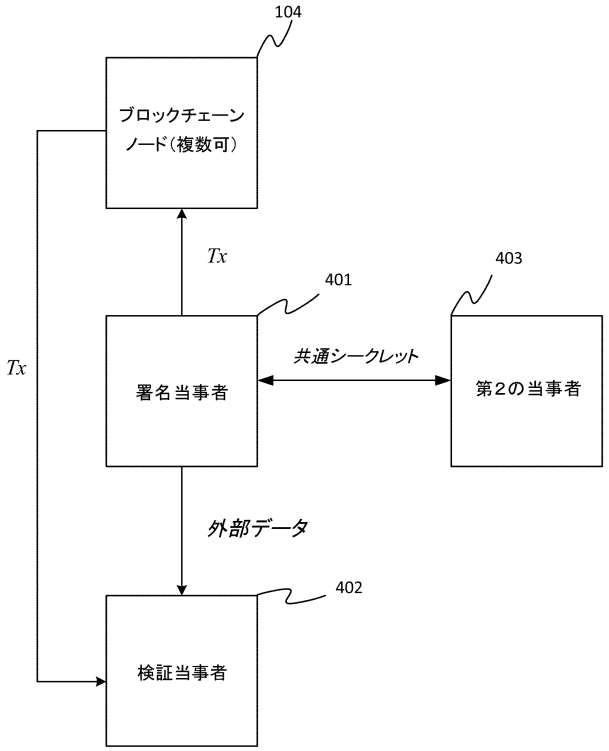
【図３Ａ】



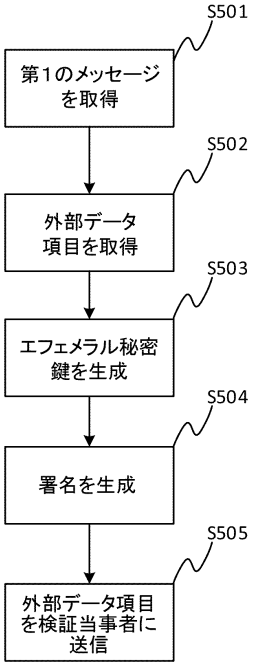
【図３Ｂ】



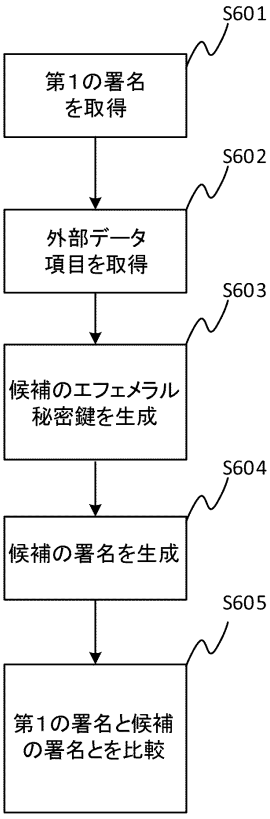
【図４】


104
ブロックチェーン
ノード（複数可）
Tx
401
403
Tx
署名当事者
共通シークレット
第2の当事者
外部データ
402
検証当事者

【図５】


S501
第1のメッセージ
を取得
S502
外部データ
項目を取得
S503
エフェメラル秘密
鍵を生成
S504
署名を生成
S505
外部データ項目
を検証当事者に
送信

【図６】


S601
第1の署名
を取得
S602
外部データ
項目を取得
S603
候補のエフェメラル
秘密鍵を生成
S604
候補の署名を生成
S605
第1の署名と候補
の署名とを比較

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2021/070105

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/08 H04L9/32
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, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2011/213982 A1 (BROWN DANIEL RICHARD L [CA]) 1 September 2011 (2011-09-01) paragraph [0031] - paragraph [0039]	1-36
Y	US 2011/208970 A1 (BROWN DANIEL RICHARD L [CA] ET AL) 25 August 2011 (2011-08-25) paragraph [0049] - paragraph [0050] figure 5	1-36
Y	WO 2017/145048 A1 (NCHAIN HOLDINGS LTD [AG]) 31 August 2017 (2017-08-31) paragraph [0013] - paragraph [0014] paragraph [0129] - paragraph [0241]	1-36

☐ Further documents are listed in the continuation of Box C.

☒ 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 October 2021	14/10/2021

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	Apostolescu, Radu

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

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2021/070105

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011213982 A1	01-09-2011	NONE	
US 2011208970 A1	25-08-2011	NONE	
WO 2017145048 A1	31-08-2017	CN 109314636 A	05-02-2019
		EP 3420669 A1	02-01-2019
		EP 3860037 A1	04-08-2021
		GB 2562622 A	21-11-2018
		HK 1258478 A1	15-11-2019
		JP 2019511151 A	18-04-2019
		KR 20180115768 A	23-10-2018
		US 2019066228 A1	28-02-2019
		WO 2017145048 A1	31-08-2017

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,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,IT,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

(72)発明者　ライト，クレイグ，スティーヴン
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　マッケイ，アレクサンダー
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