(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7371015号
(P7371015)

(45)発行日　令和5年10月30日(2023.10.30)

(24)登録日　令和5年10月20日(2023.10.20)

(51)国際特許分類

		FI		
H04L 9/32 (2006.01)		H04L	9/32	200Z
H04L 9/08 (2006.01)		H04L	9/08	C
		H04L	9/32	200B

請求項の数　9　（全27頁）

(21)出願番号　特願2020-562171(P2020-562171)
(86)(22)出願日　令和1年5月8日(2019.5.8)
(65)公表番号　特表2021-523609(P2021-523609A)
(43)公表日　令和3年9月2日(2021.9.2)
(86)国際出願番号　PCT/IB2019/053771
(87)国際公開番号　WO2019/220270
(87)国際公開日　令和1年11月21日(2019.11.21)
審査請求日　令和4年4月11日(2022.4.11)
(31)優先権主張番号　1807813.9
(32)優先日　平成30年5月14日(2018.5.14)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　PCT/IB2018/053347
(32)優先日　平成30年5月14日(2018.5.14)
最終頁に続く

(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】　ブロックチェーンを使って原子的スワップを実行するためのコンピュータ実装されるシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
コンピュータ実装される交換方法であって、当該方法は：
（ｉ）第1のユーザーに関連する装置によって、前記第1のユーザーから第2のユーザーに第1のベールに包まれた秘密値を、前記第2のユーザーに関連する装置によって、前記第2のユーザーから前記第1のユーザーに第2のベールに包まれた秘密値を通信する段階と；
（ｉｉ）前記第1のユーザーに関連する装置および／または前記第2のユーザーに関連する装置によって、第1のブロックチェーン・トランザクションおよび／または第2のブロックチェーン・トランザクションを構築する段階であって、前記第1のブロックチェーン・トランザクションおよび前記第2のブロックチェーン・トランザクションはそれぞれ前記第1のベールに包まれた秘密値および前記第2のベールに包まれた秘密値を含み、前記第1のブロックチェーン・トランザクションおよび前記第2のブロックチェーン・トランザクションは、第1の秘密値および第2の秘密値の両方がそれぞれのブロックチェーン・トランザクションに提供されると、それぞれの第1または第2の資源の制御を移転するためにロック解除可能となるように構成される、段階とを含み、
前記第1のブロックチェーン・トランザクションのロック解除によって前記第1の秘密値が前記第2のユーザーに明かされ、前記第2のブロックチェーン・トランザクションのロック解除（48B）によって前記第2の秘密値が前記第1のユーザーに明かされる、
方法。
【請求項２】

前記第1のブロックチェーン・トランザクションおよび前記第2のブロックチェーン・トランザクションの少なくとも一方は、それぞれの第1の秘密鍵および第2の秘密鍵の適用の際にのみ、償還可能であるように構成される、請求項１に記載の方法。

【請求項３】

前記第1のユーザーに関連する装置および／または前記第2のユーザーに関連する装置によって、(a)少なくとも部分的には前記第1のユーザーの第1の公開鍵に基づく第1の派生公開鍵および(b)少なくとも部分的には前記第2のユーザーの第2の公開鍵に基づく第2の派生公開鍵のうちの少なくとも一方を計算する段階をさらに含み、前記第1の派生公開鍵は、前記第1の秘密鍵を含む暗号鍵ペアの一部であり、前記第2の派生公開鍵は、前記第2の秘密鍵を含む暗号鍵ペアの一部である、請求項２に記載の方法。

【請求項４】

(a)少なくとも部分的には前記第1のユーザーの第1の公開鍵に基づく第1の派生公開鍵および(b)少なくとも部分的には前記第2のユーザーの第2の公開鍵に基づく第2の派生公開鍵のうちの少なくとも一方を計算する段階はさらに、前記第1および第2のベールに包まれた秘密値の組み合わせを含む、請求項３に記載の方法。

【請求項５】

前記第1および第2のベールに包まれた秘密値の組み合わせは、前記第1のベールに包まれた秘密値と前記第2のベールに包まれた秘密値との連結、および少なくとも1つのベールに包まれた秘密値の、ランダムまたは擬似ランダム値との連結の少なくとも一方を含む、請求項４に記載の方法。

【請求項６】

前記第1のユーザーに関連する装置および／または前記第2のユーザーに関連する装置によって、前記第1のブロックチェーン・トランザクションが償還されない第1の時間期間の経過に応答して、前記第1の資源の制御を前記第1のユーザーに返すように構成された第3のブロックチェーン・トランザクション；および前記第2のブロックチェーン・トランザクションが償還されない第2の時間期間の経過に応答して、前記第2の資源の制御を前記第2のユーザーに返すように構成された第4のブロックチェーン・トランザクション、のうちの少なくとも一方を構築する段階をさらに含む、請求項１ないし５のうちいずれか一項に記載の方法。

【請求項７】

前記第1のベールに包まれた秘密値および前記第2のベールに包まれた秘密値の少なくとも一方は、前記第1の秘密値および前記第2の秘密値のうち少なくとも一方と、前記第1のユーザーおよび前記第2のユーザーの両方によってアクセス可能な共有秘密値との組み合わせ（32）を含む、請求項１ないし６のうちいずれか一項に記載の方法。

【請求項８】

前記共有秘密値は、段階(i)の前に、共通の秘密（CS）として確立される、請求項７に記載の方法。

【請求項９】

（ｉｉｉ）前記第1のユーザーに関連する装置によって、前記第1の秘密値から始まるベールに包まれた秘密値の少なくとも1つのシーケンスを生成する、および／または前記第2のユーザーに関連する装置によって、前記第2の秘密値から始まるベールに包まれた秘密値の少なくとも1つのシーケンスを生成する段階と；

（ｉｖ）前記第1のユーザーに関連する装置および／または前記第2のユーザーに関連する装置によって、少なくとも1つのブロックチェーン・トランザクションを償還して、前記第1の秘密値および前記第2の秘密値のうち少なくとも一方を明かし、それにより、前記シーケンスの少なくとも1つのベールに包まれた秘密値を明かす段階とをさらに含む、
請求項１ないし８のうちいずれか一項に記載の方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、概括的には、コンピュータで実装されるセキュリティ方法および暗号技法に関する。より詳細には、本発明は、資源の制御を原子的に交換するための方法に関する。本発明は、限定されるものではないが、一つまたは複数のブロックチェーンおよび関連するプロトコル上での使用に特に適している。

【背景技術】

【０００２】

本稿では、「ブロックチェーン」という用語を、電子的、コンピュータベース、分散型のあらゆる形態の台帳を含むものとして使う。これらは、コンセンサスに基づくブロックチェーンおよびトランザクションチェーン技術、許可式および非許可式の台帳、共有される台帳、およびそれらの変形を含む。ブロックチェーン技術の最も広く知られている応用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が提案され、開発されている。本明細書では、便宜上、説明のためにビットコインに言及することがあるが、本発明は、ビットコイン・ブロックチェーンと一緒に使用することに限定されず、代替的なブロックチェーン実装およびプロトコルが、本発明の範囲内にはいることに注意しておくべきである。用語「ユーザー」は、本明細書中では、人間またはプロセッサ・ベースの資源を指しうる。また、本明細書中で使用される用語「ビットコイン」は、ビットコイン・プロトコルから派生するプロトコルまたは実装のすべてのバージョンおよび変形を含むことが意図されている。

【０００３】

ブロックチェーンは、ピアツーピアの電子台帳であり、これは、ブロックから構成される、コンピュータベースの脱中心化された分散型のシステムとして実装され、ブロックはトランザクションから構成される。各トランザクションは、ブロックチェーン・システムにおける参加者間のデジタル資産または資源の制御〔コントロール〕の移転をエンコードするデータ構造であり、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。各ブロックは、前のブロックのハッシュを含み、それにより諸ブロックがチェーン化されて、ブロックチェーンの端緒以来、そのブロックチェーンに書き込まれたすべてのトランザクションの永続的で変更不可能なレコードを作成する。トランザクションは、その入力および出力に埋め込まれた、スクリプトとして知られる小さなプログラムを含んでおり、それが、トランザクションの出力に誰がどのようにアクセスできるかを指定する。ビットコイン・プラットフォームでは、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。

【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、トランザクションは「有効確認」される必要がある。ネットワーク・ノード（採鉱者）が、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し、無効なトランザクションはネットワークから拒否される。ノードにインストールされているソフトウェア・クライアントが、そのロック・スクリプトおよびアンロック・スクリプトを実行することによって、未使用トランザクション（unspent transaction、UTXO）に対してこの有効確認（validation）作業を実行する。ロックおよびアンロック・スクリプトの実行が真と評価される場合、そのトランザクションは有効であり、そのトランザクションはブロックチェーンに書き込まれる。このように、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、トランザクションは、i)そのトランザクションを受け取る最初のノードによって有効確認されなければならない。もし、トランザクションが有効確認されれば、ノードはそれをネットワーク内の他のノードに中継する。トランザクションは、ii)採鉱者によって構築された新しいブロックに追加され、iii)採掘されなければならない。すなわち、過去のトランザクションの公開台帳に追加される。

【０００５】

ブロックチェーン技術は、暗号通貨実装の使用について最も広く知られているが、デジタル起業家は、ビットコインのベースとなる暗号セキュリティ・システムと、ブロックチェーンに格納できるデータの両方を使用して、新しいシステムを実装することを検討し始

めている。ブロックチェーンが、暗号通貨の領域に限定されない自動化されたタスクおよびプロセスに使用できれば、非常に有利であろう。そのような解決策は、応用がより多様でありながら、ブロックチェーンの恩恵（たとえば、事象の恒久的な、改竄不能な記録、分散処理など）を利用することができるであろう。

【０００６】

原子的スワップ〔アトミック・スワップ〕の概念は、以前から暗号通貨コミュニティで議論されてきた。当事者間の交換は、すべての参加者が所望の資源（たとえば、暗号通貨のトークンまたはコイン）を受け取るか、または誰も受け取らないという意味で「原子的」である。執筆時点で、ウィキペディアは、原子的スワップを「信頼できる第三者を必要とせずに、ある暗号通貨を別の暗号通貨と交換することを可能にする、暗号通貨における提案された機能」と説明している。伝統的な暗号通貨では、当事者が対価として通貨を受け取ることなく通貨を送ってしまうことを防止するためには、暗号通貨のスワップを実行するために暗号通貨交換所のような信頼できる第三者が必要になる。原子的スワップ・システムは、ハッシュ時間ロック・スマート・コントラクトを使用して、当事者はスワップされる通貨を指定された時間内に引き渡さなければならず、さもなければトランザクションはキャンセルされる。これは、スワップが行なわれるか、または通貨が全くスワップされないかのいずれかである」という意味で「原子性」を保存する。

【０００７】

このように、原子的スワップは、ブロックチェーンを通じて実施される移転に関して、向上したセキュリティを提供する。信頼される第三者の必要性をなくすことは、悪用や悪意ある介入のリスクをなくすからである。現に、マウントゴックスの一件のように、暗号通貨交換所に関して、いくつかのセキュリティ破りまたは「ハッキング」が行なわれてきた。

【先行技術文献】

【非特許文献】

【０００８】

【文献】https://en.wikipedia.org/wiki/Atomic_swap

【発明の概要】

【発明が解決しようとする課題】

【０００９】

しかしながら、提案された原子的スワップ解決策は、ただ1つの秘密の使用を含み、スワップは非同期的に実行される。これは、あるトランザクションが消費された後になってはじめて、他のトランザクションが消費できるという欠点を生み出す。

【００１０】

このように、ブロックチェーン技術によって提供される信頼不要性および不変性を有する資源または資産を原子的に交換し、ブロックチェーン実装ネットワークを通じて実施される移転に関するセキュリティを向上させる、暗号学的に実施される資源交換方法を提供することが望ましい。

【００１１】

そのような改善された解決策が今、考案された。

【課題を解決するための手段】

【００１２】

本発明によれば、コンピュータ実装された交換、取り換え、または移転方法が提供される。追加的または代替的な定義によれば、本発明は、資源が送信者から受信者にネットワークを横断していつ伝送されてもよく、またはいつ伝送されなくてもよいかを制御するセキュリティ方法を提供する。追加的または代替的に、本発明は、ブロックチェーンを介した資源の原子的交換または伝送を実行するように構成された方法および対応するシステムを提供する。

【００１３】

本方法は、以下のステップを含んでいてもよい：

（ｉ）第1のユーザーから第2のユーザーに第1のベールに包まれた秘密値を、第2のユーザーから第1のユーザーに第2のベールに包まれた秘密値を通信すること；および
（ｉｉ）それぞれ前記第1のベールに包まれた秘密値および前記第2のベールに包まれた秘密値を含む第1のブロックチェーン・トランザクションおよび第2のブロックチェーン・トランザクションを構築することであって、それらのトランザクションは、第1の秘密値および第2の秘密値の両方がそれぞれのブロックチェーン・トランザクションに提供されると、それぞれの第1または第2の資源の制御を移転するためにロック解除可能（unlockable）であるように構成されること、
ここで、前記第1のブロックチェーン・トランザクションのロック解除によって前記第1の秘密値が前記第2のユーザーに明かされ、前記第2のブロックチェーン・トランザクションのロック解除によって前記第2の秘密値が前記第1のユーザーに明かされる。

【００１４】

本方法は、少なくとも2つのトランザクション（Tx1およびTx2）を含み、各トランザクションは、それぞれの出力に関連する複数のパズルまたはスクリプトへの要求される基準が提供される際にのみロック解除されることができる少なくとも1つの未使用出力（unspent output、UTXO）を有する、ことによって、原子的な交換機構を提供しうる。言い換えると、第1のトランザクションにおける未使用出力についてのロックおよびアンロック基準は、他方のまたはある別のトランザクションにおける未使用出力についてのロックの基準と同じであってもよく、それによってミラーされてもよい。
第1のトランザクションにおける未使用出力への要求されるロック解除基準の提供は、他方のまたはある別のトランザクションにおける未使用出力をロック解除するために要求される一つまたは複数の秘密値を明かしてもよい、または、該秘密値をアクセス可能にしてもよい。

【００１５】

秘密値を含むロック解除スクリプトは、第1のトランザクションまたは第2のトランザクションにおける出力を消費するその後のトランザクションの入力において提供されてもよい。ひとたび後続トランザクションのロック解除スクリプトが、第1または第2のトランザクションのロック・スクリプトと一緒に実行されると、後続トランザクションは、有効確認され、その後、ブロックチェーン上で公表されうる。これにより、後続トランザクションの入力において提供される一つまたは複数の秘密値は、ブロックチェーンからアクセス可能または読み取り可能となる。

【００１６】

この方法は、秘密値が信頼のない環境で原子的に交換されることを確実にする安全な仕方を提供し、この方法のどのユーザーも、他のユーザーよりもこの方法に対してより大きな制御をもつことはない。

【００１７】

ベールに包まれた秘密値から秘密値を決定するのは現実的に可能ではないが、秘密値からベールに包まれた秘密値を決定するのは現実的に可能であるという仕方で、秘密値は、対応するベールに包まれた秘密値に関係する。この関係の例として、ハッシュやモジュロ算術のような一方向性関数を秘密値に適用することで、ベールに包まれた秘密値が得られる。このように、1つの定義によれば、ベールに包まれた（秘密の）値は、もとの（秘密の）値から導出できる、または導出されたが、もとの（秘密の）値を決定するために使用することはできない値でありうる。もとの値を提供するためにリバース・エンジニアリングすることは現実的に可能ではないかもしれない。

【００１８】

語句「トランザクションのロック解除」は、トランザクションにおいて提供される少なくとも1つの未使用出力（UTXO）をロック解除するまたは消費することの意味を含みうる。これは、未使用出力に関連するロック・スクリプトを満たすために、要求されるデータ／ロック解除スクリプトを提供することによって達成されうる。

【００１９】

第1のトランザクションおよび第2のトランザクションの少なくとも1つは、それぞれの第1の秘密鍵および第2の秘密鍵の適用または提供の際にのみ、償還可能（支出可能）であるように構成されてもよい。これは、秘密鍵によって示される意図された受信者のみがトランザクションをロック解除できるという利点を提供する。

【００２０】

本方法は：(a)少なくとも部分的には前記第1のユーザーの第1の公開鍵に基づく第1の派生公開鍵および(b)少なくとも部分的には前記第2のユーザーの第2の公開鍵に基づく第2の派生公開鍵のうちの少なくとも一方を計算するステップをさらに含んでいてもよい。前記第1の派生公開鍵は、前記第1の秘密鍵を含む暗号鍵ペアの一部であり、前記第2の派生公開鍵は、前記第2の秘密鍵を含む暗号鍵ペアの一部である。

【００２１】

これにより、資産または資源を、公知なアドレスではなく派生したアドレスに格納することができ、本方法のユーザーに追加的なプライバシーおよびセキュリティを提供する。「資産」および「資源」という用語は、本明細書において交換可能に使用されうることに注意しておくべきである。「資産」という用語は、財務上の文脈または用途を有するもののみとして解釈されるべきではない。資産は、たとえば、ブロックチェーン上またはブロックチェーン外の何らかの他のエンティティを表わすトークンでありうる。

【００２２】

(a)少なくとも部分的には前記第1のユーザーの第1の公開鍵に基づく第1の派生公開鍵および(b)少なくとも部分的には前記第2のユーザーの第2の公開鍵に基づく第2の派生公開鍵のうちの少なくとも一方を計算するステップはさらに、第1および第2のベールに包まれた秘密値の組み合わせを含んでいてもよい。

これは、トランザクションと実行される原子的スワップとの間の、記録された、消去できないリンクを提供するという利点を提供する。

【００２３】

第1および第2のベールに包まれた秘密値の組み合わせは、第1のベールに包まれた秘密値と第2のベールに包まれた秘密値との連結、および少なくとも1つのベールに包まれた秘密値の、ランダムまたは擬似ランダム値との連結の少なくとも一方を含んでいてもよい。

これは、追加的な決定論的な難読化を介して、トランザクションのセキュリティをさらに高めるという利点を提供する。

【００２４】

本方法は、さらに、第1のトランザクションが償還されない第1の時間期間の経過に応答して、第1の資源の制御を第1のユーザーに返すように構成された第3のブロックチェーン・トランザクション；および第2のトランザクションが償還されない第2の時間期間の経過に応答して、第2の資源の制御を第2のユーザーに返すように構成された第4のブロックチェーン・トランザクション、のうちの少なくとも一方を構築するステップを含んでいてもよい。

これにより、別のユーザーが交換に完全には参加しない場合に、本方法の少なくとも一のユーザーが、それぞれの資源の制御を返してもらうことが可能となり、それにより、本方法の汎用性が増す。

【００２５】

第1のベールに包まれた秘密値および第2のベールに包まれた秘密値の少なくとも一方は、第1の秘密値および第2の秘密値のうち少なくとも一方と、第1のユーザーおよび第2のユーザーの両方によってアクセス可能な共有秘密値との組み合わせを含んでいてもよい。

これは、本方法によって提供されるプライバシーおよびセキュリティを増すという利点を提供する。

【００２６】

共有秘密値は、ステップ(i)の前に、共通の秘密として確立されてもよい。

これは、本方法のセキュリティをさらに高めるという利点を提供する。

【００２７】

本方法は、以下のステップをさらに含んでいてもよい：
（ｉｉｉ）第1の秘密値および第2の秘密値のうち少なくとも一方から始まるベールに包まれた秘密値の少なくとも1つのシーケンスを生成すること；
（ｉｖ）第1の秘密値および第2の秘密値のうち少なくとも一方を使用して、先行する請求項のいずれかの方法を実行すること；
（ｖ）少なくとも1つのブロックチェーン・トランザクションを償還して、第1の秘密値および第2の秘密値のうち少なくとも一方を明かし、それにより、前記シーケンスの少なくとも1つのベールに包まれた秘密値を明かすこと。
【００２８】
これは、秘密を格納するために必要とされるストレージ・スペースが少なくなるので、本方法の単純な繰り返しよりも高い効率で、一連の安全な原子的交換を実行することを可能にする。さらに、通信のラウンド数が少なくなる。これにより時間が節約され、セキュリティが改善される。
【００２９】
本方法の少なくともステップ(ii)を実行するステップは、本方法のステップ(v)で開示された少なくとも1つのベールに包まれた秘密値を使用してもよい。
これは、本方法の効率をさらに向上させるという利点を提供する。
【００３０】
本発明のこれらおよび他の側面は、本明細書に記載される実施形態から明白となり、それを参照して解明される。
【００３１】
ここで、添付の図面を参照して、本発明の実施形態を、限定的意味ではなく単に例として、説明する。
【図面の簡単な説明】
【００３２】
【図１】本発明を具現する方法において取られるステップを示すフローチャートを示す。
【図２】機密性の高い情報を安全に伝送するために本発明に従って使用されうる、第1のノードおよび第2のノードについての共通の秘密を決定するための例示的なシステムの概略図である。
【図３】機密性の高い情報を安全に伝送するために本発明に従って使用されうる、共通の秘密を決定するためのコンピュータ実装される方法のフローチャートである。
【図４】第1および第2のノードを登録するためのコンピュータ実装される方法のフローチャートである。
【図５】機密性の高い情報を安全に伝送するために本発明に従って使用されうる、共通の秘密を決定するためのコンピュータ実装される方法の別のフローチャートである。
【発明を実施するための形態】
【００３３】
ブロックチェーン上の原子的トランザクション交換は、第1のユーザー、アリスから第2のユーザー、ボブへの1つのトランザクションとボブからアリスへの別のトランザクションとの2つのトランザクションについて、両方のトランザクションが完了するか、さもなければどちらも完了しないことを意味する。
【００３４】
図1を参照するに、本発明は、アリスおよびボブがそれぞれ、それぞれA_0およびB_0と記される秘密を生成（30）できるようにすることを含む。もしアリスとボブが信頼できるなら、本ブロックチェーン・プロトコルの一部ではない通信チャネルを使って、これらの秘密を含む情報を交換できる。二人は、小見出し「共通の秘密の決定」の下に後述される安全な秘密交換を使用してもよい。
【００３５】
一方の当事者が信頼できず、自分の秘密を共有しないとする。本発明により、この当事者が自分の資金を消費する唯一の方法は、ブロックチェーン上で自分の秘密を明かし、そ

れにより秘密を公知にし、他のユーザーが利用できるようにすることである。これは、交換において使用されるトランザクションの構成によるものである。したがって、本方法は、いずれの当事者にも、相手方当事者を信頼することを要求しない。

【００３６】

本発明のある実施形態では、2つの秘密がある：1つはアリスによって生成され、アリスにとってアクセス可能であり、もう1つはボブによって生成され、ボブにとってアクセス可能である。これらはブロックチェーン外のチャネルを通じて通信される。

【００３７】

単一の原子的スワップ

P_{A0}が対応する秘密鍵S_{A0}をもつアリスの楕円曲線デジタル署名アルゴリズム（elliptic curve digital signature algorithm、ECDSA）公開鍵を示し、P_{B0}が秘密鍵S_{B0}をもつボブの公開鍵を示すとする。

【００３８】

１．30において、アリスは自分だけが知っている秘密

【数１】

$$A_0 \in \mathbb{Z}_n^*$$

を選び、ボブは自分だけが知っている秘密

【数２】

$$B_0 \in \mathbb{Z}_n^*$$

を選ぶ。（これらの秘密は、アリスおよびボブの公開鍵または秘密鍵とは関係していない。）ここで、nは楕円曲線生成点Gの位数である。秘密は、SHA256(mod n)アルゴリズムを通された一般的なデータ構造の形であってもよい。

【００３９】

２．アリスとボブが、両者間の通信チャネルを開く。これは、下記の小見出し「共通の秘密の決定」の下で後述される方法を使って生成される安全な通信チャネルであってもよい。次いで二人は、それぞれの秘密をハッシュし（ステップ34）、公開鍵とそれぞれの秘密のハッシュとを共有する（ステップ36）。A_0およびB_0のハッシュ値は$H(A_0)$および$H(B_0)$と記され、ここで、SHA-256のような標準的なハッシュ関数が使用されてもよい。値$H(A_0)$および$H(B_0)$は、公にも共有されうる。アリスとボブは、二人とも、

P_{A0}、P_{B0}、$H(A_0)$、$H(B_0)$

を知っている。

【００４０】

３．38で、アリスとボブは決定論的鍵

$H(A_0) \mid H(B_0)$

（ここで、"|"はOP_CAT演算を示す）または代替的には

$H(H(A_0) \mid H(B_0))$

のような派生ハッシュを計算する。

【００４１】

４．40で、アリスとボブはここで派生公開鍵

【数３】

$$\text{アリス: } P_{A_1} = P_{A_0} + (H(A_0) \mid H(B_0)) \cdot G$$

$$\text{ボブ: } \quad P_{B_1} = P_{B_0} + (H(A_0) \mid H(B_0)) \cdot G.$$

を生成する。これらは次の対応する秘密鍵をもつ。

【数４】

$$\text{アリス: } S_{A_1} = S_{A_0} + H(A_0) \mid H(B_0)$$

$$\text{ボブ: } \quad S_{B_1} = S_{B_0} + H(A_0) \mid H(B_0).$$

アリスとボブは派生公開鍵P_{A1}、P_{B1}を使って原子的スワップを実行する。原理的には、二人はもとの公開鍵P_{A0}、P_{B0}を使うことができるのだが、派生公開鍵は、原子的スワップに結びつけられており、アリスとボブが簡単に計算できるが、（$H(A_0)$および$H(B_0)$が公開されない限り）他の誰にとっても簡単に計算できないという利点がある。

【００４２】

38において、

$H(A_0) \mid H(B_0) \mid Z$

のように決定論的な疑似ランダムに見える値も組み込まれれば、追加的なプライバシーが達成されうる。ここで、Ｚは、共有された開始値に基づいて、事前に合意されたゼータ関数のように、両方の当事者が計算できるものである。

【００４３】

５．42で、アリスとボブは次のロック・スクリプトを構築する。ここではスクリプトは概略的に記述され、ビットコイン・スクリプトにおける例示的な実装は後に示す。

【数５】

$$LockingScript(A) = CheckSig\ H(P_{A_1})\ \ AND\ \ Solve\ H(A_0)\ \ AND\ \ Solve\ H(B_0)$$

$$LockingScript(B) = CheckSig\ H(P_{B_1})\ \ AND\ \ Solve\ H(A_0)\ \ AND\ \ Solve\ H(B_0)$$

プロセス$CheckSigH(P_{A1})$は、公開鍵／秘密鍵ペアP_{A1}、S_{A1}のための標準的なECDSA署名有効確認演算である。その代わりに、公開鍵／秘密鍵ペアP_{A0}、S_{A0}のための標準的なECDSA署名有効確認である$CheckSigH(P_{A0})$が使われてもよい。プロセス$SolveH(A_0)$は、解A_0をもつハッシュ・パズルである。つまり、ロック解除スクリプトは、ハッシュされたときにロック・スクリプトにおいて与えられている$H(A_0)$と等しい有効な値A_0を含まなければならない。ロック解除スクリプトは、次のように与えられる。

【数６】

$$UnlockingScript(A) = [B_0][A_0][Sig\ P_{A_1}][P_{A_1}]$$

$$UnlockingScript(B) = [B_0][A_0][Sig\ P_{B_1}][P_{B_1}]\ .$$

ここでわかるように、もしアリスかボブのどちらかが自分の資金をロック解除すれば、必然的に値A_0およびB_0をブロックチェーン上で曝露することになる。

【００４４】

６．42では、アリスはロック・スクリプトLockingScript(B)を用いてP_{B1}へのトランザクションtx_1を生成し、ボブはロック・スクリプトLockingScript(A)を用いてP_{A1}へのトランザクションtx_2を作成する。この段階では、アリスもボブもP_{A1}およびP_{B1}における資金を使うことはできない。なぜなら、どちらの当事者もA_0およびB_0の両方は知らないからである。これらのトランザクションはネットワークに送信され、その後ブロックチェーン上に現われる。

【００４５】

７．46Ｃでは、アリスはボブに秘密A_0を送り、ボブはアリスに秘密B_0を送る。これは、上記で確立されたアリスとボブ間の通信チャネルを使って実行される。アリスとボブは、これらのハッシュ値が$H(A_0)$および$H(B_0)$に等しいことを確認することによって、これ

らが正しい値であることを検査しうる。

【００４６】

８．アリスとボブがどちらも正直であり、正しい秘密を共有しているとすると、両方の当事者は両方の秘密を知り（ステップ48C）、どちらもP_{A1}およびP_{B1}においてロックされた資金を消費でき（ステップ50C）、原子的スワップは完了する。

【００４７】

９．たとえば、ボブが自分の正しい秘密B_0をアリスに送らなかったとしよう。つまり、アリスだけが自分の秘密を送り、ステップ46Cではなく46Bが生起する。ロック・スクリプトLockingScript(B)の形のため、ボブがP_{B1}でロックされた資金を使うためには、ボブはロック解除スクリプトにおいて自分の秘密B_0を公開しなければならない。結果として、ボブが資金を使うとすぐに、アリスはボブの秘密を知り（ステップ48B）、よってP_{A1}における自分の資金を使うことができるようになる（ステップ50B）。これにより、アリスとボブのどちらも自分の資金を使うことができるか、どちらも資金を使うことができないかのどちらかであることが保証される。

【００４８】

下記は、ビットコイン・ブロックチェーンと互換性がある、上記のステップ４におけるアリスについての例示的なロックおよびアンロック・スクリプトである。

アリスについてのロック・スクリプト：

OP_DUP OP_HASH160〈Hash160 P_{A1}〉OP_EQUALVERIFY OP_CHECKSIG OP_HASH256〈Hash256 A_0〉OP_EQUALVERIFY OP_HASH256〈Hash256 B_0〉OP_EQUALVERIFY

アリスについてのロック解除スクリプト：

〈B_0〉 〈A_0〉 〈SigP_{A1}〉 〈P_{A1}〉

公開鍵ハッシュへの支払い（Pay To Public Key Hash、P2PKH）アドレスおよびスクリプト・ハッシュへの支払い（Pay To Script Hash、P2SH）アドレスへのトランザクションはどちらも、上記のタイプのロック・スクリプトおよびロック解除スクリプトを許容することを注意しておく。P2SHアドレスについては、ロック・スクリプトは、同じ情報を含む償還スクリプトのハッシュとして提示される。

【００４９】

上記の方法は、ビットコイン・ブロックチェーンで使用されるECSDAに類似した公開／秘密鍵暗号化システムを使用するブロックチェーンを参照して記述されている。しかしながら、本方法は、ロック解除スクリプトにおいて一般的な形の秘密（これは任意のデータ構造でありうる）が開示されることを要求する、一般的な暗号化機構に一般化できる。必要とされるのは、ロック・スクリプト、トランザクション、およびブロックチェーンであり、これは安全で検証可能な通信チャネルである。

【００５０】

<u>時間ロック還付トランザクション</u>

もしボブがアリスに自分の正しい秘密B_0を与えることを拒否し、アドレスP_{B1}に格納されている資金をロック解除することもしない場合、ボブの秘密はアリスに明かされず、アリスはP_{A1}に格納されている資金をロック解除することはできない。さらに、アリスはボブに送った、P_{B1}に格納されている資金を回収することもできない。

【００５１】

この問題は、資金が使われていない場合は、一定時間後に資金を送り返すように構成された、ボブからアリスへの新しいトランザクションを導入することで解決できる。これは、LockingScript(A)およびLockingScript(B)を若干修正することをも必要とし、この修正は後述する。

【００５２】

この新しいトランザクションは、ある事前に指定された時間が経過した後にのみ、トランザクションがブロックによって受け入れられることを許容する、ロック・スクリプトにおける時間依存動作を利用する。たとえば、ビットコイン・スクリプトでは、これは、指

定された値以来の相対的な時間についてはチェック・シーケンス検証（Check Sequence Verify、CSV） 動作、または固定した時間値についてはチェック・ロック時間検証（Check Lock Time Verify、CLTV）動作であってもよい。

【００５３】

上記のステップ４のロック・スクリプトは、アリスとボブの両方が署名することに同意する場合に消費するオプションを含むように、次のように修正される：

【数７】

$$LockingScript'(A) = \begin{cases} CheckSig\,H(P_{A_1})\ \ AND\ \ Solve\,H(A_0)\ \ AND\ \ Solve\,H(B_0) \\ OR \\ CheckSig\,H(P_{A_1})\ \ AND\ \ CheckSig\,H(P_{B_1}) \end{cases}$$

$$LockingScript'(B) = \begin{cases} CheckSig\,H(P_{B_1})\ \ AND\ \ Solve\,H(A_0)\ \ AND\ \ Solve\,H(B_0) \\ OR \\ CheckSig\,H(P_{A_1})\ \ AND\ \ CheckSig\,H(P_{B_1}) \end{cases}$$

【００５４】

44では、上記の方法のステップ４の後、ステップ５の前に、2つの新規トランザクションが生成される。アリスは、ボブの資金全部を返す、P_{A1}からボブへのトランザクションtx_4を生成する。このトランザクションは、一定の時間（たとえば、24時間）後にのみブロック内で受け入れられるよう時間ロック〔タイムロック〕される。ボブはP_{B1}からアリスへの同様のトランザクションtx_3を生成する。トランザクションtx_3およびtx_4はそれぞれのロック・スクリプト

【数８】

$$LockingScript2(A) = CheckSig\,H(P_{A_1})\ \ AND\ \ CSV(24\ hours)$$

$$LockingScript2(B) = CheckSig\,H(P_{B_1})\ \ \text{AND}\ \ CSV(24\ hours)$$

をもつ。

【００５５】

アリスはtx_4に署名し、それをボブに送信し、ボブは署名してネットワークに送信する。同様に、ボブtx_3は署名し、それをアリスに送信し、アリスは署名してネットワークに送る。

【００５６】

この段階で、いずれの当事者も従わない場合は、プロセスは中止され、資金は移転されない。両方の当事者が従っている場合は、上記の方法のステップ5が実行される（42）。ここで、どちらの当事者も原子的スワップ（46A）において交換された資金を使わなければ、資金は24時間後にもとの所有者に返還される（48A、50A）。

【００５７】

ここで、24時間というCSVの相対時間が例として使用されたが、将来の任意の相対時間、または将来の任意の特定の時間を（たとえば、CLTV演算子を使用して）使用することが可能であろう。

【００５８】

ビットコイン・ブロックチェーンを使用して24時間後にアリスに資金を返却するロック・スクリプトの例は次のとおり。

"24h" OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160〈Hash160 P_{A1}〉OP_EQUALVERIFY OP_CHECKSIG

対応するアンロック・スクリプトは、〈Sig P_{A1}〉 〈P_{A1}〉によって与えられる。

【００５９】

秘密値のマスキング

さらなる代替的な実施形態は、値A_0およびB_0がアリスおよびボブのみに知られ、決して公開されないように、マスキング・ステップ32を含む。

【００６０】

最初、アリスとボブの両名は、二人だけが知っている共有される秘密S_Cについて合意する。これは、後述する「共通の秘密の決定」という小見出しで述べられる方法を用いた、秘密の安全な交換を通じて達成できる。

【００６１】

次いで、アリスとボブは新しい秘密を定義する。

【００６２】

$A'_0 = A_0 + S_C$

$B'_0 = B_0 + S_C$

次いで、二人は上記で概説した方法のように進めるが、もとの秘密ではなく、マスクされた秘密A'_0、B'_0を用いる。原子的スワップの間は、マスクされた秘密だけがブロックチェーン上で一般に公開される。

これは、秘密値A_0およびB_0が、のちに記載されるさらなる実施形態のような他のコンテキストにおいても使用される場合に有用である。

【００６３】

さらなる代替的な実施形態は、アリスとボブが一連のn回の原子的スワップを行なうことができるようにする。各当事者はランダムな秘密をもって開始し、アクセス・チェーンと呼ばれるこの秘密のハッシュ値のシーケンスを作成する。原子的スワップが実行されると、次の原子的スワップで使用される次の秘密のハッシュ値が開示される。このプロセスは、最大n回まで逐次反復可能に繰り返される。

【００６４】

この方法では、アリスとボブは一度に1つの秘密を格納するだけでよいので、秘密のための要求される格納スペースがより少なくなるという点で、個別のスワップの系列に比べ、効率が節約される。前の秘密をハッシュすることから次の秘密を計算することができる。秘密のハッシュを毎回通信する必要がないため、相互間の通信ラウンドが少なくてすむ。これは時間を節約し、セキュリティを改善する。

【００６５】

本方法は、以下の通りである：

アリスとボブが、反復交換の数nについて合意する。二人は、それぞれランダムな値A_nおよびB_nを生成する。アリスは次のアクセス・チェーンを計算する：

【数９】

$$A_n = random$$

$$A_{n-1} = hash\,(A_n)$$

$$A_{n-2} = hash\,(A_{n-1})$$

$$\vdots$$

$$A_{i-1} = hash\,(A_i)$$

$$\vdots$$

$$A_0 = hash\,(A_1)$$

ボブは、B_nから開始して等価なチェーンを計算する。これらのチェーンは、一連のスワップにおいて使用される秘密値に対応する。可能なスワップの数は、シーケンス{0,1,…,n}

内にある。すなわち、両当事者は、新しいチェーンを再初期化することが必要になる前に、0ないしn回のトランザクションのスワップのためにこれらの値を使用することができる。

【００６６】

これらのスワップの実施方法を下記で概説する。ボブが等価なプロセスをたどることを理解しておくべきである。

【００６７】

１．アリスは、自分のチェーン$A_0,A_1,\cdots,A_n$、ボブの公開鍵P_{B0}、ボブの秘密のハッシュ$H(B_0)$から始める。これまでと同様に、$H(B_0)$はボブによって公に共有されてもよい。

【００６８】

２．アリスが派生公開鍵

【数１０】

$$\text{アリス: } P_{A_1} = P_{A_0} + (H(A_0)|\, H(B_0)) \cdot G$$

$$\text{ボブ: } \quad P_{B_1} = P_{B_0} + (H(A_0)|\, H(B_0)) \cdot G,$$

を、次いでロック・スクリプト

【数１１】

$$LockingScript(A)_0 = CheckSig\ H(P_{A_1})\ \ AND\ \ Solve\ H(A_0)\ \ Solve\ H(B_0)$$

$$LockingScript(B)_0 = CheckSig\ H(P_{B_1})\ \ AND\ \ Solve\ H(A_0)\ \ \ Solve\ H(B_0).$$

を計算する。

先の実施形態で述べた時間依存の返金は、論理に実質的な変更を加えることなく、上記のロック・スクリプトに含めることができることを注意しておく。

【００６９】

３．アリスとボブが最初のスワップを実行する。前述したように、これはアリスとボブの間のA_0とB_0の交換に関わる。これは、スワップ後にアリスが$H(B_1)=B_0$を知っていることを意味する。

【００７０】

４．アリスは、この方法のステップ２を繰り返すが、チェーン内のボブの第2の秘密のハッシュ$H(B_1)$を用いる。明示的には、アリスは派生公開鍵

【数１２】

$$\text{アリス: } P_{A_2} = P_{A_1} + (H(A_1)|\, H(B_1)) \cdot G$$

$$\text{ボブ: } \quad P_{B_2} = P_{B_1} + (H(A_1)|\, H(B_1)) \cdot G,$$

およびロック・スクリプト

【数１３】

$$LockingScript(A)_1 = CheckSig\ H(P_{A_2})\ \ AND\ \ Solve\ H(A_1)\ \ Solve\ H(B_1)$$

$$LockingScript(B)_1 = CheckSig\ H(P_{B_2})\ \ AND\ \ Solve\ H(A_1)\ \ \ Solve\ H(B_1).$$

を計算する。

【００７１】

５．ひとたび第2のスワップが完了すると、アリスは$H(B_2)=B_1$を知っている。アリス

は、ボブの第3の秘密のハッシュ$H(B_2)$を使って再びステップ２を繰り返す。

【００７２】

６．このプロセスは、スワップが完了しないか、またはn回のスワップの最大数に達するまで逐次反復可能に繰り返される。

【００７３】

先の実施形態で述べたように、疑似ランダム値Z_iを演算$H(A_i) \| H(B_i) \| Z_i$に導入することによって、さらなるセキュリティを組み込むことができる。この場合、関数は、たとえば、ハッシュ関数$Z_{i-1} = H(Z_i)$を使用することによって、逐次反復毎に変換すべきである。

【００７４】

上記で概説した原子的スワップは、ビットコイン・ブロックチェーンに制約されない。上述の原子的スワップ方法における重要な構成要素は、一方の当事者がステップ７で資金を使うとき、ブロックチェーン上で自分の秘密を明かすということである。このことは、上記の方法は、ステップ４で与えられた形のロック・スクリプトおよびロック解除スクリプトを許容する任意のブロックチェーンに対して、原子的スワップを実行するために用いられうることを意味する。

【００７５】

さらに、原子的スワップ方法は、暗号通貨を交換するために使用されうる。たとえば、アリスがビットコイン（Bitcoin）・ブロックチェーン上でボブにビットコインを送信し、ボブがイーサリアム（Ethereum）・ブロックチェーン上でアリスにイーサリアムを送信するために使用されてもよい。

【表１】

	送信	受信
アリス	BCH	Eth
ボブ	Eth	BCH

【００７６】

2つの異なるブロックチェーン間の原子的スワップに関する唯一の制約は、それらのブロックチェーンが、ロック・スクリプトにおけるハッシュ・パズルにおいて同じハッシュ関数の使用を許容するということ（またはそれと等価なこと）である。その理由は次のとおり：アリスのブロックチェーンがSHA-256ハッシュ・アルゴリズムの使用を許容するのみであり、ボブのブロックチェーンがSHA-384アルゴリズムを許可するのみであるとする。ボブは、ある秘密のSHA-256ハッシュをアリスに送信するが、ボブのロック・スクリプトでは、異なる秘密のためにSHA-384ハッシュ・パズルを設定する。ボブが資金を使うとき、ロック解除スクリプトはアリスには役に立たない秘密を明かすことになり、ボブが資金を使い果たしてしまうまで、アリスにはこれを知るすべはない。

【００７７】

さらなる実施形態によれば、二当事者がそれぞれ公開鍵を生成できるようにする方法であって、該公開鍵について、対応する秘密鍵は、両方の当事者にとってアクセス可能にされるか、どちらの当事者にとってもアクセス可能にされないかのどちらかのみである、方法が提供される。この方法は、両当事者間で2つの秘密の値を交換するために上記の原子的スワップ方法を利用する。これらの秘密値は、秘密鍵を計算するために使用される。

【００７８】

この方法の1つの応用は、二当事者が、単一の公開鍵／秘密鍵ペアによって制御される複数のタイプの暗号通貨を交換することを許容することである。この方法は、アリスとボブがそれぞれ公開鍵を生成できるようにし、該公開鍵についての秘密鍵は、原子的スワップが行なわれるまで知られない。原子的スワップは、アリスとボブのどちらも対応する秘密鍵を計算することができるか、どちらも秘密鍵を計算できないかのどちらかであること

を保証する。

【００７９】

この方法は、たとえばビットコイン、イーサリアム、ダッシュ（Dash）で使用されるようなECSDAの秘密鍵と公開鍵のペアを使用して以下に記述される。しかしながら、この方法はECDSAプロトコルに決定的には依存せず、既存の秘密鍵と公知の決定論的鍵から新しい安全な公開鍵を決定論的に生成できる任意の公開鍵／秘密鍵ベースの暗号システムに容易に適応させることができる。

【００８０】

この方法は、新しい秘密鍵に関する部分的な情報が、オープンな台帳である一つまたは複数のブロックチェーンに保存されるという意味で、仮名的である。しかしながら、プロセスに関与する当事者だけがこの情報を復号することができ、セキュリティは決して損なわれない。

【００８１】

１．アリスは、対応する公開鍵$P_A = S_A \cdot G$をもつ秘密鍵S_Aと、自分だけが知っている秘密S_2から始める。ボブは、対応する公開鍵$P_B = S_B \cdot G$をもつ秘密鍵S_Bと、自分だけが知っている秘密S_1から始める。

【００８２】

２．アリスはボブに$P_2 = S_2 \cdot G$を送信し、ボブはアリスに$P_1 = S_1 \cdot G$を送信する。それらの秘密は楕円曲線の基底点を乗算されるので、このプロセスでは開示されず、P_2およびP_1は公知であってもよい。

【００８３】

３．アリスは、ビットコイン・トランザクション（またはアルトコインについての同様のもの）を受信するためのアドレスとして使用されうる新しい公開鍵$P_{AE} = P_A + P_1$を作成する。ボブは新しい公開鍵$P_{BE} = P_B + P_2$を作成する。

【００８４】

楕円曲線暗号の特性によれば、P_{AE}に対応する秘密鍵は$S_{AE} = S_A + S_1$であり、つまり$P_{AE} = S_{AE} \cdot G$である。P_{BE}に対応する秘密鍵は$S_{BE} = S_B + S_2$である。

【００８５】

この段階では、アリスはS_1を知らず、よってP_{AE}についての秘密鍵を知らない。ボブはS_1を知っているが、S_Aを知らず、よってP_{AE}についての秘密鍵を知らない。同じ論理により、アリスもボブもP_{BE}についての秘密鍵を知らない。

【００８６】

４．アリスはボブのアドレスP_{BE}へのトランザクションを行ない、ボブはアリスのアドレスP_{AE}へのトランザクションを行なう。これらのトランザクションは、公開鍵／秘密鍵システムを使用する何らかの暗号通貨の交換であってもよく、あるいはトークンまたはさらには物理的な資産を公開鍵P_{AE}およびP_{BE}の所有に移してもよい。上記の組み合わせであってもよい。

【００８７】

５．アリスとボブは今や、S_2およびS_1をそれぞれの秘密として、任意のブロックチェーンを使って、上述したように原子的スワップを初期化する。

【００８８】

６．アリスとボブが秘密を交換する。つまり、

【表２】

	送信	受信
アリス	S_2	S_1
ボブ	S_1	S_2

アリスとボブは、公式$P_1 = S_1 \cdot G$、$P_2 = S_2 \cdot G$を用いて正しい秘密を受け取ったことを検査してもよい。正しい値を交換しなければ、その原子的スワップの出力を消費することはできない。

【００８９】

７．今やアリスはS_1を保持しており、P_{AE}に対応する秘密鍵を計算することができる。アリス以外にはアリスの秘密鍵S_Aを知らないので、たとえS_1が公知であっても、他の誰もP_{AE}に対応する秘密鍵を計算することはできない。同様に、今やボブは秘密S_2を保持しており、P_{BE}に対応する秘密鍵を計算することができるが、ボブ以外には誰もこれを行なうことはできない。

アリスもボブも、その原子的スワップの自分のトランザクション出力を使わなければ、アリスの秘密S_2はボブに開示されず、ボブの秘密S_1はアリスに開示されない。この場合、アリスもボブもP_{AE}およびP_{BE}に対応する秘密鍵を計算することはできない。

【００９０】

ブロックチェーンは、トランザクションに署名し、トランザクション出力の所有権を証明するために、公開鍵／秘密鍵暗号化システムを使用する。これは、上記の実施形態の方法を使用して、同時にいくつかの暗号通貨において、P_{AE}およびP_{BE}にトランザクションを送ることを可能にする。たとえば、上記のステップ３においてP_{AE}およびP_{BE}を確立した後、

アリスは、BCHおよびETHの資金をP_{BE}に移動し、

ボブは、BCHおよびDASHの資金をP_{AE}に移す。

【００９１】

ひとたび原子的スワップが実行されたら、P_{BE}およびP_{AE}への秘密鍵はロック解除される。これらは、アリスが保有するビットコインおよびイーサリアム公開鍵ならびにボブが保有するビットコインおよびダッシュ公開鍵における資金をロック解除する。よって、アリスからボブへの次のトランザクションが安全に完了されることができる。

【表３】

	送信	受信
アリス	BCH, Eth	BCH, DASH
ボブ	BCH, DASH	BCH, Eth

【００９２】

これらのブロックチェーンは、ロック・スクリプトにおいて同じハッシュ関数を許容する必要がないことを注意しておく。

【００９３】

上記では、二当事者が原子的スワップを使用した秘密の交換を通じて公開鍵をロック解除する一般的な方法を与えている。これは、暗号通貨の交換を超えた用途をもち、ECDSAと同様の公開鍵／秘密鍵暗号方式を使用する任意のシステムに関連する。たとえば、他の使用事例は、以下を含むが、これらに限定されない：

１．分散ハッシュテーブル（Distributed Hash Table、DHT）へのアクセスの提供；
２．暗号化された計算；
３．プライベート電子メール・クライアント；
４．物流データおよび交換へのアクセス；
５．商品およびサービスのスワップ；
６．私的な価値交換；
７．鍵の階層。

【００９４】

共通の秘密の決定

適切な場合には、以下に述べるような公開鍵／秘密鍵システムを使用して、2者間の情報交換の安全な方法を使用することにより、セキュリティを向上させることができる。

【００９５】

共通の秘密（common secret、CS）が2者間で確立され、次いで、シェアのうち一つまたは複数のシェアの伝送のための安全な暗号化鍵を生成するために使用されることができる。共通の秘密（CS）は、何らかの秘密（$S_{A,B,1,2}$）、たとえば秘密の値、鍵、またはそのシェアの安全な交換を可能にするために生成され、使用される。

【００９６】

以下、便宜上、アリスおよびボブは第1のノード（C）、第2のノード（S）と称される。目的は、両方のノードが知っている共通の秘密（CS）を生成することであるが、その共通の秘密は、通信チャネルを介して送信されたものではないとする。こうして、その不正な発見の可能性が排除される。

【００９７】

安全な伝送技法は、独立した仕方で伝送の各端で生成されるCSに関わる。よって、両ノードはCSを知っているが、CSは潜在的に安全でない通信チャネルを通じて移動する必要はなかった。ひとたびCSが両端で確立されると、CSは、両ノードがその後通信のために使用できる安全な暗号化鍵を生成するために使用できる。

【００９８】

図2は、通信ネットワーク5を通じて第2のノード7と通信する第1のノード3を含むシステム1を示す。第1のノード3は、関連する第1の処理装置23を有し、第2のノード5は、関連する第2の処理装置27を有する。第1および第2のノード3、7は、コンピュータ、電話、タブレットコンピュータ、移動通信装置、コンピュータサーバー等の電子装置を含みうる。一例では、第1のノード3はクライアント（ユーザー）装置であってよく、第2のノード7はサーバーであってもよい。サーバーは、デジタル・ウォレット・プロバイダーのサーバーであってもよい。

【００９９】

第1のノード3は、第1のノードのマスター秘密鍵（V_{1C}）および第1のノードのマスター公開鍵（P_{1C}）を有する第1の非対称暗号ペアと関連付けられている。第2のノード（7）は、第2のノードのマスター秘密鍵（V_{1S}）および第2のノードのマスター公開鍵（P_{1S}）を有する第2の非対称暗号ペアと関連付けられている。言い換えると、第1および第2のノードはそれぞれ、それぞれ、公開鍵‐秘密鍵のペアを保持している。

【０１００】

それぞれの第1および第2のノード3、7についての第1および第2の非対称暗号ペアは、財布のための登録のような登録プロセスの間に生成されてもよい。各ノードについての公開鍵は、通信ネットワーク5を通じて公に共有されてもよい。

【０１０１】

第1のノード3および第2のノード7の両方における共通の秘密（CS）を決定するために、ノード3、7は、通信ネットワーク5を通じて秘密鍵を通信することなく、それぞれの方法300、400のステップを実行する。

【０１０２】

第1のノード3によって実行される方法300は、少なくとも第1のノードのマスター秘密鍵（V_{1C}）および生成子値（Generator Value、GV）に基づいて、第1のノードの第2の秘密鍵（V_{2C}）を決定すること（330）を含む。生成子値は、第1のノードと第2のノードとの間で共有されるメッセージ（M）に基づいてもよく、これは、のちにさらに詳細に説明するように、通信ネットワーク5を通じてメッセージを共有することを含んでいてもよい。方法300はまた、少なくとも第2のノードのマスター公開鍵（P_{1S}）および生成子値（GV）に基づいて第2のノードの第2公開鍵（P_{2S}）を決定すること（370）を含む。方法300は、第1のノードの第2の秘密鍵（V_{2C}）および第2のノードの第2の公開鍵（P_{2S}）に基づいて、共通の秘密（CS）を決定すること（380）を含む。

【０１０３】

同じ共通の秘密（CS）は、方法400によって第2のノード7においても決定できる。方法400は、第1のノードのマスター公開鍵（P_{1C}）および生成子値（GV）に基づいて、第1のノードの第2の公開鍵（P_{2C}）を決定すること（430）を含む。方法400は、さらに、第2のノードのマスター秘密鍵（V_{1S}）および生成子値（GV）に基づいて第2のノードの第2の秘密鍵（V_{2S}）を決定すること（470）を含む。方法400は、第2のノードの第2の秘密鍵（V_{2S}）および第1のノードの第2の公開鍵（P_{2C}）に基づいて、共通の秘密（CS）を決定すること（480）を含む。

【０１０４】

通信ネットワーク5は、ローカルエリアネットワーク、ワイドエリアネットワーク、セルラーネットワーク、無線通信ネットワーク、インターネットなどを含んでいてもよい。データが電気ワイヤ、光ファイバーのような通信媒体を介して、または無線で伝送されうるこれらのネットワークは、盗聴者11などによる盗聴を受けることがありうる。方法300、400は、通信ネットワーク5を通じて共通の秘密を伝送することなく、第1のノード3と第2のノード7の両方が独立して共通の秘密を決定することを許容しうる。

【０１０５】

このように、1つの利点は、共通の秘密（CS）が、潜在的に安全でない通信ネットワーク5を通じて秘密鍵を伝送する必要なしに、各ノードによって安全かつ独立に決定されうることである。次に、共通の秘密は、通信ネットワーク5を通じた第1のノード3と第2のノード7との間の暗号化された通信のための秘密鍵として（または秘密鍵の基礎として）使用されてもよい。

【０１０６】

方法300、400は、追加的なステップを含んでいてもよい。方法300は、第1のノード3において、メッセージ（M）および第1のノードの第2の秘密鍵（V_{2C}）に基づいて、署名されたメッセージ（SM1）を生成することを含んでいてもよい。方法300は、さらに、通信ネットワークを通じて、第1の署名されたメッセージ（SM1）を第2のノード7に送信すること（360）を含む。次に、第2のノード7は、第1の署名されたメッセージ（SM1）を受信するステップ440を実行してもよい。方法400はまた、第1のノードの第2の公開鍵（P_{2C}）を用いて第1の署名されたメッセージ（SM2）を有効確認するステップ450と、第1の署名されたメッセージ（SM1）を有効確認した結果に基づいて第1のノード3を認証するステップ460とを含む。有利には、これは、第2のノード7が、標榜される第1のノード（第1の署名されたメッセージが生成されたところ）が第1のノード3であることを認証することを許容する。これは、第1のノード3のみが第1のノードのマスター秘密鍵（V_{1C}）にアクセスでき、よって、第1のノード3のみが、第1の署名されたメッセージ（SM1）を生成するための第1のノードの第2の秘密鍵（V_{2C}）を決定することができるという想定に基づいている。同様に、第2の署名されたメッセージ（SM2）が第2のノード7において生成され、第1のノード3に送信されることができ、それにより第1のノード3がピアツーピア・シナリオにおけるように第2のノード7を認証できることが理解される。

【０１０７】

第1のノードと第2のノードの間でメッセージ（M）を共有することは、多様な仕方で達成されうる。一例では、メッセージは、第1のノード3において生成され、次いで、通信ネットワーク5を通じて第2のノード7に送られる。あるいはまた、メッセージは、第2のノード7において生成され、次いで通信ネットワーク5を通じて第2のノード7に送信されてもよい。さらに別の例では、メッセージは第3のノード9において生成され、メッセージは第1のノード3および第2のノード7の両方に送信されてもよい。さらに別の代替では、ユーザーがユーザー・インターフェース15を通じてメッセージを入力し、それが第1および第2のノード3、7によって受信されてもよい。さらに別の例では、メッセージ（M）はデータストア19から取り出され、第1および第2のノード3、7に送られてもよい。いくつかの例では、メッセージ（M）は公開であってもよく、よって、安全でないネットワーク5を通じて伝送されてもよい。

【０１０８】

さらなる例では、一つまたは複数のメッセージ（M）がデータストア13、17、19に記憶されてもよく、ここで、メッセージは、デジタル・ウォレットなどの何らかのエンティティ、または第1のノード3と第2のノード7との間に確立された通信セッションと関連付けられてもよい。このように、メッセージ（M）は、それぞれの第1および第2のノード3、7において、その財布またはセッションに関連付けられた共通の秘密（CS）を再生成するために、取り出され、使用されてもよい。

【０１０９】

有利には、共通の秘密（CS）の再生成を許容する記録は、記録自体が秘匿的に保存されるか、または安全に伝送される必要なしに、保持されうる。これは、多数のトランザクションが第1および第2のノード3、7において実行され、すべてのメッセージ（M）をノード自体に記憶することが実用的でない場合に有利でありうる。

【０１１０】

登録方法100、200の例を、図4を参照して説明する。図4では、方法100は第1のノード3によって実行され、方法200は第2のノード7によって実行される。これは、第1および第2の非対称暗号ペアをそれぞれの第1および第2のノード3、7について確立することを含む。

【０１１１】

非対称暗号ペアは、公開鍵暗号化で使用されるような、関連する秘密鍵と公開鍵を含む。この例では、楕円曲線暗号（ECC）と楕円曲線演算の特性を用いて、非対称暗号ペアが生成される。

【０１１２】

ECCについての標準は、効率的暗号標準グループ（Standards for Efficient Cryptography Group）（www.sceg.org）によって記述されるような既知の標準を含んでいてもよい。楕円曲線暗号はまた、US5,600,725、US5,761,305、US5,889,865、US5,896,455、US5,933,504、US6,122,736、US6,141,420、US6,618,483、US6,704,870、US6,785,813、US6,078,667、US6,792,530にも記述されている。

【０１１３】

方法100、200では、これは、第1および第2のノードが共通のECCシステムについて合意し、基底点（G）を使用することを含む。（注：基底点は共通生成子（Common Generator）と称することができるが、生成子値GVとの混同を避けるために基底点（base point）という用語が用いられる。）一例において、共通のECCシステムは、ビットコインによって使用されるECCシステムであるsecp256K1に基づいていてもよい。基底点（G）は、選択される、ランダムに生成される、または割り当てられることができる。

【０１１４】

ここで第1のノード3に目を転じると、方法100は、共通のECCシステムおよび基底点（G）を決めること110を含む。これは、第2のノード7または第3のノード9から共通のECCシステムおよび基底点を受信することを含んでいてもよい。あるいはまた、ユーザー・インターフェース15が第1のノード3に付随していて、それによりユーザーが、共通のECCシステムおよび／または基底点（G）を選択的に提供してもよい。さらに別の代替では、共通のECCシステムおよび／または基底点（G）の一方または両方が、第1のノード3によってランダムに選択されてもよい。第1のノード3は、通信ネットワーク5を通じて、基底点（G）とともに共通のECCシステムを使用することを示す通知を、第2のノード7に送信してもよい。次いで、第2のノード7は、共通のECCシステムおよび基底点（G）を使用することへの確認応答を示す通知を送信することによって、決着してもよい（210）。

【０１１５】

方法100は、第1のノード3が第1のノードのマスター秘密鍵（V_{1C}）および第1のノードのマスター公開鍵（P_{1C}）を含む第1の非対称暗号ペアを生成120することをも含む。これは、少なくとも部分的には、共通のECCシステムにおいて指定された許容可能な範囲内のランダムな整数に基づいて第1のマスター秘密鍵（V_{1C}）を生成することを含む。これはまた、第1のノードのマスター秘密鍵（P_{1C}）と基底点（G）の公式：

$P_{1C}=V_{1C}\times G$　　(式1)
に従った楕円曲線点乗算に基づいて、第1のノードのマスター公開鍵（P_{1C}）を決定することを含む。
【０１１６】
このように、第1の非対称暗号ペアは次を含む：
V_{1C}：第1のノードによって秘密に保持される、第1のノードのマスター秘密鍵、
P_{1C}：公知にされている第1のノードのマスター公開鍵。
【０１１７】
第1のノード3は、第1のノードのマスター秘密鍵（V_{1C}）および第1のノードのマスター公開鍵（P_{1C}）を、第1のノード3に関連付けられた第1のデータストア13に格納してもよい。セキュリティのために、第1のノードのマスター秘密鍵（V_{1C}）は、鍵が秘匿されたままであることを保証するために、第1のデータストア13の安全な部分に格納されてもよい。
【０１１８】
方法100は、さらに、通信ネットワーク5を通じて、第1のノードのマスター公開鍵（P_{1C}）を第2のノード7に送信すること130を含む。第2のノード7は、第1のノードのマスター公開鍵（P_{1C}）を受信220すると、第1のノードのマスター公開鍵（P_{1C}）を第2のノード7に関連付けられた第2のデータストア17に格納230してもよい。
【０１１９】
第1のノード3と同様に、第2のノード7の方法200は、第2のノードのマスター秘密鍵（V_{1S}）および第2のノードのマスター公開鍵（P_{1S}）を含む第2の非対称暗号ペアを生成240することを含む。第2のノードのマスター秘密鍵（V_{1S}）も、前記許容可能な範囲内のランダムな整数である。次に、第2のノードのマスター公開鍵（P_{1S}）は、次の公式によって決定される：
$P_{1S}=V_{1S}\times G$　　(式2)
【０１２０】
このように、第2の非対称暗号ペアは、次を含む：
V_{1S}：第2のノードによって秘密に保持されている第2のノードのマスター秘密鍵、
P_{1S}：公知にされている第2のノードのマスター公開鍵。
【０１２１】
第2のノード7は、第2の非対称暗号ペアを第2のデータストア17に格納してもよい。方法200は、さらに、第1のノード3に第2のノードのマスター公開鍵（P_{1S}）を送信すること250を含む。次に、第1のノード3は、第2のノードのマスター公開鍵（P_{1S}）を受信140し、格納150してもよい。
【０１２２】
いくつかの代替において、それぞれの公開マスター鍵は、第3のノード9（たとえば、信頼される第三者）に付随する第3のデータストア19において受信され、格納されてもよいことが理解される。これは、認証局などの公開ディレクトリとして機能する第三者を含みうる。このように、いくつかの例において、第1のノードのマスター公開鍵（P_{1C}）は、共通の秘密（CS）が必要とされるときにのみ、第2のノード7によって要求され、受信されうる（逆も同様）。
【０１２３】
登録ステップは、たとえば、デジタル・ウォレットの初期セットアップとして一度だけ必要とされる場合がある。
【０１２４】
次に、共通の秘密（CS）を決定する例が図5を参照して記述される。共通の秘密（CS）は、第1のノード3と第2のノード7との間の特定のセッション、時間、トランザクション、または他の目的のために使用されてもよく、同じ共通の秘密（CS）を使用することは望ましくない、または安全でないことがありうる。このように、共通の秘密（CS）は、異なるセッション、時間、トランザクションなどの間で変更されることがある。

【０１２５】
下記は、上述した安全な伝送技法を説明するために与えられる。
【０１２６】
この例では、第1のノード3によって実行される方法300は、メッセージ（M）を生成310することを含む。メッセージ（M）は、ランダム、疑似ランダム、またはユーザー定義であってもよい。一例では、メッセージ（M）はユニックス時間およびナンス（および任意の値）に基づいている。たとえば、メッセージ（M）は、次のように与えられてもよい。
【０１２７】
メッセージ（M）＝ユニックス時間＋ナンス　　（式3）
【０１２８】
いくつかの例では、メッセージ（M）は任意（arbitrary）である。しかしながら、メッセージ（M）は、いくつかのアプリケーションにおいて有用でありうる選択的な値（たとえば、ユニックス時間など）を有しうることが理解される。
【０１２９】
方法300は、通信ネットワーク3を通じて、メッセージ（M）を第2のノード7に送信315することを含む。メッセージ（M）は秘密鍵に関する情報を含まないので、メッセージ（M）は安全でないネットワークを通じて送信されてもよい。
【０１３０】
方法300は、メッセージ（M）に基づいて生成子値（GV）を決定するステップ320をさらに含む。この例では、これはメッセージの暗号学的ハッシュを決定することを含む。暗号学的ハッシュ・アルゴリズムの例は、256ビットの生成子値（GV）を生成するSHA-256を含む。すなわち、
GV＝SHA-256(M)　　(式4)
【０１３１】
他のハッシュ・アルゴリズムが使用されてもよいことが理解される。これは、セキュアハッシュアルゴリズム（SHA）ファミリーの他のアルゴリズムを含みうる。いくつかの具体例は、SHA3-224、SHA3-256、SHA3-384、SHA3-512、SHAKE128、SHAKE256を含むSHA-3サブセット内のインスタンスを含む。他のハッシュ・アルゴリズムには、RACE完全性プリミティブ評価メッセージダイジェスト（RACE Integrity Primitives Evaluation Message Digest、RIPEMD）ファミリーのものを含みうる。具体例は、RIPEMD-160を含みうる。他のハッシュ関数は、ゼモール‐ティリック（Zemor-Tillich）ハッシュ関数およびナップザック・ベースのハッシュ関数に基づくファミリーを含みうる。
【０１３２】
次いで、方法300は、第2のノードのマスター秘密鍵（V_{1C}）および生成子値（GV）に基づいて第1のノードの第2の秘密鍵（V_{2C}）を決定するステップ330を含む。これは、次の公式による、第1のノードのマスター秘密鍵（V_{1C}）と生成子値（GV）とのスカラー加算に基づくことができる：
$V_{2C}=V_{1C}+GV$　　(式5)
【０１３３】
このように、第1のノードの第2の秘密鍵（V_{2C}）はランダムな値ではなく、第1のノードのマスター秘密鍵から決定論的に導出される。暗号ペアにおける対応する公開鍵、すなわち、第1のノードの第2の公開鍵（P_{2C}）は、次の関係をもつ：
$P_{2C}=V_{2C}\times G$　　(式6)
式5からのV_{2C}を式6に代入すると、次式が得られる：
$P_{2C}=(V_{1C}+GV)\times G$　　(式7)
ここで'＋'演算子は楕円曲線点加算を指す。
楕円曲線暗号代数が分配則を満たすことに注意すると、式7は次のように表わせる：
$P_{2C}=V_{1C}\times G+GV\times G$　　(式8)
最後に、式1を式7に代入して、次式を与えることができる：
$P_{2C}=P_{1C}+GV\times G$　　(式9.1)

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

【０１３４】

このように、対応する第1のノードの第2の公開鍵（P_{2C}）は、第1のノードのマスター公開鍵（P_{1C}）およびメッセージ（M）の知識が与えられれば導出可能である。第2のノード7は、第1のノードの第2の公開鍵（P_{2C}）を独立して決定するためのそのような知識を有してもよく、このことは、方法400に関して以下でさらに詳細に議論される。

【０１３５】

方法300は、さらに、メッセージ（M）および決定された第1のノードの第2の秘密鍵（V_{2C}）に基づいて、第1の署名されたメッセージ（SM1）を生成350することを含む。署名されたメッセージを生成することは、デジタル署名アルゴリズムを適用して、メッセージ（M）にデジタル的に署名することを含む。一例では、これは、楕円曲線デジタル署名アルゴリズム（Elliptic Curve Digital Signature Algorithm、ECDSA）においてメッセージに第1のノードの第2の秘密鍵（V_{2C}）を適用して、第一の署名されたメッセージ（SM1）を得ることを含む。

【０１３６】

ECDSAの例は、secp256k1、secp256r1、secp384r1、sec3cp521r1を用いたECCシステムに基づくものを含む。

【０１３７】

第1の署名されたメッセージ（SM1）は、第2のノード7における対応する第1のノードの第2の公開鍵（P_{2C}）を用いて検証されることができる。第1の署名されたメッセージ（SM1）のこの検証は、第1のノード3を認証するために第2のノード7によって使用されてもよく、これについては下記の方法400において論じられる。

【０１３８】

次いで、第1のノード3は、第2のノードの第2の公開鍵（P_{2S}）を決定370してもよい。上記で論じたように、第2のノードの第2の公開鍵（P_{2S}）は、少なくとも第2のノードのマスター公開鍵（P_{1S}）および生成子値（GV）に基づいていてもよい。この例では、公開鍵は、秘密鍵を基底点（G）と楕円曲線点乗算したものとして決定370'されるので、第2のノードの第2の公開鍵（P2S）は、式6と同様の仕方で、次のように表わせる：

$P_{2S}＝V_{2S}×G$　(式10.1)

$P_{2S}＝P_{1S}＋GV×G$　(式10.2)

式10.2の数学的証明は、第1のノードの第2の公開鍵（P_{2C}）について式9.1を導出するために上述したものと同じである。第1のノード3は、第2のノード7とは独立して、第2のノードの第2の公開鍵を決定370することができることが理解される。

【０１３９】

次いで、第1のノード3は、決定された第1のノードの第2の秘密鍵（V_{2C}）および決定された第2のノードの第2の公開鍵（P_{2S}）に基づいて、共通の秘密（CS）を決定380してもよい。共通の秘密（CS）は、第1のノード3によって次の式によって決定されうる：

$S＝V_{2C}×P_{2S}$　(式11)

【０１４０】

<u>第2のノード7において実行される方法400</u>

ここで、第2のノード7で実行される対応する方法400について述べる。これらのステップのいくつかは、第1のノード3によって実行された上記のステップと同様であることが理解される。

【０１４１】

方法400は、第1のノード3から、通信ネットワーク5を通じてメッセージ（M）を受信すること410を含む。これは、ステップ315において第1のノード3によって送信されたメッセージ（M）を含みうる。次いで、第2のノード7は、メッセージ（M）に基づいて、生成子値（GV）を決定420する。第2のノード7によって生成子値（GV）を決定するステップ420は、上述の第1のノードによって実行されるステップ320と同様である。この例では、第2のノード7は、第1のノード3から独立して、この決定ステップ420を実行する。

【０１４２】
次のステップは、第1のノードのマスター公開鍵（P_{1C}）および生成子値（GV）に基づいて、第1のノードの第2の公開鍵（P_{2C}）を決定するステップを含む。この例では、公開鍵は、秘密鍵を基底点（G）と楕円曲線点乗算したものとして決定430'されるので、第1のノードの第2の公開鍵（P_{2C}）は、式9と同様の仕方で、次のように表わすことができる：

$$P_{2C}=V_{2C}\times G \quad (式12.1)$$

$$P_{2C}=P_{1C}+GV\times G \quad (式12.2)$$

式12.1および12.2の数学的証明は、式10.1および10.2について上記で論じたものと同じである。

【０１４３】
方法400は、主張される第1のノード3が第1のノード3であることを認証するために、第2のノード7によって実行されるステップを含んでいてもよい。上記で論じたように、これは、第1のノード3から第1の署名されたメッセージ（SM1）を受信すること440を含む。次いで、第2のノード7は、ステップ430において決定された第1のノードの第2の公開鍵（P_{2C}）を用いて、第1の署名されたメッセージ（SM1）上の署名を有効確認450してもよい。

【０１４４】
デジタル署名の検証は、上記で論じた楕円曲線デジタル署名アルゴリズム（ECDSA）に従ってなされてもよい。重要なことに、第1のノードの第2の秘密鍵（V_{2C}）を用いて署名された第1の署名されたメッセージ（SM1）は、V_{2C}とP_{2C}が暗号ペアを形成するので、対応する第1のノードの第2の公開鍵（P_{2C}）を用いたときにのみ正しく検証されるべきである。これらの鍵は、第1のノード3の登録時に生成された第1のノードのマスター秘密鍵（V_{1C}）および第1のノードのマスター公開鍵（P_{1C}）をもとに決定論的であるため、第1の署名されたメッセージ（SM1）を検証することは、第1の署名されたメッセージ（SM1）を送信する主張された第1のノードが、登録の際の同じ第1のノード3であることを認証する基礎として使用できる。このように、第2のノード7は、第1の署名されたメッセージを有効確認（450）した結果に基づいて、第1のノード3を認証（460）するステップをさらに実行してもよい。

【０１４５】
上記の認証は、2つのノードのうち一方が信頼されるノードであり、それらのノードのうちの1つだけが認証される必要があるシナリオに好適でありうる。たとえば、第1のノード3はクライアントであってもよく、第2のノード7は、財布プロバイダーのようなクライアントによって信頼されるサーバーであってもよい。このように、サーバー（第2のノード7）は、クライアントがサーバー・システムにアクセスすることを許容するために、クライアント（第1のノード3）のクレデンシャルを認証する必要があることがありうる。サーバーがクライアントに対するサーバーのクレデンシャルを認証することは必要はないことがありうる。しかしながら、いくつかのシナリオでは、ピアツーピア・シナリオのように、両方のノードが互いに対して認証されることが望ましいことがある。

【０１４６】
方法400は、さらに、第2のノード7が、第2のノードのマスター秘密鍵（V_{1S}）および生成子値（GV）に基づいて第2のノードの第2の秘密鍵（V_{2S}）を決定すること470を含んでいてもよい。第1のノード3によって実行されるステップ330と同様に、第2のノードの第2の秘密鍵（V_{2S}）は、次の公式に従って、第2のノードのマスター秘密鍵（V_{1S}）および生成子値（GV）のスカラー加算に基づくことができる：

$$V_{2S}=V_{1S}+GV \quad (式13.1)$$

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

次いで、第2のノード7は、第1のノード3とは独立に、第2のノードの第2の秘密鍵（V_{2S}）および第1のノードの第2の公開鍵（P_{2C}）に基づいて、次の式に基づいて、共通の秘密（CS）を決定480してもよい：

$S = V_{2S} \times P_{2C}$ (式14)

第1のノード3によって決定された共通の秘密（CS）は、第2のノード7において決定された共通の秘密（CS）と同じである。ここで、式11と式14が同じ共通の秘密（CS）を与えることの数学的証明を述べる。

【０１４７】

第1のノード3によって決定された共通の秘密（CS）を見ると、式10.1を式11に代入して、次のようになる：

$S = V_{2C} \times P_{2S}$ (式11)

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

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

第2のノード7によって決定された共通の秘密（CS）を見ると、式12.1を式14に代入して、次のようになる：

$S = V_{2S} \times P_{2C}$ (式14)

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

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

ECC代数は可換なので、式15と式16は等価である。なぜなら：

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

【０１４８】

今や共通の秘密（CS）は、秘密鍵として、または、第1のノード3と第2のノード7との間の安全な通信のための対称鍵アルゴリズムにおける秘密鍵の基礎として使用されうる。この通信は、秘密鍵の一部、秘密鍵の表現もしくは秘密鍵のための識別子、または秘密鍵についてのニーモニックを伝達するために使用されてもよい。よって、ひとたび本発明が、たとえば、デジタル・ウォレットまたは他の制御された資源のセットアップの際に使用されたら、その後は両当事者間の安全な通信が実行できる。

【０１４９】

共通の秘密（CS）は、楕円曲線点（x_S,y_S）の形であってもよい。これは、ノード3、7によって合意された標準的な公知の操作を使用して、標準的な鍵フォーマットに変換されてもよい。たとえば、x_S値は、AES_{256}暗号化のための鍵として使用できる256ビットの整数であってもよい。また、RIPEMD160を使用して、160ビットの整数に変換されてもよい。この長さ鍵を必要とする任意の用途のためである。

【０１５０】

必要に応じて共通の秘密（CS）が決定されてもよい。重要なことに、第1のノード3は、共通の秘密（CS）を格納する必要がない。これはメッセージ（M）に基づいて再度決定できるからである。いくつかの例では、使用されるメッセージ（M）は、マスター秘密鍵に必要とされるのと同じレベルのセキュリティなしに、データストア13、17、19（または他のデータストア）に格納されてもよい。いくつかの例では、メッセージ（M）は公に利用可能であってもよい。

【０１５１】

しかしながら、何らかの用途によっては、共通の秘密（CS）が第1のノードのマスター秘密鍵（V_{1C}）と同じくらい安全に保たれていれば、共通の秘密（CS）を第1のノードに付随する第1のデータストア（X）に格納することができる。

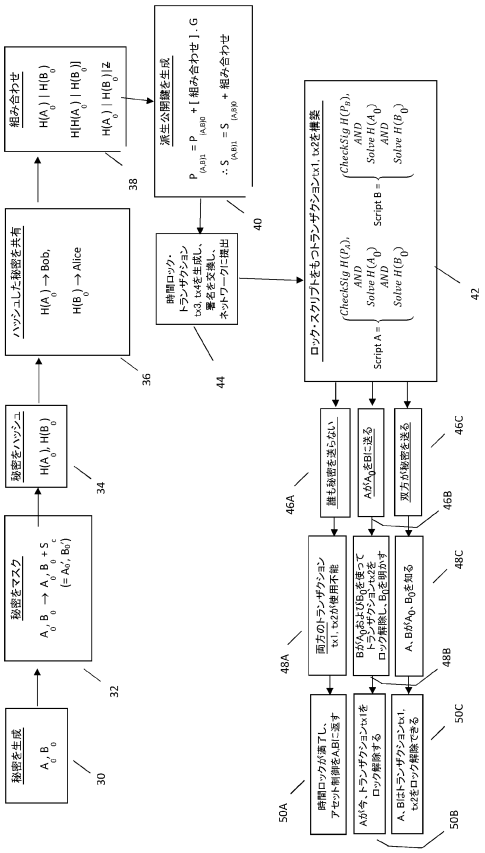
【０１５２】

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、添付の特許請求の範囲によって定義される本発明の範囲から逸脱することなく、多くの代替実施形態を設計することができるであろうことに注意しておくべきである。特許請求の範囲においては、括弧内に付した参照符号があったとしても、請求項を限定するものと解釈してはならない。「含む」および「有する」等の語は、いずれかの請求項または明細書全体に列挙されたもの以外の要素また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、「含む」は「…を含むまたは…からなる」ことを意味し、「有する」は「…を含むまたは…からなる」を意味する。要素の単数形での言及は、そのような要素の複数

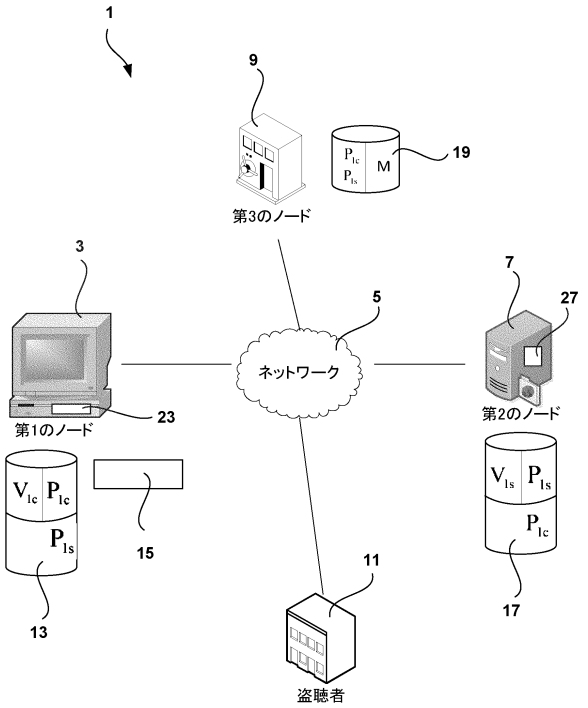
形での言及を除外するものではなく、その逆もまた同様である。本発明は、いくつかの別個の要素を有するハードウェアによって、および好適にプログラムされたコンピュータによって実装されうる。いくつかの手段を列挙する装置請求項においては、これらの手段のいくつかは、同一のハードウェア項目によって具現されてもよい。ある種の措置が相互に異なる従属請求項に記載されているというだけの事実が、これらの措置の組み合わせが有利に利用できないことを示すものではない。

【図面】

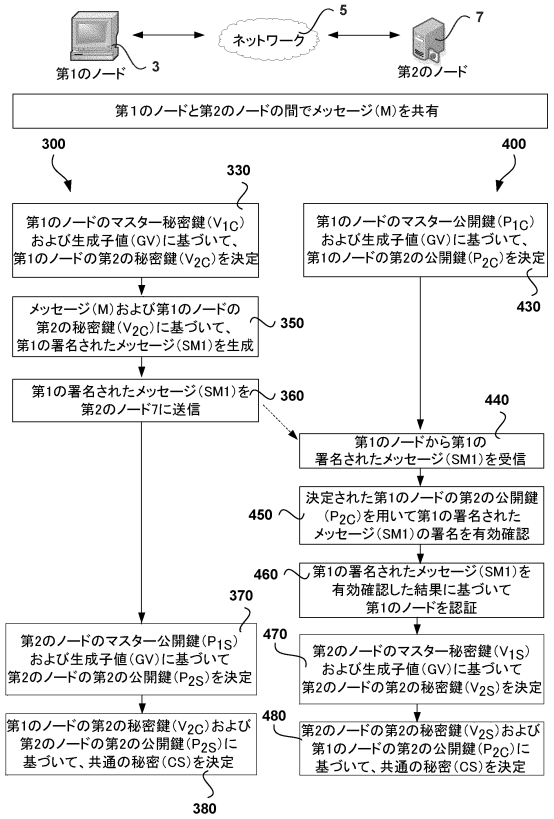
【図１】



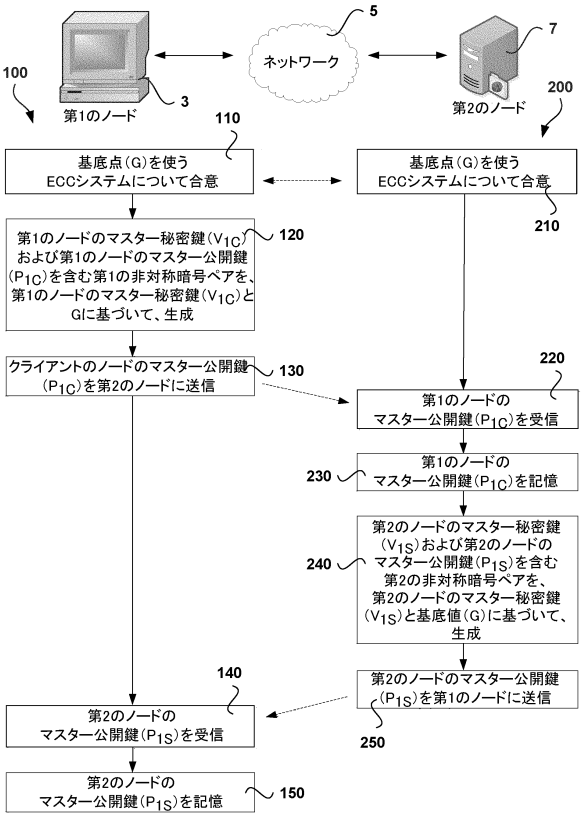
【図２】



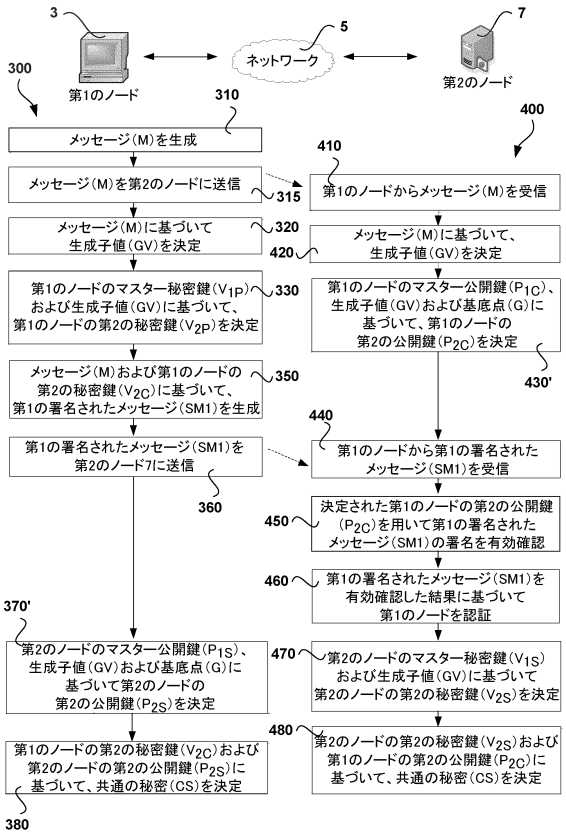
【図３】


第1のノード
3
ネットワーク
5
第2のノード
7
第1のノードと第2のノードの間でメッセージ(M)を共有
300
400
330
第1のノードのマスター秘密鍵(V_{1C})および生成子値(GV)に基づいて、第1のノードの第2の秘密鍵(V_{2C})を決定
第1のノードのマスター公開鍵(P_{1C})および生成子値(GV)に基づいて、第1のノードの第2の公開鍵(P_{2C})を決定
430
メッセージ(M)および第1のノードの第2の秘密鍵(V_{2C})に基づいて、第1の署名されたメッセージ(SM1)を生成
350
第1の署名されたメッセージ(SM1)を第2のノード7に送信
360
440
第1のノードから第1の署名されたメッセージ(SM1)を受信
決定された第1のノードの第2の公開鍵(P_{2C})を用いて第1の署名されたメッセージ(SM1)の署名を有効確認
450
460
第1の署名されたメッセージ(SM1)を有効確認した結果に基づいて第1のノードを認証
370
第2のノードのマスター公開鍵(P_{1S})および生成子値(GV)に基づいて第2のノードの第2の公開鍵(P_{2S})を決定
470
第2のノードのマスター秘密鍵(V_{1S})および生成子値(GV)に基づいて第2のノードの第2の秘密鍵(V_{2S})を決定
480
第1のノードの第2の秘密鍵(V_{2C})および第2のノードの第2の公開鍵(P_{2S})に基づいて、共通の秘密(CS)を決定
第2のノードの第2の秘密鍵(V_{2S})および第1のノードの第2の公開鍵(P_{2C})に基づいて、共通の秘密(CS)を決定
380

【図４】


100
第1のノード
3
ネットワーク
5
第2のノード
7
200
110
基底点(G)を使うECCシステムについて合意
基底点(G)を使うECCシステムについて合意
210
第1のノードのマスター秘密鍵(V_{1C})および第1のノードのマスター公開鍵(P_{1C})を含む第1の非対称暗号ペアを、第1のノードのマスター秘密鍵(V_{1C})とGに基づいて、生成
120
クライアントのノードのマスター公開鍵(P_{1C})を第2のノードに送信
130
220
第1のノードのマスター公開鍵(P_{1C})を受信
230
第1のノードのマスター公開鍵(P_{1C})を記憶
240
第2のノードのマスター秘密鍵(V_{1S})および第2のノードのマスター公開鍵(P_{1S})を含む第2の非対称暗号ペアを、第2のノードのマスター秘密鍵(V_{1S})と基底値(G)に基づいて、生成
140
第2のノードのマスター公開鍵(P_{1S})を第1のノードに送信
250
第2のノードのマスター公開鍵(P_{1S})を受信
第2のノードのマスター公開鍵(P_{1S})を記憶
150

【図５】


3
第1のノード
ネットワーク
5
第2のノード
7
300
310
メッセージ(M)を生成
400
410
メッセージ(M)を第2のノードに送信
315
第1のノードからメッセージ(M)を受信
メッセージ(M)に基づいて生成子値(GV)を決定
320
420
メッセージ(M)に基づいて、生成子値(GV)を決定
第1のノードのマスター秘密鍵(V_{1P})および生成子値(GV)に基づいて、第1のノードの第2の秘密鍵(V_{2P})を決定
330
第1のノードのマスター公開鍵(P_{1C})、生成子値(GV)および基底点(G)に基づいて、第1のノードの第2の公開鍵(P_{2C})を決定
430'
メッセージ(M)および第1のノードの第2の秘密鍵(V_{2C})に基づいて、第1の署名されたメッセージ(SM1)を生成
350
第1の署名されたメッセージ(SM1)を第2のノード7に送信
360
440
第1のノードから第1の署名されたメッセージ(SM1)を受信
決定された第1のノードの第2の公開鍵(P_{2C})を用いて第1の署名されたメッセージ(SM1)の署名を有効確認
450
460
第1の署名されたメッセージ(SM1)を有効確認した結果に基づいて第1のノードを認証
370'
第2のノードのマスター公開鍵(P_{1S})、生成子値(GV)および基底点(G)に基づいて第2のノードの第2の公開鍵(P_{2S})を決定
470
第2のノードのマスター秘密鍵(V_{1S})および生成子値(GV)に基づいて第2のノードの第2の秘密鍵(V_{2S})を決定
480
第1のノードの第2の秘密鍵(V_{2C})および第2のノードの第2の公開鍵(P_{2S})に基づいて、共通の秘密(CS)を決定
第2のノードの第2の秘密鍵(V_{2S})および第1のノードの第2の公開鍵(P_{2C})に基づいて、共通の秘密(CS)を決定
380

フロントページの続き

(33)優先権主張国・地域又は機関
国際事務局(IB)
(31)優先権主張番号　1807816.2
(32)優先日　平成30年5月14日(2018.5.14)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　PCT/IB2018/053350
(32)優先日　平成30年5月14日(2018.5.14)
(33)優先権主張国・地域又は機関
国際事務局(IB)
(31)優先権主張番号　1807807.1
(32)優先日　平成30年5月14日(2018.5.14)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　PCT/IB2018/053346
(32)優先日　平成30年5月14日(2018.5.14)
(33)優先権主張国・地域又は機関
国際事務局(IB)
(31)優先権主張番号　1807811.3
(32)優先日　平成30年5月14日(2018.5.14)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　PCT/IB2018/053349
(32)優先日　平成30年5月14日(2018.5.14)
(33)優先権主張国・地域又は機関
国際事務局(IB)
内
審査官　青木 重徳

(56)参考文献　国際公開第２０１７／１８７３９６（WO，Ａ１）
国際公開第２０１８／０２０３７０（WO，Ａ１）
国際公開第２０１７／１４５０１６（WO，Ａ１）
Marcin Andrychowicz et al.，Fair Two-Party Computations via Bitcoin Deposits，Cryptology ePrint Archive，Paper 2013/837，［オンライン］，2014年03月05日，〈URL: https://eprint.iacr.org/2013/837.pdf〉，（検索日 令和５年４月２５日）、インターネット

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／３２
Ｈ０４Ｌ　９／０８