

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7653211号
(P7653211)

(45)発行日 令和7年3月28日(2025.3.28)

(24)登録日 令和7年3月19日(2025.3.19)

(51)国際特許分類

F I

H 0 4 L 9/32 (2006.01) H 0 4 L 9/32 2 0 0 C

請求項の数 18 外国語出願 (全30頁)

(21)出願番号	特願2023-171811(P2023-171811)	(73)特許権者	318001991
(22)出願日	令和5年10月3日(2023.10.3)		エヌチェーン ライセンシング アーゲー
(62)分割の表示	特願2020-549646(P2020-549646)の分割		スイス・6 3 0 0・ツーク・グラーフエ
原出願日	平成31年3月18日(2019.3.18)	(74)代理人	100107766
(65)公開番号	特開2023-171896(P2023-171896 A)		弁理士 伊東 忠重
(43)公開日	令和5年12月5日(2023.12.5)	(74)代理人	100070150
審査請求日	令和5年10月3日(2023.10.3)		弁理士 伊東 忠彦
(31)優先権主張番号	1804739.9	(74)代理人	100135079
(32)優先日	平成30年3月23日(2018.3.23)		弁理士 宮崎 修
(33)優先権主張国・地域又は機関	英国(GB)	(72)発明者	トレヴェサン, トーマス
(31)優先権主張番号	1804742.3		イギリス国 シーエフ10 2エイチエイ
(32)優先日	平成30年3月23日(2018.3.23)		チ カーディフ チャーチル ウェイ チャ
	最終頁に続く		ーチル ハウス 7ス フロア アーカート
			ーダイクス アンド ロード エルエルビー
			最終頁に続く

(54)【発明の名称】 ゼロ知識証明を可能にするためのコンピュータ実装されたシステム及び方法

(57)【特許請求の範囲】

【請求項1】

証明者コンピュータが、検証者コンピュータに対して、ステートメントが真であることを、該ステートメントに対する証拠（w）を秘密にしたまま証明するものである、ステートメント（S）のゼロ知識証明又は検証を可能にするためのコンピュータ実装された方法であって、当該方法は、

証明者コンピュータが検証者コンピュータに：

関数回路を実装して、所与の関数回路出力（h）及び楕円曲線点（P）に対して、前記関数回路のワイヤへの関数回路入力（s）が対応する楕円曲線点乗数（s）に等しいかを判定する、ように構成されたm個のゲート及びn個のワイヤを有する算術回路によって表されるステートメント（S）を有するデータであり、前記関数回路は、ハッシュ関数の機能を実装する回路である、データと、

前記回路のワイヤについての個々のワイヤコミットメント及び／又はバッチにされたコミットメントと、

関数回路出力（h）と、

証明キー（PrK）と、

を送信することを含み、

これにより、前記検証者コンピュータが、前記回路が満たされることを決定し、前記楕円曲線点（P）を計算して、前記ステートメントを検証すること、故に、前記証明者コンピュータが前記ステートメントに対する前記証拠（w）を保持していると決定することを

可能にする、

コンピュータ実装された方法。

【請求項 2】

各コミットメントが暗号化される、請求項 1 に記載のコンピュータ実装された方法。

【請求項 3】

前記証明者 コンピュータ は、前記証拠 (w) の知識を証明するために、個々のワイヤコミットメントを送信し、Σ プロトコルを使用して前記検証者 コンピュータ と通信する、請求項 1 に記載のコンピュータ実装された方法。

【請求項 4】

前記証明者 コンピュータ は、前記検証者 コンピュータ からチャレンジ値 (x) を受信し、オープニングで応答する、請求項 1 乃至 3 のいずれか一項に記載のコンピュータ実装された方法。

【請求項 5】

前記証明者 コンピュータ は、前記検証者 コンピュータ が前記ステートメントが真であると判定して前記楕円曲線点 (P) を計算することを可能にするランダム値 (x) を前記検証者 コンピュータ に送信する、請求項 1 乃至 3 のいずれか一項に記載のコンピュータ実装された方法。

【請求項 6】

前記ランダム値 (x) は、少なくとも 1 つのコミットメントの関数である、請求項 5 に記載のコンピュータ実装された方法。

【請求項 7】

前記ランダム値 (x) は、前記証明者 コンピュータ によって生成されて前記検証者 コンピュータ に送信された全ての前記コミットメントの連結をハッシュすることによって計算される、請求項 5 又は 6 に記載のコンピュータ実装された方法。

【請求項 8】

前記コミットメント W_i は、 $W_i = \text{Com}(w_i, r_i)$ であり、

Com は前記関数回路へのコミットメントであり、

w_i はワイヤ値であり、

r_i は、ワイヤコミットメントごとに異なる乱数であり、

i はワイヤ種類であり、

$\text{Com}(w, r) = w \times G + r \times F$ とされ、

F 及び G は楕円曲線点である、

請求項 1 乃至 7 のいずれか一項に記載のコンピュータ実装された方法。

【請求項 9】

前記算術回路内のワイヤ l に対する入力が、 $k_o = r_l \times F$ であり、

k_o はキーオープニング入力であり、

r_l は乱数であり、

F は楕円曲線上の点である、

請求項 8 に記載のコンピュータ実装された方法。

【請求項 10】

前記検証者 コンピュータ は、前記回路が満たされることを確認し、楕円曲線点減算：

$$pk_l = \text{Com}(w_l, r_l) - k_{o_l}$$

を介してワイヤ l に対する公開鍵を計算することができる、請求項 9 に記載のコンピュータ実装された方法。

【請求項 11】

前記証明者 コンピュータ は、ワイヤコミットメントのバッチを送信し、各ワイヤについて楕円曲線点を計算するための乱数を生成して、前記証明キー (PrK) を形成する、請求項 1 に記載のコンピュータ実装された方法。

【請求項 12】

前記証拠に関する前記バッチにされたコミットメントは、

10

20

30

40

50

【数 1】

$$Com(w) = r \times F + \sum_{i=1}^{n-1} w_i \times K_i + w_n \times G$$

であり、

r は、証明者コンピュータによって生成された乱数であり、

証明者コンピュータは、ワイヤ値 w_i (for $i = 1, \dots, n$) のベクトル w へのコミットメントを計算し、

w_n は、キーオープンされるものであり、

K_i は、計算された楕円曲線点であり、

w_i はワイヤ値であり、

F は楕円曲線上の点である、

請求項 1 1 に記載のコンピュータ実装された方法。

【請求項 1 3】

前記算術回路内のワイヤ n に対する入力は、

【数 2】

$$ko_n = r \times F + \sum_{i=1}^{n-1} w_i \times K_i$$

であり、

ko_n はキーオープニング入力であり、

r は乱数であり、

F は楕円曲線上の点である、

請求項 1 2 に記載のコンピュータ実装された方法。

【請求項 1 4】

前記検証者コンピュータは、楕円曲線演算：

【数 3】

$$pk_n = Com(w) - ko_n$$

を介して、キーステートメントワイヤの公開鍵オープニングを計算する、請求項 1 3 に記載のコンピュータ実装された方法。

【請求項 1 5】

前記証明者コンピュータは更に、完全にオープンにされたコミットメントを少なくとも 1 つのワイヤに送る、請求項 1 乃至 1 4 のいずれか一項に記載のコンピュータ実装された方法。

【請求項 1 6】

当該方法は Pedersen コミットメントを使用する、請求項 1 乃至 1 5 のいずれか一項に記載のコンピュータ実装された方法。

【請求項 1 7】

前記ステートメントは、前記関数回路に対して 1 つの算術回路のみを使用する、請求項 1 乃至 1 6 のいずれか一項に記載のコンピュータ実装された方法。

【請求項 1 8】

前記関数回路はハッシュ関数を実装する、請求項 1 乃至 1 7 のいずれか一項に記載のコンピュータ実装された方法。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

30

40

50

この明細書は、概して、例えばブロックチェーンネットワークのノードなどのコンピュータプロセッサ又は一群のそのようなプロセッサにおける実装に適した、コンピュータ実装された方法及びシステムに関する。ステートメントの効率的なゼロ知識検証を可能にするプルーフを生成する改良された方法が提供される。当該方法は、双線形ペアリングフレンドリー楕円曲線の使用を必要としない既存の回路充足可能性のための離散対数ベースゼロ知識プルーフプロトコルに組み込むのに好適である。本発明は、以下に限られないが、特に、プルーフを作成するために証明者によって実行される方法、及びプルーフを検証するために検証者によって実行される、及び2人以上の参加者間の共同作業に適している。参加者間の安全なトラストレスでのやり取りを実現するために、当事者のうちの1つが、キー又はステートメントの知識を、そのステートメントを明らかにすることなく証明することができる。

10

【背景技術】

【0002】

本文書において、用語‘ブロックチェーン’は、全ての形態の電子的な、コンピュータベースの、分散型台帳を含む。これらは、合意に基づく（コンセンサスベースの）ブロックチェーン及びトランザクションチェーンの技術、許可された及び許可されていない台帳、共有台帳、並びにこれらの変形を含む。ブロックチェーン技術の最も広く知られている用途はビットコイン台帳であるが、他のブロックチェーン実装も提案及び開発されている。

【0003】

ここでは、便宜及び説明のために、ビットコインを参照することがあるが、留意されたいことには、本発明は、ビットコインブロックチェーンとの使用に限定されず、それに代わるブロックチェーン実装及びプロトコルも本発明の範囲内にある。用語“ユーザ”は、ここでは、人間又はプロセッサベースのリソースを指す。ブロックチェーンは、トランザクションからなる複数のブロックで構成されるコンピュータベースの非中央集権的な分散システムとして実装されるピア・ツー・ピアの電子台帳である。

20

【0004】

各トランザクションは、ブロックチェーンシステムにおける参加者間でのデジタル資産の管理の移転を符号化するデータ構造であり、少なくとも1つのインプットと少なくとも1つのアウトプットとを含む。各ブロックが、そのブロックと一緒にチェーン化される先行ブロックのハッシュを含むことで、その開始以前にそのブロックチェーンに書き込まれた全てのトランザクションの恒久的で変更不可能な記録を作り出す。トランザクションは、それらのインプット及びアウトプットに埋め込まれるスクリプトとして知られた小プログラムを含み、それらが、トランザクションのアウトプットが誰によってどのようにアクセスされ得るかを指定する。ビットコインプラットフォーム上では、これらのスクリプトは、スタックベースのスクリプト言語を使用して書かれる。

30

【0005】

さらに、この文書では、算術回路を使用する既知のゼロ知識プルーフプロトコル及びシステムの構造を参照する。ブロックチェーンは、第三者仲裁又は第三者預託を必要とせずに2つの相互に信頼のない2つの当事者間で公正なやり取りすることの問題に対するソリューションを可能にする非中央集権化された許可なしでのグローバル機構を提供した。金銭的報のための又は例えばデジタルグッズなどの情報との交換での、データ又は情報の公正な交換は、ZKCP（Zero-Knowledge Contingent Payments）（非特許文献2）として知られるトランザクションプロトコルにて具体化されている。ZKCPでは、支払が確認された場合にのみ、指定されたデータが売り手から買い手に転送され、買い手から売り手への支払は、この指定されたデータが売却の条件に従って有効である場合にのみ完了される。このようなプロトコルの詳細は知られている（非特許文献1）が、それは基本的に、ハッシュ時間ロック契約（hash-time-locked contract；HTLC）とゼロ知識プルーフとの組み合わせに基づいており、これは、暗号化された情報（‘デジタルグッズ’）が有効／正確であることと、この情報を解読する‘パスワード’が支払いを請求するためにブロックチェーン上で明らかにされなければならないデータであることとを同時に検証

40

50

するものである。

【0006】

ZKCPプロトコルの中心コンポーネントは、データ／情報の妥当性又は正確性、キーの妥当性、及び対応するハッシュ値に関する一連の従属ステートメントに対するゼロ知識プルーフである。このような複雑な複合ステートメントは、全般的な計算のために効率的なゼロ知識プルーフシステムを必要とし、突き詰めると、これは、1つの当事者が、秘密の入力を用いて任意のプログラムを走らせ、そして、秘密の入力又はプログラムの実行に関する情報を何ら明らかにすることなく、他の当事者に対し、プログラムが入力を有効であると認めて正しく実行されたことを証明することを可能にする。既知のZKCPの例では、採用された汎用ゼロ知識プルーフシステムは、Pinocchioプロトコル（非特許文献3）及びC++ libsnarkライブラリ（非特許文献4）で実装されるような、知識の簡潔な非対話型論証（succinct non-interactive arguments of knowledge；SNARK）フレームワークに基づいてきた。

10

【0007】

ゼロ知識SNARK（zkSNARK）は、算術回路として表現可能な任意の計算の妥当性をゼロ知識で証明する方法を提供する。zkSNARKの2つの主な特徴的性質は、それらが非対話的である（証明者が検証者に一手（1ムーブ）でプルーフを送る）こと及び簡潔である（プルーフが小さくて検証容易である）ことである。しかしながら、それらはかなりの制限を持つ：

- － プルーフ生成が、極めて計算的要求が大きいものである。
- － 証明キーが非常に大きく、回路サイズに比例する。
- － それらは、強力にテストされていない暗号仮定（すなわち、エクスポート仮定及びペアリングベース仮定の知識）に依存する。
- － 所与のプログラム（回路）に対して、それらは、設定パラメータを削除するために信頼されなければならない第三者によって共通参照文字列（common reference string；CRS）が計算されることを必要とする。設定パラメータの知識を持つ者が、偽のプルーフを作成する能力を持つ。

20

【0008】

任意の暗号化楕円曲線キー操作を伴うステートメントを証明するzkSNARKの構築は、今日まで試みられていないが、仮説的に、何十万又は何百万ものゲートを備える算術回路で構成されることになり、結果として、プルーフ生成時間が数分かかることになるとともに、証明キーが数百メガバイトのサイズになるであろう。

30

【0009】

技術的背景

対話型ゼロ知識プルーフのための基本システムは、証明者と検証者との間での何度かの通信ステップを伴う Σ （シグマ）プロトコルを使用することができる。通常、 Σ プロトコルは三手（3ムーブ）を必要とし、すなわち、証明者が最初のコミットメント（a）を検証者に送り、次いで、検証者がランダムチャレンジ（x）で応答し、最後に証明者が最終応答すなわち‘opening’（z）で応答する。そして、検証者が、トランスクリプト（a，x，z）に基づくステートメントを受け入れる又は拒絶する。

40

【0010】

Σ プロトコルは、証明者のみに知られている証拠（w）の知識又は該証拠（w）についてのステートメントを証明するために使用されることができる。コミットメントが、証拠に関するステートメントが真実であるという事実を除いて、証拠に関する情報すなわち秘密を検証者に明らかにしない場合、このプロトコルはゼロ知識である（非特許文献5）。

【0011】

多くの対話型ゼロ知識プロトコルの中心は、算術回路の充足可能性に関して使用されるものであるコミットメント方式である。コミットメントによって、証明者が前もって秘密値にコミットし、その後、検証可能に秘密値を明らかにする（開く）ことが可能にされる。コミットメントスキームは、2つの主な性質を持つ。第一に、それは隠れていて、コミ

50

ットメントが値を秘密のままにする。第二に、それは結びついていて、コミットメントは、元のコミットされた値に対してのみ開かれることができる。Pedersenコミットメント（非特許文献5）スキームは、全ての当事者に知られた素数位数 p の群 G の中の G 及び F という、2つの楕円曲線生成点を含む。コミットする者（コミッタ）が、素な整数 $\mathbb{Z}_p$ のフィールドでセキュアな乱数 r を生成し、秘密値 s へのコミットメント：

$$\text{Com}(s, r) = s \times G + r \times F$$

を（楕円曲線加算／乗算によって）計算し、ここで、 $\times$ は楕円曲線点乗算を表す。

【0012】

コミッタは、後の段階で、値 s 及び r を提供することによって、コミットメントを完全に開くことができる（すなわち、それが検証されることができる）。コミッタはまた、秘密 s 又は乱数 r を明らかにすることなく、 Σ プロトコルの一部として特定のチャレンジ値に応答してコミットメントを開くことができる。

10

【0013】

Pedersenコミットメントは相加的に同形（ホモモルフィック）であり、すなわち、（楕円曲線上で）2つのコミットメントを加えることが、コミットされた値の和に対するコミットメントをもたらす、すなわち：

$$(s_1 \times G + r_1 \times F) + (s_2 \times G + r_2 \times F) = (s_1 + s_2) \times G + (r_1 + r_2) \times F$$

である。

【0014】

算術回路充足可能性の証明は‘ゼロ知識’で達成されることができる。（フィールド $\mathbb{Z}_p$ 上の）算術回路は、ワイヤによって接続され（有向非循環グラフを形成する）算術ゲートの仮想構成であり、これは、任意の複雑な計算を実行することが可能であり、計算は整数演算に限定され、データ依存ループ又は可変状態を持たないものでなければならない。

20

【0015】

各ゲートが、2つの入力ワイヤ及び1つの出力ワイヤを持ち、入力に対して乗算（ $\times$ ）又は加算（ $+$ ）演算を実行する。図1（a）は、左（ w_L ）及び右（ w_R ）のワイヤ入力と1つのワイヤ出力（ w_O ）を有する乗算ゲートの概略図を示し、図1（b）は、3つのゲート、3つの入力ワイヤ（ w_1, w_2, w_3 ）、1つの出力ワイヤ（ w_6 ）、及び2つの内部ワイヤ（ w_4, w_5 ）を有する単純な算術回路の概略図を示している。

30

【0016】

実際には、完全な回路は、外部（回路）入出力値を定めるフリー入力ワイヤ及びフリー出力ワイヤを持つ。合法的割り当て（リーガルアサインメント）は、ワイヤの値を、回路を満たすものとして定めるものであり、すなわち、各ワイヤが値を割り当てられ、各ゲートの出力が入力の積又は和に正確に対応する（すなわち、そのゲートは一貫性がある）。

【0017】

所与の算術回路について、証明者は、まず（Pedersenコミットメントを用いて）合法的割り当てにて各ワイヤ値にコミットし、次いで、ワイヤ値を証拠として、（並列に実行されることができる）回路内の各ゲートに対して検証者を持つ特別な Σ プロトコルを、実行することによって、ワイヤ値を明らかにすることなく、回路に対する合法的割り当てを知っていることを検証者に対して証明することができる。これらの Σ プロトコルは、後述するように、Pedersenコミットメントの同形性を活用する。

40

【0018】

（回路が満たされることの）プルーフを生成するために、最初に証明者は回路内の各ワイヤ w_i （ワイヤ数を n として、 $i = 1, \dots, n$ ）に対するコミットメント：

$$W_i = \text{Com}(w_i, r_i)$$

を生成し、これらを検証者に送る。

【0019】

回路内の各‘加算’ゲート（図1（b）に1つ示される）に対して Σ_{ero} プロトコルが実行され、これは、 $w_L + w_R - w_O = 0$ であること（すなわち、入力ワイヤ w_L 及び w_R

50

が出力ワイヤ w_0 に等しくて、加算ゲートが満たされること)を(ゼロ知識で)証明することを伴う。これは、以下のステップを含み、すなわち：

1. 証明者が、ゼロに対するコミットメント： $B = \text{Com}(0, r_B)$ を生成して検証者に送る。
2. 検証者が、ランダムチャレンジ値： $x \leftarrow Z_p$ で応答する。
3. 次いで、証明者が、開ける値(オープニング値)： $z = x(r_L + r_R - r_0) + r_B$ を計算し、それを検証者に送る。
4. 検証者が、 $w_L + w_R - w_0 = 0$ であることのプルーフとして、 $\text{Com}(0, z) = x \times (w_L + w_R - w_0) + B$ を証明する。

B は、公開鍵と同様の曲線点を表す； $B = r \times F + 0 \times G$

r_B は、対応するペアの秘密鍵を表す。

【0020】

各‘乗算’ゲート(図1(a)に示される)に対して Σ_{prod} プロトコルが実行され、これは、各乗算ゲートに対して $w_L \cdot w_R = w_0$ であること(すなわち、乗算ゲートが満たされること)を(ゼロ知識で)証明することを伴う。

1. 証明者が、5つのランダムバイディング値： $t_1, t_2, t_3, t_4, t_5 \leftarrow Z_p$ を生成する。
2. 証明者が、 $C_1 = \text{Com}(t_1, t_3)$ 、 $C_2 = \text{Com}(t_2, t_5)$ 、 $C_3 = t_1 \times w_R + t_4 \times F$ を計算し、これらを検証者に送る。
3. 検証者が、ランダムチャレンジ値： $x \leftarrow Z_p$ で応答する。
4. 証明者が、開ける値：

$$e_1 = w_L x + t_1$$

$$e_2 = w_R x + t_2$$

$$z_1 = r_L x + t_3$$

$$z_2 = r_R x + t_5$$

$$z_3 = (r_0 - w_L r_R) x + t_4$$

を計算し、これらを検証者に送る。

5. そして、検証者が、 $w_L \cdot w_R = w_0$ であることの証明として、次式：

$$\text{Com}(e_1, z_1) = x \times w_L + C_1$$

$$\text{Com}(e_2, z_2) = x \times w_R + C_2$$

$$e_1 \times w_R + z_3 \times F = x \times w_0 + C_3$$

を検査する。

【0021】

Σ_{zero} プロトコル及び Σ_{prod} プロトコルは、回路内の各ゲートの検証のために並列に動作されることができ、全てのゲートに対して同じ検証チャレンジ値(x)を使用することができる。

【0022】

一例として、図1(b)の回路を考えると、合法的割り当て(すなわち、回路を満たすワイヤ値)を知っていることを証明者が検証者に対してゼロ知識で証明するために、証明者は、最初に、各ゲートに対するワイヤコミットメント($w_1, \dots, w_6$)及び Σ プロトコルコミットメント(すなわち、各加算ゲートに対して1つの追加コミットメント、各乗算ゲートに対して5つの追加コミットメント)を検証者に送る。

【0023】

次いで、検証者がランダムチャレンジ $x \leftarrow Z_p$ で応答し、証明者が各ゲートの開ける値(各加算に1つ、各乗算に5つ)を計算して検証者に送り返す。そして、検証者が、 Σ プロトコルチェックを実行して、

$$w_1 \cdot w_2 = w_4$$

$$w_4 \cdot w_5 = w_6$$

$$w_2 + w_3 = w_5$$

であり、故に、従って、コミットメント $w_1, \dots, w_6$ がワイヤ値 $w_1, \dots, w_6$ を満たす

10

20

30

40

50

ことに対応する、ことを検証する。

【0024】

証明者が、回路を満たすことに加えて、特定のワイヤが特定の値を持つことを示すことを望む場合、それらは、関連するワイヤに対するコミットメントを完全に開くことができる。この例では、検証者は更に、 w_6 が特定の合法的な割り当てからの実際の出力であることを示すために、値 w_6 及び r_6 を検証者に送ることができる（そして、検証者は、 $W_6 = \text{Com}(w_6, r_6)$ であることを検証することができる）。

【0025】

図1(b)の例は、簡単な回路である。実際には、有用な回路はもっと多くのゲートで構成される。特に関心があるのは、SHA-256ハッシュ関数に関する算術回路であり、この回路は、証明者が、特定の（出力）値にハッシュするSHA-256関数へのプリイメージ（入力）を知っていることを、プリイメージを明らかにすることなく示すことを可能にする。SHA-256アルゴリズムに関する回路の最も効率的な実装の1つは、27,904個の算術ゲートからなる（Zcash2016）。すると、SHA-256プリイメージの知識を証明することは、上のプロトコルの最初のコミットメント及びオープニングラウンドの両方で約5MBのデータを送信することを必要とすることになり、証明者と検証者の両方に対して約200,000回の楕円曲線演算を必要とする（各々が数秒のプロセッサ時間を要する）。

【0026】

算術回路充足可能性を証明するためについての並列Σプロトコルアプローチの性能を有意に改善するために開発された方法が幾つか存在する。既知のアプローチ（非特許文献5、非特許文献6）は、証明者から検証者に送られなければならないデータのサイズを実質的に減らす（すなわち、通信の複雑さを減らす）ために、回路ワイヤ値へのコミットメントをバッチにすることを伴う。これらの方法は、通信複雑性が $O(n)$ から $O(\sqrt{n})$ 又は $O(\log(n))$ に低減される証明システムを可能にする。

【0027】

これまた、同じSHA回路の充足可能性を証明することについての比較として、そのプロトコル（非特許文献5）は、ほんの5KBの証明キーサイズ及び180msの鍵生成時間を持つ。プルーフサイズは24KBであり、生成するのに約4秒かかり、そのプルーフはまた、検証するのに約4秒かかる。

【0028】

これらの方法については、採用されているメインベクトルバッチングプロトコルを以下のステップで説明することを除いて、ここで完全に説明することはしない。これは、通常のPedersenコミットメントと同じ性質に従うが、 n 個の要素($m = m_1, \dots, m_n$)へのコミットメントが、単一のグループ要素を送信することを要するのみであり：

1. 証明者及び検証者が、グループ要素 $F \leftarrow G$ について合意する。
2. 証明者が、 n 個の乱数 $x_1, \dots, x_n \leftarrow Z_p$ を生成する。
3. 証明者が、点 $K_i = x_i \times F$ ($\text{for } i = 1, \dots, n$)を計算する。これらの値は、検証者に送られる証明キー PrK を形成する。
4. 証明者が、乱数： $r \leftarrow Z_p$ を生成する。
5. 証明者が、コミットメント：

【数1】

$$\text{Com}(\mathbf{m}) = r \times F + \sum_{i=1}^n m_i \times K_i$$

を計算し、それを検証者に送信する。

【先行技術文献】

【非特許文献】

【0029】

10

20

30

40

50

【文献】Campanelli, Matteo, et al. "Zero-knowledge contingent payments revisited: Attacks and payments for services." Commun. ACM (2017).

【文献】<https://github.com/zcash-hackworks/pay-to-sudoku>

【文献】Parno, Bryan, et al. "Pinocchio: Nearly practical verifiable computation." Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013.

【文献】<https://github.com/scipr-lab/libsnark>

【文献】Bootle, Jonathan, et al. "Efficient zero-knowledge proof systems." Foundations of Security Analysis and Design VIII. Springer, Cham, 2015. 1-31.

【文献】Groth, Jens. "Linear Algebra with Sub-linear Zero-Knowledge Arguments." CRYPTO. Vol. 5677. 2009.

【文献】<https://bitcointalk.org/index.php?topic=81865.msg901491#msg901491>

【文献】Bootle, Jonathan, et al. "Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2016.

【文献】Standards for Efficient Cryptography (SEC) (Certicom Research, <http://www.secg.org/sec2-v2.pdf>)

【文献】Bitcoin Developer Reference (<http://bitcoin.org/en/developer-reference>)

【発明の概要】

【0030】

全体として、本発明は、ステートメントのゼロ知識証明又は検証を可能にするためのコンピュータ実装された方法にある。証明者は、ここでの方法を用いて、ステートメントに対する証拠を秘密にしながら、ステートメントが真であることを検証者に対して証明することができる。これらのステートメントは、算術回路充足可能性と公開鍵の妥当性に関する従属ステートメント（キーステートメントブルーフ）の両方を同時に必要とする複合ステートメントである。

【0031】

ここでの方法は、例えば既存の離散対数ベースのゼロ知識ブルーフプロトコルなどの、回路充足可能性に関する既知のプロトコルにおいて使用されることができる。当該方法は特に、双線形ペアリングフレンドリー楕円曲線の使用を必要としないプロトコルに適している。

【0032】

当該方法においては、証明者が検証者に、所与の関数回路出力及び楕円曲線点に関するものであるステートメントを含む一組のデータを送信することを含み、関数回路入力、対応する楕円曲線点乗数（ s ）に等しい。該データは、ステートメントの回路についての個々のワイヤコミットメント及び／又はバッチにされたコミットメント、入力、及び出力を含む。証明者は、ステートメントで使用される楕円曲線又は各楕円曲線の仕様を、データに含めることができ、又は予め共有していることができる。そして、証明者は、検証者からのチャレンジに応答してオープニングを送信する。あるいは、証明者は更に証明キーを含める。

【0033】

証明者から受信したデータを用いて、検証者は、回路が満たされると判定して、ステートメントを検証すること、故に、証明者がステートメントに対する証拠を保持していると決定することができる。楕円曲線点も計算されることができる。データを受信すると、検証者は、計算を通じて、データがステートメントに適合することを決定する。本発明は特に、ハッシュプリイメージ及び楕円曲線秘密鍵の等価性のゼロ知識証明に適する。

【0034】

従って、本発明によれば、添付の特許請求の範囲に規定される方法及びシステムが提供

10

20

30

40

50

される。

【0035】

故に、証明者が、検証者に対して、ステートメントが真であることを、該ステートメントに対する証拠（W）を秘密にしたまま証明するものである、ステートメント（S）のゼロ知識証明又は検証を可能にするためのコンピュータ実装された方法を提供することが望ましい。この証明は、明示的な証明とすることができる。

【0036】

証明者が、検証者に対して、ステートメントが真であることを、該ステートメントに対する証拠（w）を秘密にしたまま証明するものである、ステートメント（S）のゼロ知識証明又は検証を可能にするためのコンピュータ実装された方法を提供することができ、当該方法は、

10

証明者が検証者に：

関数回路を実装して、所与の関数回路出力（h）及び楕円曲線点（P）に対して、関数回路のワイヤへの関数回路入力（s）が対応する楕円曲線点乗数（s）に等しいかを判定する、ように構成されたm個のゲート及びn個のワイヤを有する算術回路によって表されるステートメント（S）と、

回路のワイヤについての個々のワイヤコミットメント及び／又はバッチにされたコミットメントと、

関数回路出力（h）と、

証明キー（PrK）と、

20

を送信することを含み、

これにより、検証者が、回路が満たされることを決定し、楕円曲線点（P）を計算して、ステートメントを検証すること、故に、証明者がステートメントに対する証拠（w）を保持していると決定することを可能にする。

【0037】

当該方法は、証明者が一組のデータ（データのセット）を検証者に送信することを含む。データのセットは、関数回路を実装し、所与の関数回路出力（h）及び楕円曲線点（P）に対して、関数回路への又は関数回路内のワイヤへの関数回路入力（s）が対応する楕円曲線点乗数（s）に等しいかを判定する、ように構成されたm個のゲート及びn個のワイヤを有する算術回路を持つステートメントを含む。関数回路は、ハッシュ関数の機能を実装する回路とし得る。ハッシュ関数回路に対する又は関数回路内のワイヤに対するプリイメージは、対応する楕円曲線点乗数に等しいことができる。

30

【0038】

データはまた、個々のワイヤコミットメント及び／又はバッチにされたコミットメントを含む。該コミットメント又は各コミットメントは、回路のゲートに関するワイヤ入力及び出力（これらは暗号化される）とし得る。データはまた、入力を含む。入力は、算術回路〔楕円曲線点（P）〕のワイヤに対するキーオープニングとしての働きをする。証明者又は検証者のいずれかがワイヤに名前を付け得る。入力又はキーオープニングは、回路内の最初のワイヤに対するものとし得る。データはまた、関数回路出力を含む。ステートメント内で使用される楕円曲線又は各楕円曲線の仕様をデータに含めることができる。

40

【0039】

データを送信した後、証明者は、検証者からチャレンジ値を受信し、オープニングで応答する。オープニングは、 Σ （シグマ）プロトコルに従った値ステートメントとし得る。オープニング値は、検証者が、ステートメントが真であると判定して楕円曲線点を計算することを可能にする回路の各ゲートに対するものとし得る。

【0040】

チャレンジを待つことの代わりとして、証明者は更に証明キーを検証者に送信し得る。証明キーは、プルーフの一部であるデータから生成され得る。証明キーは、プルーフに使用される乱数のうちの1つ以上の乱数のハッシュとし得る。

【0041】

50

検証者に送信されるデータは、検証者が、回路が満たされることを決定し、楕円曲線点を計算し、ステートメントを検証すること、故に、証明者がステートメントに対する証拠を保持していると決定することを可能にする。

【0042】

検証者に送信されるデータのセット、及び／又は検証者に送信されるチャレンジへのオープンニングは、検証者とは無関係に作成されるキーのように機能することができる。検証者からのチャレンジは、証明者の身元及びキーの完全性を決定することに類似している。

【0043】

入力又はキーオープンニングは、算術回路内の最初のワイヤに対するものとし得る。しかしながら、中間ワイヤの知識を証明することは、最初のワイヤの知識を証明することよりも困難であるため、ランダムなワイヤが選択されることが好ましい。さらに、最初のワイヤ以外のランダムなワイヤを選択することは、よりロバストであり、悪意ある第三者によるプルーフ又は証拠の発見を阻止する。

【0044】

検証者が、証明者から受信したデータを分析することによって、ステートメントに対する証拠（w）を知ることなく、ステートメントが真であることを検証するものであるステートメントのゼロ知識証明又は検証を可能にするための相補的なコンピュータ実装された方法を提供することが等しく望ましい。明確にしておくに、本発明の方法は、プラグソケット式に検証者によってとられる逆のアクションにまで及ぶ。本発明は、証明者と検証者との間の完全な共同作業にまで及ぶ。

【0045】

証明者は、チャレンジ値を待つことに加えて、あるいは代えて、検証者が、ステートメントが真であると判定して楕円曲線点を計算することを可能にするランダム値を検証者に送信し得る。証明者からのデータの受信を受けて、検証者は代わりに、検証者が、ステートメントが真であると判定して楕円曲線点を計算することを可能にするランダム値を受信し得る。ランダム値は、少なくとも1つのコミットメントの関数とし得る。この関数は、ハッシュ関数とすることができる。

【0046】

ランダム値又はチャレンジは、プロセスの利便性及び効率を向上させるために置き換えられ得る。また、証拠に関する情報を抽出しようという試みにおいて検証者が非ランダムなチャレンジを生成することに伴うリスクも存在する。さらに、チャレンジ値を、証明者によって提供されるランダム値で置き換えることは、この方法に対話型のものから非対話型のものに変換する。証明者は、独立的且つ公開的に検証されることが可能なプルーフをオフラインで生成することができる。ランダム値は、ハッシュ関数からの出力とし得る。ランダム値（x）の代わりに、1つ以上のコミットメントのハッシュからの出力を用いることは、フィアットーシャミア（Fiat-Shamir）原理を利用する。

【0047】

ランダム値は、証明者によって生成されて検証者に送信された全てのコミットメントの連結をハッシュすることによって計算され得る。

【0048】

コミットメントは、 $W_i = \text{Com}(w_i, r_i)$ とすることができ、

Com は関数回路へのコミットメントであり、

w_i はワイヤ値であり、

r_i は乱数であり、すなわち、これはワイヤコミットメントごとに異なり、

i はワイヤ種類であり、

$\text{Com}(w, r) = w \times G + r \times F$ とされ、

F 及び G は楕円曲線点である。

【0049】

算術回路内のワイヤ l に対する入力は、 $k_o = r_l \times F$ とすることができ、

k_o はキーオープンニング入力であり、

10

20

30

40

50

r_l は乱数であり、
 F は楕円曲線上の点である。

【0050】

ワイヤは、回路内の最初のワイヤとし得る。

【0051】

検証者は、回路が満たされることを確認することができ、楕円曲線点減算：

$$pk_l = Com(w_l, r_l) - ko_l$$

を介してワイヤ l に対する公開鍵を計算することができる。

【0052】

証明者は、ワイヤコミットメントのバッチを送信し得るとともに、各ワイヤについて楕円曲線点を計算するための乱数を生成して、証明キーを形成し得る。

10

【0053】

証拠に関するバッチにされたコミットメントは、

【数2】

$$Com(w) = r \times F + \sum_{i=1}^{n-1} w_i \times K_i + w_n \times G$$

とすることができ、

r は、証明者によって生成された乱数であり、

20

証明者は、ワイヤ値 w_i （for $i = 1, \dots, n$ ）のベクトル w へのコミットメントを計算し、

w_n は、キーオープンされるものであり、

K_i は、計算された楕円曲線点であり、

w_i はワイヤ値であり、

F は楕円曲線上の点である。

【0054】

算術回路内のワイヤ n に対する入力は、

【数3】

$$ko_n = r \times F + \sum_{i=1}^{n-1} w_i \times K_i$$

30

であり、

ko_n はキーオープニング入力であり、

r は乱数であり、

F は楕円曲線上の点である。

【0055】

入力は、最初のワイヤに対するものとし得る。

【0056】

検証者は、楕円曲線演算：

40

【数4】

$$pk_n = Com(w) - ko_n$$

を介して、キーステートメントワイヤの公開鍵オープニングを計算し得る。

【0057】

証明者は更に、完全にオープンにされたコミットメントを少なくとも1つのワイヤに送り得る。当該方法はPeder senコミットメントを使用し得る。ステートメントは、関数回路に対して1つの算術回路のみを使用し得る。関数回路は、好ましくはSHA-2

50

56 ハッシュ関数であるハッシュ関数を実装し得る。

【0058】

当該方法は、証明者によって、暗号鍵などのデータに対するゼロ知識付帯トランザクション（これは、ゼロ知識付帯トランザクションとし得る）を可能にするために使用されることができ、証明者は、提供されるデータ（これは、バニティアドレスとし得る）及び受信されるデータ（該データは、UTXOの形態の支払とし得る）を確認するために検証者と連携し、検証者との通信チャンネル（該チャンネルはオープンとし得る）を確立し、証明者は、検証者から、安全なランダム秘密鍵 $s_k B$ から検証者が生成した楕円曲線公開鍵 $p k_B$ を受信し、 $p k_V = s k_V \times G$ であり、 G は楕円曲線点であり、

証明者は、データ $= p k_V + i \times G$ であるように、提供されるデータをロック値 i で保護する。

【0059】

証明者は、 i を変えることによって得られたBase58符号化アドレスにおける必要パターンの検索を実行しているとし得る。証明者は、 $p k_P = i \times G$ であるそれらの公開鍵と、関数回路入力（例えば、プリイメージ）がロック値 i である関数回路からの出力 $f(i)$ と、を検証者に送信する。

【0060】

証明者は、関数回路への入力が、 $p k_P$ に対応する秘密鍵であることを検証者に証明するステートメントプルーフを検証者に送信することができ、故に、検証者がプルーフを検証し、 $p k = p k_V + p k_P$ に対応するアドレスが合意されたパターンと一致することを確認することで、ロック値 i を知ることが、データに関する完全な秘密鍵の導出を可能にすることと、ロック値 i が関数回路への関数回路入力であることと、を特定することを可能にし得る。

【0061】

証明者は、証明者からの署名と関数回路入力とによってアクセスされることが可能な、受信されるデータを含む出力を含んだトランザクション $T x_1$ を受信し得る。該トランザクションは、ハッシュ時間ロック関数とし得る。受信されるデータが、UTXOへのアクセスを提供し得る。

【0062】

証明者は、該トランザクションに署名して該トランザクションをブロックチェーン上でブロードキャストすることができ、それがブロックへとマイニングされて、該トランザクションをアンロックするための署名及び値 i を供給する第2のトランザクション $T x_2$ を提供することによって、トランザクション $T x_1$ の出力からデータに証明者がアクセスすることを可能にし、そして、該トランザクションは、ブロックチェーン上で明らかにされ、斯くして、検証者が、ロック値 i を特定して、証明者によって提供されたデータにアクセスすることを可能にし、

$s k = s k_B + i$ であり、

$p k = s k \times G$ である。

【0063】

証明者によって提供されるデータはバニティアドレスを含み得る。検証者から受信されるデータは暗号通貨支払（例えば、UTXO）を含み得る。

【0064】

トランザクションは、完全にアトミックでトラストレスとすることができ、買い手は有効な値 i を提供した場合にのみ支払いを受け、値 i はブロックチェーン上で公然に明かされる。秘密鍵の分割により、ブロックチェーン上で露わにされる値は、他の誰にも役に立たず、完全な秘密鍵のセキュリティを損なわない。

【0065】

あるコンピュータ実装された方法は、証明者が検証者との（第三者による集中化された交換を用いない）トラストレスでの公正なデータ交換を実行することを伴い得る。これは、クロスチェーン・アトミック・スワップ又はアトミックトレードとして記述されること

10

20

30

40

50

ができる。何故なら、この文脈では、両者がトランザクションを完了するか両者とも完了しないかのいずれかであるという公正な交換性を参照するからである。このスワップは、ハッシュされた時間ロックされた契約を可能にするスクリプト機能をサポートするブロックチェーン間で実行されることができる。

【0066】

証明者は、第1のブロックチェーン上の例えば1ビットコインのUTXOなどの第1のデータへのアクセスを持ち、検証者は、第2のブロックチェーン上にある例えば100LTCなどの第2のデータへのアクセスを持ち、証明者及び検証者は、データを交換することに合意する。当該方法は、証明者が、第2のブロックチェーンに対する鍵ペアを生成し、公開鍵を検証者に送信し、秘密鍵を保持することと、証明者が、第1のブロックチェーンに対する検証者公開鍵を受信し、検証者は、第1のブロックチェーンに対する鍵ペアを生成して秘密鍵(s_B)を保持していることと、証明者が、ステートメントと、1つ以上のコミットメントと、入力若しくはキーオープニング及び関数回路出力(h)と、楕円曲線仕様とを送信することと、を含む。

【0067】

証明者は、第1のデータを共通の公開鍵アドレスに送信する第1のブロックチェーントランザクション Tx_A を作成することができ、そして、該トランザクションを第1のブロックチェーンネットワーク上でブロードキャストし、該アドレスは、入力と検証者公開鍵との和によって定められる。このデータは、24時間以内にスワップが実行されなかった後に、証明者によってアクセスされることができる。

【0068】

証明者は、第2のブロックチェーントランザクション Tx_B を検証することができ、該トランザクションは、第1のブロックチェーンに第1のブロックチェーントランザクション Tx_A が含まれることを確認した後に、検証者によって作成されて第2のブロックチェーンネットワーク上でブロードキャストされ、該トランザクションは、100LTCの形態の第2のデータを証明者公開鍵アドレスに送り、該証明者公開鍵アドレスは、該証明者公開鍵アドレスに関する有効な署名、及び関数回路出力を決定する関数回路入力である値、を用いて証明者によってアクセス可能である。このデータは、24時間以内にスワップが実行されなかった後に、検証者によってアクセスされることができる。

【0069】

証明者は、第2のブロックチェーントランザクション Tx_B が第2のブロックチェーン上に含まれることを確認し、署名と、関数回路出力の関数回路入力である上記値と、を提供することによって第2のデータにアクセスし、斯くして、検証者が、関数回路出力を決定する関数回路入力である上記値を観測し、(楕円曲線点乗算の同形性から $s_B + s$ である P_C)の秘密鍵を用いて署名を提供することによって第1のデータにアクセスすることを可能にする。

【0070】

交換されるデータは暗号通貨とすることができ、第1のデータは、好ましくはビットコイン(Bitcoin)である第1の暗号通貨の量に対応し、第2のデータは、好ましくはライトコイン(Litecoin)である第2の暗号通貨の量に対応する。

【0071】

上述のように、証明者による全てのアクションは、プルーフを検証するための検証者による逆のアクションを必要とする。本発明は、検証者によって実行される方法又はアクションにまで及ぶ。従って、証明者が、検証者に対して、ステートメントが真であることを、該ステートメントに対する証拠を秘密にしたまま、好ましくは明示的に、証明するものである、ステートメントのゼロ知識証明又は検証を可能にするためのコンピュータ実装された方法が提供され、当該方法は、検証者が証明者から：好ましくはハッシュ関数である関数回路を実装して、所与の関数回路の、そして好ましくは指定された関数回路の出力及び楕円曲線点に対して、関数回路への関数回路入力又はプリイメージが楕円曲線点乗数に等しいかを判定する、ように構成された m 個のゲート及び n 個のワイヤを有する算術回路

を持つステートメント、を受信することを含む。検証者はまた、回路のワイヤについての、暗号化されたワイヤ入力及び出力である個々のワイヤコミットメント及び／又はバッチにされたコミットメントと、算術回路内のワイヤ（好ましくは、最初のワイヤ以外のワイヤ）に対する入力又はキーオープニングと、関数回路出力（ h ）と、を受信する。検証者はまた、ステートメントで使用される楕円曲線又は各楕円曲線の仕様を受信し得る。検証者は、証明者にチャレンジ値を送信することができ、その後、オープニングを受信する。オープニングは、 Σ プロトコルに従うものとして行うことができ、検証者が、ステートメントが真であると判定して楕円曲線点を計算する、ことを可能にする回路の各ゲートについての値を含むことができる。加えて、あるいは代わりに、検証者は、証明者から証明キーを受信し得る。

10

【0072】

そして、検証者は、回路が満たされると判定し、楕円曲線点（ P ）を計算し、故に、証明者がステートメントに対する証拠（ w ）を保持していると判定する。

【0073】

これは、ステートメント回路の各ゲートについての値を証明者が知っていることを、証明が対話型である場合にはシグマプロトコルを用いて、あるいは、フィアットーシャミアヒューリスティックが使用される場合には証明キーを用いて、ゼロ知識で証明することによって達成され得る。検証者は、証明者から各ゲートに対する Σ_zero 及び Σ_prod コミットメントを受信し、チャレンジ値で応答し、証明者からオープニング値を受信し、そして、コミットメントと照合することができる。検証者は、楕円曲線点減算を介してワイヤ i に対する公開鍵を計算することによって、回路が満たされることを確認し得る。検証者は、各ワイヤについての各公開鍵が、ステートメントで指定された（1つ以上の）鍵と一致することを確認し得る。検証者は、完全にオープンにされたワイヤが、ステートメント内の指定可能な値と一致すると判定して、検証を完了することができる。

20

【0074】

コンピュータ実行可能命令を有したコンピュータ読み取り可能記憶媒体を提供することも望ましく、該コンピュータ実行可能命令は、実行されるときに、証明者によって、検証者によって、又は証明者と検証者とが協働することによって実行される方法を実行するようにプロセッサを構成する。

【0075】

インタフェース装置と、該インタフェース装置に結合された1つ以上のプロセッサと、該1つ以上のプロセッサに結合されたメモリとを有するエレクトロニクス装置を提供することも望ましく、メモリはコンピュータ実行可能命令を格納しており、該コンピュータ実行可能命令は、実行されるときに、ここに特許請求される方法を実行するように上記1つ以上のプロセッサを構成する。特許請求される方法を実行するように構成された、ブロックチェーンネットワークのノードを提供することも望ましい。そのようなノードを有するブロックチェーンネットワークを提供することも望ましい。

30

【図面の簡単な説明】

【0076】

本発明の態様が、ここに記載される実施形態を参照して解明されて明らかになる。以下、本発明の実施形態が、単なる例として、以下を含む添付の図面を参照して説明される。

40

【図1】図1（a）及び（b）は、技術的背景のセクションで対話型ゼロ知識証明に関する基本システムを上述した際に使用した図であり、図1（a）、左及び右のワイヤ入力と1つのワイヤ出力とを有する乗算ゲートの概略図であり、図1（b）は、3つのゲート、3つの入力ワイヤ、1つの出力ワイヤ、及び2つの内部ワイヤを有する算術回路の概略図である。

【図2】ハッシュ関数及び楕円曲線乗算のための算術回路を含む、ステートメントに関する複合回路の概略図である。

【図3】1つの算術回路のみが必要とされる、図1の複合ステートメントに関する算術回路の代替概略図である。

50

【図4】4つのゲートと5つのワイヤを持つ算術回路の概略図であり、ワイヤは、その公開鍵を有し、その公開鍵は、キーオープニング値でワイヤコミットメントから明らかにされる又はオープンにされる自身の公開鍵を持つ。

【図5】回路記述を有し且つ最初のワイヤが対応する公開鍵を持つステートメントSの証明のために証明者と検証者との間で交換されるデータの概略表現である。

【図6】証明者と検証者との間で交換されるデータの他の概略表現である。

【図7】図4の回路が満たされること、入力ワイヤが必要な公開鍵を持つこと、及び出力ワイヤのハッシュが必要な値を持つことを検証するために検証者によって実行されるチェックの概略図である。

【発明を実施するための形態】

【0077】

概説

本発明は、算術回路充足可能性と公開鍵の妥当性に関する従属ステートメント（キーステートメントプルーフ）の両方を同時に必要とする複合ステートメントの効率的なゼロ知識検証を可能にする。公開鍵楕円曲線仕様が、回路充足可能性を証明するために、同形コミットメント関数内で採用される。これは、効率的なやり方で回路入力及び／又は出力として使用される秘密鍵に対応する公開鍵ステートメントの証明を可能にする。

【0078】

回路充足可能性及び楕円曲線キーペアの両方を含むステートメントに関するプルーフを生成することのプルーフサイズ及び計算費用を大きく低減することができる。ここでの方法は、双線形ペアリングフレンドリー楕円曲線の使用を必要としないものである回路充足可能性に関する既存の離散対数ベースのゼロ知識プルーフプロトコルに容易に組み込むことができる。当該方法は、ビットコイン `secp256k1` 標準に完全に適合する。

【0079】

例えばビットコインブロックチェーンなどのブロックチェーン上の2者間の公正な交換トランザクションに関係する2つの用途が記述される。1つ目は、アウトソーシングされたバニティアドレスのトラストレス販売に関するゼロ知識付帯支払いを含み、これは、SHA256ハッシュプリイメージと楕円曲線（例えば、ビットコイン）秘密鍵とが等しいことのゼロ知識証明を必要とする。2つ目は、クロスチェーン・アトミック・スワップのセキュリティを向上させることを含み、これは、SHA256ハッシュプリイメージが、未知の秘密鍵（提供された公開鍵を有する）に、提供されたナンスを乗じたものに等しいことの証明を必要とする。

【0080】

全般的ソリューション

この発明は、（楕円曲線点乗算に基づく）楕円曲線公開／秘密鍵ペアとの関係を含む特定のクラスの複合ステートメントの証明を可能にする方法に関する。

【0081】

任意の暗号楕円曲線キー操作を含むステートメントを証明するために `zkSNARK` を用いることは非現実的と考えられ、従って、当該方法は、一般的な回路充足可能性に関するプルーフの構築に使用される“同形隠蔽（homomorphic hiding）”（又はコミットメントスキーム）から直接抽出される楕円曲線公開鍵に関する情報を使用する。当該方法のステートメントに関与する特定タイプの楕円曲線は、回路コミットメントスキームで使用されるものと同じである。

【0082】

しかしながら、`SNARK`法はペアリング操作を必要とし、それ故に、特別な双線形ペアリングフレンドリー楕円曲線を必要とする。このことは、一部のブロックチェーン上で使用される楕円曲線は双線形ペアリングフレンドリー楕円曲線と互換性がないので、`zk-SNARK`の使用を不可能にする。

【0083】

例として、ビットコイン公開鍵に関係するステートメントは、互換性のないものである

10

20

30

40

50

ビットコイン $secp256k1$ 曲線を使用している。

【0084】

従って、本発明の方法は、ペアリングに頼らずにいっそう少ない暗号仮定を有した、算術回路充足可能性を証明するための他のプロトコルに適合する。全体的に、本発明の方法は $zkSNARKS$ よりも効率的である。何故なら、より少ない計算しか必要とされず、トラストレス交換用途でのプルーフサイズが小さくなるからである。

【0085】

例として、後述の“ステートメント1”に関する複合回路を表すものである図2の概略図は、ハッシュ関数と楕円曲線乗算の両方のサブ回路を含む。図2において、その概略図は、秘密鍵‘ s ’並びにそれが有する対応する対を為す公開鍵‘ P ’及び値‘ h ’（これは秘密鍵 $10s$ のハッシュである値）という、3つの入力（Input）を持っている。この概略図は2つの算術回路を含んでおり、1つ目は、秘密鍵に対するハッシュ（Hash）を実行し、2つ目は、秘密鍵に対する楕円曲線乗算（EC mult）を実行する。これらの回路の出力（Out）が入力と比較される。

【0086】

なお、これら内部ゲートは単に例示目的のものである。この回路は、ハッシュの出力が楕円曲線（EC）公開鍵に等しいことを検査する。入力‘ h ’、‘ P ’、及び出力のみが検証者に対して完全に明らかにされる。そのたの値は全て暗号化される。

【0087】

ステートメント1

20

“ハッシュ関数（ H ）の出力 h と楕円曲線点 P （公開鍵）を所与として、ハッシュ s （すなわち、 $h = H(s)$ ）のプリイメージが、楕円曲線点乗数（秘密鍵、すなわち、 $P = s \times G$ 、ただし、 G は楕円曲線生成点）に等しい”。

【0088】

当該方法は、証明者がこの特定のステートメントをゼロ知識で証明することを可能にする。このような方法からの恩恵を受ける用途の例が、トラストレスでのデータ交換（例えば、アウトソーシングされたビットコインバニティアドレスの販売）、及び匿名化されたセキュアなクロスチェーン・アトミック・スワップに関して後述される。

【0089】

ステートメント1が真であることの検証は、例として、入力‘ h ’、‘ P ’、‘ s ’をとって、30ステートメントが真である場合に‘1’を出力し、それ以外の場合には‘0’を出力するものである以下の擬似コード関数を用いて決定可能である：

```
int verify(h,P,s)
{
    if(h == H(s) && P == s x G) {
        return 1
    }
    else {
        return 0
    }
}
```

40

【0090】

‘ステートメント1’を、ゼロ知識で検証すること、すなわち、 $zkSNARK$ システムで証明者が‘ s ’の値を検証者から秘密にしたままで検証することは、図2のように、ハッシュ関数と楕円曲線点乗算の両方について算術回路を必要とすることになる。

【0091】

$SHA-256$ ハッシュ関数に関する算術回路は広く使用され、最適化されて、典型的に30,000未満の乗算ゲートを含むが、文献において、暗号楕円曲線点乗算実装に関する算術回路の例は知られていない。そのような回路が知られていたとしても、それらは、そのサイズ及び複雑さのために実用的でなく、より多くのゲートを含む。

50

【0092】

当該方法は、図2に示すような、単一ハッシュ関数のための完全な算術回路で機能し、これは、キーステートメントプルーフと、ハッシュ関数用の1つの算術回路のみを使用する複合ステートメント1に関する算術回路の概略図を有している。この回路は、ハッシュの出力が正しいこと、及び公開鍵がECの暗号化された入力（キーステートメントプルーフ）に等しいことを検査する。青色でハイライトされた値、すなわち、入力‘h’、‘P’及び出力‘1’は検証者に対して明らかにされ、その他全ての値は暗号化される。

【0093】

図3の回路を用いて、証明者は、回路充足可能性を介して、秘密鍵‘s’のハッシュが‘h’になること、及び、楕円曲線生成点をGとして、鍵ペアの対応する公開鍵‘P’が‘ $s \times G$ ’に等しいことを、明示的に証明することができる。秘密鍵‘s’は、ハッシュのプリイメージ、又は関数への入力であり、ステートメントを証明するときに検証者には明らかにされない。

【0094】

言うまでもなく、‘ $s \times G$ ’が‘P’に等しいことを検証することは、プルーフプロトコルの一部としてコミットメントスキームにおいて必要な楕円曲線を使用することにより、無視できる追加の計算コストで回路プルーフから抽出されることができる。このような操作は‘キーステートメントプルーフ（key-statement proof）’と呼ばれており、‘キーオープニング（key-opening）’と呼ばれるコミットメントオープン手順を使用する。

【0095】

技術的効果

既知のzk-SNARK（Zero-knowledge Succinct Non-interactive Arguments of Knowledge）は、算術回路充足可能性に関する汎用証明システムの一実装である。SNARKフレームワークにおいて、算術回路としてエンコードされるステートメントは、多項式の集合から成る二次算術プログラム（Quadratic Arithmetic Program；QAP）と呼ばれる構成へと変換される。そして、この式の集合の妥当性を単一の点で実証することによって、ステートメントを証明することができる。SNARK法の主な利点は、検証者が数少ない楕円曲線（ペアリング）演算（数ミリ秒を要する）を実行するだけでよく、プルーフが非常に小さく（288バイト）いとともに回路サイズに依存しないということである。

【0096】

SNARK法によって達成される非常に小さいプルーフ及び検証時間は、トラストのあるセットアップ、標準的でない暗号仮定、及び証明者に課される遥かに重い計算負荷という犠牲のもとでもたらされる。SNARK法はまた、楕円曲線双線形ペアリングの使用を必要とする。しかしながら、計算的に実現可能な双線形ペアリングの使用は、特別な‘ペアリングフレンドリー’楕円曲線の使用を必要とする。これは、ビットコインsecp256k1を含む、多くの標準的な暗号楕円曲線パラメータセットの使用を排する。すると、一般的な楕円曲線点乗算を含むステートメントは、明示的な回路（非常に大きくなり得る）を使用しなければならない。

【0097】

前節で説明したSHA回路充足可能性を証明するためのΣプロトコルアプローチとの比較として、SNARK（ピノキオ）フレームワークを用いる場合、証明キーは生成するのに約10秒を要し、約7MBのサイズになり、プルーフも生成するのに約10秒を要することになる。しかしながら、プルーフサイズは288Bとなり、検証するのに要するのは約5msのみである（非特許文献8）。

【0098】

さらに、明示的な楕円曲線乗算（キーステートメント）を回路に組み込むことは、証明キーサイズ及びプルーフ生成時間の両方に、少なくとも1桁の大きさを乗じることになる。

【0099】

本発明は、一般的な算術回路充足可能性と同時での、楕円曲線公開鍵－秘密鍵関係を含

20

30

40

50

むステートメントのゼロ知識証明を可能にする。これは、算術回路充足可能性を証明することの域を超えて無視できる計算費用で達成され、証明の計算費用をかなり増加させるものである楕円曲線点乗算演算用の明示的算術回路を作成することを不要にする。

【0100】

実装

以下、本発明の実装を、バッチ式及び非バッチ式の両方のコミットメントベースゼロ知識証明システムについて説明する。

【0101】

これら例では、ゼロ知識証明プロトコルに、証明者（P）及び検証者（V）の二者が関与する。このプロトコルの目的は、証明者が、所与のステートメント（S）が真であることを、そのステートメントに対する証拠についての情報を秘密にしたまま、検証者に確信させることである。ステートメントは、 m 個のゲート及び n 個のワイヤを有する算術回路（C）と、回路ワイヤ値のうちの1つ（又はそれ以上）に対応する楕円曲線公開鍵： $p k_l$ についての従属アサーションとで構成される。ここで、下付き文字の l は、キーステートメントのワイヤインデックスである。さらに、ステートメントはまた、完全にオープンにされた（公開）ワイヤ値（すなわち、回路の公開入力／出力）についてのアサーションを含んでもよい。

【0102】

ステートメントで指定される（1つ以上の）楕円曲線公開鍵は、ターゲット楕円曲線仕様（これは、楕円曲線パラメータの全集合： $T = (p, a, b, G, n, h)$ によって規定される）に対応する。

【0103】

ビットコインスクリプトの場合、これらのパラメータは $secp256k1$ （非特許文献9）の仕様によって規定される。この使用は、ベース生成点 G を含む。ベース点を指定することに加えて、ステートメントはまた、第2の点 F も指定しなければならない（ここで、 $F = f \times G$ であり、 f は Z_p の要素である）。証明者に f の自由選択を許すことは証明者が偽のプルーフを生成することを可能にすることになり得るので、 f の値は、証明的にランダムでなければならず（例えば、ビットコインジェネシスブロックハッシュ）、あるいは、例えば π のバイナリ表現の最初の256ビットなどの“奥の手以外の何物でもない（nothing up my sleeve）”数字でなければならない。

【0104】

バッチにされたコミットメント及びバッチにされていないコミットメントを、4つのゲート及び5つのワイヤを有する代表的な算術回路である図4に関連して説明する。入力ワイヤ（ w_1 ）は、‘キーオープニング’値 k_{10} を有するワイヤコミットメント W_1 から明らかにされる又はオープンにされるその公開鍵を持つ。

【0105】

実装 — 個々のワイヤコミットメント

図4を一例として用いるに、‘キーオープニング’は、証明者によって作成されて検証者に送信される、回線内の各ワイヤに対する個々のコミットメントである。これらのキーオープニングは、算術回路充足可能性に関する既知の Σ プロトコルに従う。図5は、証明者と検証者との間で交換されるデータを示している。

【0106】

充足可能性は、以下のように、幾つかのステップを含めることによって達成される：

1. 回路の各ワイヤ i （ $i = 1, \dots, n$ ）に対して、Pedersenコミットメントを用いてコミットされる：

$$W_i = \text{Com}(w_i, r_i)$$

ここで、

$$\text{Com}(w, r) = w \times G + r \times F$$

2. それに対応する公開鍵の照明（キーステートメントプルーフ）を必要とする回路ワイヤ l に対して、証明者はまた、キーオープニング：

$$k_{o_l} = r_l \times F$$

も送信する。

3. オプションで、回路ワイヤ j が公然に明らかにされていること（完全に公開されたワイヤ）を必要とする場合、証明者は完全なオープニングタプル：

$$(w_j, r_j)$$

を送信する。

4. 次いで、 Σ プロトコルを用いてゼロ知識で、回路の各ゲートが満たされることが証明され、これは、証明者が、各ゲートについて Σ_{zero} 及び Σ_{prod} コミットメント（すなわち、それぞれ、 B 、又は C_1, C_2, C_3 ）を計算して送信し、検証者がチャレンジ値（ x ）で応答し、次いで証明者がオープニング値（ z 及び e 値）を送信し、そして、検証者がコミットメントと照合することを含む。

10

5. 回路が満たされることを検証者が確認すると、次いで、検証者が、楕円曲線点減算：

$$pk_l = Com(w_l, r_l) - k_{o_l}$$

を介してワイヤ l に関する公開鍵を計算する。

6. そして、検証者が、各 pk_l がステートメント中で指定された（1つ以上の）キーと一致すること（及び完全にオープンにされたワイヤが指定値と一致すること）を確認して、検証を完了する。

【0107】

詳細な実装 — 個々のワイヤコミットメント

図4を続けて参照して、この例の個々のコミットメント及び検証を詳述する明示的な例を提供する。これは、ワイヤのうちの1つのキーステートメントプルーフ及び他の1つのワイヤの完全な開示の両方で、単純な算術回路の充足可能性を検証することを記述するものである。

20

【0108】

図4に示す回路 C は、5つのワイヤ w_i （ $i = 1, \dots, 5$ ）と4つのゲート g_j （ $j = 1, \dots, 4$ ）とを有している。ゲート1及び3は加算ゲートであり、ゲート2及び4は乗算ゲートである。

【0109】

証明者及び検証者は、楕円曲線及びコミットメント仕様と共に、回路、ワイヤ5の値、及びワイヤ1の公開鍵を含むステートメントに同意する。それが真であることを証明者が検証者に対して証明することを望むステートメント（ S ）は：

30

“私は、回路 C への充足割り当て（すなわち、全てのゲートを満たすワイヤ値 $\{w_i\}_{i=1}^5$ ）を知っており、ただし、ワイヤ1は公開鍵 P （すなわち、 $P = w_1 \times G$ ）を持っており、ワイヤ5は値 h （すなわち、 $w_5 = h$ ）を持っている”

である。

【0110】

ワイヤ1から4の値は明らかにされない。そして、図6に示すように、また以下に説明するように、証明者及び検証者がインタラクトする：

1. 証明者が、5つのランダムなブラインディング値（ $r_1, \dots, r_5$ ）を生成し、次いで、5つのワイヤコミットメント（ $W_1, \dots, W_5$ ）を計算し、それらを検証者に送信する。

40

2. 証明者が、ワイヤ1に対するキーオープニング： $k_{o_1} = r_1 \times F$ を計算し、それを検証者に送信する。

3. 証明者が、ワイヤ5に関する完全なオープニング情報（ w_5, r_5 ）を検証者に送信する。

4. 加算ゲート（ g_1 及び g_3 ）について、証明者が、（ランダムなナンス r_{B1} 及び r_{B3} を用いて）ゼロに対するコミットメント： $B_1 = Com(0, r_{B1})$ 及び $B_3 = Com(0, r_{B3})$ を生成し、それらを検証者に送信する。

5. 乗算ゲート（ g_2 及び g_4 ）について、証明者が、次のようにコミットメントを生成し、すなわち：

50

ゲート 2 について：

$$\begin{aligned} C_{12} &= \text{Com}(t_{12}, t_{32}) \\ C_2 &= \text{Com}(t_{22}, t_{52}) \quad \text{及び} \\ C_3 &= t_{12} \times W_1 + t_{42} \times F \end{aligned}$$

ゲート 4 について：

$$\begin{aligned} C_{14} &= \text{Com}(t_{14}, t_{34}) \\ C_2 &= \text{Com}(t_{24}, t_{54}) \quad \text{及び} \\ C_3 &= t_{14} \times W_3 + t_{44} \times F \end{aligned}$$

のようにコミットメントを生成し、ここで、 t_{xx} 値は、ランダムなブラインディングナンスである。証明者は、これらのコミットメントを検証者に送信する。

10

6. 次いで、検証者が、ランダムチャレンジ値 x を生成し、それを証明者に送信する。あるいは、検証者は、フィアットーシャミアヒューリスティックを用いて、全てのコミットメントの連結をハッシュすることによって、値 x を生成してもよい。

7. 加算ゲート (g_1 及び g_3) について、証明者が、以下のオープニング：

$$\begin{aligned} z_1 &= x(r_1 + r_1 - r_2) + r_{B1} \\ z_3 &= x(r_2 + r_1 - r_4) + r_{B3} \end{aligned}$$

を計算し、それらを検証者に送信する。

8. 乗算ゲート (g_2 及び g_4) について、証明者が、以下のオープニング：

$$\begin{aligned} e_{12} &= w_1 x + t_{12} \\ e_{22} &= w_2 x + t_{22} \\ z_{12} &= r_1 x + t_{32} \\ z_{22} &= r_2 x + t_{52} \\ z_{32} &= (r_3 - w_1 r_2) x + t_{42} \\ e_{14} &= w_3 x + t_{14} \\ e_{24} &= w_4 x + t_{24} \\ z_{14} &= r_3 x + t_{34} \\ z_{24} &= r_4 x + t_{54} \\ z_{34} &= (r_5 - w_3 r_4) x + t_{44} \end{aligned}$$

20

を計算し、それらを検証者に送信する。

9. 最後に、検証者が、等価性を検査する。これらが合格すると、証明が検証される。

30

【0111】

検証者によって実行される検証は、図 7 にまとめられており、内側のボックス内の検査が、回路が満たされること、及び、第 1 のワイヤが必要な公開鍵を有し、第 5 のワイヤが必要な値を有することを検証する。

【0112】

図 5 及び図 6 のチャレンジ ' x ' は、対話的な証明を提供し、証明者と検証者との間で通信が往復する。

【0113】

売り手と買い手が同時には対応できない、あるいはオンラインでないことがあり得るので、このインタラクションは、ゼロ知識付帯支払い (zero-knowledge contingent payment; ZKCP) が行われるときに不便であり得る。また、買い手 (検証者) が、プルーフが公開的に検証可能であることを望むことがあり、例えば、それがデジタルグッズの広告の一部であることがある。

40

【0114】

さらに、証明が厳密にゼロ知識であるのは、完全特別正直検証者 (perfect special-honest verifier) モデルにおいてのみであり、すなわち、検証者がチャレンジとして真の乱数を生成するのであって、証拠についての情報を試して取り出すためのチャレンジ値を選択したりはしないと仮定されるときのみである。

【0115】

これらの問題を解決するために、フィアットーシャミアヒューリスティックが適用され

50

、これは、ランダムチャレンジ値‘x’を、証明者によって作成されたコミットメントのハッシュの出力で置き換える。ランダムオラクルモデル（暗号ハッシュ関数の出力が真にランダムとみなされる）では、証明者は不正を働くことができず、検証者は生成されたチャレンジ値を検査することができる。

【0116】

従って、この例は、フィアットーシャミアヒューリスティックを使用して対話型証明システムを非対話型のものに変換することによって改善することができ、また、証明者は、オフラインで独立して公開的に検証されることができるプルーフを生成することができる。

【0117】

より具体的には、チャレンジ値（x）が、証明者によって生成されたコミットメントの全て（すなわち、ワイヤコミットメントの全てと、和ゲート及び積ゲートそれぞれに対するB及びC₁、C₂、C₃コミットメントの全て）の連結を（例えば、SHA-256で）ハッシュすることによって計算される値で置き換えられる。

【0118】

実装 — バッチ化ベクトルコミットメント

ベクトルコミットメントのバッチ化を伴う回路充足可能性に関する圧縮証明システム（非特許文献8、非特許文献6）は、以下に記載される方法を使用し、当該方法は、バッチにされた回路ワイヤコミットメントからキーステートメントプルーフを抽出することを可能にする。

【0119】

繰り返しを避けるため、完全なプロセスを記載することはせず、以下のステップは、バッチにされたワイヤコミットメントの生成と、それが指定された公開鍵を含むことを例証することとに焦点を当てている。以下のステップでは、ワイヤ₁が、キーオープニングを与えられるものであり、n個のワイヤがベクトルコミットメント内で共にバッチにされるとして、以下のように、バッチにされたコミットメントが生成される。

1. 証明者が、n-1個の乱数 $x_1, \dots, x_{n-1} \leftarrow \mathbb{Z}_p$ を生成する。
2. 証明者が、楕円曲線点 $K_i = x_i \times G$ （for $i = 1, \dots, n-1$ ）を計算する。これらの値に $K_n = G$ を足したものが、検証者に送られる証明キーPrKを形成する。
3. 証明者が、ランダム値： $r \leftarrow \mathbb{Z}_p$ を生成する。
4. w_n がキーオープンされるものであるとして、証明者が、ワイヤ値 w_i （for $i = 1, \dots, n$ ）のベクトルwへのコミットメント：

【数5】

$$Com(w) = r \times F + \sum_{i=1}^{n-1} w_i \times K_i + w_n \times G$$

を計算し、それを検証者に送信する。

5. 証明者がまた、ベクトルコミットのためのキーオープニング：

【数6】

$$ko_n = r \times F + \sum_{i=1}^{n-1} w_i \times K_i$$

を送信する。

6. 検証者が、楕円曲線演算：

【数7】

$$pk_n = Com(w) - ko_n$$

を介して、キーステートメントワイヤの公開鍵オープニングを計算する。

【0120】

発明概要

ハッシュブライメージ及び楕円曲線秘密鍵の等価性の証明は、数多くの用途で利用されることができる。以下、利用のためのキーステートメントゼロ知識プルーフの具体例の構築を概説するものである2つの用途について説明する。

【0121】

以下のステートメントSは、適用例の目的で、上述のステートメント1のより具体的なバージョンである：

S：

“公開出力hを有するSHA-256ハッシュ関数(H)及びsecp256k1楕円曲線上の公開点Pが所与であり、ハッシュの秘密のブライメージs（すなわち、 $h = H(s)$ ）が、楕円曲線点乗数（すなわち、対応する秘密鍵、すなわち、 $P = s \times G$ ）に等しい”

10

【0122】

提供される例において、このステートメントは、入力ワイヤ(w_1)が公開点Pの秘密鍵である、及び出力ワイヤ(w_n)がhに等しい、というアサーションとともに、SHA-256ハッシュ関数C_{SHA256}(n個のワイヤ w_i ($i = 1, \dots, n$)及びm個のゲートを有する)に関する単一の算術回路で構成され、すなわち、

【数8】

$S: C_{SHA256}$ はワイヤ $\{w_i\}_{i=1}^n$ によって満たされる $AND\ w_1 \times G = P\ AND\ w_n$

20

で構成される。

【0123】

従って、このステートメントを完全に検証するために、証明者は、検証者に対して、secp256k1ベースのコミットメントスキームを使用して、SHA256回路への充足割り当てを知っていることを実証し、そして、ワイヤ1についてのキーオープニング(k_{o1})及びワイヤnについての完全オープニング(w_n, r_n)を単純に提供しなければならない。検証者は、入力ワイヤ(w_1)の値を教わず、すなわち、完全にオープンにされる出力ワイヤ w_n を除く他のワイヤのいずれの値も教わらない。

30

【0124】

用途I

上の実装セクションで説明した本発明の例は、支払い又はリソースへのアクセスのために交換されるデータを表すものである、アウトソーシングされたビットコインバニティアドレスに関するZKCPに適用されることができる。

【0125】

ビットコインアドレスは、公開、コピー及び転記するのが容易なものとするために、人間読み取り可能な英数字フォーマット(Base58エンコーディング)でエンコードされる。このフォーマットの使用は、所謂バニティアドレスの人気につながっており、例えば下に示すもの：

40

[外1]

1nChainRHtKNngkdXEeobR76b53LEtTpyT

version byte vanity pattern

などの、所望の(名前のような)文字列を含むアドレスを生じる秘密鍵を見つけるために、鍵空間がブルートフォース(総当たり)探索される。

【0126】

有意なパターンを有するバニティアドレスを導出することは計算的に高くつくため(例

50

例えば、上に示したアドレスは、一致が見つかるまでにおよそ 10^{13} 個の異なる公開鍵の生成を必要とした)、検索をアウトソーシングすることが一般的であり、バニティアドレスが委託販売されているオンラインマーケットプレイスが幾つか存在している。これは、楕円曲線点乗算の同形性を用いて安全に行われることができる(非特許文献7)。

【0127】

生成をアウトソーシングすることは安全であるが、バニティアドレスの販売は信用できない。売り手が支払いを受ける前に買い手が必要値を得たり、売り手が必要値を渡す前に支払いを受けたり、あるいは両者ともが第三者預託サービスを信頼しなければならなかったりする。本発明は、ZKCPを介したバニティアドレスのトラストレス販売を可能にするために使用されることができる。以下にて、買い手／検証者と売り手／証明者との間でとられるステップを説明する。

1. 買い手と売り手が、必要なバニティパターン(Str)及び価格(aビットコイン)について合意し、セキュアである必要のない通信チャネルを確立する。

2. 買い手が、セキュアなランダム秘密鍵 $s_k B$ 及び対応する楕円曲線公開鍵を生成し、公開鍵 $p_k B = s_k B \times G$ である。

3. 買い手が売り手に $p_k B$ を送信する。

4. 次いで、売り手が、 i を変化させることによって $p_k = p_k B + i \times G$ から導出される Base58 エンコードされたアドレスにおいて必要なパターンの探索を実行する。

5. 必要なパターンを持つアドレスが見つかった場合、売り手は i を保存し、買い手にシグナリングして、 $p_k s = i \times G$ 及び SHA256 ハッシュ $H(i)$ を送信する。

6. 売り手はまた、上の例で説明したように、 $H(i)$ に対するプリイメージが、 $p_k s$ に対応する秘密鍵であることのプルーフを買い手に提供する。

7. 買い手が、プルーフを検証し、また、 $p_k = p_k B + p_k s$ に対応するアドレスが合意したパターンに一致することを確認する。この時点で(プルーフによって)、買い手は、値 i を教わることで、自身がバニティアドレスに関する完全な秘密鍵($s_k B + i$)を導出することが可能になること、及び特定の値 i が $h = H(i)$ にハッシュすることを知る。

8. 次いで、買い手が、合意した手数料(a)を含むアウトプットを含んだ、ハッシュ時間ロック契約(HTLC)トランザクション Tx_1 を構築する。この出力は、2通りのやり方でアンロックされることができ、すなわち：

i. 任意の時に、売り手からの署名と、ハッシュプリイメージ i とを用いて、

ii. 例えば、指定された時間又はブロック高さまで出力が消費されることを防ぐために使用され得るものである CHECKLOCKTIMEVERIFY (OP_CLTV) スクリプト op コードを使用して、指定された時間の後に買い手からの署名を用いて、アンロックされることができる。

9. 次いで、買い手が、このトランザクションに署名してブロックチェーンにブロードキャストし、それがブロックへとマイニングされる。

10. 確認されると、売り手は、自身の署名とハッシュロックを解除するための値 i を供給するトランザクション Tx_2 を提供することによって、 Tx_1 の出力における手数料を請求することができ、そして、値 i はブロックチェーン上で明らかにされる。

11. 買い手は、最終的なバニティアドレス秘密鍵 $s_k = s_k B + i$ を計算することができ、 $p_k = s_k \times G$ である。

12. 買い手が指定の OP_CLTV 時間の前に値 i を供給しない場合、売り手は(非協力的な買い手によって手数料が失われることを防ぐために)署名を提供して手数料を再請求することができる。

【0128】

すると、トランザクションは、完全にアトミックでトラストレスであり、買い手は有効な値 i を提供した場合にのみ支払いを受け、値 i はブロックチェーン上で公然に明かされる。秘密鍵の分割により、この値は、他の誰にも役に立たず、完全な秘密鍵のセキュリティを損なわない。

【0129】

用途II

上の実装セクションで説明した本発明の例は、各々がそれぞれの、異なるブロックチェーン上に記録された交換されるべきデータを持つ二者間での私的なデータ交換に適用されることができる。

【0130】

より具体的には、本発明は、アトミックトレードとしても知られた、ブロックチェーントランザクション機構を活用するトラストレスでの公正な交換プロトコルであるプライバシーを保護するクロスチェーン・アトミック・スワップに適用されることができる。このプロトコルは、第三者集中型の交換を用いずに、2つの異なるブロックチェーン上で2つの異なる暗号通貨トークンを取引するために使用される。この文脈での単語‘アトミック’は、公正な交換の性質を指し、両者がトランザクションを完了するか、あるいは両者とも完了しないかのいずれかである。

【0131】

既知の基本プロトコルの一例は、以下のステップに従って実行される。安全であるために、スワップで使用される暗号通貨の両方が、ハッシュされ且つ時間ロックされた契約を可能にするスクリプト機能を持たなければならない。このスワップには、アリスとボブの二者が関与する。この例では、アリスが1ビットコインを保有しており、それをボブの100ライトコインと取引することに合意している。

1. アリスが、ボブに送信するライトコイン公開鍵 P_A を生成する。
2. ボブが、アリスに送信するビットコイン公開鍵 P_B を生成する。
3. アリスが、セキュアな乱数 x を生成する。
4. アリスが、 x の $SHA-256$ ハッシュ: $h = H(x)$ を計算する。
5. アリスが、以下であるビットコイントランザクション T_{xA} 、すなわち、
 - i. 有効な署名 AND h にハッシュする値を用いて、1ビットコインを P_B に支払う、
 - ii. OR 24時間後に1ビットコインをアリスに払い戻す、ものであるビットコイントランザクション T_{xA} を作成する。
6. アリスが、該トランザクションをビットコインネットワークにブロードキャストする。
7. ボブが、ビットコインブロックチェーン上で T_{xA} が確認されたことを観測すると、以下であるライトコイントランザクション T_{xB} 、すなわち、
 - i. 有効な署名 AND h にハッシュする値を用いて、100ライトコインを P_A に支払う、
 - ii. OR 24時間後に100ライトコインをボブに払い戻す、ものであるビットコイントランザクション T_{xB} を作成する。
8. ボブが、該トランザクションをライトコインネットワークにブロードキャストする。
9. トランザクションが確認されると、アリスは、自身の署名及び値 x を提供することによって、ライトコイン出力を請求することができる。
10. ボブは、ライトコインブロックチェーン上で値 x を観測すると、自身の署名及び値 x を提供することによって、ビットコイン出力を請求することができる。

【0132】

この例は、両者ともコインを手に入れる又は両者とも手に入れないのいずれかであることを確実にする。アリスはハッシュ値を生成し、彼女のみがプリイメージを知っているが、彼女は、コインを請求するためにこのプリイメージを明らかにすることを要求され、そして、それが、ボブが彼のコインを請求することを可能にする。どちらかの者が完了に向けてプロトコルに従わない場合、それらの両者がロックアウト期間後に自身のコインを再請求することができる。

【0133】

上述した既知のプロトコルの1つの重大な欠点は、両方のブロックチェーン上のトラン

ザクションが自明にリンク付け可能なことであり、いったん確認されると、固有値 x が、永久に、両方のブロックチェーン上で公然と見えてしまう。これは、コインの交換性及びトランザクションのプライバシーの両方に影響を及ぼす。

【0134】

2つのトランザクションをリンク付けないためには、各チェーン上の出力に異なるキーが使用されなければならないが、プロトコルがセキュア且つトラストレスであるために、ボブは、アリスによって、彼女が彼女のハッシュプリメージを明らかにするときに彼が彼のコインをアンロックするのに必要な情報を教わることになる、というプルーフを与えられなければならない。

【0135】

上の例で説明したキーステートメントプルーフを採用することにより、第2のブロックチェーン上のハッシュロックされた出力を、通常の $pay-to-public-key-hash$ (P2PKH) 出力に変換することができ、トランザクションの性質を隠すとともに如何なる可能なリンクも破ることができる。

【0136】

アリスが1ビットコインを保有しており、それをボブの100ライトコインと取引することに合意しているという上の例に適用して、改良プロセスは、以下のアクションを含むことになる：

2. アリスが、ボブに送信するライトコイン公開鍵 P_A を (秘密鍵 s_A で) を生成する。
3. ボブが、アリスに送信するビットコイン公開鍵 P_B を (秘密鍵 s_B で) 生成する。
4. アリスが、セキュアな乱数 $x \leftarrow \mathbb{Z}_p$ を生成する。
5. アリスが、 x の $SHA-256$ ハッシュ： $h = H(x)$ と、 x に対応する楕円曲線公開鍵： $P_x = x \times G$ とを計算する。
6. アリスが、 h 及び P_x の両方をボブにセキュアに送信する。
7. アリスはまた、 h のプリイメージが P_x を生成した秘密鍵に等しいことのキーステートメントプルーフをボブに送信する。
8. アリスが、以下であるビットコイントランザクション T_{xA} 、すなわち、
 - i. 1ビットコインを公開鍵 $P_C = P_B + P_x$ に支払う、
 - ii. OR 24時間後に1ビットコインをアリスに払い戻す、

ものであるビットコイントランザクション T_{xA} を作成する。

9. アリスが、該トランザクションをビットコインネットワークにブロードキャストする。

10. ボブが、ビットコインブロックチェーン上で T_{xA} が確認されたことを観測すると、以下であるライトコイントランザクション T_{xB} 、すなわち、

- i. 有効な署名 AND h に $SHA-256$ ハッシュする値を用いて、100ライトコインを P_A に支払う、
- ii. OR 24時間後に100ライトコインをボブに払い戻す、

ものであるビットコイントランザクション T_{xB} を作成する。

11. ボブが、該トランザクションをライトコインネットワークにブロードキャストする。

12. トランザクションが確認されると、アリスは、自身の署名及び値 x を提供することによって、ライトコイン出力を請求することができる。

13. ボブは、ライトコインブロックチェーン上で値 x を観測すると、楕円曲線点乗算の同形性から $s_B + x$ である P_C の秘密鍵を用いて署名を提供することによってビットコイン出力を要求することができる。

【0137】

用途全般

本発明は、証明者が、検証者に対して、ステートメントが真であることを、該ステートメントに対する証拠 (w) を秘密にしたまま証明するものである、ステートメント (S) のゼロ知識証明又は検証に適する。秘密は、例えばハッシュ関数などの関数によって処理

10

20

30

40

50

されることができるが、例えば公開鍵に関するステートメントの妥当性といった、暗号楕円曲線キー操作を更に含むことができる。上の例において、本発明の方法は、バニティアドレスに関するトラストレスでのZKCPを可能にするために使用されている。これはまた、例えば、パスワードの導出、例えばパスポート又は身元証明書などの有効な機械読み取り可能文書の検証、又は他のそのような機密トランザクションにも適用されることができる。

【0138】

なお、上述の実施形態は、本発明を限定するものではなく、例示するものであり、当業者は、添付の請求項によって規定される本発明の範囲から逸脱することなく、数多くの代替の実施形態を設計することができる。

10

【0139】

請求項において、括弧内に置かれた如何なる符号も、請求項を限定するものと解釈されるべきでない。用語“有している”及び“有する”並びにこれらに類するものは、いずれかの請求項又は明細書全体に列挙されたもの以外の要素又はステップの存在を除外するものではない。本明細書において、“有する”は“含む又はからなる”を意味し、“有している”は“含んでいる又はからなっている”を意味する。

【0140】

単数形での要素の参照は、それらの要素の複数形での参照を除外するものではなく、その逆もまた然りである。本発明は、幾つかの別個の要素を有するハードウェアによって、また、適切にプログラムされたコンピュータによって実装され得る。

20

【0141】

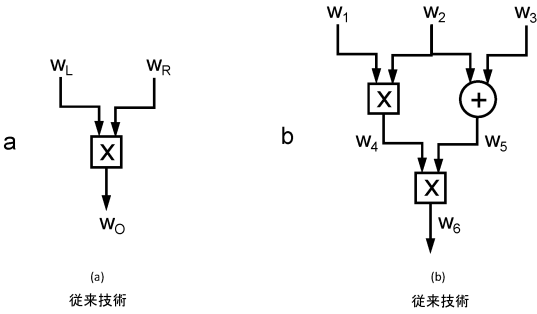
幾つかの手段を列挙するデバイスクレームにおいて、それらの手段のうちの幾つかが、同一のハードウェアアイテムによって具現化されてもよい。特定の複数の手段が相互に異なる従属請求項に記載されているという単なる事実は、それらの手段の組み合わせが有利に使用され得ないということを指し示すものではない。

30

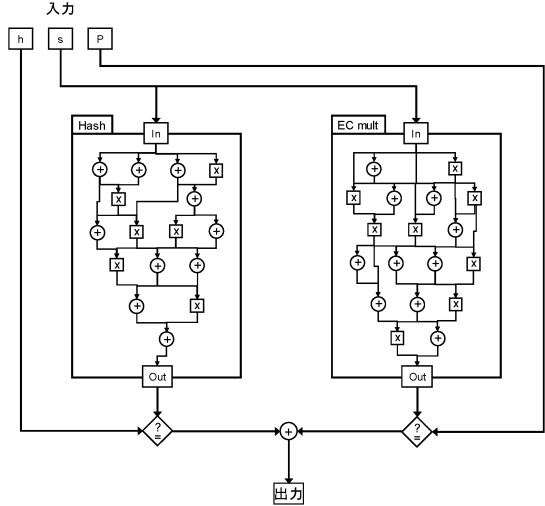
40

50

【図面】
【図 1】

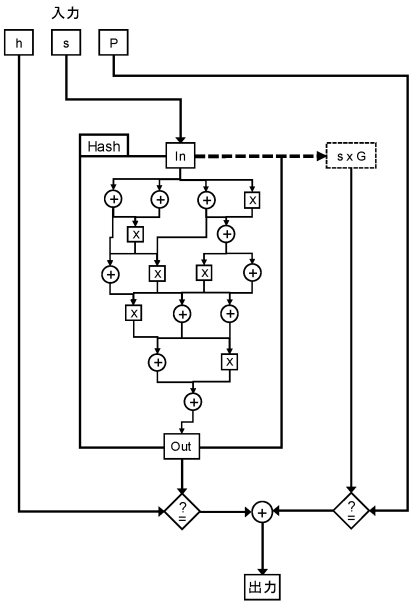


【図 2】

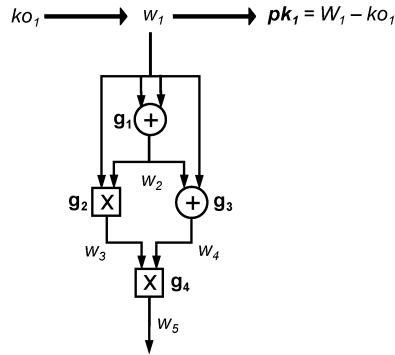


10

【図 3】



【図 4】



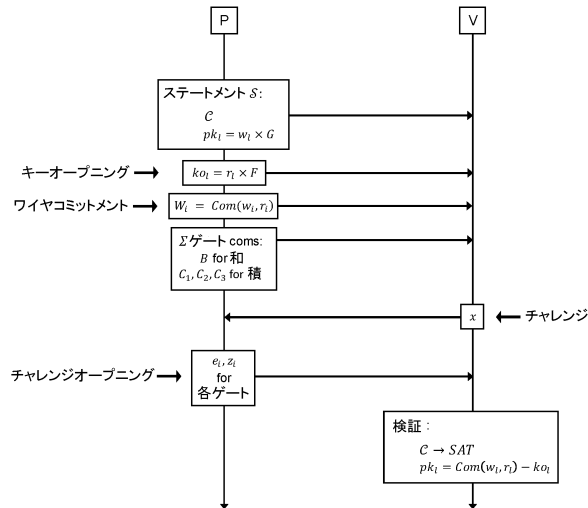
20

30

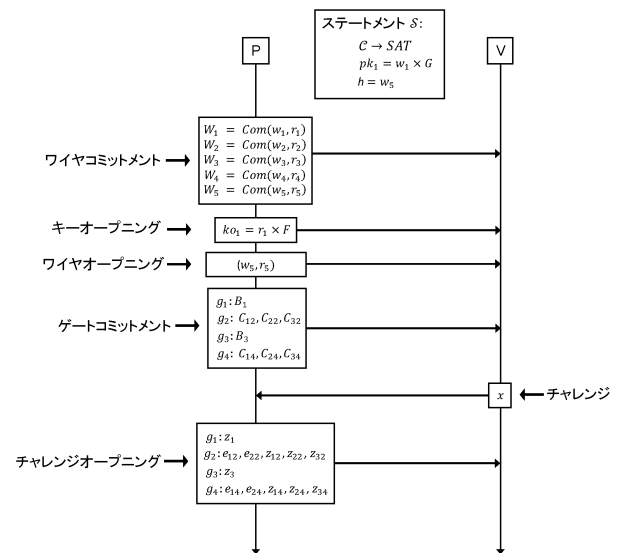
40

50

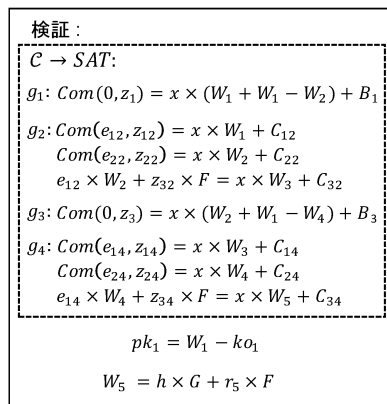
【図 5】



【図 6】



【図 7】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

英国(GB)

(31)優先権主張番号 1804740.7

(32)優先日 平成30年3月23日(2018.3.23)

(33)優先権主張国・地域又は機関

英国(GB)

内

審査官 中里 裕正

(56)参考文献 DANEZIS, G. et al., Pinocchio Coin: Building Zerocoin from a Succinct Pairing-based Proof System, PETShop'13: Proceedings of the First ACM workshop on Language support for privacy-enhancing technology, 2013年11月, pp.27-30

BEN-SASSON, E. et al., Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs, 2015 IEEE Symposium on Security and Privacy, 2015年05月, pp.287-304

KLUCZNIAK, K. and AU, M. H., Fine-Tuning Decentralized Anonymous Payment Systems based on Arguments for Arithmetic Circuit Satisfiability, Cryptology ePrint Archive, [online], 2018年02月14日, pp.1-51, <URL:https://eprint.iacr.org/2018/176>

(58)調査した分野 (Int.Cl., DB名)

H04L 9/32

JSTPlus/JMEDPlus/JST7580(JDreamII)

IEEE Xplore