(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7545489号
(P7545489)

(45)発行日　令和6年9月4日(2024.9.4)

(24)登録日　令和6年8月27日(2024.8.27)

(51)国際特許分類

Ｇ０６Ｆ　21/62　(2013.01)		Ｇ０６Ｆ	21/62
Ｇ０６Ｆ　21/64　(2013.01)		Ｇ０６Ｆ	21/64

ＦＩ

請求項の数　15　（全55頁）

(21)出願番号　特願2022-558420(P2022-558420)
(86)(22)出願日　令和3年3月23日(2021.3.23)
(65)公表番号　特表2023-520372(P2023-520372A)
(43)公表日　令和5年5月17日(2023.5.17)
(86)国際出願番号　PCT/US2021/023633
(87)国際公開番号　WO2021/195052
(87)国際公開日　令和3年9月30日(2021.9.30)
審査請求日　令和6年1月12日(2024.1.12)
(31)優先権主張番号　16/828,003
(32)優先日　令和2年3月24日(2020.3.24)
(33)優先権主張国・地域又は機関
米国(US)
早期審査対象出願

(73)特許権者　522377837
スパイダーオーク， インク．
ＳＰＩＤＥＲＯＡＫ， ＩＮＣ．
アメリカ合衆国 ６６２１４ カンザス州 ，レネックサ マーシャル ドライブ ８２１６
(74)代理人　100107364
弁理士　斉藤 達也
(72)発明者　ムーア，ジョナサン アンドリュウ クロケット
アメリカ合衆国 ９７２０２ オレゴン州 ポートランド， エスイー チベッツ ストリート ２９４７
審査官　平井 誠

最終頁に続く

(54)【発明の名称】　企業環境におけるブロックチェーンの統合、グループ権限とアクセスの管理

(57)【特許請求の範囲】

【請求項１】

コンピュータ実装方法であって、

複数のブロックを含むブロックチェーンを作成するステップを、

ユーザの権限を定義するブロックを作成するステップであって、前記ブロックは、ユーザを識別する暗号ユーザＩＤと、前記暗号ユーザＩＤに関連付けられた権限とを含み、前記権限は、前記ブロックチェーンで実行する為に前記暗号ユーザＩＤに関連付けられた少なくとも１つの操作を定義している、ステップと、

前記ブロックを前記ブロックチェーンの末尾に追加するステップであって、前記ブロックチェーンは暗号化されていない、ステップと、によって行うステップと、

前記ブロックチェーンにアクセスする要求を要求元デバイスから受信するステップであって、前記要求は、前記要求を行う前記ユーザに関連付けられた暗号ユーザＩＤを含む、ステップと、

前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする権限を有するかどうかを判断するステップを、

前記ブロックチェーンに記録された前記権限を計算することなく、前記ブロックチェーンを初期ブロックから最終ブロックまで確認することを含めて、前記要求がリカバリー鍵で署名されているかどうかを判断することによって行うステップと、

前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする前記権限を有すると判断した場合に、前記要求を行う前記ユーザに前記ブロックチェーンへのアクセスを付与

するトークンを生成するステップと、
前記トークンを前記要求元デバイスに送信するステップと、
を含む方法。
【請求項２】
前記トークンを生成するステップは、
秘密のルート鍵を識別する鍵識別子を作成するステップと、
前記トークンによって付与される前記ブロックチェーンへのパーミッションを作成するステップと、
前記秘密のルート鍵と前記パーミッションとの暗号化ハッシュを作成するステップと、
前記鍵識別子、前記パーミッション、及び前記暗号化ハッシュを前記トークンに追加するステップと、を含む、請求項１に記載の方法。
【請求項３】
前記リカバリー鍵を複数の部分に分離するステップと、
前記複数の部分の内の少なくとも一部のサブセットを暗号化するステップと、
暗号化された前記一部のサブセットと前記複数の部分の残りを複数のデバイスに配布するステップと、を含む請求項１に記載の方法。
【請求項４】
第１の暗号鍵をサーバに格納するステップと、
第２の暗号鍵をユーザデバイスに送信するステップと、
前記ユーザデバイスから前記第１の暗号鍵の要求を受信するステップと、
前記要求を受信した時に、前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されているかどうかを判断するステップと、
前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されていないと判断した場合、前記第１の暗号鍵の送信を拒否するステップと、を含む請求項１に記載の方法。
【請求項５】
前記ブロックチェーンのセマンティクスの解釈に関する更新を受信するステップと、
前記ブロックチェーン内に前記更新を格納することにより、複数のユーザデバイスに亘る前記ブロックチェーンの前記セマンティクスの解釈の一貫性を確保するステップと、を含む請求項１に記載の方法。
【請求項６】
コンピュータシステムであって、
１つ以上のコンピュータプロセッサと、
コンピュータ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、前記コンピュータ命令は、前記１つ以上のコンピュータプロセッサによって実行されたときに、前記コンピュータシステムに、
複数のブロックを含むブロックチェーンを作成するステップを、
ユーザの権限を定義するブロックを作成するステップであって、前記ブロックは、ユーザを識別する暗号ユーザＩＤと、前記暗号ユーザＩＤに関連付けられた権限とを含み、前記権限は、前記ブロックチェーンで実行する為に前記暗号ユーザＩＤに関連付けられた少なくとも１つの操作を定義している、ステップと、
前記ブロックを前記ブロックチェーンの末尾に追加するステップであって、前記ブロックチェーンは暗号化されていない、ステップと、によって行うステップと、
前記ブロックチェーンにアクセスする要求を要求元デバイスから受信するステップであって、前記要求は、前記要求を行う前記ユーザに関連付けられた暗号ユーザＩＤを含む、ステップと、
前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする権限を有するかどうかを判断するステップを、
前記ブロックチェーンに記録された前記権限を計算することなく、前記ブロックチェーンを初期ブロックから最終ブロックまで確認することを含めて、前記要求がリカバリー鍵で署名されているかどうかを判断することによって行うステップと、

前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする前記権限を有すると判断した場合に、前記要求を行う前記ユーザに前記ブロックチェーンへのアクセスを付与するトークンを生成するステップと、
前記トークンを前記要求元デバイスに送信するステップと、を行わせる、
前記非一時的コンピュータ可読記憶媒体と、
を備えるコンピュータシステム。
【請求項７】
前記トークンを生成するステップを行わせる前記コンピュータ命令が、前記コンピュータシステムに、
秘密のルート鍵を識別する鍵識別子を作成するステップと、
前記トークンによって付与される前記ブロックチェーンへのパーミッションを作成するステップと、
前記秘密のルート鍵と前記パーミッションとの暗号化ハッシュを作成するステップと、
前記鍵識別子、前記パーミッション、及び前記暗号化ハッシュを前記トークンに追加するステップと、を行わせる請求項６に記載のコンピュータシステム。
【請求項８】
前記コンピュータ命令が、前記コンピュータシステムに、
前記リカバリー鍵を複数の部分に分離するステップと、
前記複数の部分の内の少なくとも一部のサブセットを暗号化するステップと、
暗号化された前記一部のサブセットと前記複数の部分の残りを複数のデバイスに配布するステップと、を行わせる請求項６に記載のコンピュータシステム。
【請求項９】
前記コンピュータ命令が、前記コンピュータシステムに、
第１の暗号鍵をサーバに格納するステップと、
第２の暗号鍵をユーザデバイスに送信するステップと、
前記ユーザデバイスから前記第１の暗号鍵の要求を受信するステップと、
前記要求を受信した時に、前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されているかどうかを判断するステップと、
前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されていないと判断した場合、前記第１の暗号鍵の送信を拒否するステップと、を行わせる請求項６に記載のコンピュータシステム。
【請求項１０】
前記コンピュータ命令が、前記コンピュータシステムに、
前記ブロックチェーンのセマンティクスの解釈に関する更新を受信するステップと、
前記ブロックチェーン内に前記更新を格納することにより、複数のユーザデバイスに亘る前記ブロックチェーンの前記セマンティクスの解釈の一貫性を確保するステップと、を行わせる請求項６に記載のコンピュータシステム。
【請求項１１】
コンピュータ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、前記コンピュータ命令は、コンピュータシステムの１つ以上のコンピュータプロセッサによって実行されたときに、前記コンピュータシステムに、
複数のブロックを含むブロックチェーンを作成するステップを、
ユーザの権限を定義するブロックを作成するステップであって、前記ブロックは、ユーザを識別する暗号ユーザＩＤと、前記暗号ユーザＩＤに関連付けられた権限とを含み、前記権限は、前記ブロックチェーンで実行する為に前記暗号ユーザＩＤに関連付けられた少なくとも１つの操作を定義している、ステップと、
前記ブロックを前記ブロックチェーンの末尾に追加するステップであって、前記ブロックチェーンは暗号化されていない、ステップと、によって行うステップと、
前記ブロックチェーンにアクセスする要求を要求元デバイスから受信するステップであって、前記要求は、前記要求を行う前記ユーザに関連付けられた暗号ユーザＩＤを含む、

ステップと、
前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする権限を有するかどうかを判断するステップを、
前記ブロックチェーンに記録された前記権限を計算することなく、前記ブロックチェーンを初期ブロックから最終ブロックまで確認することを含めて、前記要求がリカバリー鍵で署名されているかどうかを判断することによって行うステップと、
前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする前記権限を有すると判断した場合に、前記要求を行う前記ユーザに前記ブロックチェーンへのアクセスを付与するトークンを生成するステップと、
前記トークンを前記要求元デバイスに送信するステップと、を行わせる、
非一時的コンピュータ可読記憶媒体。
【請求項１２】
前記トークンを生成するステップを行わせる前記コンピュータ命令が、前記コンピュータシステムに、
秘密のルート鍵を識別する鍵識別子を作成するステップと、
前記トークンによって付与される前記ブロックチェーンへのパーミッションを作成するステップと、
前記秘密のルート鍵と前記パーミッションとの暗号化ハッシュを作成するステップと、
前記鍵識別子、前記パーミッション、及び前記暗号化ハッシュを前記トークンに追加するステップと、を行わせる請求項１１に記載の非一時的コンピュータ可読記憶媒体。
【請求項１３】
前記コンピュータ命令が、前記コンピュータシステムに、
前記リカバリー鍵を複数の部分に分離するステップと、
前記複数の部分の内の少なくとも一部のサブセットを暗号化するステップと、
暗号化された前記一部のサブセットと前記複数の部分の残りを複数のデバイスに配布するステップと、を行わせる請求項１１に記載の非一時的コンピュータ可読記憶媒体。
【請求項１４】
前記コンピュータ命令が、前記コンピュータシステムに、
第１の暗号鍵をサーバに格納するステップと、
第２の暗号鍵をユーザデバイスに送信するステップと、
前記ユーザデバイスから前記第１の暗号鍵の要求を受信するステップと、
前記要求を受信した時に、前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されているかどうかを判断するステップと、
前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されていないと判断した場合、前記第１の暗号鍵の送信を拒否するステップと、を行わせる請求項１１に記載の非一時的コンピュータ可読記憶媒体。
【請求項１５】
前記コンピュータ命令が、前記コンピュータシステムに、
前記ブロックチェーンのセマンティクスの解釈に関する更新を受信するステップと、
前記ブロックチェーン内に前記更新を格納することにより、複数のユーザデバイスに亘る前記ブロックチェーンの前記セマンティクスの解釈の一貫性を確保するステップと、を行わせる請求項１１に記載の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】
（関連出願の相互参照）
本出願は、２０２０年３月２４日に出願された米国特許出願第１６／８２８，００３号の優先権及び利益を主張し、同出願の全体が参照により本明細書に組み込まれる。
【０００２】
本出願は、セキュアファイルシステムへのアクセスを管理することに関し、より具体的

には、企業環境においてグループ権限及びセキュアファイルシステムへのアクセスを管理する方法及びシステムに関するものである。

【背景技術】

【０００３】

今日、我々のコンピューティングシステムが直面している大きな問題は、内部脅威、即ち、集中型インフラストラクチャ及びそれを管理及び提供する人々への依存による脅威が、全体的な脅威モデルの一環であるということである。ＭｉｃｒｏＳｏｆｔ ＳｈａｒｅＰｏｉｎｔ管理者ロールのドキュメントによると、「グローバル管理者とＳｈａｒｅＰｏｉｎｔ管理者は、全てのサイトと各ユーザのＯｎｅＤｒｉｖｅに自動的にアクセスできるわけではないが、任意のサイトやＯｎｅＤｒｉｖｅへのアクセスを自分に与えることができる。」とある。セキュリティとアクセスコントロールの管理と提供を管理者に一元化する現在のインフラへのアプローチでは、人々に、その情報を知る必要がなくても情報にアクセスすることを可能にしている。その結果、自分が管理しているサーバに格納されているデータを読む権限を有していない管理者でも、その権限を有さずにアクセスしてデータを読むことができる。

【発明の概要】

【０００４】

本明細書では、ファイルシステムにアクセスする権限を有するユーザにのみアクセスを付与することによって、セキュアファイルシステム、及びファイルシステムにアクセスする権限を管理するシステムを提示する。ユーザに付与されるアクセスは、ユーザの権限を超えることはできない。システム内のユーザは、各ユーザに固有の暗号鍵を用いて識別される。ユーザの権限は、ブロックチェーンのような線形シーケンスに記録され、複数のデバイスに分散して配置される。複数のデバイスは夫々独立に線形シーケンスの各ブロックの有効性を検証し、侵害された中央のサーバが不正なアクセスを付与することを防止する。線形シーケンスの有効性は、線形シーケンスの分岐、線形シーケンス内のブロックの削除、線形シーケンス内のブロックの変更等、線形シーケンスに対して特定の操作が行われないようにすることで保証される。新たなブロックを線形シーケンスに追加する前に、ブロックの有効性は、ブロックの追加を要求するユーザが線形シーケンスにブロックを追加する権限、より具体的にはブロックの内容によって指定される操作を実行する権限を有することを保証する為に、システム内の各デバイスによって独立して計算される。

【０００５】

ブロックチェーン自体は暗号化されていなくてもよく、ブロックチェーンへのアクセスは、ブロックチェーン自体及び企業の情報技術（ＩＴ）環境で動作するアクセス制御サーバによって規制され得る。ブロックチェーンとアクセス制御サーバのような複数のソースで定義された権限を組み込む為に、複数のソースから来る複数層のパーミッション、即ち制約を含むトークンを作成することができる。パーミッションが追加される毎に、トークンによって付与される権限は減衰する。ブロックチェーンへのアクセスを制御するプロセッサがトークンを受信すると、プロセッサは、トークンの有効性とトークンによって付与された権限を確認し、要求者がブロックチェーンの少なくとも一部にアクセスする権限を付与されているかどうかを判断できる。

【図面の簡単な説明】

【０００６】

【図１】分散型環境において、グループの権限と暗号的に安全なデータへのアクセスを管理する為のシステムを示す図である。

【図２】チーム線形シーケンスと空間線形シーケンスを示す図である。

【図３】チーム線形シーケンスと空間線形シーケンスとの間の線形順序付けを示す図である。

【図４】ブロックの構造を示す図である。

【図５】システム内に存在し得るポリシーの様々なレイヤーの検証を示す図である。

【図６】システム内に存在し得る様々な暗号ＩＤを示す図である。

【図７】ブロックがどのように複数のデバイスに分散され得るかを示す図である。
【図８】ブロックを含む線形シーケンスを示す図である。
【図９】チーム線形シーケンスと空間線形シーケンスを示す図である。
【図１０Ａ】乃至
【図１０Ｂ】悪意のあるアクターがシステムに侵入しようとする場合の権限計算を示す図である。
【図１１Ａ】乃至
【図１１Ｃ】暗号化されたデータへのアクセスが、権限の失効時にどのように制御され得るかを示す図である。
【図１２】分散型台帳を介して、１つ以上の信頼できるデバイスによる暗号化されたデータへのアクセスとは別に権限を管理する方法のフローチャートであり、信頼できるデバイスの各々は、少なくとも１つの暗号鍵ベースのアイデンティティに対応しているのを示すフローチャートである。
【図１３】分散型台帳を使用して暗号化されたデータへのアクセスを管理する方法のフローチャートである。
【図１４】一実施形態による、セキュアファイルシステムがどのように企業情報技術（ＩＴ）インフラストラクチャに統合され得るかを示す図である。
【図１５】別の実施形態による、セキュアファイルシステムが企業ＩＴインフラストラクチャにどのように統合され得るかを示す図である。
【図１６Ａ】ブロックチェーンを使用してどのようにクロックが実装され得るかを示す図である。
【図１６Ｂ】クロックブロックチェーンの内容を示す図である。
【図１７】暗号化ツリーを示す図である。
【図１８】トークンの構造を示す図である。
【図１９】リプレイ攻撃を防止するトークンを示す図である。
【図２０】リカバリー鍵がどのように使用され得るかを示す図である。
【図２１】ユーザデバイスが侵害された時に暗号化されたデータへの攻撃を制限する分割鍵システムを示す図である。
【図２２】ブロックチェーンのセマンティクスの解釈に対する更新を示す図である。
【図２３】認可クレデンシャルを提供するトークンを生成する方法のフローチャートである。
【図２４】減衰トークンを作成する方法のフローチャートである。
【図２５】本明細書で論じる方法論又はモジュールの何れか１つ以上を機械に実行させる為の命令のセットが実行され得る、コンピュータシステム２５００の例示的形態の機械の図式的表現である。
【発明を実施するための形態】
【０００７】
（分散型環境におけるグループ権限及びセキュアデータへのアクセスの管理）
図１は、分散型環境においてグループ権限及び暗号的に安全なデータへのアクセスを管理する為のシステムを示す。サーバ１００は、エンドポイントとも呼ばれる複数のデバイス１１０、１２０、１３０、１４０、１５０と通信している。デバイス１１０～１５０の各々は、夫々、アリス、ボブ、キャロル、デイブ、エレン等のエンティティ又はユーザと関連付けられ得る。各ユーザアリス～エレンは、本出願で後に説明するように、固有暗号化ユーザ識別（「ＩＤ」）を有し得、各デバイスは固有暗号デバイスＩＤを有し得る。各暗号ユーザＩＤは、それに関連する１つ以上の暗号デバイスＩＤを有し得る。
【０００８】
固有暗号化ユーザＩＤは、図１に示すように、チーム１及びチーム２のようなチームに分けられ得る。例えば、図１に示すように、アリスの暗号ユーザＩＤ、ボブの暗号ユーザＩＤ、及びキャロルの暗号ユーザＩＤはチーム１のメンバーであり得るのに対し、デイブの暗号ユーザＩＤ及びエレンの暗号ユーザＩＤはチーム２のメンバーであり得る。チーム

１及びチーム２は、図１に示すように、相互排他的なメンバーシップを有し得、又は部分的に重複するメンバーシップを有し得る。チームメンバーシップは、例えば、チーム線形シーケンス１６０、１７０（簡潔にする為、チーム線形シーケンスの１つのインスタンスのみにラベルを付けている）のような線形シーケンスで記録され得る。各チーム１、２は、夫々１つのチーム線形シーケンス１６０、１７０を有し得る。チーム線形シーケンス１６０、１７０や空間線形シーケンス１９０、１９２、１９４等の線形シーケンスは、台帳に類似した暗号化されたデータ構造である。線形シーケンスは複数のデバイスに分散され得、各デバイスは独立して線形シーケンスを検証するので、線形シーケンスは分散型台帳を表し得る。

【０００９】

各チーム１、２は、１つ以上の空間１８０、１８２、１８４（簡潔にする為、空間の１つのインスタンスのみにラベルを付けている）を有し得る。各空間１８０、１８２、１８４は、暗号化されたデータ及び暗号化されたデータへのアクセスを有するメンバーを含む仮想区画であり得る。チームメンバーのサブセットは、１つ以上の空間１８０、１８２、１８４に含まれ、１つ以上の空間１８０、１８２、１８４に関連付けられた暗号化されたデータにアクセスする権限を与えられ得る。例えば、チーム１は空間１８０を有し、チームメンバーであるアリス、ボブ及びキャロルは全員が空間１８０に招待される。別の例では、チーム２は空間１８２と１８４を有する。空間１８２にはデイブのみがメンバーとして存在するのに対し、空間１８４にはデイブとエレンの両方がメンバーとして存在する。各空間１８０、１８２、１８４は、空間メンバーのみがアクセスできるように暗号化されたデータを有し得る。暗号化されたデータは、コンテンツ又はデータ、或いはその両方を含み得る。例えば、暗号化されたデータは、ファイル、ファイルシステム、文書、及び／又はインスタントメッセージ、電子メール、チャットメッセージ、テキストメッセージ等のメッセージを含み得る。

【００１０】

一例では、ユーザアリス、ボブ、キャロルのみが、空間１８０に関連付けられた暗号化されたデータに対する権限を有する。別の例では、ユーザデイブのみが空間１８２に関連付けられた暗号化されたデータに対する権限を有するのに対し、ユーザデイブ及びエレンの両者が空間１８４に関連付けられた暗号化されたデータへのアクセス権限を有している。暗号化されたデータに対する権限は、暗号化されたデータの読み取り、書き込み、及び／又は修正するパーミッションを含み得る。暗号化されたデータへのアクセスは、アクセスを要求する暗号ユーザＩＤが、アクセスを包含する権限を有することを確認した時に付与され得る。

【００１１】

例えば、ユーザエレンの暗号ユーザＩＤは、暗号化されたデータを読み取る権限を有し得る。しかし、ユーザエレンの暗号ユーザＩＤが暗号化されたデータに書き込むことを要求した場合、ユーザエレンの暗号ユーザＩＤは、権限がない為、暗号化されたデータに書き込む為のアクセスを拒否されることになる。言い換えると、本明細書で開示されたシステムでは、暗号化されたデータへのアクセスは、暗号化されたデータに関連付けられた権限を超えることはできない。

【００１２】

別の実施形態では、チーム１、２は存在せず、ユーザは１つ以上の空間１８０、１８２、１８４にグループ化され得る。空間を生成する為に、システム内に存在する暗号ユーザＩＤの一般的なプールを検索して、空間のメンバーを定義することができる。チーム線形シーケンス１６０、１７０は、対応する空間線形シーケンス１９０、１９２、１９４に統合され得る（簡潔にする為、空間線形シーケンスの１つのインスタンスのみにラベルを付けている）。例えば、空間線形シーケンス１９０は、チーム線形シーケンス１６０及び空間線形シーケンス１９２を含み得、空間線形シーケンス１９２は、チーム線形シーケンス１７０及び空間線形シーケンス１９２を含み得るのに対し、空間線形シーケンス１９４は、チーム線形シーケンス１７０及び空間又はリスト１９４を含み得る。

【００１３】
暗号化されたデータに関連する権限のレコードは、本願で更に説明するように、チーム線形シーケンス１６０、１７０と対応する空間線形シーケンス１９０、１９２、１９４を組み合わせることによって計算され得る。権限及びメンバーシップを格納することに加えて、空間線形シーケンス１９０、１９２、１９４は、暗号化されたデータ及び／又は暗号化されたデータへの参照も格納できる。
【００１４】
チーム線形シーケンス１６０、１７０及び空間線形シーケンス１９０、１９２、１９４のコピーが、サーバ１００と同様に、その暗号ユーザＩＤが対応するチーム及び空間のメンバーである全てのデバイス１１０～１６０に配布され得る。例えば、デバイス１１０～１３０は、デバイス１１０～１３０に関連する暗号ユーザＩＤがチーム２及び空間１８０のメンバーである故に、チーム線形シーケンス１６０及び空間線形シーケンス１９０のコピーを有する。別の例では、デバイス１４０は、デバイス１４０に関連するユーザデイブの暗号ユーザＩＤがチーム２と空間１８２、１８４のメンバーである故に、チーム線形シーケンス１７０と空間線形シーケンス１９２、１９４のコピーを有する。第３の例では、デバイス１５０は、デバイス１５０に関連付けられたユーザエレンの暗号ユーザＩＤがチーム２及び空間１８４のメンバーである故に、チーム線形シーケンス１７０及び空間線形シーケンス１９４のコピーを有する。
【００１５】
チーム線形シーケンス１６０、１７０及び空間線形シーケンス１９０、１９２、１９４に含まれるメタデータは平文で格納され得るのに対し、データの残りは暗号化され得る。メタデータは、チーム線形シーケンス１６０、１７０及び空間線形シーケンス１９０、１９２、１９４内に格納される権限情報、ポリシー情報、ロール等を含み得る。データの残りは、ファイル、ファイルシステム、メッセージ、及び／又は他の機密データ等の機密データを含み得る。例えば、暗号化されたデータがファイル及び／又はファイルシステムを含む場合、ファイル名は機密データの一部となり得る。ファイルシステム、ファイル、メッセージは、データの暗号化に使用された暗号鍵を知ることによってのみアクセスされ得る。攻撃者が暗号鍵、ユーザの秘密鍵、及び／又は認可されたエンドポイントデバイスの制御に成功したとしても、システムの侵害は限定的であろう。
【００１６】
例えば、攻撃者が空間１８２に関連する暗号鍵の制御を獲得した場合、攻撃者は空間１８２内の機密データにのみアクセスでき、空間１８４及び１８０内の機密データにはアクセスできないであろう。攻撃者がエレンの秘密鍵を入手した場合、攻撃者は空間１８４内の機密データにのみアクセスでき、空間１８０及び１８２内の機密データにはアクセスできない。このように、空間１８０、１８２、１８４への権限とアクセスを区分けすることで、システムの侵害を限定できる。
【００１７】
図２は、チーム線形シーケンスと空間線形シーケンスを示す。チーム線形シーケンス２００は、チームメンバーのアイデンティティ及びチーム内の権限を追跡する為に使用され得る。空間線形シーケンス２１０は、チームメンバーのサブセットを認めることができる安全な区画を形成する為に使用され得る。安全な区画は、データを管理し、空間メンバー間で共有鍵をネゴシエートする為に使用される。チーム線形シーケンス２００は、システムポリシーを含むプログラム２２０に接続され得、又、ファクトを含むデータベース２３０に接続され得る。空間線形シーケンス２１０は、チーム線形シーケンス２００に依存して空間内のポリシーを決定できる。空間線形シーケンス２１０は、ファクトを含むデータベース２５０に接続され得る。
【００１８】
チーム線形シーケンス２００及び空間線形シーケンス２１０は、夫々複数のブロック２０５、２０７、２０９、２１５（簡潔にする為４つのみにラベルを付けている）を含み得る。チーム線形シーケンス２００の初期ブロック２０５は、チームの為のポリシーを定義

することができる。ポリシーは、ロールと、ロールに関連する権限とを指定できる。例えば、ポリシーは、「管理者のみがチーム内に空間を作成できる」と指定できる。チームポリシーは、ポリシーデータベースに格納されたポリシーテンプレートから取得され得るものであり、及び／又は、第１のブロック２０５をインスタンス化する際に修正され得る。或いは、第１ブロック２０５は、ポリシーテンプレートを参照することなく、チームポリシーを定義できる。ポリシープログラム２２０は異なるチーム間で共有され得る。しかし、異なるチームは、異なるファクトデータベース２３０を有し得る。又、チームポリシーは、チームポリシー定義ブロック２０５が修正を許可する場合、チーム線形シーケンス２００に追加された後時のブロックによって修正され得る。

【００１９】

ポリシープログラム２２０は、初期ブロック２０５にポリシーとして含まれる及び／又は修正され得るポリシーテンプレートを格納することができる。ファクトデータベース２３０は、システム内のユーザ及びユーザ権限を定義できる固有キー‐値ペアを含み得る。例えば、ファクトデータベース２００に追加され得るキー‐値ペアは、アリスが管理者であることを指定するブロックが検証された後で、「アリス‐管理者」となり得る。

【００２０】

チーム線形シーケンス２００のブロック２０７は、チーム２００のメンバーであるユーザのプロファイルを含み得る。ユーザプロファイルは、Ｒｉｖｅｓｔ‐Ｓｈａｍｉｒ‐Ａｄｌｅｍａｎ（ＲＳＡ）又はＤｉｆｆｉｅ　Ｈｅｌｌｍａｎ（ＤＨ）等の非対称暗号アルゴリズムにおける公開鍵等の暗号ユーザＩＤによって表され得るユーザのアイデンティティ２４０を含み得る。秘密鍵は、そのユーザのみが所有し得る。ユーザプロファイルは、ユーザが承認した全てのデバイスに関連付けられた暗号デバイスＩＤ２４２、２４４も含み得る。

【００２１】

デバイスがシステムに追加される方法は複数ある。例えば、ユーザは、デバイスをシステムに追加する要求を送信することによってデバイスを承認することができ、この要求は、ユーザの秘密鍵で署名される。別の例では、デバイスを承認する為に、ユーザは複数段階のプロセスを実行することができる。第１のステップでは、ユーザは非対称暗号アルゴリズムを使用して新たなデバイス鍵のセットを作成することができる。第２のステップでは、ユーザは、ユーザの秘密鍵でデバイス鍵に署名し、デバイス公開鍵とユーザの秘密鍵の署名を含むデバイス証明書を作成することができる。第３のステップでは、デバイスは、第２のステップからの証明書を含む、チームに追加する要求を送信することができ、この要求は、デバイスの秘密鍵を使用して署名される。システムは、ユーザの公開鍵を用いて要求を検証することにより、チームメンバーが要求を行ったことを認証することができる。暗号デバイスＩＤは、非対称暗号アルゴリズムの公開鍵の暗号化ハッシュであり得、一方、秘密鍵は、デバイスにのみ知られ得るものであり、デバイスによって実行されるアクションを認証する為に使用され得る。

【００２２】

チーム線形シーケンス２００のブロック２０９は、新規ユーザの追加、空間線形シーケンス２１０の作成、ポリシー２０５の変更、既存ユーザの削除、及び／又はユーザのロールの変更等のイベントを含み得る。

【００２３】

空間線形シーケンス２１０のブロック２１５は、空間へのユーザの追加、空間への暗号化されたデータの追加、空間からのユーザの削除等のイベントを含み得る。空間線形シーケンス２１０の各イベント２１５は、チーム線形シーケンス２００で定義されたポリシーに準拠する。幾つかの実施形態では、ポリシーは、空間線形シーケンス２１０では変更できず、チーム線形シーケンス２００で変更されなければならない。チームは、異なるポリシーを有する複数の空間タイプを定義し、異なる空間タイプに対応する空間を確立することによって、異なるポリシーを有する複数の空間を有し得る。例えば、空間タイプは、全てのユーザが管理者のロールを有し、管理者が他のユーザを追加及び削除し、暗号化され

たデータに対する読み取り及び書き込みアクセスを有し得る「管理者空間タイプ」を含み得る。別の例では、空間タイプは「層別空間タイプ」を含み得、その場合一部のユーザが管理者ロールを有し、一部のユーザがユーザロールを有し、管理者ロールはユーザロールより多くの権限を有する。ユーザの権限を変更するチーム線形シーケンス２１０内のイベントは、ファクトデータベース２５０に格納され得る。

【００２４】

ユーザポリシー２０５は、特定の属性に一致するユーザが、限られた時間だけ空間線形シーケンス２１０及び空間暗号化されたデータにアクセスすることを可能にするように定義され得る。時間の経過は、空間線形シーケンス２１０の各ブロック２１５に対してタイムスタンプ２６０、２７０、２８０、２９０を提供できる常に増加するクロックによって測定され得る。タイムスタンプ２６０は、例えば、「２０１９年１１月２１日ＡＭ１１:４６（ＰＳＴ）」と記載され得る。空間線形シーケンス２１０において、タイムスタンプ２６０、２７０、２８０、２９０は、後続のブロック間で常に増加している。空間線形シーケンス２１０及び関連する暗号化されたデータへの時間制限されたアクセスを可能にする為に、ユーザポリシー２０５は、「プロファイル１に関連するユーザは、２０１９年１２月２日まで線形シーケンスにアクセスすることができる」と述べることができる。

【００２５】

図３は、チーム線形シーケンスと空間線形シーケンスとの間の線形順序付けを示す。権限を少なくとも部分的に定義するポリシーはチーム線形シーケンス３１０に格納されるので、空間線形シーケンス３００における権限を計算する為に、チーム線形シーケンス３１０への参照がなされる必要がある。

【００２６】

例えば、空間線形シーケンス３００のブロック３２０において、空間ユーザは、別のユーザを追加することを要求する。チーム線形シーケンス３１０のブロック３３０において、ポリシーは、別の空間ユーザを追加する空間ユーザの権限を定義したが、チーム線形シーケンス３１０のブロック３４０において、ポリシーは、空間ユーザが他の空間ユーザを追加することを防ぐ為に修正された。ブロック３２０が有効であり、空間線形シーケンス３００に追加されるべきかどうかを決定する為に、チーム線形シーケンス３１０のブロック３３０、３４０と空間線形シーケンス３００のブロック３２０、３５０、３６０の線形シーケンスが確立される必要がある。

【００２７】

線形シーケンスを確立する為に、空間線形シーケンス３００におけるブロック３２０、３５０、３６０とチーム線形シーケンス３１０におけるブロック３３０、３４０との間に時間的関係３７０、３８０、３９０が確立され得る。時間的関係３７０、３８０、３９０は、空間ブロック３２０、３５０、３６０から、空間ブロック３２０、３５０、３６０の直前又は直後の後継者であるチームブロックへのポインタを含み得る。図３において、時間的関係３７０、３８０、３９０は、空間ブロック３２０、３５０、３６０から直前のチームブロックを指し示す。例えば、時間的関係３７０、３８０は、チームブロック３３０が空間ブロック３２０、３５０の直前であることを示し、空間ブロック３２０、３５０がチームブロック３３０の後であるがチームブロック３４０の前に作成されたことを意味する。同様に、時間的関係２９０は、空間ブロック３６０がチームブロック３４０の後に作成されたことを示す。

【００２８】

図４は、ブロックの構造を示す。ブロック４００、４１０は、１つ以上のイベント１～６を含み得る。ブロック４００、４１０のイベント１～６はアトミックであり得る。各イベント１～６は単一のブロック４００、４１０においてコミットされる。

【００２９】

トランザクションは、最後のイベントが署名される１つ以上のイベントを含み得る。トランザクションは、図１におけるクライアント１１０のような単一のクライアントによって生成され得る。トランザクションが複数のイベント、例えばイベント１及び２を含むト

ランザクション１を含む場合、イベントはサーバに送信される前にクライアントによって順序付けされ得る。イベントの順序付けは、矢印４２０、４３０、４４０、４５０を使用して示される。複数のクライアントからトランザクション１～４を受信すると、サーバは、矢印４２０、４３０、４４０、４５０によって示される順序に従ってトランザクションを順序付けることができる。

【００３０】

１つのトランザクション内のイベントは互いに指し示し合い、最後のイベントが署名される。例えば、イベント１と２は単一のトランザクション１を形成する。同様に、イベント５と６は単一のトランザクション４を形成する。イベント３とイベント２の間の矢印４２０は、サーバがイベント１の後にイベント３を線形シーケンスにコミットすべきことを示す。同様に、イベント５とイベント６の間の矢印４３０は、サーバがイベント５の後にイベント６を線形シーケンスにコミットすべきことを示す。

【００３１】

一実施形態では、イベント１及び２は単一のクライアントから来るが、イベント３はイベント１及び２と同じクライアントから来ることも、異なるクライアントから来ることもあり得る。同様に、イベント５及び６は、単一のクライアントから来るが、イベント４は、同じクライアントから来ることも、異なるクライアントから来ることもあり得る。更にこの実施形態では、図４に示すように、トランザクション１～４を作成するクライアントが少なくとも１つ、最大で４つ存在することが可能である。例えば、単一のクライアントが、イベント１及び２を含むトランザクション１をオーサリングすることができる。第２のクライアントは、イベント３を含むトランザクション２をオーサリングすることができる。第３のクライアントは、イベント４を含むトランザクション３をオーサリングすることができ、第４のクライアントは、イベント５及び６を含むトランザクション４をオーサリングすることができる。

【００３２】

複数のイベント、例えばイベント１～３をアトミックな方法で追加する必要がある場合、イベント１～３を単一のブロック４００に結合し、ブロック４００は線形シーケンスで格納され得る。ブロック４００，４１０では、夫々イベント３、６のような最後のイベントのみが署名され得、ブロック４００、４１０のイベント１～３、４～６は、意図したブロックに意図した順序で全て現れなければ、何れも有効でない。

【００３３】

ブロック４００、４１０内のイベント１～６は、本願で説明するように、暗号的に署名され、認証された線形シーケンスで記録される。線形シーケンスの構造は、或る指数ｎを有するブロックを受け入れる図１のデバイス１１０～１６０が、ｎ未満の指数を有する全てのブロックの内容に関して確実に合意することを保証する。ブロック４００、４１０は、図４において夫々指数１、２を有する。

【００３４】

ブロック４００、４１０が確定されると、ブロック４００、４１０はブロックＩＤを得る。ブロックを確定することは、ブロックを耐久ストレージにコミットし、ブロックを変更しないことを意味する。ブロックＩＤは、ブロックの指数と、そのコンテンツの暗号化ハッシュとのタプルである。所与のブロックｎについて、ブロックｎ＋１において線形シーケンスに追加されることが意図される全てのイベントは、ブロックｎのブロックＩＤを含み得る。これにより、デバイス１１０～１６０のイベントの意図した順序付けの保持が確実になる。更に、ブロック０からｎまでのブロック順序及び内容に同意するデバイスのみが、ブロックｎ＋１におけるイベントを受け入れることができる。

【００３５】

図５は、システム内に存在し得るポリシーの様々なレイヤーの検証を示す。イベント５００は、イベント５００の拒否又は受け入れの何れかを行い、任意にファクトデータベース５１０、５２０を更新し得る正規のポリシーを使用して処理される。ファクトデータベース５１０、５２０は、ポリシーに関する線形シーケンス上の指数である。指数は鍵値ペ

アの集合である。例えば、ファクトデータベース５１０、５２０は、イベント５００及びそれを許可したポリシールールによって生成されたファクトを記録できる。ファクトデータベース５１０、５２０は、イベント５００が有効であり、チーム線形シーケンス、又は空間線形シーケンス等の線形シーケンスで受け入れられるべきであるかどうかを判断する時に、ポリシープレッサーによって読み取られ得る。

【００３６】

図１のサーバ１００がイベント５００を受信すると、サーバは、イベント５００を次のブロックに含めることができる。デバイス１１０～１６０がイベント５００を受信すると、デバイスは、ビジネスニーズに基づいてイベント５００を処理できる。

【００３７】

システムポリシー５３０は、デバイス１１０～１６０及びサーバ１００によって実装される。全ての協働デバイス１１０～１６０は、同じブロックを処理するに当たり、同じシステムポリシー５３０を使用しなければならない。一実施形態では、システムポリシー５３０は、Ｒｕｓｔ、Ｐｙｔｈｏｎ、Ｇｏ、又は独自のポリシー言語等のプログラミング言語におけるソースコードとして実装され得る。システムポリシー５３０は、ブロック自体の構造を記述する。システムポリシー５３０は、「ブロックｎの全てのイベントはブロックｎ－１を参照しなければならない」等の線形シーケンスのコアプロトコルを記述する。

【００３８】

ユーザポリシー５４０は、図２におけるチームポリシー２０５であり得る。チームポリシー２０５は、顧客によって提供され、顧客の組織に合わせられ得る。全ての当事者がユーザポリシー５４０に同意することを保証する為に、ユーザポリシー５４０は、図３のチーム線形シーケンス３１０等の線形シーケンスに格納され得、特定の線形シーケンスで協働する全てのユーザに対して同じであることが保証される。チームポリシールールの一例は、「チーム管理者のみがチームに新メンバーを追加できる」である。

【００３９】

アプリケーションポリシー５５０は、図１のチーム１６０、１７０毎に定義され得る。アプリケーションポリシーと他のポリシーとの間の重要な違いは、アプリケーションポリシーが、任意の暗号文が復号された後に動作することである。その結果、アプリケーションポリシーは、サーバによって確認され得ない。アプリケーションポリシーは、権限を記録しなくてもよく、「２つのファイルが同じ名前を有することはできない」等の低レベルのルールを指定することができる。アプリケーションポリシー５５０は、「２つのファイルが同じ名前を有する場合、２番目のファイルは無視される」等の低レベルの競合を解決するルールを符号化することができる。システムポリシー５３０、ユーザポリシー５４０及び／又はアプリケーションポリシー５５０を含むポリシーは、図２のチーム線形シーケンス２００及び空間線形シーケンス２１０等の線形シーケンスを使用して、プログラム中に固定されるか、又は管理され得る。

【００４０】

図６は、システム内に存在し得る様々な暗号ＩＤを示す。アイデンティティは、セキュリティの基盤を形成する。アイデンティティは、ＲＳＡ、ＤＨ又は他の非対称アルゴリズムを使用してオフラインで生成された非対称鍵ペア等の暗号識別（ＩＤ）により表される。非対称鍵ペアからの公開鍵はエンティティの暗号アイデンティティとして使用されるのに対し、非対称鍵ペアからの秘密鍵はエンティティのみが知ることができる。ルート非対称鍵ペアをオフラインで生成することで、鍵ペアが悪意あるアクターによって侵害されることを防ぐことができる。デバイス鍵はオンラインで生成され得るが、ハードウェアセキュリティモジュール（ＨＳＭ）に格納されてもよい。

【００４１】

暗号デバイスＩＤや暗号ユーザＩＤ等の暗号ＩＤは、夫々デバイスやユーザ等、システム内の単一のエンティティを表す。その結果、暗号ＩＤを使用して確立されたＩＤはグローバルに区別される。一実施形態では、管理者がユーザを区別しなくても、ユーザはチーム間で一意である。又、メンバーを共有しないチーム等の非協働グループ間でも、暗号Ｉ

Ｄを使用して確立されたアイデンティティはグローバルに区別されたままである。
【００４２】
暗号ユーザＩＤは、システム内で運用されるデバイス鍵の署名／認証に使用され得る。デバイスは、管理署名鍵、メッセージ署名鍵、及びデバイス暗号化鍵の３つの鍵タイプを含む暗号デバイスＩＤを使用できる。これらの鍵タイプは全て非対称鍵ペアであり、アプリケーションの外部で、オペレーティングシステム（ＯＳ）鍵ストア又はハードウェアセキュリティモジュール（ＨＳＭ）で管理され得る。
【００４３】
アイデンティティ証明書６００は、復元秘密から決定論的に生成され得、新たなデバイスをプロビジョニングする時に復元秘密の効率的なハンドトランスクリプションを可能にする。
【００４４】
管理署名鍵６１０は、高リスクの操作で使用され得、任意の署名要求に対してユーザの存在証明を要求することができる。高リスクの操作の例は、ユーザを追加すること、又はパーミッションを変更することである。
【００４５】
メッセージ署名鍵６２０は、図１の装置１１０～１６０によって送信される殆どのデータに署名する為に使用され得、存在証明を必要としない。メッセージ署名キー６２０の一使用例は、チャットで送信されるメッセージに署名すること、又はアップロードされるファイルに署名することである。
【００４６】
デバイス暗号化鍵６３０は、デバイスに機密メッセージを送信することが必要な場合に使用され得る。デバイス暗号化鍵６３０は、デバイス間通信の前方秘匿性を提供する為に、頻繁にローテーションされ得る。
【００４７】
図７は、ブロックがどのように複数のデバイスに分散され得るかを示す。デバイス７００、７１０、７２０は、同じ空間及び／又はチームに属し得る。それらは全て、線形シーケンス７３０のコピーを有し得る。デバイス７００等のデバイスの１つが線形シーケンス７００に新たなブロック７４０を追加すると、デバイス７００は新たなブロック７４０をサーバ７５０に送信する。
【００４８】
サーバ７５０も線形シーケンス７３０のコピーを有する。サーバ７５０は、ユーザがブロック７４０に表される操作を実行する権限を有するかどうかを計算することによって、ブロック７４０が有効であるかどうかを計算できる。ユーザが権限を有するかどうかを判断する為に、サーバ７５０は、線形シーケンス７３０から権限を計算できる。権限計算を以下に説明する。
【００４９】
ブロックが有効であることをサーバ７５０が確認すると、サーバは、ブロック７４０をデバイス７１０、７２０に配布することができ、デバイスのほうも、ブロック７４０が有効であるかどうかを、線形シーケンス７３０に記録された権限に基づいて計算できる。デバイス７１０、７２０が、ブロックが有効でないと判断した場合、システム侵害が発生した可能性が高い為、デバイスはシャットダウンすることができる。
【００５０】
デバイス７００、７１０、７２０は、それらが同じ空間にある場合、暗号化されたデータ７６０を共有し得る。暗号化されたデータ７６０は、以下に説明する機密セッション鍵等の少なくとも１つの暗号キーを用いて暗号化される。異なるデバイス７００、７１０、７２０は、読み取り専用、書き込み専用、及び読み取りと書き込みの権限等、暗号化されたデータ７６０に対して異なる権限を有し得る。
【００５１】
サーバ７５０は、暗号化された機密データ７６０を格納することもできるが、サーバ７

５０は、機密セッション鍵を含めて如何なる暗号鍵も格納しない。サーバ７５０は、暗号化されたデータ７６０に対する如何なる権限も有さない。サーバ７５０は、データの可用性を確保する為に、暗号化されたデータ７５０のコピーを有する。例えば、デバイス７００、７１０がオフラインであり、空間の新たに追加されたメンバーとしてのデバイス７２０が、暗号化された機密データ７６０を要求した場合、サーバ７５０は、デバイス７００、７１０がオフラインであっても、デバイス７２０に暗号化された機密データ７６０を提供できる。

【００５２】

図８は、ブロックを含む線形シーケンスを示す。線形シーケンス８００は、少なくともブロック８１０、８２０、８３０を含む。ブロック８１０は複数のイベント８１２、８１４を含むのに対し、ブロック８２０はイベント８２２、８２４を含む。ブロック８１０は線形シーケンス８００の初期ブロックであり、線形シーケンス８００の権限を定義するポリシー８１６を含む。ブロック８２０等の後続のブロックにおける各イベントは、イベント８２２、８２４がポリシー８１６と一致することを確認する為に検証され得る。

【００５３】

初期ブロックに続く、ブロック８２０等の各ブロックは、先行ブロックの暗号化ハッシュ８２６を含み、それは、ブロック８２０についてはブロック８１０のハッシュとなる。先行ブロックの暗号化ハッシュを含むことによって、線形シーケンス８００内のブロックの順序付けを保証することができ、既存のブロックの順序変更又は編集、及び／又は線形シーケンス８００内の新たなブロックの挿入が検出され、自動的に拒否され得る。

【００５４】

線形シーケンス８００は、分岐のない線形シーケンスである為、ブロックの有効性を検証する為のプルーフ・オブ・ワークを必要としない。更に、ブロックがリストに追加された時点で、そのブロックの有効性が線形シーケンス８００に記録されたポリシーと権限に準拠していることが確認されており、そのブロックは削除され得ない。即ち、順序リスト８００はロールバックされ得ない。

【００５５】

初期ブロック８１０内では、イベント８１２がアリスをユーザとして確立する。このイベントはアリスによって署名されており、これは、アリスが自身の秘密鍵を使用して、ステートメント「アリスはユーザである」を暗号化することを意味する。アリスが本当にユーザとして確立されることを要求していることを検証する為に、プロセッサは、アリスの公開鍵を使用して、署名されたステートメント「アリスはユーザである」を検証し、検証が成功すれば、プロセッサには、アリスが本当にユーザを確立することを要求したことが分り得る。

【００５６】

イベント８１４はアリスを管理者として確立する。同様に、イベントは署名され、プロセッサは、上記で説明したように、アリスのアイデンティティを検証することができる。ブロック８１０は初期ブロックである為、線形シーケンスの為のポリシー８１６が確立される。例えば、ポリシー８１６は、「管理者のみがユーザを追加できる」と述べることができる。ブロック８１０が線形シーケンス８００にコミットされると、システムの有効なロールは、アリスが管理者であり、アリスがユーザであることである。

【００５７】

ブロック８２０のイベント８２２は、ボブがユーザであることを確立する。アリスがユーザを追加する権限を有するかどうかを計算する為に、プロセッサは、イベント８１２、８１４によって確立されたシステム内のポリシー８１６及びアリスのロールを確認することができる。ポリシーは、管理者のみがユーザを追加できることを指定するので、プロセッサは、アリスが管理者であるかどうかを確認する。アリスが管理者であることを確認すると、プロセッサは、アリスがボブをユーザとして追加する権限を有することを確認することができる。初期ブロック８１２、８１４に含まれるイベントに続く各イベントは、ポリシー８１６に対して確認され得、イベントを認可するポリシーは記録され得る。例えば

、イベント８２２について、ブロック８２０は、「管理者のみがユーザを追加できる」というポリシーを指し示すことができ、プロセッサは、「アリスが管理者である」というファクトデータベースに格納されたファクトがあることを確認できる。

【００５８】

イベント８２４はキャロルをユーザとして追加し、又、上記で説明したように、アリスによって署名されなければならない。イベント８２２は、イベントを認可したポリシー、即ち、「アリスが管理者である」というファクトによってサポートされる「管理者のみがユーザを追加できる」と述べるポリシーも指し示すことができる。ハッシュ８２６は、ブロック８１０、８２０の線形シーケンスを作成する。ブロック８２０が線形シーケンス８００にコミットされた後で、システムにおける有効なロールは、アリスが管理者であり、アリス、ボブ、及びキャロルがユーザであるということである。

【００５９】

他のロールは、法務、技術、営業等のチーム内で定義され得る。各ロールは、対応する空間タイプへのアクセス権を付与され得る。例えば、アリスが「法務」のロールを有する場合、アリスは、タイプ「法務」を有する全ての空間へのアクセスを付与され得る。アリスの「法務」ロールが取り消された場合、アリスは、タイプ「法務」を有する空間へのアクセスを自動的に失い得る。

【００６０】

図９は、チーム線形シーケンスと空間線形シーケンスを示す。この例では、チーム線形シーケンス９００とブロック線形シーケンスは、１つのみのイベントを含むブロックを含む。チーム線形シーケンス９００は、初期ブロック９１０と、チームと任意の空間のポリシーを確立するポリシー９２０にリンクした接続９１５を使用して初期化される。ブロック９３０で、アリスがボブをユーザとして追加し、このイベントは、接続９３５を使用して、それを認可するポリシー９２０にリンクさせることができる。

【００６１】

ブロック９４０で、ボブは空間「計画」を作成し、そのイベントは、ユーザとしてのボブが、空間を作成する権限を有することを保証する為に、ポリシー９２０にリンクされる。デフォルトで、空間の作成者のみが、空間線形シーケンス９７０上での空間に対する管理権限を付与される。ポリシー９２０がユーザが空間を作成する権限を有する場合、ブロック９３０は検証され、イベントは接続９４５を使用してポリシー９２０にリンクされ、ブロック９４０はチーム線形シーケンス９００に追加される。ボブが空間「計画」を作成すると、空間線形シーケンス９７０が確立され、初期ブロック９８０がブロック９４０を指し示し、リストがブロック９４０の後であるが以下に説明するブロック９５０の前に作成されたことを示す。

【００６２】

ブロック９５０で、アリスは、空間作成を管理者に限定する。このアクションはポリシー９２０を変更する。ブロック９５０が有効であるかどうかを検証する為に、プロセッサは、ポリシー９２０を確認して、ポリシーが、管理者がポリシーを変更することを許可しているかどうかを確認する必要がある。ポリシー９２０が管理者がポリシーを変更することを許可している場合、イベントは、接続９５５を使用して、管理者がポリシーを変更することを許可するポリシー９２０の部分にリンクされ、新たなポリシー９２５が確立される。ブロック９５０の後で、ボブが空間を作成できないとしても、ブロック９４２でボブが空間を作成する権限を有していたので、ボブが作成した空間は有効である。ブロック９６０で、アリスはキャロルを追加し、このイベントは新たなポリシー９２５に対して確認される。このイベントが新たなポリシー９２５によって承認されると、接続９６５は、イベントを承認する新たなポリシー９２５のその部分に対して確立される。

【００６３】

空間線形シーケンス９７０のブロック９９０において、ボブはキャロルをユーザとして追加する。ブロック９９０が有効であるかどうかを確認する為に、プロセッサは、新たなポリシー９２５を確認して、ポリシー９２５が空間ユーザが他の空間ユーザを追加するこ

とを許可しているかどうかを確認する必要がある。ポリシー９２５は、ユーザがユーザを追加することを許可し、イベントは、接続９９５を使用して、イベントを許可するポリシー９２５の部分にリンクされる。

【００６４】

先に説明したように、チームメンバーのみがその空間に追加され得る。ボブがブロック９５０の前にキャロルを追加しようとすると、ファクトベース及び／又はチーム線形シーケンス９００を確認した後で、プロセッサは、キャロルがチームのメンバーでないと判断できるので、プロセッサはブロック９９０の追加を承認しないことになる。しかし、ブロック９６０の後にボブがキャロルを空間線形シーケンス９７０に追加しようとすると、プロセッサは、ポリシーが空間ユーザが空間ユーザを追加することを許可し、キャロルがチームのメンバーであるので、ブロック９９０の追加を承認することになる。

【００６５】

図１０Ａ～Ｂは、悪意のあるアクターがシステムに侵入しようとする場合の権限計算を示す。この例は、各デバイス１０１０～１０３０が線形シーケンス１０４０の有効性と線形シーケンス１０４０に記録された権限を独立して確認し保証するので、サーバ１０００の侵害がデバイス１０１０、１０２０、１０３０を侵害しない様子を示すものである。

【００６６】

サーバ１０００が、例えば、不正なブロック１０５０を不正に検証してデバイス１０１０～１０３０に配布するようにサーバ１０００を強制する悪意のあるサーバ管理者によって侵害された場合、各デバイス１０１０～１０３０は、ブロック１０５０の有効性を独立して検証できる。

【００６７】

図１０Ｂにおいて、各デバイス１０１０～１０３０は、線形シーケンス１０４０の最終ブロックのハッシュ１０６０が有効であることを独立して検証することができる。各デバイス１０１０～１０３０は、ユーザが有効なロールであることを検証できる。各デバイス１０１０～１０３０は、ブロック１０５０を提出する前に、マル（Ｍａｌ）が、サーバ１０００が自分の為に公開鍵と秘密鍵とを生成し、公開鍵をデバイス１０１０～１０３０に配布することを要求したので、マルの署名が有効であることを確認することができる。しかし、システムポリシーによれば、管理者やユーザ等の既存のメンバーのみが新たなユーザを追加でき、且つマルは管理者でもユーザでもないので、デバイス１０１０～１０３０はブロック１０５０が有効でないと判断することができる。ブロック１０５０はポリシーを満たさないので、全てのデバイス１０１０～１０３０はブロック１０５０を拒否することができる。更に、デバイス１０１０～１０３０は、サーバ１０００が侵害されたと結論付けることができる為、サーバ１０００から無効なトランザクションを受信すると、デバイス１０１０～１０３０はシャットダウンすることができる。

【００６８】

図１１Ａ～Ｃは、暗号化されたデータへのアクセスが、権限失効時にどのように制御され得るかを示す。デバイス１１００、１１１０、１１２０は、例えば、同じ空間の一部であることによって、暗号化されたデータ１１３０を共有し得る。暗号化されたデータ１１３０は、データの暗号化及び復号の両方に同じ鍵を使用する対称鍵アルゴリズムである高度暗号化標準（Ａｄｖａｎｃｅｄ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ、ＡＥＳ）を使用して暗号化され得る。暗号化された機密データ１１３０は、デバイス１１００～１１２０及びサーバ１１４０に格納され得る。ＡＥＳキーは、暗号化されたデータ１１３０にアクセスする権限を有するデバイス１１００～１１２０にのみ知られ得る。

【００６９】

アリスが管理者であり、ユーザを削除する権限を有すると仮定すると、アリスは、「キャロルはユーザではない」というブロック１１５０をサーバに提出することができ、従って、アリスとボブとの間で共有される将来のあらゆる暗号化されたデータに対するキャロルの権限を失効させることができる。サーバ１１４０は、図１１Ｂに見られるように、ブロック１１５０を全てのデバイス１１１０～１１２０に配布することができる。

【００７０】
デバイス１１１０～１１２０によってブロック１１５０が検証されると、キャロル及び彼女のデバイス１１２０は、任意の将来の暗号化されたデータにアクセスする権限を喪失する。キャロルと彼女のデバイス１１２０が、アリスとボブの間で共有される任意の将来の暗号化されたデータにアクセスできないことを保証する為に、デバイス１１００、１１１０は、新たなチャネルセッション鍵を生成する。
【００７１】
新たなチャネルセッション鍵は、例えば、Ｐ‐３８４楕円曲線を用いた楕円曲線暗号等の暗号方式を用いて生成され得る。チャネルセッション鍵は、楕円曲線暗号の場合、ドメインパラメータ等の秘密鍵生成材料１１７０を使用して生成され得る。ドメインパラメータは、楕円曲線$Y^2 = X^3 + AX + B$を定義する定数Ａ、Ｂを含み得る。
【００７２】
鍵を共有する新たなデバイスのグループは、線形シーケンス１１６０に基づいて計算される。秘密鍵生成材料１１７０は、新たなデバイスグループに残ったデバイスに属する非対称暗号アルゴリズムの公開鍵の夫々を使用して暗号化される。暗号化された秘密鍵生成材料１１７０は、グループ内の全てのデバイスに配布される。デバイス１１００、１１１０は、自身の非対称暗号アルゴリズムの秘密鍵を使用して、受信した暗号化メッセージを復号してもよい。その結果、暗号化に用いた公開鍵に対応する秘密鍵を有するデバイス１１００、１１１０のみが、新たなチャネルセッション鍵を計算することができる。
【００７３】
秘密鍵生成材料１１７０を受信するデバイス１１００、１１１０は、チャネルセッション鍵の秘密部分１１７２とチャネルセッション鍵の公開部分１１７４とを計算することができ、チャネルセッション鍵の公開部分１１７４を線形シーケンス１１６０に記録できる。チャネルセッション鍵の公開部分１１７４を線形シーケンス１１６０にコミットした結果、線形シーケンス１１６０へのアクセスを有するクライアントは、公開部分１１７４にアクセスし、線形シーケンス１１６０への書き込み専用アクセスを有し得る。クライアントは、秘密鍵生成材料１１７０及び／又はチャネルセッション鍵の秘密部分１１７２を有していないので、クライアントは、線形シーケンス１１６０に関連付けられた暗号化されたデータの何れも読み取ることができない。チャネルセッション鍵の秘密部分１１７２は線形シーケンス１１６０に記録されない。
【００７４】
線形シーケンス１１６０は、プルーフ・オブ・ワークを必要とせず、線形シーケンス１１６０を複製することは計算上実行可能であり得るので、図１１Ｃに見られるように、侵害されたサーバ１１４０は、２つの異なる線形シーケンス１１６２、１１６４を２つのユーザグループに提示することができる。例えば、侵害されたサーバ１１４０は、デバイス１１２０へのブロック１１５０の配布を拒否することができ、従って、デバイス１１２０に、依然としてグループ内にあると信じさせ、デバイス１１２０の暗号化されたデータ１１９０をデバイス１１００及び１１１０と共有しようとさせることができる。その結果、新たなチャネルセッション鍵１１８０が、線形シーケンス１１６０の最終ブロック１１５０のハッシュに基づいて計算され得る。例えば、新たなチャネルセッション鍵１１８０は、ＨＫＤＦ(図１１Ｂのチャネルセッション鍵、ブロック１１５０のハッシュ)を実行することによって取得され得る。その結果、デバイス１１００及び１１１０は、デバイス１１２０と同じ最終ブロックを共有しないので、デバイス１１２０は、同じチャネルセッション鍵１１８０を計算できない。
【００７５】
同じ空間アリス及びボブのユーザに加えて、ゲストユーザは、アリス及びボブと同じ空間に含まれる暗号化されたデータへのアクセスを一時的に許可され得る。ゲストユーザは、チーム線形シーケンス１１６０にアクセスできず、空間線形シーケンス１１６０内のユーザの権限を検証することができない。しかしながら、ゲストユーザは、依然として、アリス及びボブとチャネルセッション鍵をネゴシエートし、暗号化されたデータ１１３０へ

の一時的なアクセスを付与され得る。

【００７６】

図１２は、分散型台帳を介して、１つ以上の信頼できるデバイスによる暗号化されたデータへのアクセスとは別に権限を管理する方法のフローチャートであり、信頼できるデバイスの夫々は、少なくとも１つの暗号鍵ベースのアイデンティティに対応する。ステップ１２００において、プロセッサは、ユーザの権限を定義するブロックを作成することができる。ブロックは、ユーザを一意に識別する暗号ユーザＩＤと、暗号ユーザＩＤに関連付けられた権限とを含み得る。権限は、暗号化されたデータに対して実行する為に、暗号ユーザＩＤに関連付けられた少なくとも１つの操作を定義することができる。操作は、読み取り専用、書き込み専用、又は読み取り及び書き込みを含み得る。作業に承認を必要とするビットコイン台帳とは異なり、ブロックは作業を承認するエントリを必要としない為、ブロックの作成は、ブロックのプルーフ・オブ・ワークを生成する為に平均で１０分のプロセッサ時間を必要とするビットコインと比較して、プロセッサ負荷がより少ない。

【００７７】

ステップ１２１０において、プロセッサは、暗号化されたデータに関連する権限を定義する複数のブロックを含む線形シーケンスの末尾にブロックを付加し、複数のブロック内の各ブロックのメンバーシップ及び順序付けを保持することができる。ブロックのメンバーシップを保持する為に、線形シーケンス内のどのブロックも削除することはできない。言い換えると、削除は線形シーケンスに対して許されない操作である。複数のブロック間の各ブロックの順序を保持する為に、ロールバックは線形シーケンスの操作として許可されない。つまり、線形シーケンスは分岐できず、線形シーケンスに追加されたブロックの内容は変更及び／又は編集され得ない。ブロックの削除と修正の禁止は線形シーケンスの完全性を保証する。言い換えると、ブロックが線形シーケンスに追加されると、そのブロックは線形シーケンスに恒久的に存在する。更に、ブロックが線形シーケンスに追加される前に、ブロックの内容が線形シーケンスの先行するブロックと一致することを確認する為に、ブロックの内容が検証されなければならない。

【００７８】

ステップ１２２０において、プロセッサは、暗号化されたデータにアクセスする要求を受信することができる。要求は、要求を行うユーザに関連付けられた暗号ユーザＩＤを含み得る。暗号化されたデータへのアクセスは、読み取り専用、書き込み専用、又は読み取り及び書き込みの両方のアクセスを含み得る。

【００７９】

ステップ１２３０において、プロセッサは、図８、図９、及び図１０Ａ～Ｂに示すように、線形シーケンスに記録された権限を計算することによって、要求を行うユーザが暗号化されたデータにアクセスする権限を有するかどうかを判断することができる。権限を計算する為に、プロセッサは、初期ブロックから最終ブロックまでの線形シーケンスを確認して、ユーザのロールと各ロールに関連する権限を判断し、ユーザからの要求をユーザのロールと比較することができる。言い換えると、プロセッサは、線形シーケンスに記録された権限によって暗号ユーザＩＤによるアクセスが許可されていることを確認することによって、暗号ユーザＩＤによる暗号化されたデータへのアクセスを管理することができる。

【００８０】

ステップ１２４０において、プロセッサは、要求を行ったユーザが暗号化されたデータにアクセスする権限を有すると判断した場合に、要求を行ったユーザに対して暗号化されたデータへのアクセスを付与できる。

【００８１】

プロセッサは、ロール及びロールに関連付けられた権限を指定するポリシーを定義する線形シーケンスの初期ブロックを作成することができる。例えば、図８における初期ブロック８１０は、図８におけるポリシー８１６を定義する。更に、プロセッサは、暗号ユーザＩＤに関連するロールを定義する線形シーケンスのブロックを作成することができる。ブロックは、ポリシーを定義することに加えて、図８のイベント８１４でアリスが管理者

であることを定義する図８に示すような初期ブロック８１０であってもよく、又は、ブロックは、図８のブロック８２０等、ボブ及びキャロルをユーザとして定義する後続ブロックであってもよい。

【００８２】

複数のブロックにおける各ブロックの順序を保持する為に、プロセッサは、後続の各ブロックに各先行ブロックの暗号化ハッシュ（「ハッシュ」）を含めることができ、従って、図８で説明したように、順序付けシーケンスにおける如何なる変化も検出することを可能にする。例えば、プロセッサは、複数のブロックの中の第２のブロックに含まれるデータの第２の暗号化ハッシュを計算できる。第２のブロックは、線形シーケンスにおける初期ブロック、又は任意のブロックであり得る。プロセッサは、第２の暗号化ハッシュを第２のブロック内に格納することができ、第２のブロックに続くブロックに含まれるデータに第２の暗号化ハッシュを含めることができる。

【００８３】

線形シーケンス内のブロックのメンバーシップを保持する為に、プロセッサは、線形シーケンスに対して実行される操作のセットを定義することができ、定義されたセット以外の操作は、線形シーケンス内で実行され得ない。定義されたセットは、線形シーケンスの削除、分岐、及び／又は線形シーケンス内のデータの修正等の操作を除外することができる。

【００８４】

侵害された中央サーバによって線形シーケンス及び／又は暗号化されたデータが破損されるような障害の可能性を低減する為に、プロセッサは、線形シーケンスを複数のデバイスに分散することができる。複数のデバイスの各デバイスは、図２で説明したように、線形シーケンスに関連付けられた暗号ユーザＩＤによって暗号的に検証され得る。例えば、認可されたデバイスのリストにデバイスを追加するには、図２で説明したように、既に認可された暗号ユーザＩＤが、デバイスのアクセスを要求する必要がある。要求を受信すると、プロセッサは、暗号ユーザＩＤの公開鍵を使用して要求を復号することにより、暗号ユーザＩＤが実際に要求を行ったことを検証することができる。検証後、プロセッサは、暗号デバイスＩＤをデバイスに割り当てることができる。

【００８５】

暗号デバイスＩＤを有する複数のデバイスの内の各デバイスは、図１０Ａ～Ｂで説明したように、線形シーケンスに基づいて計算された権限に基づいて、要求の有効性を独立して検証することができる。或るデバイスが、計算された権限に基づいてブロックが有効であることを検証できない場合、そのデバイスはブロックの追加を拒否することができる。検証に失敗すると、デバイスは、デバイス上の暗号化されたデータの改竄を防止する為に、シャットダウンすることができる。

【００８６】

プロセッサは、効率化の為に、暗号化されたデータを共有できる人々のグループを見つける為に全ての暗号ＩＤを検索する必要がないように、暗号ユーザＩＤのチームを定義することができる。チームを作成する為に、プロセッサは、複数のブロックを含むチーム線形シーケンスを作成することができる。複数のブロックは、１つ以上のポリシーブロック、１つ以上のプロファイルブロック、及び１つ以上の権限ブロックを含み得る。１つ以上のポリシーブロックは、ロール及びロールに関連する権限を確立するポリシーを定義することができ、１つ以上のプロファイルブロックは、暗号ユーザＩＤ及び暗号ユーザＩＤに関連する暗号デバイスＩＤを確立することができ、１つ以上の権限ブロックは、暗号ユーザＩＤに関連するロールを定義することができる。チーム線形シーケンスは、本願で説明した線形シーケンスのインスタンスであり、同じ特性を有する。

【００８７】

チーム線形シーケンス等の線形シーケンスの初期ブロックに記録されたポリシーは、初期ブロックに記録されたポリシーが修正を許可する場合に修正され得る。初期ブロックに記録されたポリシーが修正を許可しない場合、ポリシーを修正しようとするブロックは、

複数のデバイスによって検証されない。

【００８８】

ポリシーを修正する為に、プロセッサは、１つ以上のポリシーブロックに定義されたポリシーを修正する要求と、その要求を行うユーザの暗号ＩＤを取得することができる。プロセッサは、チーム線形シーケンスから暗号ＩＤに関連する権限を決定することによって、暗号ユーザＩＤがポリシーを修正する権限を有するかどうかを確認することができる。通常、管理者のみがポリシーの修正を許可されており、プロセッサは、暗号ユーザＩＤが管理者又はユーザのロールを有するかどうかを確認することができる。暗号ユーザＩＤがユーザのロールを有する場合、プロセッサはブロックの検証を拒否することができる。暗号ユーザＩＤが許可されていると判断すると、プロセッサは、修正を指定するポリシーブロックを作成し、チーム線形シーケンスの末尾に修正を定義するポリシーブロックを付加することができる。

【００８９】

プロセッサがチームを定義すると、プロセッサは、チーム内の空間を定義することができ、この空間は、暗号化されたデータを秘密に共有し得るチームメンバーのサブセットを有している。空間メンバーシップは、チームのメンバーシップと同じであっても、又はチームのメンバーシップよりも小規模であってもよい。空間は、暗号化されたデータ及び暗号化されたデータへのアクセスを定義する仮想区画である。空間は、空間のメンバーのみが知っている暗号鍵を使用した暗号化されたデータを含み得る。

【００９０】

空間を定義する為に、プロセッサは、空間線形シーケンスを作成することによって、メンバー及び暗号化されたデータを表現することができる。効率化の為に、空間線形シーケンスは、複数の線形シーケンスに細分化され得る。例えば、空間線形シーケンスは、権限線形シーケンスと、暗号化されたデータ線形シーケンスとに細分化され得る。権限線形シーケンスは、暗号ユーザＩＤの空間内でのロールを定義できる。暗号ユーザＩＤは空間のメンバーであり、そのロールは、チームの１つ以上のポリシーブロックに定義されたポリシーと一致する。暗号化されたデータ線形シーケンスは、暗号化されたデータの少なくとも一部の追加、削除、又は修正等、暗号化されたデータに対して実行された操作を記録できる。

【００９１】

暗号化されたデータは、ファイル、電子メール、メッセージ等、複数のタイプの暗号化されたデータを含み得る。プロセッサは、暗号化されたデータの種類の夫々について線形シーケンスを作成することができる。因って、１つの暗号化されたデータ線形シーケンスを作成する代わりに、プロセッサは、ファイル用の線形シーケンス、電子メール用の線形シーケンス、及びメッセージ用の線形シーケンスを作成することができる。

【００９２】

権限用と、暗号化されたデータの種類毎に別々の線形シーケンスを作成することにより、プロセッサは、権限を計算する為に、権限ブロックと暗号化されたデータブロックを含む線形シーケンスを調べるのとは対照的に、権限に関連するデータを含むブロックの線形シーケンスだけを調べればよいので、権限の計算を高速化することができる。暗号化されたデータブロックと同じ数の権限ブロックがあると仮定すると、空間線形リストを権限線形リストと暗号化されたデータ線形リストに分割することで、プロセッサは権限の計算を２倍高速化することができる。同様に、プロセッサは、暗号化されたデータを検索する為に、暗号化されたデータブロックと権限ブロックの両方を含む線形シーケンスとは対照的に、暗号化されたデータ線形シーケンスを調べるだけでよいので、暗号化されたデータの検索をおよそ２倍高速化することができる。

【００９３】

プロセッサは、空間に関連付けられた暗号ユーザＩＤのメンバーシップを失効させることができる。メンバーシップが失効すると、暗号ユーザＩＤは、暗号ユーザＩＤのメンバーシップの失効後に、空間内で共有されるアクセス及び暗号化されたデータにアクセスす

ることを防止されなければならない。暗号ユーザＩＤが、メンバーシップが失効した後の空間に追加された暗号化されたデータにアクセスすることを防ぐ為に、プロセッサは、メンバーシップが失効した暗号ユーザＩＤが知らない暗号セッション鍵を生成し、暗号セッション鍵を用いて、失効後の空間に追加された暗号化されたデータを暗号化することが可能である。又、失効前に空間に含まれていたデータも、新たな暗号セッション鍵を用いて暗号化され得る。

【００９４】

新たな暗号セッション鍵は、以下の４つのステップを使用して計算されたＡＥＳ鍵であり得る。ステップ１で、ＡＥＳ鍵は、Ｐ‐３８４等の楕円曲線アルゴリズムを使用して計算され得る。ステップ２で、残りのデバイス群を空間線形シーケンス、例えば空間内の権限線形シーケンスから計算する。ステップ３で、空間内に残っている各デバイスの公開デバイス鍵を用いてＡＥＳ鍵を暗号化し、暗号化されたＡＥＳ鍵を空間内の各デバイスに配布する。各デバイスは自分のデバイス秘密鍵を知っている為、暗号化されたＡＥＳ鍵を復号することができる。他のデバイスはデバイスの秘密鍵を知らないので、盗聴者は暗号化されたＡＥＳ鍵を復号することができない。ステップ４で、ＡＥＳ鍵を使用して暗号化されたメッセージを空間内のデバイスに配布して、誰もがメッセージを復号できることを確実にする。

【００９５】

図１１Ｃで説明したような幾つかの例では、セッション鍵の計算は、ステップ３の前に実行される追加のステップを含み得、このステップでは、セッション鍵の計算は、空間の権限線形シーケンスのような空間の線形シーケンスの最終ブロックの暗号化ハッシュも含む。具体的には、ステップ１でＡＥＳ鍵が計算されると、最終鍵を生成する為に、ＡＥＳ鍵と最終ブロックの暗号化ハッシュを組み合わせる追加のステップが実行され得る。この組み合わせはＨＫＤＦ暗号関数を用いて計算され得るものであり、ＨＫＤＦ暗号関数は、ＡＥＳ鍵と最終ブロックの暗号化ハッシュを引数にとり、最終鍵を生成する。次に、最終鍵は、暗号化され、全てのデバイスに配布される。

【００９６】

メンバーがリストから削除されると新たなセッション鍵を計算することに加えて、新たなメンバーが追加された時に新たなセッション鍵を計算してもよく、その意図は、メンバーが参加する前に空間内で共有された暗号化されたデータにメンバーがアクセスすることを防止することである。

【００９７】

プロセッサは、図３で説明したように、チーム線形シーケンスに属する複数のチームブロックと空間線形シーケンスに属する複数の空間ブロックとの間に時間的関係を確立することによって、空間線形シーケンスにおけるチーム線形シーケンスの線形順序付けを可能にすることができる。時間的関係としては、空間ブロックの直前のチームブロックに空間ブロックがバインドされる、又は空間ブロックの直後のチームブロックに空間ブロックがバインドされる等、種々のものがあり得る。線形順序付けを確立することは、監査だけでなく、権限計算においても重要である。

【００９８】

例えば、権限を決定する為に、ブロックの追加より前の時間にチーム空間に定義された現在のポリシーを参照する必要がある。別の例では、線形シーケンスの監査を実行する時、ブロックが空間の線形シーケンスに正しく追加されたかどうかを判断する為に、チーム線形シーケンスに部分的に定義され得る現在の権限が計算される必要がある。現在のポリシーを決定する為に、空間線形シーケンス及びチーム線形シーケンス内のブロックの線形順序付けを決定する必要があり、これにより、ブロックの追加前にチームリストに記録された権限を計算できる。

【００９９】

その結果、ブロックが空間線形シーケンスに追加される度に、ブロックはチーム線形シーケンスにバインドされ、チーム線形シーケンスと空間線形シーケンスとの間の線形順序

付けが決定される。一方の空間線形シーケンス内の権限は、他方の空間線形シーケンス内の権限に影響を与えないので、２つの空間線形シーケンス間の線形順序付けを確立する必要はない。

【０１００】

図１３は、分散型台帳を使用して暗号化されたデータへのアクセスを管理する方法のフローチャートである。ステップ１３００において、プロセッサは、配置された複数のブロックを含む線形シーケンスに記録された権限によってアクセスが許可されていることを確認することによって、暗号化されたデータへのアクセスを管理でき、線形シーケンス内の初期ブロックは、ロールとロールに関連付けられた権限を指定するポリシーを定義する。

【０１０１】

ステップ１３１０において、プロセッサは、線形シーケンスに関連付けられたユーザを暗号的に識別することによって、線形シーケンスに記録された権限の有効性を保持し、従って、許可されたユーザが線形シーケンスにアクセスするのを防止し得る。ユーザ権限に関連付けられたブロックを複数のブロックに追加する前に、プロセッサは、線形シーケンスを確認して、ユーザ権限がポリシーと一致することを保証することができる。ユーザ権限がポリシーと一致することを保証すると、プロセッサは、ユーザ権限に関連するブロックを複数のブロックに追加することができる。

【０１０２】

プロセッサは、ユーザに関連付けられたユーザロール、ユーザロールに関連付けられた権限を決定することができ、ブロックに関連付けられたユーザ権限が、ユーザロールに関連付けられた権限の範囲内にあることを保証することができる。

【０１０３】

例えば、シーケンスに追加されるブロックは、ポリシーの修正を要求することができる。シーケンスにブロックを追加する前に、プロセッサは、ポリシーが検証を許可するかどうか、及びどのロールがポリシーを修正することができるかを確認することができる。例えば、ポリシーは、ポリシーは修正され得るが、管理者のみがポリシーを修正することができると述べることができる。その場合、プロセッサは、修正を要求するユーザが管理者であるか否かを確認することができる。

【０１０４】

侵害された中央サーバによるデータへの不正アクセスを防止する為に、プロセッサは、複数のユーザに関連付けられた複数のデバイスに線形シーケンスを配布することができ、複数のデバイスにおける各デバイスは、複数のユーザにおけるユーザによって暗号的に認証される。線形シーケンスにブロックを追加するかどうかの判断は、単一障害点をもたらす中央サーバによって行われるのではなく、デバイスの各々によって独立して行われ得る。

【０１０５】

プロセッサは、非対称暗号アルゴリズムを使用して生成された公開鍵であり得る暗号ユーザＩＤを使用して各ユーザを認証することができる。暗号ユーザＩＤは２０４８ビットの文字列であり得る。非対称暗号アルゴリズムは、ＲＳＡ又はＤＨであり得、公開鍵と秘密鍵の２つの鍵を生成できる。プロセッサは、非対称暗号鍵ペアの内の第１の鍵（即ち、秘密鍵）をそのユーザにのみ提供し、非対称暗号鍵ペアの内の暗号ユーザＩＤ（即ち、公開鍵）を複数のユーザにシステム全体に提供できる。プロセッサは、システム全体でユーザを識別する方法として公開鍵を使用できる。その結果、ユーザは、異なるチーム又は空間において異なる名前を想定することができ、様々な名前は、１つの暗号ユーザＩＤに結び付けられ得る。

【０１０６】

例えば、ユーザのアイデンティティを確認する為に、プロセッサはテキストメッセージを受信し、そのテキストメッセージはユーザの秘密鍵を使用して署名され得る。ユーザのアイデンティティを検証する為に、プロセッサは、ユーザの公開鍵（即ち、暗号ユーザＩＤ）を使用して署名プロセスを逆行させることができる。次に、プロセッサは、テキストメッセージと、署名プロセスを逆行させて得られたメッセージとを比較することができる

。テキストメッセージと署名プロセスを逆行させて得られたメッセージが完全に一致する場合、プロセッサはユーザのアイデンティティを検証できる。そうでない場合、プロセッサはユーザのアイデンティティを検証できない。

【０１０７】

効率化の為に、プロセッサはチームを作成することができる。チームを作成する為に、プロセッサは、複数のユーザを識別する複数の暗号ユーザＩＤを取得することができる。プロセッサは、線形シーケンスに配置された複数のブロックを含む線形シーケンスを作成することができ、線形シーケンスにおける初期ブロックは、ロール及びロールに関連付けられた権限を指定するポリシーを定義し、複数のブロックにおけるブロックは、複数のユーザにおけるユーザを識別する複数の暗号ユーザＩＤの１つの暗号ユーザＩＤに関連付けられたロールを定義する。

【０１０８】

チーム内に空間を作成する為に、プロセッサはシステム内の全ての暗号ＩＤを検索する必要はなく、チームに含まれる暗号ＩＤのみを検索するだけでよいので、ＣＰＵサイクルを節約することができる。例えば、１つのチームに１０人のユーザが含まれる場合、システム全体では数万人のユーザが含まれる為、空間の作成に使用するプロセッササイクルは約１,０００回分削減される。空間は、チームの暗号ユーザＩＤのサブセットを含み得る。空間は、空間のメンバーのみが知っている暗号鍵を用いて暗号化されたデータを含み得る。空間は、メンバー及び暗号化されたデータを表す空間線形シーケンスを含み得る。

【０１０９】

本願で説明するように、空間線形シーケンスは、効率上の理由から、２つ以上のサブシーケンスを含み得る。空間線形シーケンスは、ユーザ及び／又は管理者を追加又は削除するような、システム内の権限を修正するブロックを含む権限線形シーケンスを含み得る。暗号化されたデータ線形シーケンスは、システム内で暗号化されたデータを追加、削除、及び修正する線形シーケンスを含み得る。暗号化されたデータ線形シーケンスは、ファイル及び／又はメッセージ等の暗号化されたデータの種類に応じて、複数の線形シーケンスに更に細分化され得る。

【０１１０】

チーム線形シーケンス、空間線形シーケンス、及び暗号化されたデータは、中央サーバ等のネットワークを介して１つ以上のコンピュータから継続的に利用できるように構成されたメモリに格納され得る。その為、空間内のデバイスの殆どがオフラインである場合、空間内のデバイスは、中央サーバに暗号化されたデータを要求することができ、及び／又は空間順序シーケンスにブロックを追加することができる。

【０１１１】

（企業環境におけるグループ権限とアクセスを管理するブロックチェーンの統合）

図１４は、一実施形態により、セキュアファイルシステムが企業情報技術（ＩＴ）インフラストラクチャに如何に統合され得るかを示す。サーバ１４００は、ファイルシステム、電子メール、インスタントメッセージ等の機密データを含み得る暗号化されたデータ１４１０を格納することができる。サーバ１４００は、本願で説明するように、チーム線形シーケンス又は空間線形シーケンスを表し得るブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２を格納することもできる。ブロックチェーンは、暗号を使用してリンクされるブロックと呼ばれるレコードの成長するリストである。各ブロックは、例えば図８の８２６のような先行ブロックの暗号化ハッシュと、例えば図８の８１２、８１４、８１６、８２２、８２４のようなデータとを含む。ブロックは又、例えば、図２における２６０、２７０、２８０、２９０のようなタイムスタンプを含み得る。

【０１１２】

ブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２は、本願で説明したように、暗号ユーザＩＤに関連する権限を記録できる。ブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２は、平文で格納され得、サーバ１４００は、平文へのアクセスを権限のある要求者のみに許可することによって、ブロックチェーン１４

０２、１４０４、１４０６、１４０８、１４１２へのアクセスを制御することができる。要求者に権限を与える為に、サーバ１４００は、以下に説明するように、トークンを発行し管理することができる。

【０１１３】

システム１４２０は、企業ＩＴインフラストラクチャの一部として、顧客構内に実装され得る。システム１４２０は、アクセス制御サーバ１４３０、トークン発行元１４４０、及びユーザデバイス１４５０を含み得る。

【０１１４】

アクセス制御サーバ１４３０は、一連の企業ポリシーに基づいてユーザデバイス１４５０にパーミッションを付与又は拒否することによって、企業インフラストラクチャ上で実行されているウェブアプリケーション、サービス及び／又はファイルへのユーザデバイス１４５０のアクセスを制御できる。アクセス制御サーバ１４３０は、マイクロソフト・アクティブディレクトリ、又はアップル・オープンディレクトリ等の様々なソフトウェアを実行できる。

【０１１５】

トークン発行元１４４０は、アクセス制御サーバ１４３０、ユーザデバイス１４５０、及びサーバ１４００の間のミドルウェアとして機能することができる。トークン発行元１４４０は、ブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２の一部へのアクセスを要求するトークン要求１４６０をユーザデバイス１４５０から受信することができる。トークン要求１４６０は、要求を行うユーザに関連する暗号ユーザＩＤ、及び要求されるブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２の部分の仕様を含み得る。例えば、トークン要求１４６０は、「９ＥＤａｌｅＭＮ９ＣＵｙｌＶ７ＶＳｙＡＵＴｋｆＥＧＣ７ＭＵＤＭｋｕｇｍＸＶ　ＶｓＭ７Ｚ５ｒ０１Ｗｐｇ」等の英数字列の形態の暗号ユーザＩＤ、及びチームブロックチェーン１４０２、１４０８又は空間ブロックチェーン１４０４、１４０６、１４１２の識別を含み得る。

【０１１６】

トークン発行元１４４０は、ブロックチェーンの指定された部分にアクセスするパーミッションをユーザデバイス１４５０に与えるトークンの要求１４７０をサーバ１４００に送信することができる。トークン要求１４７０は、暗号ユーザＩＤと、トークン要求１４６０に含まれるチームブロックチェーン１４０２、１４０８又は空間ブロックチェーン１４０４、１４０６、１４１２の識別情報とを含み得る。

【０１１７】

トークン要求１４７０を受信すると、サーバ１４００は、ブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２に格納されたメンバーシップ情報に基づいて、暗号ユーザＩＤがブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２の要求された部分にアクセスする権限を有するかどうかを計算できる。例えば、暗号ユーザＩＤが空間ブロックチェーン１４０４へのアクセスを要求した場合、サーバ１４００は、暗号ユーザＩＤが空間ブロックチェーン１４０４のメンバーであるか否かを確認することができる。暗号ユーザＩＤが空間のメンバーである場合、サーバ１４００は、暗号ユーザＩＤが必要な権限を有すると判断し、トークン１４８０を発行することができる。そうでない場合、サーバ１４００は、暗号ユーザＩＤが空間ブロックチェーン１４０４にアクセスする権限を有していないと判断し、トークン１４８０を発行することを拒否することができる。

【０１１８】

トークン１４８０は、空間ブロックチェーン１４０４等のブロックチェーンの要求された部分への無制限の読み取りアクセスを付与することができる。言い換えると、サーバ１４００及び／又はトークン発行元１４４０が、任意の源からトークン１４８０を受信する度に、サーバ１４００は、上述の権限計算を行わず、トークン１４８０で指定されたブロックチェーンの部分、例えば空間ブロックチェーン１４０４へのアクセスを即座に付与する。事実上、トークン１４８０は、暗号ユーザＩＤが空間ブロックチェーン１４０４への

アクセスを要求する度に、サーバ１４００が権限を計算するという高価な計算を実行しないようにすることによって、効率的な利点を創出する。代わりに、トークン１４８０は、以下に説明するように、サーバ１４００が、トークン１４８０を単に検証するという、より安価な計算を実行することを可能にする。

【０１１９】

トークン１４８０をユーザデバイス１４５０に渡す前に、トークン発行元１４４０は、ユーザデバイス１４５０が空間ブロックチェーン１４０４に関してどのようなパーミッションを有しているかについて企業アクセス制御サーバ１４３０に確認することができる。確認を実行する為に、ユーザデバイス１４５０は、アクセス制御サーバ１４３０にチケット要求１４９０を送信することができる。チケット要求１４９０は、ユーザのログインＩＤ及びユーザのパスワード等のアクセス制御サーバのユーザ識別を含み得る。アクセス制御サーバ１４３０にユーザを識別する為に用いられるユーザのログインＩＤ及びユーザのパスワードは、サーバ１４００にユーザを識別する為に用いられる暗号ユーザＩＤとは異なるものである。

【０１２０】

ログインＩＤ及びパスワード等のユーザの識別を検証すると、アクセス制御サーバ１４３０は、企業ポリシーに従ってユーザが有するパーミッションを確認し、パーミッションを含むチケット１４９２をトークン発行元１４４０に送信することができる。例えば、企業ポリシーは、ユーザが休暇中、ユーザが電子メールにアクセスできないことを指定することができる。その結果、パーミッションは、「ユーザが会社ビル内等の特定の場所にいる時のみアクセスを許可される」、「ユーザが特定の時間帯のみアクセスを許可される」、及び／又は「ユーザデバイス１４５０が会社のネットワーク等の特定のネットワークに接続している時のみアクセスを許可される」等の種々の制限を指定できる。

【０１２１】

トークン発行元１４４０は、チケット１４９２で指定されたパーミッションをトークン１４８０に組み込んで、減衰トークンを取得することができる。チケット１４９２で指定されたパーミッションは、トークン１４８０によって付与されるパーミッションを増加させる可能性はないが、パーミッションを変わらない状態に保つか、又はトークン１４８０によって付与されるパーミッションを減衰させる、即ち、減少させるかの何れかはあり得る。

【０１２２】

更に、トークン発行元１４４０は、減衰トークンの有効期限（例えば、３分又は５分以内）、場所制限、インターネットアドレス制限等の追加の制限を、減衰トークンに追加することができる。追加の制限を減衰トークンに追加して減衰トークン１４９４を生成し、それをユーザデバイス１４５０に送信してもよい。

【０１２３】

ユーザデバイス１４５０は、チームブロックチェーン１４０２、１４０８に追加されることを要求することができる。チームに追加される為に、ユーザデバイス１４５０はアクセス制御サーバ１４３０に自身を認証することができ、アクセス制御サーバ１４３０は、ユーザを認証するチケット１４９２をトークン発行元１４４０に送信できる。更に、トークン発行元１４４０は、サーバ１４００に、ユーザに関連する暗号ユーザＩＤについて、チームブロックチェーン１４０２、１４０８に格納された権限を計算するよう求めることによって、ユーザをサーバ１４００に認証することができる。アクセス制御サーバ１４３０とサーバ１４００の両方がユーザを認証した場合、ユーザをチームブロックチェーン１４０２、１４０８に追加することができる。

【０１２４】

アクセス制御サーバ１４３０が敵対者によって制御されたとしても、アクセス制御サーバ１４３０はトークン１４８０によって付与されたパーミッションを増やすことができないので、敵対者にあるのは依然として、ブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２内にありトークン１４８０によって付与された権限のみである。敵

対者がトークン発行元１４４０を制御する場合、敵対者は、トークン発行元１４４０がサーバから受信したチームの為のトークン１４８０のみを制御することができる。何れの場合も、敵対者は、平文データのみを読むことができ、暗号化されたデータ１４１０を読むことはできないことになる。更に、敵対者は、平文データを修正することができない。

【０１２５】

トークン発行元１４４０は、アクセス制御サーバ１４３０がどの程度信頼されているかの表示を受信することができる。アクセス制御サーバ１４３０が信頼されていない場合、アクセス制御サーバ１４３０を介さずにトークン１４９４を発行することができ、又はユーザをチーム１、２に追加することができる。

【０１２６】

図１５は、別の実施形態による、セキュアファイルシステムが企業ＩＴインフラストラクチャにどのように統合され得るかを示す。ブロックチェーン１５０２、１５０４、１５０６、１５１２へのアクセスは、図１４のアクセス制御サーバ１４３０を介さずに管理され得る。アクセス制御サーバ１４３０によって実装される企業ポリシーは、ブロックチェーン１５２０に記録され得るか、又はファクトデータベース１５３０の一部であり得る。ブロックチェーン１５２０及びファクトデータベース１５３０は、チーム１、２又は空間１、２から独立して存在することができる。サーバ１５００及びトークン発行元１５４０は、企業のＩＴインフラストラクチャの一部であり得、及び／又はクラウドサービスとして提供され得る。アクセス制御サーバ１４３０を削除することにより、多くの潜在的なセキュリティ問題が減少し、アクセス制御サーバ１４３０が信頼できない場合でも、システムを機能させることができる。

【０１２７】

ユーザデバイス１５５０がトークン要求１５６０を使用してトークンを要求すると、トークン発行元は、トークン要求１５６０をサーバ１５００に転送することができる。トークン要求１５６０は、暗号ユーザＩＤ、及びアクセスが要求されているブロックチェーン１５０２、１５０４、１５０６、１５０８、１５１２の部分の識別を含み得る。例えば、ブロックチェーンの部分の識別は、チームブロックチェーン１５０８を識別することができる。

【０１２８】

サーバ１５００は、チームブロックチェーン１５０８に記録された暗号ユーザＩＤの権限を計算し、暗号ユーザＩＤがチームのメンバーであるか否かを判断することができる。暗号ユーザＩＤがチームのメンバーでない場合、サーバ１５００は、トークン発行元１５４０にトークンを送信することを拒否することができる。暗号ユーザＩＤがチームのメンバーである場合、サーバ１５００は、ファクトデータベース１５３０及び／又は会社ポリシーブロックチェーン１５２０を確認して、会社ポリシーが暗号ユーザＩＤのアクセスに何らかの制限を加えているかどうかを判断することができる。この場合、図１４で説明したように、別々の認証を必要とするのとは対照的に、単一の暗号ユーザＩＤを使用して、サーバ１５００に関連するパーミッションと会社ポリシーに関連するパーミッションの両方を確認することができる。

【０１２９】

サーバ１５００は、トークン発行元１５４０にメッセージ１５８０を送信することができる。メッセージ１５８０は、チームブロックチェーン１５０８への無制限の読み取りアクセス、及び企業ポリシーに関連付けられたパーミッションを付与するトークンを含み得る。トークン発行元１５４０は、トークン及び会社ポリシーに関連付けられたパーミッションを組み合わせることによって、減衰トークン１５９４を作成し、減衰トークン１５９４をユーザデバイス１５５０に転送することができる。

【０１３０】

ユーザデバイス１５５０は、チームブロックチェーン１５０８にアクセスするパーミッションを第三者に付与する為に、減衰トークン１５９４を更に減衰させたい場合がある。そうする為に、ユーザデバイス１５５０は、減衰トークン１５９４と、減衰トークンに課

された追加のパーミッション、例えば時間的パーミッションとを含む要求をトークン発行元１５４０に送信することができる。トークン発行元１５４０は、追加のパーミッションを減衰トークン１５９４に組み込み、ユーザデバイス１５５０に送信する為の新たなトークンを発行し、ユーザデバイス１５５０はこれを第３者に転送できる。

【０１３１】

（クロック）

図１６Ａは、ブロックチェーンを使用してどのようにクロックが実装され得るかを示す。図１５で論じた時間的パーミッションを組み込む為に、プロセッサは、例えば図２で説明したように、ブロックチェーン１６１０、１６１５、１６２０、１６２５内の各ブロック１６３５、１６４５（簡潔にする為、２つのみにラベルを付けている）がタイムスタンプされ、ブロック１６３５、１６４５内の任意のタイムスタンプ１６３０、１６４０が常に増加するような、常に増加するクロックを作成し得る。

【０１３２】

トークンにおける時間的なパーミッションは、例えば、トークンはタイムスタンプ１６３０の５分後に有効である、というようにタイムスタンプで表現され得る。その為、タイムスタンプ１６４０がタイムスタンプ１６３０から５分を超えて経過している場合、トークン保有者はブロック１６４５への読み取りアクセスを許可されない。ブロック１６３５、１６４５は、チーム及びシーケンス内で順序付けされ得るが、２つの異なるチーム又は２つの異なるシーケンス間で順序付けることができない場合がある。タイムスタンプ１６３０、１６４０は、夫々、ブロック１６３５、１６４５のヘッダ内に配置され得る。

【０１３３】

代わりに、又は付加的に、クロックは、ブロックチェーン１６００を使用して実装され得る。クロックブロックチェーン１６００は、夫々が１秒、１分、５分、半時間、１時間等のクロックの刻みを表すブロック１６５０、１６６０、１６７０等を含み得る。ブロック１６５０、１６６０、１６７０の頻度は、企業の計算機資源に関連し得る。計算資源が高いほど、ブロック１６５０、１６６０、１６７０の頻度は高くなり、計算資源が低いほど、ブロック１６５０、１６６０、１６７０の頻度は低くなる。

【０１３４】

チームブロックチェーン１６１０、１６１５の各ブロック１６２５、１６３５、１６４５（簡潔にする為に３つのみラベルを付けている）は、クロックブロックチェーン１６００の対応するブロック１６５０、１６６０、１６７０に対するバインディング１６２８、１６３８、１６４８を夫々有し得る。例えば、チームブロック１６２５は、ブロック１６５０に対するバインディング１６２８を有し、これは、チームブロック１６２５が、ブロック１６５０の作成後且つブロック１６６０の作成前に作成されたことを意味する。

【０１３５】

空間ブロックチェーン１６２０、１６２５のブロック１６８０、１６９０（簡潔にする為に２つのみラベルを付けている）は、チームブロックチェーン１６１０、１６１５への対応するブロック１６３５、１６２５に対して、夫々バインディング１６８２、１６９２を有し得る。例えば、バインディング１６８２は、ブロック１６８０がブロック１６３５の後且つブロック１６４５の前に作成されたことを示す。

【０１３６】

その結果、空間ブロックチェーン１６２０、１６２５の各ブロック１６８０、１６９０は、チームブロックチェーン１６１０、１６１５の対応するブロックにバインドされ、クロックブロックチェーン１６００のブロック１６５０、１６６０、１６７０に間接的にバインドされる。例えば、図１６Ａに示すバインディングに基づき、ブロック１６９０はブロック１６５０の後且つブロック１６６０の前に作成される。

【０１３７】

トークンにおける時間的パーミッションは、クロックブロックチェーン１６００、ブロック１６５０、１６６０、１６７０の観点から、又はウォールクロックの観点から表現され得る。クロックブロックチェーン１６００は、クロックブロックチェーン１６００のブ

ロック１６５０、１６６０、１６７０が常に増加しているという制約のもと、ウォールクロックに対応し得る。

【０１３８】

例えば、時間的パーミッションがウォールクロックの観点から定式化される場合、時間的パーミッションは、トークンが２０２０年１２月１日まで有効であると述べることができる。指定された日付の後にタイムスタンプを有する最初のブロック１６５０、１６６０、１６７０は、トークンが、それ以降はもはや有効でなくなる時間を指定する。このような時刻を指定する最初のブロックにバインドされている全てのブロックに、トークン保有者はアクセスできない。

【０１３９】

別の例では、時間的パーミッションがブロック１６５０、１６６０、１６７０の観点から定式化される場合、時間的パーミッションは、ブロック１６５０の後に１時間半の間トークンが有効であると述べることができる。指定された時間後にタイムスタンプを有する最初のブロック１６６０、１６７０は、トークンがそれ以降はもはや有効でなくなる時間を指定する。そのような時間を指定する最初のブロックにバインドされた全てのブロックに、トークン保有者はアクセスできない。

【０１４０】

図１６Ｂはクロックブロックチェーンの内容を示す。クロックブロックチェーン１６００のブロック１６５０、１６６０、１６７０は、ウォールクロックフィールド１６５２、１６６２、１６７２、シーケンスクロックフィールド１６５４、１６６４、１６７４及び暗号化ハッシュツリーフィールド１６５６、１６６６、１６７６のルートを含む幾つかのフィールドを含み得る。

【０１４１】

ウォールクロックフィールド１６５２、１６６２、１６７２は、クロックブロックチェーン１６００を格納するサーバのクロックによって示される時間のレコードであり得る。ウォールクロックフィールド１６５２、１６６２、１６７２は常に増加している必要はないので、フィールド１６５２、１６６２は同じ値を有し得るか、又はフィールド１６７２はフィールド１６５２によって示される時間より前の時間を示し得る。

【０１４２】

シーケンスクロック１６５４、１６６４、１６７４は常に増加する。シーケンスクロックは、現在時刻を測定し合意している複数のサーバによって示され得る。複数のサーバは冗長性を提供し、その結果、１つのサーバが故障した場合、クロックブロックチェーン１６００は時間の測定を継続できる。合意された現在時刻は、以前の現在時刻よりも大きいという特性を有する。現在時刻は、複数のサーバによって測定された最新の時刻であってもよいし、現在時刻が前に合意された現在時刻よりも大きい限り、殆どのサーバが合意する時刻であってもよい。

【０１４３】

暗号化ハッシュツリーフィールド１６５６、１６６６、１６７６のルートはメルクルルートであり得、ブロック１６５０、１６６０、１６７０にバインドされているブロックチェーン内の全ての最新のブロックの暗号化ハッシュを含み得る。例えば、暗号化ハッシュツリーフィールド１６６２は、ブロック１６３５とブロック１６８５の暗号化ハッシュを含み得る。全ての最新のブロックのリストではなく暗号化ハッシュツリーのルートを提供する理由は帯域幅及び格納リソースを保存することであり得るが、それは、ルートを通信及び格納することが、全てのブロックのリストを通信及び格納することよりも低コストであるからである。別の理由は、ルートのみを提供することによって、全てのブロックのリストを秘密にし、そこから、幾つかの追加情報と共に、全てのブロックのリストを計算できるようにすることであり得る。

【０１４４】

図１７は暗号ツリーを示す。エンドポイントは、図１６のクロックブロックチェーン１６００内のブロックにバインドされたブロックチェーン内の最新のブロックの全てである

ブロック１７００、１７１０、１７２０、１７３０を表している。暗号ツリー１７４０は、各ノードにおいて、子ノードのＳＨＡ等の暗号化ハッシュを計算することにより構築される。例えば、ノード１７５０はブロック１７１０の暗号化ハッシュを計算することによって計算され、ノード１７９２はノード１７７０、１７８０の暗号化ハッシュを計算することによって計算される。最後に、ルート１７９０はノード１７９２、１７９４の暗号化ハッシュを計算することによって計算される。

【０１４５】

ルート１７９０はクロックブロックチェーン１６００に格納され得る。ルート１７９０の値は、暗号ツリー１７４０を生成したブロック１７００、１７１０、１７２０、１７３０のシーケンスに固有のものである。ルート１７９０のような単一の値をクロックブロックチェーン１６００に格納することは、全てのブロック１７００、１７１０、１７２０、１７３０の値を格納することと比較して、帯域幅及び格納領域を保存することができる。更に、ルート１７９０を格納することは、暗号ツリー１７４０を構築する際に使用される全てのブロック１７００、１７１０、１７２０、１７３０を開示しない。

【０１４６】

ブロックが暗号ツリー１７４０の一部であるかどうかを確認する為に、暗号ツリー１７４０の全ての要素のサブセットのみを供給する必要がある。例えば、Ｎ個のエンドポイント、即ち暗号ツリー１７４０を構築する際に使用されるＮ個のブロックがある場合、ルート１７９０が特定の要素を含むかどうかを確認する為に、ｌｏｇ（Ｎ）個の要素のみを供給する必要がある。

【０１４７】

より具体的な例では、ブロック１７１０がルート１７９０のメンバーであるかどうかを確認する為には、ブロック１７１０及びノード１７６０、１７９２のみを供給すればよい。供給されると、ノード１７５０の値はブロック１７１０の値から計算され得る。ノード１７９４の値は、ノード１７５０、１７６０の値から計算され得るが、例えば、ＳＨＡ（ノード１７６０、ノード１７５０）を計算することによって計算され得る。ルートノードはノード１７９４及び１７９２の値から計算され得る。このように計算されたルートノードが、クロックブロックチェーン１６００に格納されているルート１７９０と一致すれば、暗号ツリー１７４０におけるブロック１７１０のメンバーシップが確認され得る。

【０１４８】

（トークン）

図１８はトークンの構造を示している。トークン１８００は、図１４のサーバ１４００、図１５の１５００が、図１４の権限定義ブロックチェーン１４０２、１４０４、１４０６、１４０８、１４１２、図１５の１５０２、１５０４、１５０６、１５０８、１５１２を格納するような第１の権限源によって付与された、図１４のトークン１４８０、図１５の１５８０であり得る。

【０１４９】

トークン１８００は、秘密のルート鍵を識別する鍵識別子（ＩＤ）１８１０と、パーミッション１８２０と、パーミッション１８２０と秘密のルート鍵との暗号化ハッシュ１８３０とを含み得る。秘密のルート鍵は、サーバ１４００、１５００及び／又は図１４のトークン発行元１４４０、図１５のトークン発行元１５４０に知られ得る。鍵識別子１８１０を使用して、サーバ１４００、１５００及び／又はトークン発行元１４４０、１５４０は、秘密のルート鍵を取得することができる。暗号化ハッシュ１８３０はＨＭＡＣであり得、鍵付きハッシュメッセージ認証コード又はハッシュベースメッセージ認証コードの何れかとして拡張される場合もある。ＨＭＡＣは、任意のＭＡＣと同様に、メッセージのデータの完全性と真正性の両方を同時に検証する為に使用され得る。ＨＭＡＣの計算には、ＳＨＡ‐２５６やＳＨＡ‐３等、任意の暗号化ハッシュ関数を使用してもよい。全ての暗号化ハッシュと同様に、暗号化ハッシュ１８３０は可逆的ではなく、それは、暗号化ハッシュ１８３０の出力が与えられて、暗号化ハッシュへの入力を割り出すことは計算上実行可能でないことを意味する。

【０１５０】
第２のパーミッション１８５０が追加されたトークン１８００は変化して、減衰トークン１８４０になり得る。減衰トークン１８４０は、図１４のトークン１４９４、図１５のトークン１５９４であり得る。第２のパーミッション１８５０は、図１４のアクセス制御サーバ１４３０又は図１５のブロックチェーン１５２０に記録された企業ポリシー等の第２の権限源によって付与され得る。トークン１８００は、制約又は注意事項とも呼ばれる、任意の数のパーミッションを含み得る。
【０１５１】
減衰トークン１８４０を得る為に、サーバ１４００、１５００は、トークン１８４０に第２のパーミッション１８５０を追加し、第１の暗号化ハッシュ１８３０と第２のパーミッション１８５０との第２の暗号化ハッシュ１８６０を計算し、トークンから第１の暗号化ハッシュ１８３０を除去し、それにより減衰トークン１８４０を得ることができる。
【０１５２】
第２のパーミッション１８５０は、第１のパーミッション１８２０を減少させるのみと解釈される。第１のトークンから暗号化ハッシュ１８３０を除去することによって、第１のパーミッション１８２０が第２のパーミッション１８５０によって制限されるトークン１８４０が作成される。暗号化ハッシュ１８６０は可逆ではないので、攻撃者は暗号化ハッシュ１８３０を推測することができず、トークン１８４０は安全である。その結果、最大の権限は、１つのパーミッション１８２０のみを有する元のトークン１８００によって付与される。
【０１５３】
例えば、第１のパーミッション１８２０は、暗号ユーザＩＤが、ブロックチェーン１４０２、１４０４、１４０８、１４１２、１５０２、１５０４、１５０８、１５１２に関連するメタデータを読む為のアクセスを付与されていることを指定することができる。第２のパーミッション１８５０は、暗号ユーザＩＤがアクセス権を有するチームブロックチェーン１４０２、１４０８、１５０２、１５０８等のチームブロックチェーンを指定することができる。追加のパーミッションを減衰トークン１８４０に追加して、更に減衰トークン１８７０を作成することができる。
【０１５４】
一実施形態では、減衰トークン１８７０はユーザデバイスによる要求に応じて生成され得る。例えば、第３のパーミッション１８８０は、暗号ユーザＩＤがアクセスを有し得るチームブロックチェーン１４０２、１４０８、１５０２、１５０８内の空間ブロックチェーン１４０４、１４０６、１４１２、１５０４、１５０６、１５１２を指定することができる。第３の暗号化ハッシュ１８９０を減衰トークン１８７０に含めてもよく、又は暗号化ハッシュ１８９０は、暗号化ハッシュ１８６０と第３のパーミッション１８８０との暗号化ハッシュであり得る。暗号化ハッシュ１８６０は、トークン１８７０から除去することができる。
【０１５５】
別の実施形態では、ユーザデバイスは、減衰トークン１８７０を第三者に付与することができる。例えば、第三者は、独自のビデオ処理アルゴリズムを有し得、減衰トークン１８７０は、ユーザデバイスがアクセス権を有するビデオへのアクセスを付与することができる。第三者にアクセスを付与する理由は、第三者がユーザデバイスよりも高速なネットワーク接続にアクセスできることであり得る。減衰トークン１８７０は、減衰トークン１８７０がアクセスを付与するビデオを指定する第３のパーミッション１８８０を含み得る。
【０１５６】
ビデオを要求する為に、第三者は、減衰トークン１８７０を、サーバ１４００、１５００及び／又はトークン発行元１４４０、１５４０に送信することができる。サーバは、以下に説明するように、秘密ルート鍵を使用してトークンを検証することができ、ビデオへのアクセスを第三者に付与することができる。
【０１５７】

図１９は、リプレイ攻撃を防止するトークンを示す。攻撃者がトークンを取得し、トークンのコピーを図１４のトークン発行元１４４０、図１５の１５４０、及び／又は図１４のサーバ１４００、図１５の１５００に提供することによってリプレイ攻撃を行うことを防止する為に、図１８の減衰トークン１８４０、１８７０は、減衰トークン１９４０、１９７０を取得する単回使用パーミッション１９００、１９１０を含み得る。暗号化ハッシュ１９２０は、図１８の暗号化ハッシュ１８６０と第３のパーミッション１９００との暗号化ハッシュであり得、暗号化ハッシュ１９３０は、図１８の暗号化ハッシュ１８９０と第４のパーミッション１９１０との暗号化ハッシュであり得る。トークン１９４０、１９７０を要求に応じて傍受した場合、１つの要求を満たした後のトークン１９４０、１９７０は有効ではないので、トークン１９４０、１９７０を再生しても攻撃者にアクセス権を付与することはない。

【０１５８】

（リカバリー鍵）

図２０は、リカバリー鍵がどのように使用され得るかを示す。ブロックチェーン２０００はブロック２０１０を含み得る。ブロック２０１０はブロックチェーン２０００の初期ブロックであり得るか、又はブロックチェーン２０００の他のブロックであり得る。ブロック２０１０は複数のイベント２０１２、２０１４、２０１６を含み得る。イベント２０１２や２０１４のようなブロックチェーン２０００のイベントの殆どは、アリス等の、イベントを入力するユーザの公開鍵によって署名されている。しかしながら、イベント２０１６のようなイベントは、リカバリー鍵２０５０によって署名され得る。

【０１５９】

リカバリー鍵２０５０は、権限とポリシーの計算全体を回避する特別な鍵である。ポリシーがイベント２０１６を認可するかをプロセッサが判断する前であったとしても、プロセッサは、イベントがリカバリー鍵２０５０によって署名されているかどうかを判断することができる。イベントがリカバリー鍵２０５０で署名されている場合、プロセッサは、権限及びポリシーに関係なくイベントが許可されると判断する。このように、リカバリー鍵２０５０は、権限及びポリシー全体を上書きする。

【０１６０】

リカバリー鍵２０５０は厳重に保護される必要がある。その結果、リカバリー鍵２０５０は、３０個の部分のような複数の部分２０５２、２０５４、２０５６（簡潔にする為に３つのみにラベルを付けている）に分割され得る。異なる部分２０５２、２０５４、２０５６は、夫々が異なる操作手順を有する異なるＨＳＭのような異なる安全な場所に配置され得る。異なる部分２０５２、２０５４、２０５６は暗号化され得、或る人は、鍵部分２０５２、２０５４、２０５６をファイルに格納する為のパスワードを有し得、別の人は、鍵部分２０５２、２０５４、２０５６をファイルから取り出す為のパスワードを有し得る。リカバリー鍵２０５０の組み立ては、リカバリー鍵２０５０の２０５２、２０５４、２０５６の部分を有するデバイス及びユーザの全て、又は少なくとも過半数の参加を必要とする可能性があり、それにより、リカバリー鍵２０５０の偶発的な使用を保護することが可能である。

【０１６１】

リカバリー鍵２０５０の存在及び使用は任意であり得る。リカバリー鍵は全てゼロ等の所定値に設定され得、それは、リカバリー鍵２０５０が生成されておらず存在しないことをプロセッサに示す。プロセッサは、その所定値に設定されたリカバリー鍵２０５０によって署名された任意のイベントを無視してもよい。

【０１６２】

（分割鍵）

図２１は、ユーザデバイスが侵害された時に、暗号化されたデータへの攻撃を制限する分割鍵システムを示す。ユーザデバイス２１００は、本願で説明したようなチャネルセッション鍵２１２０のような複数の鍵と、分割鍵２１３０とを用いて暗号化されたデータ２１１０を格納することができる。分割鍵２１３０は、少なくとも２つの部分２１３２、２

１３４に分離され得る。第１の鍵部分２１３２はサーバ２１４０に格納され得るのに対し、第２の鍵部分２１３４はユーザデバイス２１００に格納され得る。

【０１６３】

データを暗号化する為に、ユーザデバイス２１００は、サーバ２１４０に第１の鍵部分２１３２を要求することができる。第１の鍵部分２１３２を受信すると、ユーザデバイス２１００は、第１の鍵部分２１３２と第２の鍵部分２１３４との組み合わせである鍵導出関数（ＫＤＦ）を計算し、データを暗号化する為の分割鍵２１３０を得ることができる。同様に、暗号化されたデータ２１１０を復号する為に、ユーザデバイス２１００は、サーバ２１４０に第１の鍵部分２１３２を要求し、第１の鍵部分２１３２と第２の鍵部分２１３４との組み合わせであるＫＤＦを用いて分割鍵２１３０を計算できる。

【０１６４】

ユーザデバイス２１００が分割鍵２１３０を計算してしまうと、ユーザデバイス２１００は、所定のルールに基づいて第１の鍵部分２１３２を失することができる。所定のルールは、ユーザデバイス２１００がサスペンド及び／又はリブートされる毎に、第１の鍵部分２１３２を失すること、分割鍵２１３０を使用して３つのファイルが開かれる毎に、第１の鍵部分２１３２を失すること、鍵導出関数に関連するアプリケーションが閉じられる毎に、第１の鍵部分２１３２を失すること、ユーザデバイス２１００の地理位置及び／又はＩＰアドレスが所定の空間外になる毎に、第１の鍵部分２１３２を失すること等と述べることができる。

【０１６５】

ユーザデバイス２１００は、所定のルールに基づいて第１の鍵部分２１３２を失するので、サーバ２１４０は、ユーザデバイス２１００へのアクセスを取り消すことができる。例えば、サーバ２１４０が、攻撃者によって盗まれる及び／又はハッキングされる等してユーザデバイス２１００が危険に曝されたことを通知された場合、サーバ２１４０は、ユーザデバイス２１００からの第１の鍵部分２１３２の要求を拒否すべきことを記録できる。次回ユーザデバイス２１００が第１の鍵部分２１３２を要求した時、サーバ２１４０は、第１の鍵部分２１３２の提供を拒否することができる。その結果、暗号化されたデータ２１１０は、ユーザデバイス２１００上で暗号化されたままであり、攻撃者は利用できない。

【０１６６】

第１の鍵部分２１３２のセキュリティを高める為に、サーバ２１４０は、第１の鍵部分２１３２を提供する前に多要素認証を要求することができる。例えば、サーバ２１４０は、第２のデバイス２１５０がサーバ２１４０に認証を提供することを要求することができる。サーバ２１４０が第２のデバイス２１５０から認証を受信すると、サーバ２１４０は、第１の鍵部分２１３２をユーザデバイス２１００に提供することができる。

【０１６７】

（システム更新）

図２２は、ブロックチェーンのセマンティクスの解釈に対する更新を示す。ブロックチェーン２２００は、ブロック２２１０、２２２０、２２３０等を含み得る。ブロックチェーン２２００のセマンティクスの解釈の為のルールは、ブロック２２１０、２２２０、２２３０にコード化され得るものであり、Ｒｕｓｔ、Ｐｙｔｈｏｎ、Ｇｏ、又は独自の言語等のプログラミング言語のソースコードとして表され得る。ブロックチェーン２２００のセマンティクスの更新解釈（権限、ポリシー等を含む）は、ブロックチェーン２２００において起こり得る。

【０１６８】

例えば、ブロック２２１０は、ブロック２２３０を解釈する方法を更新するソースコードを、ブロック２２１０に続いて含み得る。ブロックチェーン２２００にセマンティクスを入れることによって、複数のクライアントに亘り、異なるブロックチェーンを含むシステムの全てのインスタンスは、ブロック２２１０を受信した時に、ブロックチェーンのセマンティクスを解釈する為のローカルルールを更新することができる。従って、ブロック

２２１０に先行するブロック２２２０は、第１のルールのセットに従って解釈されるのに対し、ブロック２２１０に後続するブロック２２３０は、ブロック２２１０によって確立された第２のルールのセットに従って解釈される。

【０１６９】

例えば、ブロックチェーン２２００のセマンティクスを解釈する為のルールは、レース条件がどのように解決されるかを支配し得る。そのようなルールは、図５のシステムポリシー５３０、図５のユーザポリシー５４０、又は図５のアプリケーションポリシー５５０で指定され得る。その結果、ブロック２２１０は、システム５３０、ユーザ５４０、及び／又はアプリケーション５５０のポリシーを更新することができる。

【０１７０】

ブロックチェーン２２００のセマンティクスを解釈する為のルールにバグを招くことを防ぐ為に、ポリシーエンジン５３０、５４０、５５０はフォーマル検証され得る。フォーマル検証とは、数学の形式的方法を用いて、特定の形式的仕様又は特性に関して、ポリシー５３０、５４０、５５０の正しさを証明する行為である。フォーマル検証により、ポリシー５３０、５４０、５５０にバグがないことを保証することができる。

【０１７１】

（フローチャート）

図２３は、認可クレデンシャルを提供するトークンを生成する方法のフローチャートである。ステップ２３００において、プロセッサは、ブロックをブロックチェーンの末尾に付加する際のユーザの権限を定義するブロックを作成することによって、複数のブロックを含むブロックチェーンを作成することができる。ブロックは、ユーザを識別する暗号ユーザＩＤ、及び暗号ユーザＩＤに関連付けられた権限を含み得る。権限は、ブロックチェーン上で実行する為に暗号ユーザＩＤに関連付けられた少なくとも１つの操作を定義することができる。例えば、操作は、ブロックチェーン及び／又はブロックチェーンに関連する暗号化されたデータへの読み取りアクセス、書き込みアクセス、及び／又は読み取り／書き込みアクセスを含み得る。ブロックチェーンは暗号化されていなくてもよい。

【０１７２】

ステップ２３１０において、プロセッサは、ブロックチェーンにアクセスする要求を要求元デバイスから受信することができる。要求は、要求を行うユーザに関連付けられた暗号ユーザＩＤを含み得、暗号ユーザＩＤは、特定の操作を実行する為にブロックチェーンによって認可され得る。要求元デバイスはユーザデバイスであり得る。

【０１７３】

ステップ２３２０において、プロセッサは、ブロックチェーンを初期ブロックから最終ブロックまで確認することを含めてブロックチェーンに記録された権限を計算することによって、要求を行うユーザがブロックチェーンにアクセスする権限を有するかどうかを判断することができる。権限計算を実行することは高価になり得る。

【０１７４】

ステップ２３３０において、プロセッサは、要求を行うユーザがブロックチェーンにアクセスする権限を有すると判断した場合に、要求を行うユーザにブロックチェーンへのアクセスを付与するトークンを生成することができる。トークンは、プロセッサが、例えば、ブロックチェーン上の権限を計算すること、又はユーザのパスワードを確認すること等の高価な操作を行ったことを証明する証明書であり得る。従って、プロセッサが次回トークンを受信する時、プロセッサは高価な操作を行う必要がなく、代わりにトークンを確認することができ、これは、例えば、ブロックチェーンにおける権限を計算するよりも安価な操作となり得る。ステップ２３４０において、プロセッサは、トークンを要求元デバイスに送信することができる。

【０１７５】

トークンは、本願で説明するように、メッセージと、２つの引数、即ちメッセージと秘密のルート鍵とを取るＨＭＡＣとを含み得る。例えば、メッセージは、ブロックチェーンにアクセスすることを許可されたユーザを識別するパーミッションであり得る。

【０１７６】
一実施形態では、ユーザがブロックチェーンを読み取りたい場合、ユーザは、暗号ユーザＩＤを提供し、ユーザが読み取りたいチーム及び／又は空間等のブロックチェーンの一部を特定する署名付きメッセージを送信することができる。ブロックチェーンにアクセスするプロセッサは、暗号ユーザＩＤに関連付けられた公開鍵、署名、及びファクトデータベースを確認し、暗号ユーザＩＤが要求されたチーム及び／又は空間にアクセスするパーミッションを有していることを確認できる。プロセッサが確認を行わない場合、プロセッサは、ユーザへのアクセスを拒否することができる。プロセッサは、ユーザがチーム及び／又は空間にアクセスする権限を有することを確認した場合、暗号化ユーザと、要求されたチーム及び／又は空間の表示とを含むトークンを作成することができる。トークンは、暗号ユーザＩＤに読み取りパーミッションを付与することができる。

【０１７７】
トークンを生成する為に、プロセッサは、秘密のルート鍵を識別する鍵識別子を作成することができる。鍵識別子は、サーバに関連付けられたデータベースへの指数として実装され得る。秘密ルート鍵は、公開鍵暗号化又は秘密鍵暗号化を使用して更に保護され得る。プロセッサは、ユーザＩＤ又はチーム及び／又は空間ＩＤ等のトークンによって付与されるブロックチェーンへのパーミッションを作成することができる。更に、プロセッサは、秘密のルート鍵とパーミッションとの暗号化ハッシュを作成することができる。暗号化ハッシュはＨＭＡＣであり得る。プロセッサは、鍵識別子、パーミッション及び暗号化ハッシュをトークンに追加することができる。トークンは、チーム及び／又は空間への読み取りパーミッションを与えることができる。

【０１７８】
要求を行うユーザがブロックチェーンとして機能する権限を有するかどうかを判断する為に、プロセッサは、ブロックチェーンに記録された権限を計算せずに、要求がリカバリー鍵で署名されているかどうかを判断することができる。本願で説明するように、リカバリー鍵は、ブロックチェーンに記録された権限を上書きできる。プロセッサは、要求がリカバリー鍵で署名されていると判断した時に、要求を行うユーザにブロックチェーンへの無制限アクセスを付与する第２のトークンを生成することができる。

【０１７９】
攻撃者がリカバリー鍵にアクセスすることを防止する為に、リカバリー鍵を複数の部分に分離し、各部分を異なる秘密の暗号化キーを使用して暗号化し、各暗号化部分をＨＳＭ等の複数のデバイス間で分散させることができる。

【０１８０】
攻撃者がエンドポイント、例えばユーザデバイスを侵害することによってシステムにアクセスすることを防止する為に、暗号化されたデータは、追加的に分割鍵で暗号化され得る。分割鍵は、サーバに格納される第１の部分と、ユーザデバイスに格納される第２の部分とに分離され得る。暗号化されたデータを復号する為に、ユーザデバイスは、分割鍵の第１の部分、即ち、第１の暗号鍵をサーバに要求する必要がある。第１の暗号鍵の要求を受信すると、サーバは、ユーザデバイスが第１の暗号鍵の受信を許可されているかどうかを判断することができる。

【０１８１】
例えば、ユーザデバイスは盗難品として報告され、その結果、第１の暗号鍵の受信が許可されないことがある。サーバは、ユーザデバイスが第１の暗号鍵を受信することを許可されていないと判断した場合、第１の暗号鍵の送信を拒否することができる。ユーザデバイスが暗号鍵を受信した場合、ユーザデバイスは、ＨＭＡＣ等の第１暗号鍵と第２暗号鍵の組み合わせである鍵導出関数を計算し、暗号化されたデータの復号に使用できる鍵を取得することができる。

【０１８２】
ブロックチェーンに格納された権限及び／又はポリシーの計算に対するソフトウェア更新は、それ自体がブロックチェーンに格納され得る。サーバは、ブロックチェーンのセマ

ンティクスの解釈に関する更新を受信することができる。サーバは、ブロックチェーン内に更新を格納することによって、複数のユーザデバイスに亘るブロックチェーンのセマンティクスの解釈の一貫性を確保できる。

【０１８３】

図２４は、減衰トークンを作成する方法のフローチャートである。ステップ２４００において、プロセッサは、ブロックチェーンへのアクセスを付与するトークンを取得できる。ブロックチェーンへのアクセスは、ブロックチェーンによって定義される第１の権限源によって許可され得る。言い換えると、第１の権限源は、権限及びポリシーを記録するブロックチェーン自体であり得る。トークンは、秘密のルート鍵を識別する鍵識別子と、第１の権限源によって許可されたブロックチェーンへのアクセスの第１のパーミッションと、秘密のルート鍵とパーミッションとの第１の暗号化ハッシュとを含み得る。ブロックチェーンへのアクセスは、要求者が暗号化されたデータへの読み取りアクセスを有するか、又は要求されたチーム及び／若しくは空間におけるメンバーシップを有するかどうかに基づいて許可され得る。

【０１８４】

第１のパーミッションは、第１のパーミッションが付与される暗号ユーザＩＤ、及びチームや空間等、暗号ユーザＩＤがアクセス権を有するブロックチェーンの少なくとも一部の識別を含み得る。第１のパーミッションは、暗号ユーザＩＤによって実行されることが許可された操作を含み得るか、又は、許可された操作は含まれず、操作が読み取り専用であると想定され得る。

【０１８５】

ステップ２４１０において、プロセッサは、要求元デバイスからブロックチェーンにアクセスする要求を、要求元デバイスに関連するアクセスを制限する第２の権限源からの第２のパーミッションと共に受信し得る。第２の権限源は、アクティブディレクトリ等のアクセス制御サーバであり得る。アクセス制御サーバは、企業ＩＴシステムの一部であり得る。

【０１８６】

第２のパーミッションは、時間制限又は地理的位置の制限を含み得る。例えば、第２のパーミッションは、軸が許可される時間ウィンドウ、又はアクセスが許可されるジオロケーションを指定することができる。より具体的な例では、ユーザがビルから出た場合、ブロックチェーンへのアクセスは取り消され得る。第２のパーミッションは又、インターネットアドレス制限を含み得、例えば、要求元デバイスのインターネットプロトコル（ＩＰ）アドレスが指定されたＩＰアドレス範囲内にある限り、要求元デバイスへのアクセスを許可する。

【０１８７】

ステップ２４２０において、プロセッサは、企業からの第２のパーミッションをトークンに追加し、第１の暗号化ハッシュと第２のパーミッションとの第２の暗号化ハッシュを計算し、第１の暗号化ハッシュをトークンから除去し、それにより減衰トークンを得ることによって、トークンによって付与されたアクセスを減衰させ得る。第１の権限源及び第２の権限源から夫々得られた第１のパーミッション及び第２のパーミッションは、複数のパーミッション及びトークン及び減衰トークン内のエントリに変換され得る。ステップ２４３０において、プロセッサは、減衰トークンをユーザデバイス等の要求元デバイスに送信することができる。

【０１８８】

トークンを取得する為に、プロセッサは、ブロックチェーンに格納された権限を計算できる。ブロックチェーンは複数のブロックを含み得、ブロックチェーン内のブロックは、暗号ユーザＩＤの権限を定義する。権限は、ブロックチェーン上で実行する為に、暗号ユーザＩＤに関連する少なくとも１つの操作を定義することができる。要求を行うユーザがブロックチェーンにアクセスする権限を有するかどうかを判断する為に、プロセッサは、ブロックチェーンを初期ブロックから最終ブロックまで確認することを含め、ブロックチ

ェーンに記録された権限を計算してもよい。プロセッサは、要求を行うユーザがブロックチェーンにアクセスする権限を有すると判断した場合に、要求を行うユーザにブロックチェーンへのアクセスを付与するトークンを生成することができる。

【０１８９】

プロセッサは、有効なトークンを受信すると、ブロックチェーンへのアクセスを付与できる。プロセッサは、ブロックチェーンの一部にアクセスする要求と、減衰トークンとを受信することができる。プロセッサは、減衰トークン内に格納されたキーＩＤを使用して、秘密のルート鍵を取得することができる。プロセッサは、秘密のルート鍵と第１のパーミッションとの暗号化ハッシュを計算して、第３の暗号化ハッシュを取得することができる。プロセッサは、第３の暗号化ハッシュと第２のパーミッションとの暗号化ハッシュを計算して、第４の暗号化ハッシュを取得することができる。プロセッサは、減衰トークンに含まれる第２の暗号化ハッシュと第４の暗号化ハッシュを比較することによって、減衰トークンに含まれる第２の暗号化ハッシュが第４の暗号化ハッシュに一致するかどうかを判断することができる。減衰トークンに含まれる第２の暗号化ハッシュと第４の暗号化ハッシュとが一致すると判断した場合、プロセッサは、ブロックチェーンの一部にアクセスする要求を許可することができる。第２の暗号化ハッシュと第４の暗号化ハッシュとが一致しない場合、プロセッサは、減衰トークンが有効でない為、要求の許可を拒否することができる。

【０１９０】

減衰トークンに追加のパーミッション、即ち制約又は注意事項を追加することによって、減衰トークンを更に減衰させることができる。例えば、減衰トークン保有者は、ブロックチェーンへのアクセスの一部を第三者に委任したいと思う場合があり得る。その為、減衰トークン保持者は、第三者にアクセスの一部を許可する追加の減衰トークンの作成を要求することができる。例えば、トークン保有者は、第三者がアクセスすることができるチーム内の特定のブロックを指定できる。

【０１９１】

更に減衰したトークンを作成する為に、プロセッサは、減衰トークンと、第３のパーミッションの要求とを受信することができる。一実施形態では、プロセッサは、第３のパーミッションが第１のパーミッション及び第２のパーミッションによって認可されているかどうかを判断することができる。例えば、第１のパーミッションと第２のパーミッションは、チーム１のユーザアリスへのアクセスを付与し、第３のパーミッションは、チーム２へのアクセスを要求することができる。プロセッサは、ユーザアリスにはチーム２へのアクセス権がないと判断し、減衰トークンの作成を拒否することができる。別の例では、第３のパーミッションが第１のパーミッション及び第２のパーミッションによって許可されていると判断すると、プロセッサは、第２の減衰トークンの作成を、第２の暗号化ハッシュと第３のパーミッションとの暗号化ハッシュを計算し、第２の暗号化ハッシュを減衰トークンから削除し、第３のパーミッションを減衰トークンに追加し、第３の暗号化ハッシュを減衰トークンに追加して、それにより第２の減衰トークンを作成することによって、行い得る。

【０１９２】

別の実施形態では、プロセッサは、パーミッションの有効性を判断せず、代わりに、減衰トークンを作成するのみである。パーミッションの有効性は、トークンの有効性が決定される時に決定され得る。第３のパーミッションが、第１及び第２のパーミッションによって付与されないデータを要求する場合、空白応答が要求者に提供され得る。

【０１９３】

プロセッサは、リカバリー鍵に基づいてトークンを付与することができる。プロセッサは、ブロックチェーンに記録された権限を計算することなく、要求がリカバリー鍵で署名されているかどうかを判断することによって、トークンの要求を行うユーザがブロックチェーンにアクセスする権限を有するかどうかを判断することができる。プロセッサは、要求がリカバリー鍵で署名されていると判断した場合に、要求を行ったユーザにブロックチ

ェーンへの無制限のアクセスを付与する第２のトークンを生成することができる。本願明細書に記載されているように、プロセッサは、リカバリー鍵を３０個の部分等の複数の部分に分割し、各部分を暗号化し、暗号化された鍵部分を複数のデバイスに配布し、複数の部分からリカバリー鍵を組み立てる為にデバイスの少なくとも過半数の参加を要求し得る。

【０１９４】

ユーザデバイスを侵害することによってシステムに侵入する攻撃者から保護する為に、プロセッサは分割鍵を実装でき、第１の暗号鍵はサーバに格納され、第２の暗号鍵はユーザデバイスに格納される。プロセッサは、ユーザデバイスから第１の暗号鍵の要求を受信すると、プロセッサは、要求受信時に、ユーザデバイスが第１の暗号鍵の受信を許可されているかどうかを判断することができる。例えば、ユーザデバイスが盗難品であると報告された場合、プロセッサは、ユーザデバイスがサーバに格納されている第１の暗号鍵を受信することを許可しない。ユーザデバイスが暗号鍵を受信した場合、ユーザデバイスは、ＨＭＡＣ等の第１暗号鍵と第２暗号鍵の組み合わせであるＫＤＦを実行し、暗号化されたデータの復号に用いることができる鍵を得ることができる。

【０１９５】

プロセッサは、ブロックチェーンのセマンティクスの解釈に関する更新を受信することができる。プロセッサは、ブロックチェーン内に更新を格納することによって、複数のユーザデバイスに亘るブロックチェーンのセマンティクスの解釈の一貫性を確保することができる。

【０１９６】

トークンに格納されたタイムセンシティブなパーミッション等のタイムセンシティブなパーミッションを実施する為に、プロセッサは、複数のブロックを含むクロックブロックチェーンを作成することができる。複数のブロックの内の各ブロックは、先行するブロックのタイムスタンプよりも大きいタイムスタンプを含み得る。プロセッサは、ブロックチェーン内のブロックと、クロックブロックチェーン内のクロックブロックとの間の時間的関係を作成することができる。例えば、ブロックとクロックブロックとの間のリンクは、ブロックがクロックブロックの前に、クロックブロックの後に、クロックブロックと同時に、クロックブロックの後及び／又は前の指定時間内に作成されたこと等を示し得る。

【０１９７】

プロセッサは、時間制限付きパーミッションを含むトークンと、ブロックチェーンの一部にアクセスする要求とを受信することができる。プロセッサは、時間制限付きパーミッションが、ブロックチェーンの要求された部分に関連するクロックブロックチェーンによって許可されているか否かを判断することができる。プロセッサは、時間制限付きパーミッションがクロックブロックチェーンによって許可されていないと判断した場合、ブロックチェーンの一部にアクセスする要求を拒否することができる。例えば、クロックブロックチェーンにおける最新のブロックが時間制限付きパーミッションを過ぎている、又はブロックチェーンの要求された部分が時間制限付きパーミッションウィンドウの外に作成された、等であり得る。

【０１９８】

（コンピュータ）

図２５は、本明細書で論じた方法論又はモジュールの何れか１つ以上を機械に実行させる為の命令のセットが実行され得るコンピュータシステム２５００の例示的形態での機械の図式的表現である。

【０１９９】

図２５の例では、コンピュータシステム２５００は、プロセッサ、メモリ、不揮発性メモリ、及びインターフェースデバイスを含む。様々な共通の構成要素（例えば、キャッシュメモリ）は、説明の簡略化の為に省略されている。コンピュータシステム２５００は、図１～２４の実施例で説明した構成要素（及び本明細書で説明する他の任意の構成要素）の何れかが実装可能なハードウェアデバイスを例示することを意図している。コンピュータシステム２５００は、任意の適用可能な既知の又は便利なタイプであり得る。コンピュ

ータシステム２５００の構成要素は、バスを介して、又は他の既知の若しくは便利なデバイスを介して結合され得る。

【０２００】

コンピュータシステム２５００は、図１の１００、図７の７５０、１０００及び図１０Ａ、図１１Ａ～Ｃの１１４０、図１４の１４００、図１５の１５００のようなサーバを表し得る。コンピュータシステム２５００は、図１の１１０～１１６、図７の７００～７２０、図１０Ａの１０１０～１０３０、図１１Ａ～１１Ｃの１１００～１１２０、図１４の１４４０、図１５の１５４０、図１５の１４５０、図１５の１５５０等のデバイスを表し得る。システム２５００のプロセッサは、本願で説明した様々な方法及び命令を実行することができる。システム２５００のメインメモリ、不揮発性メモリ、及び／又は駆動装置は、プロセッサによって実行される命令を格納することができる。デバイス１１１０～１１６０、７００～７２０、１０１０～１０３０、１１００～１１２０、及びサーバ１００、７５０、１０００、１１４０、１４００、１５００、１４４０、１５４０は、システム２５００のネットワークインターフェース装置を用いて互いに通信できる。例えば、図１４のトークン１４８０、１４９４、図１５のトークン１５８０、１５９４は、システム２５００のネットワークインターフェースを介して通信することができる。

【０２０１】

本開示は、コンピュータシステム２５００が任意の適切な物理的形態を取ることを企図する。例として、限定するものではないが、コンピュータシステム２５００は、埋込式コンピュータシステム、システムオンチップ（ＳＯＣ）、シングルボードコンピュータシステム（ＳＢＣ）（例えば、コンピュータオンモジュール（ＣＯＭ）又はシステムオンモジュール（ＳＯＭ）等）、デスクトップコンピュータシステム、ラップトップ若しくはノートブックコンピュータシステム、インタラクティブキオスク、メインフレーム、コンピュータシステムのメッシュ、携帯電話、パーソナルデジタルアシスタント（ＰＤＡ）、サーバ又はこれらの２以上の組合せであってもよい。適切な場合、コンピュータシステム２５００は、１つ以上のコンピュータシステム２５００を含んでよく、ユニット式又は分散式であってよく、複数の場所に亘ってもよく、複数の機械に亘ってもよく、又はクラウドに存在してもよく、このクラウドには１つ以上のネットワークにおける１つ以上のクラウドコンポーネントが含まれてもよい。適切な場合、１つ以上のコンピュータシステム２５００は、本明細書で説明又は例示される１つ以上の方法の１つ以上のステップを実質的に空間的又は時間的に制限することなく実行してもよい。例として、限定するものではないが、１つ以上のコンピュータシステム２５００は、本明細書で説明又は例示される１つ以上の方法の１つ以上のステップをリアルタイム又はバッチモードで実行してもよい。１つ以上のコンピュータシステム２５００は、適切な場合、本明細書で説明又は図示する１つ以上の方法の１つ以上のステップを異なる時間又は異なる場所で実行してもよい。

【０２０２】

プロセッサは、例えば、インテルＰｅｎｔｉｕｍマイクロプロセッサ又はモトローラｐｏｗｅｒＰＣマイクロプロセッサ等の従来のマイクロプロセッサであってもよい。関連技術の当業者であれば、「機械可読（記憶）媒体」又は「コンピュータ可読（記憶）媒体」という用語が、プロセッサによってアクセス可能な任意のタイプのデバイスを含むことを認識するであろう。

【０２０３】

メモリは、例えば、バスによってプロセッサに結合される。メモリは、限定するものではないが、例として、ダイナミックＲＡＭ（ＤＲＡＭ）及びスタティックＲＡＭ（ＳＲＡＭ）等のランダムアクセスメモリ（ＲＡＭ）を含み得る。メモリは、ローカル、リモート、又は分散型であり得る。

【０２０４】

バスは、プロセッサを不揮発性メモリ及びドライブユニットも結合する。不揮発性メモリは、多くの場合、磁気フロッピー又はハードディスク、磁気－光ディスク、光ディスク、ＣＤ‐ＲＯＭ、ＥＰＲＯＭ、又はＥＥＰＲＯＭ等の読み取り専用メモリ（ＲＯＭ）、磁

気又は光カード、或いは大量のデータ用の別の形態のストレージである。このデータの一部は、コンピュータ２５００におけるソフトウェアの実行中に、直接メモリアクセス処理によって、メモリに書き込まれることが多い。不揮発性ストレージは、ローカル、リモート、又は分散型であり得る。不揮発性メモリは任意であるが、これは、メモリ内で利用可能な全ての適用可能なデータでシステムが作成され得るからである。典型的なコンピュータシステムは、通常、少なくともプロセッサ、メモリ、及びメモリをプロセッサに結合する装置（例えば、バス）を含む。

【０２０５】

ソフトウェアは、通常、不揮発性メモリ及び／又は駆動装置に格納される。実際、大規模なプログラム全体をメモリに格納することは可能でない場合さえもあり得る。それでも、ソフトウェアを実行する為には、必要に応じて、処理に適したコンピュータ可読の場所に移動させ、説明の為に、その場所を本稿ではメモリと呼ぶことを理解されたい。ソフトウェアが実行の為にメモリに移動される場合でも、プロセッサは通常、ソフトウェアに関連する値を格納する為のハードウェアレジスタ、及び理想的には実行を高速化する為に役立つローカルキャッシュを利用する。本明細書では、ソフトウェアプログラムが「コンピュータ可読媒体に実装されている」と言及される場合、ソフトウェアプログラムは、任意の既知の又は便利な場所（不揮発性ストレージからハードウェアレジスタまで）に格納されていると仮定される。プロセッサは、プログラムに関連する少なくとも１つの値がプロセッサによって読み取り可能なレジスタに格納される時、「プログラムを実行するように構成される」と見做される。

【０２０６】

バスは又、プロセッサをネットワークインターフェースデバイスに結合する。インターフェースは、モデム又はネットワークインターフェースの内の１つ以上を含み得る。モデム又はネットワークインターフェースは、コンピュータシステム２５００の一部であると考えられ得ることが理解されよう。インターフェースは、アナログモデム、ｉｓｄｎモデム、ケーブルモデム、トークンリングインターフェース、衛星伝送インターフェース（例えば、「ダイレクトＰＣ」）、又はコンピュータシステムを他のコンピュータシステムに結合する為の他のインターフェースを含み得る。インターフェースは、１つ以上の入力及び／又は出力デバイスを含み得る。入出力デバイスは、限定するものではないが、例として、キーボード、マウス又は他のポインティングデバイス、ディスクドライブ、プリンタ、スキャナ、及びディスプレイデバイスを含む他の入出力デバイスを含み得る。ディスプレイデバイスは、限定するものではないが例として、陰極線管（ＣＲＴ）、液晶ディスプレイ（ＬＣＤ）、又は他の適用可能な既知の若しくは便利なディスプレイデバイスを含み得る。簡単にする為に、図２５の実施例に描かれていない任意のデバイスのコントローラは、インターフェースに存在すると仮定される。

【０２０７】

稼動時、コンピュータシステム２５００は、ディスクオペレーティングシステムのようなファイル管理システムを含むオペレーティングシステムソフトウェアによって制御され得る。関連するファイル管理システムソフトウェアを有するオペレーティングシステムソフトウェアの一例は、ワシントン州レッドモンドのマイクロソフト社からのＷｉｎｄｏｗｓ（登録商標）として知られるオペレーティングシステムのファミリー、及びそれらの関連するファイル管理システムである。関連するファイル管理システムソフトウェアを伴うオペレーティングシステムソフトウェアの別の例としては、Ｌｉｎｕｘ（登録商標）オペレーティングシステムとその関連するファイル管理システムがある。ファイル管理システムは、典型的には、不揮発性メモリ及び／又はドライブユニットに格納され、プロセッサに、不揮発性メモリ及び／又はドライブユニットにファイルを格納することを含む、データの入出力及びメモリへのデータの格納の為にオペレーティングシステムが必要とする様々な行為を実行させる。

【０２０８】

詳細な説明の幾つかの部分は、コンピュータメモリ内のデータビットに対する操作のア

ルゴリズム及び記号表現の観点から提示されることがある。これらのアルゴリズムの記述及び表現は、データ処理技術の当業者が、当業者に自身の仕事の実質を最も効果的に伝える為に使用する手段である。アルゴリズムとは、ここでは、一般に、所望の結果をもたらす自己矛盾のない一連の演算であると考えられている。演算は、物理量の物理的操作を必要とするものである。通常、これらの量は、必ずしもそうではないが、格納、転送、結合、比較、及びその他の操作を行うことができる電気信号又は磁気信号の形態を取る。これらの信号をビット、値、要素、記号、文字、用語、数等と呼ぶことは、主に一般的な使用上の理由から、時に便利であることが証明されている。

【０２０９】

しかし、これら及び類似の用語の全ては、適切な物理量と関連付けられるべきものであり、これらの量に適用される便利なラベルに過ぎないことに留意すべきである。以下の議論から明らかなように特に別段の記載がない限り、本明細書を通じて、「処理」又は「計算」又は「算出」又は「測定」又は「表示」又は「生成」等の用語を利用する議論は、コンピュータシステムのレジスタ及びメモリ内の物理的（電子）量として表されるデータを、コンピュータシステムのメモリ若しくはレジスタ又は他のその種の情報記憶、伝送若しくは表示デバイス内の物理的量として同様に表される他のデータに操作及び変換するコンピュータシステム又は同様の電子計算デバイスの作用及び処理を指していると理解されよう。

【０２１０】

本明細書に提示されたアルゴリズム及びディスプレイは、特定のコンピュータ又は他の装置に本質的に関連するものではない。様々な汎用システムが、本明細書の教示に係るプログラムと共に使用されてもよく、或いは、幾つかの実施形態の方法を実行する為により特殊な装置を構築することが便利であることが判明する可能性もある。これらの様々なシステムの為に必要な構造は、以下の説明から判明するであろう。加えて、本技術は、任意の特定のプログラミング言語を参照して説明されておらず、様々な実施形態は、従って、様々なプログラミング言語を使用して実施され得る。

【０２１１】

別の実施形態では、機械はスタンドアロン装置として動作するか、又は他の機械に接続（例えば、ネットワーク化）されてもよい。ネットワーク化された配備では、機械は、クライアントサーバネットワーク環境においてサーバ又はクライアント機械の能力で、或いはピアツーピア（又は分散）ネットワーク環境においてピア機械として動作してもよい。

【０２１２】

機械は、サーバコンピュータ、クライアントコンピュータ、パーソナルコンピュータ（ＰＣ）、タブレットＰＣ、ラップトップコンピュータ、セットトップボックス（ＳＴＢ）、パーソナルデジタルアシスタント（ＰＤＡ）、携帯電話、ｉＰｈｏｎｅ（登録商標）、ブラックベリー、プロセッサ、電話、Ｗｅｂアプライアンス、ネットワークルータ、スイッチ、ブリッジ、又はその機械によって行われるべき動作を指定する命令のセット（連続又はその他）を実行できる任意の機械であり得る。

【０２１３】

機械可読媒体又は機械可読記憶媒体は、例示的な実施形態では単一の媒体であるように示されているが、用語「機械可読媒体」及び「機械可読記憶媒体」は、１組以上の命令を格納する単一の媒体又は複数の媒体（例えば、集中型又は分散型データベース、及び／又は関連するキャッシュとサーバ）を含むように解釈されるべきである。又、「機械可読媒体」及び「機械可読記憶媒体」という用語は、機械による実行の為の命令のセットを記憶、符号化又は担持することができ、機械に、現在開示されている技術及び革新の方法論又はモジュールの何れか１つ以上を実行させる任意の媒体を含むものと解釈されるものとする。

【０２１４】

一般に、本開示の実施形態を実装する為に実行されるルーチンは、オペレーティングシステム或いは「コンピュータプログラム」と呼ばれる特定のアプリケーション、コンポー

ネント、プログラム、オブジェクト、モジュール又は命令のシーケンスの一部として実装され得る。コンピュータプログラムは、典型的には、コンピュータ内の様々なメモリ及び記憶装置に様々なタイミングで設定され、コンピュータ内の１つ以上の処理ユニット又はプロセッサによって読み取られ実行されると、コンピュータに本開示の様々な態様に関わる要素を実行する為の動作を実行させる１つ以上の命令から構成されている。

【０２１５】

更に、実施形態は、完全に機能するコンピュータ及びコンピュータシステムの文脈で説明されてきたが、当業者ならば、様々な実施形態が様々な形態のプログラム製品として配布可能であり、配布を実際に行う為に使用される特定のタイプの機械又はコンピュータ可読媒体に関らず、開示が同様に適用されることを理解するであろう。

【０２１６】

機械可読記憶媒体、機械可読媒体、又はコンピュータ可読（記憶）媒体の更なる例としては、特に、揮発性及び不揮発性メモリデバイス、フロッピーディスク及び他のリムーバブルディスク、ハードディスクドライブ、光ディスク等（例えば、コンパクトディスクリードオンリーメモリ（ＣＤＲＯＭ）、デジタル多用途ディスク、（ＤＶＤ）等）の記録可能媒体、デジタル及びアナログ通信リンク等の伝送型媒体があるがこれに限定されるものではない。

【０２１７】

或る状況では、例えば、２進数の１から２進数の０又はその逆への状態の変化等のメモリデバイスの動作は、物理的変換等の変換を含んでいてもよい。特定のタイプのメモリデバイスでは、そのような物理的変換は、物品の、異なる状態又は物への物理的変換を含んでいてもよい。例えば、限定するものではないが、幾つかのタイプのメモリデバイスでは、状態の変化は、電荷の蓄積及び貯蔵又は貯蔵された電荷の放出を含む場合がある。同様に、他のメモリデバイスでは、状態の変化は、磁気配向の物理的変化又は変換、或いは結晶質から非晶質又はその逆のような分子構造の物理的変化又は変換を含み得る。前述は、メモリデバイスにおける２進数の１から２進数の０又はその逆の状態変化が、物理的変換等の変換を含んでいてもよい、網羅的なリストであることを意図していない。寧ろ、上記は、例示的な例として意図されている。

【０２１８】

記憶媒体は、典型的には、非一時的であってもよいし、非一時的デバイスを構成してもよい。この文脈では、非一時的記憶媒体は、有形であるデバイス、即ち、デバイスがその物理的状態を変化させ得るが、デバイスが具体的な物理的形態を有することを意味するデバイスを含んでもよい。従って、例えば、非一時的とは、このように状態が変化するにも係らず、有形の状態を保つデバイスを指す。

【０２１９】

（備考）

本明細書で使用される言語は、主として読み易さと説明の為に選択されたものであり、発明的主題を描写又は包括する為に選択されたものではない場合がある。従って、本発明の範囲は、この詳細な説明によってではなく、寧ろ、これに基づく出願で公表される任意の請求項によって限定されることが意図される。従って、様々な実施形態の開示は、以下の特許請求の範囲に規定される実施形態の範囲を例示するものであるが、限定するものではないことを意図している。

〔付記１〕

コンピュータ実装方法であって、

複数のブロックを含むブロックチェーンを作成するステップを、

ユーザの権限を定義するブロックを作成するステップであって、前記ブロックは、ユーザを識別する暗号ユーザＩＤと、前記暗号ユーザＩＤに関連付けられた権限とを含み、前記権限は、前記ブロックチェーンで実行する為に前記暗号ユーザＩＤに関連付けられた少なくとも１つの操作を定義している、ステップと、

前記ブロックを前記ブロックチェーンの末尾に追加するステップであって、前記ブロッ

クチェーンは暗号化されていない、ステップと、によって行うステップと、
前記ブロックチェーンにアクセスする要求を要求元デバイスから受信するステップであって、前記要求は、前記要求を行う前記ユーザに関連付けられた暗号ユーザＩＤを含む、ステップと、
前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする権限を有するかどうかを、前記ブロックチェーンに記録された前記権限を計算することによって判断するステップと、
前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする前記権限を有すると判断した場合に、前記要求を行う前記ユーザに前記ブロックチェーンへのアクセスを付与するトークンを生成するステップと、
前記トークンを前記要求元デバイスに送信するステップと、
を含む方法。
〔付記２〕
前記トークンを生成するステップは、
秘密のルート鍵を識別する鍵識別子を作成するステップと、
前記トークンによって付与される前記ブロックチェーンへのパーミッションを作成するステップと、
前記秘密のルート鍵と前記パーミッションとの暗号化ハッシュを作成するステップと、
前記鍵識別子、前記パーミッション、及び前記暗号化ハッシュを前記トークンに追加するステップと、を含む、付記１に記載の方法。
〔付記３〕
前記要求を行う前記ユーザが前記権限を有するか否かを判断するステップは、
前記ブロックチェーンに記録された前記権限を計算することなく、前記ブロックチェーンを初期ブロックから最終ブロックまで確認することを含めて、前記要求がリカバリー鍵で署名されているかどうかを判断することによって、前記要求を行う前記ユーザが前記ブロックチェーンにアクセスする前記権限を有するかどうかを判断するステップと、
前記要求が前記リカバリー鍵で署名されていると判断した場合に、前記要求を行う前記ユーザに前記ブロックチェーンへの無制限のアクセスを付与する第２のトークンを生成するステップと、を含む、付記１に記載の方法。
〔付記４〕
前記リカバリー鍵を複数の部分に分離するステップと、
前記複数の部分の内の少なくとも一部のサブセットを暗号化するステップと、
前記暗号化された部分のサブセットと前記複数の部分の残りを複数のデバイスに配布するステップと、を含む付記３に記載の方法。
〔付記５〕
第１の暗号鍵をサーバに格納するステップと、
第２の暗号鍵をユーザデバイスに送信するステップと、
前記ユーザデバイスから前記第１の暗号鍵の要求を受信するステップと、
前記要求を受信した時に、前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されているかどうかを判断するステップと、
前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されていないと判断した場合、前記第１の暗号鍵の送信を拒否するステップと、を含む付記１に記載の方法。
〔付記６〕
前記ブロックチェーンのセマンティクスの解釈に関する更新を受信するステップと、
前記ブロックチェーン内に前記更新を格納することにより、複数のユーザデバイスに亘る前記ブロックチェーンのセマンティクスの解釈の一貫性を確保するステップと、を含む付記１に記載の方法。
〔付記７〕
コンピュータ実装方法であって、
ブロックチェーンへのアクセスを付与するトークンを取得するステップであって、前記

ブロックチェーンへの前記アクセスは、前記ブロックチェーンによって定義される第１の権限源によって許可され、前記トークンは、秘密のルート鍵を識別する鍵識別子と、前記ブロックチェーンによって定義される前記第１の権限源によって許可される前記ブロックチェーンにアクセスする第１のパーミッションと、前記秘密のルート鍵と前記第１のパーミッションとの第１の暗号化ハッシュとを含む、ステップと、

要求元デバイスから前記ブロックチェーンにアクセスする要求と、前記要求元デバイスに関連するアクセスを制限する第２の権限源から第２のパーミッションを受信するステップと、

前記トークンに前記第２の権限源からの前記第２のパーミッションを追加し、前記第１の暗号化ハッシュと前記第２のパーミッションとの第２の暗号化ハッシュを計算し、前記トークンから前記第１の暗号化ハッシュを削除することにより、前記トークンによって付与される前記アクセスを減衰させ、減衰トークンを取得するステップと、

前記減衰トークンを前記要求元デバイスに送信するステップと、を含む方法。

〔付記８〕

前記ブロックチェーンが複数のブロックを含み、前記ブロックチェーン内のブロックが暗号ユーザＩＤの権限を定義し、前記権限が前記ブロックチェーンで実行する為に前記暗号ユーザＩＤに関連付けられる操作を少なくとも定義し、前記トークンを取得するステップが、

前記ブロックチェーンを初期ブロックから最終ブロックまで確認することを含めて、前記要求を行うユーザが前記ブロックチェーンにアクセスする権限を有するかどうかを、前記ブロックチェーンに記録された前記権限を計算することによって判断するステップと、

前記要求を行うユーザが前記ブロックチェーンにアクセスする前記権限を有すると判断した場合に、前記要求を行うユーザに前記ブロックチェーンへのアクセスを付与するトークンを生成するステップと、を含む、付記７に記載の方法。

〔付記９〕

前記第１のパーミッションは、前記第１のパーミッションが付与される暗号ユーザＩＤと、前記暗号ユーザＩＤがアクセスする前記ブロックチェーンの少なくとも一部の識別とを含む、付記７に記載の方法。

〔付記１０〕

前記第２のパーミッションは、時間制限又は地理的位置制限を含む、付記７に記載の方法。

〔付記１１〕

前記ブロックチェーン及び前記減衰トークンにアクセスする第２の要求を受信するステップと、

前記秘密のルート鍵を取得するステップと、

前記秘密のルート鍵と前記第１のパーミッションとの暗号化ハッシュを計算して第３の暗号化ハッシュを取得するステップと、

前記第３の暗号化ハッシュと前記第２のパーミッションとの暗号化ハッシュを計算して第４の暗号化ハッシュを取得するステップと、

前記減衰トークンに含まれる前記第２の暗号化ハッシュと前記第４の暗号化ハッシュを比較することにより、前記減衰トークンに含まれる前記第２の暗号化ハッシュが前記第４の暗号化ハッシュに一致するかどうかを判断するステップと、

前記減衰トークンに含まれる前記第２の暗号化ハッシュと前記第４の暗号化ハッシュとが一致すると判断した場合に、前記ブロックチェーンにアクセスする前記第２の要求を許可するステップと、を含む付記７に記載の方法。

〔付記１２〕

前記減衰トークンと、第３のパーミッションの要求とを受信するステップと、

第２の減衰トークンの作成を、前記第２の暗号化ハッシュと前記第３のパーミッションとの暗号化ハッシュを計算して第３の暗号化ハッシュを取得し、前記減衰トークンから前記第２の暗号化ハッシュを削除し、前記減衰トークンに前記第３のパーミッションを追加

し、前記減衰トークンに前記第３の暗号化ハッシュを追加して、それにより前記第２の減衰トークンを生成することによって行うステップと、を含む付記７に記載の方法。
〔付記１３〕
前記ブロックチェーンが複数のブロックを含み、前記ブロックチェーン内のブロックが、ユーザに関連付けられた暗号ユーザＩＤの権限を定義し、前記権限が、前記ブロックチェーン上で実行する為に前記暗号ユーザＩＤに関連付けられた操作を少なくとも定義し、前記トークンを取得するステップが、
前記ブロックチェーンに記録された前記権限を計算することなく、前記ブロックチェーンを初期ブロックから最終ブロックまで確認することを含めて、前記要求がリカバリー鍵で署名されているかどうかを判断することによって、前記要求を行うユーザが前記ブロックチェーンにアクセスする前記権限を有するかどうかを判断するステップと、
前記要求が前記リカバリー鍵で署名されていると判断した時に、前記要求を行う前記ユーザに前記ブロックチェーンへの無制限のアクセスを付与する第２のトークンを生成するステップとを含む、付記７に記載の方法。
〔付記１４〕
前記リカバリー鍵を複数の部分に分離するステップと、
前記複数の部分中の部分の少なくともサブセットを暗号化するステップと、
前記部分の暗号化されたサブセットと前記複数の部分の残りを複数のデバイスに配布するステップと、を含む付記１３に記載の方法。
〔付記１５〕
第１の暗号鍵をサーバに保存するステップと、
第２の暗号鍵をユーザデバイスに送信するステップと、
前記ユーザデバイスから前記第１の暗号鍵の要求を受信するステップと、
前記要求を受信した時に、前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されているか否かを判断するステップと、
前記ユーザデバイスが前記第１の暗号鍵を受信することを許可されていないと判断した場合、前記第１の暗号鍵の送信を拒否するステップと、を含む付記７に記載の方法。
〔付記１６〕
前記ブロックチェーンのセマンティクスの解釈に関する更新を受信するステップと、
前記更新を前記ブロックチェーン内に格納することにより、前記ブロックチェーンのセマンティクスの解釈の一貫性を複数のユーザデバイスに亘って確保するステップと、を含む、付記７に記載の方法。
〔付記１７〕
システムであって、
少なくとも１つのプロセッサと、
前記少なくとも１つのプロセッサに結合されたメモリであって、前記少なくとも１つのプロセッサによって実行された時に方法を実行するコンピュータ実行可能命令を含むメモリを備え、前記方法が、
ブロックチェーンへのアクセスを付与するトークンを取得するステップであって、前記ブロックチェーンへの前記アクセスは、前記ブロックチェーンによって定義される第１の権限源によって許可され、前記トークンは、秘密のルート鍵を識別する鍵識別子と、前記ブロックチェーンによって定義される前記第１の権限源によって許可される前記ブロックチェーンにアクセスする第１のパーミッションと、前記秘密のルート鍵と前記第１のパーミッションとの第１の暗号化ハッシュとを含む、ステップと、
ユーザデバイスから前記ブロックチェーンにアクセスする要求と、前記ユーザデバイスに関連するアクセスを制限する第２の権限源から第２のパーミッションを受信するステップと、
前記トークンに前記第２の権限源からの前記第２のパーミッションを追加し、前記第１の暗号化ハッシュと前記第２のパーミッションとの第２の暗号化ハッシュを計算し、前記トークンから前記第１の暗号化ハッシュを削除し、それにより減衰トークンを取得するこ

とによって、前記トークンによって付与される前記アクセスを減衰させるステップと、
前記減衰トークンを前記ユーザデバイスに送信するステップと、を含むシステム。
〔付記１８〕
前記第２の権限源は企業アクセス制御サーバを含む、付記１７に記載のシステム。
〔付記１９〕
前記プロセッサが前記方法を実行することは、
複数のブロックを含むクロックブロックチェーンを作成するステップであって、前記複数のブロックの各ブロックが、先行ブロックのタイムスタンプよりも大きいタイムスタンプを含む、作成ステップと、
前記ブロックチェーン内のブロックと前記クロックブロックチェーン内のブロックとの間に時間的関係を作成するステップと、
を含む付記１７に記載のシステム。
〔付記２０〕
前記プロセッサが前記方法を実行することは、
時間制限付きパーミッションと、前記ブロックチェーンにアクセスする第２の要求とを含むトークンを受信するステップと、
前記時間制限付きパーミッションが、前記ブロックチェーンに関連する前記クロックブロックチェーンによって許可されているか否かを判断するステップと、
前記時間制限付きパーミッションが前記クロックブロックチェーンによって許可されていないと判断した場合に、前記ブロックチェーンにアクセスする要求を拒否するステップと、
を含む付記１９に記載のシステム。
〔付記２１〕
前記プロセッサが前記方法を実行することは、
前記ブロックチェーンにアクセスする第２の要求と、減衰されたトークンとを受信するステップと、
前記秘密のルート鍵を取得するステップと、
前記秘密のルート鍵と前記第１のパーミッションとの暗号化ハッシュを計算して、第３の暗号化ハッシュを取得するステップと、
前記第３の暗号化ハッシュと前記第２のパーミッションとの暗号化ハッシュを計算して、第４の暗号化ハッシュを取得するステップと、
前記減衰トークンに含まれる前記第２の暗号化ハッシュと前記第４の暗号化ハッシュを比較することにより、前記減衰トークンに含まれる前記第２の暗号化ハッシュが前記第４の暗号化ハッシュに一致するかどうかを判断するステップと、
前記減衰トークンに含まれる前記第２の暗号化ハッシュと前記第４の暗号化ハッシュとが一致すると判断した場合に、前記ブロックチェーンにアクセスする前記第２の要求を許可するステップと、
を含む付記１７に記載のシステム。
〔付記２２〕
前記プロセッサが前記方法を実行することは、
前記減衰トークンと、第３のパーミッションに対する要求とを受信するステップと、
前記第３のパーミッションが前記第１のパーミッション及び前記第２のパーミッションによって許可されているかどうかを判断するステップと、
前記第３のパーミッションが前記第１のパーミッション及び前記第２のパーミッションによって許可されていると判断した場合に、第２の減衰トークンの作成を、前記第２の暗号化ハッシュと前記第３のパーミッションとの暗号化ハッシュを計算して第３の暗号化ハッシュを取得し、前記第２の暗号化ハッシュを前記減衰トークンから削除し、前記第３のパーミッションを前記減衰トークンに追加し、前記第３の暗号化ハッシュを前記減衰トークンに追加し、それにより前記第２の減衰トークンを作成することによって行うステップと、

を含む付記１７に記載のシステム。

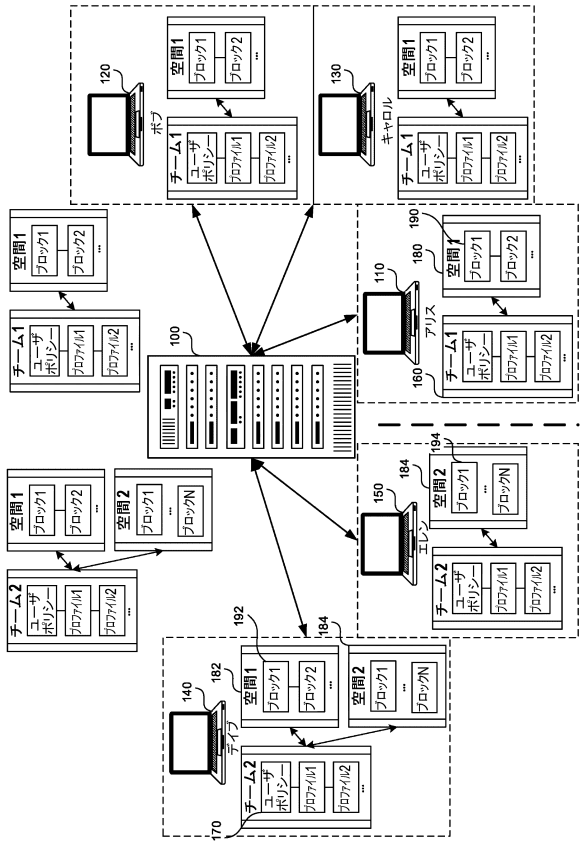
【符号の説明】

【０２２０】

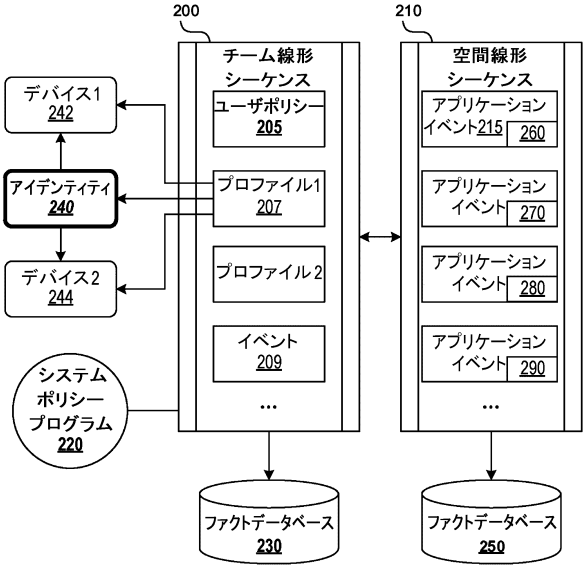
１、２　チーム
１００　サーバ
１１０、１２０、１３０、１４０、１５０　デバイス
１６０、１７０、２００　チーム線形シーケンス
１８０、１８２、１８４　空間
１９０、１９２、１９４、２１０　空間線形シーケンス
２０５　ユーザポリシー
２０９　イベント
２２０　システムポリシープログラム
２３０、２５０　ファクトデータベース
２４０　アイデンティティ
２４２　デバイス１
２４４　デバイス２
２５０　ファクトデータベース
２６０、２７０、２８０、２９０　２１５　アプリケーションイベント
１４１０　暗号化されたデータ
１４３０　アクセス制御サーバ
１４４０　トークン発行元
１４５０　ユーザデバイス
１４６０、１４７０　トークン要求
１４８０　トークン
１４９２　チケット
１４９４　減衰トークン

【図面】

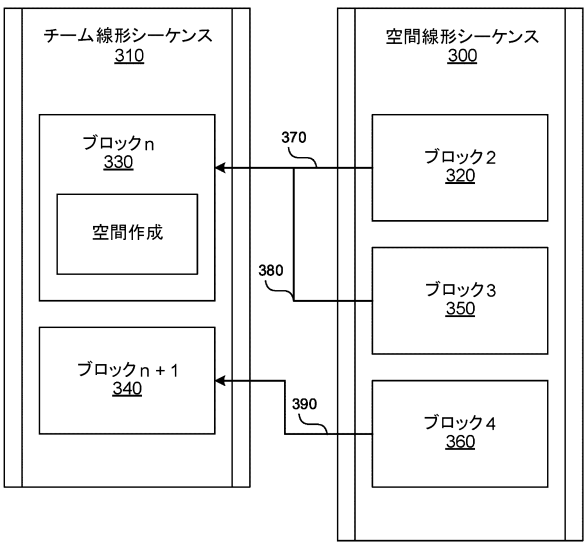
【図１】


ボブ
キャロル
アリス
エレン
テイラ
空間1
空間2
ブロック1
ブロック2
ブロックN
チーム1
チーム2
ユーザポリシー
プロファイル1
プロファイル2
100
110
120
130
140
150
160
170
180
182
184
190
192
194

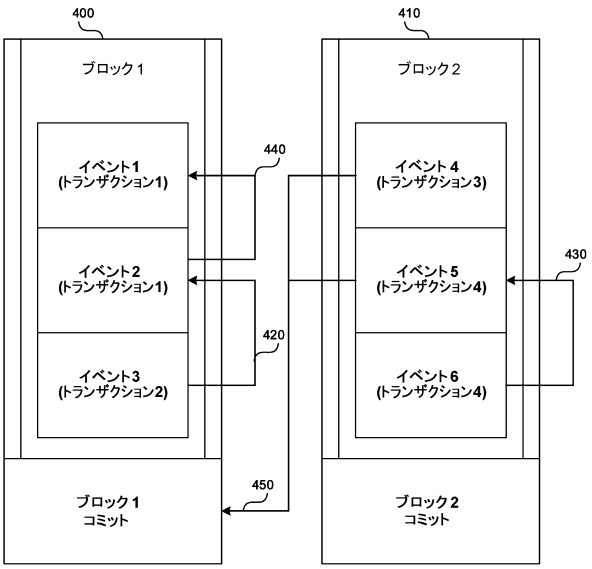
【図２】


200
210
チーム線形シーケンス
空間線形シーケンス
デバイス1 242
アイデンティティ 240
デバイス2 244
システムポリシープログラム 220
ユーザポリシー 205
プロファイル1 207
プロファイル2
イベント 209
アプリケーションイベント215 260
アプリケーションイベント 270
アプリケーションイベント 280
アプリケーションイベント 290
ファクトデータベース 230
ファクトデータベース 250

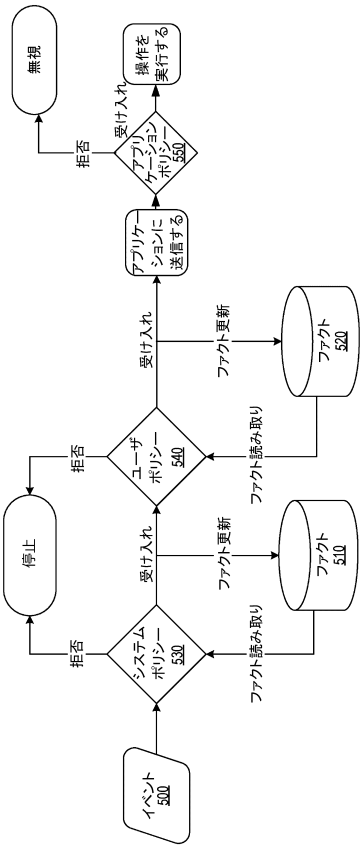
【図３】


チーム線形シーケンス 310
空間線形シーケンス 300
ブロックn 330
空間作成
ブロックn＋1 340
ブロック2 320
ブロック3 350
ブロック4 360
370
380
390

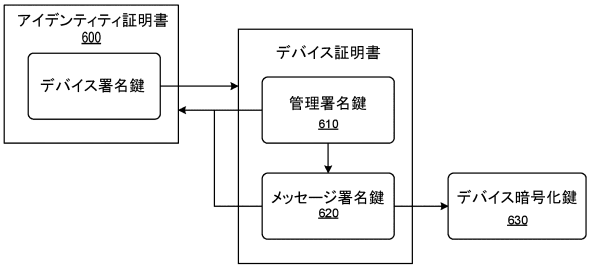
【図４】


400
410
ブロック1
ブロック2
イベント1 (トランザクション1)
イベント2 (トランザクション1)
イベント3 (トランザクション2)
イベント4 (トランザクション3)
イベント5 (トランザクション4)
イベント6 (トランザクション4)
ブロック1 コミット
ブロック2 コミット
420
430
440
450

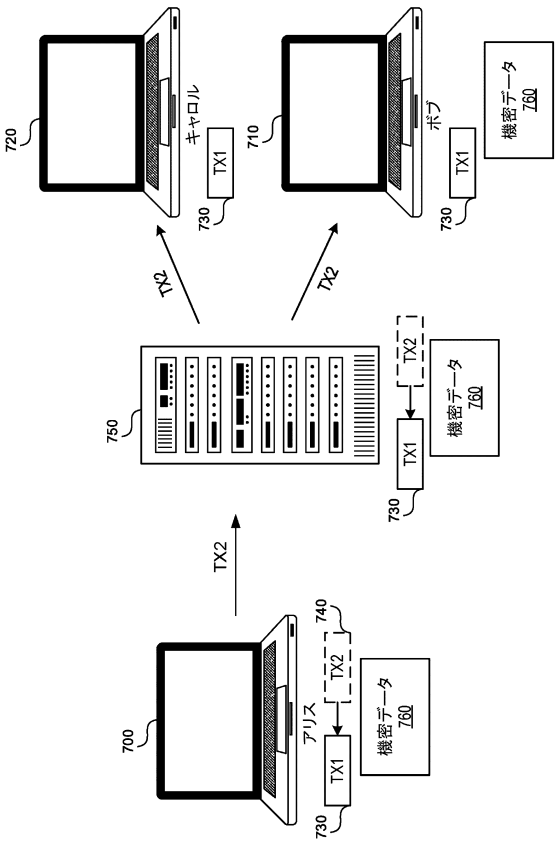
【図５】


イベント
500
システム
ポリシー
530
拒否
受け入れ
ファクト更新
ファクト読み取り
ファクト
510
停止
ユーザ
ポリシー
540
拒否
受け入れ
ファクト更新
ファクト読み取り
ファクト
520
アプリケーション
に送信する
アプリケーション
ポリシー
550
拒否
受け入れ
操作を
実行する
無視

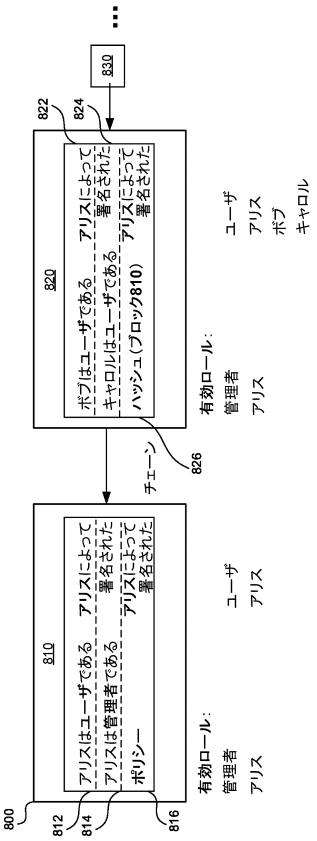
【図６】


アイデンティティ証明書
600
デバイス署名鍵
デバイス証明書
管理署名鍵
610
メッセージ署名鍵
620
デバイス暗号化鍵
630

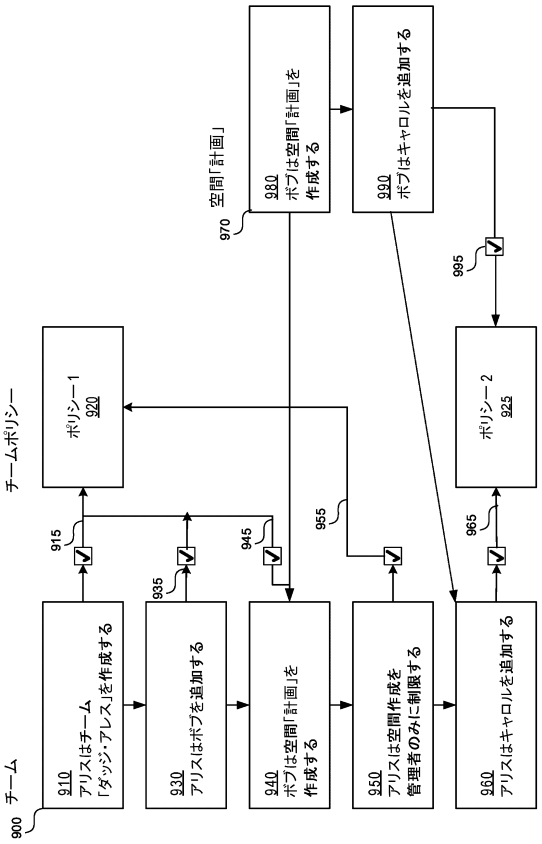
【図７】


720
キャロル
TX1
730
710
ボブ
TX1
730
機密データ
760
TX2
TX2
750
TX2
TX1
730
機密データ
760
TX2
700
アリス
740
TX2
TX1
730
機密データ
760

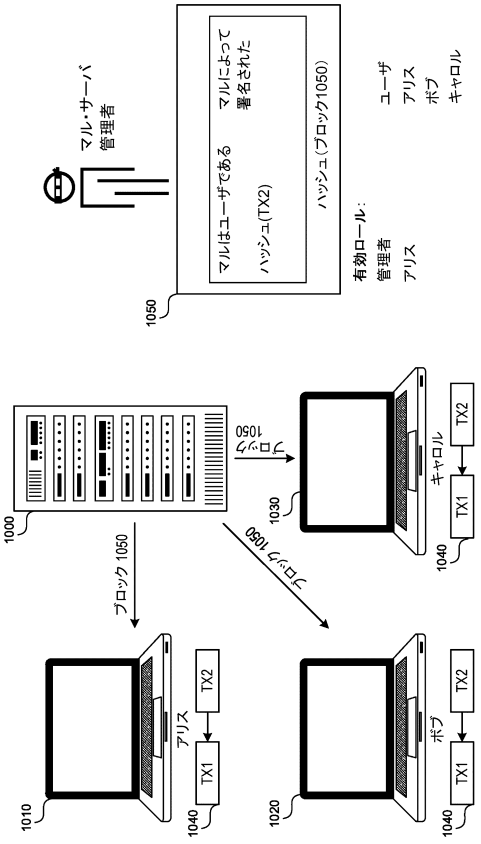
【図８】


830
822
824
820
ボブはユーザである　アリスによって署名された
キャロルはユーザである　アリスによって署名された
ハッシュ(ブロック810)　アリスによって署名された
有効ロール:
管理者
アリス
ユーザ
アリス
ボブ
キャロル
826
チェーン
810
アリスはユーザである　アリスによって署名された
アリスは管理者である　アリスによって署名された
ポリシー
有効ロール:
管理者
アリス
ユーザ
アリス
800
812
814
816

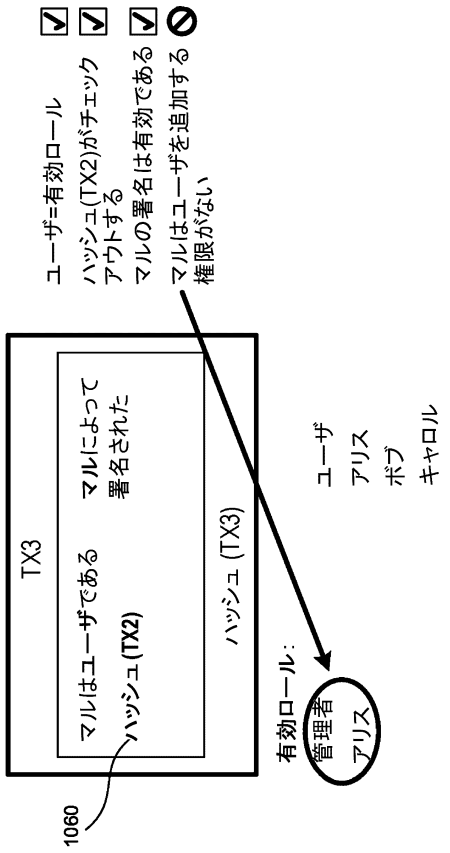
【図９】



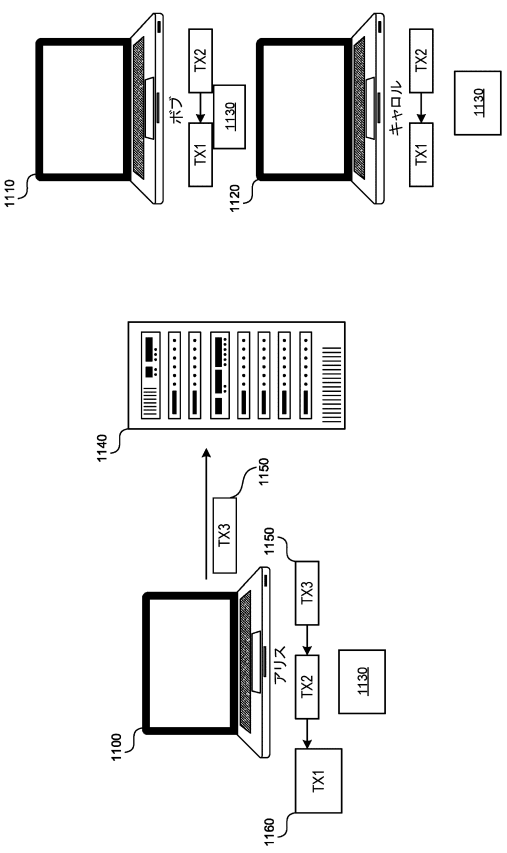
【図１０Ａ】



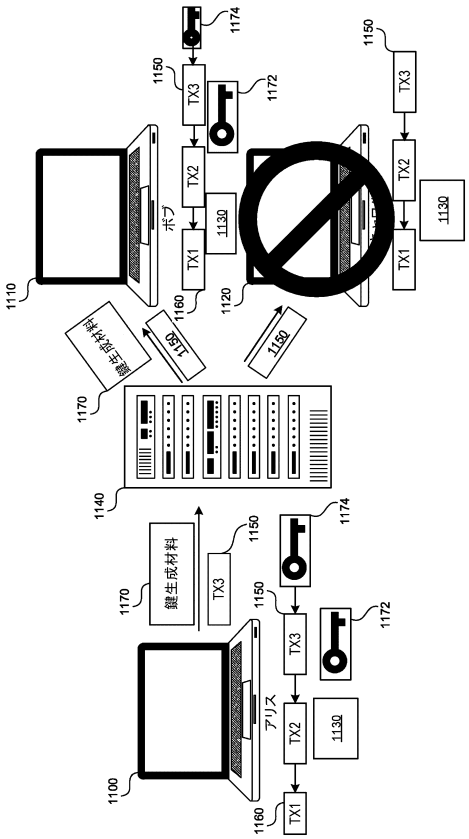
【図１０Ｂ】



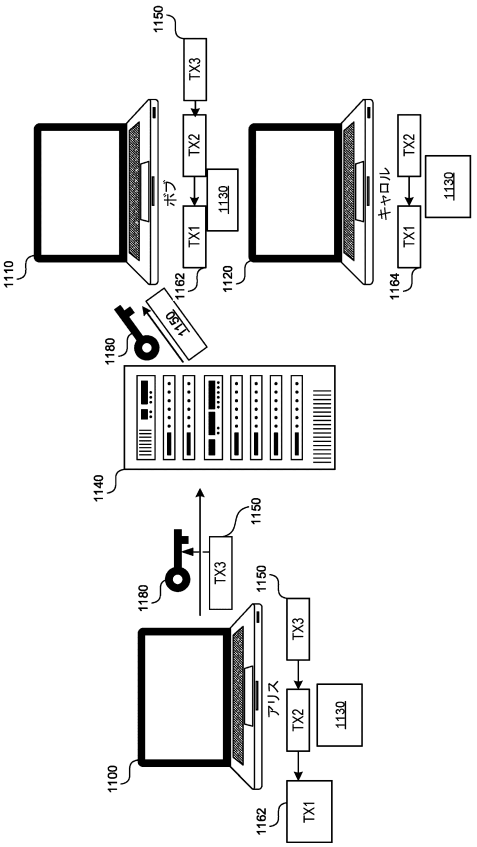
【図１１Ａ】



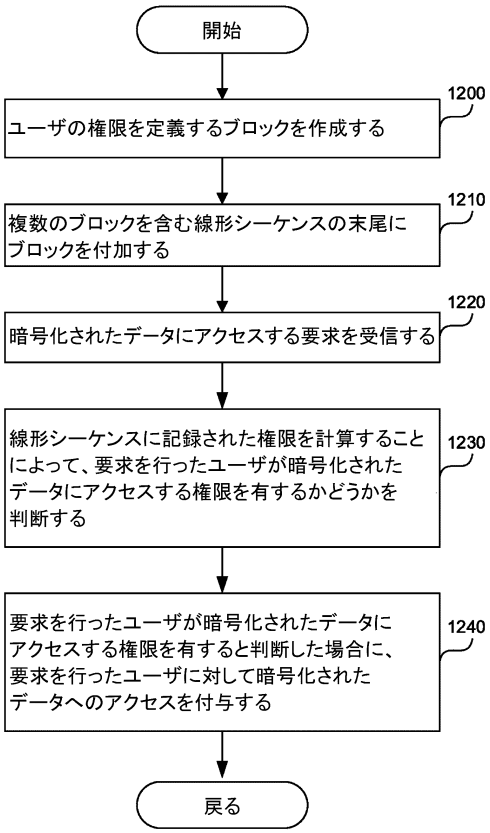
【図１１Ｂ】



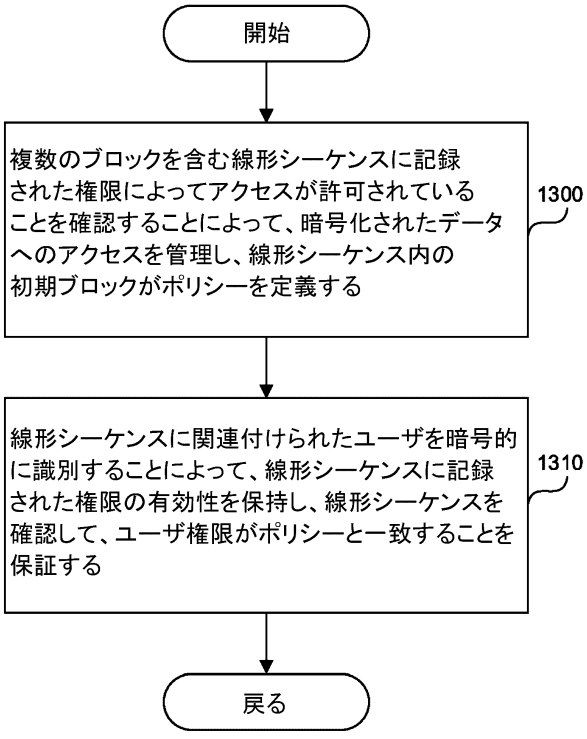
【図１１Ｃ】



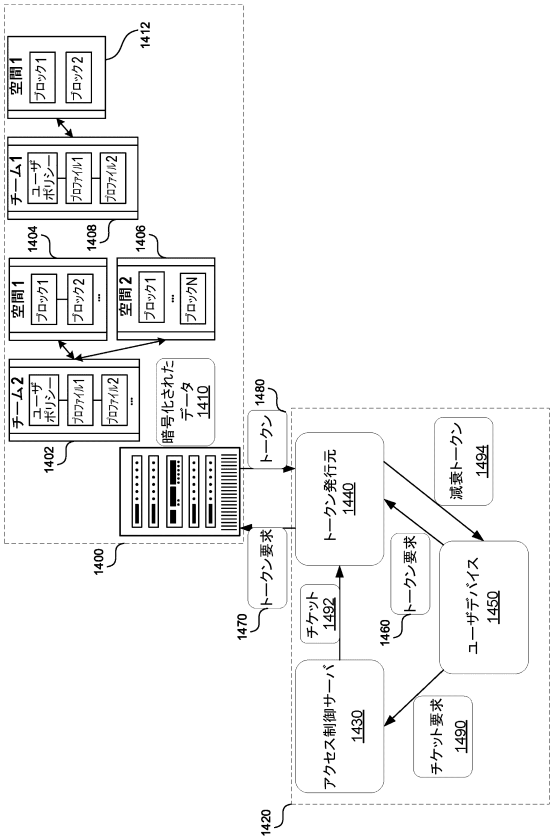
【図１２】



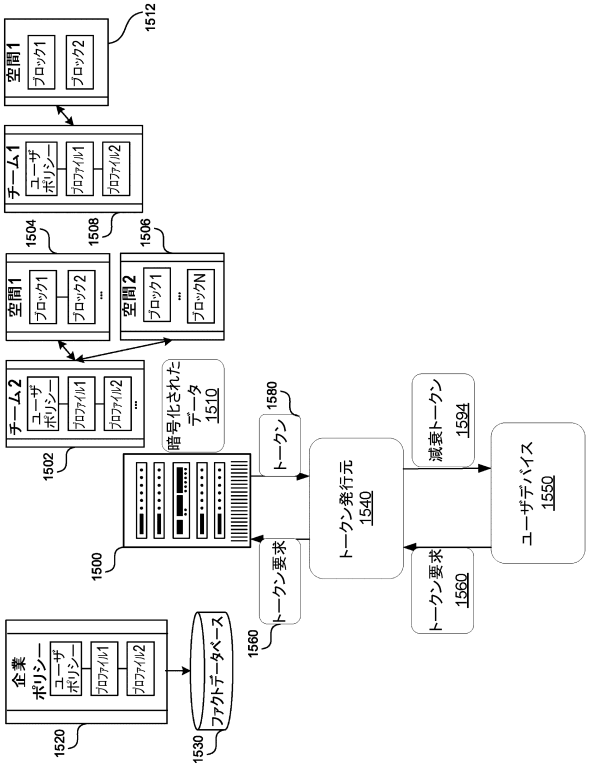
【図１３】



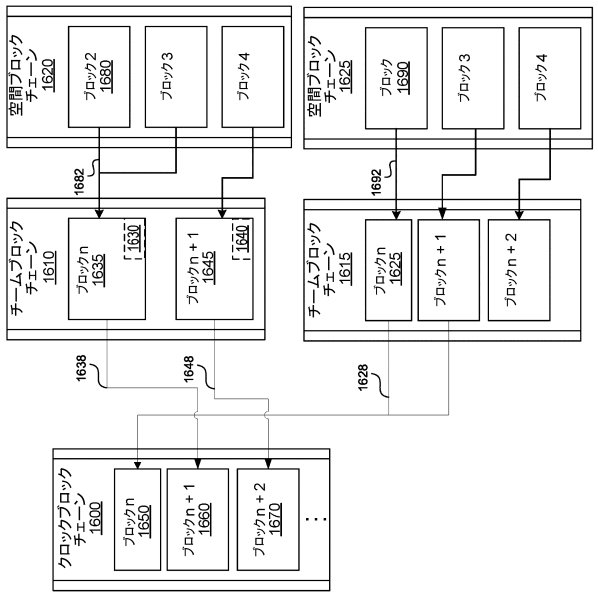
【図１４】



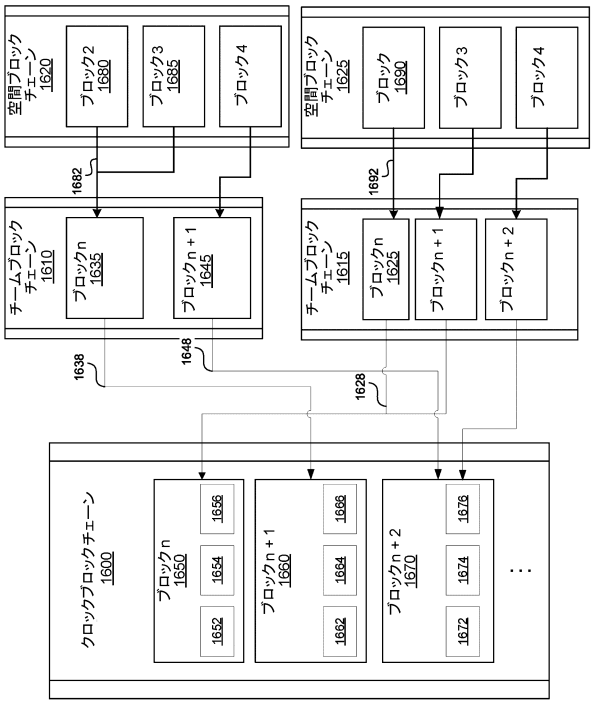
【図１５】



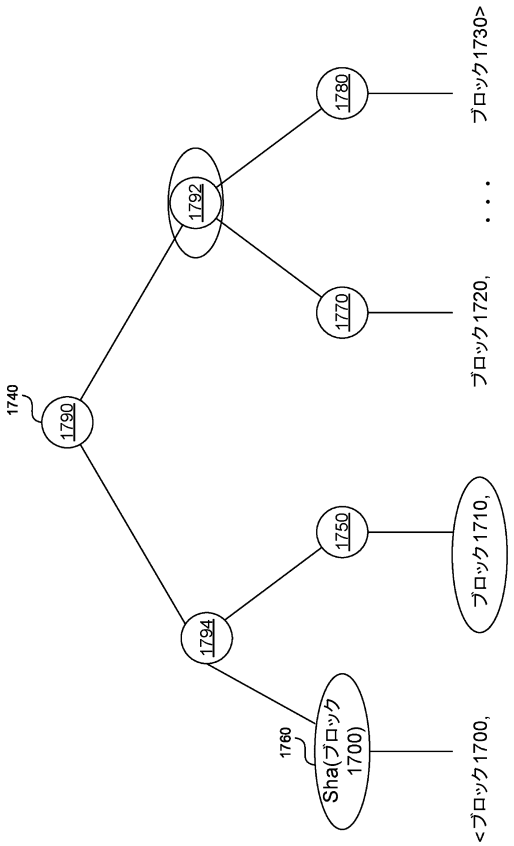
【図１６Ａ】



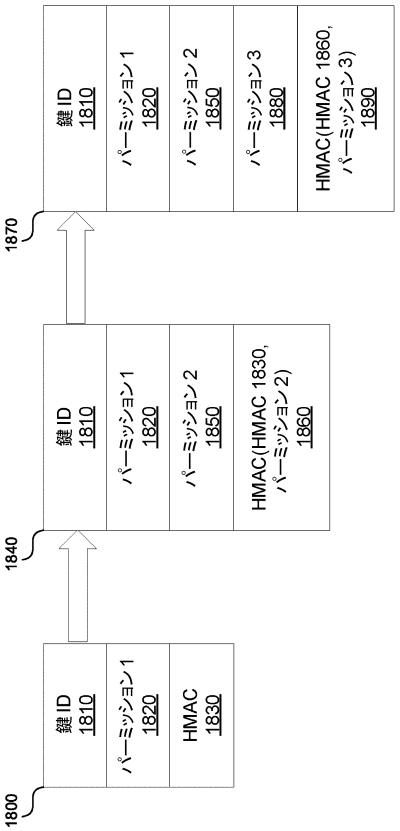
【図１６Ｂ】



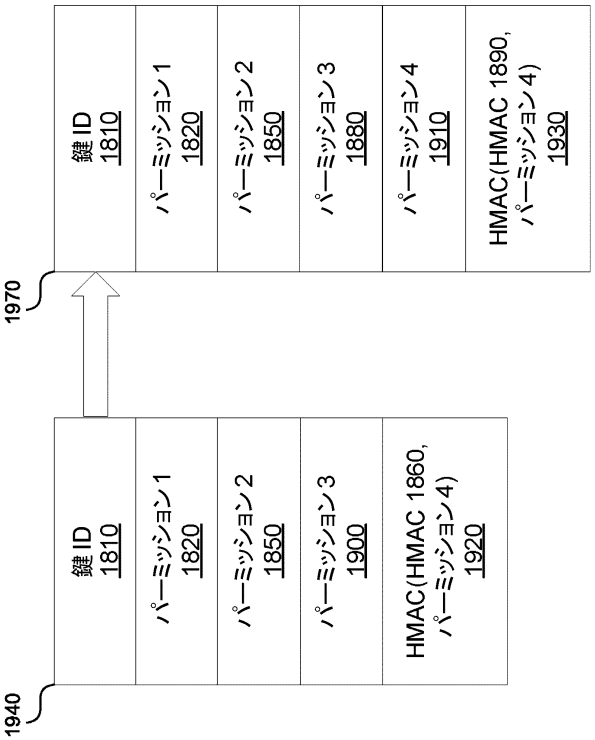
【図１７】



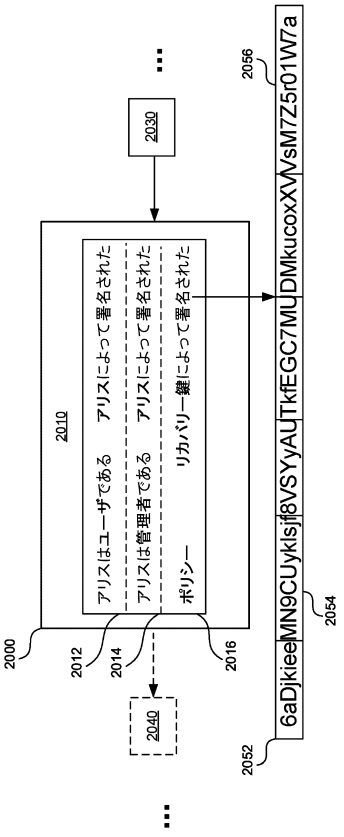
【図１８】



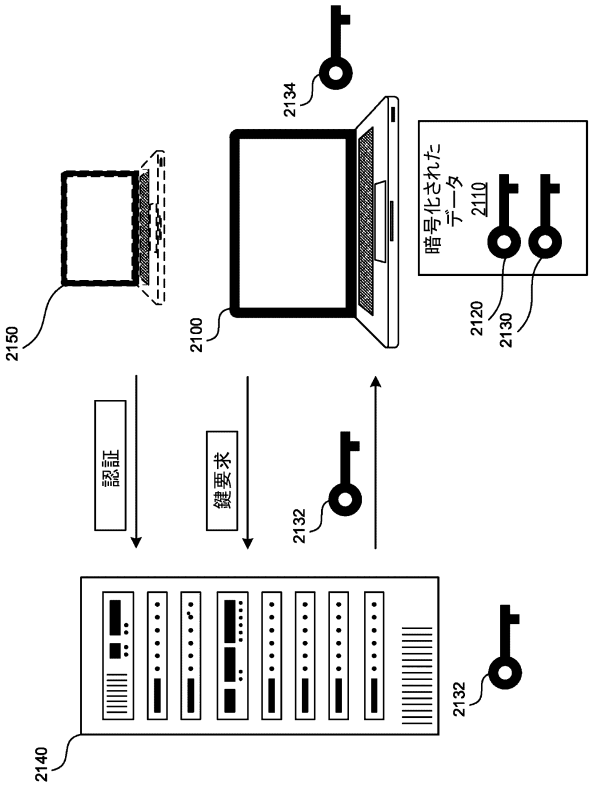
【図１９】



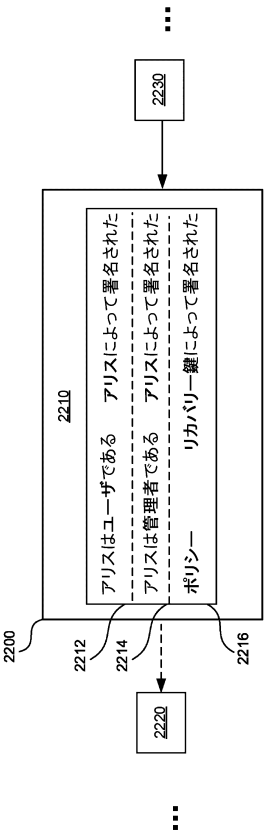
【図２０】



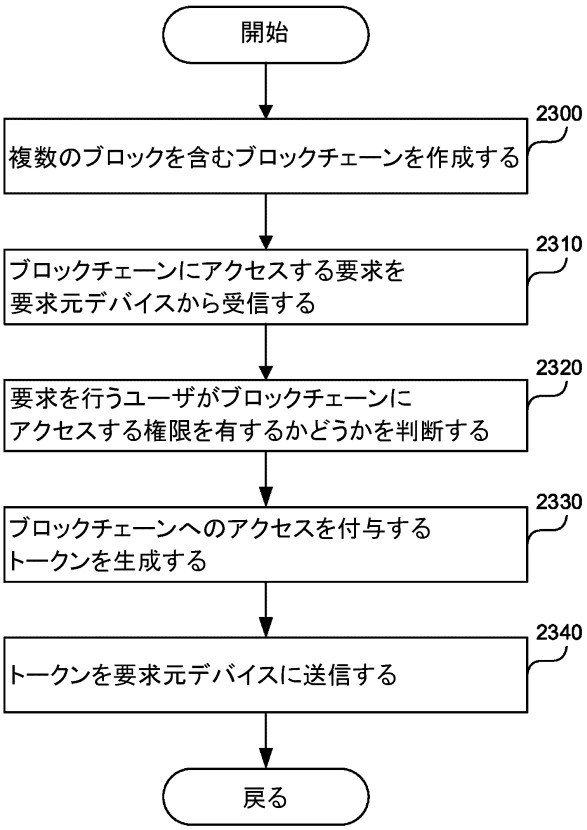
【図２１】



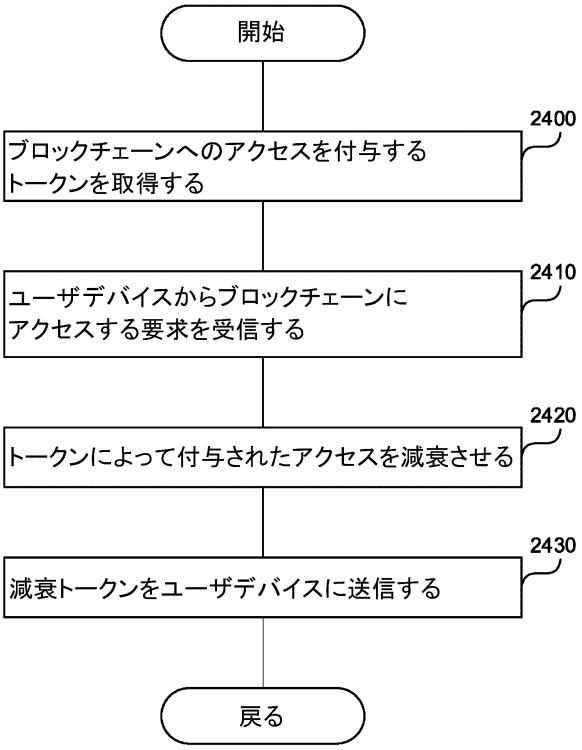
【図２２】



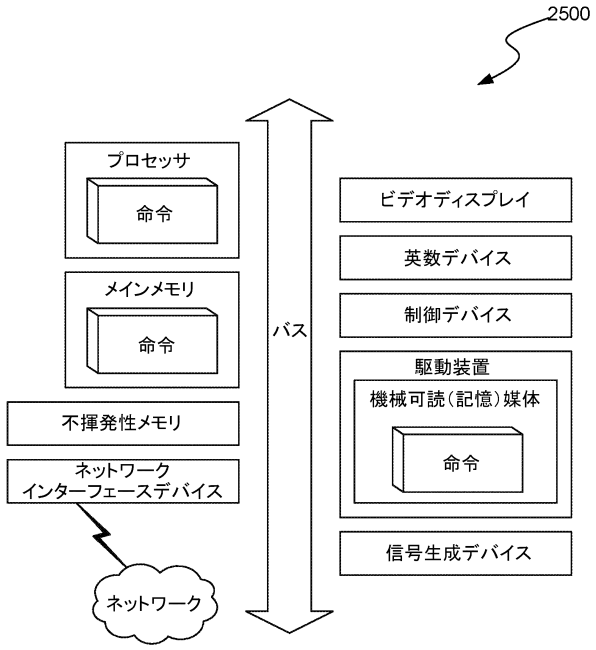
【図２３】



【図２４】



【図２５】


2500
プロセッサ
命令
メインメモリ
命令
不揮発性メモリ
ネットワーク
インターフェースデバイス
ネットワーク
バス
ビデオディスプレイ
英数デバイス
制御デバイス
駆動装置
機械可読（記憶）媒体
命令
信号生成デバイス

フロントページの続き

(56)参考文献　　米国特許出願公開第２０２０／００６７９０７（ＵＳ，Ａ１）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／００－８８