(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7554493号
(P7554493)

(45)発行日　令和6年9月20日(2024.9.20)

(24)登録日　令和6年9月11日(2024.9.11)

(51)国際特許分類
Ｈ０４Ｌ　9/08 (2006.01)

ＦＩ
Ｈ０４Ｌ　9/08　C
Ｈ０４Ｌ　9/08　F

請求項の数　13　外国語出願　（全43頁）

(21)出願番号　特願2022-206111(P2022-206111)
(22)出願日　令和4年12月23日(2022.12.23)
(62)分割の表示　特願2020-506804(P2020-506804)の分割
原出願日　平成30年8月13日(2018.8.13)
(65)公開番号　特開2023-24683(P2023-24683A)
(43)公開日　令和5年2月16日(2023.2.16)
審査請求日　令和4年12月23日(2022.12.23)
(31)優先権主張番号　1713064.2
(32)優先日　平成29年8月15日(2017.8.15)
(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/IB2017/054961
(32)優先日　平成29年8月15日(2017.8.15)
(33)優先権主張国・地域又は機関
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】　閾ボールトを生成する、コンピュータにより実施される方法

(57)【特許請求の範囲】

【請求項１】

閾ボールトを安全に生成する、コンピュータにより実施される方法であって、

前記閾ボールトは、公開鍵アドレス、秘密鍵及び閾値ｋを有し、複数のノードの中の各ノードは、各々の第１秘密及び各々のｋ－１次多項式関数を有し、各ノードの各々の第１秘密が前記ｋ－１次多項式関数の定数項としてセットされ、前記複数のノードは第１ノードを含み、該第１ノードはその各々の多項式関数である第１多項式関数を有し、前記第１ノードによって実行される当該方法は、

前記複数のノードの中の他のノードの夫々について、前記第１多項式関数内の変数を、当該他のノードに関連した値にセットし、その値に対する前記第１多項式関数の各々の結果を求め、該各々の結果を当該他のノードへ安全に送ることと、

複数の前記他のノードから、各々の第２秘密を受信することであり、該各々の第２秘密の夫々は、前記第１ノードに関連した第１値に対する当該他のノードの各々の多項式関数の結果である、ことと、

前記他のノードに対して、前記各々の結果の夫々の楕円曲線点乗算を送ることと、

前記複数の他のノードから、該他のノードによって前記複数のノードのいずれかと共有される前記各々の第２秘密の楕円曲線点乗算を受信することと、

前記他のノードから受信された前記各々の第２秘密の和に基づき、前記秘密鍵の第１秘密鍵シェアを形成及び格納することと、

少なくとも閾数のノードから、前記各々の第２秘密の前記受信された楕円曲線点乗算の

和に基づき、且つ、多項式補間を用いて、前記秘密鍵に対応する公開鍵を決定することとを有する方法。

【請求項２】

各々の多項式関数の夫々は、当該多項式関数に対応する各々のノードによって独立して選択され、

各々の多項式関数の夫々は、係数の異なる組を有する、

請求項１に記載の方法。

【請求項３】

各々の多項式関数の夫々は、次の式

【数１】

$$f_i = \sum_{t=0}^{(k-1)} a_t x^t \bmod n$$

により表され、

ｆｉは、ｉ番目のノードの多項式関数であり、ｎは、楕円曲線の整数位数であり、ｘは、当該多項式関数内の変数であり、αｔは、当該多項式関数のための係数の組であり、ｔは、インデックスである、

請求項１又は２に記載の方法。

【請求項４】

前記他のノードに対して、前記第１多項式関数の各係数の楕円曲線点乗算を送ることを更に有する、

請求項１乃至３のうちいずれか一項に記載の方法。

【請求項５】

前記複数の他のノードから、前記複数の他のノードに対応する各々の多項式関数の各係数の楕円曲線点乗算を受信することを更に有する、

請求項１乃至４のうちいずれか一項に記載の方法。

【請求項６】

前記他のノードから受信された前記各々の第２秘密を妥当性確認することを更に有する、

請求項１乃至５のうちいずれか一項に記載の方法。

【請求項７】

前記第１秘密鍵シェアを形成することは、次の式

【数２】

$$d_{A(i)} = \sum_{h=1}^{j} f_h(i) \bmod n$$

により表される和を決定することを含み、

ｄＡ（ｉ）は、前記第１秘密鍵シェアであり、ｈは、前記他のノードのインデックスであり、ｆｈ（）は、ｈ番目のノードの各々の多項式関数であり、ｎは、楕円曲線の整数位数であり、ｉは、前記第１ノードに関連した前記第１値であり、

各非ヌルｆｈ（ｉ），ｈ＝１，・・・，ｊは、前記各々の第２秘密のうちの１つである、

請求項１乃至６のうちいずれか一項に記載の方法。

【請求項８】

ジョイントゼロ秘密シェアのディーラーなし秘密分配に前記他のノードとともに協調して参加することを通じて前記第１秘密鍵シェアを変更することを更に有し、これによって、前記第１ノードは、第１ゼロ秘密シェアを受信し、該第１ゼロ秘密シェアを加えることによって前記第１秘密鍵シェアを変更して、変更された第１秘密鍵シェアを形成及び格納する、

請求項１乃至７のうちいずれか一項に記載の方法。
【請求項９】
前記公開鍵を決定することは、前記各々の第２秘密の前記受信された楕円曲線点乗算の複数の既知の値から前記多項式関数の係数を決定することによって、前記多項式関数を決定することを有する、
請求項１乃至８のうちいずれか一項に記載の方法。
【請求項１０】
前記多項式関数を決定することは、誤り訂正アルゴリズムを実行することを有する、
請求項９に記載の方法。
【請求項１１】
前記多項式関数を決定することは、バーレカンプ－ウェルチ復号化アルゴリズムを実行することを有する、
請求項９又は１０に記載の方法。
【請求項１２】
前記多項式関数を決定することは、
誤り位置多項式関数、及び前記多項式関数と前記誤り位置多項式関数との積である第２多項式関数とを定義することと、
前記各々の第２秘密の前記受信された楕円曲線点乗算の複数の既知の値から前記第２多項式関数及び前記誤り位置多項式関数の係数を決定することと、
前記第２多項式関数及び前記誤り位置多項式関数から前記多項式関数を決定することとを有する、
請求項９乃至１１のうちいずれか一項に記載の方法。
【請求項１３】
ブロックチェーントランザクションに参加するコンピューティングデバイスであって、
当該コンピューティングデバイスは、複数のノードの中の第１ノードであり、
当該コンピューティングデバイスは、
プロセッサと、
メモリと、
前記複数のノードの中の他のノードへのネットワーク接続を提供するネットワークインターフェイスと、
前記プロセッサによって実行される場合に、該プロセッサに、請求項１乃至１２のうちいずれか一項に記載の方法を実行させるコンピュータ実行可能命令を含むブロックチェーンアプリケーションと
を有する、
コンピューティングデバイス。
【発明の詳細な説明】
【技術分野】
【０００１】
本願は、概して、データ及びコンピュータに基づくリソースの安全性に関係がある。より具体的には、本願は、楕円曲線暗号化（Elliptic Curve Cryptography）、楕円曲線デジタル署名アルゴリズム（Elliptic Curve Digital Signature Algorithm）（ＥＣＤＳＡ）アプリケーション、及び閾値暗号化（Threshold Cryptography）に関係がある。例えば、本願は、ＥＣＤＳＡ要件に準拠しているデジタル署名を生成することを閾数のノードに可能にしながら、如何なるノード又はサードパーティも秘密鍵自体を生成することができないように、公開－秘密鍵対を安全に且つ非公開で協調して生成するプロセスの例を記載する。
【背景技術】
【０００２】
本文書中、あらゆる形式の電子的な、コンピュータに基づく分散台帳（distributed ledgers）を含むよう語「ブロックチェーン」が使用される。それらは、合意に基づくブロ

ックチェーン及びトランザクションチェーン技術、許可（permissioned）及び無許可（un-permissioned）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た応用は、他のブロックチェーン実施が提案及び開発されているとはいえ、ビットコイン台帳である。ビットコインが、便宜上、単に説明のために本明細書で言及され得るが、本発明は、ビットコインブロックチェーンとともに使用することに限られず、代替のブロックチェーン実施及びプロトコルが本願の適用範囲内にあることが留意されるべきである。

【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、トランザクションから成るブロックで構成された、コンピュータに基づく非中央集権型システムとして実装される。各トランザクションは、ブロックチェーンシステム内のアドレス間の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つの入力及び少なくとも１つの出力を含む。ブロックチェーンにその始まり以来書き込まれてきた全てのトランザクションの永久的且つ不変な記録を構成するようブロックどうしが連鎖するように、各ブロックは前のブロックのハッシュを含む。

【０００４】

非中央集権化（decentralisation）の概念は、ブロックチェーンシステムにおいて基本的なことである。非中央集権型システムには、分散型又は中央集権型システムとは異なり、単一障害点がないという利点がある。従って、それらは、向上した安全性及び回復力のレベルを提供する。この安全性は、楕円曲線暗号化及びＥＣＤＳＡのような既知の暗号化技術の使用によって更に高められる。

【０００５】

ブロックチェーンシステム内のデジタルアセット（通貨、有形資産、知的資産、コンピューティング資源、又は何らかの他の物若しくはサービスを代表することができる。）に対する所有権及び制御は、公開－秘密鍵対により制御される。未使用トランザクション出力（unspent transaction output）（ＵＴＸＯ）アドレスは、公開鍵と同種であり、そのＵＸＴＯに割り当てられているアセットに対する制御は、対応する秘密鍵の使用によって証明される。ＵＸＴＯを他のトランザクションへの入力として使用することによってデジタルアセットを移すために、入力は、対応する秘密鍵によってデジタル署名されなければならない。従って、秘密鍵の安全性及び秘密性に対する制御は、ブロックチェーンの安全性及び信頼性において基本的なことである。

【発明の概要】

【０００６】

よって、そのようなシステムの安全性を更に高める解決法の必要性が存在する。本願は、とりわけ、そのような利点を提供する。

【０００７】

本願は、添付の特許請求の範囲で定義される方法及びシステムを提供する。

【０００８】

本願の態様によれば、閾ボールト（threshold vault）を安全に生成する、コンピュータにより実施される方法であって、前記閾ボールトは、公開鍵アドレス、秘密鍵及び閾値ｋを有し、複数のノードの中の各ノードは、各々の第１秘密と、及び当該ノードの各々の第１秘密が自由項としてセットされている各々のｋ－１次多項式関数を有し、前記複数のノードは、第１多項式関数を有する第１ノードを含む、前記方法が提供される。前記第１ノードによって実行される前記方法は、前記複数のノードの中の他のノードの夫々について、前記第１多項式関数内の変数を、当該他のノードに関連した値にセットし、その値に対する前記第１多項式関数の各々の結果を求め、該各々の結果を当該他のノードへ安全に送ることを含む。前記方法は、

少なくとも閾数の前記他のノードから、各々の第２秘密を受信し、該各々の第２秘密の夫々は、前記第１ノードに関連した第１値に対する当該他のノードの各々の多項式関数の結果である、ことと、

前記他のノードに対して、各々の結果の夫々の楕円曲線点乗算を送ることと、少なくとも前記閾数の前記他のノードから、該他のノードによって前記複数のノードのいずれかと共有される前記各々の第２秘密の楕円曲線点乗算を受信することと、
前記他のノードから受信された前記各々の第２秘密の和に基づき、前記秘密鍵の第１秘密鍵シェアを形成及び格納することと、
前記各々の第２秘密の前記受信された楕円曲線点乗算の和に基づき、且つ、多項式補間を用いて、前記秘密鍵に対応する公開鍵を決定することと
を更に含む。
【０００９】
いくつかの実施において、各々の多項式関数の夫々は、その各々のノードによって独立して選択され、各々の多項式関数の夫々は、係数の異なる組を有する。多項式は、次の式
【数１】

$$f_i = \sum_{t=0}^{(k-1)} a_t x^t \mathrm{mod} n$$

により表され得る。
【００１０】
いくつかの実施において、前記方法は、前記他のノードに対して、前記第１多項式関数の各係数の楕円曲線点乗算を送ることを含む。前記方法は、少なくとも前記閾数の前記他のノードから、それらの各々の多項式関数の各係数の楕円曲線点乗算を受信することを更に含んでもよい。また、前記方法は、前記他のノードから受信された前記各々の第２秘密を妥当性確認することを更に含んでもよい。
【００１１】
前記第１秘密鍵シェアは、次の式
【数２】

$$d_{A(i)} = \sum_{h=1}^{j} f_h(i) \mathrm{mod} n$$

により表される和を決定することから取得され得る。ここで、f_iは、ｉ番目のノードの多項式関数であり、ｎは、楕円曲線の整数位数であり、ｘは、当該多項式関数内の変数であり、a_tは、当該多項式関数のための係数の組であり、ｔは、インデックスである。
【００１２】
いくつかの実施において、前記方法は、ジョイントゼロ秘密シェアのディーラーなし秘密分配に前記他のノードとともに協調して参加することを通じて前記第１秘密鍵シェアを変更することを含み、これによって、前記第１ノードは、第１ゼロ秘密シェアを受信し、該第１ゼロ秘密シェアを加えることによって前記第１秘密鍵シェアを変更して、変更された第１秘密鍵シェアを形成及び格納する。
【００１３】
前記公開鍵を決定することは、前記各々の第２秘密の前記受信された楕円曲線点乗算の複数の既知の値から前記多項式関数の係数を決定することによって、前記多項式関数を決定することを有してよい。
【００１４】
前記多項式関数を決定することは、誤り訂正アルゴリズムを実行することを有してよい。
【００１５】
前記多項式関数を決定することは、バーレカンプ－ウェルチ（Berlekamp-Welch）復号化アルゴリズムを実行することを有してよい。

【００１６】
前記多項式関数を決定することは、
誤り位置（error locator）多項式関数、及び前記多項式関数と前記誤り位置多項式関数との積である第２多項式関数とを定義することと、
前記各々の第２秘密の前記受信された楕円曲線点乗算の複数の既知の値から前記第２多項式関数及び前記誤り位置多項式関数の係数を決定することと、
前記第２多項式関数及び前記誤り位置多項式関数から前記多項式関数を決定することと
を有してよい。
【００１７】
他の態様で、本開示は、閾ボールトに関するメッセージのためのデジタル署名を安全に生成する、コンピュータにより実施される方法であって、前記閾ボールトは、公開鍵アドレス、秘密鍵及び閾値を有し、複数のノードの中の各ノードは、各々の秘密鍵シェア、一時鍵シェア、乗算マスクシェア、第１加算マスクシェア、及び第２加算マスクシェアを有する、前記方法を開示する。前記複数のノードの中の第１ノードによって実行される前記方法は、
前記乗算マスクシェアによって及び前記第１加算マスクシェアによってマスキングされた前記一時鍵シェアから二重マスク化された鍵シェアを、並びに前記乗算マスクシェアの楕円曲線点乗算を生成及び共有することと、
少なくとも閾数の他のノードから、それらの各々の二重マスク化された鍵シェア、及びそれらの乗算マスクシェアの各々の楕円曲線点乗算を受信することと、
前記二重マスク化された鍵シェアに対する多項式補間及び前記乗算マスクシェアの前記楕円曲線点乗算に対する多項式補間を用いて第１署名成分を決定することと、
前記第２加算マスクシェアによってマスキングされた前記メッセージ、前記秘密鍵シェア、前記第１署名成分、及び前記一時鍵シェアに基づき、第２署名成分シェアを決定及び公開することと
を含む。少なくとも前記閾数の前記他のノードの各々の第２署名成分シェアから、第２署名成分が取得される。
【００１８】
いくつかの実施において、前記方法は、ディーラーなし秘密分配を用いて、前記第１ノードの秘密鍵シェア、一時鍵シェア、及び乗算マスクシェアを生成することを含む。前記方法は、ジョイントゼロ秘密シェアを用いて、前記第１ノードの第１加算マスクシェア及び第２加算マスクシェアを生成することを更に含む。いくつかの実施において、前記二重マスク化された鍵シェアを生成することは、

$$\theta_i = D_{k(i)}\alpha_i + \beta_i \bmod n$$

を決定することを含み、θ_iは、前記二重マスク化された鍵シェアであり、$D_{k(i)}$は、前記第１ノードの一時鍵シェアであり、α_iは、前記第１ノードの乗算マスクシェアであり、β_iは、前記第１ノードの第１加算マスクシェアであり、ｎは、楕円曲線の整数位数である。
【００１９】
いくつかの実施において、前記二重マスク化された鍵シェアに対する前記多項式補間は、乗算的にマスク化された一時鍵をもたらし、前記乗算マスクシェアの前記楕円曲線点乗算に対する前記多項式補間は、乗算マスクの楕円曲線点乗算をもたらし、前記第１署名成分を決定することは、前記乗算マスクの楕円曲線点乗算と前記乗算的にマスク化された一時鍵の逆数との積を決定することを含む。
【００２０】
いくつかの実施において、前記第２署名成分シェアs_iを決定することは、

$$s_i = D_{k(i)}(e + d_{A(i)}r) + c_i \bmod n$$

を決定することを含み、$D_{k(i)}$は、前記第１ノードの一時鍵シェアであり、ｅは、前記メッセージのハッシュであり、$d_{A(i)}$は、前記第１ノードの秘密鍵シェアであり、ｒは、前記第１署名成分であり、c_iは、前記第２加算マスクシェアであり、ｎは、楕円曲線

の整数位数である。

【００２１】

更に、別の実施において、前記方法は、少なくとも前記閾数の前記他のノードの前記第２署名成分シェアに対する多項式補間により前記第２署名成分を決定することを含む。いくつかの場合に、前記方法は、前記メッセージを含むトランザクションに対して前記第１署名成分及び前記第２署名成分を付加し、該トランザクションをブロックチェーンネットワークへサブミットすることを含む。

【００２２】

更なる態様で、本願は、コンピュータにより実施される、マルチパーティ計算の方法であって、

複数の第１ノードの間で複数の秘密のシェアを分配し、夫々の前記秘密は、各々の多項式関数の自由項であり、夫々の前記シェアは、前記第１ノードに関連した前記多項式関数の各々の値であり、それにより、前記秘密は、閾数の前記シェアの多項式補間によってアクセス可能であり、前記閾数のシェアがない場合にアクセス不可能である、ことと、

第２ノードで、複数の前記第１ノードから、計算された秘密の各々のシェアを受信し、夫々の前記第１ノードは、該第１ノードに割り当てられている各々の複数の秘密のシェアに対して少なくとも１の所定の計算を実行することによって、前記計算された秘密のそのシェアを計算する、ことと、

前記第２ノードで、前記計算された秘密の少なくとも閾数の前記シェアの多項式補間によって、前記計算された秘密を決定することと

を有する前記方法を開示する。

【００２３】

これは、第１ノードが個々の秘密又は計算された秘密を決定することができないように、計算が行われることを可能にするという利点をもたらす。

【００２４】

各多項式関数、次の式

【数３】

$$f_i = \sum_{t=0}^{(k-1)} a_t x^t \bmod n$$

により表されてよく、ｆ ｉは、ｉ番目のノードの多項式関数であり、ｎは、楕円曲線の整数位数であり、ｘは、当該多項式関数内の変数であり、ａ ｔは、当該多項式関数のための係数の組であり、ｔは、インデックスである。

【００２５】

前記方法は、複数のノードに対して、複数の前記多項式関数の各係数の楕円曲線点乗算を送ることを更に有してよい。

【００２６】

前記方法は、複数のノードに対して、該複数のノードの夫々に関連した少なくとも１の前記多項式関数の値の楕円曲線点乗算を送ることを更に有してよい。

【００２７】

前記方法は、少なくとも１の前記ノードに関連した少なくとも１の前記多項式関数の値の楕円曲線点乗算と、前記多項式関数の係数の楕円曲線点乗算の和との間の一貫性を検証することを更に有してよい。

【００２８】

これは、不正直な参加者がより迅速に識別され、多項式補間に寄与することから除外されることを可能にするという利点をもたらす。

【００２９】

前記計算された秘密のシェアを決定することは、

μ ｉ＝ｘ ｉ・ｙ ｉｍｏｄｎ

を決定することを含んでよく、x_iは、第１秘密のシェアであり、y_iは、第２秘密のシェアであり、ｎは、楕円曲線の整数位数である。

【００３０】

前記計算された秘密のシェアを決定することは、

$$x_i + y_i \bmod n$$

を決定することを含んでよく、x_iは、第１秘密のシェアであり、y_iは、第２秘密のシェアであり、ｎは、楕円曲線の整数位数である。

【００３１】

前記計算された秘密のシェアを決定することは、

$$x_i^{-1} \bmod n$$

を決定することを含んでよく、x_i^{-1}は、第１秘密のシェアの、ｎを法とする逆数であり、ｎは、楕円曲線の整数位数である。

【００３２】

前記方法は、ゼロに等しい多項式関数の自由項であるゼロ秘密のシェアを加えることによって、前記計算された秘密の変更されたシェアを供給して、前記計算された秘密が、閾数の前記変更されたシェアの多項式補間によってアクセス可能であり、前記閾数のシェアがない場合にアクセス不可能であるようにすることを更に有してよい。

【００３３】

これは、更なる安全性をプロセスに加えるという利点をもたらす。

【００３４】

前記計算された秘密を決定することは、該計算された秘密のシェアの複数の既知の値から多項式関数の係数を決定することによって、前記計算された秘密のシェアに対応する前記多項式関数を決定することを有してよい。

【００３５】

前記多項式関数を決定することは、誤り訂正アルゴリズムを実行することを有してよい。

【００３６】

前記多項式関数を決定することは、バーレカンプ－ウェルチ復号化アルゴリズムを実行することを有してよい。

【００３７】

前記多項式関数を決定することは、

誤り位置多項式関数、及び前記計算された秘密のシェアに関連した前記多項式関数と前記誤り位置多項式関数との積である第２多項式関数とを定義することと、

前記計算された秘密のシェアの複数の既知の値から前記第２多項式関数及び前記誤り位置多項式関数の係数を決定することと、

前記第２多項式関数及び前記誤り位置多項式関数から前記計算された秘密のシェアに関連した前記多項式関数を決定することと

を有してよい。

【００３８】

更なる態様で、本願は、複数のノードの間で公開－秘密鍵暗号化システムの秘密鍵のシェアを分配する、コンピュータにより実施される方法であって、前記複数のノードの夫々は、各々の第１秘密、当該ノードの各々の第１秘密が自由項としてセットされた各々の第１多項式関数、及び秘密鍵の各々の第１シェアを有し、前記秘密鍵は、前記第１秘密の和であり且つ第２多項式関数の自由項であり、前記秘密鍵は、少なくとも閾数の前記第１シェアの多項式補間によってアクセス可能であり、前記閾数に満たない前記第１シェアにはアクセス不可能であり、前記複数のノードに含まれる第１ノードによって実行される当該方法は、

その各々の自由項としてセットされたゼロを有する各々の第３多項式関数を選択することと、

前記複数のノードの各他のノードについて、前記第３多項式関数内の変数を、当該他のノードに関連した値にセットし、その値に対する前記第３多項式関数の各々の結果を決定

し、該各々の結果を当該他のノードへ安全に送ることと、
　各他のノードから、各々の第２秘密の各々のシェアを受信し、該シェアは、前記第１ノードに関連した第１値に対する当該他のノードの各々の第３多項式関数の結果である、ことと、
　前記秘密鍵の各々の第２シェアを形成及び格納し、該第２シェアは、前記各々の第１シェアの和及び前記第２秘密の前記受信されたシェアの和である、ことと
　を有する前記方法を開示する。
【００３９】
　これは、秘密鍵自体を更新する必要なしに秘密鍵のシェアが更新されることを可能にし、それによって、悪意のある又は反応しない参加者が除外されることを可能にするという利点をもたらす。
【００４０】
　前記方法は、前記他のノードに対して、前記各々の結果の夫々の楕円曲線点乗算を送ることと、前記他のノードに対して、前記第１ノードに関連した前記第３多項式関数の係数の楕円曲線点乗算を送ることとを更に有してよい。
【００４１】
　前記方法は、各他のノードから、そのノードに関連した前記結果の楕円曲線点乗算を受信することと、各他のノードから、そのノードに関連した前記第３多項式関数の係数の楕円曲線点乗算を受信することとを更に有してよい。
【００４２】
　これは、悪意のある又は反応しない参加者がより容易に識別されることを可能にするという利点をもたらす。
【００４３】
　前記方法は、少なくとも１の他のノードに関連した前記結果の一貫性を、その、他のノードに関連した前記第３多項式関数の係数により確認することを更に有してよい。
【００４４】
　前記方法は、メッセージ及び前記秘密鍵の前記第２シェアに基づきデジタル署名のシェアを生成することを更に有してよい。
【００４５】
　前記方法は、少なくとも閾数のノードから、メッセージ及び前記秘密鍵の各々の第２シェアに基づくデジタル署名の各々のシェアを受信することと、前記デジタル署名の少なくとも閾数のシェアの多項式補間によってデジタル署名を生成することとを更に有してよい。
【００４６】
　本願の更なる態様によれば、上記の方法を実行するコンピュータ実装システムが提供される。
【００４７】
　本願のこれら及び他の態様は、本明細書で記載される実施形態から明らかであり、それらを参照して説明される。
【００４８】
　これより、本願の実施形態が、単なる例として、貼付の図面を参照して記載される。
【図面の簡単な説明】
【００４９】
【図１】例となる楕円曲線デジタル署名プロセスの部分を表す。
【図２】例となる楕円曲線デジタル署名プロセスの更なる部分を表す。
【図３】例となる方法によって分配されるシェアの再構成階層を表す。
【図４】閾ボールトの実施に関係があるシェアの分配の例を図式的に示す。
【図５】閾ボールトを用いてデジタルアセットを受信及び分配する方法の一例をフローチャート形式で示す。
【図６】閾ボールトを生成する方法の例をフローチャート形式で示す。
【図７】閾ボールトからアセットを分配するためのデジタル署名を生成する方法の例をフ

ローチャート形式で示す。
【図８】ブロックチェーンノードのブロック図を例示する。
【図９】本発明の、閾ボールトを用いてマルチパーティ計算を実行する方法の例をフローチャート形式で示す。
【図１０】本発明を具現する方法において多項式補間を実行するバーレカンプ－ウェルチ復号器を示す。
【発明を実施するための形態】
【００５０】
＜概要＞
本発明は、不正アクセスから保護される必要があるコンピュータシステム及びリソースの安全性を高めるために使用され得る新規且つ独創的な技術を提供する。以下の記載は、ビットコインを含む暗号通貨に関連したシステムに関係がある実施、使用ケース及び実例を提供する。しかし、本発明は、より広い適用性を有しており、他のタイプのシステム及びコンピュータに基づくリソースを守ために使用されてよく、本発明はこの点で制限されないことに留意することが重要である。
【００５１】
本発明の実施形態は、ディーラーを伴わない展開のための能力と結合された、グループに基づく閾値暗号化の付加を可能にすることによって、改善された安全性をもたらす。実施形態はまた、メッセージの非インタラクティブな署名をサポートし、且つ、個人及びグループに分配され得るシェアへの秘密鍵の分割を提供する。更に、本発明は、不正行為又は攻撃の如何なる脅威も最小限としながら如何なる中央集権的制御の必要性も排除する分散型鍵生成システムを構築する解決法を提供する。ＥＣＤＳＡに対するＤＳＡの閾値に基づく解決法の適用において、本発明は、如何なる単一障害点も緩和する完全に分配的な署名システムを提供する。
【００５２】
ビットコイン又は代わりとなるものに関して使用される場合に、それは、暗号通貨を展開する無限に拡張可能であって且つ安全な手段にＣＬＴＶ及びマルチシグウォレット（multisig wallets）を関係させる検索方式と結合可能である。グループ及びリングに基づくシステムによれば、本発明は、発行されたトランザクションに対してブラインド署名を実装するために利用可能である。
【００５３】
本願は、ビットコインと完全互換である、閾値に基づく、ディーラーなしの秘密鍵分配システムを開示する。システムは、ビットコインウォレット内で展開される従来の個別署名システムから離れるグループ署名スキームを足場とする。展開される場合に、システムは、拡張性があり且つ堅牢であって、エラー及び悪意のある敵に耐性がある。システムは、ディーラーあり及びディーラーなし両方のシステム並びに大いに柔軟な分配組み合わせでの展開を支援する。
【００５４】
個々のパーティは、単独の参加者として又は共同して、彼らの保護された鍵スライスのスライスを、安全性及び回復力のためにマシン全体に、又は役割及びアクセス制御リストの投票閾値に基づいた展開のためにグループで分配するディーラーの役割を果たすことができる。
【００５５】
どの程度までスライスが分割可能であるかの深さに制限はない。この複雑性の問題は、個々の展開に対して重み付けされる必要がある。このようにして、署名及びトランザクションの記録は、この方法を用いて、グループ内からのものであっても、その後の書類で提示される拡張子を持った全ての外部の参加者から隠され得るということで、本発明の実施形態は、ユーザに対する偽名を用いた保護（pseudonymous protection）の標準を高めながら、ある程度の匿名性及びもっともらしい否認（plausible deniability）をビットコイントランザクションに導入する。

【００５６】
Ibrahim et al.［２００３］は、初期のロバストな閾値ＥＣＤＳＡスキームを開発した。続くプロトコルは、Gennaro et al.［１９９６］によって紹介された閾値ＤＳＳの楕円曲線式を形成するものの更なる拡張である。
【００５７】
【表１】
表１　定義

m	ビットコイントランザクションを含むメッセージ
$e = H(m)$	メッセージのハッシュ
CURVE	展開される楕円曲線及び体（E と要約される。）
G	楕円曲線基底点 この点は、素数位数 n が大きい楕円曲線の生成元である。
n	n×G＝∅であるような G の整数位数 これは、楕円曲線を満足する有理点の数として定義され、曲線 E の位数を表す。
k	鍵分割アルゴリズムのための閾値 この値は、鍵を回復するのに必要な鍵の数を表す。秘密は、(k-1)個のシェア又はそれ以下について安全であり、従って、k 個のシェアを用いて取得される。
d_A	インターバル$[1,(n-1)]$においてランダムに選択される秘密鍵整数
Q_A	曲線点$Q_A = d_A \times G$から導出される公開鍵
×	スカラーによって楕円曲線点乗算を表す。
j	スキーム内の参加者の数

群数学（group mathematics）の使用は、ＲＳＡ及びＤＳＡスキームから秘密の隠蔽におけるShamir［１９７９］の研究並びにFeldman［１９８７］及びPedersen［１９９２］の研究を拡張する検証可能秘密分散スキーム（verifiable secret sharing scheme）（ＶＳＳ）の構築を可能にし、それにより、それは、ビットコイン［Koblitz，１９９８］のような、ＥＣＣ及びＥＣＤＳＡに基づく署名システム内で使用され得る。本発明の実施形態は、悪意のある敵に耐性があり、停止し（halting）、傍受に対して堅牢である。
【００５８】
本開示は、どのパーティも秘密鍵を知らない場合にＥＣＤＳＡ署名の共同署名を可能にする方法の提示から始まる。更に、それは、秘密鍵を変更する必要なしに、秘密鍵対が更新及びリフレッシュされることを可能にする。これは有意な技術的利点である。
【００５９】
既存の解決法は全て、信頼できるパーティを必要とする。本発明を利用することはまた、代替ブロックチェーン又はサイドチェーンの要件を陳腐化しながら、ビットコインブロックチェーン上で直接解決され得る電子ノード発行の真に分散された方法へと中央集権型システムからChaum［１９８３］の研究を拡張することを可能にする。

【００６０】
上記の生成元Ｇに関して、ビットコインはｓｅｃｐ２５６ｋ１を使用する。これは，ビットコインで使用されるＥＣＤＳＡ曲線のパラメータを定義し、Standards for Efficient Cryptography（ＳＥＣ）（Certicom Research，http://www.secg.org/sec2-v2.pdf）から参照され得る。
【００６１】
＜信頼の問題＞
全ての既存のシステムは、ある程度の信頼を必要とする。これまで、ビットコインは、世界から隔絶されているセキュアシステムを使用した秘密鍵の保護を必要としてきたが、これは達成するのが困難であることが分かっている。注目すべきは、ビットコインが交換又は保管され得るシステムは、中央当局（centralised authority）に対する信頼を必要とする点である。本発明は、この要件を完全に変えて、プロトコルの中核的要件のいずれも変えることなく、ビットコイン内の鍵生成及びメッセージ署名プロセスを分散化及び分権化する。本明細書で言及される方法は、ビットコインプロトコルを変更せずに実装され得、実際には、署名されたメッセージの解析を通じてこのプロセスが展開されているかどうかを判定する方法はない。
【００６２】
ビットコインのための分散型署名スキームを構築することにおいて、本発明は、人々又はシステムのグループが、個人が自分で署名を生成することができないように鍵を安全に保持することを可能にする。拡張される場合に、このスキームはまた、ビットコインの秘密鍵自体及びシェアの夫々の安全な回復も可能にする。グループにより生成された署名は、既存のプロトコルから生成されたものと区別不可能である。そのようなものとして、署名検証は、それが標準のトランザクションを使用して一人の署名者により成立されたかのようなままである。
【００６３】
ｎ人の参加者のグループ又はm個の参加者グループによって秘密鍵が共有されるということで、信頼性のこのような向上が実現される。トランザクションの署名には閾数の参加者が必要とされ、最小閾値を満足する参加者又は参加者グループの任意の連合が署名動作を実行することができる。重要なことに、このプロトコルは、個人又はグループが参加者の連合を作ろうと試みることができる一括処理として、又は同期して成立され得る。
【００６４】
＜背景研究＞
最初に、Shamir［１９７９］が、鍵の分散管理を可能にするディーラーに基づいた秘密分散スキームを紹介した。このスキームに付随する問題は、検証不可能なディーラーを信頼する必然に由来する。このような形式のスキームは、本発明と完全互換であり、本明細書で言及されるプロセスを通じて生成される個々の鍵スライスのグループ分配のために使用され得る。
【００６５】
＜ジョイントランダム秘密分散（ＪＲＳＳ）［Pedersen，１９９２］＞
このプロシージャの明言された目的は、どの参加者も秘密を知ることなく参加者グループが協調して秘密を共有し得る方法を構築することである。各参加者は、彼らの局部的秘密としてランダムな値を選択し、このランダムな値から導出される値を、グループによるシャミアの秘密分散スキーム（Shamir’s secret sharing scheme）（ＳＳＳＳ）を用いて分配する。次いで、各参加者は、自身を含む参加者から受け取られた全てのシェアを足し合わせる。この和がジョイントランダム秘密シェア（joint random secret share）である。結合された秘密値の秘密性を保つには、一人の正直な参加者によって提供されるランダムさで十分である。この状態は、たとえ（ｎ－１）人の他の全ての参加者がランダムでない秘密値を意図的に選択するとしても、依然として保たれる。
【００６６】
＜ジョイントゼロ秘密分散（ＪＺＳＳ）［Ben-Or，１９８８］＞

ジョイントゼロ秘密分散（Joint Zero Secret Sharing）（ＪＺＳＳ）はジョイントランダム秘密分散（Joint Random Secret Sharing）（ＪＲＳＳ）に似ているが、各参加者がランダムな値に代わるものとして０を共有する点で相違する。この技術により生成されるシェアは、ＪＲＳＳアルゴリズムにおける如何なる潜在的な弱点も解消することに役立つ。

【００６７】

Desmedt［１９８７］は、グループ指向の暗号化の概念を紹介した。このプロセスにより、参加者は、参加者の中の選択された一部しかメッセージを暗号解読することができないようにして人々のグループへメッセージを送ることが可能となった。システムにおいて、メンバーは、送信者が公開鍵を使用してメンバーを知る必要がある場合に知られていることを伝えられ、グループは、メンバーから独立して保持されるそのグループのための単一の公開鍵しかない場合に匿名である。本発明は、両方の方法を統合し、既知及び匿名の送信者及び署名者が同時にグループ内に存在することを可能にする。

【００６８】

＜本発明＞

大きい位数の素数を有する任意の楕円曲線（ＣＵＲＶＥ）、及び素数体Z_pにわたって定義される位数ｎの基底点$G \in CURVE(Z_p)$について、如何なる参加者も閾数に満たないシェアから元の秘密鍵を再生することがでないように、鍵シェアへのＥＣＣ秘密鍵の安全な分配及びその使用を可能にするシステムが構築され得る。

【００６９】

未知の整数d_Aについて、$1 \leqq d_A \leqq (n-1)$の場合に、$Q_A = d_A \times G$を前提として、d_Aを計算することは極めて困難であることが知られている［Kapoor，２００８］。

【００７０】

本発明の基本的な技術は、閾値暗号化の適用により得られる。このシステムにおいて、ＥＣＤＳＡ秘密鍵は、潜在的なものとしてしか存在せず、どのシステムでも再生される必要がない。それらの複数のシェアの夫々は、拡張可能であって且つグループ及び個人の両方の署名フォーマットの導入を可能にする方法で、複数の参加者［$p_{(i)}$］に分配される。よって、署名プロセスは、ビットコイン内で展開されるものとは異なる。このプロセスでは、調整参加者$p_{(c)}$が、グループに分配されるトランザクション及びメッセージ署名を生成する。各参加者は、部分署名を計算するか又はパスするかによって、その秘密鍵シェアの使用に投票することができる。

【００７１】

実際に、パスすることは、反対票（no vote）と同等である。調整参加者$p_{(c)}$は、応答を照合し、最低限の閾数の部分署名を受け取った場合にそれらを結合して完全な署名を形成する。

【００７２】

調整参加者$p_{(c)}$は、反対票を受け入れ、他のパーティからのヌル値に基づき計算を行うか、あるいは、パーティに働きかけて、メッセージに署名するように説得するか、いずれかを行うことができる。プロトコルは、設定された調整者（coordinator）により実装され得るか、あるいは、任意の個人又はグループが、この役割を形成し、署名されるべき閾グループへのトランザクションを提案することができる。本発明は、完全に分散されたＥＣＤＳＡ秘密鍵生成アルゴリズムを提供するようにIbrahim et al.［２００３］の研究を拡張する。本願はまた、ビットコインとともに使用される分散型鍵再分散アルゴリズム及び分散型ＥＣＤＳＡ署名アルゴリズムを提示する。鍵再分散アルゴリズムは、新しいものを支持して現在存在する全ての秘密鍵シェアを無効にするために、又は新しい参加者への秘密鍵シェアの再割り当てのために、使用されてよい。このプロトコルは、ＥＣＤＳＡ秘密鍵だけでなく秘密鍵シェアの分散にも及ぶ。この結果は、シェアが構成され、グループ処理として採決され得ることを意味する。

【００７３】

本発明は、信頼できるサードパーティが存在するための全ての要件を取り除く。その結

果、既存のプロトコルと完全互換であって且つ更なる拡張性を可能にしながら残りの単一障害点を全て解消するビットコイン用の新しいオーバーレイ及びウォレットを構築することが可能である。本発明はまた、ブラインド署名の導入を可能にするように拡張することもできる。

【００７４】

本発明は、秘密鍵をメモリにロードする必要がないので、信頼できるサードパーティの必要性を取り除くだけでなく、更には、広範囲の一般的な攻撃を排除する。プロトコルは、必要とされるシェア数及びシェアの分配が使用ケース、経済シナリオ及びリスク要件によって決定されることを可能にするように拡張可能である。

【００７５】

本発明は、全てのサイドチャネル攻撃、従って、あらゆるキャッシュタイミング攻撃を軽減する。このシステムは、Gennaro et al.［１９９６］の研究を取り込み、それをＤＳＳから拡張して、如何なるＥＣＤＳＡに基づくアプリケーションでも成功裏に使用され得るようにする。

【００７６】

＜ＥＣＤＳＡ＞

ビットコインは、ｓｅｃｐ２５６ｋ１曲線に基づきＥＣＤＳＡを使用する。ＥＣＤＳＡは、最初に、楕円曲線暗号化（elliptic curve Cryptography）（ＥＣＣ）を使用したDiffie-Hellmanに基づく鍵交換のための要件を変更するように、２００３年にＮＩＳＴによって標準化された［ＮＩＳＴ］。ＥＣＣの生成は、他の公開／秘密鍵システムと比べた場合に、鍵サイズ及び処理電力の低減により特に重要であった。劣指数的（sub-exponential）時間アルゴリズムはＥＣＤＬＰについて発見されていない。ＥＣＤＬＰは、扱いにくいことが知られており、楕円曲線離散対数問題（elliptic curve discrete logarithm problem）を指す［Johnson，２００１］。

【００７７】

本願を通じて使用されるパラメータは、上記の表１で提供されている。

【００７８】

＜安全性の懸案事項＞

システムは、ビット内の現在の制限であるＥＣＤＳＡの安全性によって制限される。現在、ＥＣＤＳＡは、秘密鍵が安全に展開され得る場合に安全なままである。本発明は、鍵の再生成イベントの前に閾数の参加者が危殆化されていることを要求する閾値まで、サイドチャネル攻撃及びメモリ開示攻撃を軽減する。更に、危殆化されていない閾値の大多数は、危殆化された参加者を閾値未満で識別することができる。

【００７９】

停止性問題（Halting problem）

サービスの中断は、参加者に対してサービス拒否攻撃を仕掛けようと試みる悪意のある敵が加担する可能性がある攻撃の一種である。この攻撃により、参加者は、解析に処理時間を費やすことになる無効な署名、又はその後にドロップされることになる膨大なネットワークメッセージを受け取らざるを得ない。

【００８０】

ＥＣＣ又は署名暗号化（signcryption）に基づくＥＣＣのいずれかを用いて参加者へのメッセージを暗号化する要件は、このような攻撃経路（attack vector）を軽減する。攻撃者が無効な部分的に署名されたメッセージを送るには、参加者を既に危殆化している必要があるので、このような攻撃はもはや不要である。

【００８１】

ランダム（Randomness）

アルゴリズム２は、たとえ（ｎ－１）人の参加者がランダムな値を選ぶことができないとしても、十分なランダムさが導入されるシナリオを提供する。このプロトコルに追加できるものは、署名及び鍵再生成プロセスへのランダムな値の導入のためにもっぱら設計されたグループオラクルの導入である。この任意のシナリオでは、鍵スライスの夫々が同じ

プロトコルを用いて生成され得る。例えば、ｍ　ｏｆ　ｎのプライマリスライス要件がある場合に、基礎となる鍵スライスの夫々も、ｍ’　ｏｆ　ｎ’閾値条件を用いて生成及び管理され得る。

【００８２】

このシステムを使用する参加者は、ランダムさをプロトコルに注入する以外何もしない外部オラクルを追加することができる。ｍ’個の鍵スライス（なお、ｍ’＜ｎ－１）を有するユーザは、彼らが保持する鍵スライスに基づき彼らの署名ソリューションを再現し処理することを選択するか、あるいは、ランダムさの導入のため以外に不要である外部オラクルを導入し得る。

【００８３】

各スライスは、ロバスト性及び安全性のために同様に分割され得る。鍵スライスは、ユーザが携帯電話機又はスマートカードのような外部デバイス及びコンピュータで実行されるソフトウェアプログラムでスライスを有するように分配されてよく、それにより、彼らが部分署名を生成するために、ソースの組み合わせは必要とされることになる。

【００８４】

一意のランダムな一時鍵（ephemeral key）D_kが生成されるか、あるいは、秘密鍵d_Aを再現するために情報を使用することが可能であることが重要である。

【００８５】

＜公開署名＞

このプロトコルによるトランザクション署名の主たる目的は、ビットコイントランザクションの分散署名を可能にすることである。ブロックチェーンに公開されたことがない如何なるトランザクションも参加者によって非公開で保持され得る。従って、調整参加者$p_{(c)}$が機会あるごとに、トランザクションに成功裏に署名するために、必要とされる水準の投票を達成することができなかった場合には、新しいビットコイントランザクションを生成することは不要である。鍵スライスがそれ自体閾値に対して安全である場合には、決済された如何なるトランザクションの所有権も安全なままである。

【００８６】

システムが適切に展開される場合に、危殆化される参加者を（ｋ－１）人までとする能力は、システムを、閾値未満の攻撃に対して安全なままとする。周期的な鍵再生成プロトコル（アルゴリズム２）と結合される場合に、本発明の実施形態は、サイドチャネル攻撃及びメモリ開示に耐えることができる。

【００８７】

＜本発明の方法及び実施＞

本発明の実施形態は、階層的な導出に基づいてＥＣＣを使用して参加者間で送られる必要がある秘密情報を暗号化するので［Wright，２０１６］、必要に応じて、危殆化されるか又は敵意を持たれる可能性がある参加者に対して妥当性確認が行われ得るように、全てのメッセージを、全ユーザへ送られる単一のパケットにまとめることが可能であり且つ望ましい。

【００８８】

署名生成は、調整参加者$p_{(c)}$によって提案される。デフォルトで、任意の鍵スライスが、調整参加者として機能することができ、要件は、プロトコルの個々の実施に帰着する。使用されるアルゴリズムは、以下で説明され、後の項目は、それらの展開に関して詳細を提供する。

【００８９】

＜アルゴリズム１　鍵生成＞

ドメインパラメータ（ＣＵＲＶＥ，濃度ｎ，生成元Ｇ）

入力：　ＮＡ

出力：　公開鍵Q_A

秘密鍵シェア$d_{A(1)}$，$d_{A(2)}$，・・・，$d_{A(j)}$

（ｊ）人の参加者からのｋ個のスライスの閾値について、構成される鍵セグメントd_A

（ｉ）は、参加者（ｉ）と、参加者（ｉ）が鍵に（従って、ビットコイントランザクションに）署名するために秘密をやり取りする残りのパーティである、参加者（ｈ）として指名される（ｊ－１）人の参加者とに関連付けて構成される。

【００９０】

・スキームにおいて、ｊは参加者の総数であり、このとき、ｋ≦ｊ、従って、ｈ＝ｊ－１。

・従って、（ｋ，ｊ）－閾値分散スキームが存在する。

【００９１】

アルゴリズム１のための方法は、次の通りである：

１）１≦ｉ≦ｊとして、（ｊ）の中の各参加者ｐ（ｉ）は、ＥＣＣ公開鍵（すなわち、この実施では、ビットコインアドレス）を他の全ての参加者とやり取りする。このアドレスは、グループ識別アドレスであり、如何なる他の目的にも使用される必要がない。

【００９２】

これは、国際特許出願ＷＯ２０１７／１４５０１６のプロセスから参加者の夫々の間の共有値に基づき導出されたアドレス及び鍵である点が留意されるべきである。

【００９３】

２）各参加者ｐ（ｉ）は、他の全てのパーティから秘密であるようにランダムな係数を用いて次数（ｋ－１）の多項式ｆｉ（ｘ）を選択する。

【００９４】

この関数は、多項式自由項として選択される参加者の秘密ａ０（ｉ）の形で第１秘密値を条件とする。この値は共有されない。この値は、ＷＯ２０１７／１４５０１６で開示されているように、導出された秘密鍵を用いて計算される。

【００９５】

ｆｉ（ｈ）は、点（ｘ＝ｈ）での値について参加者ｐ（ｉ）によって選択された関数ｆ（ｘ）の結果であると定義され、参加者ｐ（ｉ）の基本式は、関数：

【数４】

$$f_{(x)} = \Sigma_{\mathrm{p}=0}^{(k-1)} \; a_p x^{\mathrm{p}} \bmod n$$

として定義される。

【００９６】

この式で、ａ０は、各参加者ｐ（ｉ）の秘密であり、共有されない。

【００９７】

従って、各参加者ｐ（ｉ）は：

【数５】

$$f_{i(x)} = \Sigma_{\gamma=0}^{(k-1)} \; a_\gamma x^{\gamma} \bmod n$$

であるように、自由項ａ０（ｉ）がその参加者の秘密として定義されている次数（ｋ－１）の多項式として表される秘密に保たれた関数ｆｉ（ｘ）を有する。

【００９８】

３）各参加者ｐ（ｉ）は、上記の通りに、ＷＯ２０１７／１４５０１６で開示されているように、Ｐ（ｈ）の公開鍵を用いて参加者Ｐ（ｈ）∀ｈ＝｛１，・・・，（ｉ－１），（ｉ＋１），・・・，ｊ｝への第１シェアｆｉ（ｈ）を暗号化し、Ｐ（ｈ）が暗号解読するために値をやり取りする。

【００９９】

素数ｐの位数ｎの如何なる基底点Ｇ∈Ｅ（Ｚｐ）についてもｎ×Ｇ＝φであることが留意されるべきである。ビットコインの場合に、値は：

楕円曲線式： $y^2 = x^3 + 7$
素数モジュロ： $2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
＝FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F
基底点＝04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8
位数＝FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141
である。

【０１００】

そのようなものとして、（ｂ，b_1，b_2，・・・）として表現可能な整数の任意の組Ｂ：｛$b_i \in Z_n$｝について、ｂＧ＝［$b_1G + b_2G$＋・・・］ｍｏｄｐの場合に、ｂ＝［$b_1 + b_2$＋・・・］ｍｏｄｎ。更に、ｂＧ＝［$b_1 b_2$・・・］Ｇｍｏｄｐの場合に、ｂ＝［$b_1 b_2$・・・］ｍｏｄｎ。

【０１０１】

Z_nが体（field）であり、ｎを法とするラグランジュ補間がＥＣＣ秘密鍵として選択された値に対して有効に行われ得るとすれば、シャミアの秘密分散スキーム（ＳＳＳＳ）［５］がZ_nに対して実施され得るという結論に至る条件が存在する。

【０１０２】

４）各参加者$p_{(i)}$は、以下の値

【数６】

$$\text{a)}\quad a_\kappa^{(i)} G \quad \forall \kappa = \{0,\ldots,(k-1)\}$$

$$\text{b)}\quad f_i(h)G \qquad \forall h = \{1,\ldots,j\}$$

を全参加者へブロードキャストする。

【０１０３】

上記の式中の変数ｈに関連した値は、参加者$P_{(h)}$がスキームにおける第３の参加者に相当する場合にはｈ＝３であるように参加者$P_{(h)}$のポジションであることができ、あるいは、同様に、整数として参加者によって使用されるＥＣＣ公開鍵の値を表し得る。使用ケース及びシナリオは、いずれの実施についても存在する。後者の実施では、値ｈ＝｛１，・・・，ｊ｝は、個々の参加者の利用される公開鍵にマッピングされた値のアレイによって置換される。

【０１０４】

５）各参加者$P_{(h \neq i)}$は、受け取られたシェアと、各他の参加者から受け取られるものとの一貫性を検証する。すなわち：

【数７】

$$\Sigma_{k=0}^{(k-1)} h^k a_k^{(i)} G = f_i(h)G$$

そして、f_i（ｈ）Ｇは、参加者のシェアと一致する。

【０１０５】

６）各参加者$P_{(h \neq i)}$は、その参加者（$P_{(h \neq i)}$）によって所有されている受け取られたシェアが、他の受け取られたシェアと一致することを妥当性確認する：

【数８】

$$a_0^{(i)}G = \Sigma_{h\in \mathrm{B}} b_h f_i(\mathrm{h})G \qquad \forall P_{(h\neq i)}$$

実際に、このステップは、シェアｆｉ（ｈ）の楕円曲線暗号化されたバージョン（すなわち、ｆｉ（ｈ）Ｇ）に対して、ｆｉ（ｈ）の暗号化されていないバージョンに対して実行された場合に秘密値ａ０（ｉ）を回復する動作を実行して、Ｇａ０（ｉ）を回復することから成る。従って、シャミアの秘密分散スキームの場合に、係数ｂｈは、秘密をその対応するシェアから回復するのに必要なラグランジュ補間係数を表す。これに一貫性がない場合には、参加者はプロトコルを拒否し、再び開始する。

【０１０６】

７）このとき、参加者ｐ（ｉ）は各々、彼らのシェアｄＡ（ｉ）を、

【数９】

$$\mathrm{SHARE}(p_{(i)}) = d_{A(i)} = \Sigma_{\mathrm{h}=1}^{j} f_h(i) \bmod n$$

として計算する。ここで、Σｈ＝１ｉｆｈ（ｉ）ｍｏｄｎは、各参加者Ｐ（ｈ≠ｉ）から受け取られる各々の第２秘密値ａ０の第２シェアであり、そして、

ＳＨＡＲＥ（ｐ（ｉ））∈Ｚｎ　且つ　ｄＡ（ｊ）

このとき、

【数１０】

$$Q_A = Exp - Interpolate\left(f_1, \cdots, f_j\right) \triangleright \left[= G \times d_A\right]$$

演算Ｅｘｐ－Ｉｎｔｅｒｐｏｌａｔｅ（）は、楕円曲線暗号化されたシェアから楕円曲線暗号化された秘密を回復する演算として定義される。この演算は、次のように更に詳細に定義され得る。

【０１０７】

せいぜい（ｋ－１）個の値がヌル（ｎｕｌｌ）であり、残りの値がＧ×αｉの形をとるとして、｛ω１，・・・，ωｊ｝（ｊ≧（２ｋ－１））がセットされ、各αｉが何らかの（ｋ－１）次の多項式Ｈ（・）に存在するならば、その場合に、θ＝Ｇ×Ｈ（０）。

【０１０８】

この値は、θ＝Σｉ∈Ｖωｉ×λｉ＝Σｉ∈Ｖ（Ｇ×Ｈ（ｉ））×λｉβによって計算され得る。この場合に、Ｖは、正しいωｉ値の（ｋ）－サブセットであり、更には、λｉは、結果として起こるラグランジュ補間係数を表す。多項式は、バーレカンプ－ウェルチ復号器を使用することによって計算され得る。

リターン　（ｄＡ（ｉ），ＱＡ）

このとき、ｄＡ（ｉ）は、第３秘密値の第３シェアである。

【０１０９】

参加者ｐ（ｉ）は、このとき、署名を計算することにおいてシェアを使用する。この役割は、署名を収集するプロセスにおいて調整者となるいずれかの参加者又はパーティｐ（ｃ）によって行われ得る。参加者ｐ（ｃ）は変化してよく、トランザクションに署名するために十分なシェアを収集しようとする各試みにおいて同じパーティである必要がない。

【０１１０】

従って、秘密鍵シェアｄＡ（ｉ）←Ｚｎ＊は、他の参加者のシェアを知ることなしに生成された。

【０１１１】

＜アルゴリズム２　秘密鍵の更新＞

入力：　ｄＡ（ｉ）と表される秘密鍵ｄＡの参加者Ｐｉのシェア

出力：　参加者Ｐｉの新しい秘密鍵シェアｄＡ（ｉ）

アルゴリズム２は、秘密鍵を更新すること及びランダムさをプロトコルに加えることの両方のために使用され得る。

【０１１２】

［Wright，２０１６］のフォーマットの鍵を使用すると、このプロセスは、秘密鍵の再構成又は計算された存在さえなしで、階層的なサブキーの再計算をもたらすことができる。このようにして、正確に展開される場合に、過去に起こった如何なる大規模な不正行為又はデータベース窃盗も排除するビットコインアドレス及び秘密鍵スライスのヒエラルキーを構成することが可能である。

【０１１３】

１）各参加者は、その自由項としてゼロに従う次数（ｋ－１）のランダム多項式を選択する。これは、アルゴリズム１に類似しているが、参加者は、全ての他の参加者の選択された秘密がゼロであることを妥当性確認しなければならない。

【０１１４】

φＧ＝ｎＧ＝０であることが留意されるべきであり、このとき、０は、楕円曲線上で無限遠にある点である。

【０１１５】

この式を用いると、全てのアクティブな参加者は、関数：

【数１１】

$$a_0^{(i)}G=\varnothing \qquad \forall i=\{1,\cdots,j\}$$

を妥当性確認する。例えば、Feldman（１９８７）を参照されたい。

【０１１６】

ゼロシェアを生成：　$z_i \leftarrow Z_n^*$

２）$d_{A(i)}' = d_{A(i)} + z_i$

３）リターン：　$d_{A(i)}'$

このアルゴリズムの結果は、元の秘密鍵に関連する新しい鍵シェアである。このアルゴリズムの変形は、とり得るビットコインアドレスを変更する必要なしに新しい鍵スライスをもたらす再分散実行に従事するか、あるいは、第１アルゴリズムのランダムさを増大させることができる。このようにして、本発明は、基礎となる秘密鍵を変えずに秘密鍵シェアを追加的にマスキングすることをグループに可能にする。このプロセスは、基礎となるビットコインアドレス及び秘密鍵を変えずに、個々の鍵シェアの継続的な使用及び展開に関連した如何なる潜在的な鍵漏えいも最小限にするために使用され得る。

【０１１７】

＜アルゴリズム３　署名生成＞

ドメインパラメータ：　ＣＵＲＶＥ，濃度ｎ，生成元Ｇ

入力：　署名されるメッセージ　ｅ＝Ｈ（ｍ）

秘密鍵シェア　$d_{A(i)} \in Z_n^*$

出力：　署名　ｅ＝Ｈ（ｍ）に対して（ｒ，ｓ）$\in Z_n^*$

Ａ）分散型鍵生成

１）アルゴリズム１により一時鍵シェアを生成する：

$D_{k(i)} \leftarrow Z_n^*$

２）アルゴリズム１によりマスクシェアを生成する：

$\alpha_i \leftarrow Z_n$

３）アルゴリズム２によりマスクシェアを生成する：

$\beta_i, c_i \leftarrow Z_n^2$

次数２（ｋ－１）の多項式を用いてアルゴリズム２を２度実行することによって、それらのプロトコルで生成されるシェアは、

【数１２】

$$\left(\beta_1,\cdots,\beta_j\right)^{(2(k-1),j)} \leftrightarrow \beta \bmod n \text{ および } \left(c_1,\cdots,c_j\right)^{(2(k-1),j)} \leftrightarrow c \bmod n$$

と表される。これらは、加算マスク（additive masks）として使用される。マスキングされる数が次数（ｋ－１）の２つの多項式の積を含むということで、多項式は次数２（ｋ－１）でなければならない。これは、秘密を回復するために必要とされるシェアの必要数を２倍にする。

【０１１８】

βおよびｃのシェアは、次いで、参加者によって秘密に保たれる。

【０１１９】

Ｂ）署名生成

４）ｅ＝Ｈ（ｍ）　メッセージｍのハッシュを妥当性確認する。

【０１２０】

５）$\theta_i = D_{k(i)}\alpha_i + \beta_i \bmod n$及び$\omega_i = G \times \alpha_i$をブロードキャストする。

【０１２１】

６）

【数１３】

$$\mu = Interpolate(\vartheta_1,\ldots,\vartheta_n) \bmod n$$

$$\triangleright \left[= D_k \alpha \bmod n\right]$$

ここで、演算μ＝Interpolate（θ_1，・・・，θ_n）ｍｏｄｎは、シェアから秘密を回復する演算として定義される。この演算は、次のように更に詳細に定義され得る。

【０１２２】

大部分の（ｋ－１）がヌル（null）であり、且つ、全ての残余値が（ｋ－１）次多項式Ｆ（．）にあるように、｛θ_1，・・・，θ_n｝（ｊ≧（２ｋ－１））が集合を形成する場合に、μ＝Ｆ（０）。

【０１２３】

７）

【数１４】

$$\theta = Exp - Interpolate\left(\omega_1,\ldots,\omega_n\right)$$

$$\triangleright \left[= G \times \alpha\right]$$

８）（R_x，R_y）を計算する。ここで

【数１５】

$$r_{x,y} = \left(R_x, R_y\right) = \theta \times \mu^{-1}$$

$$\triangleright \left[= G \times D_k^{-1}\right]\Big]$$

９）$r = r_x = R_x \bmod n$

$r = 0$の場合に、再び（すなわち、初期分布から）開始する。

【０１２４】

１０）$s_i = D_{k(i)}(e + d_{A(i)} r) + c_i \bmod n$をブロードキャストする。

【０１２５】

１１）$S = \text{Interpolate}(s_i, \cdots, s_n) \bmod n$

ｓ＝０の場合に、アルゴリズム３を最初からやり直す（Ａ．１）。

【０１２６】

１２）リターン（ｒ，ｓ）

１３）ビットコインでは、スタンダードトランザクションを形成するように（ｒ，ｓ）対によりトランザクションを再構成する。

【０１２７】

図１０は、デジタルメッセージの部分署名を表す多項式関数を取得するための従来のバーレカンプ－ウェルチ復号器７０の新規の使用を示す。

【０１２８】

伝送データ内のエラーを訂正するためのバーレカンプ－ウェルチアルゴリズムの従来の使用では、メッセージｍは、符号器７２で連続するｋ個のバイトに分けられ、各バイト$c_0, c_1, \cdots, c_{k-1}$は、整数モジュロｐとして符号化される。次いで、メッセージは、多項式関数：

【数１６】

$$m(x) = \sum_{i=0}^{k-1} c_i x^i$$

によって表される。次いで、多項式関数ｍ（ｘ）の値は、（ｘ、ｙ）対の連続を生成するようにｘの多数の既知の値について決定され、次いで、それは、送信器７４によって受信器７６へ送られる。

【０１２９】

受信器７６で受信されたデータＭ（すなわち、受信されたメッセージ）は、元のメッセージを表す多項式関数上の点に対応する対（$a_1, b_1, \cdots, a_n, b_n$）を有する：

【数１７】

$$P(x) = m(x) = \sum_{i=0}^{k-1} c_i x^i$$

送信された（ｘ，ｙ）対のいくつかが送信中に破損したと考えられる場合に、誤り位置多項式関数が次のように定義され得る：

Ｐ（a_i）≠b_iの場合に、Ｅ（a_i）＝０

他の場合に、Ｅ（a_i）≠０

積多項式関数（product polynomial function）Ｑ（a_i）がＱ（a_i）＝b_iＥ$_i$（a_i）と定義される場合に、夫々の受信される（a_i，b_i）対について、b_iの値が破損しているかどうかにかかわらず、Ｐ（a_i）≠b_iの場合にＥ（a_i）＝０であるから、Ｑ（a_i）＝b_iＥ$_i$（a_i）＝Ｐ（a_i）Ｅ$_i$（a_i）。

【０１３０】

（a_i，b_i）のｎ個の既知の値について、Ｅ（a_i）は次数ｅの多項式関数であり、Ｐ（a_i）は次数（ｋ－１）の多項式関数であるから、Ｑ（a_i）は次数（ｅ＋ｋ－１）の多項式関数である。従って、（a_i，b_i）の既知の値は、線形系として表され得る：

【数１８】

$$Q(a_i) = \sum_{j=0}^{e+k-1} q_i a_i^j = b_i \sum_{j=0}^{e} e_j a_i^j = b_i E_i(a_i)$$

線形系は、２ｅ＋ｋ－１個の未知の項（Ｅ（ｘ）からｅ及びＱ（ｘ）からｅ＋ｋ－１）を

含み、その結果として、Q（a_i）及びE（a_i）の係数は、n≧２e＋k－１の場合に決定され得る。Q（a_i）及びE（a_i）が決定され得る場合に、元のメッセージm（x）を回復するためにP（a_i）を決定することが可能である。

【０１３１】

従って、バーレカンプ－ウェルチ復号器７０は、多項式関数上の点を表す対を入力として受け取り、多項式関数を出力する。従って、復号器７０は、多項式によって表される閾数のシェアからその多項式を決定するために、本発明ではラグランジュ補間に代わるものとして使用され得る。

【０１３２】

＜モデル－閾値ＥＣＤＳＡ（Ｔ，ＥＣＤＳＡ）＞

本発明の実施形態に従って、参加者として指名されるn個のグループ又は個人のシステムが認められる。各プレイヤーは、単独の参加者若しくはグループ又はそれらの組み合わせとして個人であることができる。参加者$p_{(i)}$は、一般的に導出された公開鍵計算を用いて識別子（identity）に対してマッピングされてよく、あるいは、参加者$p_{(i)}$は、個人にマッピングされ直すことなしに、このプロトコルのためにのみ使用される参加者の公開鍵を用いて偽名エンティティ（pseudonymous entity）として残されてもよい。

【０１３３】

本発明は、スキームの有効なプレイヤー及びメンバーとしての他の参加者の認識を可能にすると同時に、グループ内のメンバーが識別されないままであることを可能にする専用のブロードキャストチャネルを導入する。メッセージが参加者$p_{(i)}$からブロードキャストされるとき、グループ内のメンバーは、鍵に関連するエンドユーザ又は個人を識別することが必ずしもできなくても、メッセージを認定されたパーティに由来するものとして認める。また、そのようなシステムが保証されているならば、鍵の識別子を個人とリンクすることも可能である。

【０１３４】

プロセスフローは、次のように要約される。

【０１３５】

図１中：

ステップ１）

図１を参照されたい。

【０１３６】

ステップ２）

Ｐｃが生の（raw）トランザクションをグループへ送る。これが妥当性確認する場合に（すなわち、生のトランザクションが署名されるハッシュと一致する場合に）、参加者は、それに署名することによって投票する。

【０１３７】

ステップ３）

肯定判断である場合に、各参加者は、部分署名されたトランザクションを返す。

【０１３８】

図２を参照されたい。

【０１３９】

ステップ４）

閾数の部分署名されたトランザクションが受け取られる場合に、Ｐｃ（又は任意の他の参加者）は完全な署名を再構成する。

【０１４０】

ステップ５）

Ｐｃは、署名されたビットコイントランザクションとしてトランザクションをブロードキャストする。

【０１４１】

メッセージ署名の計算は、変化しない個人によって、又は一時的なブロードキャストパ

ーティを通じて、開始され得る。プロトコル調整者の役割は、署名を収集するプロセスにおいて調整者となる任意の参加者によって又はパーティｐ（ｃ）によって果たされ得る。

【０１４２】

＜鍵生成＞

変更されたＥＣＤＳＡ鍵生成アルゴリズムは、署名スキームを完全に分散させるために使用される。このスキームでは、秘密鍵は、隠匿されたランダム秘密の組み合わせを用いて分散グループによって共同して選択される。

【０１４３】

閾値鍵導出アルゴリズムは、アルゴリズム１で与えられる。

【０１４４】

アルゴリズムは拡張可能であり、アルゴリズムの各ステップは、ディーラーなしであらゆる参加者によって同時に、あるいは、グループ又は個人又はディーラーにおいて、実行され得る。この実施は、現在のビットコインプロトコルと完全互換である。如何なる署名者も、あたかも標準の方法で署名されたかのように外部の観測者又は検証者に見える。結果として、標準フォーマットで又は本発明の高度なプロトコルを用いて鍵が生成されたかどうかを伝える方法はない。

【０１４５】

＜署名生成＞

閾値署名生成の概念は、［Shamir，１９７９］に記載されている。アルゴリズム３は、ＤＨに基づくシステムに基づいており且つＥＣＤＳＡを可能にするように変更されている［Feldman，１９８７］で報告されたプロシージャに関係がある。

【０１４６】

本発明は、ビットコイントランザクションの処理及び署名の両方と完全互換であるようにこのプロセスを拡張する。これはまた、マルチシグトランザクションに及び、必要である複数の署名の夫々に対して、分散される鍵を求めることが可能である。

【０１４７】

＜秘密鍵の再分散＞

このプロセスは、完全分散型鍵再分散スキームを導入するように拡張され得る。この再分散は、現在の参加者がアルゴリズム２を１回実行して、結果として得られたゼロシェアを参加者の秘密鍵シェアに加える場合に、完了される。新しいシェアは、１人の参加者がランダムな値を導入した場合にランダムに分散される。

【０１４８】

このプロセスは、実際の秘密鍵を変更せずに、秘密鍵シェアの付加的なマスキング（additive masking）を可能にする。

【０１４９】

＜閾値ＥＣＤＳＡ署名導出＞

閾値ＥＣＤＳＡ署名生成システムは、［Shamir，１９７９］で開発されたスキームに従った［Feldman，１９８７］で見られる閾値ＤＳＳ署名生成プロトコルに関するアイディアを用いて導出される。

【０１５０】

＜検証＞

本発明は、如何なる値も既知のビットコインアドレスへ転送される前に、メッセージのオフラインの署名及び検証を可能にする。各パーティは、アルゴリズム１で言及されたプロセスを用いて独立してアドレスを計算及び妥当性確認することができる。従って、全ての参加者は、ビットコインアドレスに資金を出すことを求める如何なる実行よりも前に、彼らのシェアが有効であると認識することができる。このプロセスでは、検証スキームが可能であるが、それらは不要である。無効な署名スライスを送ることを選択する如何なる閾値参加者も、実際には、否定的な投票を行っている。すなわち、メッセージに署名せず、従って、ビットコインでトランザクションを完了しないという投票が、無活動から達成される。その影響は、彼らが全くメッセージ署名しなかったかのようである。

【０１５１】
アルゴリズム２は、参加者が彼らのシェアの一貫性を検証され得る方法を提供する。閾数の悪意のない参加者が保たれる場合には、鍵再生成時に如何なる既知の悪意のある参加者も除外することが可能である。従って、鍵スライスは、新しいスライスを既知の悪意のある参加者に割り当てずに更新可能であり、スライスの再割り当てを更に可能にする方法で鍵のリフレッシュを可能にする。
【０１５２】
信頼性が乏しく、悪意のある敵が標準として予想されるべき環境で、安全なマルチパーティ計算を完了するときに、ｊ／２の受動的な及びｊ／３の能動的な敵［Ben-Or，１９８９；Rabin，１９８８］に対して守る能力を高めながら検証プロセスのロバスト性を更に強化することが可能である。システムのロバスト性は、追加のプロセスを用いて強化され得る：
１．次数（ｋ－１）の多項式Ａ（ｘ）でｊ人の参加者の間で共有される秘密をD_aとする。
２．個別に参加者ｐ（ｉ）は、グループに利用可能にされるD_aのシェア$D_{a(i)}$及び$D_{a(i)}$ Ｇ∀（ｉ∈０，・・・，ｊ）を有する。
３．全ての参加者は、次に、アルゴリズム２を使用することによって秘密ｂを共有し、それにより、各参加者ｐ（ｉ）は、次数（ｋ－１）の多項式でD_bの新しい隠匿されたシェア$D_{b(i)}$を有する。
【数１９】

$$D_{b(i)} = \sum_{h=1}^{j} D_{b(i)}^{(h)}$$

に留意すべきであり、ここで、$D_{b(i)}^{(h)}$は、参加者ｐ（ｈ≠ｉ）から参加者ｐ（ｉ）にサブミットされたサブシェアである。
４．参加者は、アルゴリズム２を使用し、それにより、各参加者ｐ（ｉ）は、自由項がゼロに等しい次数（２ｋ－１）の多項式で新しい隠匿されたシェア$Z_{(i)}$を有する。
５．各参加者ｐ（ｉ）は、$D_{a(i)}D_{b(i)}^{(h)}$ Ｇ∀（ｈ∈０，・・・，ｊ）及び$D_{a(i)}D_{b(i)}$ Ｇをグループへ発行する。
６．各参加者ｐ（ｈ≠ｉ）は、彼らが$D_{b(i)}^{(h)}$及び$D_{a(i)}$ Ｇを有するということで、$D_{a(i)}D_{b(i)}^{(h)}$ Ｇの妥当性を検証することができる。
７．また、参加者ｐ（ｉ）は、
【数２０】

$$D_{a(i)}D_{b(i)}G = \sum_{h=1}^{j} D_{a(i)}D_{b(i)}^{(h)}$$

を更に検証することができる。
【０１５３】
如何なる参加者も、このシステムにより、他の参加者が悪意を持って行動しているかどうかを判定することができる。
【０１５４】
＜分散型鍵生成＞
このスキームによれば安全な方法で分散型自律企業（distributed autonomous corporations）（ＤＡＣ）及び分散型自律社会組織（distributed autonomous social organisations）（ＤＡＳＯ）の両方の実施を完了することが可能である。如何なるｋ人のメンバーも、識別スキームを通じて（認証局によって署名及び発行されたデジタル証明書によることを含む。）そのようなグループを表すことができること、及び如何なるｋ人のメンバーも、組織に代わってデジタル署名を構成することができることが示されている。このシステムは、如何なる識別機能もなしで検証し、価値の移動を提供するビットコイ

ントランザクションの署名に拡張される。これらの認証スキームは、安全であることが証明されている。

【０１５５】

＜方法及び実施＞

プロトコルは、国際特許出願ＷＯ２０１７／１４５０１６で開示されている技術に基づき、ＥＣＣを用いて、参加者間で送られる必要がある秘密情報を暗号化するということで、必要に応じて、危殆化されるか又は敵意を持たれる可能性がある参加者に対して妥当性確認が行われ得るように、全てのメッセージを、全ユーザへ送られる単一のパケットにまとめることが可能であり且つ望ましい。

【０１５６】

署名生成は、調整参加者ｐ（ｃ）によって提案される。デフォルトで、任意の鍵スライスが、調整参加者として機能することができ、要件は、プロトコルの個々の実施に帰着する。ｐ（ｃ）による有効な生のトランザクションの生成時に、トランザクション及びトランザクションのメッセージハッシュは、暗号化されたチャネルを用いて全参加者ｐ（ｉ≠ｃ）へブロードキャストされる。

【０１５７】

Ａ．一時鍵シェア$D_k(i)$を生成する。

【０１５８】

参加者は、アルゴリズム１を用いて、次数（ｋ－１）の多項式により、Z_n^*において一様分布した一時鍵D_kを生成して、シェア

【数２１】

$$\left(D_{k(1)},\cdots,D_{k(j)}\right) \overset{((k-1),j))}{\leftrightarrow} D_k \bmod n$$

を生成する。

【０１５９】

D_kのシェアは、各参加者によって個々に保持されながら秘密に保たれる。

【０１６０】

Ｂ．マスクシェアα_iを生成する。

【０１６１】

各参加者は、シェア

【数２２】

$$\left(\alpha_1,\cdots,\alpha_j\right) \overset{((k-1),j)}{\leftrightarrow} \alpha \bmod n$$

を生成するように、アルゴリズム１を用いて、次数（ｋ－１）の多項式により、Z_n^*において一様分布したランダムな値α_iを生成する。これらは、$D_k(i)$を乗算的にマスキングするために使用される。

【０１６２】

α_iのシェアは秘密であり、対応する参加者によって保持される。

【０１６３】

Ｃ．マスクシェアβ_i，c_iを生成する。

【０１６４】

次数２（ｋ－１）の多項式を用いてアルゴリズム２を２度実行する。

【０１６５】

これらのプロトコルで生成されるシェアを

【数２３】

$$\left(\beta_1,\cdots,\beta_j\right)^{(2(k-1),j)} \leftrightarrow \beta \bmod n \text{ および } \left(c_1,\cdots,c_j\right)^{(2(k-1),j)} \leftrightarrow c \bmod n$$

と表す。これらは、加算マスクとして使用される。マスキングされる数が次数（ｋ－１）の２つの多項式の積を含むということで、多項式は次数２（ｋ－１）でなければならない。これは、秘密を回復するために必要とされるシェアの必要数を２倍にする。

【０１６６】

βｉ及びｃｉのシェアは、参加者によって秘密に保たれるべきである。

【０１６７】

Ｄ．メッセージｍ：ｅ＝Ｈ（ｍ）のダイジェストを計算する。

【０１６８】

この値は、ｐ（ｃ）から取得されるトランザクションの受け取られたハッシュに対してチェックされる。

【０１６９】

Ｅ．ｖｉ＝Ｄｋ（ｉ）αｉ＋βｉｍｏｄｎ及びωｉ＝Ｇ×αｉをブロードキャストする。

【０１７０】

参加者Ｐｉは、ｖｉ＝Ｄｋ（ｉ）αｉ＋βｉｍｏｄｎ及びωｉ＝Ｇ×αｉをブロードキャストする。

【０１７１】

Ｐｉから応答が受け取られない場合に、使用される値はヌル（null）にセットされる。

【０１７２】

【数２４】

$$\left(\upsilon_1,\cdots,\upsilon_j\right)^{(2(k-1),j)} \leftrightarrow D_k\alpha \bmod n$$

が留意されるべきである。

【０１７３】

Ｆ．μ＝Interpolate（ｖ１，・・・，ｖｊ）ｍｏｄｎを計算する。

【０１７４】

Interpolate（）［２］：

ここで、大部分の（ｋ－１）がヌル（null）であり、且つ、全ての残余値が（ｋ－１）次多項式Ｆ（．）にあるように、｛ｖ１，・・・，ｖｎ｝（ｊ≧（２ｋ－１））が集合を形成する場合に、μ＝Ｆ（０）。

【０１７５】

多項式は、一般の多項式補間を用いて計算され得る。関数“Interpolate（）”は、バーレカンプ－ウェルチ補間［２］であり、ＢＣＨ及びリード・ソロモン符号のためのエラー訂正アルゴリズムとして定義される[2]。これは、http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.htmlで及びWhittaker, E. T. and Robinson, G. "Lagrange's Formula of Interpolation." §17 in The Calculus of Observations: A Treatise on Numerical Mathematics, 4th ed. New York: Dover, pp.28-30, 1967, 更に、https://jeremykun.com/2015/09/07/welch-berlekamp/で更に詳細に記載されている。

【０１７６】

Ｇ．θ＝Exp－Interpolate（ω１，・・・，ωｊ）を計算する。

【０１７７】

Exp－Interpolate（）［１０］：

せいぜい（ｋ－１）個の値がヌル（null）であり、残りの値がＧ×αｉの形をとるとして、｛ω１，・・・，ωｊ｝（ｊ≧（２ｋ－１））がセットされ、各αｉが何らかの（ｋ－

１）次の多項式Ｈ（・）に存在するならば、その場合に、θ＝Ｇ×Ｈ（０）。
【０１７８】
この値は、$\theta = \Sigma_{i \in V} \omega_i \times \lambda_i = \Sigma_{i \in V}（G \times H（i））\times \lambda_i \beta$によって計算され得る。この場合に、Ｖは、正しいω$_i$値の（ｋ）－サブセットであり、更には、λ$_i$は、結果として起こるラグランジュ補間係数を表す。多項式は、バーレカンプ－ウェルチ復号器を使用することによって計算され得る。
【０１７９】
Ｈ．（R_x，R_y）＝θ×μ^{-1}を計算する。
【０１８０】
Ｉ．ｒ＝R_xｍｏｄｑを割り当てる。ｒ＝０の場合に、ステップＡへ進む。
【０１８１】
各参加者ｐ（ｉ）は、ステップＪでｒ$_i$の彼らのスライスを計算する。調整者ｐ（ｃ）は、閾数の応答を受け取った場合に、ｓを再構成するためにそれらの値を使用することができる。
【０１８２】
Ｊ．$s_i = D_{k(i)}（e + D_{A(i)} r）+ c_i$ｍｏｄｎをブロードキャストする。
【０１８３】
応答がＰ$_i$から懇願／受信されない場合に、使用される値はヌル（null）にセットされる。
【０１８４】
【数２５】

$$\left(s_1,\cdots,s_j\right) \overset{(2(k-1),j)}{\leftrightarrow} D_k\left(m + D_A r\right) \bmod n \quad (s_1,\ldots,s_n) \longleftrightarrow k(m + D_A r) \bmod \mathrm{n}$$

が留意されるべきである。
【０１８５】
Ｋ．ｓ＝Interpolate（s_1，・・・，s_n）ｍｏｄｎを計算する。
【０１８６】
ｓ＝０の場合に、ステップＩへ進む。
【０１８７】
ここで、関数Interpolate（）は、先に定義された通りである。
【０１８８】
各参加者ｐ（ｉ）は、ステップＪでｒ$_i$の彼らのスライスを計算する。調整者ｐ（ｃ）は、閾数のｓ$_i$個の応答を受け取った場合に、ｓを再構成するためにそれらの値を使用することができる。
【０１８９】
Ｌ．（ｒ，ｓ）を返す。
【０１９０】
Ｍ．生のトランザクションの署名セクションを置換し、これをネットワークへブロードキャストする。
【０１９１】
＜スライスのディーラー分配＞
上記の実施形態は、グループシェアの導入を通じてはるかに柔軟にされ得る。このようにして、シェアの割り当ては、任意のレベルの階層的深さで、ディーラー、複数のディーラー、ディーラーを含まないグループ、又はそれらの任意の可能な組み合わせの間で分割され得る。
【０１９２】
値d_A及びその対応する鍵スライス$d_{A(i)}$を、同じアルゴリズムを用いて導出された値により置換することによって、投票のヒエラルキーが生成され得る。例えば、スキーム

は：
１）ディーラーに基づく分配
２）複数のディーラー
３）ディーラーなし
から導出されるシェアを同時に統合して構築され得る。
【０１９３】
従って、スキームは拡張可能であり、任意のビジネス構造又は組織システムを組み込むように作られ得る。
【０１９４】
スライスの割り当ても拡張可能である。一様でない割り当てプロセスを展開することは、シェアに対する重み付けが加えられることを可能にする。図３に示されるスキームでは、５つの上位メンバーにより仮想組織を構築することが可能である。なお、これには、ｎ＝５の均等に重み付けされたシェアの値をセットする必要はない。この仮想組織において、次のように、上位スキーマのための投票構造をセットすることが可能である：
・閾値（０）　６１シェア
・D_{L02}　１５シェア
・D_{L1}　１５シェア
・D_{L2}　１５シェア
・D_1　４５シェア
・D_2　１０シェア
ここで、ｎ＝１００がセットされている。述べられているように、これは、任意の組織構造を反映することができる任意の値である。図３の組織は、否認（veto）シナリオ（D_1）を可能にし、重層的な割り当ての導入を通じて、想像可能な任意の投票構造を可能にする。
【０１９５】
マルチレベル下位層構造でしばしば見落とされるものは、たとえ秘密のスライスが割り当てられているとしても、それらが等しく分配される必要がなく、更には、サブグループの所有権が他のレベルの所有権を反映する必要がないことである。図３には、閾値の７５％でシェアの総数の４５％を制御する一見強力なブロックが存在する。シェアの下位レベル割り当てがその場合に考えられるならば、シナリオははるかに複雑になる。テーブル上で複数のレベル及び位置で議決権付きシェア（voting shares）を保持する個人との相互所有権（cross-owenership）を生成することが可能である。
【０１９６】
表３の分布は、（保持されるシェア，閾値，割り当て｛ｎ｝）と定義される。
【０１９７】
表３から、参加者Ｐ１及びＰ２は夫々投票を支配するが、参加者Ｐ４との連携は、Ｐ１又はＰ２が投票を拒否しない限り、Ｐ１又はＰ２のどちらか一方に、十分な投票ブロック（voting block）をもたらす。
【０１９８】
本発明における投票フォーマットの実施及び構造には制限がないので、これは、安全なバックアップ及び回復方法を確かにしながら、想像され得る如何なる組織的ヒエラルキーも構築するために使用され得る。
【０１９９】

【表２】

表３　仮想組織構造

参加者	レベル０	レベル１	投票数（最大）	投票数（最小）
Ｐ１	D_{L0_2} (15,61,100)	D_{1A} (5,6,10) D_{2A} (3,8,10)	70	15
Ｐ２	D_{L1} (15,61,100)	D_{1B} (5,6,10) D_{2B} (3,8,10)	70	15
Ｐ３	D_{L2} (15,61,100)		15	0
Ｐ４		D_{2C} (6,8,10)	10	0

結果として、より高いレベルの署名シェアに割り当てられる拒否権及び投票権を有することが可能である。本例では、Ｓにおけるシェアの所有権は、D_{L02}、D_{1A}及びD_{2A}で保持され得る。

【０２００】

＜安全なマルチパーティ計算＞

ｎ人の参加者ｐ（１），・・・，ｐ（ｉ），・・・，ｐ（ｎ）による安全なマルチパーティ関数計算は、ｘ（ｉ）を含む関数Ｆ（ｘ１，・・・，ｘｉ，・・・，ｘｎ）を評価する必要性に基づく問題である。ｘ（ｉ）は、ｐ（ｉ）によって供給される秘密値であって、参加者ｐ（ｊ≠ｉ）又は外部のパーティがｘ（ｉ）を全く知り得ないように秘密に保たれる必要がある。従って、目的は、計算の正確さを保証することが可能でありながら、各参加者の値の秘密性を保つことである。

【０２０１】

このシナリオで、信頼できるサードパーティＴは、様々な参加者ｐ（ｉ：１・・・ｎ）から全ての値ｘ（ｉ：１・・・ｎ）を収集し、計算を返す。この設計は、暗黙的にＴを信頼することができる理想的な世界でのみ機能する。Ｔが悪意があるか、無法者であるか、あるいは、危殆的であるかのいずれかである何らかの可能性がある場合に、信頼できるサードパーティの使用は実行可能でなくなる。このシナリオは、参加者が投票者であり、信頼できるサードパーティが政府によって行われる既存の選挙を反映する。

【０２０２】

信頼できるサードパーティを用いて安全な方法で計算され得る如何なる値も、個々の秘密ｘ（ｉ）の安全性を保ちながら、信頼できるパーティなしでも計算され得ることが証明されている［Bar-Ilan，１９８９］。ここで提示されるプロトコルは、個人の計算に対して安全であり、危険な参加者の非闘グループが協働することができる場合でさえ安全な計算を提供する。

【０２０３】

＜単純乗算＞

ｎ人の参加者ｐ（ｉ：１・・・ｎ）の間で分配される２つの秘密値ｘ及びｙが存在する場合に、両方の入力変数ｘ及びｙの秘密性を保ち、同時に、参加者ｐ（ｉ）が秘密性を維持することによって個々の秘密ｘ（ｉ：１・・・ｎ）及びｙ（ｉ：１・・・ｎ）が保持されることを確かにしながら、積ｘｙを計算することが可能である。

【０２０４】

このスキームで、ｘ及びｙは、次数（ｋ－１）の多項式を用いて参加者の閾グループの間で夫々共有される。各参加者ｐ（ｉ）は、ｘの次数（ｋ－１）の多項式上のｘ（ｉ：１・・・ｎ）及びｙに対する次数（ｋ－１）の多項式内のｙ（ｉ：１・・・ｎ）の彼らの共有を乗算することができる。

【０２０５】

アルゴリズム２を導入することは、参加者ｐ（ｉ）に、次数（２ｋ－１）の多項式であるｚ（ｉ）のシェアを返す。この値により、各参加者ｐ（ｉ）は、値ｘ（ｉ）ｙ（ｉ）＋ｚ（ｉ）を計算する。

【０２０６】

ｘ（ｉ）ｙ（ｉ）＋ｚ（ｉ）の戻り値は、次数（２ｋ－１）の多項式に対するｘ・ｙの計算の有効なシェアを表す。閾数のシェアのために行動する任意の参加者又は調整者は、個々のシェアの如何なる知識も得ることなしに、ｘ・ｙの真の値を計算するために、各参加者によって保持されている戻り値を使用することができる。

【０２０７】

＜単純加算＞

ｎ人の参加者ｐ（ｉ：１・・・ｎ）の間で分配される２つの秘密値ｘ及びｙが存在する場合に、両方の入力変数ｘ及びｙの秘密性を保ち、同時に、参加者ｐ（ｉ）が秘密性を維持することによって個々の秘密ｘ（ｉ：１・・・ｎ）及びｙ（ｉ：１・・・ｎ）が保持されることを確かにしながら、和ｘ＋ｙを計算することが可能である。

【０２０８】

簡単な乗算のためのプロセスのように、各参加者ｐ（ｉ）は、値ｘ（ｉ）＋ｙ（ｉ）＋ｚ（ｉ）を計算する。ｚ（ｉ）の計算は不要であるが、プロセスに更なるレベルのランダムさ及び秘密性を加える。

【０２０９】

ｘ（ｉ）＋ｙ（ｉ）＋ｚ（ｉ）の戻り値は、次数（２ｋ－１）の多項式に対するｘ＋ｙの計算の有効なシェアを表す。閾数のシェアのために行動する任意の参加者又は調整者は、個々のシェアの如何なる知識も得ることなしに、ｘ＋ｙの真の値を計算するために、各参加者によって保持されている戻り値を使用することができる。

【０２１０】

参加者がそれほど敵対的でない場合に、これは、付加的なステップなしでｘ（ｉ）＋ｙ（ｉ）加算として簡単化され得る。

【０２１１】

＜逆数（inverse）又は逆数（reciprocal）＞

分配された秘密値、すなわち、ｊ人の参加者の間でｘ（ｉ：１・・・ｊ）として秘密に分配されるｘｍｏｄｎについて、値ｘ（ｉ）、ｘ又はx^{-1}を開示する如何なる情報も公開せずに、x^{-1}ｍｏｄｎの値に関連した多項式のシェアを生成することが可能である［Gennaro，１９９６］。先と同じく、各参加者ｐ（ｉ）は、次数（ｋ－１）の多項式上でｘ（ｉ）によって表される値ｘのシェアを保持する。

【０２１２】

アルゴリズム１を使用して、各参加者は、次数（ｋ－１）の多項式で未知の秘密ｘ・ｙのシェアｘ（ｉ）を生成する。次いで、各参加者は、アルゴリズム２を実行して、次数（２ｋ－１）の多項式でゼロ秘密の（ｋ－１）を計算する。各参加者（２ｋ－１）は、値ｘ（ｉ）ｙ（ｉ）＋ｚ（ｉ）を計算する計算を行う。

【０２１３】

上記のInterpolate（）ルーチンを使用して、各参加者は、μ＝ｘ（ｉ）ｙ（ｉ）＋ｚ（ｉ）の値を計算して、μ_iの収集された値から値μを返すことができる。次いで、各参加者は、μ^{-1}ｍｏｄｎの値を計算することができる。

【０２１４】

これらの値は、任意の参加者ｐ（ｉ）が次数（２ｋ－１）の多項式で$\zeta_i=\gamma_i\mu^{-1}$を用いてx_i^{-1}の関連するシェアを計算することができるほど十分である。バーレカンプ－

ウェルチ復号化スキーム［Berlekamp，１９６８］は、このプロセスを完了するために使用され得るいくつかの方法のうちの１つを提供する。

【０２１５】

図９は、数量$\zeta = x^{-1} + w$の安全なマルチパーティ計算を実行する方法の一例をフローチャート形式で示す。ここで、安全なマルチパーティ計算は、複数の第１ノードによって実行され、計算された秘密は、第２ノードによって回復される。最初に、ステップ９０１で、秘密ｘ、ｙ、z_1のシェアx_i、y_i、x_{1i}が、第２ノードによって第１ノード間で分配される。ここで、ｘ、ｙ、z_1は、各々の多項式関数の自由項であり、z_1はゼロである。

【０２１６】

ステップ９０２で、各第１ノードは、$\mu_i = x_i y_i + z_i$の各々のシェアを計算する。多項式補間によって、積$\mu = xy$が、ステップ９０３で、閾数のシェアμ_iを受け取っているノード（１以上の第１ノード又は第２ノードであってよい。）によって決定される。ステップ９０４で、逆数$\mu^{-1} \bmod n = x^{-1} \cdot y^{-1} \bmod n$が次いで決定される。ステップ９０５で、秘密ｗ、$z_2$のシェア$w_i$、$z_{2i}$が次いで第２ノードによって第１ノードの間で分配される。ここで、ｗ及びz_2は、多項式関数の各々の自由項であり、z_2はゼロである。ステップ９０６で、各第１ノードは、共通の秘密のそのシェアζ_iを決定する。ここで、$\zeta_i = y_i \cdot \mu^{-1} + w_i + z_{2i} = y_i \cdot x^{-1} \cdot y^{-1} + w_i + z_{2i}$。次いで、第２ノードは、ステップ９０７で閾数のシェアζ_iを第１ノードから受け取り、ステップ９０８で多項式補間を実行して、個々の秘密ｘ、ｙ又はｗが第１ノードのいずれにも利用可能でないように、計算された秘密$\zeta_i = x^{-1} + w \bmod n$を決定する。

【０２１７】

＜割り当て＞

検証可能且つ立証可能な方法でトランザクションに署名する能力は、非公開で所有権を証明し、更には、ブロックチェーン上で何も公に動かさずに、ビットコイン秘密鍵の所有権及び関連するビットコインアドレスを破棄又は交換する機会をもたらす。このようにして、ビットコインアドレスに資金を提供することができ、そのアドレスの内容は、公的な記録を残さずに移転又は販売され得る。このプロセスは閾値システムであるから、鍵スライスの割り当ては、ブロックチェーン上で追加の決済を記録することなしに安全に実現され得る。

【０２１８】

このようにして、ブロックチェーン上で既に精算されたノートの所有権を、そのノートを処理するプロセスから分離することが可能である。

【０２１９】

＜ＣＬＴＶ＞

ビットコインメッセージ、又はより一般的な用語では、トランザクションは、ＣＬＴＶ［BIP 65］エントリの包含により生成され得る。この追加により、トランザクションは、全ての鍵スライスの壊滅的な喪失が発生した場合、又はエンティティからの複数のスライスが、最小閾値による署名の安全な再構成を可能にしないように失われるか若しくは信用できないと見なされる場合でさえ、回復可能にされ得る。

【０２２０】

これは更に、エンティティがサードパーティサービスを使用しており、サービスが鍵へのアクセスを保持又は拒絶することができないことを確かにすることを望む場合に、可能である。時間に基づくフェイルセーフ（failsafe）でビットコイントランザクションを構成することにおいて、ユーザは、悪意のあるサードパーティ又は危殆化された交換サイト若しくは銀行が彼らの鍵へのアクセスを彼らに強要することができないことを知っている。最悪のシナリオとして、壊滅的なレベルへの危殆化は、ＣＬＴＶ条件に基づき、事前に定義されたアドレスへのトランザクションの時間に基づく逆戻りをもたらすことになる。この事前に定義されたアドレスは、本願で開示されるプロトコルを用いて生成され得る。そのようなものとして、容易に危殆化され得ない一連のトランザクション及び鍵を構成す

ることが可能である。

【０２２１】

＜安全性の懸案事項＞

Benger et. al.（２０１４）は、フラッシュ（Flash）及びリロード（reload）方法によるＥＣＤＳＡ秘密鍵回復の一例を提案している。この出来事は、システムＲＡＭ及びキャッシュに対する攻撃の一例にすぎない。これらの方法は、それらが秘密鍵を再構成するということで、シャミアのＳＳＳ［１９７９］のような手順の使用を欠いたままとする。更に、秘密鍵がいつでも再構成される任意のシナリオで、信頼性に対する要件が導入される。このシナリオでは、秘密鍵を保持するエンティティのシステム及びプロセスに依存する必要がある。

【０２２２】

信頼できるパーティがたとえ悪意がないとしても、彼らのプロセスに依存する必要がある。最近の多数の妥協案から分かるように、秘密鍵を再構成することへのこの依存は、攻撃の道を残している。

【０２２３】

既存のＥＣＤＳＡ実装のためのドロップイン置換と、現在のビットコインプロトコルとの完全な透明性及び互換性との両方を備えているために、その実装にはハードフォーク又はソフトフォークが不要であり、実装は、如何なる現在のトランザクションとも区別不可能である。本発明は、回復機能を備えた鍵のグループ署名を可能にする個別の参加者として個人を扱うことができる。一例として、ツー・オブ・ツー（two of two）のスキームが４つの鍵スライスを用いて実装可能であり、このとき、オンラインウォレットプロバイダ又は取引所が２つの鍵スライスを保持し、エンドユーザが２つのスライスを保持する。取引所及びユーザは夫々、彼らの鍵スライスに対するツー・オブ・ツーのプロセスを有し、それらは、必要に応じてメッセージの安全な署名のためにお互いに連結して使用され得る。

【０２２４】

＜閾ボールト＞

上記のプロセス及び技術は、どのノードも秘密鍵を独立して生成することができないように、公開－秘密鍵対を生成するために使用され得る。各ノードによって個々に保持される秘密鍵シェアは、やり取りにおいて決して公開されず、安全性を更に高めるために又は鍵シェア喪失に対処するために必要に応じてリフレッシュされ得る。この機能は、ブロックチェーンネットワーク内の「閾ボールト」（Threshold Vault）の実装を促す。閾ボールトは、ブロックチェーン上のアドレスであり、それと関連付けてデジタルアセットは記録され得る。それにより、秘密鍵シェアを保持している閾数のノードが、閾ボールト内でデジタルアセットにアクセスし、それを転送し、又は別なふうにそれを扱うために協調するよう求められる。閾ボールトは、全て又は一部の鍵シェアが、入力として閾ボールトを有するトランザクションにデジタル署名するために必要とされるように、確立され得る。閾値は、ボールトの目的及び目標に応じてセットされてよい。例えば、第１目的が秘密鍵の開示に対して保護することであるいくつかの場合に、閾値は、鍵シェアの１００％にセットされてよい。第１目的が投票スキームであるいくつかの場合に、鍵シェアの５１％以上が閾値としてセットされてよい。一例において、鍵シェア喪失に対する安全性及び保護の両方が目的であってもよく、その場合に、（例えば）３／５閾値が、鍵シェアの３／５を制御する一次ボールト所有者により且つ鍵シェアの残り２／５を保持する１以上のサードパーティリポジトリによりセットされてよく、それによって、一次ボールト所有者が彼らの鍵シェアの１以上（しかし、全てではない。）へのアクセスを失った状況で、サードパーティリポジトリと協調して彼らに彼らの鍵シェアを使用させてもらうことを一次ボールト所有者に可能にする。このようにして、サードパーティリポジトリは、一次ボールト所有者の鍵シェアの１つが失われた場合に、“バックアップ”鍵シェアの守護者となり得る。

【０２２５】

一例として説明するために、これより図４を参照する。図４は、閾ボールトの実装に関連するシェアの分配の例を図式的に示す。この例には、ブロック１０２を含む部分ブロックチェーン１００が表されている。ブロック１０２は、トランザクションから部分的になり、ボールトアドレスＰに関連した出力１０６を特徴とするトランザクション１０４を含む。ボールトアドレスＰは、公に利用可能である。その場合に、ボールトアドレスに関連した未使用トランザクション出力（ＵＸＴＯ）が、他のトランザクションへの入力となるよう利用可能である。

【０２２６】

ＵＸＴＯが他のトランザクションへの入力として使用されるために、ボールトアドレスに対応する秘密鍵Ｖが、デジタル署名を生成することにおいて使用される必要がある。この場合に、秘密鍵Ｖは生成されておらず且つ生成されることはない。代わりに、秘密鍵は、複数のノード１１０にわたって秘密鍵シェア（V_1，V_2，・・・V_n）の形で保持される。これらの鍵シェアは、完全ブロックチェーンノード（例えば、サーバ、パーソナルコンピュータ、など）、ウォレットノード（例えば、パーソナルコンピュータ、タブレット、ラップトップ、モバイルデバイス、又はブロックチェーンウォレットソフトウェアを実行可能な何らかの他のネットワーク化されたコンピューティングデバイス内で実装される。）、オフラインメモリ／ストレージ（例えば、ＵＳＢスティック、メモリカード、又は他の永続性デジタル記憶媒体）、オンラインリポジトリ（例えば、ブロックチェーン交換アカウント、銀行取引ネットワークデバイス、又はブロックチェーンデータを記憶する他の遠隔のサードパーティ操作サービス）、あるいは、何らかの他のそのようなコンピューティングデバイスに記憶されてよい。

【０２２７】

いくつかの例となる実施において、各ノードは、閾数の所有者の間の協調がボールトアドレスＰに関するデジタル署名を生成するために必要とされるように、別個のエンティティによって操作又は制御され得る。いくつかの例となる実施において、たとえ鍵シェアが鍵保護のために、物理的又は論理的に、別個の電子デバイスにあるとしても、閾数のノードは単一のエンティティによって制御され得る。閾数ではなく、追加の鍵シェアが、主要所有者が鍵シェアを失った場合に鍵回復サービスの機能を果たす働きをし得る他のエンティティによって操作される電子デバイスに記憶されている。更なる他の例となる実施において、２つ（又はそれ以上）のエンティティは、例えば、２人（又はそれ以上の）パートナー、すなわち、長及び連署当局（例えば、銀行、親会社、など）の場合に、デジタル署名生成で協働するのに十分な各々の鍵シェアを制御し得る。当業者であれば、パーティ間の鍵シェアの分配及び署名のための関連閾値の設定において反映され得る論理的及び法的な関係の範囲及び種類は十分に理解されるだろう。

【０２２８】

これより図５を参照する。図５は、閾ボールトを用いてデジタルアセットを受け取り且つ転送する方法２００を一例としてフローチャート形式で示す。この例では、複数のノードが、閾ボールトアドレスを生成することと、如何なる中央ディーラー又は当局もなしで対応する鍵シェアを秘密に且つ安全に生成及び分配することとにおいて協働すべきであり、それによって、生成プロセス中又はデジタル署名プロセス中に如何なる他のノードの鍵シェアへもアクセスするノードがないことを確かにする。

【０２２９】

動作２０２で、ノードは、閾ボールトアドレス（例えば、公開鍵）を協働して生成する。そのために、個々のノードは独立して、各個別ノードによって秘密に選択された各々の多項式に基づき、各々の秘密鍵を生成及び記憶する。各ノードによって選択された多項式は、ノード間で同意された閾値に基づく位数を有する。例えば、ｊ個のノードの間で、閾ボールト生成プロセスを開始することにおいて、ノードは、秘密鍵を使用する（語「使用する」は、本願では、秘密鍵が如何なるノードによっても決して実際に再構成されないことを意味する。）ために必要とされるｋ個の協調ノードの閾値をセットすることに同意している。ここで、$k \leqq j$。閾ボールトアドレスは、ノードによって共有されている特定の

多項式値に基づき且つ多項式補間を用いてノードのいずれか又は夫々によって決定され得る。

【０２３０】

動作２０４で、閾ボールトアドレスへデジタルアセットを移動させるブロックチェーン上のブロックにトランザクションが加えられる（そして、妥当性確認され且つ承認される）。

【０２３１】

将来のある時点として、動作２０６で、トランザクションが生成される（より一般的に「メッセージ」と称され得る。）。トランザクションは、そのトランザクションへの入力としてデジタルアセットを使用することを提案する。この提案されたトランザクションは、１つのノードによって又は他の無関係のノードによって生成され得る。入力として閾ボールトアドレスを特徴とするメッセージを受け取ると、ノードは、トランザクションにデジタル署名することによってそのトランザクションを承認すべきかどうかを判定する。

【０２３２】

ノードがトランザクションを承認するように指示されるとすると、動作２０８で、少なくとも閾数ｋのノードが、メッセージのためのデジタル署名を生成するように協働する。この協働は、使用される如何なる鍵シェアもマスキングするためにプロセス内で使用されるノード間のランダム化されたマスクシェアを生成するようにジョイントゼロ秘密分散の使用に依存し、それによって、如何なる他のノードの鍵シェアも受け取るノードがないことを確かにする。デジタル署名プロセスで使用される如何なるマスクシェア又は鍵シェアも、秘密ディーラーなし分配を用いて生成及び分配される。デジタル署名プロセスは、第１署名成分ｒ及び第２署名成分ｓの決定を含む。決定プロセスにおいて、如何なる一時鍵シェア又は秘密鍵シェアも、分散される前にマスクをかけられ、如何なる分散も、先に詳述され且つ以下で更に説明されるように、他の式内に埋め込まれたマスキングされた鍵シェアを含む値を有する。従って、メッセージのためのデジタル署名は、完全な秘密鍵を生成することさえなしに、且つ、ノードのいずれかによって保持されている如何なる秘密鍵シェアも公開せずに、生成される。

【０２３３】

これより図６を参照する。図６は、閾ボールトアドレスを生成する方法３００を一例としてフローチャート形式で示す。閾値ｋは前もってセットされる。ノードの夫々は、他のノードの夫々との安全な通信を確立する。これは、［Wright，２０１６］で記載されるように、情報の安全なやり取りのための共通の秘密を決定することを含み得る。なお、如何なる他の暗号鍵生成プロセスも、各ノードが暗号化された通信を各他のノードへ送り、暗号化された通信を各他のノードから受信し暗号解読することができるという条件で、用いられてよい。

【０２３４】

方法３００は、ノードp_i（ｉ＝０～ｊ）の夫々がそれ自身の各々の第１秘密$a_0^{(i)}$を有していることから始まる。第１秘密は、如何なる適切な方法でも取得又は決定されてよい。

【０２３５】

各ノードはまた、次数ｋ－１のそれ自身の各々の多項式$f_i(x)$を選択する。多項式の自由項、例えば、$f_i(0)$は、ノードの第１秘密、すなわち、係数a_0である。各々の多項式は夫々異なった係数a_iを有し得るが、夫々が次の式：

【数２６】

$$f_i = \sum_{t=0}^{(k-1)} a_t x^t \bmod n$$

で表される。

【０２３６】

ノードの夫々は、他のノードの夫々のために自身の多項式の値を決定する。これに関連して、ｘは、他のノードｈ＝１，・・・，（ｉ－１），（ｉ＋１），・・・，ｊに関連したインデックスｈにセットされ得る。すなわち、各ノードは、全てのｈ（ｈ≠ｉ）について、$f_i(h)$の値を決定する。方法３００の動作は、「第１ノード」ｉとの関連で後述される。なお、同じ又は類似した動作が、他のノードの夫々によって実行される。

【０２３７】

動作３０２によって示されるように、第１ノードｉは、ノードｈのための第２秘密$f_i(h)$をそのノードｈと安全に（すなわち、暗号化されたチャネル上で）共有する。すなわち、各他のノードｈに対して、第１ノードｉは値$f_i(h)$を送る。次いで、第１ノードｉはまた、全ての他のノードに対して、自身が決定した全ての第２秘密の楕円曲線点乗算（ＥＣＰＭ）をブロードキャストする。すなわち、第１ノードｉは、他の全てのノードｈに、全ｈについての$f_i(h)\times G$の値を送る。第１ノードｉはまた、Ｋ＝０，・・・，（ｋ－１）についての自身の多項式係数のＥＣＰＭ$a_K{}^{(i)}\times G$をブロードキャストする（すなわち、他のノードの夫々へ送る）。従って、第１ノードｉは、自身の多項式係数を公開せず、むしろ、各係数に対応する「公開鍵」を公開し、且つ、決定された第２秘密全般を公開せず、むしろ、各決定された第２秘密に対応する「公開鍵」を公開する。第１ノードｉは、１つの第２秘密を各他のノードと内密にのみ共有している。

【０２３８】

動作３０４で、第１ノードｉは、他のノードの各ノード（又はそれらのうちの少なくともｋ－１個）から、インデックスｘについて当該他のノードによって決定された「第２秘密」を受信する。すなわち、第１ノードｉは、夫々のｈ≠ｉ（又はｊ－１個のノードのうちの少なくともｋ－１個）についての値$f_h(i)$を受信する。第１ノードｉはまた、他のノードによってブロードキャストされたＥＣＰＭ値、すなわち、第２秘密のＥＣＰＭ及び各々の多項式の係数のＥＣＰＭを受信する。

【０２３９】

図６には明示的に示されていないが、ノードは独立して、ブロードキャストされた値に基づき、他のノードから受信された第２秘密を検証及び妥当性確認し得る。すなわち、各ノードは、ノードｈから取得された第２秘密が有効であることを、そのＥＣＰＭ値を計算し、それを当該他のノードからのブロードキャストされたＥＣＰＭ値と比較することによって、決定し得る。

【０２４０】

動作３０６で、第１ノードｉは、他（少なくともｋ－１個）のノードの夫々から受信された各々の第２秘密$f_h(i)$の和として第１秘密鍵シェア$d_{A(i)}$を決定し格納する。ノードｉによって決定されたそれらの各々の第２秘密の和は：

【数２７】

$$d_{A(i)}=\sum_{h=1}^{j} f_h(i)\mathrm{mod}n$$

として表現され得る。

【０２４１】

動作３０８で、ノードの１つ以上（第１ノード以外のノードに委託され得るか、又はノ１よりも多いノード若しくは全てのノードによって行われ得る。）は、閾ボールトアドレス、すなわち、ノードが夫々秘密鍵シェア$d_{A(i)}$を有している秘密鍵d_Aに対応する公開鍵Q_A、を決定する。公開鍵は、第２秘密の閾数のＥＣＰＭの和から、多項式補間を用いて取得される。上述されたように、これは」

【数２８】

$$Q_A = Exp - Interpolate\left(f_1, \cdots, f_j\right) \triangleright \left[= G \times d_A\right]$$

として表現され得る。

【０２４２】

明らかなように、方法３００は、公開鍵Q_Aによって与えられる閾ボールトアドレスに対応するノードで独立して生成された秘密鍵シェア$d_{A(i)}$をもたらす。

【０２４３】

これより図７を参照する。図７は、閾ボールトに関するメッセージにデジタル署名する方法４００を一例としてフローチャート形式で示す。方法４００は、メッセージハッシュｅ＝Ｈ（ｍ）によって表され得るメッセージｍのデジタル署名を生成するために使用される。メッセージｍは、閾ボールトが入力である提案されたブロックチェーントランザクションと関係すると見なされ得る。従って、トランザクションが妥当性確認されるために、それは、閾ボールト公開鍵Q_Aに対応する秘密鍵の所持を裏付けるデジタル署名を必要とする。当然、ノードのどれもが、秘密鍵を有さず、且つ、秘密鍵を生成する能力を有さず、それらは、秘密鍵シェア$d_{A(i)}$しか保持していない。それでもなお、方法４００は、閾数のノードがそれらの秘密鍵シェアを公開することさえせずに有効なデジタル署名を生成することにおいて協働するメカニズムを提供する。

【０２４４】

動作４０２によって示されるように、方法４００は、ディーラーなし秘密分配を用いて、各ノードｉについて一時鍵シェア$D_{k(i)}$及び乗算マスクシェアa_iを生成することを含む。ノードｉは、例えば、方法３００を用いて生成された秘密鍵シェア$d_{A(i)}$を既に保持している。動作４０４で、ジョイントゼロ秘密分散（ＪＺＳＳ）が、第１及び第２の加算マスクシェアβ_i及びc_iを生成するために（例えば、アルゴリズム２に関連して上述されたように）使用される。第１加算マスクシェア及び第２加算マスクシェアは関係しない。

【０２４５】

次いで、動作４０６で、各ノードは、乗算マスクシェア及び第１加算マスクシェアによってマスキングされた一時鍵シェアに基づき形成された二重マスク化された鍵シェアを公開する。一例において、二重マスク化された鍵シェアは、

$$\theta_i = D_{k(i)} \alpha_i + \beta_i \mathrm{mod} n$$

と表される。

【０２４６】

各ノードはまた、乗算マスクシェアのＥＣＰＭに対応する公開乗算マスクシェアを公開する。これは、

$$\omega_i = G \times \alpha_i$$

と表され得る。

【０２４７】

いずれのノードも、自身の公開鍵シェア、一時鍵シェア、又はマスクシェアを決して公開しない。

【０２４８】

動作４０８で、第１署名成分ｒが、公開されたデータから決定される。決定は、ノードの夫々によって独立して、又はノードの１つによって行われ、他のノードによって共有及び／又は検証され得る。

【０２４９】

楕円曲線デジタル署名アルゴリズム（ＥＣＤＳＡ）は、ランダムな値ｋの逆数を決定することを含む。ランダムな値は、本例では、一時鍵D_kである。すなわち、逆数は、D_k^{-1}を含む。第１署名成分ｒを決定するために、マスク化された一時鍵μが、二重マスク化された鍵シェアに対して多項式補間を用いて決定される：

$$\mu = \mathrm{Interpolate}(\theta_1, \cdots, \theta_n) \bmod n$$

加算マスクはジョイントゼロシェアに基づくので、それは補間から落ちて、結果$\mu = D_k \alpha \bmod n$を残す。同様に、乗算マスクシェアのＥＣＰＭに対する指数関数的多項式補間は、

$$\theta = \mathrm{Exp}-\mathrm{Interpolate}(\omega_1, \cdots, \omega_n)$$

$$\theta = G \times \alpha$$

を与える。次いで、第１署名成分ｒは、$r_{x,y} = (R_x, R_y) = \theta \times \mu^{-1}$を計算することによって取得され、これは、

$$r_{x,y} = (G \times \alpha) \times [D_k \alpha \bmod n]^{-1} = G \times D_k^{-1}$$

である。第１署名成分ｒは、$R_x \bmod n$である。

【０２５０】

動作４１０で、各ノード（又はそれらの少なくともｋ個）は、第２署名成分シェアs_iを決定する。第２署名成分は、一時鍵、秘密鍵、メッセージのハッシュ、及び第１署名成分を含む。どのノードも一時鍵又は秘密鍵を保持せず、それらは鍵シェアしか有さない。従って、各ノードは、第２署名成分シェアs_iを

$$s_i = D_{k(i)}(e + d_{A(i)} r) + c_i \bmod n$$

のように生成する。次いで、この値は、ブロードキャストされるか又は他のノードと共有される。第２加算マスクシェアは、第２署名成分シェアを公開することにおける安全性を改善するために使用される。

【０２５１】

次いで、動作４１２で、第２ＥＣＤＳＡ署名成分ｓが、多項式補間を通じて第２署名成分シェアから取得される：

$$s = \mathrm{Interpolate}(s_1, \cdots, s_j) \bmod n$$

ここで、先と同じく、第２加算マスクシェアc_iは、それがジョイントゼロシェアに由来するので、補間から落ちる。それでもなお、それは、公開された第２署名成分シェアにランダムマスクを提供するのに役立つ。動作４１２は、ノードの１以上によって、又は代理ノードによって実行されてよい。２つの成分（ｒ，ｓ）を生成すると、メッセージｍのデジタル署名は、秘密鍵シェアのいずれか又は一時鍵シェアのいずれかを公開せずに、このように生成される。

【０２５２】

動作４１４で、デジタル署名は、現在デジタル署名されているメッセージを含むブロックチェーントランザクションに組み込まれる。次いで、そのトランザクションは、妥当性確認及び最終的な確認のためにネットワークにサブミットされ得る。

【０２５３】

これより図８を参照する、図８は、ノード８００の簡略化された例をブロック図形式で示す。ノード８００は、プロセッサ８０２を含み、プロセッサ８０２は、１以上のマイクロプロセッサ、特定用途向け集積チップ（ＡＳＩＣ）、マイクロコントローラ、又は同様のコンピュータ処理デバイスを含んでよい。ノード８００は、メモリ８０６を更に含み、メモリ８０６は、値、変数、及びいくつかの場合には、プロセッサ実行可能プログラム命令を記憶するよう、永続性及び非永続性メモリを含んでよい。ノード８００は、有線又は無線ネットワーク上でネットワーク接続性を提供するネットワークインターフェイス８０４を更に含む。

【０２５４】

ノード８００は、プロセッサ実行可能命令を含むプロセッサ実行可能ブロックチェーンアプリケーション８０８を含む。プロセッサ実行可能命令は、実行される場合に、プロセッサ８０２に、本明細書で記載される機能又は動作の１つ以上を実行させる。

【０２５５】

本明細書で記載されるデバイス及びプロセス、並びにノードを構成するための記載された方法／プロセスを実装するあらゆるモジュール、ルーチン、プロセス、スレッド、アプリケーション、又は他のソフトウェアコンポーネントが、標準のコンピュータプログラミ

ング技術及び言語を用いて実現され得ることが理解されるだろう。本願は、特定のプロセッサ、コンピュータ言語、コンピュータプログラミング仕様、データ構造、又は他のそのような実施詳細に限定されない。

【０２５６】

＜おわりに＞

本発明は、グループ署名プロセスの導入により、ビットコインが達成しようとしたものの基礎を形成する。分散型鍵生成システムの結合による耐障害性署名システムの付加は、全ての中央集権化及び信頼要件を除く。

【０２５７】

更に、暗黙的に非中央集権型のシステムの導入は、より堅牢であり且つ回復力のあるプロトコルの生成を可能にする。ＥＣＤＳＡ［Johnson，２００１］とシャミアのＳＳＳ［Shamir，１９７９］との間の互換性により、本発明は、新しい検証可能な秘密分散スキームによりビットコインを拡張するシステムを導入することができた。このシステムは、安全性において何も失うことなく、フェルドマン［Feldman，１９８７］又はペダーセン［Pedersen，１９９２］によって導出されたものよりもはるかに有効である。

【０２５８】

本願では、基本プロトコルの変更を必要とせずにビットコインの機能性を拡張するシステムが記載されてきた。本発明によれば：

１．信頼できるサードパーティは、鍵秘密の選択又は分散のためにもはや必要とされない。

２．サードパーティの信用に依存しない分散型銀行取引システムが構築され得る。

３．夫々のメンバー又はメンバーのグループは、保持されている秘密鍵のシェアが、公示されているビットコインアドレス及び公開鍵に対応することを独立して検証し得る。

４．傍受及び関連する攻撃の影響を軽減するように秘密鍵スライスをリフレッシュするプロトコルが存在する。

５．信頼できるサードパーティは、トランザクション及びメッセージのグループ署名のために不要である。

【０２５９】

本発明は、機密事項を扱うデータが決してメモリに現れないようにするということで、多くの現存している安全性リスクを完全に解決する。

【０２６０】

上記の実施形態は、本発明を制限するのではなく説明しているのであって、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適用範囲から外れることなしに、多くの代替の実施形態を設計することが可能である点が留意されるべきである。特許請求の範囲において、かっこ内の如何なる参照符号も、特許請求の範囲を制限するものとして解釈されるべきではない。語「有する」（comprising及びcomprises）などは、いずれかの請求項又は明細書の全文に挙げられているもの以外の要素又はステップの存在を除外しない。本明細書中、「有する」（comprises）は、「～を含むか、又はそれらから成る」（includes or consists of）を意味し、「有する」（comprising）は、「～を含むか、又はそれらから成る」（including or consisting of）を意味する。要素の単一参照は、そのような要素の複数参照を除外せず、逆もまた同じである。本発明は、いくつかの個別要素を有するハードウェアを用いて、且つ、適切にプログラムされたコンピュータを用いて、実施されてよい。いくつかの手段を列挙している装置クレームでは、それらの手段のうちのいくつかが、ハードウェアの同一アイテムによって具現されてもよい。特定の手段が相互に異なる請求項で挙げら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ことを示すものではない。

【０２６１】

＜参考文献＞

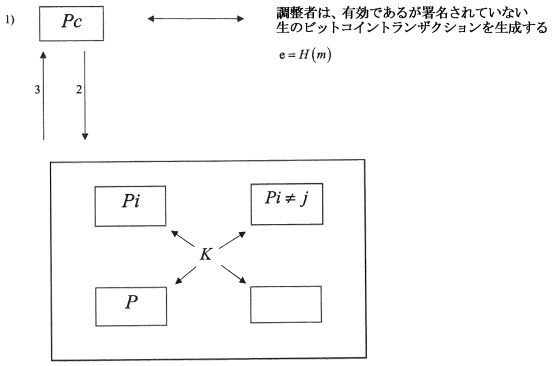
１）Bar-Ilan, J. Beaver, "Non-Cryptographic Fault-Tolerant Computing in a Constant Number of Rounds", Proc. of 8th PODC, pp.201-209, 1989.

2) Berlekamp, Elwyn R. (1968), Algebraic Coding Theory, McGraw‐Hill, New York, NY.
3) Benger, N., van de Pol, J., Smart, N.P., Yarom, Y.: "Ooh Aah... Just a Little Bit": A Small Amount of Side Channel Can Go a Long Way. In: Batina, L., Robshaw, M. (eds.) Cryptographic Hardware and Embedded Systems ¦ CHES 2014, LNCS, vol.8731, pp.75-92. Springer (2014)
4) Ben-Or, M., Goldwasser, S., Wigderson, A.: “Completeness theorems for noncryptographic fault-tolerant distributed computation”. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. pp.1‐10. STOC ’88, ACM, New York, NY, USA (1988)
5) BIP 65 OP_CHECKLOCKTIMEVERIFY https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki
6) Chaum, David (1983). "Blind signatures for untraceable payments" (PDF). Advances in Cryptology Proceedings of Crypto. 82 (3): 199‐203.
7) Dawson, E.; Donovan, D. (1994), "The breadth of Shamir's secret-sharing scheme", Computers & Security, 13: Pp.69‐78
8) Desmedt. You (1987). “Society and Group Oriented Cryptography: A New Concept”. In A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology (CRYPTO '87), Carl Pomerance (Ed.). Springer-Verlag, London, UK, UK, 120-127.
9) Feldman. P. “A practical scheme for non-interactive verifiable secret sharing”. In Proceedings of the 28th IEEE Annual Symposium on Foundations of Computer Science, pages 427‐437, 1987.
10) Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: “Robust threshold DSS signatures”. In: Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques. pp. 354‐371. EUROCRYPT’96, SpringerVerlag, Berlin, Heidelberg (1996)
11) Ibrahim, M., Ali, I., Ibrahim, I., El-sawi, A.: “A robust threshold elliptic curve digital signature providing a new verifiable secret sharing scheme”. In: Circuits and Systems, 2003 IEEE 46th Midwest Symposium on. vol.1, pp.276‐280 (2003)
12) Johnson, D., Menezes, A., Vanstone, S.: “The elliptic curve digital signature algorithm (ecdsa)”. International Journal of Information Security 1(1), 36‐63 (2001)
13) Kapoor, Vivek, Vivek Sonny Abraham, and Ramesh Singh. "Elliptic Curve Cryptography." Ubiquity 2008, no. May (2008): 1-8.
14) Knuth, D. E. (1997), “The Art of Computer Programming, II: Seminumerical Algorithms” (3rd ed.), Addison-Wesley, p.505.
15) Koblitz, N. "An Elliptic Curve Implementation of the Finite Field Digital Signature Algorithm" in Advances in Cryptology - Crypto '98. Lecture Notes in Computer Science, vol.1462, pp.327-337, 1998, Springer-Verlag.
16) Liu, C. L. (1968), “Introduction to Combinatorial Mathematics”, New York: McGraw-Hill.
17) National Institute of Standards and Technology: FIPS PUB 186-4: “Digital Signature Standard” (DSS) (2003)
18) Pedersen, T.: “Non-interactive and information-theoretic secure verifiable secret sharing”. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO ’91, LNCS, vol. 576, pp.129‐140. Springer (1992)
19) Rabin T. & Ben-Or. M. (1989) "Verifiable secret sharing and mul

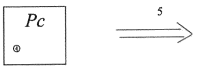
tiparty protocols with honest majority". In Proc. 21st ACM Symposium on Theory of Computing, pages 73--85, 1989.
２０）Shamir, Adi (1979), "How to share a secret", Communications of the ACM, 22 (11): Pp.612‐613
２１）Wright, C. & Savanah, S. (2016) “Determining a common secret for two Blockchain nodes for the secure exchange of information” International Patent Application Number Application Number: WO 2017/145016.

【図面】

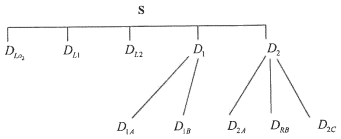
【図１】



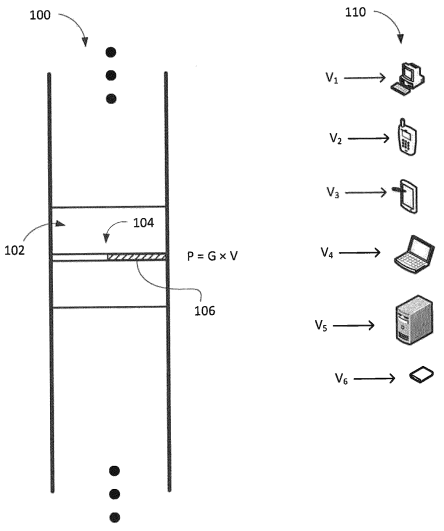
【図２】



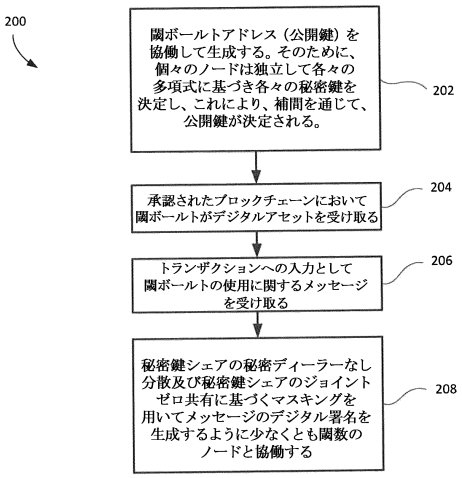
【図３】



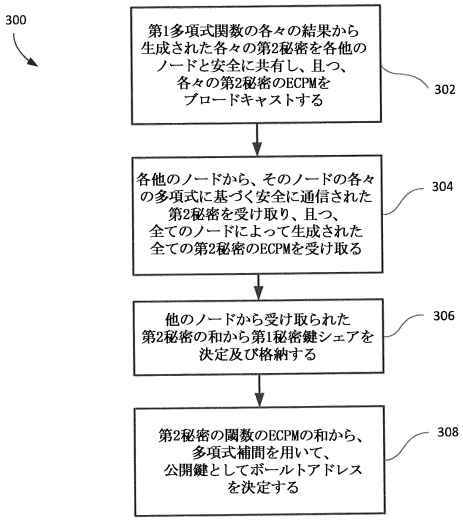
【図４】



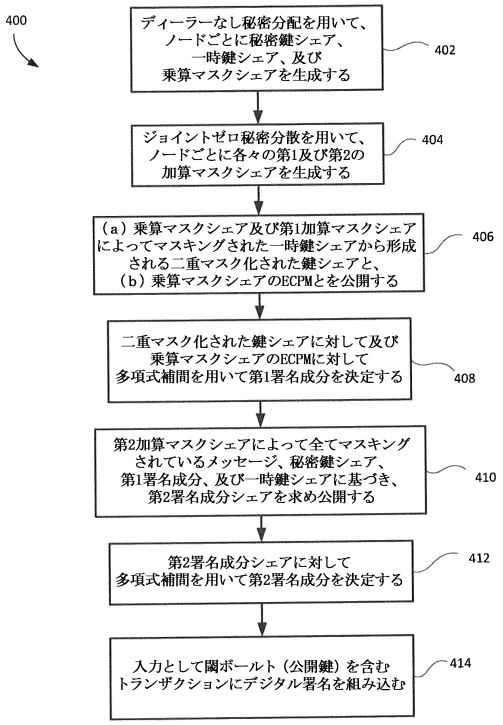
【図５】


200
閾ボールトアドレス（公開鍵）を協働して生成する。そのために、個々のノードは独立して各々の多項式に基づき各々の秘密鍵を決定し、これにより、補間を通じて、公開鍵が決定される。
202
承認されたブロックチェーンにおいて閾ボールトがデジタルアセットを受け取る
204
トランザクションへの入力として閾ボールトの使用に関するメッセージを受け取る
206
秘密鍵シェアの秘密ディーラーなし分散及び秘密鍵シェアのジョイントゼロ共有に基づくマスキングを用いてメッセージのデジタル署名を生成するように少なくとも閾数のノードと協働する
208

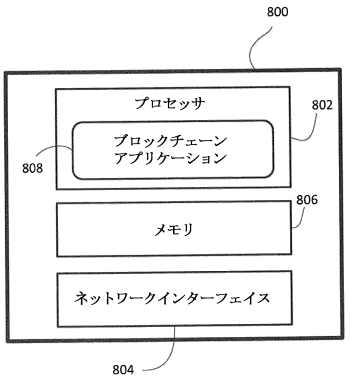
【図６】


300
第1多項式関数の各々の結果から生成された各々の第2秘密を各他のノードと安全に共有し、且つ、各々の第2秘密のECPMをブロードキャストする
302
各他のノードから、そのノードの各々の多項式に基づく安全に通信された第2秘密を受け取り、且つ、全てのノードによって生成された全ての第2秘密のECPMを受け取る
304
他のノードから受け取られた第2秘密の和から第1秘密鍵シェアを決定及び格納する
306
第2秘密の閾数のECPMの和から、多項式補間を用いて、公開鍵としてボールトアドレスを決定する
308

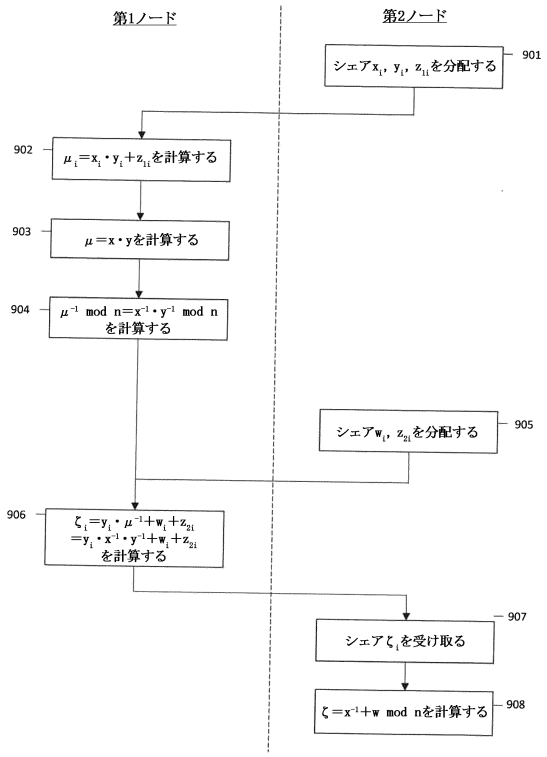
【図７】


400
ディーラーなし秘密分配を用いて、ノードごとに秘密鍵シェア、一時鍵シェア、及び乗算マスクシェアを生成する
402
ジョイントゼロ秘密分散を用いて、ノードごとに各々の第1及び第2の加算マスクシェアを生成する
404
（a）乗算マスクシェア及び第1加算マスクシェアによってマスキングされた一時鍵シェアから形成される二重マスク化された鍵シェアと、（b）乗算マスクシェアのECPMとを公開する
406
二重マスク化された鍵シェアに対して及び乗算マスクシェアのECPMに対して多項式補間を用いて第1署名成分を決定する
408
第2加算マスクシェアによって全てマスキングされているメッセージ、秘密鍵シェア、第1署名成分、及び一時鍵シェアに基づき、第2署名成分シェアを求め公開する
410
第2署名成分シェアに対して多項式補間を用いて第2署名成分を決定する
412
入力として閾ボールト（公開鍵）を含むトランザクションにデジタル署名を組み込む
414

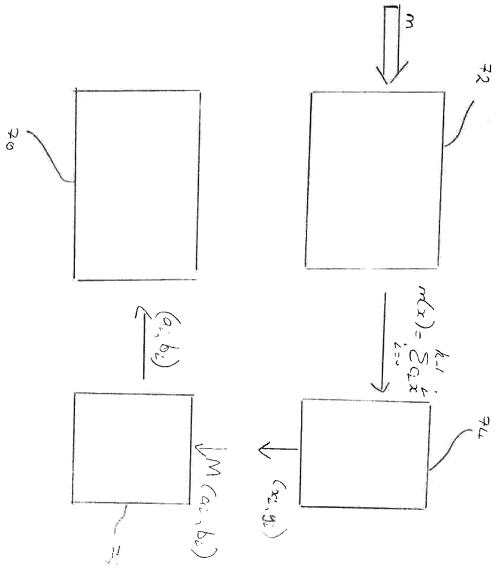
【図８】


800
プロセッサ
802
ブロックチェーンアプリケーション
808
メモリ
806
ネットワークインターフェイス
804

【図９】


第1ノード
第2ノード
シェアx_i, y_i, z_{1i}を分配する
901
902
$\mu_i = x_i \cdot y_i + z_{1i}$を計算する
903
$\mu = x \cdot y$を計算する
904
$\mu^{-1} \bmod n = x^{-1} \cdot y^{-1} \bmod n$
を計算する
シェアw_i, z_{2i}を分配する
905
906
$\zeta_i = y_i \cdot \mu^{-1} + w_i + z_{2i}$
$= y_i \cdot x^{-1} \cdot y^{-1} + w_i + z_{2i}$
を計算する
907
シェアζ_iを受け取る
908
$\zeta = x^{-1} + w \bmod n$を計算する

【図１０】


3
72
70
74

フロントページの続き

国際事務局(IB)
(31)優先権主張番号　1714660.6
(32)優先日　平成29年9月12日(2017.9.12)
(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/IB2017/055497
(32)優先日　平成29年9月12日(2017.9.12)
(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　PCT/IB2017/057782
(32)優先日　平成29年12月11日(2017.12.11)
(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　PCT/IB2018/055604
(32)優先日　平成30年7月26日(2018.7.26)
(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　Steven Goldfeder et al.，Securing Bitcoin wallets via threshold signatures，SEMANTIC SCHOLAR，Corpus ID: 16244349，［オンライン］，2014年，〈URL: http://diyhpl.us/ ~ bryan/papers2/bitcoin/Securing%20Bitcoin%20wallets%20via%20threshold%20signatures.pdf〉，（検索日 令和４年３月２３日）、インターネット
Pratyush Dikshit et al.，Efficient Weighter Threshold ECDSA for Securing Bitcoin Wallet，2017 ISEA ASIA SECURITY AND PRIVACY (ISEASP)，IEEE，2017年01月29日，PAGE(S):1 - 9，http://dx.doi.org/10.1109/ISEASP.2017.7976994
Steven Goldfeder et al.，Securing Bitcoin wallets via threshold signatures，[ONLINE]，2014年06月03日，https://www.semanticscholar.org/paper/Securing-Bitcoin-wallets-via-threshold-signatures-Goldfeder-Felten/3ff6215ea5060d1fffb6f46be0e23ff94c1e9ef4
Maged H. et al.，A ROBUST THRESHOLD ELLIPTIC CURVE DIGITAL SIGNATURE PROVIDING A NEW VERIFIABLE SECRET SHARING SCHEME，MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS[ONLINE]，INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS，2003年12月27日，VOL:1，PAGE(S):276 - 280，http://dx.doi.org/10.1109/MWSCAS.2003.1562272
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／０８
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩＩ）
ＩＥＥＥ Ｘｐｌｏｒｅ
ＴＨＥ ＡＣＭ ＤＩＧＩＴＡＬ ＬＩＢＲＡＲＹ