(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7513609号
(P7513609)

(45)発行日　令和6年7月9日(2024.7.9)

(24)登録日　令和6年7月1日(2024.7.1)

(51)国際特許分類
H04L 9/32 (2006.01)

FI
H04L 9/32 200Z

請求項の数 15 （全48頁）

(21)出願番号　特願2021-527870(P2021-527870)
(86)(22)出願日　令和1年11月14日(2019.11.14)
(65)公表番号　特表2022-507797(P2022-507797A)
(43)公表日　令和4年1月18日(2022.1.18)
(86)国際出願番号　PCT/IB2019/059795
(87)国際公開番号　WO2020/109909
(87)国際公開日　令和2年6月4日(2020.6.4)
審査請求日　令和4年10月19日(2022.10.19)
(31)優先権主張番号　1819290.6
(32)優先日　平成30年11月27日(2018.11.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　1819286.4
(32)優先日　平成30年11月27日(2018.11.27)
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】　ブロックチェーンネットワークを介するデータの効率的且つセキュアな処理、アクセス、及び送信のためのシステム及び方法

(57)【特許請求の範囲】

【請求項１】

ブロックチェーンの中のトランザクション（TX）を識別する、コンピュータにより実施される方法であって、前記ブロックチェーンはブロックチェーンプロトコルを有し、前記トランザクションは前記ブロックチェーンプロトコルの必要な部分であるトランザクションID（TX_{ID}）を含み、前記方法は、

前記トランザクションを識別するためのユニークなインデックスにニーモニックをマッピングするステップ、を含み、

ｉ）前記ニーモニックは、平文の識別子、用語、又はラベルであり、

ｉｉ）前記インデックスは、前記トランザクションに関連付けられたアウトプットの中で提供され、以下：

前記トランザクション（TX）に関連付けられた公開鍵（PK）と、

前記トランザクション（TX）のトランザクションID（TX_{ID}）と、

の組み合わせを含み、

前記インデックスの前記公開鍵（PK）と前記トランザクションID（TX_{ID}）は、前記ブロックチェーンプロトコルにより要求されない、方法。

【請求項２】

マッピングする前記ステップは、

前記公開鍵（PK）及び前記トランザクションID（TX_{ID}）を、アウトプットを生成する演算へのオペランドとして使用するステップと、

前記ニーモニックを前記アウトプットにマッピングするステップと、
を含む、請求項１に記載の方法。
【請求項３】
前記ニーモニックをマッピングする前に、前記アウトプットをハッシングするステップ、及び／又は、
前記公開鍵（PK）と前記トランザクションID（TX_{ID}）を連結するステップ、
を更に含む請求項２に記載の方法。
【請求項４】
前記演算は連結演算である、請求項２又は３に記載の方法。
【請求項５】
ｉ）前記公開鍵（PK）は平文プレフィックスを含む、及び／又は、
ｉｉ）前記アウトプットは、前記トランザクションが前記ブロックチェーンプロトコル以外のプロトコルに従い形成されることを示すフラグを更に含む、及び／又は、
ｉｉｉ）前記インデックス及び／又はフラグは、前記アウトプットの中で、後のトランザクションへのインプットとしての後の使用について無効としてアウトプットをマークするオペコード又はOP_RETURNオペコードの後に提供される、
請求項１～４のいずれかに記載の方法。
【請求項６】
前記トランザクション（TX）について前記ブロックチェーンを検索するステップ、
を更に含む請求項１～５のいずれかに記載の方法。
【請求項７】
前記トランザクション（TX）と同じ公開鍵を有する１つ以上の更なるトランザクションを識別するステップ、
を更に含む請求項１～６のいずれかに記載の方法。
【請求項８】
前記トランザクション（TX）の前記公開鍵は、前記トランザクション（TX）のインプットの中で提供され、
前記１つ以上の更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのインプットの中で提供される、
請求項７に記載の方法。
【請求項９】
前記トランザクション（TX）の前記公開鍵は、前記トランザクションのインプットの中で提供され、
前記１つ以上の更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのアウトプットの中で提供される、
請求項７に記載の方法。
【請求項１０】
前記トランザクション（TX）の前記公開鍵は、前記トランザクションのアウトプットの中で提供され、
前記１つ以上の更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのインプットの中で提供される、
請求項７に記載の方法。
【請求項１１】
前記トランザクション（TX）の前記公開鍵は、前記トランザクションのアウトプットの中で提供され、
前記１つ以上の更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのアウトプットの中で提供される、
請求項７に記載の方法。
【請求項１２】
前記トランザクション（TX）及び前記１つ以上の更なるトランザクションを調べて、前

記ブロックチェーンのプロトコルに従い、どちらが最新のトランザクションかを決定するステップ、を含む請求項７～１１のいずれかに記載の方法。
【請求項１３】
前記最新のトランザクションは、
ブロック高、又は、
ｐｒｏｏｆ－ｏｆ－ｗｏｒｋ、又は、
ブロックの中のトランザクション順序、
により決定される、請求項１２に記載の方法。
【請求項１４】
コンピュータにより実装されるシステムであって、
プロセッサと、
前記プロセッサによる実行の結果として、前記システムに請求項１～１３のいずれかに記載のコンピュータにより実施される方法を実行させる実行可能命令を含むメモリと、
を含むシステム。
【請求項１５】
実行可能命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記実行可能命令は、コンピュータシステムのプロセッサにより実行された結果として、少なくとも、前記コンピュータシステムに、請求項１～１３のいずれかに記載の方法を実行させる、非一時的コンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、概して、電子ネットワーク、特にブロックチェーンネットワークのようなピアツーピアネットワークに渡るデータ通信及び交換のための改良に関する。本発明は、データ記憶、アクセス、検索、及び処理に関し、特に、ブロックチェーン上のこのようなデータ関連動作に関する。本発明は、特に、限定ではないが、ウェブサイト及びウェブページにより提供されるものと同様の方法で、しかし基礎となるメカニズム又はプラットフォームとしてウェブサーバではなくブロックチェーンを用いて、データを処理する際に使用することに適する。従って、本発明は、セキュアな効率的な暗号法により実施される、データ処理及び転送のための代替的インフラストラクチャを提供する。
【０００２】
本願明細書では、私たちは、全ての形式の電子的な、コンピュータに基づく、分散型台帳を包含するために用語「ブロックチェーン」を使用する。これらは、総意に基づくブロックチェーン及びトランザクションチェーン技術、許可及び未許可台帳、共有台帳、並びにこれらの変形を含む。他のブロックチェーン実装が提案され開発されているが、ブロックチェーン技術の最も広く知られているアプリケーションは、Ｂｉｔｃｏｉｎ台帳である。Ｂｉｔｃｏｉｎは、ここでは、便宜上及び説明の目的で参照されることがあるが、本発明はＢｉｔｃｏｉｎブロックチェーンと共に使用することに限定されず、代替のブロックチェーン実装及びプロトコルが本発明の範囲に包含されることに留意すべきである。用語「ユーザ」は、ここでは、人間またはプロセッサに基づくリソースを表してよい。用語「Ｂｉｔｃｏｉｎ」は、ここでは、Ｂｉｔｃｏｉｎプロトコルから派生するプロトコルの全ての変形及びバージョンを包含するために使用される。
【０００３】
ブロックチェーンは、コンピュータに基づく非集中型の分散型システムとして実装されるピアツーピアの電子台帳であり、ブロックにより構成され、ブロックはまたトランザクションにより構成される。各トランザクションは、ブロックチェーンシステムの中の参加者間でデジタルアセットの制御の移転を符号化するデータ構造であり、少なくとも１つのインプット及び少なくとも１つのアウトプットを含む。各ブロックは前のブロックのハッシュを含み、これらのブロックは一緒に繋げられて、起源以来ブロックチェーンに書き込まれている全てのトランザクションの永久的な変更不可能な記録を生成する。トランザク

ションは、スクリプトとして知られている小さなプログラムを含む。スクリプトは、それらのインプット及びアウトプットを埋め込まれ、トランザクションのアウトプットがどのように及び誰によりアクセス可能であるかを指定する。Ｂｉｔｃｏｉｎプラットフォームでは、これらのスクリプトはスタックに基づくスクリプト言語を用いて記述される。

【０００４】

トランザクションがブロックチェーンに書き込まれるためには、検証されなければならない。ネットワークノード（マイナー）は、無効なトランザクションがネットワークから拒否され、各トランザクションが有効であることを保証するために作業を実行する。ノードにインストールされたソフトウェアクライアントは、未使用トランザクション（unspent transaction, UTXO）のロック及びアンロックスクリプトを実行することにより、UTXOに対してこの検証作業を実行する。ロック及びアンロックスクリプトの実行が真（TRUE）と評価する場合、トランザクションは有効であり、トランザクションはブロックチェーンに書き込まれる。従って、トランザクションがブロックチェーンに書き込まれるためには、（ｉ）トランザクションを受信した第１ノードにより検証され、トランザクションが有効な場合には、ノードが該トランザクションをネットワーク内の他のノードに中継する、（ｉｉ）マイナーにより構築された新しいブロックに追加される、（ｉｉｉ）マイニングされる、つまり過去のトランザクションの公開台帳に追加される、ことが必要である。

【０００５】

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

【０００６】

１つのこのような関心分野は、ユーザ間のデータの記憶、共有、アクセス及び制御のためにブロックチェーンを使用することである。今日、これは、インターネットにより達成され、標準的に検索エンジンにより所望のデータにアクセスするためにユーザが訪問するウェブサイトおよびページをホスティングするサーバを伴う。

【０００７】

しかしながら、一部の観察者は、中央パーティによる膨大なデータ及びコンテンツの制御のような、インターネットの欠点の幾つかを解決するために、ブロックチェーンの使用を考え始めている。例えば、以下を参照のこと：Life After Google: The Fall of Big Data and the Rise of the Blockchain Economy”, George Gilder, Gateway Editions, July ２０１８, ISBN－１０: ９７８１６２１５７５７６４及びISBN－１３:９７８－１６２１５７５７６４.。

【０００８】

従って、ブロックチェーンの分散型、不変性、且つ永久的特性を有利に利用して、このようなデータがブロックチェーン上で格納され、処理され、読み出され、検索され、及び／又は共有されることを可能にする構成を提供することが望ましい。このような改良されたソリューションがここで考案される。本開示の実施形態は、少なくとも、ブロックチェーンソリューションを実装し、及びブロックチェーンに又はブロックチェーンからデータを格納し、処理し、検索し、及び／又は読み出す、代替的な効率的且つセキュアな技術を提供する。実施形態は、更に、少なくとも、コンピューティングノード間でデータを格納し、処理し、読み出し、転送し、検索し、及び／又は共有するための、代替的なブロックチェーンにより実装される技術的インフラストラクチャを提供する。実施形態は、更に、ブロックチェーン及びブロックチェーンプロトコルを含む技術的に異なる改良されたコンピューティングプラットフォームを介してデジタルリソースへのアクセスのセキュアな制御のためのソリューションを提供する。更に、ブロックチェーン上の複数の関連付けられ

た又は論理的にリンクされたトランザクションが効率的に、セキュアに、及び迅速に処理できる。論理的に関連付けられたトランザクションは、連続したブロック高でブロックチェーンに格納されなくてよいが、それらは、容易に且つセキュアに識別され及び／又はアクセスできる。

【発明の概要】

【０００９】

本発明は、添付の請求の範囲に定められる。

【００１０】

本発明に従い、コンピュータ／ブロックチェーンにより実施される方法及び対応するシステムが提供され得る。当該方法は、ブロックチェーンを介してデータの処理、格納、読み出し、識別、及び／又は共有を可能にする又は制御する方法として記載され得る。追加又は代替として、当該方法は、該データの識別、読み出し及び／又は共有を可能にするために、（別個の／異なる）ブロックチェーントランザクション内に格納されたデータを関連付ける又はリンクする方法として記載され得る。ここで、「共有」は、ノード又はユーザに、データ部分へのアクセスを送信し、通信し、伝送し、又は提供することを提供することを含んでよい。

【００１１】

追加又は代替として、当該方法は、ブロックチェーン内のデータ及び／又はトランザクション（ＴＸ）の識別を可能にする方法として記載されてよい。ブロックチェーンは、ピアツーピアネットワーク上で動作するピアツーピアブロックチェーンであってよい。トランザクションは、ユーザにより共有され、送信され、格納され、及び／又はアクセスされる必要のあるコンテンツ／データ、又はこのようなコンテンツ／データへの参照を格納してよい。

【００１２】

当該方法は、簡略記憶を提供するステップを含んでよい。

【００１３】

当該方法は、

前記トランザクション（ＴＸ）に関連付けられた公開鍵（ＰＫ）と、

前記トランザクション（ＴＸ）のトランザクションID（TX_{ID}）と、

に簡略記憶をマッピングするステップを含んでよい。

【００１４】

前記簡略記憶は、人間が読める識別子、用語、又はラベルであってよい。これは、ブロックチェーン上のコンテンツの検索が、簡易に、素早く、及びより少ない入力誤りにより、実行できるという利点を提供し、従って、従来に比べて増強され改良された検索／記憶／共有ソリューション及びアーキテクチャを提供する。

【００１５】

当業者は、各ブロックチェーントランザクションがそれ自体のユニークな識別子（ＩＤ）を有し、その結果、該ブロックチェーントランザクションがブロックチェーン上で識別できることを直ちに理解するだろう。有利なことに、本発明は、この既存のIDを、トランザクションに関連付けられた及び／又はその中で提供される公開鍵（ＰＫ）と結合し、その結果、該結合が、トランザクションのユニークな識別子を形成する。これは、次に、少なくとも上述の利点を提供するために、簡略記憶にマッピングされ得る。

【００１６】

望ましくは、マッピングする前記ステップは、

前記公開鍵（ＰＫ）及び前記トランザクションID（TX_{ID}）を、アウトプットを生成する演算へのオペランドとして使用するステップと、

前記簡略記憶を前記アウトプットにマッピングするステップと、

を含む。

【００１７】

従って、前記公開鍵及び前記トランザクションIDは、前記簡略記憶にマッピングされる

前に、何らかの方法で処理されてよい。これは、前記簡略記憶をマッピングする前に、前記アウトプットをハッシングするステップを含んでよい。前記演算は連結演算であってよい。

【００１８】

望ましくは、前記公開鍵（PK）は人間が読めるプレフィックスを含む。これは、従来知られているような「バニティ（vanity）アドレス」であってよい。これは、公開鍵が、人間によりより読みやすく又は認識しやすいテキスト部分を含むという利点を提供し、検索、処理、アクセス、及び読み出しを一層容易に、速く、且つ誤りを少なくする。

【００１９】

当該方法は、前記トランザクション（TX）について前記ブロックチェーンを検索するステップを更に含んでよい。追加又は代替として、当該方法は、前記トランザクション、及び／又は前記トランザクションに関連付けられた、その中の、又はそれから参照されるデータ部分を識別するステップを含んでよい。これは、前記公開鍵（PK）及びトランザクションIDの結合を用いて実行されてよい。

【００２０】

当該方法は、コンピューティングリソースから別のコンピューティングリソースへ、前記トランザクションに関連付けられたデータを送信するステップを含んでよい。これらのリソースのうちの一方又は両方は、ブロックチェーンネットワーク上のノードであってよい。

【００２１】

当該方法は、プロトコルフラグと任意公開鍵（discretionary public key (DPK)）と任意トランザクションID（discretionary transaction ID (DTxID)）とを含む少なくとも１つのブロックチェーンノードトランザクション（Node）を処理するステップを含んでよい。

【００２２】

特徴のこの組合せは、ノードトランザクションに格納されたデータ部分をブロックチェーン上で識別できるようにし、更に、複数のトランザクションの中で提供されるとき、互いにリンクされ／関連付けられることを可能にする。それは、データ部分の間の階層的関係を反映し、それらの記憶、処理及び共有を実現する、グラフ又はツリーのような構造を構成することを可能にする。これは、本発明が考案される前には可能ではなかった改良された方法で、ピアツーピアネットワーク（ブロックチェーン）上の複数のノード及びトランザクションに渡るデータの記憶、識別、通信、及び読み出しを可能にする。本発明は、ブロックチェーンネットワークを新しい方法で使用することを可能にし、並びに、改良された技術的結果を提供するために、本発明は、改良されたブロックチェーンにより実装されるネットワークを提供する。

【００２３】

望ましくは、ブロックチェーントランザクション（Tx）は、データ部分、又はデータ部分への参照を含む。データ部分への参照は、該データが格納された場所のポインタ、アドレス、又は他の指示子であってよい。データ部分は、任意の種類のデータ又はデジタルコンテンツ、例えばコンピュータ実行可能アイテム、テキスト、ビデオ、画像、音声ファイル、等であってよい。前記データ部分は「コンテンツ」と呼ばれてよい。前記データ部分又はそれへの参照は、処理済み形式であってよい。例えば、それは、前記データ部分のハッシュダイジェストであってよい。データは、ブロックチェーン上又はブロックチェーン外（つまり、「オフチェーン」）に格納されてよい。

【００２４】

任意公開鍵（DPK）及び／又は任意トランザクションID（DTxID）は、基礎にあるブロックチェーンのプロトコルにより記述されるようにトランザクションの必須成分ではなく、それらが本発明の部分として提供される点で、「任意（discretionary）」であってよい。別の方法では、それらは、基礎にあるブロックチェーンのプロトコルに従いトランザクションを検証するために必要とされない。それらは、ブロックチェーンがそれらを必要

とするからではなく、本発明の部分として提供される追加の必須ではないアイテムである。
【００２５】
望ましくは、プロトコルフラグは、１つ以上のブロックチェーントランザクションの中のデータを検索し、格納し、及び／又は読み出す、ブロックチェーンに基づくプロトコルに関連付けられ及び／又はそれを示す。プロトコルフラグは、指示子又はマーカであってよい。プロトコルフラグは、トランザクションが所定のプロトコルに従い形成されていることを示してよい。これは、基礎にあるブロックチェーンのプロトコル以外のプロトコルであってよい。それは、ここに記載される任意の実施形態による検索プロトコル（つまり、ここに記載される「メタネット（metanet）」プロトコル）であってよい。
【００２６】
用語「処理（processing）」は、トランザクションに関する、生成、送信、検証、アクセス、検索、及び／又は識別を含む、任意の活動を意味するとして解釈されるべきである。
【００２７】
任意トランザクションIDは、本発明の実施形態に従いトランザクション（Tx）に関連付けられた識別子、ラベル、指示子、又はタグであってよい。私たちは、これらの用語の全部を含む用語「指示子」を使用する。留意すべきことに、従来知られているように、及び当業者により直ちに理解されるように、ブロックチェーン上の各トランザクションは、従来では標準的にTXIDと呼ばれる識別子によりユニークに識別される。TXIDは、基礎にあるブロックチェーンプロトコルの基本的な、必要な、且つ任意的ではない部分である。この任意的ではないTXIDは、ここで参照される任意トランザクションID(DTxID)と混同してはならない。
【００２８】
当該方法は、前記トランザクション（TX）と同じ公開鍵を有する１つ以上の更なるトランザクションを識別するステップ、を更に含んでよい。有利なことに、２つ以上のトランザクションが同じ公開鍵を有する場合、それらは、同じコンテンツ／データの異なるバージョンとして解釈できる。従って、本発明は、ブロックチェーン上のコンテンツのバージョン付けのための簡易であるか強力なメカニズムを提供する。これは、データ完全性を保証するだけでなく、データの正しいバージョンが使用されるので、誤りが回避され、データの正しいバージョンについてのより少ない検索しか実行される必要がないので、少ない処理しか必要ないことも保証する。
【００２９】
１つ以上の実施形態では、前記トランザクション（TX）の前記公開鍵は、及び前記更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのインプットの中で提供される。これは、グラフの中の兄弟（sibling）により、兄弟の識別を可能にする。
【００３０】
１つ以上の実施形態では、前記トランザクション（TX）の前記公開鍵は、前記トランザクションのインプットの中で提供され、前記１つ以上の更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのインプットの中で提供される。
【００３１】
この場合、前記更なるトランザクションの前記公開鍵は、基礎にあるブロックチェーンプロトコルにより記述される又は要求されるトランザクションの中で提供されないという意味で、任意であるが、本発明のプロトコルの部分として提供される。これは、グラフの中の論理的子を有する論理的親の識別及び／又は関連付けを可能にする。
【００３２】
１つ以上の実施形態では、前記トランザクション（TX）の前記公開鍵は、前記トランザクションのアウトプットの中で提供され、前記１つ以上の更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのインプットの中で提供される。これも、論理的子を有する論理的親の識別及び／又は関連付けを可能にする。
【００３３】

１つ以上の実施形態では、前記トランザクション（ＴＸ）の前記公開鍵は、前記トランザクションのアウトプットの中で提供され、前記１つ以上の更なるトランザクションの前記公開鍵は、前記１つ以上の更なるトランザクションのアウトプットの中で提供される。これは、同じコンテンツ／データの異なるバージョンを含む又は有するトランザクションの識別を可能にする。

【００３４】

当該方法は、前記トランザクション（ＴＸ）及び前記１つ以上の更なるトランザクションを調べて、どちらが最新のトランザクションかを決定するステップ、を更に含んでよい。この決定は、基礎にあるブロックチェーンのプロトコルの特長を用いて、又はそれに従い／参照して、行われ又は実行されてよい。最新のトランザクションは、ブロック高、又はｐｒｏｏｆ－ｏｆ－ｗｏｒｋ、又はブロック内のトランザクション順序、を含む任意の適切な指標により決定されてよい。

【００３５】

本発明は、ここの記載する方法の任意の実施形態のステップを実行するよう配置され及び構成される対応するシステムも提供する。当該システムは、プロセッサと、プロセッサによる実行の結果として、システムに本願明細書に記載のコンピュータにより実施される方法のいずれかの実施形態を実行させる実行可能命令を含むメモリと、を含むコンピュータにより実装されるシステムであってよい。

【００３６】

本発明は、実行可能命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記実行可能命令は、コンピュータシステムのプロセッサにより実行された結果として、少なくとも、前記コンピュータシステムに、本願明細書に記載の方法の実施形態を実行させる、非一時的コンピュータ可読記憶媒体も提供する。

【００３７】

従って、本発明は、ブロックチェーンによりデータを検索し、識別し、及び／又はアクセスするための改良されたソリューションであり得る。当該方法は、電子ネットワーク、特にブロックチェーンネットワークに渡るデータ通信及び交換のための改良を提供する。

【００３８】

本発明の方法／システムの幾つかの実施形態は、特に「命名及びアドレス指定」と題された章で後述する１つ以上の特徴を含んでよい。

【００３９】

本発明の上述の及び他の態様は、本願明細書に記載の実施形態から明らかであり、及びそれを参照して教示される。本発明の実施形態は、単なる例を用いて及び添付の図面を参照して以下に説明される。

【図面の簡単な説明】

【００４０】

【図１】データが複数のアウトプットに格納される、本発明を具現化するブロックチェーントランザクションを示す。

【図２】データがインプットに格納される、本発明を具現化するブロックチェーントランザクションを示す。

【図３】データが複数のブロックチェーントランザクションのアウトプットに渡り格納される、本発明を具現化するブロックチェーントランザクションのシリーズを示す。

【図４】アトミックスワップを用いてデータへのアクセスを許可するために、暗号通貨支払いを送金する、本発明を具現化するブロックチェーントランザクションを示す。

【図５】図４のトランザクションの支払いを償還する、本発明を具現化するブロックチェーントランザクションを示す。

【図６】アトミックスワップを用いてデータへのアクセスを許可するために、トークンを発行する、本発明を具現化するブロックチェーントランザクションの中で参加者により保持されるシークレット値を示す。

【図７】アトミックスワップを用いてデータへのアクセスを許可するために、トークンを

発行する、本発明を具現化するブロックチェーントランザクションを示す。
【図８】アトミックスワップを用いてデータへのアクセスを許可するために、トークンを発行する、本発明を具現化するブロックチェーントランザクションを示す。
【図９】図７及び８のトランザクションを用いて発行されたトークンを償還する、本発明を具現化するブロックチェーントランザクションを示す。
【図１０】図７及び８のトランザクションを用いて発行されたトークンを償還する、本発明を具現化するブロックチェーントランザクションを示す。
【図１１】図９及び１０のトランザクションにより交換されるシークレットにアクセスするためのブロックチェーントランザクションを示す。
【図１２】図９及び１０のトランザクションにより交換されるシークレットにアクセスするためのブロックチェーントランザクションを示す。
【図１３】本発明の実施形態によるメタネットグラフ構造の説明を提供する。
【図１４】本発明の実施形態による、MURL検索パスを含むドメイン「bobslog」のためのメタネットグラフツリーの説明を示す。
【図１５】本発明の一例によるブラウザ－ウォレットの説明のための実施形態の概略、及びそのコア機能が本願の異なるコンポーネントに渡りどのように分割できるかを示す。
【図１６】コンテンツの検索が、本発明の実施形態のインフラストラクチャ内でどのように実行できるかを示す図を提供する。
【図１７】本発明の実施形態による、ローカルフルコピーピアとグローバルフルコピーピアとの間の説明のための相互作用を示す。
【図１８】後述する説明のための使用例を参照する際に使用するためのメタネットツリー（又はグラフ）を示す。
【図１９】後述する説明のための使用例により具現化される処理を示すフローチャートを示す。
【図２０】種々の実施形態が実装できるコンピューティング環境を示す概略図である。
【発明を実施するための形態】
【００４１】
用語「Bitcoin」は、ここでは単に便宜上使用され、限定ではないが、Bitcoinプロトコルから導出される全ての変形、及び他のブロックチェーンのための任意の代替プロトコルを含むあらゆる暗号通貨／ブロックチェーンプロトコルを含むことを意図する。本願明細書の残りの部分では、本発明の実施形態の動作を決定するプロトコルは「メタネット（Metanet）プロトコル」と呼ばれる。
【００４２】
用語「コンテンツ」、「デジタルコンテンツ」及び「データ」は、本発明の実施形態によりブロックチェーントランザクションに格納される、それにより参照される、又はそれを介してアクセスされるデータを表すために、ここでは同義的に使用されてよい。データは、トランザクションコード自体の部分又は基礎にあるブロックチェーンプロトコルにより要求されるデータとは異なり、ブロックチェーンにより伝達され、通信され、又は格納される追加の／任意のデータである。
【００４３】
＜概要＞
上述のように、コンピューティングノードの間で及びそれらにより、データを格納し、書き込み、アクセスし、及び閲覧するための改良された及び／又は代替的なインフラストラクチャの必要性が認識されている。ブロックチェーン技術に固有の利点（例えば、変更不可能な記録、暗号法の実施された制御及びアクセス、埋め込み型支払いメカニズム、台帳を公に検査する能力、分散型アーキテクチャ、等）を使用することは有利である。しかしながら、「ブロックチェーンにより実装されるインターネット」の構成は、多くの技術的観点から困難である。
【００４４】
これらの困難は、限定ではないが、ネットワーク内のデータの特定の部分、認可された

パーティのみがアクセスを得ることができるように、該データへのアクセスをどのように保護及び制御するか、ピアツーピア方式でパーティからパーティへとデータをどのように転送するか、ネットワーク内の異なる位置に格納されたまま論理的に関連付けることができるように、どのようにデータを構成し、集合的及び増強された結果を提供するために異なる位置からのデータをどのように後に結合するか、階層的方式でデータをどのように提供し及び／又は格納するか、異なるコンピューティングプラットフォームを有するユーザ及びパーティに所望のデータへのアクセスをどのように許可するか、大規模ストレージサーバ及び集中型データ制御部に頼らず又はそれらを必要としないで、（グローバルである可能性のある）コンピューティングネットワークに渡りどのようにデータを格納し、提供し、及び共有するか、を含んでよい。

【００４５】

本発明は、幾つかの点でインターネットと類似する方法で、しかし、従来知られているものと完全に異なるハードウェア及びソフトウェアコンポーネントのプラットフォームを用いて、その結果を完全に異なる方法で達成する、このような改良されたソリューションを提供する。本発明の実施形態によると、インターネット／ウェブデータを格納し、それをエンドユーザに提供するサーバは、ブロックチェーンネットワークに存在するブロックチェーントランザクションにより置き換えられる。これを達成するために、幾つかの革新が考案されなければならなかった。これらは、以下の章で説明される。

【００４６】

＜ブロックチェーン「Metanet」にデータを挿入する＞

図１を参照すると、本発明を具現化するブロックチェーントランザクションが示される。ここで、ブロックチェーン上に格納されるべき第１データは、トランザクションの１つ以上の第１アウトプットに格納され、第１データの属性を表す第２データは、トランザクションの１つ以上の第２アウトプットに格納される。第１データの１つ以上の第１部分〈Content　１〉は、トランザクションの使用可能（spendable）アウトプットに格納される。データ〈Attribute　１〉及び〈Attribute　２〉は、データがMetanetプロトコルに従い格納されていることを示すフラグと一緒にそれぞれ第１データの属性を表し、トランザクションの第２の使用不可能（unspendable）アウトプットに格納される。用語「使用不可能（unspendable）」は、トランザクションの少なくとも１つの第１及び／又は第２アウトプットが、後のトランザクションへのインプットとしての後の使用のためにアウトプットを無効であるとマークするスクリプトオペコード（OP　RETURN）を含み得ることを示すために使用される。

【００４７】

コンテンツ及びデータの属性を、トランザクションの別個のアウトプットに格納することは有利である。

【００４８】

図２は、ブロックチェーン上に格納されるべき第１データ〈Content　１〉がトランザクションのインプットに格納される、本発明を具現化するブロックチェーントランザクションを示す。Metanetフラグ、及び属性データ〈Attribute　１〉及び〈Attribute　２〉は、図１に示した構成と同様の方法で、トランザクションの使用不可能アウトプットに格納される。

【００４９】

＜データ挿入＞

＜データ挿入方法＞

以下のデータをブロックチェーンに挿入できることが望ましい。

ａ）Metanetフラグ

ｂ）属性

ｃ）コンテンツ

【００５０】

コンテンツは、ブロックチェーン上に格納されるべきデータであり、Metanetフラグは

、Metanetプロトコルに関連する任意のデータのための識別子として機能する４バイトのプレフィックスであり、一方で、属性は、コンテンツに関するインデックス、許可、及び符号化情報を含む。

【００５１】

これは、限定ではないが、データタイプ、暗号化及び／又は圧縮方式を含み得る。このような属性は、メタデータと呼ばれることがある。本願明細書において、この用語は、トランザクションメタデータとの混同を避けるために、使用を避ける。

【００５２】

以下の技術が、このデータをBitcoinスクリプト内に埋め込むために使用できる。

【００５３】

１．OP_RETURN：この方法では、全部のデータ（属性及びコンテンツ）は、明らかな使用不可能トランザクションアウトプットのロックスクリプト内のOP_RETURNの後に置かれる。

この演算子を用いたアウトプットスクリプトの例は以下の通りである。

UTXOO：OP_RETURN 〈Metanet Flag〉 〈attributes〉 〈content〉

【００５４】

２．OP_DROPを伴うOP_RETURN：この場合、OP_RETURNは属性を含み、一方、コンテンツは、使用可能トランザクションスクリプト（ロック又はアンロックのいずれか）内のOP_DROPの前に格納される。

コンテンツは、トランザクションインプット及びアウトプットの中の複数のデータパケットに分割できる。しかしながら、Bitcoinプロトコルにおいて署名され得るのはアウトプットスクリプトだけなので、トランザクションアウトプットにデータを挿入することが有利である。データがトランザクションインプットに挿入された場合、マイナー検証の代わりに、データの有効性を保証するためにデータに対するチェックサムとして、OP_MODが使用できる。例えば、３２ビットOP_MOD演算を実行し、それが予め計算された値に等しいことをチェックし得る。

【００５５】

この場合、属性は、コンテンツデータパケットがどのように再結合されるかに関する情報を含んでよい。更に、再結合されたデータパケットのハッシュH(content１ +content２)を属性として提供することは、推奨される再結合方式が使用されていることを検証することを可能にする。

【００５６】

第２データ挿入方法を実施するトランザクションが、図１に示される。簡単のため、このトランザクションは、その単一のインプットにより署名された、そのアウトプットに挿入されたコンテンツのみを含む。追加インプットに挿入されたコンテンツも、図２に示す本方法を用いて、OP_DROP文を用いて可能であり得る。

【００５７】

コンテンツが非常に大きい場合、該コンテンツを複数のトランザクションに分割することが有利であり得る。このような構成は、図３に示される。図３は、本発明を具現化するブロックチェーントランザクションのペアを示す。ここで、ブロックチェーンに格納されるべき第１データ〈Content〉は、２つのチャンク〈Content chunk １〉及び〈Content chunk ２〉に分割され、これらは、〈Content〉=〈Content chunk １〉┆┆〈Content chunk ２〉のように後に再結合できる。ここで、演算子「┆┆」は、コンテンツデータの２個のチャンクを連結する。この連結演算子は、任意の所望のビット毎、又は同様のピース毎のバイナリ演算により置き換えられてよい。２つのチャンク〈Content chunk １〉及び〈Content chunk ２〉は、次に、別個のブロックチェーントランザクションのそれぞれ適切なアウトプットに格納される。一方で、コンテンツデータの属性に関連するデータは、それぞれブロックチェーントランザクションの使用不可能アウトプットに格納される。ここでも、属性は、再結合方式に関する情報を含み得る。例えば、コンテンツは、未処理データ、実行可能プログラム、又はHTMLウェブページであってよい。更に、co

ntent１は、content２のブロックチェーン上の位置へのポインタを含んでよい。該ポインタは、ウェブページ内の埋め込まれたHTMLリンクと同じ方法で機能する。

【００５８】

留意すべきことに、両方のトランザクションは同じ公開鍵P（及びECDSA署名）をインプットとして取り入れる。その結果、〈Content chunk １〉及び〈Content chunk ２〉は、それぞれTxID１及びTxID２を有する異なるトランザクションに格納されるにも拘わらず、同じ公開鍵Pにより関連付けられ得る。

【００５９】

＜マイナー検証の役割を開発する＞

ここで、マイナーにより実行されるトランザクション検証処理は、このデータを格納するときに利益を得るために使用される。これは、トランザクションアウトプットの中の全部のデータが、少なくとも１つのトランザクションインプットの中の公開鍵Pの所有者により署名されるからであり（SIGHASH┆ALLフラグが存在する場合）、この署名は、全部のマイナーが実行するトランザクション検証処理の中でチェックされる。

【００６０】

これは以下を保証する：

・データ完全性：データが壊れている場合、CHECKSIG演算は失敗する。

・データ信頼性：Pの所有者は、データを証明しそれに署名していることが証明される。

【００６１】

これは、特に、図３に示した構成を参照して上述したように、Pのインプット署名がデータの分割されたコンポーネントの間の証明可能なリンクを提供するとき、複数のトランザクションに分割されたコンテンツについて有利である。

【００６２】

＜Rabin署名＞

データ信頼性を証明するための別の方法は、Rabin署名を使用することである。これは、メッセージ全体ではなく、データ自体に署名するために使用できる。これは、署名者がデータの現れる個々の全てのトランザクションに署名する必要がなく、署名が複数のトランザクションで再利用できるので、有利であり得る。

【００６３】

Rabin署名は、スクリプト内で容易に検証できる。これらは、OP_DROPコマンドの前にRabin署名検証を挿入することにより、上述のケース（２）に組み込むことができる。つまり、

〈content１〉 〈Rabin Sig (content１)〉 FUNC_CHECKRABSIGOP_DROP 〈H(P$_1$)〉[CheckSig P$_1$]

留意すべきことに、これは、OP_RETURNを含むスクリプトがいずれにしても失敗し、従って検証に達することができないので、上述のケース（１）では行うことができない。

【００６４】

＜Rabin署名の使用の特定の例＞

＜序論＞

デジタル署名は、Bitcoinプロトコルの基本的部分である。それらは、ブロックチェーン上に記録された任意のBitcoinトランザクションが、送信されているBitcoinの正当な保持者により認証されていることを保証する。標準的なBitcoin P２PKHトランザクションでは、トランザクションメッセージは、楕円曲線デジタル署名アルゴリズム（elliptic curve digital signature algorithm (ECDSA)）を用いて署名される。しかしながら、ECDSA署名は、通常、トランザクション全体に対して適用される。

【００６５】

ネットワークの外部からの参加者が、ネットワーク参加者により使用可能になる任意のデータタイプに対する署名を提供することを望得る、Bitcoinブロックチェーンの幾つかの使用例がある。Rabinデジタル署名を使用することにより、データの任意のピースが、それがBitcoinブロックチェーンの外部から生じて１又は複数のトランザクション内に配

置されたものである場合でも、署名され得る。

【００６６】

以下に、Rabin暗号システムの代数構造を利用することにより、どのようにBitcoinスクリプト内でデータが直接署名され検証できるかを示す。

【００６７】

＜Rabinデジタル署名＞

＜Rabinデジタル署名アルゴリズム＞

＜背景にある数学的処理＞

定義：Integers mod p

pを法とする整数（integers）は、以下の集合として定義される。

【数１】

$$\mathbb{Z}_p \coloneqq \{1,2,\dots,p-1\}$$

【００６８】

フェルマーの小定理

pを素数とする。次に、任意の整数aに以下を適用する。

【数２】

$$a^{p-1} \equiv 1 \bmod p$$

【００６９】

オイラーの基準

pを素数とする。rは、以下の場合及び以下の場合にのみ、pを法とする平方余剰である。

【数３】

$$r^{\frac{p-1}{2}} \equiv 1 \bmod p$$

【００７０】

モジュラ平方根（p=３ mod ４）

pを素数とし、次式のようにする。

【数４】

$$p \equiv 3 \bmod 4$$

次に、オイラーの基準を満たす任意の整数rについて、aが整数ならば、次式の通りである。

【数５】

$$a^2 \equiv r \bmod p$$

次に、次式の形式のaの解が存在する。

【数６】

$$a \equiv \pm r^{\frac{p+1}{4}} \bmod p$$

【００７１】

中国の余剰定理

互いに素な正整数$n_1,n_2,\dots,n_k$と任意の整数$a_1,a_2,\dots,a_k$がペアで与えられると、以下の連立合同式システムは、$N=n_1n_2\dots n_k$を法とするユニークな解を有する。

【数７】

$$x \equiv a_1 \bmod n_1$$
$$x \equiv a_2 \bmod n_2$$
$$\vdots$$
$$x \equiv a_k \bmod n_k$$

中国の余剰定理の特別な場合として、

【数８】

$$x \equiv r \bmod n_1 \cdot n_2$$

の場合に且つその場合にのみ、次式の通りであることが示される：

【数９】

$$x \equiv r \bmod n_1 \text{ and } x \equiv r \bmod n_2$$

【００７２】

＜Rabinデジタル署名アルゴリズム＞

Rabinデジタル署名アルゴリズムは、以下の通り説明できる。

【００７３】

任意のメッセージmについて、Hを、k個のアウトプットビットを有する衝突耐性のあるハッシュアルゴリズムであるとする。

【００７４】

鍵を生成するために、次式

【数１０】

$$p \equiv 3 \bmod 4, \quad q \equiv 3 \bmod 4$$

になるように、それぞれ長さが約k/２である素数p及びqを選択し、積n=p・qを計算する。秘密鍵は（p,q）であり、公開鍵はn=p・qである。

【００７５】

メッセージmに署名するために、署名者は、次式を満たすように、パディングUを選択する。

【数１１】

$H(m||U)$ は以下を満たす

$$H(m||U)^{\frac{p-1}{2}} \equiv 1 \bmod p$$
$$H(m||U)^{\frac{q-1}{2}} \equiv 1 \bmod q$$

署名Sは、次式を用いて計算される。

【数１２】

$$S \equiv \left[\left(p^{q-2} \cdot H(m||U)^{\frac{q+1}{4}} \bmod q\right) \cdot p + \left(q^{p-2} \cdot H(m||U)^{\frac{p+1}{4}} \bmod p\right) \cdot q\right] \bmod n$$

メッセージmの署名は、ペア（S,U）である。検証は、所与のm、U、及びSについて、次式をチェックすることにより簡単に行うことができる。

【数１３】

$$H(m||U) \equiv S^2 \bmod n \qquad \text{(式 1)}$$

これは、範囲０,...,ｎ－１の中に次式のような整数が存在する場合且つその場合にのみ、

真である。
【数１４】

$$H(m||U)+\lambda\cdot n=S^2 \qquad (式\ 2)$$

因子λは、組合せ（S,λ,U）を提供するために署名の中に安全に含まれてよい。
【００７６】
Rabin署名方式の有利な特徴は以下の通りである。
ａ）署名生成は計算的に高価であるが、署名の検証が計算的に容易である。
ｂ）署名のセキュリティが、素因数分解の困難さにのみ依存する。その結果、Rabin署名は、（RSAと異なり）本質的に偽造不可能である。
ｃ）以下のハッシュ関数値が、公開鍵nと同様の大きさでなければならない。
【数１５】

$$H(m||U)$$

スクリプト内の検証は、所与の署名を二乗し、モジュラ簡約（modular reduction）を実行し、次に結果が次式に等しいことを確認するだけでよいので、簡単である。
【数１６】

$$H(m||U)$$

【００７７】
＜Rabin署名証明＞
p、１を素数とし、n=p・qである。中国の余剰定理により、次式：
【数１７】

$$S^2\equiv H(m||U)\bmod n$$

が示される。
【数１８】

$$S^2\equiv H(m||U)\bmod p$$

$$S^2\equiv H(m||U)\bmod q$$

の場合且つその場合にのみ、
次式が示される。
【数１９】

$$S^2\equiv H(m||U)\bmod q$$

次式を用い
【数２０】

$$S \equiv \left(\left(p^{q-2} H(m||\, U)^{\frac{q+1}{4}} \bmod q \right) \cdot p \; + \left(q^{p-2} H(m||\, U)^{\frac{p+1}{4}} \bmod p \right) \cdot q \right) \bmod q$$

$$\equiv \left(\left(p^{q-2} H(m||U)^{\frac{q+1}{4}} \bmod q \right) p \right) \bmod q$$

$$\equiv p^{q-2} H(m||\, U)^{\frac{q+1}{4}} . p \bmod \mathrm{q}$$

$$\equiv (p^{q-1} \bmod q\,) H(m||\, U)^{\frac{q+1}{4}} \bmod \mathrm{q}$$

$$\equiv H(m||\, U)^{\frac{q+1}{4}} \bmod \mathrm{q}$$

従って

【数２１】

$$S^2 \equiv H(m||U)^{\frac{q+1}{2}} \equiv H(m||U)^{\frac{q-1}{2}} \cdot H(m||U) \equiv H(m||U) \bmod q$$

ここで、次式がオイラーの基準を満たすと仮定された。

【数２２】

$$H(m||\, U)$$

同様の計算により、次式が示される。

【数２３】

$$S^2 \equiv H(m||U) \bmod p$$

【００７８】

＜BitcoinにおけるRabin署名＞

＜スクリプト内の署名検証＞

Rabin署名を検証するために、少数の算術的及びスタック操作オペコードしか必要ない。以下の形式のRedeemスクリプトを考える。

OP_DUP OP_HASH１６０ 〈H_{160}〉 OP_EQUALVERIFY OP_MUL OP_SWAP OP_２ OP_ROLL OP_CAT FUNC_HASH３０７２ OP_ADD OP_SWAP OP_DUP OP_MUL OP_EQUAL

【００７９】

ここで、nは署名者の公開鍵である。これは、以下のインプットが提供された場合且つその場合にのみ、真（TRUE）と評価される。

〈S〉〈U〉〈m〉〈λ〉〈n〉

【００８０】

ここで、mは任意のメッセージであり、（S,λ,U）は有効なRabin署名である。代替として、Rabin署名が上述の式１を用いて確認された場合、Redeemスクリプトは以下のように与えられる。

OP_DUP OP_HASH１６０ 〈H_{160}〉 OP_DUP OP_TOALTSTACK OP_SWAP 〈roll index〉 OP_ROLL OP_CAT FUNC_HASH３０７２ OP_SWAP OP_MOD OP_SWAP OP_DUP OP_MUL OP_FROMALTSTACK OP_MOD OP_EQUAL

【００８１】

この場合、スクリプトは、以下のインプットが提供された場合且つその場合にのみ、真（TRUE）と評価される。

〈S〉〈U〉〈m〉〈n〉

【００８２】

両方のRedeemスクリプトで、３０７２ビットのハッシュ射影関数「FUNC_HASH３０７２」が使用されている。所与のメッセージ／パディング連結では、FUNC_HASH３０７

２ハッシュ射影は、以下のスクリプトを用いて生成される。
OP_SHA２５６ {OP_２ OP_SPLIT OP_SWAP OP_SHA２５６ OP_SWAP} (x１１)
OP_SHA２５６ OP_SWAP OP_SHA２５６ {OP_CAT}(x１１)
【００８３】
＜データの圧縮＞
インターネットのデータは、JavaScript及び共通のファイルタイプ、例えばテキストファイル（SML、HTML、等）、ビデオファイル（MPEG、M－JPEG、等）、画像ファイル（GIF、JPEG、等）、及びオーディオファイル（AU、WAV、等）で、例えばhttps://www.doc.ic.ac.uk/~nd/surprise_９７/journal/vol１/mmp/#text詳細に記載されるように、構成される。上述のデータ挿入技術を用いて、これらの異なるデータタイプも、ブロックチェーンに埋め込むことができる。
【００８４】
大きなファイルサイズは、ブロックチェーンに埋め込む前に、幾つかの既存の符号化方式のうちの１つを用いて圧縮できる。ランレングスハフマン符号化のような無損失データ圧縮アルゴリズムは、ZIPファイル、実行可能プログラム、テキスト文書、及びソースコードを含む幾つかのアプリケーションで使用できる。
【００８５】
特定の入力データに依存して、多くの異なるアルゴリズムが存在する。Apple無損失及び適応型変換音響符号化（Apple Lossless and Adaptive Transform Acoustic Coding）は、オーディオファイル、グラフィックファイルの圧縮のためのPNG及びTIFF、を圧縮するために使用できる。一方で、動画ファイルは、多くの無損失ビデオコーデックのうちの１つを用いて圧縮できる。データコンテンツの任意の圧縮は、属性の中のフラグを用いて示すことができる。例えば、属性の中のLZW無損失符号化方式のためのフラグは、〈LZW〉であり得る。
【００８６】
＜暗号化及び有料復号（paid decryption）＞
＜データの暗号化＞
コンテンツの所有者は、コンテンツをブロックチェーンに埋め込む前に、コンテンツを保護することを選択してよい。これは、コンテンツが、必要な許可を取得しなくては閲覧できないことを保証する。
【００８７】
データ（平文又は他のデータタイプ）の暗号化のために多くの確立された技術が存在する。これらは、非対称暗号化、又は対称暗号化として分類できる。
【００８８】
楕円曲線暗号法（Elliptic Curve Cryptography (ECC)）は、公開－秘密鍵ペアに依存するので、非対称である。それは、最も安全な暗号システムのうちの１つであり、標準的にBitcoinのような暗号通貨において使用される。ECC暗号法では、データを暗号化するために、Koblitzアルゴリズムが使用できる。
【００８９】
対称方式では、データの暗号化及び解読の両方のために、単一の鍵が使用される。高度暗号化標準（Advanced Encryption Standard (AES)）アルゴリズムは、シークレットのようなものをシードとする最も安全な対称アルゴリズムのうちの１つであると考えられ、例えば以下に詳細に記載されている：C. Paar and J. Pelzl, Chapter 4 in “Understanding Cryptography,” Springer－Verlag Berlin Heidelberg 2nd Ed., ２０１０, pp. ８７－１１８。
【００９０】
ブロックチェーンに格納されたデータを暗号化するとき、基礎にあるブロックチェーンと同じ暗号システムを使用することに利点がある。Bitcoinでは、これは、非対称暗号法ではECC鍵ペアのためのsecp２５６k１規則、及び対称暗号法ではSHA－２５６ハッシュ関数である。これらの利点は次の通りである。

・暗号化のセキュリティレベルが、データが格納される基礎にあるシステムと同じである。
・暗号化データを格納するために必要なソフトウェアアーキテクチャは、より小さなコードベースを有する。
・ウォレットの中の鍵の管理が、トランザクション及び暗号化／解読の両方のために使用できる。
・暗号化と暗号通貨における支払いとの両方に同じ鍵を使用できるので、より効率的であり、より少ない鍵しか必要ない。これは、記憶空間も削減する。
・データを解読する能力を交換／購入するために、より少ない通信チャネルしか必要ない。
・暗号化及びトランザクションのために使用される鍵が同じデータ構造であるので、セキュリティが向上し、特定タイプの鍵を狙った攻撃が軽減される。
・鍵は、基礎にある暗号通貨を用いて購入できる。

【００９１】

説明の目的で、ECCを用いてデータを暗号化するために、どのようにKoblitzアルゴリズムが使用できるかを説明する。

【００９２】

＜Koblitzアルゴリズム＞

ECC鍵ペア$P_1=S_1 \cdot G$が与えられると、Koblitzアルゴリズムは、誰もが公開鍵P_1を用いてメッセージを暗号化できるようにする。従って、対応する秘密鍵S_1を知っている者だけがメッセージを解読できる。

【００９３】

Koblitz法を用いてメッセージ「hello world」を暗号化することが望ましいと仮定する。これは、文字毎に行われる。最初の文字「h」は、以下のように暗号化され解読される。

【００９４】

１．文字「h」は、secp２５６k１曲線上の点にマッピングされる。これは、平文文字を８ビット数値にマッピングするためにASCII規則を用いて達成される。

曲線上の点は、次に、基点（base point）Gをこの数値により乗算することにより計算される。本例では、「h」は、ASCIIの１０４にマッピングされ、楕円曲線点は$P_m=104 \cdot G$により与えられる。

【００９５】

２．点P_mは、次に公開鍵P_1を用いて暗号化される。これは、ランダムな一時鍵（ephemeral key）k_0を選択し、点のペア$C_m=\{k_0 \cdot G, Q\}$を計算することにより達成される。ここで、$Q:=P_m+k_0 \cdot P_1$である。これは次にブロードキャストされてよい。

【００９６】

３．秘密鍵S_1の所有者は、$P_m=Q-S_1 \cdot k_0 \cdot G$を計算することにより、元の点を解読できる。それらは、次に、試行錯誤により又はルックアップテーブルを用いて、元のASCII数値を復元し、どの数値xが$P_m=x \cdot G$に対応するかを確立してよい。

【００９７】

＜許可を購入するためのブロックチェーンの使用＞

ブロックチェーンにデータを格納することは、支払いメカニズムがシステムに組み込まれるという明らかな利点を有する。支払いは、以下を購入するために使用できる。
・閲覧／使用するための解読データ。
・特定のアドレスにデータを挿入するための許可。

【００９８】

両方の場合に、買い手は、彼らに何かを行う許可を与えるシークレットを購入するために、暗号通貨、例えばBitcoinを使用している。このシークレットは、ハッシュプレイメージ、又は秘密鍵であってよい。

【００９９】

このような購入を行うために効率的且つセキュアな方法は、アトミックスワップを使用することである。これは、セキュアな通信チャネルを最小限に抑え、売り手が支払われ、

シークレットが買い手に開示され、又は何もイベントが生じないことを保証する。

【０１００】

暗号通貨での支払いに加えて、アクセストークンを用いて許可を購入することも便利であり得る。これは、買い手が購入を行うために使用できる、買い手の所有するシークレット値（標準的にハッシュプレイメージ）である。このようなトークンは、前もって、買い手によりバルクで購入され、彼らが実際に許可を使用したいときに活性化されてよい。

【０１０１】

以下では、図４及び５を参照して、アトミックスワップがどのように実行されるかを説明する。

【０１０２】

＜ハッシュパズル又は秘密鍵パズルを用いるアトミックスワップ＞

Aliceがシークレットの所有者であるとする。このシークレットは、知られているハッシュダイジェストのハッシュプレイメージ、又は知られている公開鍵の秘密鍵であってよい。BobがAliceからこのシークレットを購入するためにBitcoinを使用したいとする。アトミックスワップとして知られているメカニズムは、このトランザクションが生じることを可能にするものと説明される。それは、AliceがBitcoinを支払われ、シークレットがBobに開示される、又は何もイベントが生じない、という意味でアトミックである。

【０１０３】

当該方法は、以下の通りである。

Aliceは、公開／秘密鍵ペア$P_A=S_A \cdot G$の秘密鍵S_Aを所有し、Bobは、公開／秘密鍵ペア$P_B=S_B \cdot G$の秘密鍵S_Bを所有する。

Aliceは、知られているハッシュダイジェストH(X)のプレイメージX、又は知られている公開鍵$P_1=S_1 \cdot G$の秘密鍵S_1であるシークレットを所有する。

彼らは、AliceがBobにシークレットを売るBitcoinの価格に合意している。

これらに先立ち、Bobは、Aliceがデジタル署名のコンポーネントであるr_0を計算できるように、Aliceに一時鍵k_0を送信するためのトランザクションをオフブロックで設定しなければならない。

【０１０４】

ここで、図４を参照する。

１．Ｂｏｂは、Ａｌｉｃｅに、ＲｅｄｅｅｍスクリプトR（概略的に記述される）によりロックされた資金を送金する。

ハッシュプレイメージについて、

【数２４】

$$R = [\text{Hash Puzzle } H(X)]\ [\text{CheckSig } P_A]$$

これは、Redeemスクリプトのインプットの中でプレイメージXを公開させる。

秘密鍵について、

【数２５】

$$R = [\text{Private Key Puzzle } P_1, r_0]\ [\text{CheckSig } P_A]$$

これは、Redeemスクリプトへのインプットから秘密鍵S_1を計算できるようにする。この場合、Bob及びAliceは、r_0を構成するために使用される一時鍵k_0について合意しなければならない。ここで、$(r_0, R_y) = k_0 \cdot G$である。

【０１０５】

２．Aliceは、彼女のシークレット（X又はS_1）を知っているので、彼女は、図５に示すトランザクションを用いて、Bitcoinブロックチェーン上で彼女の資金を使用できる。これは、Bobが彼女のシークレットを決定することを可能にする。

【０１０６】

任意的なセキュリティの特徴として、Alice及びBobは、両方のパーティにのみ知られている共有シークレットSを確立するために、彼らの公開鍵P_A、P_Bを使用してよい。これは、国際特許公開番号WO２０１７/１４５０１６号に概説された方法で達成され得る。この場合、Xがブロックチェーン上で公衆に開示されないようにするために、Sがハッシュパズルの中のプレイメージXに加算されてよい。同様に、秘密鍵パズルでは、Alice又はBobのみが秘密鍵を計算できることを保証するために、Sが、一時鍵k_0として使用されてよい。

【０１０７】

Aliceが彼女の資金を支払わない場合に、Bobの資金がAliceによりロックされるのを防ぐために、タイムロックされた資金が、手順（procedure）に導入され得る。

【０１０８】

＜トークンを用いる購入＞

上述のものと同じ状況が存在するとする。しかし、Aliceのシークレットのために暗号通貨を支払う代わりに、その使用時点で、Bobは、シークレットと引き換えに、前もって購入しておいたアクセストークンを償還（redeem）することを望み得る。

【０１０９】

Alice及びBobが従わなければならない手順は、前の章で説明した場合と同様であるが、代わりに同様のアトミックスワップのシーケンスを使用する。処理の２つのフェーズが存在する。つまり、トークン発行及びトークン償還である。

【０１１０】

フェーズ１：トークン発行

トークン発行フェーズは、事実上、Bobによるトークンの１回限りの購入である。例えば、Aliceが１０個の異なるシークレット$X_1,X_2,...,X_{10}$を有し、Bobが、それぞれ彼にそれぞれのシークレットへのアクセスを許可する１０個のトークン$T_1,T_2,...,T_{10}$を１回購入することを望むシナリオを考える。

【０１１１】

先ず、Bobは、彼にのみ知られているシークレットシード値Yから１０個のトークンのセットを生成する。これらのトークンは、シードの順次ハッシングにより生成され、ハッシュチェーンを形成する。ここで、各トークンは次式のように計算される。

【数２６】

$$T_i = H^{10-i}(Y) \quad for \quad i \in \{1,2,\dots,10\}$$

Alice及びBobは、今や１０個のシークレット値を有する。これらは、例えばトークンの償還のために、ハッシュパズルの中で開示され得る。これらのトークンを発行するために、しかしながら、彼らは、それぞれシークレット初期化値I_{Alice}及びI_{Bob}も生成しなければならない。これらは次式のように与えられる。

【数２７】

$$I_{Alice} = k, \qquad k \in \mathbb{Z}_{256},$$
$$I_{Bob} = H^{10}(Y)$$

留意すべきことに、Aliceのイニシャライザは単なるランダム整数であり、特別な意味を持たないが、Bobのイニシャライザは、彼の最初のトークン$T_1=H^9(Y)$のハッシュでなければならない。この方法でトークンのチェーンをイニシャライザに拡張することは、トークンの発行が、連続する償還のために後に使用されるべきトークンを定めることも可能にする。全体で、各参加者により保持されるシークレット値は図６に示される。

【０１１２】

ここで、Alice及びBobは、１０個のトークンの購入のために１０個の暗号通貨単位の価格について合意できる。これらのトークンの購入は、多数の方法で生じることができ、これは、ここではアトミックスワップを用いて説明される。アトミックスワップは、それぞ

れ図７及び８に示されるトランザクションをブロードキャストするAlice及びBobにより開始される。両方のトランザクションの中で、アウトプットは、２つのハッシュパズルに対する解と、有効な署名と、を必要とする。

【０１１３】

両方のトランザクションがブロックチェーンに現れると、Alice及びBobは、彼らのイニシャライザ値I_{Alice}及びI_{Bob}を共有し、トークン発行のためのアトミックスワップを完了できる。

【０１１４】

このアトミックスワップの結果として、Aliceは、１０個のトークンの購入のための支払いを受け取り、両方のイニシャライザシークレットが開示される。留意すべきことに、ここでBobのシークレット$I_{Bob}=H^{10}(Y)$のみが意味がある。なぜなら、それが解かれるべき最初のハッシュパズル[HashPuzzle (T_1)]を定義するからである。このパズルの解は、イニシャライザ$H^{10}(Y)$のプレイメージ$H^{9}(Y)$である。

【０１１５】

フェーズ２：トークン償還

将来のある時点で、Bobは彼の最初のトークン$T_1=H^{9}(Y)$を償還し、彼の最初のシークレットX_1を受け取りたいと望むが、彼は既に、有効なトークンを購入することにより、このシークレットに対して支払っている。トークンを償還する処理は、別のアトミックスワップの形式を取る。ここで、ロックハッシュパズルの解は、トークンT_i、及び対応するシークレットX_iである。

【０１１６】

彼のトークンを償還するために、Bobは、図９に示す、アウトプットが２つのハッシュパズルによりロックされているトランザクションをブロードキャストしなければならない。Aliceは、このトランザクションを見ると、は、図１０に示す、アウトプットが同じ２つのハッシュパズルによりロックされている彼女自身の同様のトランザクションをブロードキャストする。２人の参加者は、今や、彼らのシークレットT_1及びX_1を交換し、これらのトランザクションのアウトプットをアンロックできる。両方のパーティは、今や、両方のシークレットも公開する正しいアンロックスクリプトを提供することにより、僅かな料金ｘを償還できる。これらのアンロックスクリプトを有するトランザクションは、図１１及び１２に示される。

【０１１７】

トークンを償還するこのアトミックスワップの完了は、額ｘが両方のパーティにロックされたアウトプットを使用させるのを奨励するのに十分大きいならば、Aliceの最初のシークレットX_1をBobに開示し、Bobの最初のシークレットT_1をAliceに開示し、暗号通貨資金の総額ゼロの交換を有する。重大なことに、これは、Bobが使用できる次のトークンが、ハッシュパズル[Hash Puzzle $H(T_2)$]に対する解T_2でなければならないことも確立する。ここで、ターゲットハッシュ$H(T_2)=T_2$はAliceに開示される。この処理は、Bobが彼の最後のトークンT_{10}Yを使用するまで、再帰的に繰り返すことができる。

【０１１８】

＜命名及びアドレス指定＞

＜ノード及びエッジ構造＞

トランザクション内でデータを提供することにより、ブロックチェーンにデータをどのように挿入できるかを以上に説明した。私たちは、以下に、ノードのアドレス指定、許可、及びコンテンツバージョン制御を可能にする論理的な方法で、これらのトランザクションを構造化するプロトコルを提示する。この分散型ピアMetanetの構造は、既存のインターネットと類似している。

【０１１９】

留意すべきことに、これは、基礎にあるブロックチェーンのプロトコル又は総意のルールを変更しない「tier－２」プロトコルである。

【０１２０】

ここで説明する構造の目的は以下のものである。
（ｉ）異なるトランザクション内の関連するコンテンツを関連付け、データの検索、識別、及びアクセスを可能にする。
（ｉｉ）人間の読めるキーワード検索を使用したコンテンツの識別を可能にして、検索の速度、精度及び効率を向上する。
（ｉｉｉ）ブロックチェーン内にサーバのような構造を構築しエミュレートする。
【０１２１】
私たちのアプローチは、有向グラフとしてMetanetに関連付けられたデータを構造化することである。このグラフのノード及びエッジは、以下に対応する。
【０１２２】
ノード（Node）：Metanetプロトコルに関連付けられたトランザクション。ノードはコンテンツを格納する。（用語「コンテンツ」及び「データ」は、本願明細書の中で同義的に使用されることがある）。
【０１２３】
ノードは、直後に〈Metanet Flag〉のあるOP_RETURNを含むことにより、生成される。各ノードは、公開鍵P_{node}で署名される。公開鍵とトランザクションIDとの組合せは、以下のノードのインデックスをユニークに指定する。
【数２８】

$$ID_{node} := H(P_{node}||TxID_{node})$$

【０１２４】
使用されるハッシュ関数は、基礎にあるブロックチェーンプロトコルと一貫していなければならない。本発明は、Bitcoinの場合にはSHA－２５６又はRIPEMD－１６０と共に使用されるべきである。
【０１２５】
エッジ（Edge）：子ノードの親ノードとの関連付け。
エッジは、署名Sig P_{parent}がMetanetトランザクションのインプットの中に現れるときに生成される。従って、親のみがエッジを生成することを許可され得る。全部のノードは、最大で１個の親を有してよく、親ノードは、任意の数の子を有してよい。グラフ理論の言葉で言うと、各ノードの入次数は最大で１であり、各ノードの出次数は任意である。
【０１２６】
留意すべきことに、エッジは、Metanetプロトコルの一側面であり、それ自体が基礎にあるブロックチェーンに関連付けられたトランザクションではない。
【０１２７】
（親を有する）有効なMetanetノードは、以下の形式のトランザクションにより与えられる。
［表１］
【表１】

$TxID_{node}$	
Input	Output
$<Sig\ P_{parent}><P_{parent}>$	OP_RETURN <Metanet Flag> $<P_{node}>$ $<TxID_{parent}>$

【０１２８】
このトランザクションは、ノードのインデックス及びその親を指定するために必要な全部の情報を含む。
【数２９】

$$ID_{node} = H(P_{node}||TxID_{node}) \ , \quad ID_{parent} = H\left(P_{parent}||TxID_{parent}\right)$$

【０１２９】
更に、親ノードの署名が要求されるので、親のみが子へのエッジを生成できる。〈T_xID$_{parent}$〉フィールドが存在しない、又はそれが有効なMetanetトランザクションを指さない場合、ノードは親のない子である。それは、それにより到達可能な、より上のレベルのノードを有しない。
【０１３０】
追加属性が、各ノードに追加されてよい。これらは、フラグ、名称、及びキーワードを含んでよい。これらは、本願明細書の中で後述される。
【０１３１】
図示のように、ノード（トランザクション）のインデックスは以下に分解できる。
ａ）公開鍵（P_{node}）：これは、ノードのアドレスとして解釈される。
ｂ）トランザクションＩＤ（$TxID_{node}$）：これは、ノードのバージョンとして解釈される。
【０１３２】
この構造化から、２つの有利な特徴が生まれる。
１．バージョン制御：同じ公開鍵を有する２個のノードが存在する場合、最大のproof－of－workを有するトランザクションIDを有するノードを、該ノードの最新バージョンとして解釈する。ノードが異なるブロックに存在する場合、これは、ブロック高によりチェックできる。同じブロック内のトランザクションについては、これは、トポロジトランザクション順序付けルール（Topological Transaction Ordering Rule (TTOR)）により決定される。
２．許可：ノードの子は、公開鍵P_{node}の所有者が子ノードの生成においてトランザクションインプットに署名した場合にのみ、生成され得る。従って、P_{node}は、ノードのアドレスだけでなく、子ノードの生成の許可も表す。これは、標準的なBitcoinトランザクションと意図的に類似している。つまり、公開鍵は、アドレスだけでなく、該アドレスに関連付けられた許可でもある。
【０１３３】
親ノードの署名がUXTOアンロックスクリプト内に現れるので、トランザクションがネットワークに受け入れられた時点で、標準的なマイナー検証処理を通じて検証されることに留意する。これは、子ノードを生成する許可が、Bitcoinネットワーク自体により検証されることを意味する。
【０１３４】
標準的なインターネットプロトコル（IP）アドレスは、特定の時点でネットワーク内でのみユニークであることは留意する価値がある。他方で、Metanet内のノードのインデックスは、いつでもユニークであり、別個のネットワークという概念がない。これは、データが単一のオブジェクトID$_{node}$に永久にアンカーされることを可能にする。
【０１３５】
ノード及びエッジ構造は、図１３に示すように、Metanetをグラフとして視覚化することを可能にする。
【０１３６】
＜Metanet内のドメイン、命名、及びコンテンツの位置特定＞
Metanetグラフの階層構造は、豊富なドメインのような構造を出現させる。私たちは、親のないノードを最上位ドメイン（top－level domain (TLD)）として、親のないノードの子をサブドメインとして、孫をサブサブドメイン等として、子のないノードをエンドポイントとして解釈する。図１３を参照する。
【０１３７】
ドメイン名は、ID$_{node}$として解釈される。Metanet内の各最上位ドメインは、親のない

ノードであるルートと子のないノードであるリーフとを有する木として考えられてよい。Metanet自体は、グラフを形成する木のグローバルコレクションである。

【０１３８】

Metanetプロトコルは、任意のノードがコンテンツデータを含むことを規定しないが、リーフ（子）ノードは、データ木における有向パスの端を表し、従って通常、コンテンツデータを格納するために使用される。しかしながら、コンテンツは、木の中の任意のノードに格納されてよい。ノードに属性として含まれるプロトコル固有のフラグは、データ木の中のノードの役割（ディスク空間、フォルダ、ファイル、又は許可の変更）を指定するために使用されてよい。

【０１３９】

インターネットはドメインネームシステム（Domain Name System（DNS））を使用して、インターネットプロトコル（IP）に人間の読める名称を関連付けることを思い出してほしい。DNSは、ある意味で非集中化されているが、実際には、政府や大企業のような少数の主要プレイヤにより制御される。DNSプロバイダによっては、同じ名称でも異なるアドレスに移動する場合がある。この問題は、短い人間の読める名称をコンピュータの生成した数値にマッピングするとき、発生する。

【０１４０】

私たちは、人間の読める上位レベルのドメイン名をルートノードの非集中化インデックスID_{root}にマッピングする、等価分散型システムが存在すると仮定する。言い換えると、人間の読める名称をMetanetルートノードインデックスにマッピングする、以下の１対１関数kが存在する。

【数３０】

$$\kappa('bobsblog') = ID_{bobsblog}\left(= H\left(P_{bobsblog}||TxID_{bobsblog}\right)\right)$$

左辺への入力は、人間の読める単語である。一方で、右辺の出力は、ハッシュダイジェストであり、標準的に２５６ビットのデータ構造である。$P_{bobsblog}$及び$TxID_{bobsblog}$も、通常、人間には読めないことに留意する。標準的なIPプロトコルでは、これは、www.bobsblog.comから、ネットワーク内の対応するドメインのIPアドレスにマッピングされる。

【０１４１】

マップｋは、DNSの発行したドメイン名の人間による可読性を再現する際に、Metanetのインターネットとの後方互換性を保証するための手段として解釈されるべきであるが、Metanetの構造を提供する命名及びアドレス指定方式は、このマップに明示的に依存しない。

【０１４２】

マッピング関数kの存在し得る形式は、IPFS（Interplanetary File System）より利用されるDNSLinkシステム、又はOpenNICサービス（https://www.openic.org）を含む。このマッピングは、既存のTXTレコードの中に、DNSの部分として格納できる。これは、IPFSにおけるDNSLinkと類似している。https://docs.ipfs.io/guides/concepts/dnslink/を参照のこと。しかしながら、通常、これらは１対１であるマップを提供するために、非集中化の何からの要素を犠牲にする。https://hackernoon.com/ten－terrible－attempts－to－make－the－inter－planetary－file－system－human－friendly－e４e９５df０c６faを参照のこと。

【０１４３】

＜バニティアドレス＞

Metanetノードのアドレスとして使用される公開鍵は、人間の読めるオブジェクトではない。これは、活動の検索、参照、及び入力を、人間のユーザにとって間違えやすく遅いものにし得る。しかしながら、ユーザにより直接解釈可能な平文プレフィックスを含む、人間の認識できる公開鍵アドレス、つまりバニティアドレスP_{vanity}を生成することが可能

である。バニティアドレスは、従来知られている。

【０１４４】

このようなアドレスを生成する際の難しさは、所望のプレフィックスの文字長に依存する。これは、人間の認識可能なバニティアドレスが、中央による発行ではなく、生成するための所有者の努力にのみ依存するノードアドレスとして使用されてよいことを意味する。所与のプレフィックスに対して、サフィックスの中の残りの文字により、多数の異なるバニティアドレスが存在する。従って、ユニーク性を保持したまま、多くのノードアドレスが共通のプレフィックスを共有できる。

【０１４５】

望ましいプレフィックスを有するバニティアドレスの例は、以下の通りである。

Pbobsblog：bobsblogHtKNngkdXEeobR７６b５３LETtpyT

Prefix：bobsblog

Suffix：HtKNngkdXEeobR７６b５３LETtpyT

【０１４６】

上記のバニティアドレスは、名称「bobsblog」からノードインデックス$ID_{bobsblog}$へのマップを確認（sense check）するため、及びアドレスによりMetanet内ノードの検索可能性を支援するために使用されてよい。プレフィックスはここではユニークではないが、全体のアドレス自体はユニークなエンティティであることに留意する。

【０１４７】

選択されたアドレスP_{vanity}のTxIDとの組合せは、ID_{node}を形成し、それがドメイン名の中央ではない発行者が存在することを意味し（TxIDは非集中化されたproof－of－workにより生成される）、名称がブロックチェーン自体から復元可能なので、有利である。有利なことに、インターネットDNS内に存在する障害点はもはや存在しない。

【０１４８】

Metanetドメインは既に許可システム（公開鍵）を提供しているので、所有権を証明するための証明書を発行する必要がない。この目的のためにブロックチェーンを使用することは、例えばnamecoin（https://namecoin.org/）において探求されている。本発明によれば、しかしながら、全てのことが１つのブロックチェーンの中で達成されるので、この機能のために別個のブロックチェーンを使用する必要がない。

【０１４９】

これは、従来技術と比べて、本発明により必要とされるリソース（ハードウェア、処理リソース、及びエネルギ）の量を有意に低減する。本発明は、また、機器及びシステムコンポーネントの構成の観点で、完全に異なるアーキテクチャを提供する。

【０１５０】

この命名システムの利点は、ユーザが、ハッシュダイジェストではなく記憶しやすい単語（例えば、会社名）によりMetanet内の最上位ドメインを識別できることである。これはまた、ハッシュダイジェストよりもキーワードに対して検索する方が速いので、ドメインに対する検索を速くする。それは、入力誤りを低減し、従って、ブロックチェーンに格納されたデータに対する改良された検索ツールを提供する。

【０１５１】

私たちはドメイン名からノードインデックスへのマップを有するので、インターネットのURL（Uniform Resource Locator）のものと同様のリソースロケータを構築できる。私たちは、これをMetanet URL（MURL）と呼び、以下の形式である。

【数３１】

$$MURL = \text{'mnp:'} + \text{'//domain name'} + \text{'/path'} + \text{'/file'}$$

【０１５２】

URLのコンポーネントの各々、つまり、プロトコル、ドメイン名、パス、及びファイルは、MURLの構造にマッピングされており、オブジェクトをユーザにとってより直感的に

し、インターネットの既存の構造と統合できるようにする。

【０１５３】

これは、各ノードが、ドメイン木の中のレベルにおいてユニークな、その公開鍵（アドレス）に関連付けられた名称を有することを前提としている。この名称は、常に、所与のノードについて、MURLの最も右のコンポーネントである。木の中の同じレベルにある２個のノードが同じ名称を有する場合、それらは、同じ公開鍵を有し、従って最新のバージョンが取り入れられる。

【０１５４】

以下の表は、Metanetプロトコルとインターネットプロトコルとの間の類似性を与える。

［表：インターネットとMetanetプロトコルとの間の類似性のまとめ］

【表２】

Internet	Metanet
ウェブサイト／ファイル	ノード
所有者	公開鍵 P_{node}
IPアドレス(非ユニーク)	ノードインデックス (ユニーク) $ID_{node} = H(P_{node}\|\|TxID_{node})$
ドメイン構造	ノード木構造
ドメインネームシステム（DNS）	ルートノード名からノードインデックスへのマッピング
URL http://www.bobsblog.com/path/file	**MURL** mnp://bobsblog/path/file

【０１５５】

＜Metanetの検索＞

私たちは、各ノードがユニークなインデックスを有し、それに属する名称を有し得る、Metanetグラフ構造の説明のための実施形態を定めた。これは、MURLを使用してコンテンツの位置を特定することを可能にする。また、迅速な検索機能を可能にするために、私たちは、追加キーワードをノードに帰属させた。

【０１５６】

ノードの固定属性は、インデックス及び親ノードのインデックスであり、任意属性は名称及びキーワードである。

【数３２】

Node attributes

{

index: $H(P_{node}||TxID_{node})$;

index of parent: $H(P_{parent}||TxID_{parent})$; (NULL if orphan)

name: ‘bobsblog’;

kwd1: ‘travel’;

kwd2: ‘barbados’;

⋮

}

【０１５７】

一例では、Metanetを検索する実際の方法は、先ず、ブロックエクスプローラを使用し

て、ブロックチェーンを通じてトロール（trawl）してよく、Metanetフラグを有する全部のトランザクションを識別し、それらが有効なMetanetノードであることを調べ、有効である場合、それらのインデックス及びキーワードをデータベース又は他の記憶リソースに記録する。このデータベースは、次に、所望のキーワードを有するノードを効率的に検索するために使用できる。所望のキーワードを有するノードのインデックスが見付かると、そのコンテンツが、ブロックエクスプローラから読み出され、閲覧できる。

【０１５８】

例として、図１４のブランチP_1を考えると、公開鍵P_0、P_1、$P_{1,1}$に対応するノードは、それぞれホームページ、トピックページ及びサブトピックページを表す。これらのノードは、名称「bobsblog」、「summer」、及び「caribbean」を与えられ、それらの属性は以下に示される。

【数３３】

Home page node P_0

MURL: mnp://bobsblog

{

index: $H(P_0||TxID_0)$;

index of parent: $NULL$

name: ‘bobsblog’;

kwd1: ‘travel’;

kwd2: ‘barbados’;

⋮

}

Topic page node P_1

MURL: mnp://bobsblog/summer

{

index: $H(P_1||TxID_1)$;

index of parent: $H(P_0||TxID_0)$;

name: ‘summer’;

kwd1: ‘travel’;

kwd2: ‘barbados’;

⋮

}

Sub-topic page node $P_{1,1}$

MURL: mnp://bobsblog/summer/caribbean

{

index: $H(P_{1,1}||TxID_{1,1})$;

index of parent: $H(P_1||TxID_1)$;

name: ‘caribbean’;

kwd1: ‘travel’;

kwd2: ‘barbados’;

⋮

}

【０１５９】

この例では、リーフノードＰ１,１,１、Ｐ１,１,２、及びＰ１,１,３は、それぞれ名称「beaches」、「nightlife」、及び「food」を与えられ、別個のブログ投稿を格納するために使用される。木の中の各ノードに関連するＭＵＲＬ検索パスを含む、完全なドメイン構造が、図のリーフの上に示される。

【０１６０】

Ｍｅｔａｎｅｔは、追加属性としてノードトランザクションにより格納されたコンテンツのハッシュを格納することにより、コンテンツアドレス可能なネットワーク（ｃｏｎｔｅｎｔ ａｄｄｒｅｓｓａｂｌｅ ｎｅｔｗｏｒｋ （ＣＡＮ））も組み込むことができる。これは、Ｍｅｔａｎｅｔノードが、イン

デックス付けされ、コンテンツハッシュにより検索できることも意味する。
【０１６１】
上述の命名及びアドレス指定方法は、従来技術に対し、以下を含む多数の技術的利点を提供する。
１．公開鍵アドレス：システムは、ブロックチェーンのように、ノードアドレスを割り当てるために同じ公開－秘密鍵ペアを使用する。これは、同じ鍵セットが、暗号通貨資金の管理、及びコンテンツデータの許可の両方のために使用されることを意味する。これは、効率的且つセキュアなソリューションを提供する。
２．非集中化ドメイン：proof－of－workによってのみ生成できる$TxID_{node}$を含むことを通じて、ドメイン名の発行が、完全に非集中化される。ドメイン名は、所望のドメイン公開鍵の公平な分配を可能にする、人間の認識可能な公開鍵P_{vanity}（バニティアドレス）も組み込むことができる。ここでも、このソリューションは、向上した効率及びセキュリティを提供する。
３．グラフ構造：命名及びアドレス指定アーキテクチャは、Metanetノードを含むブロックチェーンデータのサブセットから構成できるグラフを指定する。この設計は、順序付けられた構造を用いて、インターネットの複雑性をブロックチェーンにマッピングする。その結果、それは、セキュアなままに、その機能及び拡張性を完全に再現する。
【０１６２】
＜ブラウザ－ウォレットアプリケーション＞
Metanetプロトコルでは、全部のデータがブロックチェーン自体に直接存在することを思い出してほしい。この章で、私たちは、説明のためのコンピュータアプリケーションの実施形態を提示する。ここで、私たちは、便宜上、ブロックチェーンに格納されたMetanetデータに効率的にアクセスし、表示し、及び相互作用できる「ブラウザ－ウォレット」を参照する。
【０１６３】
私たちは、この章の残りの部分で詳細な説明を提供する前に、ブラウザ－ウォレットが分散型ピアインターネットとどのようにインタフェースするかというコアコンポーネント及び機能の議論から始める。
【０１６４】
＜概要＞
＜コンポーネント＞
ブラウザ－ウォレットは、エンドユーザがブロックチェーン上のMetanetインフラストラクチャと相互作用することを可能にするアプリケーションである。このアプリケーションは、木に埋め込まれた特定のコンテンツについて、Metanetグラフの探索的検索を可能にするべきである。更に、ブラウザ－ウォレットは、コンテンツの読み出し、解読、再結合及びキャッシュ（任意）を扱う。
【０１６５】
ブラウザ－ウォレットアプリケーションは、ネイティブ（又は外部）ウォレットをサポートすることにより、これらの要素を暗号通貨支払いメカニズムと結合する。ブラウザ－ウォレットは、単一のコンピュータアプリケーションに結合される以下のコア要素を含む。
【０１６６】
ブロックチェーン検索エンジン：ID_{node}、ノード名、キーワード、ブロック高、及びTx IDを含む種々のインデックスにより、Metanetノードをクエリするために第三者検索エンジンをサポートする。
【０１６７】
ディスプレイウインドウ：フルコピーブロックチェーンピアによりブラウザに返されるコンテンツをパック解除（unpack）するソフトウェア。これは、アクセストークンの解読、再結合、キャッシング、及び償還をカバーする。
【０１６８】
暗号通貨ウォレット：ブロックチェーンの通貨の専用鍵管理。アプリケーションに本来

備わっているか（ネイティブ）、又は外部ウォレット（ソフトウェア又はハードウェア）との通信及び同期を許可できる。標準的なブロックチェーントランザクション、並びに新しいMetanetノードトランザクションを書き込むことができる。アクセス鍵及びアクセストークンのオンチェーンでの購入を調停できる。

【０１６９】

暗号通貨公開鍵及びMetanetノードアドレスの両方のために、階層的な決定性鍵管理が利用される。

【０１７０】

アクセス鍵／トークンウォレット：購入したアクセス鍵又はトークンのための専用鍵管理。暗号通貨ウォレットを用いて購入した鍵又はトークンを受信できるが、それらに対する許可を有しない。それらは、後に満期になるよう、ユーザから隠されてよい。これは、信頼できる実行環境の使用を通じて達成されてよい。ブロックチェーンと同期化し、及び現在ブロック高をクエリすることにより、時間指定されたアクセスがセキュアになる。

【０１７１】

＜機能＞

Metanetブラウザ－ウォレットの仕様は、アプリケーションの以下の機能を保証する。

１．階層的鍵管理：資金を制御するため及びMetanet木（グラフ）を管理するために使用される鍵は同じ階層的決定性鍵インフラストラクチャを利用し、Metanetコンテンツのための鍵記録を保持するユーザの負荷を軽減する。

２．外部暗号通貨ウォレットを指す：外部（アプリケーションに本来備わっていない）ウォレットの認証及び同期化の能力は、障害点としてのブラウザ－ウォレットを除去することにより、更なるセキュリティを可能にする。

アプリケーションは、ブロックチェーントランザクションを記述し、鍵を収容する外部ウォレットの署名を要求し、この責任を別個のソフトウェア又はハードウェアへと委任する。

３．Metanetコンテンツの検索：ブラウザ－ウォレットは、グローベルデータベース内のMetanetノードトランザクションデータをクロールし、インデックス付けし、サービスし、及びランク付けすることを含み得る機能を有する第三者検索エンジンをサポートし及びクエリできる。Metanetプロトコルフラグを含むOP_RETURNトランザクションのデータベースが構成されてよい。BitDB２.０－https://bitdb.network/を参照のこと。

検索エンジンは、データを発見可能にするノードインデックスを、ブラウザ－ウォレットに提供できる。

４．データの読み出し及びブロックチェーンへの書き込み：検索エンジン及びフルノードを使用してブラウザにコンテンツを提供することに加えて、暗号通貨ウォレットのサポートは、ブラウザ－ウォレットからMetanetにコンテンツを直接書き込むことも可能にする。

５．データの伸長及び解読：ブラウザ－ウォレットは、解読鍵を扱い、Metanetコンテンツの伸長をその場で実行できる。

６．ノード識別子（ID_{node}）のキャッシング：より効率的なルックアップ及びクエリのために、ユニークなノード識別子は、ローカルにキャッシュされ得る。

７．ウェブサーバのバイパス：ノードインデックスが与えられると、ブラウザ－ウォレットは、ピアツーピア（peer－to－peer（Ｐ２Ｐ））ブロックチェーンネットワークの任意のフルコピーメンバに、ノードに置かれたコンテンツについてクエリできる。Metanetはオンチェーンに存在するので、任意のフルコピーピアは、ノードのローカルコピー及びそのコンテンツを有していなければならない。

これは、ユーザのブラウザ－ウォレットが、単一のピアをクエリするだけでよいことを意味する。これは、中間ウェブサーバを必要とせず、直接に行うことができる。

図１５は、ブラウザ－ウォレットの概略、及びそのコア機能がアプリケーションの異なるコンポーネントに渡りどのように分割されるかを示す。

【０１７２】

＜ブロックチェーン検索エンジン＞

検索エンジン：既存技術

従来知られている検索エンジン（Search engine (SE)）は、強力なウェブクローラがユーザクエリに従いウェブコンテンツの位置を特定し、インデックス付けし、ランク付けすることに頼っている。（同じ基礎にある原理は、Metanetをクロールする第三者ブロックチェーンSEへと拡張できる）。

【０１７３】

SEは、クエリの中のキーワードの検索を通じて、関連するHTMLメタタグ及びコンテンツを識別する。クロールの結果は、後にインデックス付けされる。ここで、任意の埋め込み画像／ビデオ／メディアファイルが、分析され、分類される（catalogued）。次に、ユーザの場所、言語、及び装置を考慮して、インデックスの中から最も関連する結果が、プログラムによりランク付けされる。

【０１７４】

典型的なSEは、以下の機能を有するべきである。

１．クロール：インターネットデータを識別し、ドメイン名、リンクされたページ、及び関連するキーワードのような関連するメタデータを通じてクロールする。新しいインターネットコンテンツは、既存のコンテンツを通じて発見され、任意の関連情報について更にクロールされる。

２．インデックス付け：コンテンツデータは、分析され、分類される（catalogued）。この情報は、データベースに格納される。

３．提供及びランク付け：コンテンツインデックスは、ユーザのクエリへの関連の順序でランク付けされる。

【０１７５】

＜ブロックエクスプローラ＞

インターネット検索エンジン（SE）に類似する最も近いブロックチェーンは、ブロックチェーンエクスプローラ（blockchain explorer）であり、「ブロックエクスプローラ（block explorer）」又は「ブロックチェーンブラウザ（blockchain browser）」と呼ばれることがある。ブロックチェーンエクスプローラは、高レベルでのブロックチェーンのユーザフレントリなクエリを可能にするウェブアプリケーションであり、ウェブブラウザと同様に機能するが、インターネットではなくブロックチェーンに接続される。https://en.bitcoin.it/wiki/Block_chain_browserを参照のこと。

【０１７６】

大抵の場合、これらのエクスプローラは、ブロック（ブロックヘッダのハッシュによりインデックス付けされる）、トランザクション（TxIDによりインデックス付けされる）、アドレス及び未使用トランザクションアウトプット（unspent transaction output (UTXO)）をインプットとして取り入れること及び検索することを可能にする。多くのエクスプローラは、生トランザクション及びブロックデータを読み出すために、それら自体のアプリケーションプログラミングインタフェース（application programming interface (API)）を提供する。Https://blockexplorer.com/api－refを参照のこと。

【０１７７】

ブロックエクスプローラは、それらの能力を変化させながら、通常、トランザクションを分類し、及び取引される通貨価値、コイン及びアドレスの確認及び履歴のようなそれらの基本情報をユーザにとって理解の容易な形式で表示するために有用である。Bitcoin.com https://explorer.bitcoin.com/bch及びBlockchain.com https://www.blockchain.com/explorerのような多くのエクスプローラは、これらの及びBlockchair https://blockchair.com/のようなより高度なサイトの間で、この情報を選択して提供する方法に不一致が存在するが、更に、トランザクションの個々のインプット及びロックスクリプトを閲覧可能にする。

【０１７８】

近年、ブロックチェーンデータに基づきウェブアプリケーションを実行するために使用される基本ブロックチェーンエクスプローラの多くの拡張が存在している。Memo.cash

https://memo.cash/protocol及びMatter https://www.mttr.app/homeのようなこれらのアプリケーションは、特定のプロトコル識別子を含むブロックチェーントランザクションを分類し組織化する、並びにそれらの特定のトランザクションの中で符号化されたデータを表示するブロックエクスプローラと同様に動作する。

【０１７９】

しかしながら、ブロックチェーンエクスプローラの使用に伴う以下の２つの重要な問題がある。これらは、本発明の実施形態により解決される。

１．ユニバーサル性：トランザクションに格納されたコンテンツデータを閲覧するための業界に渡る標準が現在存在しない。コンテンツデータは、基礎にあるブロックチェーンを生成し及びセキュアにするために使用されるプロトコルに関連しない任意のデータを表す。

２．キーワード検索：トランザクションに格納されたコンテンツデータは、人間の読めるキーワードにより検索可能である必要がある。現在のブロックエクスプローラは、検索入力としてキーワードを取り入れるのではなく、ブロック高、TxID、及びアドレスのようなトランザクションのプロトコルに基づく特性をクエリするために使用されるので、キーワード検索は、通常、現在のブロックエクスプローラの機能ではない。（しかしながら、幾つか、例えばBlockchairは、単語がトランザクションのスクリプトに直接含まれる場合には、単語を検索できる）。

【０１８０】

重要なことに、本発明の強力な命名及びアドレス指定構造は、上述のように、従来知られているよりも、一層洗練されたブロックチェーンエクスプローラの構築を実現し及び可能にする。

【０１８１】

＜提案されるMetanet検索エンジン＞

ブラウザ－ウォレットアプリケーションは、ノード識別子（ID_{node}）の発見のために、第三者検索エンジンと通信する。このような第三者は、既存のインターネット検索エンジンの能力を再現する強力な多様なサービスを提供し得ると考えられる。

【０１８２】

Metanet検索エンジンの第三者は、Metanetプロトコルフラグにより識別可能な、ブロックチェーンに埋め込まれた（mined）全部のMetanetトランザクションのデータベースを維持する。このデータベースは、ID_{node}、ノード名、キーワード、TxID、及びブロック高を含む範囲インデックスにより、全部のMetanetノードを分類できる。

【０１８３】

Bit DB https://bitdb.network/のようなサービスが既に存在する。これは、ブロックチェーンと絶えず同期し、標準的なデータベースフォーマットでトランザクションデータを維持する。ブラウザ－ウォレットは、Metanetトランザクションをクロールし、インデックス付けし、提供し、及びランク付けする責任を、このような第三者にオフロードし、Metanetグラフに格納されたコンテンツを特定するとき、それらのサービスへの接続を行う。

【０１８４】

Metanetデータにのみ専用のデータベースを有することにより、効率的な節約が行われ得る。Bit DBと異なり、これは、全部のトランザクションに関連付けられたデータを格納せず、Metanetフラグを含むデータのみを格納する。MongoDBのような非関係性データベースのような特定のデータベースは、Metanetのグラフ構造を格納する際に更に効率的であり得る。これは、より速いクエリ、より小さな記憶空間、及びMetanetドメイン内の関連コンテンツのより効率的な関連付けを可能にする。

【０１８５】

図１６は、Metanetインフラストラクチャ内のコンテンツをユーザが検索するとき、ブラウザ－ウォレットが第三者検索エンジンとどのように相互作用するかを示す。重要なことに、インターネットと対称的に、ルーティングが必要なく、従って、本発明は効率、速度、処理、及び必要なリソースの点で、重要な利点を提供することに留意すべきである。

【０１８６】
処理は、以下の通りである。
１．エンドユーザは、ブラウザ－ウォレット検索バーにキーワードを入力する。
２．ブラウザ－ウォレットは、キーワードクエリを第三者SEへ送信する。
３．SEは、キーワードを自身のデータベースに対してチェックし、関連コンテンツを含む任意のMetanetノードのID_{node}を返す。第三者は、関連コンテンツの提案を提供すると共に、各ノードにある他のインデックスもユーザに返すことができる。
４．ブラウザ－ウォレットは、MURLを構成するために、ノード識別子及びそれに関連付けられたドメイン名を使用する。
５．ブラウザ－ウォレットは、ブロックチェーンのフルコピーを有する任意のネットワークピアから、特定のノードに属するコンテンツを要求する。
６．ネットワークピアは、ブラウザ－ウォレットに要求されたコンテンツを提供する。ピアは、ブロックチェーンのコピーを有するので、それらは、コンテンツのコピーも有し、従って、１つの要求のみが行われ、それは決して他のネットワークピアへ転送されない。
【０１８７】
第三者SEはMetanetノードの属性の記録をインデックス付けし及び維持する責任のみを有し、一方で、ノードに格納された生コンテンツデータが、代わりに、ブロックチェーンのフルコピーを有するネットワークピア（例えば、フルコピーピア、マイナー、アーカイブ）により格納されることが強調される。
【０１８８】
＜コンテンツ表示－Metanetブラウザ＞
ブラウザ－ウォレットアプリケーションは、任意の標準的なウェブブラウザが提供すべき同じフロントエンド能力をエミュレートする。これらの機能は限定ではないが以下を含む。
１．検索（Searching）：コンテンツの位置を特定するために、検索エンジン（SE）へのアクセスを提供する。
２．読み出し（Retrieval）：既知のプロトコル、例えばハイパーテキスト転送プロトコル（Hypertext Transfer Protocol（HTTP））を用いてコンテンツの転送を実現するために、サーバと津シンする。
３．解釈（Interpreting）：（例えば、JavaScriptの）生コードをパースし、実行する。
４．レンダリング（Rendering）：エンドユーザにより閲覧されるべき、パースされたコンテンツを効果的に表示する。
５．ユーザインタフェース（User interface（UI））：ユーザ入力のための動作ボタン及びメカニズムを含む、コンテンツと相互作用するための直感的インタフェースを提供する。
６．記憶（Storage）：コンテンツへの繰り返しアクセスを向上するために、インターネットコンテンツのキャッシュ、クッキー、等の、ローカル一時的記憶能力。
【０１８９】
特定の実施形態では、ウェブブラウザとして動作することを担うブラウザ－ウォレットアプリケーションのソフトウェアコンポーネントは、それらの属性を用いて（SEを用いて）検索可能且つ（ピアから）読み出し可能なブロックチェーンに埋め込まれたMetanetコンテンツに対して、上述の機能を実行できる。
【０１９０】
＜再結合、圧縮解除、及び解読＞
本発明の特定の実施形態によると、ブラウザ－ウォレットアプリケーションのウェブブラウザソフトウェアコンポーネントは、所与のMetanetコンテンツに対して実行される必要のある全部の動作を処理することができる。一般に実行される必要のある多くのこのような動作が存在するが、私たちは、少なくとも以下のものが、Metanetプロトコル及びインフラストラクチャを用いてアプリケーションにより実行されることを想定する。
【０１９１】
再結合（Recombination）：Metanetコンテンツが構成され複数の別個のノードトラ

ンザクションに挿入される必要のある場合に、アプリケーションは、全部の関連ノードからコンテンツを要求し、元のコンテンツを再構成する。分裂したコンテンツの順序及び構造は、各ノードの属性の中の追加フラグを用いて符号化できる。

【０１９２】

圧縮解除（Decompression）：コンテンツデータが圧縮された形式でブロックチェーンに格納される場合、それは、ブラウザ－ウォレットに、どの標準的な圧縮方式が使用されているかを示すために、フラグを含むべきである。アプリケーションは、このフラグに従いコンテンツを圧縮解除する。

【０１９３】

解読（Decryption）：コンテンツが暗号化されている場合、暗号化方式を指定するために、フラグが使用されるべきである。アプリケーションは、（後述するように）自身の解読鍵ウォレットから鍵を特定し、使用された暗号化方式に従い使用するためにコンテンツデータを解読する。

【０１９４】

これらの動作をコンテンツデータに対して実行する際に、ブラウザ－ウォレットに、所与の動作が実行される必要のあることを示すために、フラグが使用できる。これは、動作の適用されるノードの属性の部分として適切な〈operation_flag〉が含まれ得る任意の他の動作に一般化される。

【０１９５】

＜キャッシング＞

ローカルファイルのキャッシング及びクッキーは、標準的なウェブブラウザの共通の及び重要な機能である。ブラウザ－ウォレットアプリケーションは、ID_{node}の記録及び関心のあるコンテンツに関連する他のノード属性を任意的に保持するために、同様の方法でローカル記憶も使用する。これは、頻繁に訪れるMetanetノードからのコンテンツのより効率的なルックアップ及び読み出しを可能にする。

【０１９６】

Metanetは、可変でありプロバイダによってはウェブ閲覧ソフトウェアにより変更され又は検閲され得るインターネットデータのキャッシングに本来伴う問題を解決する。Metanetデータをキャッシングするとき、ユーザは、それがブロックチェーン上の不変記録として初めに含まれたときと同じ状態にあるかを、常に容易に検証できる。

【０１９７】

＜暗号通貨ウォレット＞

＜階層的決定性鍵管理＞

決定性鍵DKは、単一の「シード（seed）」鍵から初期化された秘密鍵である（Andreas M. Antonopoulos, Chapter 5 in “Mastering Bitcoin” O’Reillynd2 Ed., ２０１７, pp. ９３－９８を参照のこと）。シードは、マスタ鍵として動作するランダムに生成された数値である。決定性鍵を導出するよう、インデックス番号又は「チェーンノード」（HD Wallets－BIP－３２/BIP－４４を参照のこと）のような他のデータとシードを結合するために、ハッシュ関数が使用できる。これらの鍵は、互いに関連付けられ、シード鍵により完全に復元可能である。シードは、異なるウォレット実装の間で、ウォレットの簡単なインポート／エクスポートを可能にし、ユーザがMetanetブラウザ－ウォレットと関連して外部ウォレットを使用したい場合に、更なる自由度を与える。

【０１９８】

階層的決定性（hierarchical deterministic (HD)）ウォレットは、決定性鍵のよく知られた導出方法である。HDウォレットでは、親鍵は一連の子鍵を生成し、また、子鍵は一連の孫鍵を導出する、等である。この木のような構造は、幾つかの鍵を管理するための強力なメカニズムである。

【０１９９】

好適な実施形態では、HDウォレットは、図１６に示すMetanet構造に組み込むことができる。HDウォレットを使用する利点は以下を含む。

１．構造（Structure）：追加の組織的な意味は、異なる目的のために異なる鍵束を使用することを表し得る。例えば、ユーザは、異なるブランチ（及びそれらの対応するサブ鍵）を異なるタイプのデータに専用にすることができる。
２．セキュリティ（Security）：ユーザは、対応する秘密鍵を有しないで、一連の公開鍵を生成でき、受信のみの能力でHDウォレットを機能させ、安全でないサーバでの使用に適するようにする。また、少数のシークレットしか格納される必要がないので、暴露されるリスクが低い。
３．復元（Recovery）：鍵が失われ／壊れた場合、それらはシード鍵から復元できる。
【０２００】
＜本来（内部）及び外部ウォレットのサポート＞
有利なことに、本発明の実施形態は、伝統的なウェブブラウザの機能を、１つ以上の暗号通貨ウォレットと直接マージできる。これは、機能的には、「インターネット」コンテンツに対する支払いを、「インターネット」コンテンツのエンドユーザへの配信ととのように結合するかである。
【０２０１】
これを達成するために、ブラウザ－ウォレットの実施形態は、暗号通貨ウォレットとして動作する専用のビルトインソフトウェアコンポーネントを有してよい。このウォレットは、アプリケーション自体に本来備わっており、暗号通貨鍵を管理するため、及びブラウザ－ウォレット自体の中でMetanetコンテンツについての支払いとしてトランザクションを認可するために使用できる。
【０２０２】
これは、アプリケーションのブラウザコンポーネントが、ウォレットコンポーネントに、Metanetコンテンツを閲覧するために、解読鍵、アクセストークン、又はその他のものを購入することにより、必要な支払いを認可することを促すことができることを意味する。アプリケーションは、支払いを処理するために外部の第三者を呼び出す必要がない。従って、関心のあるMetanetコンテンツは、アプリケーションにより消費され、その場で支払われる。
【０２０３】
＜外部ウォレット＞
ユーザが彼らの暗号通貨秘密鍵を代わりに外部ウォレット（ソフトウェア又はハードウェア）に管理し又は保持したい又は複数のウォレットを使用したいと望む場合でも、同じ利点及び機能が、本願の実施形態により達成できる。これは、アプリケーションの本来のウォレットの代わりに又はそれと関連して実行されてよい。
【０２０４】
このような実施形態では、アプリケーションは、外部ウォレットとのリンク又はペアリングを確立し、それと同期するが、ブラウザ－ウォレット自体の中に秘密鍵を格納しない。代わりに、ブラウザコンポーネントは、コンテンツについての支払いが行われることを促し、アプリケーションは、選択された外部ウォレットからデジタル署名による認可を要求する。この認可はユーザにより行われ、ブラウザ－ウォレットは、トランザクションをブロードキャストし、支払い済みのコンテンツを閲覧できる。
【０２０５】
＜Metanetトランザクションの読み出し及び書き込み＞
Metanetの固有の利点は、それが、支払い及びコンテンツデータを記録するために、同じデータ構造、つまりブロックチェーンを使用することである。これは、ソフトウェアウォレットが、暗号通貨の交換に純粋に基づくトランザクションを生成することに加えて、コンテンツデータをMetanetインフラストラクチャに書き込むために使用できることを意味する。
【０２０６】
アプリケーションに本来組み込まれたウォレットは、標準的な簡易支払検証（simplified payment verification (SPV)）クライアントよりも複雑なトランザクションをブロ

ックチェーンに書き込むことができる。https://bitcoin.org/en/glossary/simplified－payment－verificationを参照のこと。ウォレットは、ユーザが、彼らのコンピュータからブロックチェーンに埋め込まれるべきコンテンツデータを選択することにより、Metanetノードトランザクションをアプリケーションからブロックチェーンに直接書き込むことを選択できるようにする。

【０２０７】

ブラウザ－ウォレットアプリケーションがユーザインタフェース（user interface（UI））を有するので、ウォレットコンポーネントが、ブラウザコンポーネントの中で又は前もってユーザのコンピュータ上に構成されたコンテンツデータを含むトランザクションを生成しブロードキャストすることを可能にする。この能力は、専用のウォレットがそれ自体で処理することを達成するのは非常に困難である。

【０２０８】

＜アクセス鍵／トークンウォレット＞

以上の説明から、Metanetプロトコルに内蔵された、ECC鍵ペア又はAES対称鍵を用いてコンテンツを暗号化する能力、及び対応する解読鍵又はトークンを購入する能力を思い出してほしい。私たちは、これらをアクセス鍵又はアクセストークンと呼ぶ。

【０２０９】

このような鍵／トークンは、コンテンツを閲覧又は編集（一回限りの使用又は複数回の使用）する許可をユーザに与え、（望ましい場合には同じ鍵が両方の目的で使用されてよいが）ユーザの暗号通貨ウォレットを制御する鍵とは異なる役割を果たす。この理由のために、アプリケーションの本来備わっている暗号通貨ウォレットと別個に、アクセス鍵及びトークンを格納し及び管理するために使用される新しいウォレットを導入することは有利である。

【０２１０】

特定の時間期間の後に、アクセス鍵／トークンを消滅させることにより、Metanetコンテンツへの時間指定されたアクセスの概念を導入することも可能である。これは、アクセス鍵／トークンが信頼できる実行環境（Trusted Execution Environment（TEE））に格納され、ユーザにより直接アクセス可能でないことを要求することにより、達成できる。

【０２１１】

アクセス鍵／トークンが「消滅」し得るという事実は、暗号通貨秘密鍵が消滅するというリスクを負わないことを保証するために、アクセス鍵／トークンを暗号通貨ウォレットに格納しないことの動機付けの要因になり得る。

【０２１２】

暗号通貨ウォレットと同様の方法で、解読鍵及びアクセストークンは、効率的な処理及び展開を実現するために決定論的に格納され管理されることができる。解読鍵（例えば、ECC秘密鍵）は、マスタ鍵への後の追加により生成され復元できる。一方で、アクセストークンは、何らかの初期トークンによりシードされるハッシュチェーンを用いて再構成できる。

【０２１３】

ここで、暗号通貨ウォレットが、他のユーザと取引する際に及び新しいMetanetノードを生成する際に使用される鍵ペアの決定性鍵の生成を扱うこと、一方で、鍵／トークンウォレットが、暗号通貨ウォレットにより購入された鍵及びトークンを扱うこと、を区別することが重要である。

【０２１４】

＜ブロック高許可（Block height permissioning）＞

タイムロックは、ブロック高許可を可能にするためにBitcoinスクリプト言語に含まれ得る。oOp_code OP_CHECKLOCKTIMEVERIFY (CLTV)は、トランザクションアウトプット（UTXO）が使用のために許容されるブロック高を設定する。

【０２１５】

ブロック高許可の利点は以下の２つである。

１．バージョン制御（Version control）：Metanetプロトコルでは、最新バージョンのノードが、最大ブロック高にあるノードから識別できる。ブラウザ－ウォレットは、ブロック高によりファイルの最新バージョンを表示するよう設定できるだけでなく、proof－of－workバージョン制御も可能にする。
２．時間指定されたアクセス（Timed access）：ブラウザ－ウォレットアプリケーションは、ユーザにより購入された解読鍵を自動的に周期的に消滅できる。これは、ビューアが、彼らの支払った時間期間の間にのみ、コンテンツデータにアクセスできることを保証する。解読鍵のクローン化は、信頼できる実行環境（trusted execution environment (TEE)）にそれらを格納することにより防ぐことができる。更に、アトミックスワップは、（コンテンツデータの解読のための）決定性鍵Dkの購入を含む。この決定性鍵は公衆に見えるが、TEEは、Dkとセキュアにエンクレーブされた（enclaved）秘密鍵との結合に署名するために使用できる。

【０２１６】

ブラウザ－ウォレットは、任意の外部クロック又は第三者の時間のお告げ（time oracle）に頼るのではなく、自身の時間のプロキシとしてブロック高を使用するために、ブロックチェーンの現在状態と同期するよう構成できる。

【０２１７】

＜ウェブサーバのバイパス＞

本発明は、ブラウザ（クライアント）及びウェブサーバが分散型ピアインターネットを介して情報を通信し及び交換するための新しいメカニズムを可能にし、これは、ドメインネームシステム（domain name system (DNS)）サーバ及び標準的なネットワークルーティング手順をバイパスする。Http://www.theshulers.com/whitepapers/internet_whitepaper/を参照のこと。本発明は、ブロックチェーンのフルコピーを維持するピアを含む、ブラウザ－ウォレットアプリケーションがコンテンツを提供され得る新しいネットワークアーキテクチャを提供する。

【０２１８】

＜ローカルフルコピーピア＞

各々の地理的領域、例えば、郵便番号、町、都市、にあるローカルピアのシステムを考える。私たちは、このローカルネットワークの中で、少なくとも１つのピアが、ブロックチェーンのフルコピーを維持することを前提とする。私たちはこれをローカルフルコピーピア（Local Full－Copy Peer (LFCP)）と呼ぶ。私たちの目的のためには、LFCPは、Metanetフラグを含むブロックチェーントランザクションを格納するだけでよいが、それに限定されない。

【０２１９】

全部のユーザは、デフォルトで、LFCPへ「get」要求を送信する。ピアがブロックチェーン全体の完全な最新のコピーを維持するとき、要求される任意のノードIDがLFCPに利用可能なので、全ての要求がサービス可能である。SEがMetanetコンテンツを格納し及び標準的なSEの主要機能を実行するために十分に強力且つ大きい場合、Metanet検索エンジンがLFCPとして動作してもよいことに留意する。

【０２２０】

最も簡単な例では、全てのLFCPが、完全なブロックチェーンを格納可能である必要があるとき、同じ記憶及びディスク空間オーバヘッドを有する（出願時に約２００GB）。各LFCPの間の区別は、Metanetユーザからのローカルな要求に応じて、それらがそれらの能力をスケーリングすることである。従って、世界中の各Metanetユーザが彼らの最も近いLFCPをデフォルトで要求する場合、各LFCPは、彼らのローカルな要求を満たすために、彼らの運用能力をスケーリングしようとするべきである。都市のような人口密集地域は、多くのクラスタ化されたサーバを含むLFCP動作を要求する。一方で、より小さな町のような疎な地域は、より小さなLFCP動作を要求する。

【０２２１】

ディスク空間要件は汎用的であるが、LFCP毎のCPU要件はローカルネットワークの要

求に適応することに留意することが重要である。これは、Freenetのような適応型ネットワークの例である。https://blockstack.org/papers/を参照のこと。

【０２２２】

このようなシステムの１つの利点は、所与のID_{node}に関連付けられたコンテンツを読み出すとき、ユーザが、彼らのLFCPと単一の（ローカルな）接続を生成するだけでよいことである。他のピアはそれら自身の要求されたコンテンツを提供されることを保証されるので、LFCPが要求を他のピアへ送信する必要はない。

【０２２３】

Metanetは、非集中化及び非冗長性のような、インターネットに勝る多くの利点を提供する。これらは、IFPSのような他のピアツーピア（Ｐ２Ｐ）ファイル共有サービスと類似している。しかしながら、Metanetは、不変性及び重大なことにネットワークを所与のコンテンツに対する要求で溢れさせる必要を除去することにより、これらの既存のＰ２Ｐモデルを改良する。

【０２２４】

Metanetインフラストラクチャは、これらのピアのネットワークを利用することにより、任意の１つのLFCPの障害に対してもロバストである。これは、LFCPが無効にされた場合に、エンドユーザが単にデフォルトで彼らの次に近いLFCPを用いることを意味する。これは、任意の所与の時間に要求の観点で能力を下回る又は上回る近くのピアを示すために、LFCPが互いに通信する場合、より効率的にできる。これは、ユーザが彼らの要求を最も適切なピアに送信し、近くのLFCPの間で要求の動的な平衡を確立することを可能にする。

【０２２５】

＜グローバルフルコピーピア＞

私たちは、ここで、より小さなピアにとってユニバーサルディスク空間要件が大きくなり過ぎるシナリオを考える。これは、ブロックチェーンのMetanet部分が採択（adoption）によりスケーリングし増大するために生じ得る。

【０２２６】

この場合、より小さなLFCPは、人気システムに基づき、Metanetノードトランザクションを格納するために、自身のディスク空間能力を使用するべきである（要求の量及び特性によりコンテンツをランク付けする既存の技術がある）。これは、LFCPが、（要求処理能力のために）それらのCPU及び（コンテンツ提供能力のために）それらの記憶割り当ての両方を、コンテンツの量及び特性の両方においてそれらのローカルな地理的要求に適するよう調整することを意味する。

【０２２７】

LFCPが全てのMetanetトランザクションコンテンツを格納できないという事実を解決するために、グローバルフルコピーピア（Global Full－Copy Peer (GFCP)）の概念が利用できる。GFCPは、以下の特性を有するフルコピーピアである。

１．GFCPは、常にブロックチェーンのフルコピーを維持するために、自身のディスク空間能力を増大させる。

２．GFCPは、相当なCPUリソースを有するので、LFCPよりも想到に多くの要求を処理できる。グローバルフルコピーピアは、多数のLFCPが障害になった場合に、要求の急激な増加を処理できなければならない。

【０２２８】

GFCPの２つの主な機能がある。第１に、LFCPからオーバーフローリクエストの際に、Metanetコンテンツのユーザ要求のためのフェールセーフとして機能することである。第２に、GFCPは、過去にマイニングされた全部のMetanetコンテンツを格納するためのアーカイブピアとして動作する。これは、多数のLFCPがそれらの記憶準備（provision）から幾つかのコンテンツを省略した場合でも、任意のMetanetノードコンテンツがアクセスできることを保証する。

【０２２９】

＜グローバルデータバンク＞

GFCPの概念は、強力なものであり、Metanetの全体的なアーキテクチャが、包括的なグローバルデータバンクを生成するという既存の問題に対するソリューションをどのように提供するかを説明する。

【０２３０】

従来は、ユニバーサルなグローバルにアクセス可能なデータバンクは、中央当局により維持される必要があるので、安全に構成することができなかった。この中央当局は、障害点及び信頼点の両方をシステムに注入する。重要なことに、ある組織が全部のインターネットデータを格納し維持することに依存している場合、私たちは、それらが情報を壊すことなく、正しく合法的にそうしていることを信頼する必要がある。

【０２３１】

Metanetインフラストラクチャにより、私たちは、これらの問題の両方、信頼の問題及び中央化の問題を、グローバルデータセンタの概念から効果的に除去した。ここで、このようなGFCPは、記憶のために必要なディスク空間を提供することに依存し、格納されるべき情報を検証し及び認証することに依存しないので、このようなGFCPが生成できる。

【０２３２】

Metanetにより、何が格納されるかの検証処理は、マイナーにより行われる。従って、ユニバーサルなグローバルデータバンクは、それらがブロックチェーン情報を壊さないので、信頼できる。GFCPは、信頼される必要はなく、記憶を提供するだけでよい。

【０２３３】

全部のGFCPが同じ情報を格納できること、ブロックチェーン自体に対して常に検証可能であり証明可能であることは、多数のこのようなGFCPに渡り再現できることを意味する。

【０２３４】

これは、多数のグローバルデータバンクを並列に存在させ、証明可能に同じ情報を格納させることにより、単一の障害点を有するという問題を解決することも意味する。

【０２３５】

図１７は、２個のLFCP及び１個のGFCPのシステムを示し、個々のピアの障害に対してロバストであるネットワーク内で各ピアが他のピアをどのようにサポートできるかを説明する。

【０２３６】

上述のブラウザーウォレットアプリケーションの実施形態において実施され得る本発明の態様は、以下を含むがそれに限定されない、従来技術に勝る多数の顕著な特徴及び利点を提供する。

１．決定性鍵：暗号通貨及びMetanetアドレスの両方のための階層的決定性鍵管理は、アプリケーションの同じウォレットコンポーネントの中で実行される。これは、複数の機能を有する鍵の編成が可能になり、それらの記憶要件を軽減し、鍵復元を可能にする。

２．支払いメカニズム：アプリケーションは、消費者が、従来認証し及び信頼を提供する別のアプリケーション又は第三者支払いサービスを指す必要なしに、商人に直接支払うことを可能にする。これは、同じブロックチェーンプラットフォームを介して、デジタルコンテンツの購入及び配信が行われることを可能にする。アプリケーションは、小さな価値の交換を含むBitcoin支払い、又は複数のパーティを含むより複雑なトランザクションの利点を本来備える。

３．ウェブサーバのバイパス：アプリケーションは、従来高容量トラフィック、要求、及びルーティングを処理する伝統的なウェブサーバのバイパスを実現する。これは、アプリケーションが、他のLFCPへ要求を転送する必要がなくユーザにサービスすることを保証されている単一のLFCPからコンテンツを要求するだけでよいからである。これは、全体のトラフィック量、及び各要求の完了時間を削減する。

４．時間指定されたアクセス：アプリケーションは、ブロックチェーンと同期すること及びブロックチェーンを用いてブロックチェーンの現在状態に基づきアクセス許可を実施することにより、コンテンツへの時間指定されたアクセスを実現する。これは、第三者サー

ビスが、元の所有者の権利を保護しながら、時間に渡りユーザの権利を監視する必要を除去する。

【０２３７】

使用例：非集中化アプリストア（Swapp store、Swappストア）

ここに提示されるMetanetアーキテクチャについての（説明のみを目的とする）第１の使用例は、アプリケーション（アプリ、app）の非集中化支払い及び配信に関する。

【０２３８】

私たちは、アプリ開発者Alice及び消費者Bobが、互いに取引したいと望むシナリオを考える。このトランザクションは、アトミックスワップの形式を取り、その中で、Bobがアプリケーションデータにアクセスすることを許可するシークレット鍵のために、金銭が交換される。暗号化アプリケーションデータは、既に、Metanetノードトランザクションの部分として公開されている。

【０２３９】

アトミックスワップアプリケーションは、Swapp（スワップ）として知られている。第三者プラットフォーム（Swappストア）は、Metanetに存在するアプリケーションを分類し広告するために使用されてよい。しかしながら、アクセス鍵のための支払い及びBobのようなユーザへの転送は、第三者を含む必要はなく、商人と消費者との間で直接行うことができる。

【０２４０】

以下の章は、Aliceによるアプリの生成から、Bobによるその展開まで、Swappを買い及び売るために使用され得る処理を詳述する。処理を通じて、Alice及びBobは、彼らのそれぞれのブラウザ－ウォレットを使用して、Metanetと相互作用する。

【０２４１】

＜発行（Publishing）＞

１．Aliceは、アプリケーションを記述する。このアプリケーションを構成するデータは、〈App〉により示されるコンテンツである。彼女は、シークレット鍵S_kを用いて、それを暗号化する〈e(App)〉。

２．Aliceは、ノードトランザクションID_{Alice_App}を生成し、彼女の第１Metanetドメイン（木）を設定する。彼女は、ノードアドレスとして使用されるべき１AliceAppHtKNngkdXEeobR７６ｂ５３LETtpy（$P_{AliceApp}$)を生成する。

３．Aliceは、次に、第１ノードの子を生成して、彼女のアプリケーションのMetanetライブラリに対応する木を形成する。Aliceの木ドメインは、図１８に示される。

この木にあるリーフノードのうちの１つは、インデックスID_{App}を有する彼女のアプリケーション〈App〉に対応するノードである。このノードでは、Aliceは、暗号化アプリケーションデータ〈e(App)〉をノードのインプットスクリプト（scriptSig）に挿入する。アプリデータは、シークレット鍵s_kを用いてKoblitz法を用いて暗号化される。

このノードトランザクションは、以下に示される。

［表３］

【表３】

$TxID_{App}$	
Input	Output
$< Sig\ P_{Puzzle} >$ $< P_{Puzzle} >$ <e(App)> OP_DROP	OP_RETURN <Metanet Flag> $< P_{App} >$ $<TxID_{Puzzle}>$

４．Aliceは、$ID_{AliceApp}$、$P_{AliceApp}$、及びドメイン名「AliceApp」を公にブロードキ

ャストする。これは、ソーシャルメディア、インターネットウェブサイトを介して、又は第三者Metanetウェブサイトを使用して、可能である。

【０２４２】

＜購入＞

１．Bobは、彼が彼のブラウザ－ウォレット上で閲覧するMetanetウェブサイト（Swappストア）にリストされたパズルゲームをダウンロードし、Aliceのアプリを見たいと望む。

２．Bobは、次に、ウェブサイトからの情報を用いてAliceと連絡を取り、アトミックスワップを設定する。スワップは、BobがBitcoinの合意した価格をAliceに支払い、Aliceがシークレット鍵s_kを開示するよう、又は何もイベントが生じないよう、設計される。

３．アトミックスワップが完了し、Bobのブラウザ－ウォレットは、シークレット鍵s_kを自身のアクセス鍵／トークンウォレットに格納する。

【０２４３】

＜展開＞

Bobは、今や、前もってAliceが発行したアプリケーションデータを彼が解読することを可能にする鍵s_kを有する。アプリをダウンロードし、それを展開するために、Bobは、以下を行う。

１．Bobは、Metanet検索エンジン（SE）を用いて、暗号化されたアプリデータ〈e(App)〉に関連付けられたMURLを見付ける。彼は、キーワード「AliceApp」及び「App」を、彼のブラウザ－ウォレット内の検索バーへの入力として使用する。第三者SEは、クエリを解決し、以下のMURLを返す。

mnp://aliceapp/games/puzzle/app

このロケータは、自身のインプットスクリプトの中に暗号化アプリデータを含むユニークなMetanetノードID_{App}に対応する。

２．Bobのブラウザ－ウォレットは、このMURLを受信し、最も近い適切なLFCPへ要求を送信する。このピアは、Bobに要求されたデータ〈e(App)〉を提供する。

３．ブラウザ－ウォレットは、ID_{App}の属性に従いデータを処理する。これは、シークレット鍵s_kを用いてアプリケーションデータを解読し〈App〉を処理することを含む。

４．Bobは、アプリケーション〈App〉を彼のブラウザから彼のコンピュータへダウンロードする。Bobは、今や、アクセスを予め購入することなく、アプリケーションをローカルに展開できる。

【０２４４】

図１９は、上述の説明のための使用例において概説された全体処理を説明する。フローチャートは、２つのアクションブランチ、つまりAliceのもの（左側から開始する）とBobのもの（右側から開始する）を示す。Aliceに対応するブランチは初期発行フェーズを示し、Bobのものはアトミックスワップを介して購入を設定するフェーズを示す。

【０２４５】

Bobのブランチで、アトミックスワップ設定フェーズとして、彼は、以下のトランザクション$TxID_{Bob}$をブロードキャストする。

［表４］

【表４】

$TxID_{Bob}$			
Inputs		Outputs	
Value	Script	Value	Script
x BCH	$<Sig\ P_B>$ $<P_B>$	x BCH	[Private Key Puzzle P_k, r_0] [CheckSig P_A]

【０２４６】

このトランザクションでは、アウトプットは、Aliceが使用（spend）するためにBobに開示されるべきシークレット解読鍵s_kを必要とする秘密鍵パズルによりロックされる。

【０２４７】
この図の中のAlice及びBobのブランチは、Aliceがアトミックスワップトランザクションの完了に成功する時点で収束する。これは、Aliceがトランザクション$TxID_{Alice}$をブロードキャストするとき、達成される。
［表５］
【表５】

$TxID_{Alice}$			
Inputs		Outputs	
Value	Script	Value	Script
x Bitcoin	$< Sig\ P_A >$ $< P_A >$ $< Sig\ P_1, r_0 >$ $< P_1 >$	x Bitcoin	[CheckSig P_A]

【０２４８】
このトランザクションがブロードキャストすると直ぐに、Alice及びBobのアクションは再び分岐する。Aliceはx Bitcoinの支払いを受け、一方で、Bobはシークレット解読鍵s_kを受信し、MetanetからAliceのアプリケーションを読み出し解読することができる。
【０２４９】
図２０を参照すると、本開示の少なくとも一実施形態を実施するために使用され得るコンピューティング装置２６００の説明のための簡略ブロック図が提供される。種々の実施形態で、コンピューティング装置２６００は、上述の図示のシステムのうちのいずれかを実装するために使用されてよい。例えば、コンピューティング装置２６００は、データサーバ、ウェブサーバ、ポータブルコンピューティング装置、パーソナルコンピュータ、又は任意の電子コンピューティング装置として使用するために構成されてよい。図２０に示すように、コンピューティング装置２６００は、主メモリ２６０８及び永久記憶装置２６１０を含む記憶サブシステム２６０６と通信するよう構成され得る１つ以上のレベルのキャッシュメモリ及びメモリ制御部（集合的に２６０２とラベル付けされる）を備える１つ以上のプロセッサを含んでよい。主メモリ２６０８は、図示のように、動的ランダムアクセスメモリ（ＤＲＡＭ）２６１８及び読み出し専用メモリ（ＲＯＭ）２６２０を含み得る。記憶サブシステム２６０６及びキャッシュメモリ２６０２は、本開示で説明されたようなトランザクション及びブロックに関連付けられた詳細事項のような情報の記憶のために使用されてよい。プロセッサ２６０２は、本開示で説明されたような任意の実施形態のステップ又は機能を提供するために利用されてよい。
【０２５０】
プロセッサ２６０２は、１つ以上のユーザインタフェース入力装置２６１２、１つ以上のユーザインタフェース出力装置２６１４、及びネットワークインタフェースサブシステム２６１６とも通信できる。
【０２５１】
バスサブシステム２６０４は、コンピューティング装置２６００の種々のコンポーネント及びサブシステムが意図した通りに互いに通信できるようにするメカニズムを提供してよい。バスサブシステム２６０４は、単一のバスとして概略的に示されるが、バスサブシステムの代替の実施形態は、複数のバスを利用してよい。
【０２５２】
ネットワークインタフェースサブシステム２６１６は、他のコンピューティング装置及びネットワークへのインタフェースを提供してよい。ネットワークインタフェースサブシステム２６１６は、幾つかの実施形態では、コンピューティング装置２６００の他のシステムからデータを受信し及びそれへデータを送信するインタフェースとして機能してよい。例えば、ネットワークインタフェースサブシステム２６１６は、データ技術者が、装置をネットワークに接続することを可能にする。その結果、データ技術者は、データセンタ

のような遠隔地にいがなら、データを装置へ送信し、データを装置から受信できる。
【０２５３】
ユーザインタフェース入力装置２６１２は、キーボード、統合型マウス、トラックボール、タッチパッド、又はグラフィックタブレットのような指示装置、スキャナ、バーコードスキャナ、ディスプレイに組み込まれたタッチスクリーン、音声認識システム、マイクロフォンのようなオーディオ入力装置、及び他の種類の入力装置のような、１つ以上のユーザ入力装置を含んでよい。通常、用語「入力装置」の使用は、コンピューティング装置２６００に情報を入力する全ての可能な種類の装置及びメカニズムを含むことを意図する。
【０２５４】
１つ以上のユーザインタフェース出力装置２６１４は、ディスプレイサブシステム、プリンタ、又はオーディオ出力装置のような非視覚的ディスプレイ、等を含んでよい。ディスプレイサブシステムは、陰極線管（ＣＲＴ）、液晶ディスプレイ（ＬＣＤ）、発光ダイオード（ＬＥＤ）ディスプレイ、又はプロジェクションのような平面装置、又は他のディスプレイ装置を含んでよい。通常、用語「出力装置」の使用は、コンピューティング装置２６００から情報を出力する全ての可能な種類の装置及びメカニズムを含むことを意図する。１つ以上のユーザインタフェース出力装置２６１４は、例えば、ユーザインタフェースを提示して、ここに記載したプロセス及び変形を実行するアプリケーションとのユーザ相互作用が適切であるとき、そのような相互作用を実現するために使用されてよい。
【０２５５】
記憶サブシステム２６０６は、本開示の少なくとも１つの実施形態の機能を提供する基本プログラミング及びデータ構造を記憶するコンピュータ可読記憶媒体を提供してよい。アプリケーション（例えば、プログラム、コードモジュール、命令）は、１つ以上のプロセッサにより実行されると、本開示の１つ以上の実施形態の機能を提供し、記憶サブシステム２６０６に格納されてよい。これらのアプリケーションモジュール又は命令は、１つ以上のプロセッサ２６０２により実行されてよい。記憶サブシステム２６０６は、更に、本開示に従い使用されるデータを格納するレポジトリを提供する。例えば、主メモリ２６０８及びキャッシュメモリ２６０２は、プログラム及びデータのための揮発性記憶を提供できる。永久記憶装置２６１０は、プログラム及びデータの永久（不揮発性）記憶を提供でき、磁気ハードディスクドライブ、取り外し可能媒体に関連付けられた１つ以上のフロッピディスクドライブ、取り外し可能媒体に関連付けられた１つ以上の光ドライブ（例えば、ＣＤ－ＲＯＭ、又はＤＶＤ、又はＢｌｕｅ－Ｒａｙ）ドライブ、及び他の同様の記憶媒体を含んでよい。このようなプログラム及びデータは、本開示に記載した１つ以上の実施形態のステップを実行するためのプログラム、及び本開示に記載したトランザクション及びブロックに関連付けられたデータを含み得る。
【０２５６】
コンピューティング装置２６００は、ポータブルコンピュータ装置、タブレットコンピュータ、ワークステーション、又は後述する任意の他の装置を含む種々のタイプのものであってよい。さらに、コンピューティング装置２６００は、１つ以上のポート（例えば、USB、ヘッドフォンジャック、光コネクタ、等）を通じてコンピューティング装置２６００に接続可能な別の装置を含み得る。コンピューティング装置２６００に接続され得る装置は、光ファイバコネクタを受けるよう構成される複数のポートを含んでよい。従って、この装置は、光信号を、処理のために装置を接続するポートを通じてコンピューティング装置２６００に送信される電気信号に変換するよう構成されてよい。コンピュータ及びネットワークの絶えず変化する特性により、図２０に示したコンピューティング装置２６００の説明は、装置の好適な実施形態を説明する目的の特定の例としてのみ意図される。図２０に示したシステムより多くの又は少ないコンポーネントを有する多くの他の構成が可能である。
【０２５７】
上述の実施形態は、本発明を限定するのではなく、説明すること、及び当業者は添付の特許請求の範囲により定められる本発明の範囲から逸脱することなく多くの代替的実施形

態を考案できることに留意すべきである。特許請求の範囲において、括弧内の任意の参照符号は、請求項を限定することを意図しない。用語「有する」及び「含む」（comprising、comprises）等は、任意の請求項又は明細書全体に列挙されたもの以外の要素又はステップの存在を排除しない。本願明細書では、「有する」は「有する又は構成される」を意味し、「含む」は「含む又は構成される」を意味する。要素の単数の参照は、該要素の複数の参照を排除しない。逆も同様である。本発明は、幾つかの別個の要素を含むハードウェアにより、及び適切にプログラムされたコンピュータにより、実装できる。幾つかの手段を列挙する装置クレームでは、これらの手段のうちの幾つかは、１つの同じハードウェアアイテムにより具現化されてよい。単に特定の手段が相互に異なる従属請求項に記載されるという事実は、これらの手段の組み合わせが有利に使用されないことを示さない。

【図面】

【図１】

$TxID$	
Inputs	Outputs
< $Sig\ P$> <P>	OP_RETURN <Metanet Flag> <Attribute 1> <Attribute 2>
	<Content 1> OP_DROP <$H(P')$> [CheckSig P']

【図２】

$TxID$	
Inputs	Outputs
< $Sig\ P$> <P> <Content 1> OP_DROP	OP_RETURN <Metanet Flag> <Attribute 1> <Attribute 2>

【図３】

$TxID_1$	
Inputs	Outputs
< $Sig\ P$> <P>	OP_RETURN <Metanet Flag> <Attribute 1,1 = Content name> <Attribute 1,2 = Recombination scheme> <Attribute 1,3 = Content chunk index 1>
	<Content chunk 1> OP_DROP <$H(P')$> [CheckSig P']

$TxID_2$	
Inputs	Outputs
< $Sig\ P$> <P>	OP_RETURN <Metanet Flag> <Attribute 2,1 = Content name> <Attribute 2,2 = Recombination scheme> <Attribute 2,3 = Content chunk index 2>
	<Content chunk 2> OP_DROP <$H(P')$> [CheckSig P']

【図４】

$TxID_{Bob}$			
Inputs		Outputs	
Value	Script	Value	Script
x	< $Sig\ P_B$> <P_B>	x	[Private Key Puzzle P_1, r_0] [CheckSig P_A]

【図５】

$TxID_{Alice}$			
Inputs		Outputs	
Value	Script	Value	Script
x	$< Sig\ P_A >\ <P_A>\ < Sig\ P_1, r_0 >\ <P_1>$	x	[CheckSig P_A]

【図６】

Use		Alice	Bob
Token Issuance	Purchase of 10 tokens	$I_{Alice} = k$	$I_{Bob} = H^{10}(Y)$
Token redemption	Redeem token 1	X_1	$T_1 = H^9(Y)$
	Redeem token 2	X_2	$T_2 = H^8(Y)$
	...	...	...
	Redeem token 10	X_{10}	$T_{10} = Y$

【図７】

$TxID_{Alice}$			
Inputs		Outputs	
Value	Script	Value	Script
x	$< Sig\ P_A >\ <P_A>$	x	[Hash Puzzle $H(I_{Alice})$] [Hash Puzzle $H(I_{Bob})$] [CheckSig P_B]

Phase 1.1 (Alice to Bob):

【図８】

$TxID_{Bob}$			
Inputs		Outputs	
Value	Script	Value	Script
$10 + x$	$< Sig\ P_B >\ <P_B>$	$10 + x$	[Hash Puzzle $H(I_{Alice})$] [Hash Puzzle $H(I_{Bob})$] [CheckSig P_A]

Phase 1.2 (Bob to Alice):

【図９】

Phase 2.1 (Bob to Alice):

$TxID_{Bob}$			
Inputs		Outputs	
Value	Script	Value	Script
x	$< Sig\ P_B >\ <P_B>$	x	[Hash Puzzle $H(X_1)$] [Hash Puzzle $H(T_1)$] [CheckSig P_A]

【図１０】

Phase 2.2 (Alice to Bob):

$TxID_{Alice}$			
Inputs		Outputs	
Value	Script	Value	Script
x	$< Sig\ P_A >\ <P_A>$	x	[Hash Puzzle $H(X_1)$] [Hash Puzzle $H(T_1)$] [CheckSig P_B]

【図１１】

Phase 2.3 (Alice to Alice):

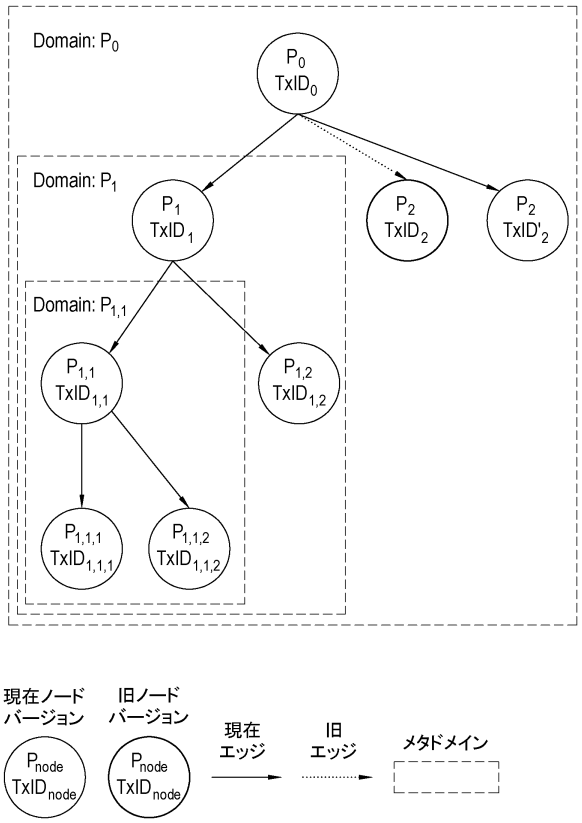
$TxID_{Alice}$			
Inputs		Outputs	
Value	Script	Value	Script
x	$< Sig\ P_A >\ <P_A>\ <T_1>\ <X_1>$	x	[CheckSig P_A]

【図１２】

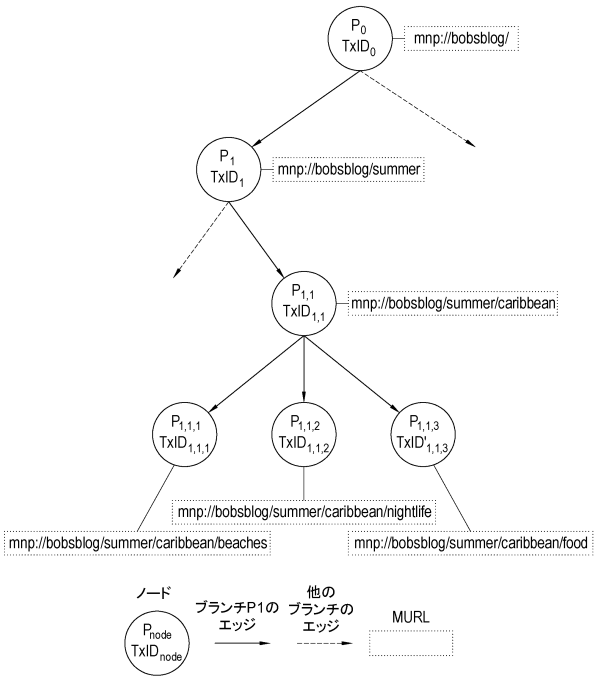
Phase 2.4 (Bob to Bob):

$TxID_{Bob}$			
Inputs		Outputs	
Value	Script	Value	Script
x	$< Sig\ P_B >\ <P_B>\ <T_1>\ <X_1>$	x	[CheckSig P_B]

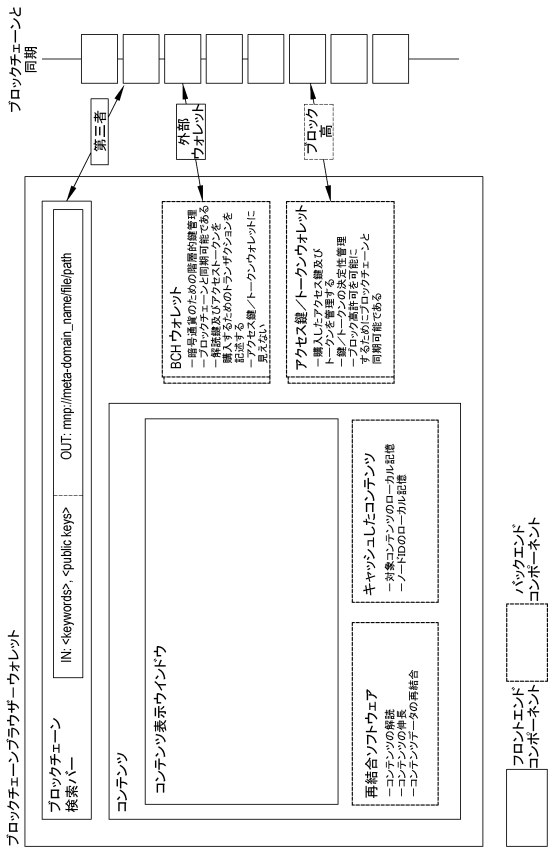
【図１３】



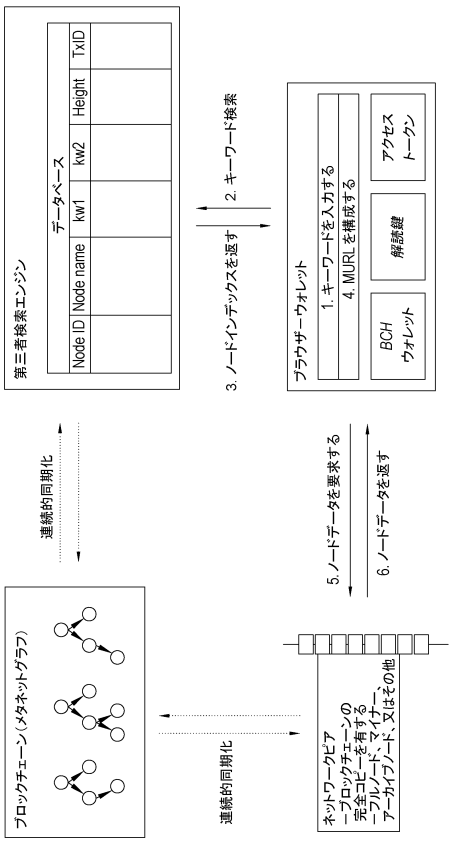
【図１４】



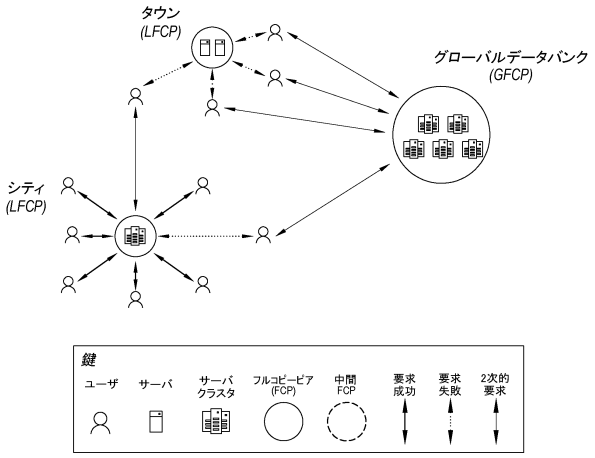
【図１５】



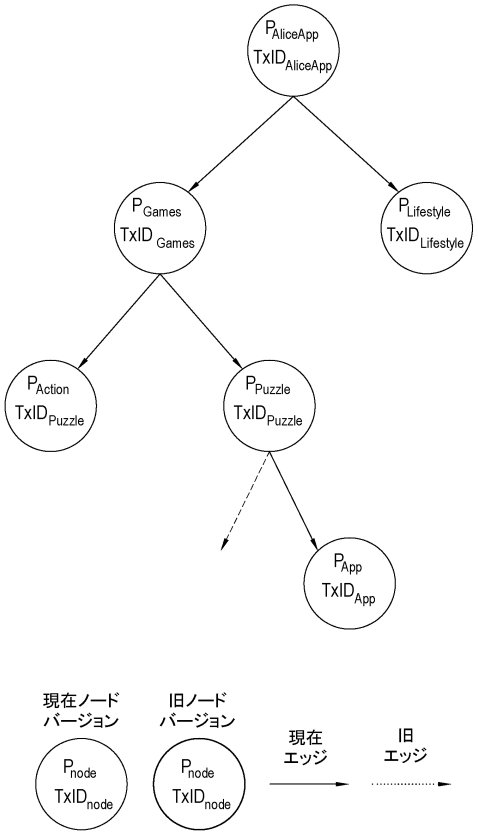
【図１６】



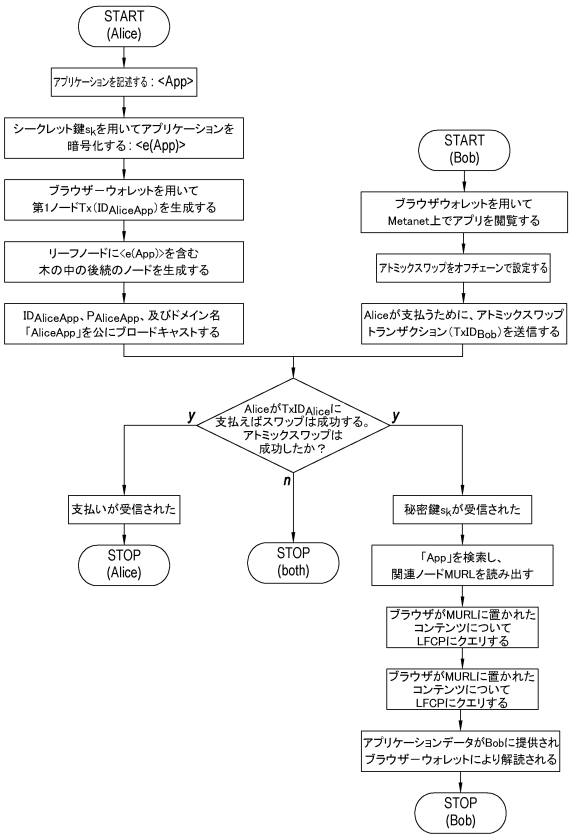
【図１７】



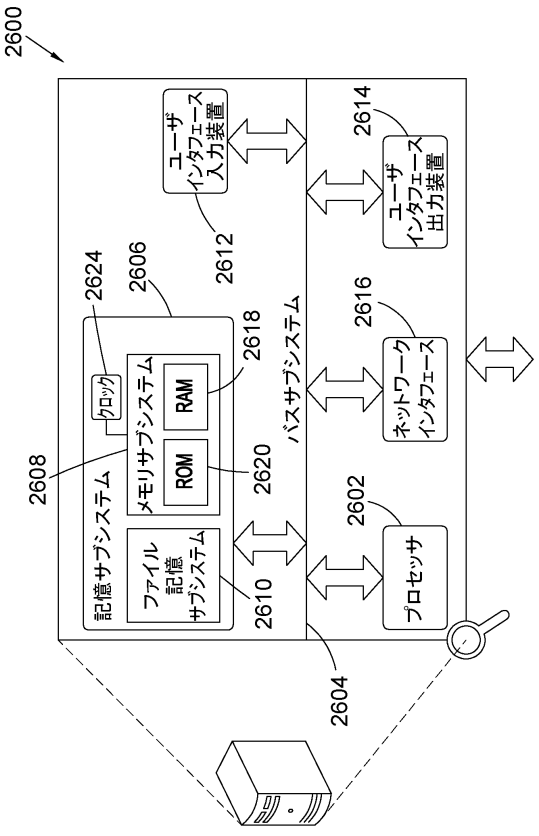
【図１８】



【図１９】



【図２０】



フロントページの続き

(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　1819284.9
(32)優先日　平成30年11月27日(2018.11.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　1819297.1
(32)優先日　平成30年11月27日(2018.11.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　1819291.4
(32)優先日　平成30年11月27日(2018.11.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　1819293.0
(32)優先日　平成30年11月27日(2018.11.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　1819299.7
(32)優先日　平成30年11月27日(2018.11.27)
(33)優先権主張国・地域又は機関
英国(GB)

内
(72)発明者　デイヴィース，ジャック　オーウェン
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　内
(72)発明者　タータン，クロイー　セレン
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　内
(72)発明者　ヴォーン，オーウェン
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　内

審査官　金沢　史明
(56)参考文献　米国特許第０９２９８８０６（ＵＳ，Ｂ１）
米国特許出願公開第２０１５／０２６９５７０（ＵＳ，Ａ１）
韓国登録特許第１０－１８３７１６８（ＫＲ，Ｂ１）
欧州特許出願公開第０３３６４３５１（ＥＰ，Ａ１）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　９／３２