(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7620564号
(P7620564)

(45)発行日　令和7年1月23日(2025.1.23)

(24)登録日　令和7年1月15日(2025.1.15)

(51)国際特許分類		FI		
H04L 9/08 (2006.01)		H04L	9/08	C
H04L 9/32 (2006.01)		H04L	9/08	E
		H04L	9/32	200Z

請求項の数　12　（全29頁）

(21)出願番号	特願2021-559820(P2021-559820)
(86)(22)出願日	令和2年4月3日(2020.4.3)
(65)公表番号	特表2022-528925(P2022-528925A)
(43)公表日	令和4年6月16日(2022.6.16)
(86)国際出願番号	PCT/IB2020/053218
(87)国際公開番号	WO2020/212796
(87)国際公開日	令和2年10月22日(2020.10.22)
審査請求日	令和5年3月6日(2023.3.6)
(31)優先権主張番号	1905348.7
(32)優先日	平成31年4月16日(2019.4.16)
(33)優先権主張国・地域又は機関	英国(GB)

(73)特許権者　318001991
エヌチェーン ライセンシング アーゲー
スイス・６３００・ツーク・グラーフェナウヴェーク・６
(74)代理人　100107766
弁理士　伊東 忠重
(74)代理人　100070150
弁理士　伊東 忠彦
(74)代理人　100135079
弁理士　宮崎 修
(72)発明者　ライト，クレイグ スティーヴン
イギリス国 シーエフ１０ ２エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス ７ス フロア アーカート－ダイクス アンド ロード エルエルピー

最終頁に続く

(54)【発明の名称】　データを暗号化するためのコンピュータにより実施される方法及びシステム

(57)【特許請求の範囲】

【請求項１】

データを暗号化する方法であって、第１参加者に関連付けられたコンピューティング装置により実行され、

前記第１参加者及び第２参加者に共通の共通シークレットに基づき、データの暗号化のための暗号鍵を決定するステップと、

暗号化データを提供するために、前記暗号鍵に基づき前記データを暗号化するステップであって、前記データを暗号化するステップは、少なくとも１つのXOR演算を含む、ステップと、

前記暗号化データをブロックチェーントランザクションに組み込むステップであって、前記共通シークレットを用いて、前記暗号化データからセッション鍵及び／又はパスワードが導出できる、ステップと、

を含み、

前記ブロックチェーントランザクションは、前記共通シークレットに基づくデータに一方向関数を適用することに基づき、データを含むスクリプトを用いて償還される、方法。

【請求項２】

以下の特徴：

（ｉ）前記暗号化データは、前記ブロックチェーントランザクションのアウトプットに含まれる、

（ｉｉ）前記暗号化データは前記ブロックチェーントランザクションの中の公衆アクセ

ス可能なデータに含まれる、
のうちの１つ以上を含む請求項１に記載の方法。
【請求項３】
前記一方向関数は、ハッシュ関数である、請求項１に記載の方法。
【請求項４】
以下の特徴：
（ｉ）前記ブロックチェーントランザクションの償還は前記データを公開する、
（ｉｉ）前記ブロックチェーントランザクションの償還は、前記ブロックチェーントランザクションを償還する参加者を識別させる、
（ｉｉｉ）暗号化データを前記第２参加者へ送信した結果として、前記第１参加者及び第３参加者に共通の第２共通シークレットに基づく暗号鍵を用いて、前記データを暗号化させるステップを更に含む、
（ｉｖ）データが前記第２参加者から第３参加者へ送信された結果として、前記第１参加者へ、前記データを送信させるステップを更に含む、又は、
（ｖ）前記共通シークレットに基づきデータに一方向関数を繰り返し適用することにより、複数の前記暗号鍵を決定するステップを更に含む、
のうちの１つ以上を含む請求項１～３のいずれか一項に記載の方法。
【請求項５】
前記方法が、前記共通シークレットに基づきデータに一方向関数を繰り返し適用することにより、複数の前記暗号鍵を決定するステップを含む場合、
前記一方向関数は、ハッシュ関数である、及び／又は、
第１データを暗号化するために使用される第１暗号鍵は、前記第１暗号鍵に基づきデータに前記一方向関数の適用を適用することにより前記第１データを暗号化した後に、第２データを暗号化するために使用される第２暗号鍵に関連する、請求項４に記載の方法。
【請求項６】
前記暗号化は、前記データの部分に少なくとも１つの暗号鍵を適用することにより実行される、及び／又は、
少なくとも１つの前記暗号鍵は行列である、請求項１～５のいずれか一項に記載の方法。
【請求項７】
少なくとも１つの前記暗号鍵が行列である場合、
前記行列の要素は、前記共通シークレットに基づきデータに一方向関数を繰り返し適用することにより生成される、及び／又は、
少なくとも１つの前記行列は、線形独立の列及び行を有する２つの行列の積から導出される、請求項６に記載の方法。
【請求項８】
前記暗号化は、前記データの少なくとも１つの部分に少なくとも１つの暗号鍵を繰り返し適用することにより実行される、及び／又は、
前記第１参加者において、前記共通シークレットを決定するステップであって、前記第１参加者は、第１秘密鍵と第１公開鍵とを有する暗号システムの第１公開－秘密鍵ペアに関連付けられ、前記第２参加者は、第２秘密鍵と第２公開鍵とを有する前記暗号システムの第２公開－秘密鍵ペアに関連付けられ、前記共通シークレットは、前記第１参加者において前記第１秘密鍵及び前記第２公開鍵に基づき決定され、前記暗号システムの特性は、前記共通シークレットが前記第２参加者において前記第２秘密鍵及び前記第１公開鍵に基づき決定できることである、ステップを更に含む、請求項１～７のいずれか一項に記載の方法。
【請求項９】
前記暗号システムは、楕円曲線暗号システムである、請求項８に記載の方法。
【請求項１０】
前記第１参加者は前記共通シークレットのシェアを有し、前記共通シークレットは、閾数の前記シェアによりアクセス可能であり、前記閾数より少ないシェアでアクセス不可能

である、請求項１～９のいずれか一項に記載の方法。
【請求項１１】
コンピュータにより実装されるシステムであって、
プロセッサと、
前記プロセッサによる実行の結果として、前記システムに請求項１～１０のいずれかに記載の方法を実行させる実行可能命令を含むメモリと、
を含むシステム。
【請求項１２】
実行可能命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記実行可能命令は、コンピュータシステムのプロセッサにより実行された結果として、前記コンピュータシステムに、請求項１～１０のいずれかに記載の方法を実行させる、非一時的コンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、データを暗号化する、コンピュータにより実施される方法及びシステムに関し、特に排他的ではなく、ブロックチェーンで使用するためのそのような方法に関する。本開示は、メッセージに認証データを適用する方法にも関する。
【背景技術】
【０００２】
本願明細書では、私たちは、全ての形式の電子的な、コンピュータに基づく、分散型台帳を包含するために用語「ブロックチェーン」を使用する。これらは、総意に基づくブロックチェーン及びトランザクションチェーン技術、許可及び未許可台帳、共有台帳、並びにこれらの変形を含む。他のブロックチェーン実装が提案され開発されているが、ブロックチェーン技術の最も広く知られているアプリケーションは、Bitcoin台帳である。Ｂｉｔｃｏｉｎは、ここでは、便宜上及び説明の目的で参照されることがあるが、本開示はＢｉｔｃｏｉｎブロックチェーンと共に使用することに限定されず、代替のブロックチェーン実装及びプロトコルが本開示の範囲に包含されることに留意すべきである。用語「ユーザ」は、ここでは、人間またはプロセッサに基づくリソースを表してよい。
【０００３】
ブロックチェーンは、コンピュータに基づく非集中型の分散型システムとして実装されるピアツーピアの電子台帳であり、ブロックにより構成され、ブロックはまたトランザクションにより構成される。各トランザクションは、ブロックチェーンシステムの中の参加者間でデジタルアセットの制御の移転を符号化するデータ構造であり、少なくとも１つのインプット及び少なくとも１つのアウトプットを含む。各ブロックは前のブロックのハッシュを含み、これらのブロックは一緒に繋げられて、起源以来ブロックチェーンに書き込まれている全てのトランザクションの永久的な変更不可能な記録を生成する。トランザクションは、スクリプトとして知られている小さなプログラムを含む。スクリプトは、それらのインプット及びアウトプットを埋め込まれ、トランザクションのアウトプットがどのように及び誰によりアクセス可能であるかを指定する。Bitcoinプラットフォームでは、これらのスクリプトはスタックに基づくスクリプト言語を用いて記述される。
【０００４】
トランザクションがブロックチェーンに書き込まれるためには、「検証」されなければならない。ネットワークノード（マイナー）は、無効なトランザクションがネットワークから拒否され、各トランザクションが有効であることを保証するために作業を実行する。ノードにインストールされたソフトウェアクライアントは、未使用トランザクション（unspent transaction, UTXO）のロック及びアンロックスクリプトを実行することにより、UTXOに対してこの検証作業を実行する。ロック及びアンロックスクリプトの実行が真（TRUE）と評価する場合、トランザクションは有効であり、トランザクションはブロックチェーンに書き込まれる。従って、トランザクションがブロックチェーンに書き込まれ

るためには、（ｉ）トランザクションを受信した第１ノードにより検証され、トランザクションが有効な場合には、ノードが該トランザクションをネットワーク内の他のノードに中継する、（ｉｉ）マイナーにより構築された新しいブロックに追加される、（ｉｉｉ）マイニングされる、つまり過去のトランザクションの公開台帳に追加される、ことが必要である。

【０００５】

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

【０００６】

パーティ間の通信が非セキュアネットワーク上で生じるシステムでは、重要な関心事は、共通された秘密鍵の確率である。この問題を解決するために、セキュアな鍵配信プロトコルが開発されており、以下を含む：Diffie－Helman（DH)対称鍵交換[Merkle, Ralph C. (April １９７８)、"Secure Communications Over Insecure Channels"、Communications of the ACM. ２１(４): ２９４－２９９；Diffie, Whitfield; Hellman, Martin E. (November １９７６)、"New Directions in Cryptography"、IEEE Transactions on Information Theory. ２２（６): ６４４－６５４]、及び３パスプロトコル（three－pass protocol）[Menezes, A.; van Oorschot, P.; Vanstone, S.（１９９６)、“Handbook of Applied Cryptography”、CRC Press. p.５００]。これらの方法はセキュアな暗号化を達成するが、大規模な鍵セットを生成し共有するには、又は連続的に鍵を生成するには、計算上高価である。

【０００７】

国際特許出願公開番号WO２０１７/１４５０１６は、公開鍵、決定性（deterministic）鍵、及び楕円曲線代数学の特性のみを用いて一連の共有シークレット鍵を確立する方法を開示している。共有シークレット鍵を公衆ネットワークを介して送信する必要がないので、それは中間者攻撃（man－in－the－middle attack）を受けない。この共有シークレット鍵から、この研究で提案されたように楕円曲線演算又は鍵生成拡張を通じて任意の数の追加の共有秘密鍵を導出することは、簡単である。留意すべきことに、このプロトコルは、鍵を連結又はトランケートして所望の長さにできるので、鍵の長さに依存しない。

【０００８】

同じレベルのセキュリティを維持しながら、国際特許出願番号WO２０１７/１４５０１６で開示されたプロトコルの更に計算効率のよい拡張を提供することが望ましい。

【発明の概要】

【０００９】

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

【００１０】

データを暗号化する方法であって、

第１参加者により、前記第１参加者及び第２参加者に共通の共通シークレットに基づき、データの暗号化のための暗号鍵を決定するステップと、

暗号化データを提供するために、前記暗号鍵に基づき前記データを暗号化するステップであって、前記データを暗号化するステップは、少なくとも１つの排他的又はXOR演算を含む、ステップと、

前記暗号化データをブロックチェーントランザクションに組み込むステップと、

を含む方法が提供され得る。

【００１１】

データを暗号化するための暗号鍵を生成する方法であって、前記方法は、

第１参加者により、データの暗号化のための暗号鍵を決定するステップであって、前記暗号鍵は、第１データの少なくとも１つの排他的又はXOR結合に基づき、第２データを有する前記第１参加者及び第２参加者に共通の共通シークレットに基づき、前記共通シークレットに基づきデータに一方向関数を適用することに基づく、ステップ、を含む方法が提供され得る。

【００１２】

メッセージに認証データを適用する方法であって、前記方法は、

第１参加者により、第２参加者へ通信されるべきメッセージのためのメッセージ認証データを決定するステップであって、前記メッセージ認証データは、前記第１参加者及び前記第２参加者に知られている関数を、前記メッセージに、及び前記第１参加者及び前記第２参加者に共通の共通シークレットに適用することにより決定される、ステップ、を含む方法が提供され得る。

【００１３】

プロセッサと、プロセッサによる実行の結果として、システムに本願明細書に記載のコンピュータにより実施される方法のいずれかの実施形態を実行させる実行可能命令を含むメモリと、を含むシステムが提供され得る。

【００１４】

実行可能命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記実行可能命令は、コンピュータシステムのプロセッサにより実行された結果として、少なくとも、前記コンピュータシステムに、本願明細書に記載のコンピュータにより実施される方法を実行させる、非一時的コンピュータ可読記憶媒体が提供され得る。

【図面の簡単な説明】

【００１５】

本開示による種々の実施形態は、図面を参照して説明される。

【図１】暗号化及び復号プロトコルの概略図である。

【図２】ブロックチェーントランザクションの償還のためのブロックチェーンスクリプトの処理を示す。

【図３】ブロックデータ暗号化及び復号方式を示す。

【図４】メッセージ認証データを組み込むブロックチェーントランザクションを示す。

【図５】メッセージ認証コードプロトコルを示す。

【図６】種々の実施形態が実装できるコンピューティング環境を示す概略図である。

【発明を実施するための形態】

【００１６】

＜XORに基づく暗号化方式＞

以下の記号：

【数１】

$$\oplus$$

により示されるXOR演算は、２つのバイナリビットの間の演算であり、入力ビットのうちの１つが１に等しい場合に且つその場合にのみ、１を返す。その他の場合、該演算は０の値を返す。これは、以下の真理値表に纏められる。

【表１】

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

【００１７】
メッセージM及びシークレット鍵Sが与えられると、メッセージの導出される暗号文は次式により与えられる：
【数２】

$$M' = M \oplus S$$

例えば、メッセージM=０１１１１０、シークレット鍵S=１０１０１０では、暗号化メッセージはビット毎のXOR演算を用いて計算される：
【数３】

$$\begin{aligned} \mathrm{M} &\quad 011110 \\ \mathrm{S} &\quad \underline{\oplus 101010} \\ \mathrm{M'} &\quad 110100 \end{aligned}$$

【００１８】
復号処理は、XOR演算と次式の結合特性を利用して、元のメッセージを回復する：
【数４】

$$A \oplus A = 0$$

つまり、
【数５】

$$M' \oplus S = (M \oplus S) \oplus S = M \oplus (S \oplus S) = M$$

従って、元のメッセージ（M）を回復させるために、共有シークレット鍵（S）の暗号文メッセージ（M'）とのビット毎XORは、１回のみでよい。
【数６】

$$\begin{aligned} \mathrm{M'} &\quad 110100 \\ \mathrm{S} &\quad \underline{\oplus 101010} \\ \mathrm{M} &\quad 011110 \end{aligned}$$

【００１９】
シークレット鍵とメッセージが任意の傍受パーティに知られていないことを前提に、各シークレット鍵が１回だけ使用されるならば、これは、通信の完璧にセキュアなワンタイムパッド（one－time－pad （OTP））方法を提供する。公衆ネットワークを介して暗号文を転送することの１つのリスクは、傍受パーティの攻撃の場合に、バイナリ文字列の順応性（malleability）である。しかしながら、これは、以下に示すように、パブリックブロックチェーンの不変性を利用することにより、回避できる。
【００２０】

＜共有秘密鍵のセキュアな確立＞

国際特許出願番号WO２０１７/１４５０１６において確立されたように、共有秘密鍵は、２つのパーティ、例えばAlice（A）及びBob（B）の間で、彼らの公開鍵、予め合意した楕円曲線（例えば、scep２５６k１）、及び生成元（generator point、G）を用いて、導出できる。彼らのそれぞれの秘密鍵（Alice及びBobのそれぞれV_A及びV_B）から開始する。彼らは、それぞれ、以下の楕円曲線演算を用いて彼らの公開鍵を計算することにより開始する：

【数７】

$$P_A = V_A \cdot G$$

$$P_B = V_B \cdot G$$

【００２１】

そして、次に、彼らの公開鍵を交換する。Aliceは、彼女の秘密鍵（V_A）及びBobの公開鍵（P_B）を用いて、共有秘密鍵（S_{AB}）を計算する：

【数８】

$$S_{AB} = V_A \cdot P_B$$

【００２２】

Bobは、彼の秘密鍵（V_B）及びAliceの公開鍵（P_A）を用いて、共有秘密鍵（S_{AB}）を計算する：

【数９】

$$S_{AB} = V_B \cdot P_A$$

【００２３】

楕円曲線点加算の結合法則と可換性を用いると、２つの計算された値が同一であることが分かる：

【数１０】

$$V_A \cdot P_B = V_A \cdot (V_B \cdot G) = (V_A \cdot V_B) \cdot G = (V_B \cdot V_A) \cdot G = V_B \cdot (V_A \cdot G) = V_B \cdot P_A$$

【００２４】

共有秘密鍵の計算の結果は楕円曲線上の点（x_{AB},y_{AB}）であり、彼らは例えば$S_1 = x_{AB}$を取り入れることに合意し得る。これは、国際特許出願番号WO２０１７/１４５０１６に記載された構成と同じ階層構造及び同じ処理セットを有するが、生成したシークレットを対称鍵として使用する代わりに、XORに基づくという意味でワンタイムパッド（one－time pad（OTP））として使用される。国際特許出願番号WO２０１７/１４５０１６におけるように、この作業の各処理を拡張して、同様の階層的鍵生成方式を用いて決定性鍵を組み込むことができる。

【００２５】

図１は、非セキュア公衆ネットワークを介する２つのパーティ間の通信のために本開示を利用する鍵生成及びXORメッセージ暗号化プロトコルを示す。AliceとBobとの間の公開鍵P_A及びP_Bの交換に続き、彼らはそれぞれ、楕円曲線乗算を用いて、共有秘密鍵S_1を計算する。Aliceは、シークレット鍵S_1を用いてメッセージMを暗号化し、結果として生じた暗号文M'をブロックチェーン２上のトランザクションに含める。Bobは、暗号文M'を読み出し、それを共有シークレット鍵を用いて復号する。図１の実線のボックスは秘密パラメータを示し、破線のボックスは公に共有される情報を示す。

【００２６】

＜ブロックチェーンを用いる非順応性XOR暗号システム＞

パブリックブロックチェーンの不変性は、非セキュアネットワークを介してバイナリ暗号文を転送することの順応性問題を解決できる。暗号文は、ネットワークへ送信されブロ

ックチェーン上に普遍的に記録されるBitcoinトランザクション内に含めることができる。暗号文が、送信者により署名されマイニングノードにより検証されるトランザクションのアウトプットに含まれる場合、それが署名後に変更されていない場合にのみブロックチェーン上に記録される。同様に、ブロックチェーンから暗号文を読み出す者は、トランザクションのハッシュがトランザクションIDに等しいことをチェックすることにより、トランザクションデータの完全性を確かめることができる。これは、暗号文のエンドツーエンドの不変性を保証する。更に、署名は公開鍵にユニークなので、送信者のアイデンティティも検証できる。

【００２７】

留意すべきことに、データがインプットスクリプトに含まれる場合、トランザクションのこの部分は送信者により署名されないので、真正さは保証できない。

【００２８】

暗号文をトランザクションのアウトプットに挿入するための２つのオプションがある：

（１）使用不可能アウトプット（Unspendable Output）（OP_RETURN）

データがOP_RETURNに含まれる場合、それは、将来のロックスクリプトで使用できないが、暗号文の格納（store）として役立つ。例えば、受信者Bobのアドレスがトランザクションに含まれる場合、Bobのウォレットは、彼のアドレスへの参照についてブロックチェーンをスキャンし（例えば、トークン化プロトコルで使用されるようなスマートウォレットを用いて［William H, Brian P, Flannery P, Saul A, et al. Numerical recipes in C: The Art of Scientific Computing（１９９２）Cambridge］）、暗号文を特定できる。Bobは、次に、上述の確立されたプロトコルにより共有秘密鍵（S_1）を計算し、次に、次式を用いて元のメッセージをローカルに取り出す：

【数１１】

$$M = M' \oplus S_1$$

【００２９】

このプロトコルは、図１に概略的に示される。この方法を用いて、データの送信は、中間者攻撃に対して回復力を有する。

【００３０】

留意すべきことに、この処理はXOR計算しか必要としないので、これは低い処理能力の装置で容易に実行できる。

【００３１】

使用例として、サービスプロバイダAlice（A）がユーザBob（B）に１回限りのセッション鍵を送信したいとする。このセッション鍵は、Bobを宛先とするメッセージとして含まれ、ログイン資格情報を検証するために、通信チャネルを開始するために使用でき、又は従来のユーザ名及びパスワードに基づくシステムを置き換えるアイデンティティ検証を要求する任意の他のアプリケーションで使用できる。

【００３２】

処理は、１回限りのセッション鍵Kを確立するという点で、次のように進行する。共有シークレット鍵を再利用できるために、それは、メッセージを隠蔽するために直接使用されてはならない。セッション鍵を暗号化するために共有鍵のハッシュ（$H(S_1)$）を用いることにより、１回限り公開されるセッション鍵はシークレット鍵を開示しない。

【数１２】

1) A は、$S_1 = x_{AB}$を計算する

2) A は、$K \oplus h(S_1) = S_{K_x}$を計算する

3) A は、B へのトランザクションを生成し、公開テキストとして
スクリプト内にS_{K_x} を含める

4) B は、Tx,を受信し、受信するとB はS_{k_x} を知るために tx を使用
する必要がない

5) B は、$S_1 = x_{AB}$

6) B は、$K = S_{k_x} \oplus h(S_1)$を計算する

7) B は、今や秘密の及び(例えば制御及び証明トークンのための)認
証又はアクセスのために他のシステムで使用可能なセッション
鍵を有する

Aliceは、次に、彼女のシステムへのアクセスを許可するために、ハッシュパズル解h(K)の解を要求し得る。ここでS_1が公開されるリスクがないので、同じ共有シークレットを再利用し続けることが安全である。

【００３３】

（２）使用可能アウトプット（Spendable Output）

上述と同様の方法が、暗号文を使用可能アウトプットとして挿入するために、OP_RETURNの代わりに、OP_DROPを用いて使用できる。しかしながら、Bitcoinスクリプトでは、オペコードOP_XORは、暗号文を復号することにより使用条件を課すことができ、使用後までデータをマスクすることができる。例えば、XOR演算は、AliceとBobがよりプライベートな及びセキュアなスクリプトを生成するために使用できるOTP鍵を人目に付かずに入力又は交換するために使用できる。Bobとの相互作用を伴わず、Aliceは、このシークレット値（S_1）を使用して、未使用のまま、彼への支払いを覆い隠すことができる。

【００３４】

これを行うために、Aliceは以下の形式のロックスクリプトを使用する：

【数１３】

$$OP_DUP\ OP_HASH\ \langle h(S_1)\rangle\ OP_EQUALVERIFY$$
$$\langle h(P_B)\oplus S_1\rangle OP_XOR\ OP_SWAP$$
$$OP_DUP\ \ OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$$

【００３５】

これは、以下のインプットにより解決される：

【数１４】

$$\langle Sig\ P_B\rangle\langle P_B\rangle\langle S_1\rangle$$

【００３６】

ロックスクリプトの第１行は、正しい共有シークレットが使用されることを保証するハッシュパズルである。第２行は、シークレットを使用して〈$h(P_B)$〉を復号する。次に、第３行は、標準的なトランザクションにおけるように、署名をチェックする。償還（redemption）処理の評価は、図２に詳細に示される。

【００３７】

留意すべきことに、上述のトランザクションが使用されるとき、共有シークレットが開示される。これは、上述のスクリプトにおいて、S_1を$S'=h(S_1)$により置き換えることにより回避できる。これは、Bob及びAliceの両者が計算できる。

【００３８】

例として、Aliceは、Bob又は第三者Charlieにより償還可能なトランザクションを生成できる。Charlieに対する償還条件は、AliceとCharlieとの間の共有鍵を用いて同じ方法でロックスクリプトに含まれ得る。上述のハッシュ関数とXOR演算の特性を用いて、Bob及びCharlieのアイデンティティは、トランザクションが償還されるまで完全に隠すことができる。償還されると、トランザクションを償還する者のアイデンティティのみが開示される。

【００３９】

これは、２つ（以上）のパーティが、先に条件を満たすことを匿名で競争している場合、又は１つのパーティが償還する条件を待機している場合に（これは、スポーツまたはその他のイベントの結果を含むことができる）、役立ち得る。複数のパーティ、例としてBob及びCharlieが存在する場合、Aliceは、同じスクリプト内に他者が知らないままBob又はCharlieにより償還可能なようにトランザクションを設定できる。つまり、Aliceは、Bob又はCharlieのいずれかにより償還可能な、条件OP_ORを用いてロックスクリプトを構成する。これは、Bob及びCharlieが、償還のために彼ら自身の公開鍵ハッシュ値のXOR値を計算できるが、他者はできないシナリオである。

【００４０】

＜マルチパーティ通信＞

XOR方式の多様性は、それぞれが彼らの間のセキュアな通信手段を有する多くのメンバを含むための、通信ネットワークの容易な拡張を可能にする。上述の鍵配信プロトコル又はマスター鍵導出アプローチを用いて、非集中型通信ネットワークは、XOR暗号化及びワンタイムパッドを直接的に確立する。XORに基づく暗号化は、低い処理能力の装置においてもセキュア且つ迅速な復号を可能にする。サービスプロバイダがアイデンティティを検証し選択されたパーティへのアクセスを与える集中型システムでは、更なる懸念が解決されなければならない。

【００４１】

例えば、管理者Alice（A）を構成員とする会社又は組織を取り上げる。Aliceは、通信の記録を保持したまま、従業員の間で暗号化通信を提供したい場合がある。例えば、AliceはBob（B）にメッセージを送信し、BobはCharlie（C）にメッセージを送信する（A→B、B→C）。Aliceは、彼らのネットワーク上のイベントの完全な記録を有するために、全部のトランザクションの記録を保持したい。Alice及びBobは、彼らの共有鍵（S_{AB1}）を用いて通信できる。BobがCharlieにメッセージ送信するとき、彼らは彼らの共有鍵（S_{BC1}）を使用するが、このメッセージは導出される秘密鍵（S_{AB2}）を用いてAliceにも中継される。

【数１５】

$$M_1' = (M_1 \oplus S_{AB1})$$

$$M_2' = (M_2 \oplus S_{AB2})$$

$$M_2'' = (M_2 \oplus S_{BC1})$$

【００４２】
これは、従って、Aliceを以降の全部の他のパーティにリンクして、彼女がネットワーク上のアクションを常に通知されることを保証する。これは、サービスプロバイダから完全に隠されないまま、相互に信頼していないユーザ間のプライベート通信を可能にする。
【００４３】
＜鍵導出方式＞
上述の処理から導出される初期共有シークレット鍵を用いて、秘密鍵の新しいセットを確立する必要を有しないで、追加鍵がセキュアに且つ効率的に生成できる。ハッシュ関数の逆算が実行不可能であることを利用して、XOR暗号方式のワンタイムパッド（one－time pad (OTP)）制限が、セキュリティを犠牲にすることなく緩和できる。これは、後述するように、マルチメディアファイルのような大規模なデータ構成をセキュアに暗号化できるよう、非常に多数の鍵へと拡張できる。
【００４４】
上述のような単一の楕円曲線計算を用いて、２つのパーティは秘密鍵$S_1 = x_{AB}$を確立する。一連の鍵は、このメッセージから導出でき、所定数、例えばn個のメッセージを符号化する。鍵セットは、次式に用に導出でき：
【数１６】

$$S_{i+1} = S_1 \oplus h(S_i)$$

【００４５】
次式を生じる：
【数１７】

$$S_1 = x_{AB}$$

$$S_2 = S_1 \oplus h(S_1)$$

$$S_3 = S_1 \oplus h(S_2) = S_1 \oplus h\big(S_1 \oplus h(S_1)\big)$$

$$\vdots$$

$$S_n = S_1 \oplus h(S_{n-1})$$

【００４６】
この鍵セットは、初期共有シークレットのみ、及び単一の初期楕円曲線点計算のみを用いて生成される。これらの鍵は、次に、n個のメッセージを符号化するために使用でき、S_nから開始する。つまり、
【数１８】

$$M_1' = M_1 \oplus S_{n+1}$$

$$M_2' = M_2 \oplus S_n$$

$$\vdots$$

$$M_n' = M_n \oplus S_2$$

【００４７】

初期共有シークレットは、メッセージを暗号化するために単独で使用されることはなく、このような単一の危険に晒されたメッセージが全部の暗号化メッセージのセキュリティを破壊することはない。更に、鍵計算においてハッシュ関数を使用することは、計算の逆算を防ぐ。従って、鍵の中で逆順に使用することにより（$S_{n+1} \rightarrow S_2$）、S_{n+1}の発見はS_nを危険に晒さない。S_1及びS_iを決定することが実現可能ではないので、メッセージ$M_{(n-i+1)}$が与えられたとしても、鍵は依然として安全に再利用できる。

【００４８】

＜ブロック暗号を用いる大規模データ暗号化＞

マルチメディアファイルのような大規模データセットを効率的に暗号化するために、データは、固定長ブロックに分割され、同一の演算が各ブロックに対して実行される。上述のように、行列乗算の特性を利用することにより、追加セキュリティがブロック暗号方式に適用できる［Sastry, V. U. K.; Anup Kumar, K.（２０１２）“A Modified Feistel Cypher Involving XOR Operation and Modular Arithmetic Inverse of a Key Matrix”. International Journal of Advanced Computer Science and Applications (IJACSA). 3(7):３５－３９］。参考文献では、大きな文字列は、各文字のASCII値を行列に加算し、Feistel暗号で（標準的には、順列とXOR演算の繰り返しが使用され、単一の文字列を符号化する）該行列をランダムに生成した鍵行列により乗算することにより符号化される。本願では、大規模データセットをセキュア且つ効率的に暗号化するために、XOR演算を用いてこの処理を拡張するために、決定性鍵がどのように導出できるかを示す。

【００４９】

本明細書では、［Sastry, V. U. K.; Anup Kumar, K.（２０１２）“A Modified Feistel Cypher Involving XOR Operation and Modular Arithmetic Inverse of a Key Matrix”. International Journal of Advanced Computer Science and Applications (IJACSA). 3(7):３５－３９］において導出された処理が概説される。大規模データセットは、ワード（word）として知られる８ビットチャンクに分割され、データが正確に$2m^2$ワードに適合することを保証するためにパディングされる。これらのワードは、次に、２ｍ×ｍ行列Pにエントリとして挿入される。素数モジュラス（prime modulus）Nは、所望のアプリケーションに十分なセキュリティを提供するよう選択される。Pに格納されたデータを暗号化するために、サイズｍ×ｍのランダムに生成された鍵行列Kが構成される。これに続いて、データ行列Pが、２つのｍ×ｍ行列$P=(P_0,Q_0)$に分割される。データ行列を暗号化及び復号するアルゴリズムは以下に示される。暗号化（及び対応する復号）アルゴリズムは、n回繰り返され、追加セキュリティパラメータを提供する。各繰り返しがデータを並べ替えるので、繰り返しの増加に伴い、データは更にスクランブリングされる。この構造は、データ暗号化標準又はDES（n=１６）及び高度暗号化標準又はAES（n=１０－１４）において使用される。

【００５０】

暗号化処理の非線形性を利用することは、複数の暗号文ペアが開示された場合でも、鍵がガウスの消去法アルゴリズム（Gaussian elimination algorithm）を用いて決定できないことを保証する。従って、単一の鍵行列Kが、複数のデータファイルを暗号化するうために使用できる。更に、暗号化及び復号処理の類似性は、計算のために必要なコード及び従ってハードウェアの量を削減する。

【００５１】
＜暗号化＞
【数１９】
Input: $P = (P_0, Q_0), n, K, \text{and } N$

1. For $i = 1\ to\ n$:
   Do
   $P_i = (K \cdot Q_{i-1} \cdot K)\ mod\ N$
   $Q_i = P_{i-1} \oplus P_i$
   End
2. Return $C = (P_n, Q_n)$

【００５２】
＜復号＞
【数２０】
Input: $C = (P_n, Q_n), n, K, \text{and } N$

1. Gauss-Jordan 消去法[8]を用いて Calculate K^{-1}を計算する
2. For $i = 1\ to\ n$:
   Do
   $Q_{i-1} = (K^{-1} \cdot P_i \cdot K^{-1})\ mod\ N$
   $P_{i-1} = Q_i \oplus P_i$
   End
3. $P = (P_0, Q_0)$を返す

【００５３】
＜Feistel暗号鍵行列の構成＞
変更されたFeistel暗号方式がここで提案され、以上に概説した処理を用いて生成されるような鍵S_1のみを使用する。詳細は図３に示される。鍵行列Kは、シークレットS_1のハッシュを繰り返しとり、次にS_1とのビット毎XOR演算を行うことにより構成される。
【００５４】
上述の鍵生成プロトコルを使用して、m(m+１)個のユニークな鍵が、先ず、上述のS_1から導出される。
【数２１】

$$
\begin{aligned}
&1)\ S_1 \oplus h(S_1) && \rightarrow S_2\\
&2)\ S_1 \oplus h^2(S_1) && \rightarrow S_3\\
&\quad\vdots\\
&i)\ S_1 \oplus h^i(S_1) && \rightarrow S_i\\
&\quad\vdots\\
&m(m+1)+1)\ S_1 \oplus h^{m(m+1)}(S_1) && \rightarrow S_{m(m+1)+1}
\end{aligned}
$$

【００５５】
要素を用いて、線形独立な列及び行を有する（従って可逆である）２つの行列を構成す

る。

【数２２】

$$K_1 = \begin{bmatrix} S_2 & 0 & \cdots & 0 \\ S_3 & S_4 & & 0 \\ \vdots & & \ddots & \vdots \\ S_{\frac{m(m-1)}{2}+1} & S_{\frac{m(m-1)}{2}+2} & \cdots & S_{\frac{m(m+1)}{2}+1} \end{bmatrix}$$

$$K_2 = \begin{bmatrix} S_{\frac{m(m+1)}{2}+2} & S_{\frac{m(m+1)}{2}+3} & \cdots & S_{\frac{m(m+3)}{2}+1} \\ 0 & S_{\frac{m(m+3)}{2}+2} & & S_{\frac{m(m+5)}{2}} \\ & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{m(m+1)+1} \end{bmatrix}$$

【００５６】

Kは、次に、$K=K_1K_2$となるよう定義され、ここで、Kも可逆であるとする（復号処理のために必要である）。

【００５７】

＜セキュアなマルチメディア転送＞

大規模データセットを暗号化する更に効率的な方法により、公衆ネットワークを介して、例えばインターネット及び移動電話サービスプロバイダを介して、エンドツーエンドで、マルチメディアファイルをセキュアに送信することが可能である。行列乗算及びGauss－Jordan消去法行列反転（Elimination matrix inversion）の両方が、Coppersmith－Winogradアルゴリズムを用いて、$O(m^{2.376})$の計算的複雑さで、実行できる。これは、最新のモバイル装置においても、１秒未満の暗号化及び復号を可能にする。データを均一なサイズのブロックに予め分割することは、単一のストリームとして、又は複数のトランザクションの間で分割して、膨大なファイルをブロックチェーンに含めることを実現可能にする。前述の章からメッセージングプロトコルを用いて、これは、ピアツーピア通信に容易に統合できる。

【００５８】

＜ストリーミング暗号化を用いる大規模データ暗号化＞

ブロック暗号プロトコルの代替として、鍵が連続的に生成され、データが一度に１ビット、迅速に符号化されるストリーミング暗号化方式がある。セキュアに生成された秘密鍵S_1及び導出された鍵S_2を用いて、ストリーム暗号シードが次式を用いて生成できる：

【数２３】

連結（$conc(\ \ )$）演算、以下で開始する:

$$Seed = SHA_{256}\left(conc(S_1, S_2)\right)$$

【００５９】

シードを標準的なPRNG（pseudo random number generator、疑似乱数生成器）の入力として用い、XOR演算をデータを符号化するために用いて、ストリーム暗号は、高速暗号化及び復号システムとして共有され得る。このようなプロトコルは、２つのパーティ間のメディアファイル又はオーディオ／ビデオ呼のセキュアなストリーミングのために不可欠である。プロトコルの詳細は、ハッシュに基づくメッセージ認証コード（hash－based message authentication code（HMAC））に関して以下に詳述される。

【００６０】

＜メッセージ認証コード（Message Authentication Code（MAC））＞

鍵配信プロトコルを用いて、メッセージ認証コード（Message Authentication Code（MAC））が生成できる。MACは、メッセージの受信者が、送信者のアイデンティティを検証できるようにするとともに、メッセージが改ざんされていないことを保証する。MACは、メッセージが送信されるのと一緒に、Alice（A）及びBob（B）により共に知られる

共有シークレット鍵の公衆に知られている関数を用いて生成される。

【００６１】

＜ブロックチェーンに基づくメッセージ認証コード＞

Alice及びBobは、以上に紹介したプロトコルを用いて共有鍵（S_1）を確立できる。つまり：

【数２４】

$$S_A \cdot S_B \cdot G = P_A S_B$$
$$= P_B S_A$$
$$= (x_{AB}, y_{AB})$$
$$S_1 = (x_{AB})$$

【００６２】

公衆に知られている関数f（例えば、XOR演算を含むもの）を用いて、MACは共有鍵（S_1又はS_1から導出される何からの鍵）及びメッセージMの観点で計算される。Aliceは、値y=f(M,S_1)を計算し、ペア（M,y）をBobへ送信する。ブロックチェーンの文脈で、この情報は、OP_RETURN内のAliceからBobへのトランザクションに、又はトランザクション内のメタデータとして組み込むことができる。Bobは、Aliceからのトランザクション内の情報を取り込み、彼はその有効性をチェックするためにyを再計算できる。メッセージがAliceから生じたものではない、又はメッセージが変更されている場合、関数fは、yの異なる値を生成し、無視され得る。

【００６３】

例として、メッセージMを、OP_RETURNの前の全部のトランザクションデータであると定義することにより、これは、トランザクションデータが送信中に変更されていないことをオフラインで検証する方法を提供し、マイナー検証の必要がない。これは、図４に概略的に示される。図４は、ブロックチェーントランザクションデータを用いるメッセージ認証コード（message authentication code（MAC））を示す。MAC内にトランザクションデータを組み込むことにより、それは、トランザクション検証方式として使用でき、ネットワークに接続する必要を有しないで追加セキュリティを提供する。これは、非常に低い能力のモバイル装置にとって重要である。

【００６４】

図５は、パブリックブロックチェーンを用いる２つのパーティ間の通信のために本開示を利用するメッセージ認証コード（message authentication code（MAC））プロトコルを示す。（図１に示されるような）共有シークレット鍵の確立に続き、Aliceは、合意した関数fを用い、メッセージMを用いて、MAC(y)を計算する。Aliceは、メッセージ及びMACをBobへのトランザクションに含める。Bobは、シークレット鍵を用いてMACを自身で計算し、それをブロックチェーンから読み出したMACと比較する。それらが等しい場合、Bobは、送信者のアイデンティティでメッセージの完全性を確認できる。

【００６５】

この検証アプローチは、２つのパーティ間でオフラインで実行できる。ここで、Bobは、Aliceからの情報M及び共有シークレットS１を使用して、オフラインで値を受信し、後に値yを認証として受け入れることができる。つまり、鍵S_1が計算された後に、Aliceは、メタデータとして含まれるメッセージ（M）を有するトランザクションtxを送信する。Aliceは、Bobにメッセージペア（y,M）をオフラインで送信して、リモートシステムへの認証を行い、これはブロックチェーン上のトランザクションを認識する。

【００６６】

処理は、以下の通りである。

【数２５】

1) A は、(Mを含む)トランザクションをBへ送信する

2) A は、yをBへ直接に又はトランザクション内に含めて送信する

3) ホストは共有シークレットS_1を用いて$y = f(M, S_1)$を計算する

メッセージが変更された場合、

$y' = f(M', S_1)$が異なる

【００６７】

MACは、符号化又は送信の際のランダムなエラーを検出するために、CRC（Cyclic Redundancy Check、巡回冗長検査）としても使用できる。同じMACが送信に続いて再生成できることを検証することにより、正しいメッセージが復元されることを保証できる。これは、リモートシステムへの認証のために、オフライン方法としても使用できる。

【００６８】

＜信頼性＞

MACは、トランザクション及び認証に信頼性を追加するために暗号化できる（例えば、サーバへ送信するとき、値y及びMを隠す）。２番目に生成された鍵はMACを暗号化し及び追加プライバシーを提供するために使用できる。これは、第２のMAC鍵を生成するためにハッシュを取り入れ、同時に認証と信頼性の両方を可能にすることにより、拡張できる。複数回認証するために、Aliceがサーバに認証しようとする度に、新しい鍵が使用されなければならない。

【００６９】

必要な場合には、サブキーを使用することにより、追加の複雑性が追加できる。例えば、TAが最後の有効なブロックハッシュを用いる時間マーカであるならば、これは、鍵階層構造を形成するために組み込むことができる。

【数２６】

$$P_A \rightarrow P_{A(1)} = P_A + T_A\, G.$$

【００７０】

$S_1(S_2, S_3)$から導出される鍵のペアは、次に、係数として取り入れられ、関数を次のように定義する：

【数２７】

$$f(M, K) = S_2\, M + S_3$$

【００７１】

f(M,K)は、何らかの合意された適切な整数Nについて、$y=(S_2 M+S_3) \bmod N$として更に強力にできる。攻撃者が１つの(M,y)ペアを知った場合、これは、次のメッセージM'についてy'を決定する際に保持される。

【００７２】

＜ショートMAC＞

少ない量のセキュリティが犠牲になり得るが、依然として十分にセキュアなMACを有する。Bitcoinでは、鍵長は、単一の座標（S_1）のみを取り入れ、２５６ビットである。低い能力の装置では、これは、モジュラスに低次の値を使用することにより削減することができ、中程度の計算応力により妥当な時間内に処理を実行可能にする。小さいN、例えば６４ビットが選択される場合：

【数２８】

$y \equiv S_2 M + S_3 \bmod N$ は 64 ビット値に限定される

【００７３】
代替として、Ｍは、Ｎ以下の長さを有するブロックに分割できる：
【数２９】
$M = M_1, M_2, \ldots, M_t$, where $0 \leqslant M_i \leqslant N$

対応する数のユニークな鍵が、上述のプロトコルを用いて導出できる：
【数３０】
$k = (S_2, S_3, \ldots, S_{t+1})$, where $0 \leqslant S_i < N$

【数３１】

$$f(M, K) = \sum S_i \, M_i \, S_2 \, (mod \, N)$$

【００７４】
最後に、次式の低次ビットのみを送信する、あまりセキュアではないが非常に簡易な手段を有することが可能である。
【数３２】

$$y = S_2 \, M + S_3 \, (mod \, N).$$

【００７５】
＜ストリーミングハッシュに基づくＭＡＣ（Streaming Hash－Based MAC (HMAC)）＞
アプリケーションが標準的なＭＡＣにより提供されるものに追加のセキュリティを要求する場合、ハッシュに基づくＭＡＣ（ＨＭＡＣ）が使用できる。ＭＡＣと異なり、ＨＭＡＣは、第３章で提案されたストリーム又はブロック暗号方式を通じて、（国際特許出願番号ＷＯ２０１７/１４５０１６に開示される構成の要件であるように、ＳＨＡ－２５６ハッシュの鍵サイズを超える）大規模ファイルへ容易に拡張できる。これは、ファイルを２５６ビット部分に分割することにより行うことができる。
【数３３】

$$B_0, B_1, \ldots, B_n$$

ここで、$n = (file \, size)/256$。B_n は０でパディングされ、全てのB_iの間でサイズの均一性を保証する

【００７６】
＜プロトコル＞
国際特許出願番号ＷＯ２０１７/１４５０１６に開示され上述した方法を用いて導出されたシークレットS_1で開始すると、後続の鍵は、次式を含む種々の演算を用いてセキュアに導出できる：
【数３４】
モジュラ加算(+), 排他的論理和(⊕), ハッシング($h(x)$), 及び連結($conc(a, b)$)

疑似乱数生成器（pseudorandom number generator (PRNG)）の使用に対する５個の効率的な代替案は、以下を使用する：
【数３５】

1. $S_i = S_{i-1} + h\big(conc(i-1, S_1, B_0)\big)$
2. $S_i = S_{i-1} + conc\big(h(B_0), h(S_{i-1})\big)$
3. $S_i = S_{i-1} \oplus h(S_{i-1})$
4. $S_i = h(S_{i-1}) \oplus h(B_{i-1})$
5. $S_i = h\big(conc(S_{i-1}, B_{i-1})\big)$

【００７７】
ここで、後続の鍵を導出するための方法により、値S_iは、B_iに対してXOR結合され、OTP（ワンタイムパッド）として機能し、ストリーム又はチャネルをセキュアにする。明らかに、暗号ブロック（C_i）は以下のように計算できる：
【数３６】

$$C_i = S_i \oplus B_i$$

【００７８】
全部の導出された鍵は、S_0及び初期ブロックハッシュ（初期の２５６ビット）を用いて計算でき、データ完全性を保証できる。
【００７９】
＜使用例＞
＜セキュアなピアツーピアメッセージングサービス＞
エンドツーエンド暗号化を伴うピアツーピア通信は、以上で検証されたマルチメディアファイル及びストリーミングデータを組み込むための拡張と共に、以上に詳細に説明された。これらのプロトコルをブロックチェーンシステムに自由に統合する能力は、アプリケーション開発者が、彼らのユーザに改ざん耐性のあるセキュリティを提供するために、更なる自由度を提供する。メッセージ認証コード（Message Authentication Code (MAC)）の組み込みにより、メッセージが請求されるアイデンティティから生じていること、及びメッセージを不明瞭にし又は変更する試みがなかったことの、追加の確実さをユーザに提供できる。非常に機密性のある情報を送信するために、これは重要な特徴である。
【００８０】
本開示は、XOR演算を用いて効率的且つセキュアな暗号システムを提供する。必要な計算の簡易性は、Bitcoin（BSV）のような最も基本的な言語によるブロックチェーンを使用しても、容易に組み込むことを可能にする。これらの方式は、多くの異なるシステム及び装置に渡りセキュア且つ多様なアプリケーションの急伸を可能にする。
【００８１】
図６を参照すると、本開示の少なくとも一実施形態を実施するために使用され得るコンピューティング装置２６００の説明のための簡略ブロック図が提供される。種々の実施形態で、コンピューティング装置２６００は、上述の図示のシステムのうちのいずれかを実装するために使用されてよい。例えば、コンピューティング装置２６００は、データサーバ、ウェブサーバ、ポータブルコンピューティング装置、パーソナルコンピュータ、又は任意の電子コンピューティング装置として使用するために構成されてよい。図６に示すように、コンピューティング装置２６００は、主メモリ２６０８及び永久記憶装置２６１０を含む記憶サブシステム２６０６と通信するよう構成され得る１つ以上のレベルのキャッシュメモリ及びメモリ制御部（集合的に２６０２とラベル付けされる）を備える１つ以上のプロセッサを含んでよい。主メモリ２６０８は、図示のように、動的ランダムアクセスメモリ（ＤＲＡＭ）２６１８及び読み出し専用メモリ（ＲＯＭ）２６２０を含み得る。記憶サブシステム２６０６及びキャッシュメモリ２６０２は、本開示で説明されたようなトランザクション及びブロックに関連付けられた詳細事項のような情報の記憶のために使用

されてよい。プロセッサ２６０２は、本開示で説明されたような任意の実施形態のステップ又は機能を提供するために利用されてよい。

【００８２】

プロセッサ２６０２は、１つ以上のユーザインタフェース入力装置２６１２、１つ以上のユーザインタフェース出力装置２６１４、及びネットワークインタフェースサブシステム２６１６とも通信できる。

【００８３】

バスサブシステム２６０４は、コンピューティング装置２６００の種々のコンポーネント及びサブシステムが意図した通りに互いに通信できるようにするメカニズムを提供してよい。バスサブシステム２６０４は、単一のバスとして概略的に示されるが、バスサブシステムの代替の実施形態は、複数のバスを利用してよい。

【００８４】

ネットワークインタフェースサブシステム２６１６は、他のコンピューティング装置及びネットワークへのインタフェースを提供してよい。ネットワークインタフェースサブシステム２６１６は、幾つかの実施形態では、コンピューティング装置２６００の他のシステムからデータを受信し及びそれへデータを送信するインタフェースとして機能してよい。例えば、ネットワークインタフェースサブシステム２６１６は、データ技術者が、装置をネットワークに接続することを可能にする。その結果、データ技術者は、データセンタのような遠隔地にいがなら、データを装置へ送信し、データを装置から受信できる。

【００８５】

ユーザインタフェース入力装置２６１２は、キーボード、統合型マウス、トラックボール、タッチパッド、又はグラフィックタブレットのような指示装置、スキャナ、バーコードスキャナ、ディスプレイに組み込まれたタッチスクリーン、音声認識システム、マイクロフォンのようなオーディオ入力装置、及び他の種類の入力装置のような、１つ以上のユーザ入力装置を含んでよい。通常、用語「入力装置」の使用は、コンピューティング装置２６００に情報を入力する全ての可能な種類の装置及びメカニズムを含むことを意図する。

【００８６】

１つ以上のユーザインタフェース出力装置２６１４は、ディスプレイサブシステム、プリンタ、又は音声出力装置のような非視覚ディスプレイ、等を含んでよい。ディスプレイサブシステムは、陰極線管（CRT）、液晶ディスプレイ（LCD）、発光ダイオード（LED）ディスプレイ、又はプロジェクションのような平面装置、又は他のディスプレイ装置を含んでよい。通常、用語「出力装置」の使用は、コンピューティング装置２６００から情報を出力する全ての可能な種類の装置及びメカニズムを含むことを意図する。１つ以上のユーザインタフェース出力装置２６１４は、例えば、ユーザインタフェースを提示して、ここに記載したプロセス及び変形を実行するアプリケーションとのユーザ相互作用が適切であるとき、そのような相互作用を実現するために使用されてよい。

【００８７】

記憶サブシステム２６０６は、本開示の少なくとも１つの実施形態の機能を提供する基本プログラミング及びデータ構造を記憶するコンピュータ可読記憶媒体を提供してよい。アプリケーション（例えば、プログラム、コードモジュール、命令）は、１つ以上のプロセッサにより実行されると、本開示の１つ以上の実施形態の機能を提供し、記憶サブシステム２６０６に格納されてよい。これらのアプリケーションモジュール又は命令は、１つ以上のプロセッサ２６０２により実行されてよい。記憶サブシステム２６０６は、更に、本開示に従い使用されるデータを格納するレポジトリを提供する。例えば、主メモリ２６０８及びキャッシュメモリ２６０２は、プログラム及びデータのための揮発性記憶を提供できる。永久記憶装置２６１０は、プログラム及びデータの永久（不揮発性）記憶を提供でき、磁気ハードディスクドライブ、取り外し可能媒体に関連付けられた１つ以上のフロッピディスクドライブ、取り外し可能媒体に関連付けられた１つ以上の光ドライブ（例えば、ＣＤ－ＲＯＭ、又はＤＶＤ、又はＢｌｕｅ－Ｒａｙ）ドライブ、及び他の同様の記憶媒体を含んでよい。このようなプログラム及びデータは、本開示に記載した１つ以上の実

施形態のステップを実行するためのプログラム、及び本開示に記載したトランザクション及びブロックに関連付けられたデータを含み得る。

【００８８】

コンピューティング装置２６００は、ポータブルコンピュータ装置、タブレットコンピュータ、ワークステーション、又は後述する任意の他の装置を含む種々のタイプのものであってよい。さらに、コンピューティング装置２６００は、１つ以上のポート（例えば、USB、ヘッドフォンジャック、光コネクタ、等）を通じてコンピューティング装置２６００に接続可能な別の装置を含み得る。コンピューティング装置２６００に接続され得る装置は、光ファイバコネクタを受けるよう構成される複数のポートを含んでよい。従って、この装置は、光信号を、処理のために装置を接続するポートを通じてコンピューティング装置２６００に送信される電気信号に変換するよう構成されてよい。コンピュータ及びネットワークの絶えず変化する特性により、図６に示したコンピューティング装置２６００の説明は、装置の好適な実施形態を説明する目的の特定の例としてのみ意図される。図６に示したシステムより多くの又は少ないコンポーネントを有する多くの他の構成が可能である。

【００８９】

＜列挙される例示的な実施形態＞

本開示の実施形態の例は、以下の項の観点で記載できる。

【００９０】

（項１）データを暗号化する方法であって、

第１参加者により、前記第１参加者及び第２参加者に共通の共通シークレットに基づき、データの暗号化のための暗号鍵を決定するステップと、

暗号化データを提供するために、前記暗号鍵に基づき前記データを暗号化するステップであって、前記データを暗号化するステップは、少なくとも１つの排他的又はXOR演算を含む、ステップと、

前記暗号化データをブロックチェーントランザクションに組み込むステップと、

を含む方法。

【００９１】

少なくとも１つの排他的又はXOR演算を含む暗号化処理を提供することにより、これは、同じ長さのシークレット鍵を有するバイナリメッセージ文字列のビット毎XOR符号化を使用することにより、XOR暗号が導出でき公衆ネットワーク上で自由に享有できるという利点を提供する。後に、暗号文は、同じシークレット鍵及びメッセージに対するビット毎XOR演算を用いて復号できる。プロトコルの簡易性は、任意の暗号に基づくシステムに組み込むことができるという更なる利点を有する。本発明の計算上の効率性が、国際特許出願番号WO２０１７/１４５０１６で提案されたセッション鍵生成を、低い能力の装置（モバイル装置又はスマートカード）において使用可能にし、及びBitcoinにおけるスクリプト内に含むことができるという利点も提供される。データを変更するという又はデータの変更を検出不可能にするというリスクを最小限にするという利点も提供される。

【００９２】

（項２）前記暗号化データは、前記ブロックチェーントランザクションのアウトプットに含まれる、項１に記載の方法。

【００９３】

これは、ブロックチェーントランザクションのアウトプットの中のデータが検証可能なデジタル署名を有し、その結果、データの送信者のアイデンティティが受信者によりチェックでき、データが署名後に変更されていないことのチェックが行われるという利点を提供する。

【００９４】

（項３）前記暗号化データは前記トランザクションの中の公衆アクセス可能なデータに含まれる、項１又は２に記載の方法。

【００９５】

これは、復号のためのデータへのアクセスが、ブロックチェーントランザクションを償還／使用する必要がなく、取得できるという利点を提供する。

【００９６】

（項４）前記共通シークレットを用いて、前記暗号化データからセッション鍵及び／又はパスワードが導出できる、項１～３のいずれか一項に記載の方法。

【００９７】

（項５）前記ブロックチェーントランザクションは前記共通シークレットに基づきスクリプトを用いて償還される、項１～４のいずれか一項に記載の方法。

【００９８】

（項６）前記トランザクションは、前記共通シークレットに基づくデータに一方向関数を適用することに基づき、データを含むスクリプトを用いて償還される、項５に記載の方法。

【００９９】

これは、共通シークレットがトランザクションの償還により開示されず、従って再利用できるという利点を提供する。

【０１００】

（項７）前記一方向関数は、ハッシュ関数である、項６に記載の方法。

【０１０１】

（項８）前記トランザクションの償還は前記データを公開する、項１～７のいずれか一項に記載の方法。

【０１０２】

（項９）前記トランザクションの償還は、前記トランザクションを償還する参加者を識別させる、項１～８のいずれか一項に記載の方法。

【０１０３】

これは、トランザクションが償還されるまでプライバシーを維持するが、１つより多くの公開鍵の場合にトランザクションの消費者を識別可能にするという利点を提供する。

【０１０４】

（項１０）暗号化データを前記第２参加者へ送信した結果として、前記第１参加者及び第３参加者に共通の第２共通シークレットに基づく暗号鍵を用いて、前記データを暗号化させるステップ、を更に含む項１～９のいずれか一項に記載の方法。

【０１０５】

これは、通信の記録を有するマルチパーティ暗号化通信を可能にするという利点を提供する。

【０１０６】

（項１１）データが前記第２参加者から第３参加者へ送信された結果として、データを前記第１参加者へ送信させるステップ、を更に含む項１～１０のいずれか一項に記載の方法。

【０１０７】

（項１２）前記共通シークレットに基づきデータに一方向関数を繰り返し適用することにより、複数の前記暗号鍵を決定するステップ、を更に含む項１～１１のいずれか一項に記載の方法。

【０１０８】

これは、共通シークレットが１回のみ確立されればよく、次に複数の鍵の生成のためのシードとして使用され、オンラインでデータを交換する必要を伴わないという利点を提供する。

【０１０９】

（項１３）前記一方向関数は、ハッシュ関数である、項１２に記載の方法。

【０１１０】

（項１４）第１データを暗号化するために使用される第１暗号鍵は、前記第１暗号鍵に基づきデータに前記一方向関数の適用を適用することにより前記第１データを暗号化した

後に、第２データを暗号化するために使用される第２暗号鍵に関連する、項１２又は１３に記載の方法。

【０１１１】

これは、共有シークレット秘密性を維持するという利点を提供する。

【０１１２】

（項１５）前記暗号化は、前記データの部分に少なくとも１つの暗号鍵を適用することにより実行される、項１～１４のいずれか一項に記載の方法。

【０１１３】

これは、大規模データセットの場合に、効率を向上するという利点を提供する。

【０１１４】

（項１６）少なくとも１つの前記暗号鍵は行列である、項１～１５のいずれか一項に記載の方法。

【０１１５】

これは、行列乗算の非線形特性のために、よりセキュアな暗号化を可能にするという利点を提供する。

【０１１６】

（項１７）前記行列の要素は、前記共通シークレットに基づきデータに一方向関数を繰り返し適用することにより生成される、項１６に記載の方法。

【０１１７】

（項１８）少なくとも１つの前記行列は、線形独立の列及び行を有する２つの行列の積から導出される、項１６又は１７に記載の方法。

【０１１８】

これは、可逆行列を提供し、それにより復号鍵の計算を可能にするという利点を提供する。

【０１１９】

（項１９）前記暗号化は、前記データの少なくとも１つの部分に少なくとも１つの暗号鍵を繰り返し適用することにより実行される、項１～１８のいずれか一項に記載の方法。

【０１２０】

これは、繰り返し暗号化のような非線形特性のために、更なるセキュリティを提供するという利点を提供する。

【０１２１】

（項２０）前記第１参加者において、前記共通シークレットを決定するステップであって、前記第１参加者は、第１秘密鍵と第１公開鍵とを有する暗号システムの第１公開－秘密鍵ペアに関連付けられ、前記第２参加者は、第２秘密鍵と第２公開鍵とを有する前記暗号システムの第２公開－秘密鍵ペアに関連付けられ、前記共通シークレットは、前記第１参加者において前記第１秘密鍵及び前記第２公開鍵に基づき決定され、前記暗号システムの特性は、前記共通シークレットが前記第２参加者において前記第２秘密鍵及び前記第１公開鍵に基づき決定できることである、ステップを更に含む、項１～１９のいずれか一項に記載の方法。

【０１２２】

（項２１）前記暗号システムは、楕円曲線暗号システムである、項２１に記載の方法。

【０１２３】

（項２２）前記第１参加者は前記共通シークレットのシェアを有し、前記共通シークレットは、閾数の前記シェアによりアクセス可能であり、前記閾数より少ないシェアでアクセス不可能である、項１～２１のいずれか一項に記載の方法。

【０１２４】

（項２３）データを暗号化するための暗号鍵を生成する方法であって、前記方法は、

第１参加者により、データの暗号化のための暗号鍵を決定するステップであって、前記暗号鍵は、第１データの少なくとも１回の排他的又はXOR結合に基づき、第２データを有する前記第１参加者及び第２参加者に共通の共通シークレットに基づき、前記共通シーク

レットに基づきデータに一方向関数を適用することに基づく、ステップ、を含む方法。
【０１２５】
これは、計算上の効率性、当該方法を低処理能力の装置により使用可能にするという利点を提供する。
【０１２６】
（項２４）前記共通シークレットに基づきデータに前記一方向関数を繰り返し適用することにより、複数の前記暗号鍵を決定するステップ、を更に含む項２３に記載の方法。
【０１２７】
（項２５）前記一方向関数は、ハッシュ関数である、項２３又は２４に記載の方法。
【０１２８】
（項２６）第３データを暗号化するために使用される第１暗号鍵は、前記第１暗号鍵に基づきデータに前記一方向関数を適用することにより前記第３データを暗号化した後に、第４データを暗号化するために使用される第２暗号鍵に関連する、項２３～２５のいずれか一項に記載の方法。
【０１２９】
（項２７）前記暗号化は、前記データの部分に少なくとも１つの暗号鍵を適用することにより実行される、項２３～２６のいずれか一項に記載の方法。
【０１３０】
（項２８）少なくとも１つの前記暗号鍵は行列である、項２３～２７のいずれか一項に記載の方法。
【０１３１】
（項２９）前記行列の要素は、前記共通シークレットに基づきデータに前記一方向関数を繰り返し適用することにより生成される、項２８に記載の方法。
【０１３２】
（項３０）少なくとも１つの前記行列は、線形独立の列及び行を有する２つの行列の積から導出される、項２８又は２９に記載の方法。
【０１３３】
（項３１）前記暗号化は、前記データの少なくとも１つの部分に少なくとも１つの暗号鍵を繰り返し適用することにより実行される、項２３～３０のいずれか一項に記載の方法。
【０１３４】
（項３２）前記第１参加者において、前記共通シークレットを決定するステップであって、前記第１参加者は、第１秘密鍵と第１公開鍵とを有する暗号システムの第１公開－秘密鍵ペアに関連付けられ、前記第２参加者は、第２秘密鍵と第２公開鍵とを有する前記暗号システムの第２公開－秘密鍵ペアに関連付けられ、前記共通シークレットは、前記第１参加者において前記第１秘密鍵及び前記第２公開鍵に基づき決定され、前記暗号システムの特性は、前記共通シークレットが前記第２参加者において前記第２秘密鍵及び前記第１公開鍵に基づき決定できることである、ステップを更に含む、項２３～３１のいずれか一項に記載の方法。
【０１３５】
（項３３）前記暗号システムは、楕円曲線暗号システムである、項３２に記載の方法。
【０１３６】
（項３４）前記第１参加者は前記共通シークレットのシェアを有し、前記共通シークレットは、閾数の前記シェアによりアクセス可能であり、前記閾数より少ないシェアでアクセス不可能である、項２３～３３のいずれか一項に記載の方法。
【０１３７】
（項３５）データを暗号化する方法であって、前記法は、
項２３～３４のいずれか一項に記載の方法により暗号鍵を生成するステップと、
暗号化データを提供するために、前記暗号鍵に基づきデータを暗号化するステップであって、前記データを暗号化する前記ステップは、少なくとも１つの排他的又はXOR演算を含む、ステップと、

を含む方法。
【０１３８】
（項３６）前記暗号化データをブロックチェーントランザクションに組み込むステップ、を更に含む項３５に記載の方法。
【０１３９】
（項３７）前記暗号化データは、前記ブロックチェーントランザクションのアウトプットに含まれる、項３６に記載の方法。
【０１４０】
（項３８）前記暗号化データは前記トランザクションの中の公衆アクセス可能なデータに含まれる、項３６又は３７に記載の方法。
【０１４１】
（項３９）前記共通シークレットを用いて、前記暗号化データからセッション鍵及び／又はパスワードが導出できる、項３６～３８のいずれか一項に記載の方法。
【０１４２】
（項４０）前記ブロックチェーントランザクションは前記共通シークレットに基づきスクリプトを用いて償還される、項３６～３９のいずれか一項に記載の方法。
【０１４３】
（項４１）前記トランザクションは、前記共通シークレットに基づくデータに一方向関数を適用することに基づき、データを含むスクリプトを用いて償還される、項４０に記載の方法。
【０１４４】
（項４２）前記一方向関数は、ハッシュ関数である、項４１に記載の方法。
【０１４５】
（項４３）前記トランザクションの償還は前記データを公開する、項３６～４２のいずれか一項に記載の方法。
【０１４６】
（項４４）前記トランザクションの償還は、前記トランザクションを償還する参加者を識別させる、項３６～４３のいずれか一項に記載の方法。
【０１４７】
（項４５）メッセージに認証データを適用する方法であって、前記方法は、
第１参加者により、第２参加者へ通信されるべきメッセージのためのメッセージ認証データを決定するステップであって、前記メッセージ認証データは、前記第１参加者及び前記第２参加者に知られている関数を、前記メッセージに、及び前記第１参加者及び前記第２参加者に共通の共通シークレットに適用することにより決定される、ステップ、を含む方法。
【０１４８】
これは、メッセージ認証データが最初に決定されて以来、データが損傷していないことをチェックするために、メッセージ認証データを第２ノードにおいて再計算可能にするという利点を提供する。
【０１４９】
（項４６）前記メッセージは、ブロックチェーントランザクションに含まれる、項４５に記載の方法。
【０１５０】
（項４７）前記メッセージは前のブロックチェーントランザクションに関するデータを含む、項４６に記載の方法。
【０１５１】
これは、トランザクションデータが送信中に変更されていないことのチェックをオフラインで行うことを可能にするという利点を提供する。これは、特に、低能力装置の場合に役立つ。
【０１５２】

（項４８）前記メッセージ認証データは、前記ブロックチェーントランザクションに含まれる、項４６又は４７に記載の方法。

【０１５３】

（項４９）少なくとも前記第２参加者へ、前記ブロックチェーントランザクションと別個に前記メッセージ認証データを通信するステップ、を更に含む項４６～４８のいずれか一項に記載の方法。

【０１５４】

（項５０）前記メッセージ認証データは前記共通シークレットに基づく複数のデータアイテムを含む、項４５～４９のいずれか一項に記載の方法。

【０１５５】

（項５１）複数の前記データアイテムは、前記メッセージのそれぞれの部分に適用される、項５０に記載の方法。

【０１５６】

これは、大規模データアイテムの場合に、メッセージデータの処理を更に効率的にするという利点を提供する。

【０１５７】

（項５２）前記関数は、少なくとも１つの排他的又はXOR演算を含む、項４５～５１のいずれか一項に記載の方法。

【０１５８】

（項５３）前記メッセージ認証データは、前記共通シークレットに基づき一方向関数をデータに繰り返し適用することにより決定される複数のデータアイテムを含む、項４５～５２のいずれか一項に記載の方法。

【０１５９】

（項５４）前記一方向関数は、ハッシュ関数である、項５３に記載の方法。

【０１６０】

（項５５）第１メッセージに適用される第１データアイテムは、前記共通シークレットに基づき前記一方向関数をデータに適用することにより前記第１データアイテムを前記第１メッセージに適用した後に、第２メッセージに適用される第２データに関連する、項５３又は５４に記載の方法。

【０１６１】

（項５６）前記第１参加者において、前記共通シークレットを決定するステップであって、前記第１参加者は、第１秘密鍵と第１公開鍵とを有する暗号システムの第１公開－秘密鍵ペアに関連付けられ、前記第２参加者は、第２秘密鍵と第２公開鍵とを有する前記暗号システムの第２公開－秘密鍵ペアに関連付けられ、前記共通シークレットは、前記第１参加者において前記第１秘密鍵及び前記第２公開鍵に基づき決定され、前記暗号システムの特性は、前記共通シークレットが前記第２参加者において前記第２秘密鍵及び前記第１公開鍵に基づき決定できることである、ステップを更に含む、項４５～５５のいずれか一項に記載の方法。

【０１６２】

（項５７）前記暗号システムは、楕円曲線暗号システムである、項５６に記載の方法。

【０１６３】

（項５８）前記第１参加者は前記共通シークレットのシェアを有し、前記共通シークレットは、閾数の前記シェアによりアクセス可能であり、前記閾数より少ないシェアでアクセス不可能である、項４５～５７のいずれか一項に記載の方法。

【０１６４】

（項５９）コンピュータにより実装されるシステムであって、

プロセッサと、

前記プロセッサによる実行の結果として、前記システムに項１～５８のいずれかに記載のコンピュータにより実施される方法のいずれかの実施形態を実行させる実行可能命令を含むメモリと、

を含むシステム。

【０１６５】

（項６０）実行可能命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記実行可能命令は、コンピュータシステムのプロセッサにより実行された結果として、少なくとも、前記コンピュータシステムに、項１～５８のいずれかに記載の方法の実施形態を実行させる、非一時的コンピュータ可読記憶媒体。

【０１６６】

上述の実施形態は、本開示を限定するのではなく、説明すること、及び当業者は添付の特許請求の範囲により定められる本開示の範囲から逸脱することなく多くの代替的実施形態を考案できることに留意すべきである。請求項において、括弧内の任意の参照符号は、請求項を限定すると考えられるべきではない。用語「有する」及び「含む」（comprising、comprises）等は、任意の請求項又は明細書全体に列挙されたもの以外の要素又はステップの存在を排除しない。本願明細書では、「有する」は「有する又は構成される」を意味し、「含む」は「含む又は構成される」を意味する。要素の単数の参照は、そのような要素の複数の参照を排除しない。逆も同様である。本開示は、幾つかの別個の要素を含むハードウェアにより、及び適切にプログラムされたコンピュータにより、実装できる。幾つかの手段を列挙する装置クレームでは、これらの手段のうちの幾つかは、１つの同じハードウェアアイテムにより具現化されてよい。単に特定の手段が相互に異なる従属請求項に記載されるという事実は、これらの手段の組み合わせが有利に使用されないことを示さない。

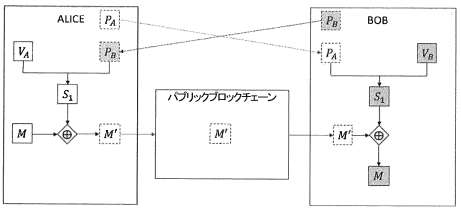
【０１６７】

参考文献

[１] Merkle, Ralph C. (April １９７８). "Secure Communications Over Insecure Channels". Communications of the ACM. ２１ (４): ２９４－２９９.
[２] Diffie, Whitfield; Hellman, Martin E. (November １９７６). "New Directions in Cryptography」 IEEE Transactions on Information Theory. ２２ (６): ６４４－６５４.
[３] Menezes, A.; van Oorschot, P.; Vanstone, S. (１９９６). “Handbook of Applied Cryptography”. CRC Press. p.５００.
[４] International patent application WO２０１７/１４５０１６
[５] Belding, J.; Georges, S.; Barr, S.; Uddeen, F.; Lee, B. (２０１８) “Tokenized: A token Protocol for the Bitcoin (BSV) Network.
[６] Sastry, V.U. K.; Anup Kumar, K. (２０１２) “A Modified Feistel Cypher Involving XOR Operation and Modular Arithmetic Inverse of a Key Matrix”. International Journal of Advanced Computer Science and Applications (IJACSA). ３(７):３５－３９.
[７] William H, Brian P, Flannery P, Saul A, et al. Numerical recipes in C: The Art of Scientific Computing (１９９２) Cambridge.

【図面】

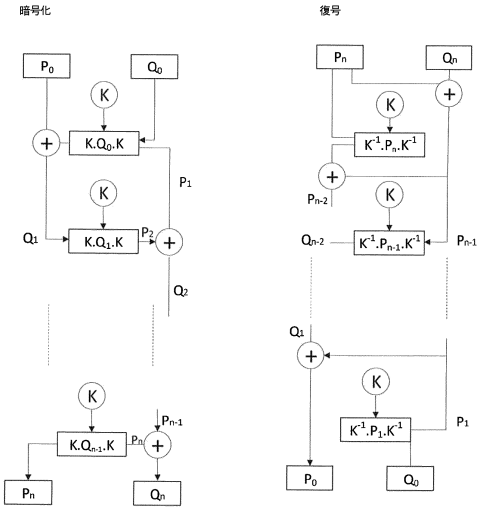
【図１】



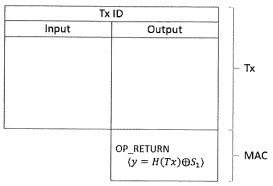
【図２】

Stack	Script
	$\langle Sig\ P_B\rangle\langle P_B\rangle\langle S_1\rangle OP_DUP\ OP_HASH\langle h(S_1)\rangle\ OP_EQUALVERIFY\ \langle h(P_B)\oplus S_1\rangle OP_XOR\ OP_SWAP\ OP_DUP\ OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle h(S_1)\rangle$ $\langle S_1\rangle$ $\langle P_B\rangle$ $\langle Sig\ P_B\rangle$	$\langle h(S_1)\rangle\ OP_EQUALVERIFY\ \langle h(P_B)\oplus S_1\rangle\ OP_XOR\ OP_SWAP$ $OP_DUP\ \ OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle S_1\rangle$ $\langle P_B\rangle$ $\langle Sig\ P_B\rangle$	$\langle h(P_B)\oplus S_1\rangle\ OP_XOR\ OP_SWAP$ $OP_DUP\ \ OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle h(P_B)\oplus S_1\rangle$ $\langle S_1\rangle$ $\langle P_B\rangle$ $\langle Sig\ P_B\rangle$	$OP_XOR\ OP_SWAP$ $OP_DUP\ \ OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle h(P_B)\rangle$ $\langle P_B\rangle$ $\langle Sig\ P_B\rangle$	$OP_SWAP\ OP_DUP\ \ OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle P_B\rangle$ $\langle h(P_B)\rangle$ $\langle Sig\ P_B\rangle$	$OP_DUP\ \ OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle P_B\rangle$ $\langle P_B\rangle$ $\langle h(P_B)\rangle$ $\langle Sig\ P_B\rangle$	$OP_HASH\ OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle h(P_B)\rangle$ $\langle P_B\rangle$ $\langle h(P_B)\rangle$ $\langle Sig\ P_B\rangle$	$OP_ROT\ OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle h(P_B)\rangle$ $\langle h(P_B)\rangle$ $\langle P_B\rangle$ $\langle Sig\ P_B\rangle$	$OP_EQUALVERIFY\ OP_CHECKSIG$
$\langle P_B\rangle$ $\langle Sig\ P_B\rangle$	$OP_CHECKSIG$
True	

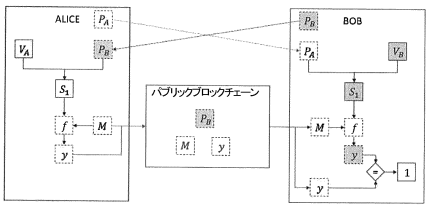
【図３】



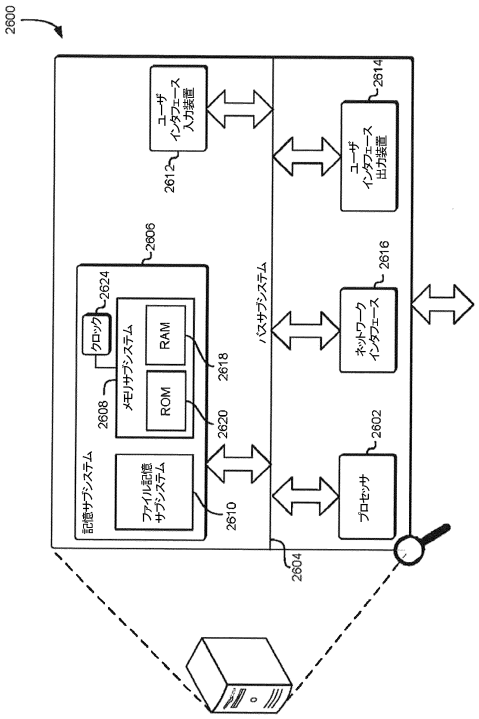
【図４】



【図５】


ALICE
BOB
パブリックブロックチェーン

【図６】


2600
ユーザインタフェース入力装置
2612
ユーザインタフェース出力装置
2614
2606
2624
ブロック
メモリサブシステム
RAM
ROM
2608
2618
2620
バスサブシステム
ネットワークインタフェース
2616
記憶サブシステム
ファイル記憶サブシステム
2610
プロセッサ
2602
2604

フロントページの続き

内
(72)発明者　ドワロン，ブロック ジルズ
イギリス国 シーエフ１０ ２エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス ７ス フロア アーカート－ダイクス アンド ロード エルエルピー 内
審査官　中里 裕正
(56)参考文献　特開平０８－００８８９５（ＪＰ，Ａ）
特開２００５－２２９１７８（ＪＰ，Ａ）
米国特許出願公開第２０１０／００２７７８３（ＵＳ，Ａ１）
米国特許出願公開第２０１６／０３０６９８２（ＵＳ，Ａ１）
国際公開第２０１８／２３５８４５（WO，Ａ１）
国際公開第２０１９／０３８６３４（WO，Ａ１）
独立行政法人情報通信研究機構，独立行政法人情報処理推進機構，2009年度版リストガイド，[online]，2010年05月28日，pp.43-66，〈URL:https://www.cryptrec.go.jp/tech_reports_html〉，〈URL:https://www.cryptrec.go.jp/tech_reports_html〉
ROTH, U. and DJOUA, T. N.， Message exchange on base of a blockchain-based layered architecture， it - Information Technology，[online]，2018年10月06日，Vol.60 No.5-6，pp.253-261，ト〈URL:https://www.degruyter.com/document/doi/10.1515/itit-2017-0037/html〉，[2023年12月14日検索]
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／０８
Ｈ０４Ｌ　９／３２