令和元年度「専修学校による地域産業中核的人材養成事業」
スマートコントラクトを使用したシステム開発人材の育成

ブロックチェーン概論

学校法人 麻生塾　麻生情報ビジネス専門学校

令和元年度「専修学校による地域産業中核的人材養成事業」
スマートコントラクトを使用したシステム開発人材の育成

ブロックチェーン概論

学校法人 麻生塾　麻生情報ビジネス専門学校

目次

環境構築

この教材で行う演習のための、環境の構築方法を説明します。

利用するツール

- Virtual Box
- Ubuntu
- Ruby

Virtual Box

Virtual Box とは使用している PC に仮想環境を構築して、他の OS をインストールすることができる仮想化ソフトです。Virtualbox を導入することにより、複数の OS を切り替えて使用することが可能になります。

インストール

- VirtualBox の公式サイトである、https://www.virtualbox.org/を開く
- 「Download VirtualBox 6.0」をクリックする
- ダウンロードを行った後は、VirtualBox のインストーラを開き、画面の指示に従い VirtualBox をインストールする

VirtualBox はバージョン 5.0 以上で問題なく演習を行うことができると確認できています。

Ubuntu

Ubuntu とは OS の１つであり、Linux の人気ディストリビューションです。

インストール

- Ubuntu のダウンロードページである、https://jp.ubuntu.com/download を開く
- Ubuntu Desktop 18.04.3 LTS のダウンロードボタンをクリックする
- Ubuntu のイメージのダウンロードが完了すると、VirtualBox で Ubuntu を起動する

Ruby

Ruby は日本人により開発が行われた、オブジェクト指向型のスクリプト言語です。

インストール

- 起動した仮想マシン上で端末アプリを開く
- `$ sudo apt-get install ruby irb` を実行する

以上で、演習に必要な環境構築は終了です。

1. ブロックチェーンの概要

ブロックチェーンは 2008 年に「Satoshi Nakamoto」によって発表された「Bitcoin: A Peer-to-Peer Electronic Cash System」という論文をベースとして研究と開発が進められた技術です。この論文で紹介されている技術が、現在のブロックチェーンの根幹になっています。

1.1 ブロックチェーンとは

1.1.1 分散型台帳

ブロックチェーンには様々な解釈があり、この条件を満たしたものをブロックチェーンと呼ぶといった厳密な定義は存在しません。そのため、ここではブロックチェーンの概要を掴むためにブロックチェーンは「**分散型台帳**」であるとして話を進めます。「台帳」とは商品の売買履歴などを記録しておく帳簿、取引記録のことを言います。

ブロックチェーンでは参加者により**ネットワーク**が形成され、その中で様々な取引が行われます。このネットワーク内で行われた取引の記録は取引を行なった当事者だけでなく、ネットワークの全ての参加者に共有されます。これにより、取引記録はネットワーク内に分散して保存されることになり、分散型台帳が形成されます。以下の図 1.1 にネットワーク内に取引記録を分散して保存する図を示します。

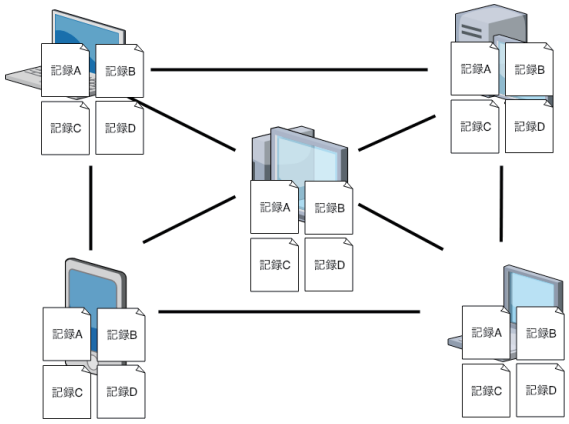


図 1.1 分散型台帳のイメージ

分散型台帳の仕組みはブロックチェーンができる以前から使われていたもので、決して新しい技術ではありません。

では、なぜブロックチェーンが注目されているのでしょうか。これは、ブロックチェーンにより実現された分散型台帳が「**管理者が存在せず、台帳の管理が参加者により自律的に行われる**」という大きな特徴を持つためです。これまで分散型台帳を作成するために利用されてきた一般的なデータベースシステムでは更新、削除などを行う際には操作に応じた権限が必要でした。これは管理者からの信頼がある人のみがデータベースを操作することで、データベースに対しての不正を防ぐことが目的です。

これに対して、ブロックチェーンでは台帳の管理者が存在せず、参加者ならば誰でも台帳に書き込む権利があります。さらに、このネットワークには誰でも参加することができます。この管理者が存在しない点がブロックチェーンが注目されている理由の１つです。

では、どうしてこれまでの技術ではこの仕組みを作ることができなかったのでしょうか。これは管理者を必要とせずに誰でも書き込むことができる台帳を作成するためには大きな課題があり、その解決が難しかったためです。課題とは「悪意のあるユーザーが不正な台帳操作を行う可能性がある」という点です。不正な台帳操作とは、実際には行われていない取引を書き込むことや、正しく書き込まれた記録を後から削除や改ざんすることなどが挙げられます。台帳にこのような不正が行われる可能性があると、台帳に対する信頼は失われ、正しく機能することはありません。

ブロックチェーンではこの問題を「**参加者が経済的合理性に基づいて行動することで、正しい記録のみが台帳に残る**」という仕組みを作ることで解決しました。ここで出てきた「経済的合理性」とは「経済的な価値基準に沿って論理的に判断した場合に、利益があると考えられる性質・状態」のことです。つまり、経済的合理性に基づいた行動とは、自身の得になるような行動のことを言います。まとめると、ブロックチェーンは**参加者が自身の利益を追求するための合理的な行動**をすることで正しく稼働し続けます。これまでの仕組みのように企業などがシステムを管理している訳ではありません。

1.2 ブロックチェーンの特徴

1.2.1 非中央集権

現在、私たちが利用しているシステムの多くは、そのサービスを提供している企業や団体が管理者として存在しています。管理者はシステムを正常に稼働させるためにシステムに対して**特別な権限**を持っています。

これに対して、ブロックチェーンは**管理者を必要とせず**に稼働します。これはブロックチェーンのネットワーク内で行われた取引の記録は、参加者全員に改ざんされることなく共有され、新たな取引が行われるとそれぞれの参加者が保有する過去の記録を参照し、これから行われる取引が正当なものであるかどうか正しく検証する仕組みになっているためです。このように管理者を必要とせず、参加者のみでシステムの維持管理をしている状態を「**非中央集権的な状態**」と言います。これに足して、管理者が存在し、管理者によりシステムが維持管理されている状態を「**中央集権的な状態**」と言います。

ブロックチェーンを用いた非中央集権的な取引の具体例として、A さんが所有している土地の権利を B さんに売却することを挙げます。土地の権利を売却する際には、A さんがいつ土地を取得し、現在も A さんが所有しているのかどうかを確認する必要があります。これは台帳に記録されている A さんの過去の取引記録からネットワーク内の全員が確認することができます。これにより土地の所有権は確実に A さんが持っていることを参加者それぞれが証明することができます。確認ができると権利の売却を行います。土地の権利の移転後にも所有権が B さんにあることが台帳に記録されるため、B さんが土地の所有者であることを参加者全員が証明することができます。

一方で、管理者が存在する中央集権的な従来の仕組みでも、取引が正当なものであるか確認するためには、過去に行われた取引記録の確認が行われます。ブロックチェーンと異なるのは、全ての取引記録を保存し、取引の正当性を確かめるのがシステムの管理者であるという点です。これは記録を安全に保存し、それを用いて正

しく取引の検証を行うためには、管理者がまとめて管理・検証する方法しか存在しなかったためです。

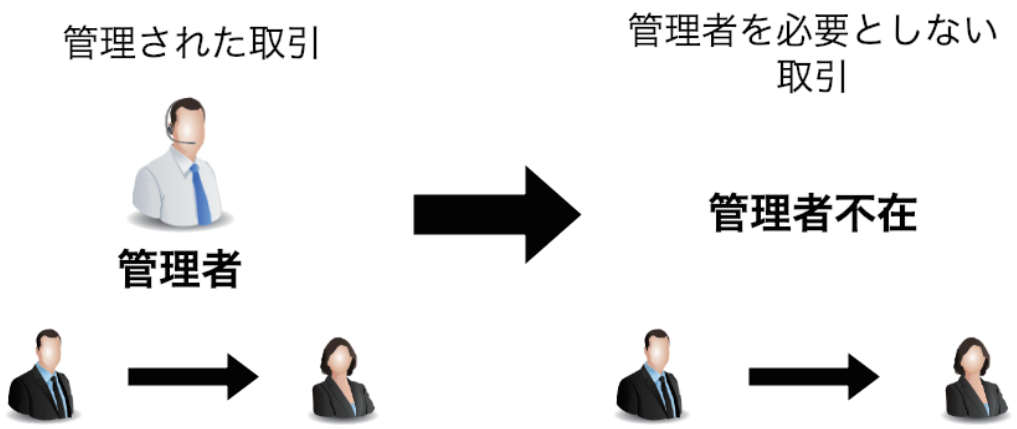


図 1.2 ブロックチェーンによる管理者不在の取引

1.2.2 セキュリティ

ブロックチェーンは、取引記録を保存するデータベースとも捉えることができます。ブロックチェーンをデータベースとして捉える時、従来のデータベースシステムに比べて、**改ざん耐性が高い**という特徴があります。この特徴はブロックチェーンの 2 つの性質により実現されています。

1 つ目にブロックチェーンのネットワーク内には取引記録の**多数の複製が存在**することが挙げられます。

2 つ目にブロックチェーンの**独自のデータ構造**と**データを記録するための仕組み**が挙げられます。データ構造については 3 章、データを記録するための仕組みについては 6 章で詳しく説明します。この 2 つの性質により、悪意のある参加者が台帳を書き換えることが極めて難しくなっています。

次に記録の紛失の可能性について説明します。ネットワーク内に多数の複製が存在することでサイバー攻撃などにより、記録が一度に失われる可能性は極めて低くなります。しかし、ブロックチェーンでは従来のデータベースシステムでは考える必要のなかった懸念があります。

ブロックチェーンは誰でもネットワークに参加することができ、逆にいつでもネットワークから抜けることができます。つまり、**誰にもシステムを維持し続けることや、記録を保存し続けることに対する責任はありません**。そのため、多数の参加者が悪意の有無に関わらず記録を紛失してしまうことや、ネットワークからの接続が途絶えてしまうことにより、ネットワーク上から記録が失われてしまう可能性があります。このようなことを防ぐためにブロックチェーンには、正しい記録を保存し続けるユーザーにはメリットがあるような仕組み作りがされています。この仕組みを実現する上で重要となっているのが、「**コンセンサスアルゴリズム**」です。コンセンサスアルゴリズムについては 6 章で詳しく説明します。

1.3 ブロックチェーンの種類

ブロックチェーンはネットワークに数万人が参加する規模の大きなものから、企業の内部で運営される小さなものまで数多く存在しています。そのため、全ての特徴を詳細に把握することは難しいですが、ブロックチェーンを大きく 2 種類に分類することはできます。1 つは誰でもネットワークに参加することのできる「**パブリックブロックチェーン(Public Blockchain)**」、もう 1 つは許可を受けた人のみがネットワークに参加することのできる「**パーミッションドブロックチェーン(Permissioned Blockchain)**」です。それぞれの特徴について説明します。

1.3.1 パブリックブロックチェーン

パブリックブロックチェーンとはパブリックという言葉の通り、許可や使用するコンピュータの性能などに関係なく誰でも参加することのできるブロックチェーンです。

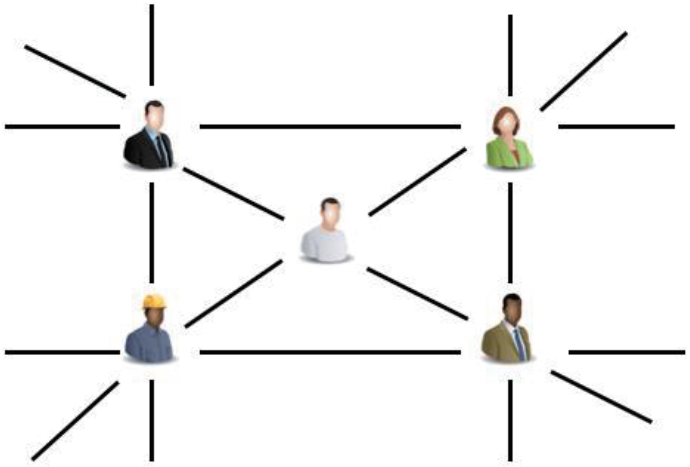


図 1.3 パブリックブロックチェーンのイメージ

特徴

管理者が存在しない

パブリックブロックチェーンには、システムを管理する企業や団体などの**管理者は存在しません**。あらかじめ作成されたプロトコルに従い、**参加者のみで自律的に運営**されます。

誰でも内容を閲覧することができる

パブリックブロックチェーンに記録されている取引の内容は、**誰でも閲覧することができます**。ネットワークに参加していない人でも、参加者に問い合わせることで内容を知ることができます。

これによりブロックチェーン上では**透明性の高い取引**を行うことができます。透明性が高いことで気になるのがプライバシーの問題ですが、ブロックチェーンでは記録された取引記録から個人を特定することはできないため、最低限のプライバシーは守ることができます。

パブリックブロックチェーンの１つである Bitcoin のネットワーク上で行われた全ての取引は、以下の Web サイトから確認することができます。

ブロックチェーンエクスプローラ（https://www.blockchain.com/ja/explorer）

利点

管理者による影響を受けない

管理者が存在するシステムでは、管理者がシステムの運営を辞めること、管理者の破綻によるシステムの停止、管理者による仕様の変更などがユーザーの意思に関わらず行われることがあります。このように管理者が存在するために**ユーザーが不利益を被る**ことがあります。

これに対して、パブリックブロックチェーンは**管理者が存在せず**、システムはユーザーによって**自律的に**管理されているため、ユーザーがいる限りシステムは動き続け、独断による仕様の変更はありません。

記録の改ざんが難しい

ブロックチェーンは改ざん耐性が高いという特徴を持つと説明しました。その中でもパブリックブロックチェーンは特に**改ざん耐性が高い**ことが特徴です。これはパブリックブロックチェーンは誰でも参加することができるため、参加者の限定されるパーミッションドブロックチェーンに比べて、多くのコンピュータによりシステムが支えられているためです。

課題点

仕様の変更が難しい

パブリックブロックチェーンにはシステムの管理者が存在せず、参加者全員が同じ権限を持ちます。そのため、仕様の変更の際には**多数の参加者による同意が必要**になります。管理者なしに多数の同意を得ることは難しく、仕様の変更には大きな労力が必要になります。特にこれまでと互換性のない仕様の変更の際には、大きな問題が発生することがあります。

安全な取引を行うためには時間がかかる

パブリックブロックチェーンでは誰でも台帳に記録を書き込むことができるという性質から、悪意を持ったユーザーがネットワークに紛れ込む可能性があります。その問題を解決するために用いられているのが「コンセンサスアルゴリズム」です。パブリックブロックチェーンで用いられるコンセンサスアルゴリズムには「Proof of Work」や「Proof of Stake」と呼ばれるものがあります。これらのアルゴリズムは、**時間の経過とともに台帳上には正しい記録のみが残る**仕組みになっています。そのため、安全な取引を行うためには時間が必要になり、即時性の必要な処理には向いていません。

ブロックチェーンへの参加

パブリックブロックチェーンへ参加するためには、そのブロックチェーン専用のソフトウェアを入手する必要があります。パブリックブロックチェーンの場合には、Web サイトから簡単に入手することができます。

Bitcoin を例に用いて説明します。Bitcoin には Bitcoin の規格に準拠したソフトウェアが複数あり、その中の 1 つに BitcoinCore があります。BitcoinCore は以下の Web サイトから入手することができます。

Bitcoin Core(https://bitcoin.org/ja/download)

Bitcoin に参加するためのソフトウェアは Bitcoin Core だけでなく多くの種類があります。その中でも最も利用されているのが Bitcoin Core です。2018 年 12 月現在で、Bitcoin ネットワーク内の参加者の 90%以上が Bitcoin Core を使用しています。これはプログラムの安定性や透明性、セキュリティ、機能などが評価されているためです。

1.3.2 パーミッションドブロックチェーン

パーミッションドブロックチェーンとは、ネットワーク内に管理者が存在し、ネットワークへの参加に管理者の許可が必要となるブロックチェーンです。企業や団体などの組織内や組織間での利用が想定されています。パーミッションドブロックチェーンはブロックチェーンを管理する主体の数によって、「**プライベートブロックチェーン(Private Blockchain)**」と「**コンソーシアムブロックチェーン(Consortium Blockchain)**」の 2 種類に分類することができます。

プライベートブロックチェーン

プライベートブロックチェーンとはブロックチェーンを**管理する主体が 1 つ**であるパーミッションドブロックチェーンです。企業や団体内での利用が想定されます。

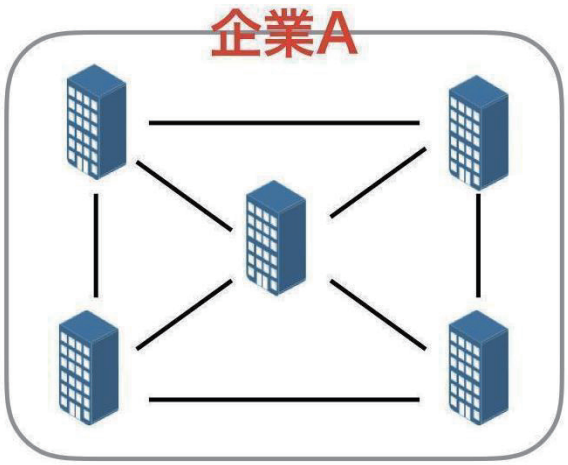


図 1.4　プライベートブロックチェーンのイメージ

特徴

書き込みには権限が必要

　プライベートブロックチェーンでは、**管理者のみ**が取引の正当性を判断し、ブロックチェーンに対して書き込みを行うことができます。

ブロックチェーンの閲覧権

　プライベートブロックチェーンに記録されている内容の閲覧は、**管理者が定めた範囲内**で行われます。そのため、公開範囲はブロックチェーンにより異なります。この閲覧制限により取引の秘匿性を高めることはできますが、透明性は低くなります。

利点

仕様の変更が容易

　従来の中央集権的な仕組みと同様に、管理者によりブロックチェーンのシステムの管理が行われているため、**管理者の判断でシステムの仕様を変更**することができます。

取引にかかる時間が短い

　ブロックチェーンへの取引記録の書き込みは管理者のみが行うため、パブリックブロックチェーンで用いる時間の経過により記録の安全性が高まるようなコンセン

サスアルゴリズムを用いる必要がありません。そのため、安全に取引を行うために必要な**時間を短く抑える**ことができます。

課題点

システムが管理者に強く依存する

管理者が存在するシステムであるため、管理者がシステムの運営を辞めることや、管理者の破綻などによりシステムが停止する可能性があるなど、システムが**管理者に依存**してしまうという特徴があります。

改ざんのリスク

プライベートブロックチェーンでは、取引記録を複数のコンピュータに分散させて保存するものの、Bitcoin などのパブリックブロックチェーンに比べるとネットワーク内の**コンピュータの数が少ない**ため、複数のコンピュータが共謀して不正を行うと、正しい取引記録が失われてしまうことや、不正な取引が承認されてしまう可能性があります。

コンソーシアムブロックチェーン

コンソーシアムブロックチェーンとは、ブロックチェーンを**管理する主体が複数存在**するパーミッションドブロックチェーンです。複数の企業や団体間での利用が想定されます。

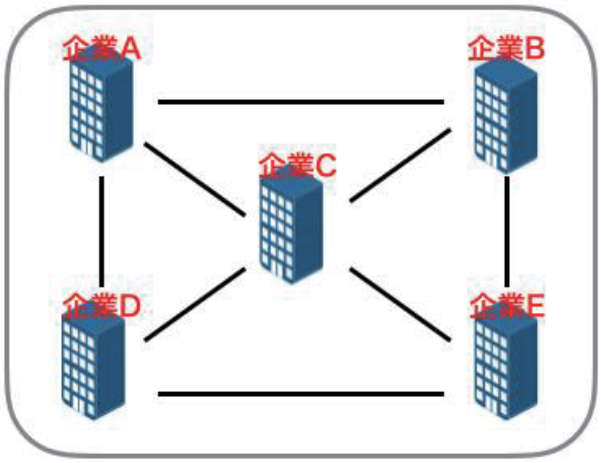


図 1.5　コンソーシアムブロックチェーンのイメージ

特徴

書き込みに制限がある

ブロックチェーンへの書き込みを行うことができるのは、プライベートブロックチェーンと同様に**管理者のみ**です。しかし、プライベートブロックチェーンと異なり、コンソーシアムブロックチェーンには複数の管理者が存在するため、書き込みの際には正しい記録が書き込まれるようにコンセンサスアルゴリズムが用いられます。

取引にかかる時間

複数の管理者が存在することから、取引の書き込みの際に管理者内での合意を得る必要があるため、プライベートブロックチェーンと比べると取引にかかる時間は長くなります。また、パブリックブロックチェーンと比べると、合意を得る必要のあるコンピュータの数が少なく、またブロックチェーンへ書き込みのできる管理者は比較的信頼できることから使用するコンセンサスアルゴリズムも簡易なものとなり、取引にかかる時間は短くなります。

今回紹介したパブリックブロックチェーンとパーミッションドブロックチェーンの特徴を以下の表 1.1 にまとめます。

	パブリックブロックチェーン	パーミッションドブロックチェーン
参加	誰でも参加することができる	管理者の許可が必要
記録の書き込み	コンセンサスアルゴリズムを用いて記録を行うノードを決める	管理者が行う
記録の閲覧	誰でも閲覧できる	管理者が閲覧権限を定める
処理速度	比較的遅い	比較的速い

表 1.1 パブリックブロックチェーンとパーミッションドブロックチェーンの比較

1.4 ブロックチェーンの実例

2009 年に Bitcoin の運用が始まって以降、世界中で様々なブロックチェーンが運用されています。それぞれのブロックチェーンには特徴があり、期待される役割が異なります。ここでは 3 つのブロックチェーンを紹介します。

1.4.1 Bitcoin(ビットコイン)

Bitcoin は 2008 年に「Satoshi Nakamoto」により発表された論文をベースとして、2009 年から運用が始まったブロックチェーンです。また、初めて稼働したブロックチェーンシステムでもあります。Bitcoin が開発される以前は、インターネット上で使用する通貨であるデジタル通貨には、同じ通貨を 2 度使用する不正方法である「二重支払い問題」が存在し、これを防ぐために管理者が全ての取引を監視する必要がありました。Bitcoin の誕生により初めて**管理者を必要とせずに**デジタル通貨の取引を二者間のみで安全に行うことができるようになりました。このことから Bitcoin は**価値の移転**に特化したブロックチェーンであると言われます。

1.4.2 Ethereum(イーサリアム)

Ethereum は Vitalik Buterin（ビタリック・ブテリン）が中心となり、2015 年にリリースされたブロックチェーンです。Ethereum は「**管理者を必要としない分散型アプリケーションを作成するプラットフォーム**」としての役割を持ちます。Ethereum ネットワークを構成する多数のコンピュータにより、1 つの大きなコンピュータのように振る舞うことができる仮想マシン(EVM、Ethereum Virtual Machine)が構築され、その上で DApps(Decentralized Applications、分散自律型アプリケーション)」を動かすことができます。このことから Ethereum は「ワールドコンピュータ」と呼ばれます。アプリケーションのコードや、DApps の実行結果はブロックチェーンに記録されます。DApps については 8 章で詳しく説明を行います。

1.4.3 Hyperledger（ハイパーレッジャー）

Hyperledger は具体的なブロックチェーンの名前ではなく、ブロックチェーン技術を様々な分野で最大限に利用することを目的として生まれたブロックチェーン技術の推進コミュニティーです。Hyperledger には様々なプロジェクトが存在しており、そのプロジェクトの１つに「**Hyperledger Fabric**」があります。Hyperledger Fabric とは、企業間や組織内でブロックチェーンを利用するために作られた**オープンソースのブロックチェーンプラットフォーム**です。プライベートブロックチェーン、コンソーシアムブロックチェーンとして利用することが想定されています。

1.5 ブロックチェーンの活用例

ここからは実際にブロックチェーンが社会でどのように用いられているのか、活用例を紹介します。

1.5.1 仮想通貨

ブロックチェーンの最大の活用例として挙げられるのが仮想通貨です。ブロックチェーンにより、これまでは管理者による取引の管理が必要だったデジタル通貨の取引を管理者に依存せず、自律的に行うことができるようになりました。仮想通貨については 2 章で詳しく説明を行います。

1.5.2 不動産取引

不動産の取引にブロックチェーンを用いることで、仲介者である不動産業者などの第三者を必要とせずに安全に取引ができるようになります。また、不動産の所有権をブロックチェーンに記録することで、ブロックチェーンの高い改ざん耐性を利用して安全に管理することができます。

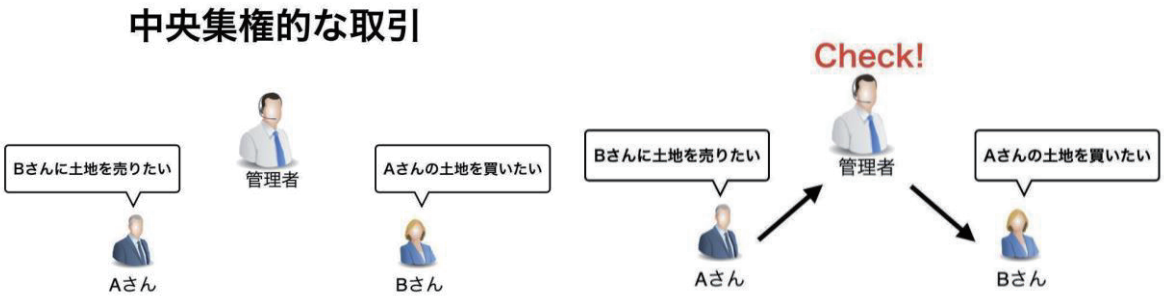


図 1.6　中央集権的なこれまでの不動産取引

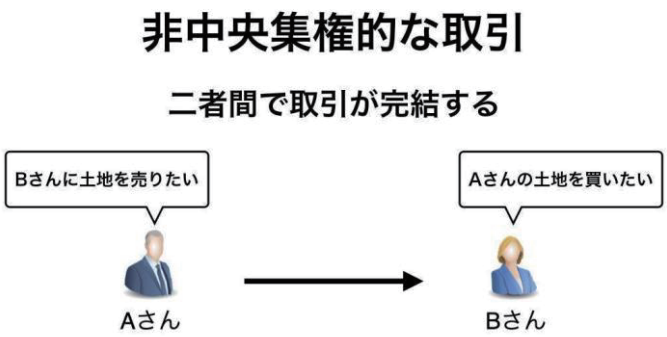


図 1.7　非中央集権的な不動産取引

1.5.3 食品管理

食品の流通経路を 1 つのブロックチェーンに記録していくことにより生産地や、輸送業者、加工者などを消費者が簡単に知ることができます。万が一、食品に問題が発生した際にも発生源の特定を迅速に行うことができます。また、農産物に使用された農薬や土壌についての情報をブロックチェーンに記録することで、生産者は自身の商品の安全性の高さを証明することもできます。

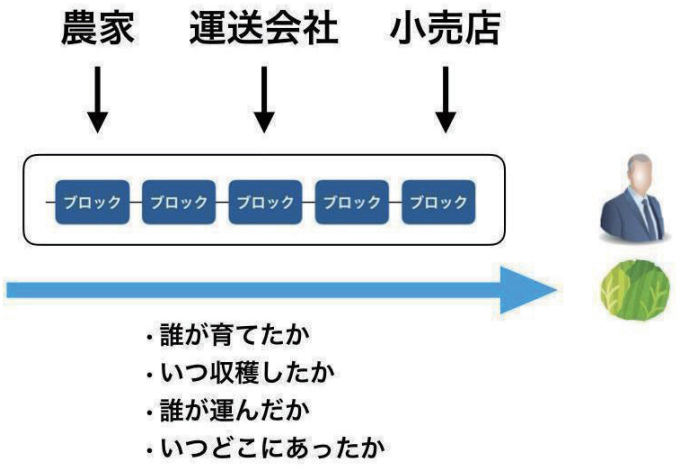


図 1.8 ブロックチェーンによる流通経路の管理

1.5.4 医療データの管理

電子カルテや処方箋の情報をブロックチェーン管理することで、どこの病院にいった場合にも過去の怪我や病気の記録、処方された薬の情報を医者は正確に知ることができます。また、患者の情報以外にもブロックチェーンの高い改ざん耐性を利用して、新薬の治験データの改ざんの防止や、遺伝子データなど複雑な利権の絡むデータの管理などにも利用されることが期待されています。

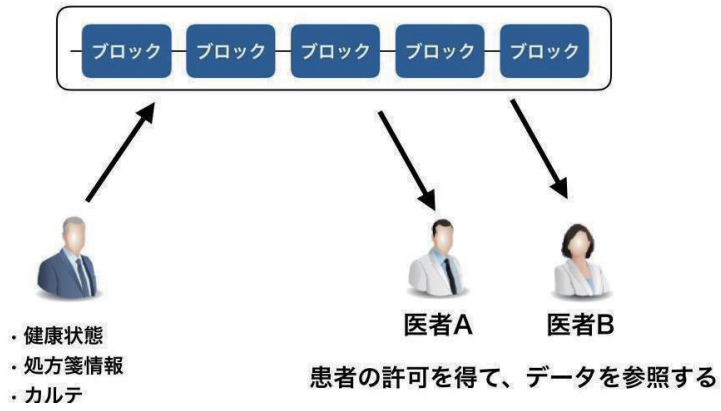


図 1.9 ブロックチェーンによる医療データの共有

以上が、ブロックチェーンの概要です。ブロックチェーンの特徴についてしっかりと理解することでブロックチェーンに対するイメージを掴み、以降のブロックチェーンの技術的な理解の手助けとしてください。

2. ブロックチェーンと仮想通貨

2017 年の年末に多くの仮想通貨の価格が高騰し「仮想通貨バブル」と呼ばれる現象が発生しました。この時に仮想通貨の投資により億単位の利益を出した人が「億り人」と呼ばれるなど、投機的な面に大きな注目が集まることにより「仮想通貨」という言葉は広く知られるようになりました。しかし、仮想通貨という言葉自体は知っている人は多くとも、その仕組みや特徴などの詳細まで理解している人はその中のわずかであると言えます。

そこで、この章では仮想通貨についての理解を深めるために、その概要について説明します。

2.1 仮想通貨

まずは、仮想通貨とブロックチェーンの関連について説明します。現在、大半のパブリックブロックチェーンでは、あらかじめ定めた機能を参加者により自律的に満たすためにシステムの構成要素の１つとして仮想通貨が利用されています。つまり、ブロックチェーンシステムの一部として仮想通貨は機能します。

仮想通貨はこれだけでなく、ブロックチェーンシステム上で作られるアプリケーションの１つの場合もあります。

つまり、ブロックチェーンと仮想通貨には密接な関係があり、仮想通貨について理解することはブロックチェーンについての理解を深めるために重要です。

2.1.1 仮想通貨の性質

まずは仮想通貨の定義を紹介します。以下は、日本国内で定められている仮想通貨の定義です。

改正資金決済法第 2 条第 5 項

物品を購入し、もしくは借り受け、または役務の提供を受ける際に、これらの代価の弁済のために不特定多数の者に対して使用することができ、かつ、不特定の者を相手方として購入及び売却を行うことができる財産的価値であり、電子情報処理組織を用いて移転することができるもの。または、不特定の者を相手方として相互に交換を行うことのできる財産的価値であって、電子情報処理組織を用いて移転することができるもの。

この定義には難しい語句や説明が多く、初見で理解するのは難しいのではないかと思います。そこで、ここでは仮想通貨の特徴を 3 つ取り上げます。

管理者が存在しない

仮想通貨には通貨の発行や、取引の監視などを行う企業や団体などの**管理者は存在しません**。あらかじめ作成されたプロトコルに従い、参加者により**自律的にシステムの維持管理**が行われます。

インターネット上で取引される

全ての仮想通貨は紙幣や硬貨のように紙や金属などの物質として存在せず、コンピュータ上での「データ」という形で存在しています。このデータを世界中のコンピュータでやりとりすることによって、仮想通貨の取引は行われます。そのため、仮想通貨を使用するためにはインターネットが必要不可欠です。

世界中で使うことができる

日本円やアメリカドルなどの通貨は、利用することのできる範囲が限定されています。これに対して、仮想通貨は使用できる国や地域が限定されておらず、世界中で使用することができます。ただし、仮想通貨の使用にはインターネットが不可欠であるため、インターネット環境がない場所や、インターネットに接続するデバイスを持っていない際には使用することができません。

2.1.2　仮想通貨の種類

仮想通貨には数多くの種類が存在し、2018 年 12 月の段階で 1600 種類以上あると言われています。

以下の表 2.1 に 2018 年 12 月 6 日時点で時価総額が大きい仮想通貨の上位 10 種類を紹介します。

	時価総額(円)
ビットコイン (Bitcoin)	7,396,220,465,748
リップル (XRP)	1,536,289,751,325
イーサリアム (Ethereum)	1,200,188,043,825
ステラー (Steller)	292,616,948,375
ビットコインキャッシュ (Bitcoin Cash)	253,845,422,208
イオス (EOS)	224,787,444,370
テザー (Tether)	208,425,647,414
ライトコイン (Litecoin)	196,652,935,720
ビットコインSV (Bitcoin SV)	175,060,586,752
トロン (TRON)	104,458,753,761

表 2.1 時価総額の大きい仮想通貨(2018 年 12 月 6 日時点)

Bitcoin

ここでは、仮想通貨の中で最も時価総額が大きいBitcoinについて詳しく紹介します。Bitcoinは**ブロックチェーン技術が採用された初めての仕組み**であり、2009年に運営が始まりました。「Bitcoin」という言葉には幅広い意味があり、システムの名前、通貨の名前、通貨の単位に使われます。通貨としてのBitcoinの基本単位は「BTC」と書き「ビットコイン」と読みます。また、「BTC」の1億分の1の単位に「satoshi」が存在し、これがBitcoinの最小単位となっています。

Bitcoinの価値は一般的な通貨と同様に需要と供給の関係で常に変動しています。2017年末の仮想通貨バブルと呼ばれた時期には1BTCあたり240万円以上に高騰しましたが、2018年12月には約40万円となっています。

Bitcoinは通貨の発行量が過剰になることで需要と供給の関係により、通貨の価値が低下すること防ぐために、通貨の総発行量は2100万BTCとあらかじめ定められています。2018年12月までに約1750万BTCが発行されており、これは総発行量の80%以上に相当します。総発行量に達するのは2140年頃になるとされています。

2.2 仮想通貨の特徴

2.2.1 法定通貨と仮想通貨

私たちの生活で馴染みのある日本円やアメリカドルなどをはじめとした国や地域が発行する通貨は「法定通貨」と呼ばれます。ここでは法定通貨と仮想通貨のそれぞれの特徴を比較しながら、仮想通貨についてさらに理解を深めます。

表2.2に仮想通貨と法定通貨の異なる点を紹介します。

	法定通貨	仮想通貨
発行者	中央銀行	プログラム
管理者	中央銀行	参加者
通貨の新規発行	経済状況を加味	一定時間ごとに 一定額
信頼	国に対する信頼	技術に対する信頼

表 2.2 法定通貨と仮想通貨の比較

発行者

法定通貨はその通貨が使用される国や地域が発行を行う仕組みになっています。これに対して、仮想通貨は特定の組織が発行するのではなく、あらかじめ定められた仕様に従い、**プログラムにより自動的に発行**されます。

通貨の管理者

法定通貨は通貨の発行者である国により管理が行われています。これに対して、仮想通貨には特定の管理者は存在せず、仮想通貨の**システムの参加者により管理**が行われています。

通貨の新規発行

法定通貨は国の経済状況や流通している通貨量などを考慮し、国により発行量が決められます。これに対して多くの仮想通貨は通貨の価格などに関係なく**一定時間毎**に、**あらかじめ定められた額**が発行されます。

信頼

通貨の価値は、通貨の発行主体に対する信頼により担保されます。法定通貨は、発行主体である国に対する信頼を担保に価値を持っています。そのため、発行主体

である国の経済状況などにより通貨の価値が変動します。また、通貨の発行主体である国が破綻するとその国が発行した通貨は価値を持たなくなります。

これに対して、仮想通貨はその仕組みを実現している**ブロックチェーンを中心とする様々な技術に対する信頼**を担保に価値を持っています。

2.2.2 仮想通貨と電子マネー

ここでは仮想通貨と比較されることの多い「電子マネー」に注目し、仮想通貨との共通点と相違点を確認しながら、仮想通貨についてさらに理解を深めます。

まずは、仮想通貨と電子マネーの共通点を説明します。

取引情報が記録される

仮想通貨と電子マネーは実際に硬貨や紙幣などの物質の交換を行うのではなく、通貨の移転の際に「誰から誰にどれだけの通貨が移転する」という**取引の記録が台帳に記録**されます。

次に、仮想通貨と電子マネーの相違点を紹介します。表 2.3 に仮想通貨と電子マネーの相違点をまとめていますので、順に確認します。

	電子マネー	仮想通貨
法定通貨との関係	依存	独立
管理者	管理会社	参加者

表 2.3 電子マネーと仮想通貨の比較

法定通貨との関係

電子マネーは企業から購入し、保有しているポイントを消費することにより、サービスや商品を提供してもらうことができる仕組みです。法定通貨に対してのポイントの価値は、電子マネーの仕組みを提供している企業が変更しない限り変わることはありません。具体的には、ある電子マネーは 1 ポイントにつき、1 円の価値を持つときに、円の価値が変動したとしても、1 ポイントと 1 円の関係は変動することはありません。つまり、電子マネーは法定通貨に依存するし、支払いを効率的に行うための「**代替通貨**」であると言うことができます。

これに対して、仮想通貨は法定通貨からは完全に独立した通貨です。BTC と円、アメリカドルの価値の関係は常に変動し続けており、日本円やアメリカドルなどの**法定通貨への依存関係はありません**。

しかし、一部の仮想通貨は価格の安定を図ることが目的で、法定通貨と価格を連動させている通貨もあります。このような仮想通貨を StableCoin(ステーブルコイン)と呼びます。

管理者の有無

仮想通貨には特定の管理者は存在せず、仮想通貨のシステムは**参加者により管理**されています。

これに対して、電子マネーにはそのシステムを提供する企業が存在し、その企業が全ての取引を管理することにより安全な取引を行うことができます。

2.2.3 仮想通貨の入手方法

仮想通貨は法定通貨と同様に「通貨」であるため、サービスや商品の対価として支払うことが可能です。しかし、2018 年現在、日本で仮想通貨での支払いはまだ一般的ではなく、オンラインや店頭での支払いに仮想通貨を使用できるのはごくわずかです。そのため、日常生活を送る上で仮想通貨を扱うことや仮想通貨を持っていないことで困ることはありません。

このような理由から、仮想通貨に興味はあるものの、その入手方法を知らない人も多いのではないかと思います。そこで、ここでは仮想通貨の入手方法を紹介します。

譲り受ける

仮想通貨の最も簡単な入手方法として、仮想通貨を既に所有している人から譲り受ける方法があります。譲り受ける際には「アドレス」と呼ばれる銀行の口座番号のようなものを準備し、そのアドレスを相手に教えて仮想通貨を振り込んでもらいます。アドレスは専用のアプリや Web サービスから簡単に作成することができます。

仮想通貨取引所で交換する

仮想通貨取引所では仮想通貨と法定通貨の交換を行うことができます。つまり、日本円で仮想通貨を買うことができます。また、法定通貨と仮想通貨の交換だけでなく、仮想通貨同士の交換を行うことも出来ます。

ただし、仮想通貨取引所では全ての仮想通貨を取り扱っているわけではなく、取引所によって扱っている仮想通貨の種類は異なります。そのため自身が欲しい仮想通貨を扱っている取引所を探してから取引を行う必要があります。また、取引所を利用するためには、身分証などによる個人の照会などが必要になるため、譲り受けることに比べて、少しハードルが高いと言うことができます。

マイニングを行う

「マイニング」という方法で仮想通貨を入手することができます。この方法はここまでに紹介した 2 つの方法とは異なり、既に流通している仮想通貨を得るのではなく、ブロックチェーンの運営に貢献することで、その対価として新規に発行される仮想通貨を受け取る方法です。マイニングはブロックチェーンの根幹となる重要な仕組みであり、6 章で詳しく説明します。

Let's TRY

２章演習

2019 年現在、日本国内で仮想通貨により支払いを行うことができる店舗はごくわずかであり、一般的な支払い方法として普及しているとは言えません。

仮想通貨での支払いや仮想通貨自体が普及しない理由を 4～5 人のグループを作り、以下の順序で考えてください。

1. 普段の生活でどのような決済方法をよく使っているか
2. なぜ、その決済方法を利用しているのか
3. その決済方法と仮想通貨による決済の違いはどのような点であるか
4. 仮想通貨決済が進まない最大の理由はなんだと考えるか

4 つの項目についてグループで話し合った結果を代表者が以下のように発表してください。

1. 普段の生活でどのような決済方法をよく使っているか　→　**現金**
2. なぜ、その決済方法を利用しているのか　→　**他の決済方法に比べて～～～であるから**
3. その決済方法と仮想通貨による決済の違いはどのような点であるか　→　**他の決済方法は～～～であるが、現金は～～～である。ただし、～～～という難点もある。**
4. 仮想通貨決済が進まない最大の理由はなんだと考えるか　→　**仮想通貨の～～～という特徴により、～～～という問題があるため普及しないのではないか**

2.3 ICO と STO

仮想通貨の利用方法として大きな注目を浴びているのが資金調達です。
ここでは仮想通貨を用いた新しい資金調達の方法である ICO(Initial Coin Offering)と STO(Security Token Offering)について説明します。

2.3.1 ICO

ICO とは日本語で「新規仮想通貨公開」と呼ばれ、起業や新たな事業を始める際の資金調達を行うための仕組みやその行為のことを言います。**企業は独自の仮想通貨を発行し、投資家にそれを売り出す**ことで資金を集めます。この時、企業が発行する仮想通貨のことを「トークン」と呼び、ICO で売り出されたトークンは、出資した企業や事業により提供されるサービスに利用することができます。また、事業が成長しサービスのユーザーが増えるとトークンの価値が上がり、投資家はトークンの売却により利益を得ることもできます。

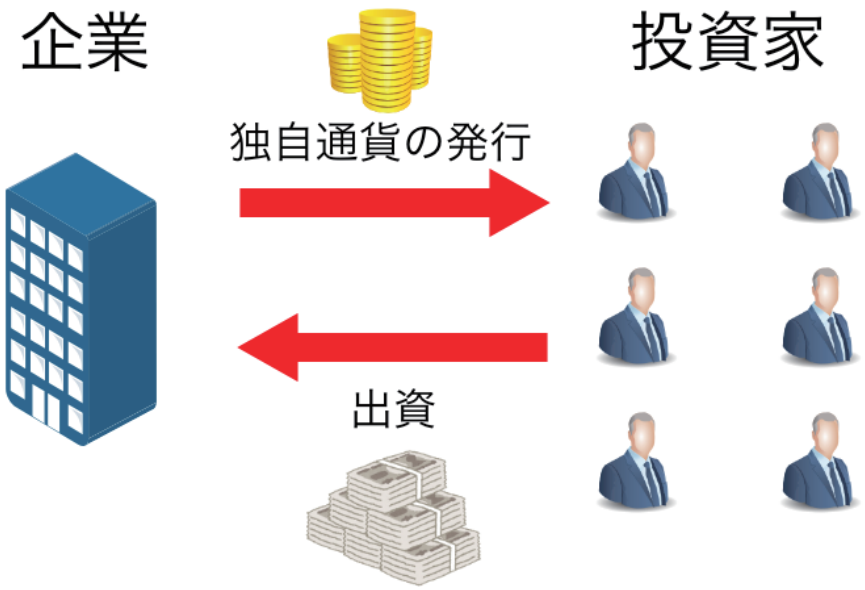


図 2.1 ICO のイメージ

ICO の元となった資金調達の方法に「IPO(Initial Public Offering)」があります。IPO とは日本語では「新規公開株」と呼ばれ、「企業が株を売り出し、証券取

引所に上場することで誰でも株の取引ができるようにする」仕組みです。これは既に多くの企業により使われている仕組みです。ここでは ICO と IPO を比較し、ICO のメリットとデメリットについて企業側、投資家側の視点から説明します。

企業側のメリット

第三者を通すことなく資金調達が可能である

ICO では証券会社などの**第三者機関を通すことなく資金の調達が可能**であり、仲介者を省くことにより取引の手数料を抑えることができます。

また、証券取引所で株を扱ってもらうためには、企業に一定の実績や規模が求められるため、小さな企業や個人では上場は難しいのが現状です。しかし、ICO では企業や事業の大小に関わらず**誰でも資金を調達することができます**。

投資家側のメリット

小さな企業、個人にも投資することができる

証券取引所への上場が難しい**小さな企業**や、**個人にも直接投資することができます**。

少額の投資が可能

IPO に比べ、**少額から投資**を行うことができるため、手軽に投資することができます。

企業側のデメリット

企業のイメージ低下の可能性

ICO はまだ**社会に浸透した仕組みであるとは言えず**、ICO を行うことにより企業のイメージが低下する可能性があります。

意図しない違法行為を行う可能性

2018 年現在、ICO に関する**法律は整備されていない部分が多い**のが現状です。また ICO 自体が法律的にグレーな部分が多いことから法律や税務に関連した予期しないトラブルに巻き込まれる可能性があります。

投資家側のデメリット

詐欺まがいの ICO が存在する

ICO は第三者機関による審査がなく誰でも行うことができることから、集めた**資金の持ち逃げや詐欺まがいの ICO も存在**します。2018 年の段階では ICO で資金は集めたものの事業が始まらなかったり、そもそも事業を行う予定がなかったりするなどといったこともありました。これは ICO に関する法律がまだ未整備である事が原因に挙げられます。投資を行う際には企業が出す事業の計画書をよく読み、自身の責任で行う必要があります。

2.3.2 STO

STO の頭文字である、「Security」には「証券」という意味があります。STO は ICO と異なり、**発行されるトークンが証券としての条件を満たします**。トークンが証券としての条件を満たすためには、以下に示す「Howey Test」の要件を全て満たす必要があります。

- 金銭または資産による投資であること
- 共同事業に対する投資であること(経営者、投資家など複数人が関係していること)
- 利益に対する期待があること(寄付は該当しない)
- 利益は第三者の努力により生じるものであること

これらの条件を満たしたトークンは「Security Token」と呼ばれ、証券と同様に扱われます。これに対して、ICO で発行される証券としての条件を満たさないトークンは「Utility Token」と呼ばれます。「Utility」とは有用性という意味があり、トークンの使用方法は出資した事業に関するものに限定されます。しかし、実際には投機的な目的で発行されるトークンも数多く存在しており、法整備が追いついていない状態です。

ICO と STO の比較

仮想通貨を用いた資金調達の方法として初めて誕生したのが「ICO」です。2018年現在、ICO に関する法律は未整備な部分も多く、詐欺まがいの ICO が乱立しているのが現状です。そのため、投資家保護の観点から今後 ICO に対する規制が厳しくなる事が予測されます。現在の法律では証券としての条件を満たさない「Utility Token」が、将来的に証券としての条件を満たす可能性があります。これにより、トークンの発行主体は証券の発行者として見なされることとなり、トークンの発行には関するあらゆる情報の開示や、厳しい審査が必要となり、簡単に ICO を行う事ができなくなります。これに対して、あらかじめ証券としての条件を満たしたトークンを売り出すのが STO です。証券としての条件を満たす事で気軽なトークンの発行はできませんが、**詐欺などの危険性の少ない安全なトークン**として保証されます。

2.4 Wallet(ウォレット)

ウォレットとは「**仮想通貨を管理する鍵の入れ物**」のことを言います。ここで出てきた「鍵」とは自身が所有する仮想通貨を使う際に必要な文字列のことです。1つの鍵からは「アドレス」と呼ばれる文字列が1つ生成されます。銀行口座に例えるならば**鍵が暗証番号**、**アドレスが口座番号**のようなものです。

銀行口座を利用する際にはパスワードが必要になることと同じように、仮想通貨を使用する際には鍵が必ず必要になります。ここで、仮想通貨の鍵の管理には注意しなければならない点があります。それは、**鍵を紛失するとそのアドレスに紐付いた通貨を使用する事ができなくなってしまう**点です。銀行の口座番号や暗証番号を忘れた時には、銀行を通じて個人情報などを照会し、手続きを行うと、暗証番号が再発行してもらえ、口座が再び使えるようになります。しかし、仮想通貨は管理者が存在しないシステムであり、**鍵の再発行を行う機関は存在しません**。そのため、鍵の管理には細心の注意を払う必要があります。

では、その鍵はどのように保管するのが良いのでしょうか。ここからは鍵の保管方法について説明します。

鍵の保管方法は様々あり、それによりウォレットの種類を分類することができます。そのウォレットの種類とそれぞれの特徴について説明します。

2.4.1 ホットウォレットとコールドウォレット

ホットウォレット

インターネットからアクセスすることのできる場所に鍵を保存するウォレット

素早く仮想通貨を利用することができる

仮想通貨を使用するための鍵をインターネットからアクセスできる場所に保管しておくことで、素早く仮想通貨を使用することができます。頻繁に仮想通貨を利用する際には便利なウォレットです。

鍵の流出のリスクが大きい

インターネットからアクセスすることのできる場所に鍵を保管しているため、**外部からの攻撃**により鍵が流出し、ウォレットに紐付いた仮想通貨が使用される可能性があります。銀行口座と異なり鍵が流出した際にも、アカウントの凍結などを行う管理者は存在しません。

ホットウォレットの種類

Web ウォレット

鍵を Web ウォレットサービスの提供している会社に管理してもらいます。自身で鍵を管理する訳ではないで鍵を紛失する可能性は低いですが、サイバー攻撃などにより、**鍵を管理している会社から鍵が流出する可能性があります**。

ソフトウェアウォレット

ソフトウェアウォレットでは自身が持つコンピュータやスマートフォンの中に鍵を保存します。Web ウォレットに比べてサイバー攻撃により鍵が流出する可能性は少ないですが、鍵を保存している**機器の破損や紛失などによる鍵の紛失の可能性があります**。

コールドウォレット

インターネットからアクセスできない場所に鍵を保存するウォレット

鍵の流出のリスクが小さい

鍵をオフラインで管理するため、インターネットを経由して外部から攻撃を受けることはなく、**鍵の流出の可能性がホットウォレットに比べて極めて低くなります**。

すぐに送金を行うことができない

鍵をオフラインで管理しているため、仮想通貨を使用する際には鍵の情報をオンラインで使用することができるようにする必要があります。そのため、即時に送金を行うことができないため、頻繁に仮想通貨を利用する場合には向いていない。

鍵の管理方法の難しさ

鍵を専用のハードウェアなどで管理するため、完全に自身で管理を行わなければなりません。そのため、**バックアップの作成や、物理的な盗難や紛失には細心の注意を払う必要があります**。

コールドウォレットの種類

ハードウェアウォレット

ハードウェアウォレットとは、**鍵を管理するための専用のハードウェアに鍵を保存**する方法です。通貨を使用する際にだけ、ハードウェアウォレットから鍵の情報を取得し、オンラインにするのでサイバー攻撃による鍵の流出の可能性は低くなります。しかし、当然ながらハードウェアウォレットを紛失や破損してしまうと鍵は失われてしまいます。また、鍵を管理する専用のハードウェアが高価な点や、全ての仮想通貨には対応していない点が導入の際の課題となります。

ペーパーウォレット

ペーパーウォレットとは言葉の通り、**鍵を紙に書いて保存**します。サイバー攻撃による流出を完全に防ぐ事ができますが、鍵を紙に書いて保存するので紙の紛失や劣化には注意が必要です。鍵の管理専用の紙も販売されています。

ブレインウォレット

鍵を覚えて、自身の**頭の中で管理**します。流出の可能性はありませんが、鍵を忘れてしまうリスクはあります。

2.4.2 決定性ウォレットと非決定性ウォレット

仮想通貨を保有するためには、「アドレス」が必要になります。アドレスとは仮想通貨を使用する際に必要となる「鍵」から一意に作成される文字列です。アドレスは銀行の口座番号に似ていますが、いくつか異なる点があります。1 つ目に銀行口座は特定の個人と紐づけて作成されますが、**仮想通貨のアドレスは一切の個人情**

報とは紐づいていないという点です。次に、一人で銀行口座を無数に開くことはできませんが、**仮想通貨のアドレスはいくつでも作ることができる**という点です。

仮想通貨の取引を行った際には、「どのアドレスからどのアドレスにいくらの送金が行われた」という情報がブロックチェーンに記録されます。Bitcoin などのパブリックブロックチェーンでは全ての取引を確認することができるため、アドレスと個人情報がなんらかの理由で紐づいてしまうと、誰から誰への送金が行われたのか特定されてしまい、プライバシーの問題が発生します。また、同じアドレスを繰り返し使用した際にも、個人が特定されやすくなり、取引の匿名性の低下に繋がります。そのため、仮想通貨の送金を行う際には**複数のアドレスを作成し、取引を行う毎に使用するアドレスを変更することが推奨**されています。そのため、アドレスを作成するための鍵も複数作成しなければなりません。鍵の作成方法は大きく分けて 2 種類存在し、それによりウォレットの種類を分けることができます。

非決定性ウォレット(ランダムウォレット)

非決定性ウォレットはランダムウォレットとも呼ばれるウォレット方式です。また、このタイプのウォレットは「Type-0 非決定性ウォレット」とも呼ばれ、Bitcoin のクライアントソフトウェアである Bitcoin Core で採用されています。Bitcoin Core では初期起動時にランダムに鍵を 100 個生成し、それぞれの鍵から 1 つずつアドレスを生成します。ランダムに鍵の生成が行われるため、鍵同士の関連はなく、**生成した全ての鍵の管理が必要**になり、鍵の管理にコストがかかります。Bitcoin では、取引の安全性の観点から個々の Bitcoin アドレスは、トランザクションは 1 件のみに使うことが推奨されており、つまり Bitcoin アドレスは取引を行う度に、使い捨てることが推奨されています。これに対して、非決定性ウォレットでは、生成した多数の鍵は全てを保存しなければならず、Bitcoin が推奨しているアドレスの利用方法とは相容れない部分があります。

決定性ウォレット

決定性ウォレットでは無作為に鍵の作成を行うのではなく、**「シード」と呼ばれる文字列によりランダムに生成された 1 つの数値を元に多数の鍵が生成**されます。決定性ウォレットでは 1 つのシードから全ての鍵が生成されるため、シードの管理

を行うことで、生成した全ての鍵を復元することができます。そのため、非決定性ウォレットのように全ての鍵の管理を行う必要はなく、鍵の管理のコストが低いことが特徴です。

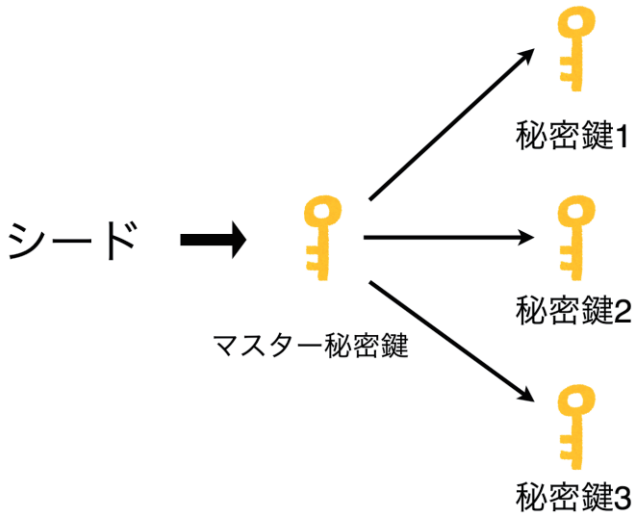


図 2.2 決定性ウォレットのイメージ

Mnemonic Code(ニーモニック コード)

決定性ウォレットではシードを「Mnemonic Code」と呼ばれる 12 から 24 個の英単語で表す方法があります。ランダムに作成された文字列であるシードのメモを取ることに比べ、馴染みのある英単語の方が転写のミスが少ないことからこのような方法が使われています。

以下に、Bitcoin でシードから Mnemonic Code が作成されるまでの流れを紹介します。

1. 128bit から 256bit のランダムな文字列を作成する
2. ランダムな配列の SHA256 ハッシュの先頭 4bit を取得し、ランダムな文字列のチェックサムを生成する
3. このチェックサムをランダムな文字列の最後に付加する
4. 2048 個のあらかじめ定められた単語の辞書のインデックスとして使うために、この文字列を 11bit ずつの部分に分解する
5. Mnemonic Code を表す 12 から 24 個の単語を生成する

Mnemonic Code で使用される 2048 個の単語は使用するウォレットの種類によって異なりますが、英語、日本語、スペイン語、中国語、フランス語、イタリア語などが用いられています。

階層的決定性ウォレット

決定性ウォレットは、1 つのシードから多くの鍵を生成しやすく、またそれらの管理が容易になるように開発されたウォレットです。決定性ウォレットで最も進んだ形は、階層性決定ウォレットです。階層性決定ウォレットでは**鍵がツリー構造を成します**。この構造においては親鍵が子鍵群を生成し、その子鍵が孫鍵を生成することにより、生成の連鎖が無限に続きます。ツリー構造であることで、ある「ブランチ」を通貨の受取に使ったり、別の「ブランチ」を支払いに使ったりと、ブランチ毎に用途を割り当てることができます。

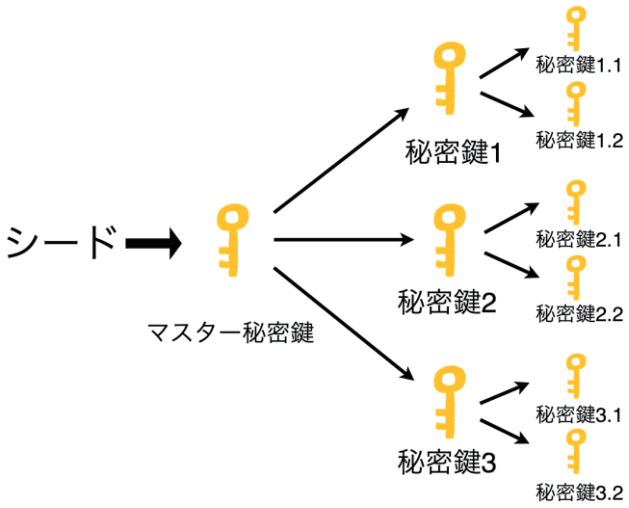


図 2.3 階層的決定性ウォレット

以上が、仮想通貨の概要です。法定通貨や電子マネーとは異なった価値の移転の仕組みである仮想通貨は、それらとは異なる特徴を持っており、この特徴を理解することでブロックチェーンに対する理解も深める事ができます。

3. ブロックチェーンの構成要素

この章ではブロックチェーンの処理の流れを追いながら、ブロックチェーンを構成する要素について説明します。

Let's TRY

３章演習

仮想通貨の送金を例にブロックチェーンの処理の流れと基本的な語句をカードゲームを通じて理解します。

ブロックチェーンゲーム

4～5 人のグループを作成してください。

必要なもの

- 電卓（スマートフォンなどでも可）
- ボールペン
- 演習に必要なカード（巻末に付いています）
    - ジェネシスブロック（1 人 1 枚）
    - ブロック（各グループに人数×4 枚）
    - トランザクション（各グループに人数×5 枚）
    - アドレス（1 人 1 枚，グループ内でアドレスが重複しないようにする）

準備

- ジェネシスブロックとアドレスのカードを 1 人 1 枚配布する。
- アドレスのカードを確認し、メンバーそれぞれのアドレスを把握する。

ジェネシスブロックを確認して、全員の現在の通貨の保有量を確認する。

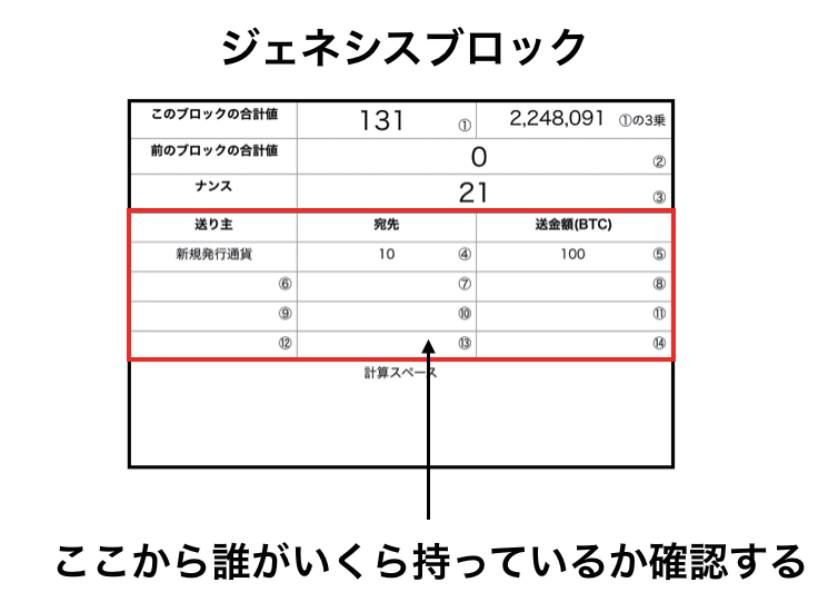


図 3.1 ジェネシスブロックの確認箇所

上図の例では、アドレス 10 の人が 100BTC 持っていることを確認できる。

1. 送金を行う

- 誰から誰に送金を行うか決める
- 送金元の人がトランザクションのカードに以下を記入する。
  - ○ 送り主のアドレス
  - ○ 宛先のアドレス
  - ○ 送金額
  - ○ 署名

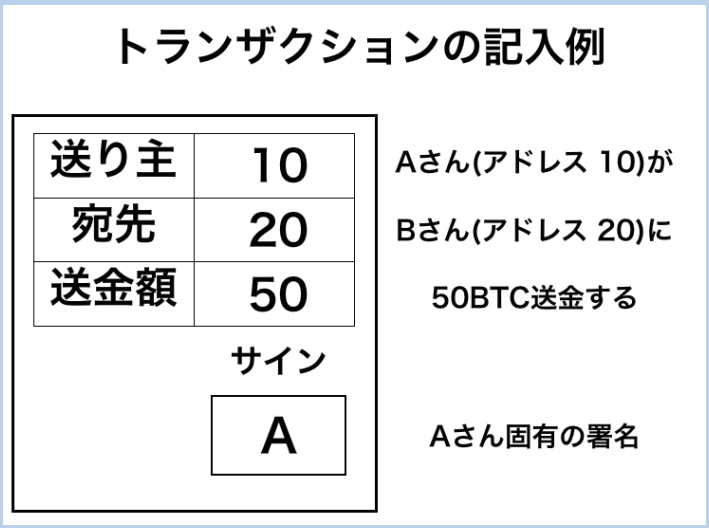


図 3.2 トランザクションの記入例

- トランザクションをグループの人数分作成し、全員に渡す。
- トランザクションを受け取った人は、そのトランザクションが正しいかどうか検証を行う。
  - 未記入の項目はないか
  - 通貨の所有者のサインが行われているか
  - 送金者が持っている額以上の送金を行なっていないか(手元にあるブロックチェーンと、既に検証が終わり手元にあるトランザクションから確認する)

- 正しいトランザクションであると判断すると、自分の手元に保存する。不正なトランザクションであると判断した時は破棄し、破棄したことをトランザクションの発行者に伝える。

2. ブロックの作成を行う

- トランザクションがいくつか(1～2 個)手元に溜まるとトランザクションをまとめてブロック作成に移る。
- 1 つ前のブロックの情報をブロックに書き写す。(項目②)
- トランザクションの情報をブロックに書き写す。(項目⑥～⑭)
- 新規発行通貨の宛先に自分のアドレスを記入する。(項目④)

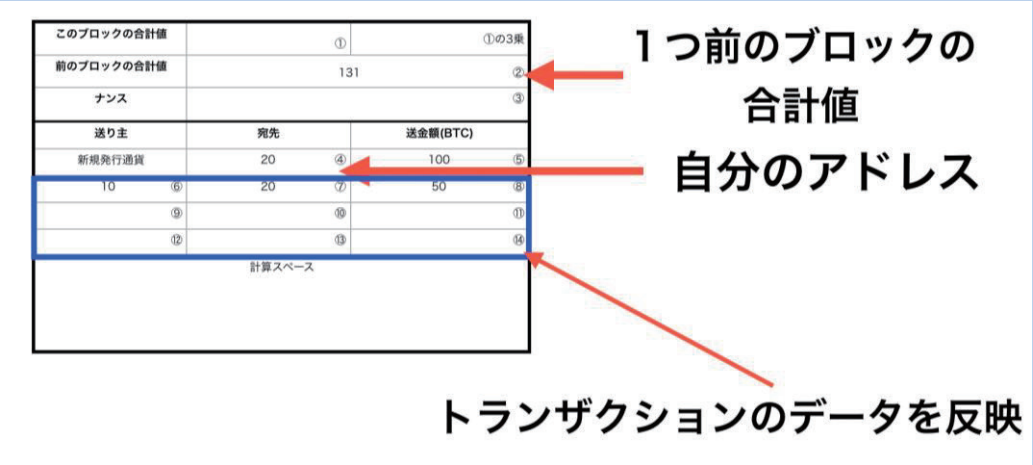


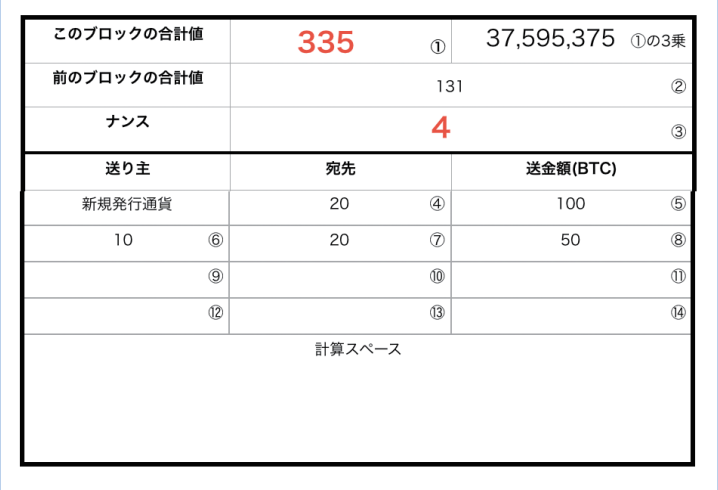
図 3.3 ブロックの記入例

- ①の項目に②～⑭までの和を記入する。そのためにそれぞれが適切なナンス(項目③)を求め、残りの項目を埋めてブロックを完成させる。
  - ○ ブロックの合計値は 3 乗すると、その値の下 3 桁が「001 以上、099 以下」でなければならない
  - ○ 過去のブロックの合計値と重複してはならない
  - ○ ナンスは 0 以上、1000 以下の整数でなければならない

ブロックの作成例

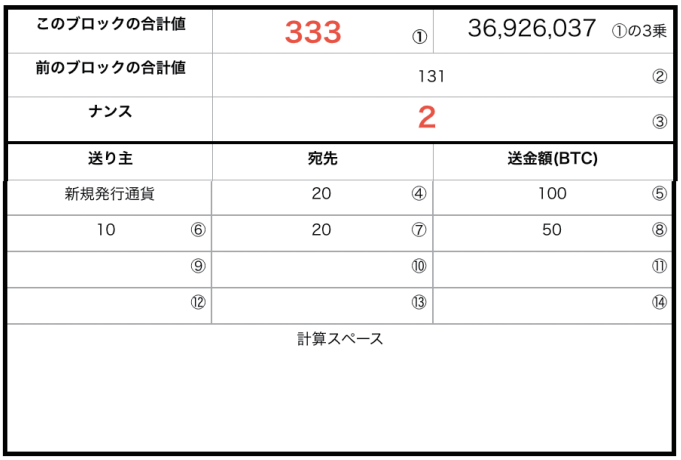
図 3.3 のブロックを例にナンスを求め、ブロックを完成させる流れについて説明します。

1. 項目②、④～⑧までの数の足し算を行う。トランザクションが複数ある場合は、⑪または⑭まで足し算を行う。今回は 331 になる。
2. 項目②のナンスに任意の数を入れて、さらに足し算を行う。例として図 3.4 のようにナンスに 4 を入れて、再度足し算を行い、項目①が 335 となる。
3. ブロックの合計値(項目①, 335)を三乗する。そうすると、値は 37, 595, 375 となる。この値は下 3 桁が 375 になっているため、ブロック作成の条件を満たしていない。そのため、再度別のナンスで計算を行う。
4. 次はナンスを 2 として計算を行う。合計値は 333 となり、3 乗すると 36, 926, 037 となる。この値の下 3 桁は 037 であり、ブロック作成の条件を満たしている。これによりブロックが完成する。
5. この計算を繰り返し適切なナンスを探し当てる。



このブロックの合計値	335	①	37,595,375	①の3乗
前のブロックの合計値	131	②		
ナンス	4	③		
送り主	宛先		送金額(BTC)	
新規発行通貨	20	④	100	⑤
10 ⑥	20	⑦	50	⑧
⑨	⑩		⑪	
⑫	⑬		⑭	
計算スペース				

図 3.4 不適なブロック
(ナンスに 4 を入れ、3 乗の値の下 3 桁が 375)



このブロックの合計値	333 ①	36,926,037 ①の3乗
前のブロックの合計値	131	②
ナンス	2	③

送り主	宛先	送金額(BTC)
新規発行通貨	20 ④	100 ⑤
10 ⑥	20 ⑦	50 ⑧
⑨	⑩	⑪
⑫	⑬	⑭

計算スペース

図 3.5 適切なブロック
（ナンスに 2 を入れ、3 乗の値の下 3 桁が 037）

- グループ内で最も早くブロックができた人は、それをメンバーに伝える。
- 最も早くブロックを作った人が同じブロックを人数分作成し、全員に渡す。
- ブロックの受け取った人は、そのブロックが正しいかどうかの検証を行う。
    - ○ トランザクションは正しく反映されているか
    - ○ 抜けている項目はないか
    - ○ ナンス、ブロックの合計値は正しいか

- 検証の結果正しいブロックであると判断すると、自分の手元に保存する。不正なブロックであると判断すると、ブロックを破棄し、再度それぞれがナンスの計算、ブロックの作成に戻る。
- 手元に保存したそれぞれのブロックは 1 つ前のブロックの合計値を持っているので、それにより手元でブロックがチェーンのように繋がる。これを狭義のブロックチェーンと呼び、これ全体が台帳となる。
- ブロックに含まれたトランザクションは手元から破棄する。

以上で、送金作業が終わる。ここまでの作業が終わると、再び送金のためのトランザクションの作成に戻る。

3.1 ブロックチェーンの処理の流れ

3.1.1 ブロックチェーン上での送金の処理

A さんが B さんに仮想通貨を送金することを例にして、ブロックチェーンの処理の流れを説明します。

図 3.6 仮想通貨送金の例

1. トランザクションの作成

A さんは「A さんから B さんに通貨の所有権を移転する」というデータを作成します。このデータを「**トランザクション**」と呼びます。

トランザクション	
送金元	A
送金先	B
送金額	1BTC
・ ・ ・	

図 3.7　トランザクションのイメージ

2. トランザクションの伝達と検証

A さんは作成したトランザクションをブロックチェーンの**ネットワークに伝達**します。

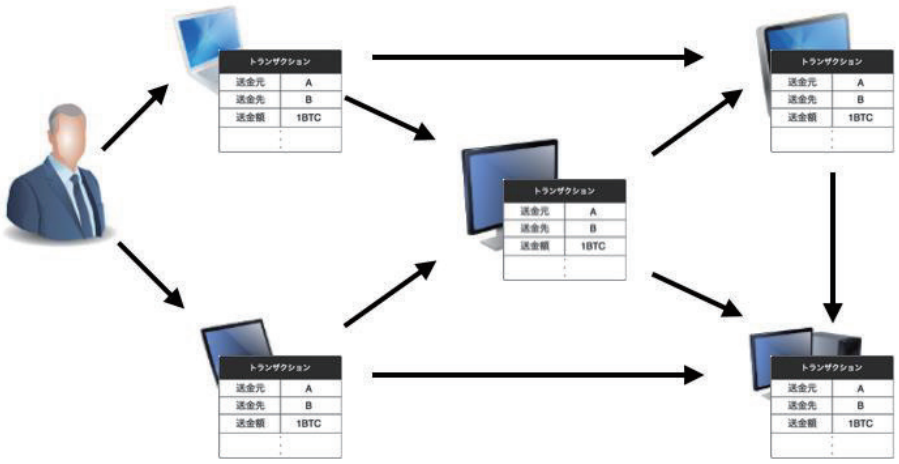


図 3.8　トランザクションの伝達

トランザクションを受け取ったノードは、トランザクションの内容が正当なものであるかの**検証**を行います。ブロックチェーンの種類により異なりますが、以下に主なトランザクションの検証内容を紹介します。

- 自分が持っている額以上の送金を行おうとしていないか
- 正しいフォーマットで記述されているか
- データサイズは適切か

仮想通貨は紙幣や硬貨のように物理的な通貨を取引するのではなく、トランザクションというデータに通貨の移転に関する情報が載っているだけです。トランザクション自体は単なるデータであるため、自身が持っている額以上の送金額を設定したトランザクションを作成する事も可能です。また、一度使用した通貨を再度利用するトランザクションなども作成することができます。このようなトランザクションが有効なものとして認められると、保有している通貨の量に関わらず好きな額の取引を行うことができてしまい、仮想通貨は通貨として機能を失ってしまいます。

これを防ぐために**トランザクションを受け取ったノードは正しい額の送金であるか確認**を行います。

　これ以外にも、トランザクション内の項目の確認やトランザクションのデータサイズなどの検証も行われます。Bitcoin では 20 項目程度の検証があります。

　それぞれのノードは検証の結果、正しいトランザクションであると判断すると、隣接するノードにトランザクションを伝達します。また、トランザクションが不正であると判断すると伝達も保存もせずにトランザクションを破棄し、トランザクションの発行元である A さんにトランザクションを破棄したことを通知します。

　トランザクションがノードからノードへの伝達が繰り返されることで、**トランザクションはネットワーク全体に伝達**されます。

3. ブロックの作成

A さんがトランザクションを作成し、そのトランザクションの検証がネットワーク内で行われている間にも、次々と他のトランザクションが発行され、伝達、検証作業が行われ続けます。Bitcoin では 2018 年 12 月の段階で平均して 1 秒間に 2～3 個のトランザクションが作成されます。

検証作業が終わりそれぞれの参加者の手元に保存されたトランザクションは、一定時間毎にネットワークの中から選ばれた 1 つのノードにより「**ブロック**」という単位にまとめられます。代表者の選出からブロックの作成までの動作を「**マイニング**」と言い、マイニングを行うノードのことを「**マイナー**」と言います。マイニングに関しては 6 章で詳しく説明を行います。

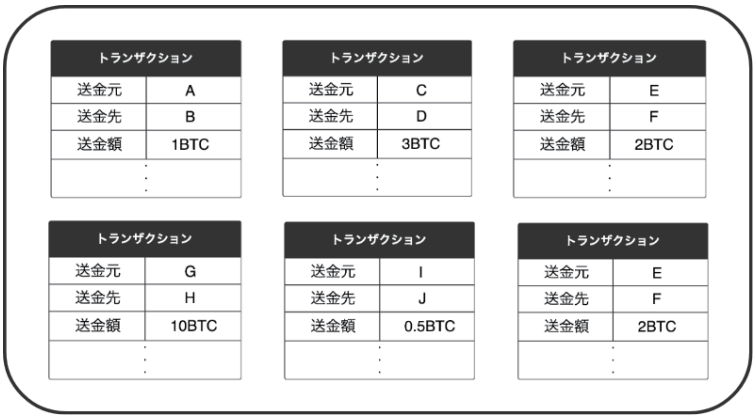


図 3.9　ブロックのイメージ

4. ブロックの伝達と検証

マイナーはブロックを作成すると、トランザクションを発行した時と同様に隣接するノードへ**ブロックの伝達**を行います。

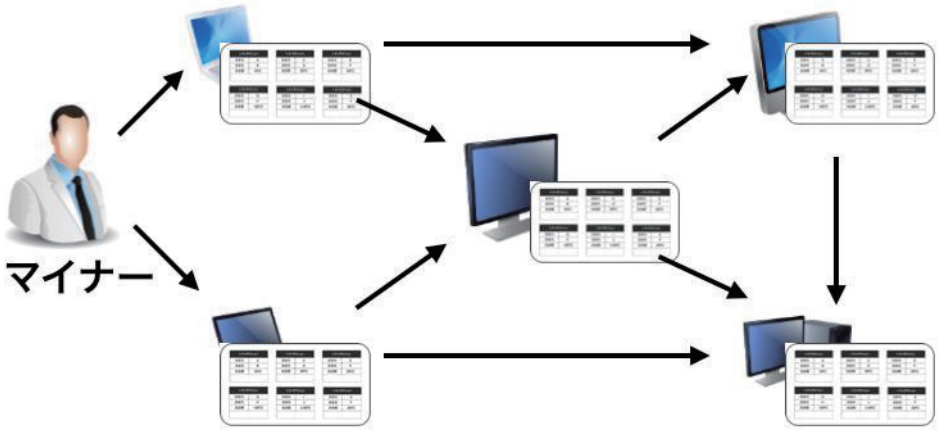


図 3.10 ブロックの伝達

ノードは新たに作成されたブロックを受け取ると、そのブロックが正当なものであるか検証を行います。検証には多くの項目があり、その中のいくつかを以下に示します。

- ブロックの構造は正しいか
- ブロックに含まれているトランザクションは全て正しいものか
- ブロックのサイズは適切か

ブロックの検証の結果、正しいブロックであると判断すると、各ノードは**ブロックを手元に保管し、かつ隣接するノードに伝達**します。検証と伝達が繰り返されることで、ブロックはネットワーク内の全てのノードに伝達され、全員が同じブロックを所有している状態になります。**全員が同じブロックを所有している状態は、全員が同じトランザクション、つまり同じ取引履歴を所有している状態**と言うことができます。

ブロックの作成は発行されるトランザクションの多い少ないに関わらず、あらかじめブロックチェーンのプロトコルにより定められた時間毎に行われます。そのため、時間の経過とともに次々とブロックが作成され、全てのノードに届けられます。

5. ブロックチェーンの形成

ブロックを受け取ったそれぞれのノードは、**受け取ったブロックを既に手元にあるブロックに繋ぎます**。これにより狭義のブロックチェーンが形成されます。

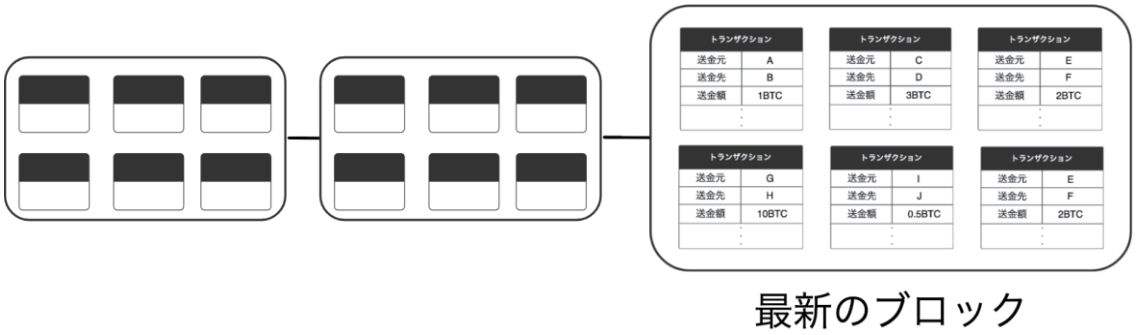


図 3.11 ブロックチェーンのイメージ

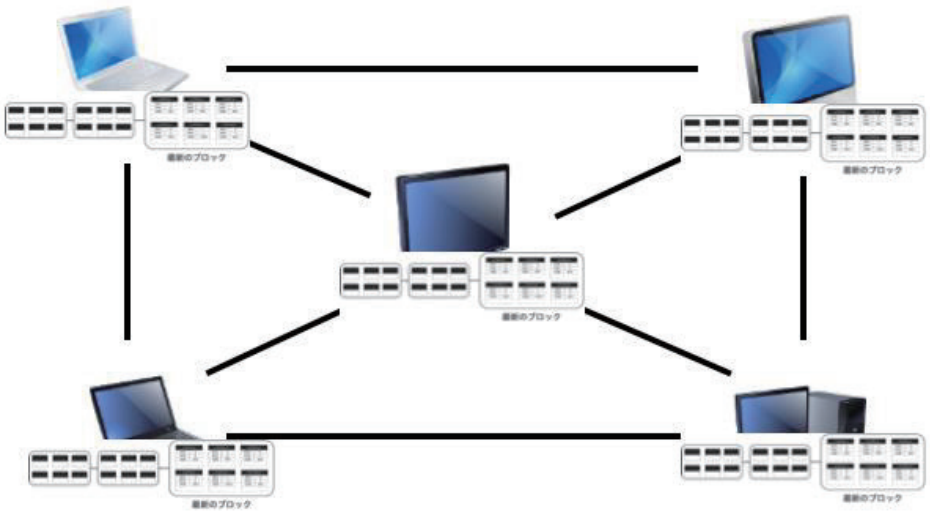


図 3.12 ブロックチェーンの共有

以上の手順を経て A さんが作成したトランザクションはブロックチェーンに記録されます。全てのトランザクションは一度ブロックチェーンに記録されると内容が書き換えられることも、削除されることもありません。

3.2 トランザクション

ブロックチェーンの処理の流れで説明した通り、ブロックチェーン上での処理はトランザクションが作成されることから始まります。トランザクションの構造や、役割はブロックチェーンにより少しずつ異なりますが、ここではBitcoinにおけるトランザクションの役割を説明します。

3.2.1 Bitcoinにおけるトランザクション

Bitcoinにおけるトランザクションは「**参加者間の価値の移転を記述したデータ**」としての役割があります。よりわかりやすく説明すると、トランザクションは小切手のようなもので、特定の相手への送金を行うという意思を表示します。トランザクションは作成されると、通貨の所有者により署名が行われることで、有効なものとなります。その後、ネットワーク内に伝達されることで、このトランザクションはブロックチェーンに記録され、Bitcoinネットワーク内での価値の移転は完了します。

トランザクションアウトプットとトランザクションインプット

Bitcoinのトランザクションの仕組みを理解する上で重要なのが、**UTXO(Unspent Transaction Output)**です。日本語では「未使用トランザクションアウトプット」と訳されます。UTXOとは**特定のアドレスに保有されたBitcoinの集まり**のことです。通貨の保有者はUTXOに署名を行うことにより、そのUTXOに記録されている額の通貨を利用することができます。ユーザーのBitcoinは、UTXOとしてブロックチェーン上に散らばっており、ユーザーの手元でまとまって管理されているわけではありません。

UTXOの特徴として、1つのUTXOに記録されている額の一部のみを利用することはできません。そのため、支払いを行う際には支払い額以上になるように複数のUTXOを組み合わせて利用します。トランザクションによって使用されたUTXOはトランザクションインプットと呼び、トランザクションにより作られたUTXOをトランザクションアウトプットと呼びます。

ある支払いで作成されたトランザクションアウトプットが、次の保有者が支払う際にトランザクションインプットとして連鎖的に使われることで、通貨の所有権が移っていきます。

3.3 ブロックとブロックチェーン

3.3.1 ブロック

ブロックとは、複数のトランザクションがまとめられたデータです。それぞれのブロックには複数のトランザクションと、ブロック固有のデータが記録されています。それぞれのブロックはブロック固有のデータの中に**1つ前のブロックの識別子**を持っています。これにより、それぞれのブロックは固有の親ブロックを特定することができ、これが繰り返されることで鎖状に繋がっていきます。この独特のデータ構造により、ブロックチェーンの高い改ざん耐性が生まれています。データ構造と改ざん耐性の関連については6章で詳しく説明します。

3.3.2 ブロックチェーン

トランザクションがまとめられたブロックが鎖状に繋がったようなデータ構造から、このシステムにはブロックチェーンという名前がつけられました。ブロックチェーンという言葉は、システム全体のことを示すこともあれば、先程説明した通り、ブロックが鎖状に繋がっている構造を示すこともあります。このようにブロックチェーンという言葉が示す対象はいくつかあるので注意する必要があります。

4. ブロックチェーンを支える暗号技術

ブロックチェーンは 1 つの大きな技術革新により実現されたのではなく、既存の様々な技術を組み合わせることによって実現された技術です。この章では、ブロックチェーンを構成する技術の中でも、特に重要な暗号技術に関連する説明を行います。

4.1 ハッシュ関数

ハッシュ関数とはメッセージダイジェスト関数とも呼ばれ、ブロックチェーンだけではなく、IT 分野において様々な場面で用いられています。

ハッシュ関数の説明に入る前に、まず「関数」についての説明からはじめます。関数とは「ある入力に対して特定の出力がされる仕組み」です。

以下の図 4.1 では「ある整数を入力するとその数の 2 倍の値を出力する関数」を f(x)、「ある文字列を入力するとその文字列の順序が反転されたものを出力する関数」を g(x)を具体例として紹介します。

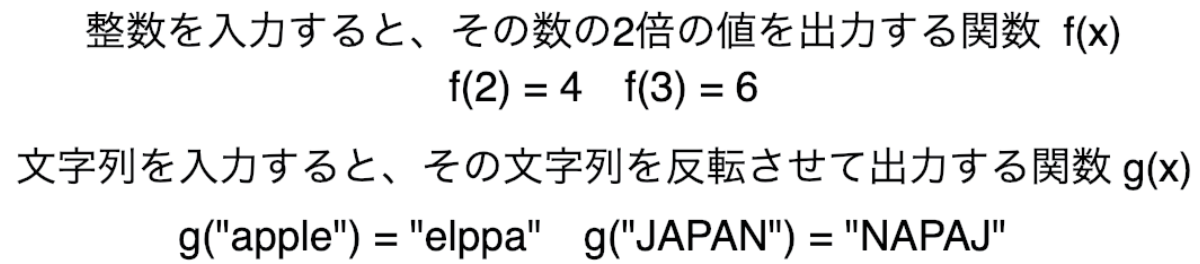


図 4.1 関数の例

上の図 4.1 の f と g が関数で、括弧の中が入力値です。関数に入力値を与えた結果、右辺に出力されている数値や文字列が関数の出力値です。一般的に関数として馴染みのある一次関数や三角関数などの関数では、入力値も出力値も数値のみを使用します。しかし、大きな意味で関数を捉えると入力値、出力値ともに数値だけでなく、文字列なども利用することができます。f(x)では、入力値は整数であり、g(x)では入力値は文字列です。

4.1.1 ハッシュ関数の特徴

ここからはハッシュ関数の説明に入ります。ハッシュ関数は**データを入力すると、そのデータ固有の文字列が出力される関数**です。この出力値はハッシュ値、ダイジェスト値と呼ばれます。

以下の図 4.2 にハッシュ関数の利用例を 3 つ示します。

図 4.2 ハッシュ関数の例

図 4.2 の左辺にある H がハッシュ関数を示します。括弧の中に入っているのが入力する文字列です。ハッシュ関数に文字列を入力した結果、出力されるものが右辺のハッシュ値です。図 4.2 で入力される 3 つの文字列はどれもアルファベットの並びは同じであり、大文字であるか小文字であるかが異なるのみです。しかし、それぞれ出力されるハッシュ値は全く異なっており、元の文字列が似ているかどうか判別することはできません。

Let's TRY

４章演習１

実際にハッシュ関数を利用しながら、その特徴を理解していきます。

準備

- 巻頭で作成した仮想マシンを立ち上げる。
- 仮想マシン上でターミナルを開く。
- ターミナルに `$ irb` と入力すると、Ruby のコンソールが開きプログラムが書けるようになる。
- `$ require 'digest'` と入力する。
- true と返ってくると、ハッシュ関数が利用できるようになる。

1. ハッシュ関数 SHA-256 を用いてハッシュ値を求める

- `$ Digest::SHA256.hexdigest('hello')` と入力する。
- 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824 が返ってくることを確認する。
- 様々な入力値を試し、異なるハッシュ値が出力されることを確認する。

2. ハッシュ関数 RIPEMD-160 を用いてハッシュ値を求める

- `$ Digest::RMD160.hexdigest('hello')` と入力する
- 108f07b8382412612c048d07d13f814118445acd が返ってくることを確認する。
- 様々な入力値を試し、異なるハッシュ値が出力されることを確認する。
- SHA-256 と RIPEMD-160 で出力されるハッシュ値の違いについて考える。

3. SHA-256 を用いて少し長い文章のハッシュ値を求める

- `$ Digest::SHA256.hexdigest('I have a pen. I have an apple.')`と入力する。
- 57784242b5094e39b96cdcc4f438c4c89958dca866fe4015e9b44fd9516fb96c が返ってくることを確認する。
- `$ Digest::SHA256.hexdigest('I have a pen. I have an apple..')`と入力する。
- d9c96c2bb0d9fb0f4eccfe24052f369039c9bd800fb7a20355e109f29248064d が返ってくることを確認する。
- 少しずつ入力値を変化させると、ハッシュ値がどのように変化するか確認する。

ここまでを踏まえて、以下にハッシュ関数の特徴を示します。

- 同じ入力値からは同じハッシュ値を得る
- 異なる入力値からは異なるハッシュ値を得る
- ハッシュ値から元の文字列の特定はできない
- 出力されるハッシュ値を予測することはできない

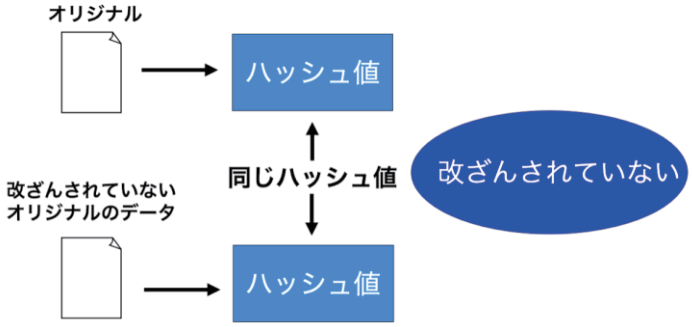
4.1.2 ハッシュ関数の利用例

ハッシュ関数はブロックチェーン以外でも様々な場面で利用されています。その一例を紹介します。

改ざんの検知

同じ入力値からは同じハッシュ値を得ることができ、入力値が異なると得られるハッシュ値も異なるというハッシュ関数の特徴を利用してデータの改ざんの検知に利用することができます。**データに改ざんが行われると、改ざんの前後でデータのハッシュ値が異なる**ため、これを比べることにより改ざんを検知することができます。データの中身を全て確認することに比べ、一定の長さで出力されるハッシュ値を比較するだけで良いので**効率的に改ざんを検知**することができます。

しかし、ハッシュ値を比較する改ざんの検知方法では、改ざんの有無はわかるものの、データ内で改ざんが行われた場所や、改ざんの内容は特定することができません。



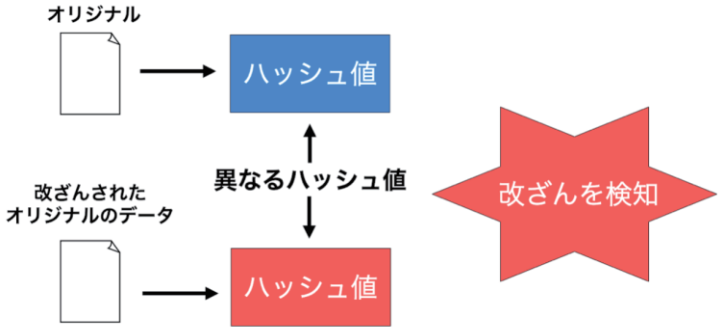


図 4.3　ハッシュ関数を用いた改ざんの検知

パスワードの保護

私たちが普段利用するオンラインサービスにおいて、パスワードによりユーザーの認証が行われることは頻繁にあります。認証を行うためのパスワードは多くの場合、**サービスを提供する企業のデータベースにハッシュ化して保存**されています。

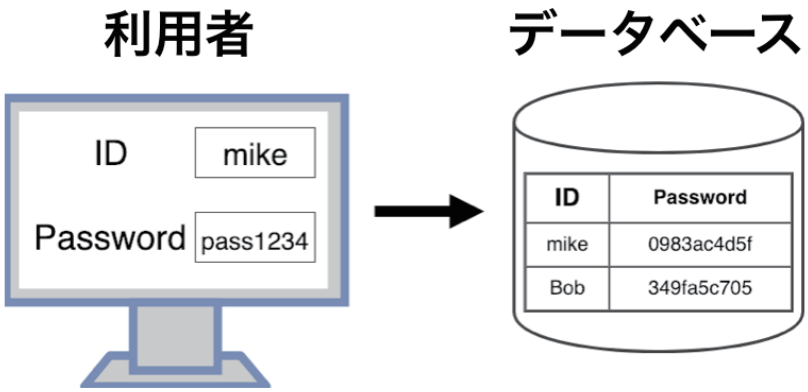


図 4.4 ハッシュ関数を用いたパスワードの保護

Let's TRY

４章演習２

ユーザーの認証の際に利用するパスワードはデータベースに保存する際にハッシュ化することが推奨されています。この理由について、4～5 人のグループで、以下の点について考えてください。

- パスワードがそのままの状態で保存されているとき、そのパスワードは誰にも悪用される可能性はないか
- パスワードが流出したとき、どのようなことが起こるか。それは、ハッシュ化して保存することで防ぐことができるか

それぞれの項目についてグループで話し合った結果を代表者が、発表してください。

4.1.3 ハッシュ関数の衝突耐性

ハッシュ関数の安全性の指標に「**衝突耐性**」があります。ハッシュ関数は入力値が異なると、異なるハッシュ値が出力されるという特徴を持っていますが、**極めて低い確率で異なる入力値から同じハッシュ値が出力される**ことがあります。これを「ハッシュ値の衝突」と呼びます。ハッシュ値の衝突に対してハッシュ関数が持っている耐性を「衝突耐性」と呼びます。衝突耐性は 2 種類に分けることができます。1 つは同一のハッシュ値を持つ 2 つの異なるデータが発見されることに対する耐性であり、これを「**強衝突耐性**」と呼びます。2 つ目はあるデータのハッシュ値を元に、同じハッシュ値を持つ異なるデータが発見されることに対する耐性であり、これを「**弱衝突耐性**」と言います。

4.1.4 ハッシュ関数の種類

ハッシュ関数には様々な種類が存在し、入力からハッシュ値を求める関数の内部の構造や出力されるハッシュ値の長さなどが異なります。ここでは 4 つのハッシュ関数を紹介します。

MD5(Message Digest Algorithm 5)

1991 年に開発されたハッシュアルゴリズムで 128bit の固定長のハッシュ値が出力されます。2018 年現在、一般的な計算速度のパソコンでも 10 数分程度で、同じハッシュ値を持つ異なる 2 つのデータを生成することができる実装が広まっています。これは強衝突耐性が容易に突破される状態にあるということであり、安全性を必要とする場面では、このハッシュ関数は使用することはできません。

RIPEMD-160(RACE Integrity Primitives Evaluation Message Digest 160)

1996 年に開発されたハッシュアルゴリズムで 160bit の固定長のハッシュ値が出力されます。他にも「RIPEMD-128」、「RIPEMD-256」、「RIPEMD-320」などの種類があり、それぞれ出力されるハッシュ値の長さが異なります。2018 年現在、このアルゴリズムは安全に使用することができます。Bitcoin に利用されているハッシュ関数の 1 つです。

SHA-1(Secure Hash Algorithm 1)

160bit の固定長のハッシュ値が出力されるハッシュアルゴリズムです。2017 年に Google により強衝突耐性が突破され、現在は安全に使用することはできません。

SHA-2(Secure Hash Algorithm 2)

SHA-2 は具体的なハッシュ関数の名前ではなく、ハッシュ関数の標準規格の 1 つです。SHA-224、 SHA-256、 SHA-384、SHA-512、SHA-512/224、 SHA-512/256 の 6 つの種類を持ちます。これらの主な違いは出力されるハッシュ値の長さです。一般的なハッシュ関数の特徴として、出力されるハッシュ値が長い方が、衝突が発生する確率が低く、セキュリティ的な強度は高くなりますが、ハッシュ値が長くなるに従って、コンピュータの負荷が大きくなります。これらを考慮して、SHA-2 を使用する際には 256bit のハッシュ値の出力が行われる SHA-256 が利用されることが多くなっています。SHA-2 に対する有効な攻撃法はまだ発見されておらず、2018 年現在、安全に使用することができます。Bitcoin に SHA-256 が利用されています。

4.1.5 ブロックチェーンでのハッシュ関数の利用

ハッシュ関数はブロックチェーンの様々な場所で利用されています。その中の 1 つとして、**ブロックの識別子**としての利用があります。

各ブロックの内部にはそのブロック固有の情報が記録されている**ブロックヘッダ**という部分があります。ブロックヘッダに記録されている内容は大きく分けて 3 種類あります。

1 つ目はタイムスタンプやナンスなどのブロックを作成するための**マイニングに関する情報**です。

2 つ目はそのブロックに含まれるトランザクションが要約されたマークルツリーの**マークルハッシュ**です。これにより、ブロック内のトランザクションに対する不正を防ぐことができます。

3 つ目は直前に作成された**ブロックのハッシュ値**です。このハッシュ値はブロック識別子やブロックハッシュと呼ばれ、ブロック固有のものでありブロックを識別

する際に使用されます。Bitcoin でのブロックの識別子は、直前のブロックのブロックヘッダを SHA-256 を用いて 2 回ハッシュ化したものです。

図 4.5 ブロックの構成要素について

4.2 暗号化アルゴリズム

4.2.1 暗号技術

暗号技術とは第三者に知られたくない機密情報、個人情報などを通信する際に、盗聴や傍受、改ざんを防ぐための技術です。あらゆる情報がインターネット上でやりとりされる現代社会では、暗号化技術は必要不可欠です。

暗号技術の基本的な語句

- 平文　　:暗号化されておらず誰でも読むことのできるデータ
- 暗号　　:平文を第三者には読み取れないようにしたもの
- 暗号化　:平文を暗号にすること
- 復号　　:暗号を平文に戻すこと
- 鍵　　:暗号化と復号の際に必要な情報

暗号技術を用いたデータの送信の流れ

暗号技術を用いた、データの送信の流れを図 4.6 で説明します。

1. 送信者は鍵を用いて平文を暗号化し、暗号を作る。

2. 送信者は暗号を受信者に送信する。

3. 受信者は暗号を受け取り、鍵を用いて復号し、平文を得る。

図 4.6 暗号技術を用いたデータ送信の流れ

シーザー暗号

暗号技術の例として、暗号理論上最もシンプルで古代ローマの時代から使用されている暗号方式であるシーザー暗号を紹介します。シーザー暗号では、平文の各文字を**辞書順にずらして暗号を作ります**。この際にずらした文字数が鍵となります。以下の図 4.6 で、アルファベットを用いた簡単な例で説明します。

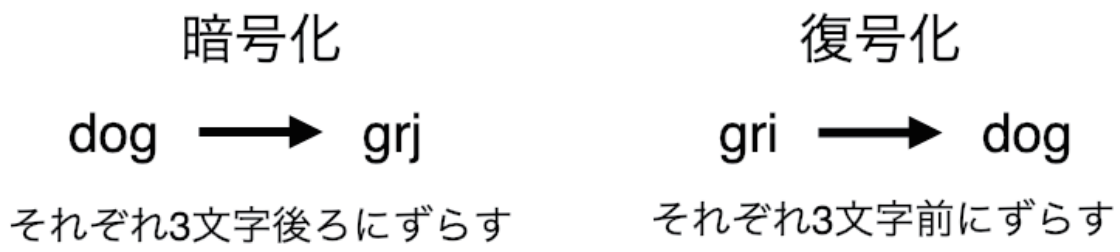


図 4.7 シーザー暗号の例

送信者は平文の各文字を辞書順にずらして暗号を作り、暗号を受信者に送信します。この際に鍵となるずらした文字数はあらかじめ伝えておく必要があります。これらの情報を元に受信者は、暗号の各文字を指定された文字数だけ暗号化とは逆方向にずらすことにより復号を行い、平文を得ることができます。

暗号化技術にはシーザー暗号のようにシンプルなものから、解読に高度な数学やコンピュータによる大量の計算が必要になるものまで様々な種類が存在します。多くの種類がある暗号化技術ですが、大きく「**共通鍵暗号方式**」と「**公開鍵暗号方式**」に分けることができます。ここではこれら 2 つの暗号方式の仕組みと、その特徴について説明します。

4.2.2 共通鍵暗号方式

暗号化と復号を同じ鍵を用いて行う暗号方式です。

共通鍵暗号の特徴

処理速度が速い

公開鍵暗号方式に比べて暗号化、復号の**処理が単純**であるため処理速度が速いという特徴があります。そのため送信するデータの容量が大きい場合にも、素早く処理を行うことができます。

鍵の配布が難しい

共通鍵暗号方式では暗号化と復号に同じ鍵を用いるため、暗号化を行った**鍵を通信相手に送らなければなりません**。鍵の伝達の際に鍵が盗聴されてしまうと、暗号

文と鍵の両方を第三者に知られてしまうことになり、暗号が簡単に解読されてしまいます。そのため、通信相手に鍵を渡す際には盗聴されないような工夫が必要になります。

管理する鍵の数が多い

共通鍵暗号方式では同じ鍵を用いて複数の人と通信を行うと、通信を行った人が全ての暗号を復号することができてしまいます。そのため、共通鍵暗号方式では安全性の問題から通信相手毎に異なる鍵を使用しなければなりません。そこで問題となるのが、管理しなければならない鍵の数です。共通鍵暗号では、**通信相手の数と同じ数の鍵を管理**することになり、不特定多数と通信を行う必要がある場合には、管理する鍵の数が膨大になるという問題があります。

共通鍵暗号の種類

DES(Data Encryption Standard)

鍵の長さは 64bit で、2 の 56 乗の鍵のパターンが存在します。2 の 56 乗は 10 進数に変換すると 17 桁の数になり、これでは鍵のパターンが少なく、コンピュータで総当たり的に鍵を試すことにより、解読が現実的な時間で行うことができます。このような問題があるため 2018 年現在、**安全に使用することはできません**。

AES(Advanced Encryption Standard)

コンピュータの計算速度の上昇に伴って、DES の相対的な強度が低下したことにより登場した暗号方式です。鍵の長さは 128bit、192bit、256bit の 3 つから用途により選ぶことができます。2018 年現在、どのような攻撃に対しても現実的な時間内で解読する方法は見つかっておらず、**安全に使用することができます**。

4.2.3 公開鍵暗号方式

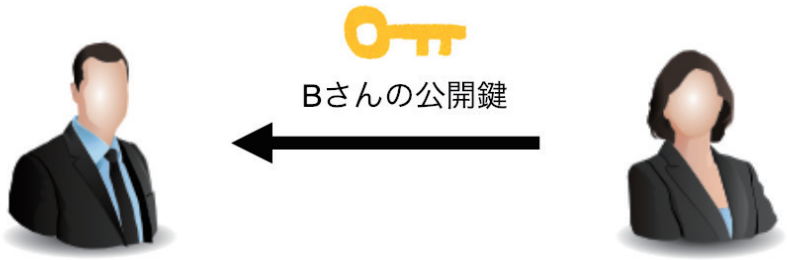
公開鍵暗号方式とは暗号化に「公開鍵」、復号に「秘密鍵」の **2 種類の異なる鍵を用いる暗号方式**です。この「公開鍵」と「秘密鍵」は 2 つで 1 対となっており、秘密鍵から公開鍵が作成されます。公開鍵暗号方式では、公開鍵を用いて暗号化を行い、暗号化されたデータは対応する秘密鍵を用いなければ復号することはできま

せん。つまり、公開鍵では暗号化することはできても、復号することはできないということです。このような仕組みであるため、通信を行う当事者以外が公開鍵を知っていたとしても、対応する秘密鍵さえ知られなければ、暗号が解読され平文を読まれる心配はありません。

使用する鍵の種類

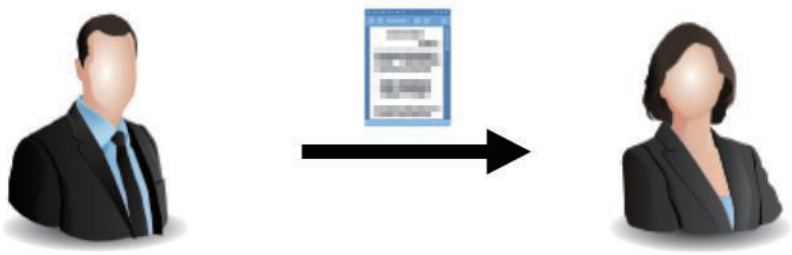
Bさんの公開鍵　　Bさんの秘密鍵

1. B さんはあらかじめ B さんの公開鍵を A さんに渡しておく



2. A さんは B さんの公開鍵を使って平文を暗号化する

3. A さんから B さんに暗号文を送信する



4. B さんは B さんの秘密鍵で復号する

図 4.8 公開鍵暗号の流れ

一方向性関数

公開鍵暗号の仕組みを理解する上で重要なのが、「一方向性関数」です。一方向性関数とは、**計算することは簡単だが、計算結果から元の情報を逆算することは極めて困難な関数**です。公開鍵暗号では、公開鍵を用いた平文の暗号化は簡単に行うことができますが、公開鍵と暗号のみを用いて平文を求めるための計算には膨大な時間が必要になります。しかし、ただ復号が難しいだけでは復号を行う際に時間がかかり、実用的ではありません。そのため、公開鍵暗号では「落とし戸付き一方向性関数」と呼ばれる特殊な一方向性関数が用いられています。この関数は計算結果から元の情報を逆算する際に、**特定の情報を用いると簡単に逆算が行うことができ**

る仕組みになっています。公開鍵暗号では秘密鍵が特定の情報にあたり、秘密鍵を用いると簡単に復号することができます。

公開鍵暗号方式では、暗号の解読は確率的には不可能ではありません。膨大な計算資源を費やし、一定時間に解読を試行する回数を増やすことで、解読にかかる時間は短くなります。

暗号化技術は情報を保護する情報の持つ価値と解読に必要な時間とコストのバランスにより守っています。現代使われているどの暗号方式も解読のためにはコンピュータによる膨大な計算が必要になります。多くのコンピュータを用意することや、より性能の良いコンピュータを準備することで、一定の時間内に多くの計算を行うことで解読までにかかる時間を短くすることはできます。しかし、これには多くのコストを費やす必要があります。このコストがデータの価値を上回る場合、暗号を解読するメリットはなくなります。また、現在安全であるとされて利用されている暗号技術を解読するためには、天文学的な時間がかかるとされています。

公開鍵暗号の特徴

不特定多数との通信に向いている

公開鍵を多くの人に配布したとしても、それに対する秘密鍵を所有していなければ復号を行うことはできません。そのため不特定多数の人との通信の際にも 2 つの鍵を管理するだけでよく、共通鍵暗号のように**多くの鍵を管理する必要はありません**。

処理速度が遅い

共通鍵暗号に比べて暗号化、復号の**処理が複雑**であるため、送信するデータ容量が大きくなると、処理に時間がかかってしまいます。

公開鍵暗号の種類

RSA(Rivest-Shamir-Adleman)

2 つの大きな素数の積を素因数分解し、元の 2 つの素数を求めることが難しいことを利用した暗号化技術です。2 つの素数の積を計算することは簡単ですが、その積を元の 2 つの素数に素因数分解するための効率的なアルゴリズムは 2018 年現在発

見されていません。そのため、計算に膨大な時間がかかり、現実的な時間内での解読は行うことができません。この計算の一方向性を利用して、暗号の安全性を担保しています。

ElGamal 暗号(エルガマル暗号)

離散対数問題を応用した暗号化技術です。離散対数問題とは整数を 300 桁程度の非常に大きな素数で割った時の余りに関する整数のまとまりに関して対数の演算を行うことで離散対数と呼ばれる数を求める計算を言います。この演算は全ての実数の範囲内において行うことは容易ですが、特殊な条件により限定された整数の集合では、大小性や連続性はなく近似という手法を用いることができないため、この計算を行うことが極めて難しくなります。

楕円曲線暗号

楕円曲線と呼ばれる曲線を用いた暗号化技術です。楕円曲線上に存在する点に関する演算の一方向性により暗号の安全性が守られています。RSA、ELGamal よりも格段に安全性が高く、計算速度も速い暗号方式です。RSA、ELGamal で 1024bit の鍵長と同程度の安全性を楕円曲線暗号では 160bit 程度の鍵長で保証することができます。

ハイブリッド暗号方式

共通鍵暗号方式と、公開鍵暗号方式を組み合わせた暗号方式です。データの受け渡しには共通鍵暗号方式を用いますが、その際に必要となる共通鍵の伝達を公開鍵暗号方式を用いて行います。これにより、共通鍵暗号方式の問題点であった鍵の受け渡しの際の盗聴、鍵の管理の問題、公開鍵暗号方式の問題点であった処理に時間がかかるという問題の両方を解決することができます。

Let's TRY

４章演習３

ここでは公開鍵暗号の１つである RSA 暗号を用いて実際に秘密鍵、公開鍵の作成とそれらを用いて暗号化、復号を行います。

準備

- 巻頭で作成した仮想マシンを立ち上げ、ターミナルを開く。

秘密鍵を作成する

- 秘密鍵の作成を行う。
  - `$ openssl genrsa > private_key`

- 秘密鍵を確認する
  - `$ cat private_key`
  - `$ openssl rsa -text < private_key`

公開鍵を作成する

- 公開鍵を作成する
  - `$ openssl rsa -pubout < private_key > public_key`

- 公開鍵を確認する
  - `$ cat public_key`
  - `$ openssl rsa -text -pubin < public_key`

暗号化する

- 暗号化する
  - message という名前でファイルを作成し、その中に文字列を書き込む
  - `$ cat message`
  - `$ openssl rsautl -encrypt -in message -pubin -inkey public_key > code`

- 暗号を確認する
  - `$ cat code`

復号する

- 復号する
  - `$ openssl rsautl -decrypt -in code -inkey private_key`

4.2.4 ブロックチェーンでの暗号化技術の利用

ブロックチェーンでの公開鍵暗号の利用例について Bitcoin を用いて説明します。**ブロックチェーンで用いられている暗号化技術は公開鍵暗号**です。Bitcoin では通貨を保有するためにはアドレスを作成する必要があります。アドレスは自身が作成した公開鍵をハッシュ化することにより作成されます。

具体的には、公開鍵に対して SHA-256 を用いてハッシュ値を求めた後に、さらにそのハッシュ値に対して RIPEMD-160 を用いてハッシュ値を求めます。このように、1 つの文字列に対して、ハッシュ関数を 2 回用いてハッシュ化を行うことを二重ハッシュ化と呼びます。最後に RIPEMD-160 を用いて求められたハッシュ値を、Bitcoin アドレス独自の形式に変換することでアドレスが作成されます。

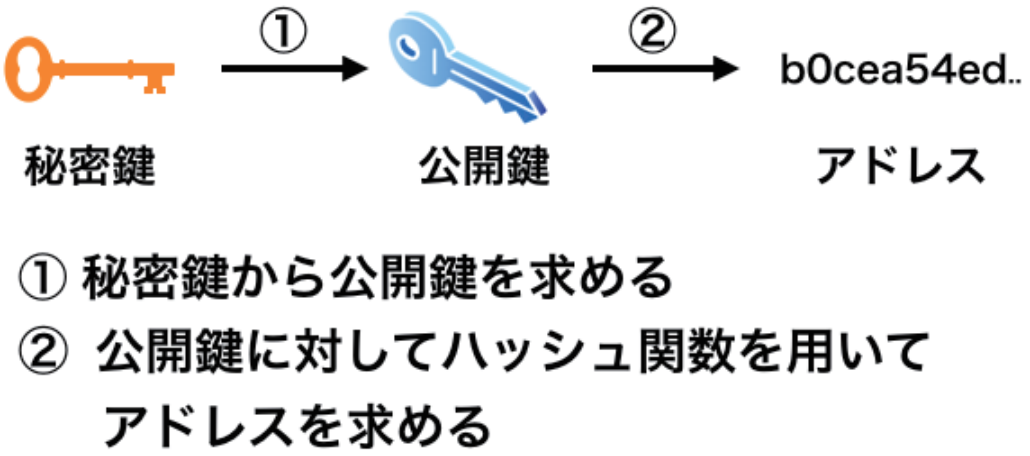


図 4.9 アドレスの生成までの流れ

4.3 電子署名

電子署名とは紙文書における印鑑やサインに相当する役割を電子文書において果たし、本人確認や、偽造、改ざんの防止のために用いられる技術です。

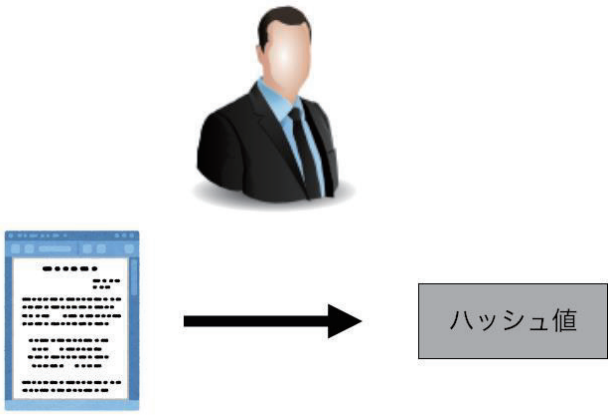
4.3.1 電子署名の仕組み

紙文書の場合、その文書が本当にその人物によって作成されたものであることは、その文書に付された作成者の印や署名によって証明することができます。しかし、電子文書において印鑑や署名データは簡単にコピーすることができるため、これらを信用することはできません。この問題を解決するために電子署名が用いられます。電子署名によって、**送信者の本人確認と、通信経路での改ざんの検知**を行うことができます。電子署名では公開鍵暗号方式で使用される秘密鍵と公開鍵を使用します。

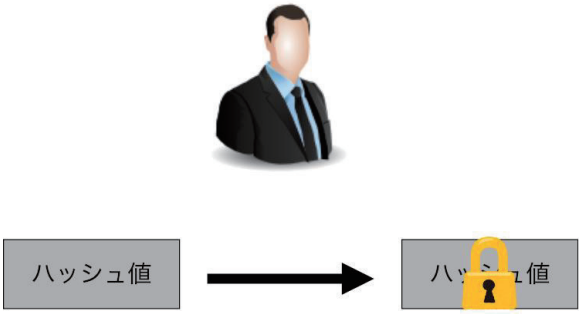
電子署名と公開鍵暗号方式を用いたデータの送信を「A さんが B さんにデータを送信すること」を例にして説明します。

説明するにあたり、前提として A さんと B さんはそれぞれ秘密鍵と公開鍵を作成しており、公開鍵の交換は終わっているものとします。

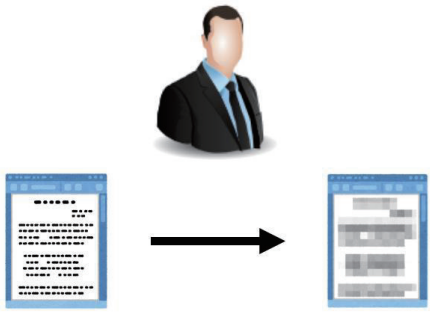
1. A さんは送信するデータのハッシュ値を求める。



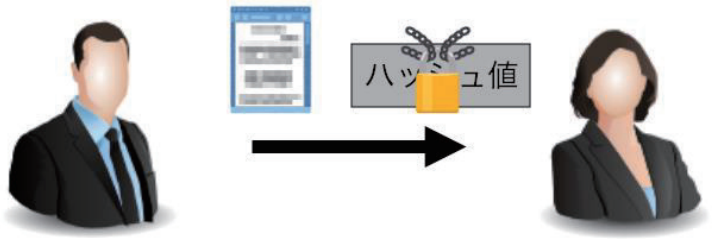
2. A さんは求めたハッシュ値を A さんの秘密鍵で暗号化する。



3. A さんは送信するデータを B さんの公開鍵で暗号化する。

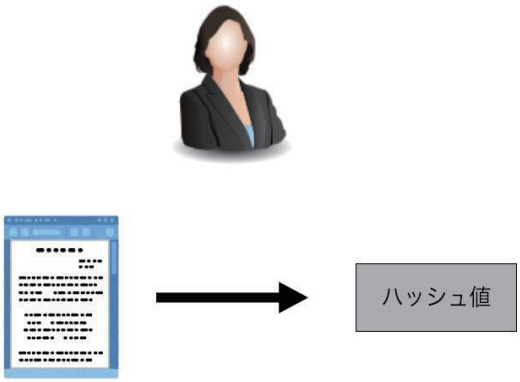


4. A さんから B さんに暗号化したデータとハッシュ値を送信する。



5. B さんは受け取ったデータを B さんの秘密鍵で復号する。

6. B さんは復号したデータのハッシュ値を求める。



7. B さんは受け取った暗号化されたハッシュ値を A さんの公開鍵を用いて復号する。

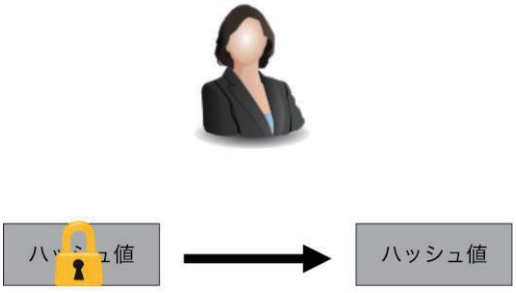


図 4.10 電子署名の流れ

B さんが受け取ったハッシュ値を A さんの公開鍵を用いて復号することができると、通信相手は A さんの公開鍵に対応する秘密鍵を持っていることがわかります。また、復号されたハッシュ値と、平文から得たハッシュ値が同じであれば、通信の過程でデータの改ざんが行われていないことを確認することができます。

4.3.2 認証局

A さんと B さんの例を用いて説明した電子署名の仕組みには、大きな問題があります。それは、「**送信相手は公開鍵に対応する秘密鍵を持っているだけであり、本当に送信相手が A さんかどうかわからない**」点です。つまり、A さんになりすました人物により、不正が行われる可能性があります。これを防ぐためには、**使用した鍵のペアは本当に A さんのものであるか確認する仕組み**が必要になります。公開鍵が A さんのものであることを証明することができれば、ハッシュ値を対応する秘密鍵で暗号化したのは、A さん本人であることがわかります。

この問題は「信頼のおける第三者」を経由し、公開鍵が A さんのものであると確認して、鍵を入手するという方法により解決します。この「信頼のおける第三者」として機能するのが、「**認証局(公開鍵証明書認証局)**」です。認証局はインターネット上の身分証明書である、デジタル証明書の発行を行います。デジタル証明書には「公開鍵」と「持ち主」の記載があり、公開鍵がその持ち主のものであると証明することができます。また、認証局はデジタル署名の所有者が秘密鍵を紛失、流出した際や、期限切れのデジタル証明書の失効作業も行います。

4.3.3 電子署名の利用例

次に、実際に電子署名の仕組みを応用した利用例を紹介します。

SSL/TLS(Secure Sockets Layer/Transport Layer Security)

SSL/TLS とは、インターネットなどのコンピュータネットワークにおいてセキュリティを要求される通信を行うためのプロトコルです。通信内容の暗号化とサイト

の運営元の確認により、**通信相手の認証、通信内容の暗号化、改ざんの検出**を行うことができます。

HTTPS(Hypertext Transfer Protocol Secure)

HTTP 通信を SSL/TLS プロトコルを用いることにより、安全に行うための仕組みです。HTTPS ではハイブリッド型暗号方式が用いられており、共通鍵を公開鍵暗号方式で受け渡しを行い、送信するデータ自体は共通鍵暗号方式で受け渡しを行います。

SSL 証明書

HTTPS でアクセスを行う Web サイトには**認証局により発行された SSL 証明書が与えられています**。これにより、アクセスした Web サイトが本当にその Web サイトであるか確認することができます。

SSL 証明書は Web ブラウザで確認することができます。SSL 証明書は認証局が発行を行いますが、信頼された認証局が発行する SSL 証明書と、信頼されていない認証局が発行する SSL 証明書が存在します。信頼された認証局とはあらかじめ、SSL 通信を要求するクライアントに登録されています。それ以外の認証局が発行した証明書はクライアント自身で認証局の信頼を判断しなければなりません。

4.3.4 ブロックチェーンでの電子署名の利用

ブロックチェーンでの電子署名の利用例を紹介します。これまでと同様に Bitcoin を例に用います。Bitcoin では、通貨は UTXO としてブロックチェーンに記録されており、通貨の移転は保有する UTXO をトランザクションインプットとして新たな UTXO を作成することで行われます。**UTXO を利用する際には、通貨の保有者はトランザクションに署名**を行います。これにより **UTXO の保有者であることを証明**します。署名が行われたトランザクションは、署名が UTXO の保有者により行われたものであるかマイナーによって検証されます。検証により正当なトランザクションであると判断されるとブロックチェーンに取り込まれ、通貨の移転が終了します。

5. P2P ネットワーク

ブロックチェーンでは、P2P ネットワークというネットワーク方式が採用されています。これに対して、現在多くの場面で利用されているネットワークの方式は「クライアント・サーバー型ネットワーク」です。この章では、これらについて説明します。

5.1 クライアント・サーバー型ネットワーク

クライアント・サーバー型ネットワークは現在主流となっているネットワーク方式です。ネットワーク内には「**クライアント**」、「**サーバー**」と呼ばれる 2 種類のコンピュータが存在します。

私たちが普段 Web サイトの閲覧や、メールの送信などを行う際には、コンピュータやスマートフォンなどのデバイスは「クライアント」としてネットワークに存在しています。

これに対して、Web サイトの情報の保持や、メールの転送などを行うのが「サーバー」です。サーバーはその役割によってメールサーバーや、Web サーバーなどに分類する事ができます。一般的にサーバーはサービスを提供する企業内や、データセンターで管理されており、日常生活で目にすることはほとんどありません。

以下の図 5.1 に Web サイトが表示されるまでの流れを説明します。

1. クライアントがサーバーにリクエストを送る
2. サーバーが受け取ったリクエストを処理する
3. サーバーがクライアントにレスポンスを返す

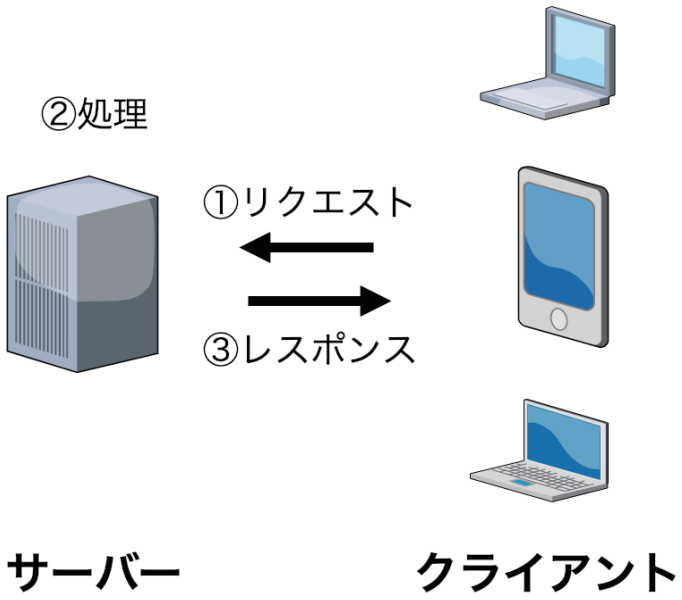


図 5.1 クライアント・サーバー型ネットワークのイメージ

5.1.1 クライアント・サーバー型ネットワークの特徴

仕様の変更が容易

クライアント・サーバー型ネットワークでは**システムの管理をサーバー側が行なっています**。そのため仕様を変更する際には、サーバーが保持しているデータやプログラムに修正を加えるだけでよく、クライアント側で操作を行う必要はなく、簡単に行う事ができます。

また、サーバーにトラブルがあった時も、その修正を P2P ネットワークと比べると簡単に行うことができます。

単一障害点が存在する

多数のコンピュータで構成されるシステムで、特定の箇所が機能不全に陥るとシステム全体が機能しなくなる箇所を「単一障害点」と言います。クライアント・サーバー型ネットワークでは、全ての通信がサーバーを経由するため、サーバーが機能しなくなると通信ができなくなり、システム全体が機能不全に陥ります。そのため、**サーバーが単一障害点**となっていると言えます。

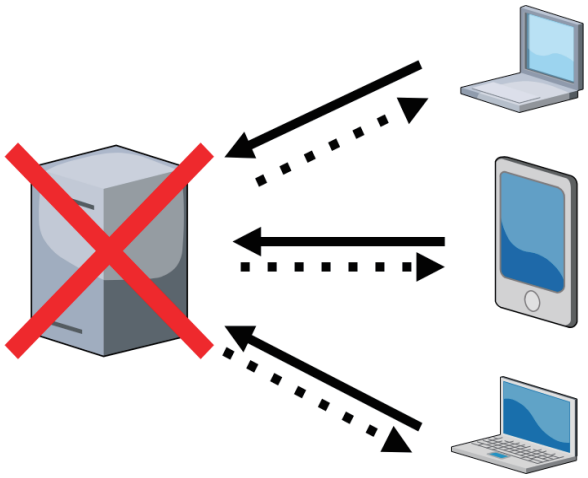


図 5.2 単一障害点

特定の箇所に負荷が集中する

通信量が増加することで、**サーバーへ大きな負荷**がかかります。これにより、サーバーでの処理が遅れによる通信速度の低下や、最悪の場合には単一障害点であるサーバーが停止することにより、システム全体が停止してしまうことが考えられます。

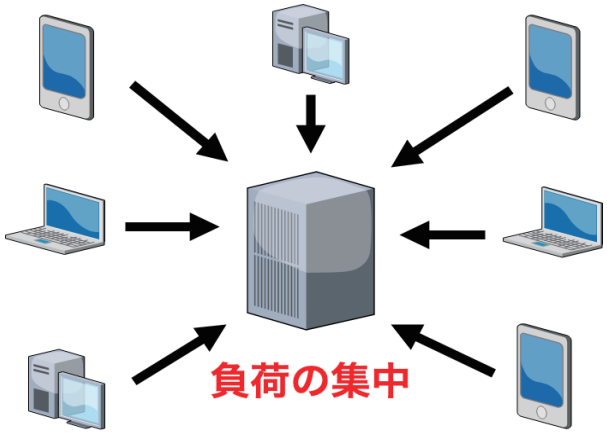


図 5.3 サーバーへの負荷の集中

管理者が存在する

サーバーには管理者が存在します。**管理者はシステムに関して特別な権限を持っており**、システムの管理を一括して行うことができます。その反面、管理者自体がこの権限を悪用することや、管理者権限が奪われることによりデータやプログラムの改ざんなどが行われる可能性もあります。

5.2 P2P ネットワーク

P2P ネットワークとは「Peer to Peer ネットワーク」を省略したもので、ネットワークを構成するそれぞれのコンピュータが互いに情報を伝達し合うことにより、通信が行われるネットワークです。P2P ネットワークを構成するそれぞれのコンピュータは「ノード」と呼ばれます。ノードには木の節という意味があり、各ノードからネットワークが枝のように広がることからこのように呼ばれています。

5.2.1 P2P ネットワークの特徴

高い稼働率

P2P ネットワークでは、**特定のノードに依存することなくノード間で通信**を行うことができます。そのため、ネットワーク内のいくつかのノードが機能しなくなった場合にも、システム全体が機能しなくなることは少ないと言うことができます。

高いスケーラビリティ

クライアント・サーバー型ネットワークでは全ての通信がサーバーを経由します。これに対して、P2P ネットワークでは特定のサーバーに依存することなく、ノード間で通信が行われます。そのため、ノード数や通信量が増加した際にも**特定の回線やコンピュータに負荷がかかることはなく**、通信速度の低下やシステム全体の機能停止などの問題が発生する可能性は低いと言うことができます。

データの書き換え

P2P ネットワークで複数のノードを中継しデータを受け取る際には、中継したノードにより**データが書き換えられる**リスクがあります。

仕様の変更

P2P ネットワークに参加するために利用する、ソフトウェアに仕様の変更があった際に、ネットワーク内の全てのノードにそれを伝達し、全員が新たなバージョンを利用し始めることは、クライアント・サーバー型の仕様変更と比べて**大きなコストがかかります**。

5.2.2 ピュア P2P とハイブリッド P2P

P2P ネットワークは大きく「ピュア P2P」と「ハイブリッド P2P」の 2 種類に分けることができます。この 2 種類についての説明を行います。

ピュア P2P

ネットワークが**ノードのみで構成されている** P2P ネットワークです。サーバーに依存することなくシステムが機能します。ピュア P2P の中にも、ノード間に権限の差があるものや、全員が同じ権限を持つものなど様々な種類があります。

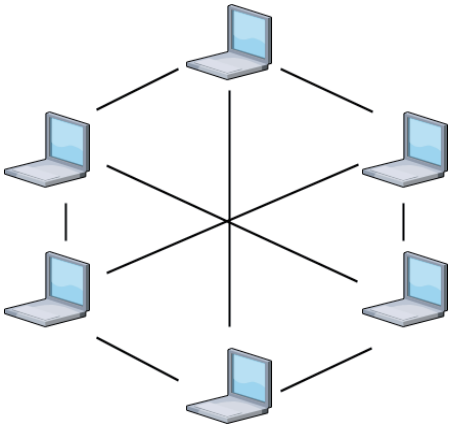


図 5.4 ピュア P2P

ハイブリッド P2P

ネットワークがノードのみで構成されているのではなく、**サーバーが存在するネットワーク**です。つまり、P2P ネットワークとクライアント・サーバー型ネットワークを組み合わせたネットワーク方式であり、それぞれの特徴を兼ね備えています。ハイブリッド P2P のネットワーク内に存在するサーバーの役割は、目的により異なりますが、ネットワーク内のノードの位置をサーバーが把握しておくことで、情報の伝達を効率的に行うことができる点などが挙げられます。

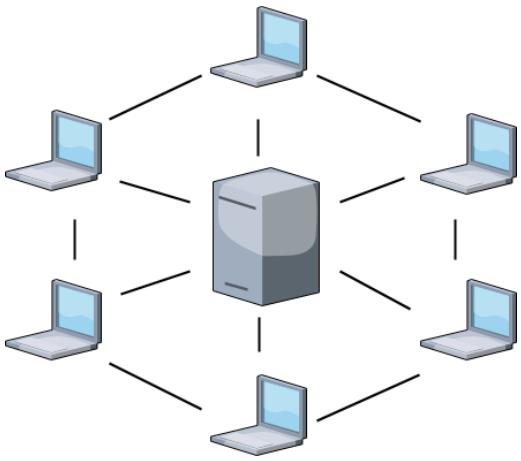


図 5.5 ハイブリッド P2P

5.2.3 構造化オーバーレイと非構造化オーバーレイ

P2P ネットワークはノード同士の接続の際に、接続するノードの制約の有無により「**構造化オーバーレイ**」と「**非構造化オーバーレイ**」の 2 種類に分類する事ができます。これらの特徴について説明します。

構造化オーバーレイ

構造化オーバーレイとは、それぞれの**ノードの接続先があらかじめ定められている P2P ネットワーク**です。各ノードに ID が割り振られ、その ID に従って接続相手が決定されます。これにより、リング型やツリー型のオーバーレイネットワークが構築されます。

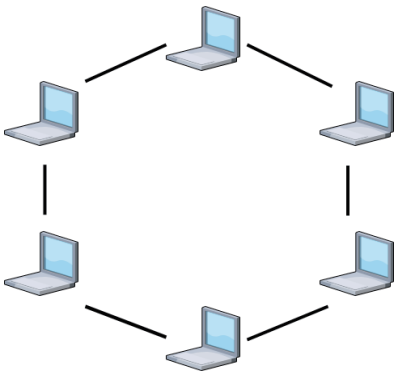


図 5.6 リング構造

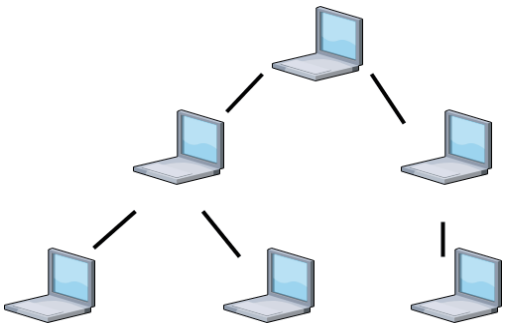


図 5.7 ツリー構造

メッセージの転送効率が良い

構造化オーバーレイではノードの接続に一定の規約を設けることで、相手の正確な場所はわからなくとも、徐々に相手のノードの場所を絞り込むことができます。例えるならば、住所を都道府県、市町村、町、番地のように絞り込んでいくことに似ています。これにより**情報の伝達を効率的に行う**ことができます。

高いスケーラビリティ

構造化オーバーレイでは、ノードが規約の元に接続されているため、相手のノードの場所を大まかに把握することができます。これによりネットワーク内に**伝達されるメッセージの量を抑える**ことができます。そのため、ノード数が増加に対して、メッセージの数の増加に伴う通信速度の低下などを回避することができます。

非構造化オーバーレイ

各ノードがネットワーク内の他のノードと接続する際に、**ノードの選択に制約のない設計のオーバーレイネットワーク**です。

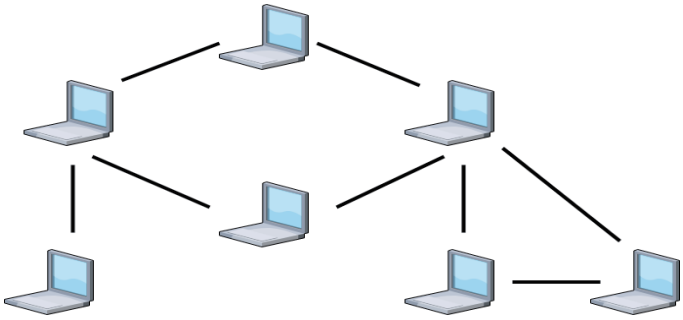


図 5.8 非構造化オーバーレイの図

隣接するノードを選ぶ際に制約がない

ネットワークに参加する際には、自身が知っているどのノードと接続しても構いません。

スケーラビリティ

非構造化オーバーレイにおけるメッセージの伝達方法では、ネットワーク内のノード数の増加に伴い、ネットワーク内で伝達されるメッセージの数が飛躍的に増加してしまいます。それにより、回線が混み合い、通信速度が低下する可能性があります。そのため、ネットワーク内のメッセージが増え過ぎないように、**メッセージの転送回数や、メッセージの生存時間などの制限を設ける**必要があります。

スーパーノード

メッセージを送信する際に、隣接するノードに対してのメッセージの伝達を繰り返し、相手に届ける方法では、メッセージを受け取る必要のないノードにまでメッセージが届き、効率が良くありません。そこで、採用されたのが「スーパーノード」です。

多くの場合、スーパーノードはネットワーク内の**ノードの情報を保持**しています。これにより、メッセージの伝達効率を上げることができます。スーパーノードはネットワーク内に存在する一般ノードの中から、ネットワークに接続されている時間が長く、回線が安定しているなど様々な項目を満たし、安定したノードが選ばれることもありますが、システムの管理者があらかじめ準備する場合もあります。

Let's TRY

５章演習

グループワーク

現在、私たちが利用しているサービスの多くはクライアント・サーバー型ネットワークが用いられており、P2P ネットワークが利用されているサービスは多くありません。P2P ネットワークがクライアント・サーバー型ネットワークと比べて利用されていない点について、その理由を以下の表に示したそれぞれの特徴を踏まえて 4～5 人のグループで考えてください。

	クライアント・サーバー型	P2P型
仕様の変更	簡単	難しい
可用性	高い	低い
スケーラビリティ	低い	高い
データの書き換え	発生しにくい	発生しやすい

グループで話し合った結果を、代表者が発表してください。

回答例

- クライアント・サーバー型ネットワークに比べて、P2P ネットワークは～～～なため利用が難しいのではないか？
- P2P ネットワークの～～～の特徴が利用の難しい最大の要因ではないか？

5.3 P2P ネットワークの利用例

ここからは、P2P ネットワークが用いられたサービスを 4 つ紹介します。

5.3.1 Bitcoin（ビットコイン）

Bitcoin では、**非構造化オーバーレイのピュア P2P ネットワーク**が用いられています。Bitcoin で P2P ネットワークが用いられている最大の理由は、「**非中央集権**」の実現です。P2P ネットワークを用いて、特定の管理者に依存することなく、参加者が対等な権限を持ち、自律的にシステムを運営する仕組みを実現しました。

ネットワークへの参加

Bitcoin のネットワークに参加するためには、Bitcoin のクライアントソフトを利用します。クライアントソフトを起動すると、あらかじめクライアントソフトが保持している仕組みにより、Bitcoin ネットワーク内のノードの IP アドレスが 1 つ以上返ってきます。このノードに接続することにより、ビッコインのネットワークに参加することができます。ネットワークに参加した後に、隣接するノードから他のノードの IP アドレスを教えてもらい、複数のノードと接続することで安定した接続を確立することができます。

5.3.2 Ethereum（イーサリアム）

Ethereum では、**非構造化オーバーレイのピュア P2P ネットワーク**が用いられています。Bitcoin と同様に、「**非中央集権**」の実現のために P2P ネットワークが用いられます。

ネットワークへの参加

Bitcoin と同様に Ethereum も専用のクライアントソフトを利用することで、ネットワークに参加することができます。Ethereum のクライアントソフトで最も利用割合が高いものは「Geth(Go-Ethereum)」です。

Ethereum には実際に価値を持った通貨が取引されるネットワークだけでなく、アプリケーションのテスト用のネットワークが複数準備されています。Geth を起動する際にオプションで参加するネットワークを選ぶことができます。

5.3.3 Skype(スカイプ)

インターネット電話サービスである Skype では、**非構造化オーバーレイのハイブリッド P2P ネットワーク**を採用しています。

Skype サーバー

Skype はハイブリッド P2P ネットワークを採用しており、ネットワークの中にサーバーが存在します。このサーバーは、ユーザーの初期登録やログイン認証、セキュリティ上、一元管理が必要なものや、技術的に分散処理が向いていない機能を提供しています。

通常ノード

ユーザーがコンピュータ上で Skype アプリケーションを起動すると、Skype のログイン用サーバーで認証を受けた後、最初は「通常ノード」として Skype ネットワークに参加することになります。その後、スーパーノードに依頼して通信相手を見つけ、見つかった相手と直接接続を確立します。

スーパーノード

スーパーノードは、通常ノードとしての機能の他に Skype に参加しているノードの情報を記録したり、探索したりする機能も担当する特殊なノードです。一般のユーザーが通常ノードとして Skype ネットワークに参加すると、Skype ユーザー名や IP アドレス、ポート番号、ログイン状態などを Skype ネットワークに登録しますが、これらのノード情報はサーバーではなくスーパーノードにより管理されています。スーパーノードは通常ノードの中から選出されますが、どのようなノードでもスーパーノードになることができる訳ではなく、以下の条件を満たす必要があります。

- 高性能 CPU と大容量のメモリを持つ
- 広いネットワーク帯域を持つ
- グローバル IP アドレスを持つ
- 内向きパケットを常に受け入れている
- 連続稼働している

これらの条件がスーパーノードになるための最低条件であり、スーパーノードの数は全ノードの 1%程度になるように調節されています。また、スーパーノードはネットワークから自動的に選ばれ、選ばれたことはそのノードには通知されません。

5.3.4 BitTorrent(ビットトレント)

BitTorrent とは、P2P ネットワークを用いたファイル転送プロトコル及びその通信を行うソフトウェアです。BitTorrent のネットワーク上に分散して保存されたファイルをダウンロードする際には、BitTorrent のプロトコルを実装したクライアントソフトウェアを利用します。ネットワークの形式としては、**非構造化オーバーレイのハイブリッド P2P ネットワーク**です。どのファイルをどのノードで管理しているかなどファイルに関する情報はサーバーが保持し、ファイル自体は、ネットワーク内の複数のノードが分割して保有しています。

トラッカー

トラッカーとはファイルを保存しているノードの情報を持っているサーバーのことを言います。

トレントファイル

トレントファイルとはネットワーク上にアップロードされているファイルをダウンロードする際に必要な情報が記録されているファイルです。トレントファイルは自体は、BitTorrent のネットワーク内に保存されるのではなく、BitTorrent のサービスからは独立した Web サーバー上に保存されます。

ファイルのダウンロード

BitTorrent のネットワーク上からファイルをダウンロードする方法について説明します。

- Web からダウンロードしたいファイルのトレントファイルを入手する
- トレントファイルの情報を元に、トラッカーにアクセスし、ファイルを保持しているノードの情報を得る
- ノードから直接ダウンロードを行う

ファイルのダウンロードを行った後にも、BitTorrent クライアントを起動させたままにすると、自身のノードがファイルの保管場所として認識されます。これにより、自身のノードから他のノードがファイルのダウンロードを行うことができるようになります。

ファイルの配信

次に、BitTorrent にファイルのアップロードを行う方法について説明します。

- アップロードするファイルについてのトレントファイルを作成する。
- トレントファイルを Web サーバーに配置する。
- 配信するファイルを保持した状態で BitTorrent クライアントを起動する。

これで、ファイルのアップロードのために必要な作業は終わりですが、この状態では自身の手元にしかファイルが存在せず、自身が BitTorrent クラインントソフトを終了すると、ネットワーク上にファイルは存在しなくなり、誰もダウンロードする事ができなくなります。それを防ぐためには、クライアントソフトを起動し続けるか、自身の手元からダウンロードが行われ、ネットワーク上の複数のノードにファイルが存在する状態になるまで待つ必要があります。

6. マイニングとコンセンサスアルゴリズム

この章ではブロックチェーンの仕組みについて理解する上で特に重要となるマイニングとコンセンサスアルゴリズムについて説明します。

6.1 マイニング

マイニングとは「**ネットワーク内から代表者を選出し、トランザクションをまとめてブロックを作成すること**」であり、マイニングを行うノードをマイナーと呼びます。

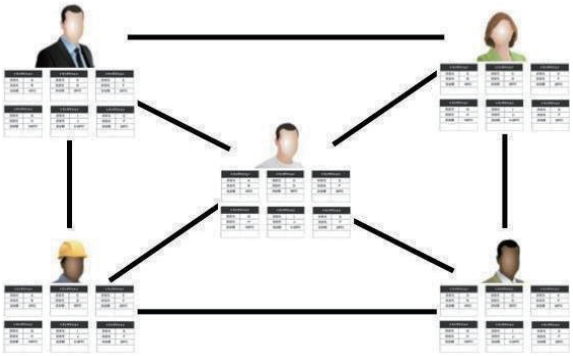
マイニングを行うためには、ネットワーク内で作成される新たなトランザクションを受け取るできる状態にしておかなければなりません。具体的には、常にブロックチェーンのクライアントソフトを起動させ、コンピュータをインターネットに繋いでおく必要があります。また、ネットワークの中からブロックを作成する代表者を選出する手法によっては、コンピュータによる大量の計算が必要になるものもあります。

このような点からマイニングを行うためには電力が必要になることがわかると思います。電力を得るためには当然電気代がかかり、つまり**マイニングを行うためにはお金がかかります**。では、マイナーがこのようなコストを投じてまでブロックを作るのはなぜなのでしょうか。

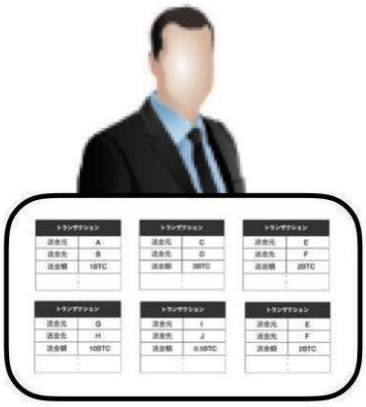
これはマイナーがブロックを作成すると「**マイニング報酬**」という報酬を受け取ることができるためです。マイナーはこの報酬目当てにマイニングを行っています。これを踏まえて以下の図 6.1 にマイニングの流れを示します。

1. マイナーはブロックチェーンネットワークに参加して、

トランザクションを集める



2. トランザクションをまとめてブロックを作る



3. ブロックを作ると報酬を得る

図 6.1 マイニングの流れ

マイニング報酬は 2 種類の報酬の合計であり、1 つは「**トランザクション手数料**」、もう 1 つは「**新規発行通貨**」です。それぞれについて説明します。

6.1.1 トランザクション手数料

トランザクション手数料とは、**トランザクションの発行者がトランザクションに付け加える手数料**です。

ネットワークの中から選出されブロックを作成したマイナーは、自身が作成したブロックに含まれるトランザクションの手数料を、自身の報酬にすることができます。

トランザクション手数料について、Bitcoin を例に挙げて説明します。Bitcoin ではトランザクション手数料は一定額に定められているのではなく、トランザクション発行者が任意に設定することができます。このトランザクション手数料はトランザクションがマイナーにより処理される優先度の 1 つの指標になり、高い手数料を設定する程トランザクションがブロックに取り込まれる優先度が高くなり、取引にかかる時間を短縮することができます。ただし、トランザクション手数料はあくまでも 1 つの指標であり、これ以外にもトランザクションのデータサイズや、取引額なども考慮されます。

6.1.2 通貨の新規発行

多くのブロックチェーンでは**ブロックが作成される度に、仮想通貨の新規発行が行われます**。新しく発行された通貨は、その**ブロックを作成したマイナーに報酬として支払われる**ことで市場に出回ります。

Bitcoin では平均 10 分に 1 回ブロックが作成され、2019 年 1 月の段階で 1 つブロックを作成したマイナーには 12.5BTC の報酬が支払われます。この報酬額はブロックが 21 万個作成される度に半減するようにあらかじめプログラムされており、Bitcoin のブロックの作成間隔は平均 10 分であることからと約 4 年に一度半減する計算になります。

一定時間毎に一定額の通貨の新規発行が行われ、その通貨の総量が決まっているという仕組みが金や銀などの地下資源の採掘に似ているため、「採掘を行う」という意味の「Mining(マイニング)」という単語が用いられています。

6.2 コンセンサスアルゴリズム

6.2.1 コンセンサスアルゴリズムとは

マイナーは新たなブロックを作成し、それがネットワーク内で正当なものであると判断されると報酬を得ることができます。マイナーがマイニングを行う最大の目的はマイニング報酬であることから、自分以外のマイナーが作ったブロックを承認することは、自身にとっては不利益になると考えることができます。このような問題が発生しないようにブロックチェーンでは、自分以外のマイナーが正当なブロックを作ったことを認めることでも、マイナーに利点があるような仕組み作りがされています。

パーミッションドブロックチェーンではブロックチェーンへの参加者が制限されるため、参加者全員に一定の信頼があり、全体として１つの意思決定を行うことは比較的簡単です。しかし、Bitcoin や Ethereum などのパブリックブロックチェーンでは誰でも参加することができるため、悪意を持った参加者がネットワークに紛れる可能性があることから、意思決定は簡単に行うことはできません。

このような状態でも悪意を持った参加者に左右されることなくネットワークとして１つの正しい判断を行うための手法が「コンセンサスアルゴリズム」です。コンセンサスアルゴリズムでは、**ネットワークの参加者の大半が経済的合理性に基づいて行動した際に、ネットワーク全体として正しい判断を行うことができます**。コンセンサスとは「複数の人の合意や意見の一致」という意味があり、アルゴリズムには「問題を解くための手法」という意味があります。つまり、ブロックチェーンの**ネットワークの参加者全員が１つの判断をするための手法**です。

6.3 コンセンサスアルゴリズムの種類

ここでは多数あるコンセンサスアルゴリズムの中から 4 つを紹介します。

6.3.1 Proof of Work

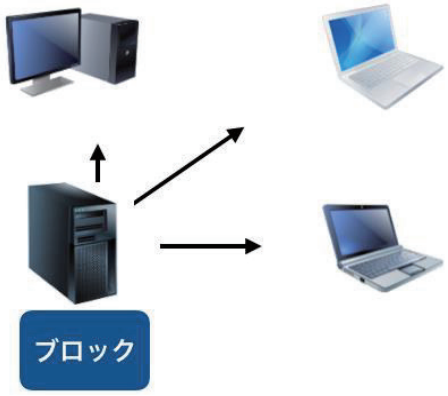
Proof of Work は Bitcoin をはじめとして複数のブロックチェーンで使用されているコンセンサスアルゴリズムです。「**最も早く計算問題を解くことができた人のブロックを採用する**」という方法でブロックの作成者を決めます。

ブロックにはトランザクションが含まれているだけでなく、ブロック内のヘッダーと呼ばれる部分にそのブロック固有の情報を記述する欄があり、その中に「**ナンス**」という項目があります。このナンスには任意の数値が入り、この値を変更することで、**ブロック全体のハッシュ値を変える**ことができます。定義で出てきた計算問題とは、**ブロックのハッシュ値を一定値以下にする**ことのできる適切なナンスを探し当てることを言います。この計算は数式を計算すれば値が求まるような問題ではなく、総当たり的に計算を繰り返し、正答を求めなければならない問題です。総当たり的とは大量の「クジ」を渡され、それを引き続けることで当たりを見つけることに似ています。様々なナンスを試しながらブロックのハッシュ値を求め続け、**ハッシュ値が一定以下になるナンスを最も早く求めることのできたマイナーが、ブロックを作成する**ことができます。

1. マイナーがナンスの計算をする

2. あるマイナーがナンスを見つける

3. ブロックを作成し、伝達する



4. ブロックを検証し、保存する

図 6.2　Proof of Work の処理の流れ

Let's TRY

６章演習

ここでは実際にハッシュ関数を利用しながら Proof of Work の体験を行います。

準備

- 巻頭で作成した仮想マシンを立ち上げる。
- 仮想マシン上でターミナルを開く。
- ターミナルに `$ irb` と入力すると、Ruby のコンソールが開きプログラムが書けるようになる。
- `$ require 'digest'` と入力する。
- true と返ってくると、ハッシュ関数が利用できるようになる。

Proof of Work の体験

- ハッシュ関数 SHA-256 に数値を入力して、16 進数表記のハッシュ値を出力する。
    - `$ Digest::SHA256.hexdigest('数値')`
- 出力されるハッシュ値の先頭が 00 から始まる入力値を 10 分以内に探す。

Bitcoin の Proof of Work の確認

- Bitcoin のブロックチェーンエクスプローラを開く。
  (https://www.blockchain.com/ja/explorer)
- Bitcoin ネットワークで作成されたブロックのハッシュ値を確認する。
- 自身が 10 分間で行った計算量と比較して、Bitcoin でのマイニングに膨大な計算資源が投じられていることを確認する。

Proof of Work のメリット

誰でも参加することができる

マイニングに参加したいブロックチェーンのクライアントソフトを利用することで、誰でも参加することができます。

マイニングの権限が平等にある

ブロックを作成することのできる権限は全員が均等に持っています。

Proof of Work の課題点

電力問題

Proof of Work ではナンスを求め、ブロックを作成するための競争を世界中のマイナーが行うため、**大量の電力が消費**されます。2017 年に Bitcoin のマイニングのために全世界のマイナーが消費した電力は、国別の消費電力のランキングで 50 位前後に相当します。

マイニングの寡占化問題

ブロックを作成する権利はマイナー全員に平等にあるにも関わらず、**ブロックの作成は一部のマイナーによる寡占化**が進んでいるのが現状です。マイニングの寡占化が進むと、ブロックチェーンが攻撃されるリスクが高まります。ブロックチェーンに対する攻撃に関してはこの章の「二重支払い問題」で説明を行います。

マイニングの寡占化は、ナンスを求めるための計算問題の難易度上昇が原因として発生します。マイナーの増加とコンピュータの性能の向上による計算速度の上昇により、一定時間に試行されるナンスの数は飛躍的に増加しました。このため、計算の難易度が同じならば、計算終了までの時間は短くなります。これを防ぐためにブロックチェーンにはブロックの作成時間が一定に保たれるように、プログラムにより自動的に計算の難易度が調整される仕組みが備わっています。このようにして計算の難易度が自動的に上昇しました。

これにより、高性能なコンピュータや大量の電力を使用できる環境など、マイニングのための専用の設備を準備しなければマイニングの競争には勝つことができな

くなりました。一般的なコンピュータでマイニングを行ってもブロックを作成し、報酬を得ることのできる可能性は極めて低くなっています。

6.3.2 Proof of Stake

Proof of Stake は Ethereum で導入予定のコンセンサスアルゴリズムです。**保有する通貨の量に応じて、ブロックの作成確率が決まります**。Proof of Stake では Proof of Work のように大量の計算は行われないので計算量、つまり消費電力を削減することができます。

Proof of Work と異なる点

- 通貨の保有量が多いほど、ブロックを作成できる確率が高くなる。
- 満たさなければならないブロックのハッシュ値の基準が緩くなることで、ブロックが作成できる確率が高くなる。

Proof of Stake のメリット

消費電力が少ない

Proof of Stake では Proof of Work のような計算の競争は発生しないため、消費電力が圧倒的に少ないという利点があります。

Proof of Stake の課題点

通貨の流動性の低下

Proof of Stake では、通貨を多く所有する人ほど多くの通貨を手に入れることができます。そのため、保有している通貨を手放さない人が増え、**通貨の流動性が低下**する可能性があります。

貧富の差の拡大

Proof of Stake では**通貨を多く所有する人ほど多くの通貨を手に入れる**ことができるため、貧富の差の拡大が懸念されています。

ブロック作成の条件

Proof of Stake が用いられているブロックチェーンで、ブロックの作成を行うためには**一定額以上の通貨を保有する必要**があります。ブロックを作成するために保有しなければならない通貨の最低量が高く設定された場合には、既に多くの通貨を保有している人のみが新たな通貨を得ることができ、通貨の集中が発生してしまいます。

6.3.3 Proof of Importance

Proof of Importance はブロックチェーン NEM などで使用されているコンセンサスアルゴリズムです。仕組みは Proof of Stake に似ていますが、Proof of Importance では**通貨の保有量だけでなく、取引を行った通貨量や取引相手が考慮**されます。これらの項目を考慮した上で**各ノードに Importance Score がつけられ、これによりブロックが作成される確率が決まります**。通貨の取引量を重要度に組み込むことで通貨の流動性低下の問題を解決しました。

Proof of Importance のメリット

消費電力が少ない

Proof of Stake と同様に、Proof of Work に比べると、消費電力が圧倒的に少ないという利点があります。

通貨の流動性低下の解決

Proof of Importance では重要度の指標に、**通貨の取引量が考慮されることで、Proof of Stake で問題となった通貨の抱え込みを防ぐ**ことができ、通貨の流動性の低下を抑えることができます。

Proof of Importance の課題点

貧富の差の拡大

ブロックの作成には Proof of Stake と異なり、通貨の保有量だけでなく通貨の取引量などが考慮されています。これにより単純に多くの通貨を保有している人が多

くの報酬を受け取ることはできませんが、多くの通貨を保有している人の方が取引の量が大きくなることや、ブロックの作成のために最低限保有しなければいけない通貨の量が定められているなど、**依然として多くの通貨を保有する人が優位**になる仕組みには変わりません。

6.3.4 Practical Byzantine Fault Tolerance(PBFT)

PBFT は**コンソーシアムブロックチェーンで使用されることの多いコンセンサスアルゴリズム**です。今回紹介した 3 つのアルゴリズムと大きく異なるのは、ブロックの作成を誰もが行うことができるのではなく、ネットワーク内の代表者のみが行うことができる点です。

PBFT の特徴

PBFT では、ネットワーク内のノードに権限の差があり、トランザクションの検証や、ブロックの作成を行うことができるノードを Validating-Peer と呼び、それ以外のノードを Non-Validating-Peer と呼びます。この Validating-Peer における多数決により、ネットワークの意思決定が行われます。

PBFT の流れ

1. Validating-Peer から 1 つのリーダーノードを選出する
2. リーダーノードが Non-Valadating-Peer からトランザクションを受け取る
3. リーダーノードは他の Validating-Peer にトランザクションを送信する
4. トランザクションを受け取った Validating-Peer は受け取ったトランザクションが改ざんされていないことを確認し、その結果を他の Validating-Peer に伝える
5. それぞれの Validating-Peer は一定数の Validating-Peer から「トランザクションが改ざんされていない」という報告を受けると「トランザクションは正しく配信されている」と判断し、他の Validating-Peer にその旨を伝える

6. それぞれの Validating-Peer は一定の数の Validating-Peer から「トランザクションが正しく配信された」と報告を受けると、トランザクションの処理を実行し、その結果を記録する
7. トランザクションを実行したことを、Non-Validating-Peer に伝える
8. Non-Validating-Peer は一定数の Validating-Peer からトランザクションが記録されたと報告を受けると、トランザクションが実行されたとする

PBFT のメリット

ファイナリティーがある

Validating-peer の合意によってのみブロックが作成されるため、**取引終了のタイミングがわかりやすい**という特徴があります。

処理速度が速い

Proof of Work などのように適切なハッシュ値を求めるための計算を必要としないため、処理速度が速くなります。

PBFT のデメリット

Validating-Peer の数に制約がある

Validating-Peer が一定数以下になるとシステムが停止してしまいます。また、Validating-Peer 同士は全てのノード間で通信を行う必要があるので、**ノード数が増加すると飛躍的に通信量が増加**し、それに伴い処理に時間がかかるようになります。

6.4 二重支払い問題

オンラインで通貨を取引する際には「二重支払い」という問題を解決しなければなりません。二重支払いとは言葉の通り、同じ通貨が 2 回以上使われてしまう問題です。現金で支払う際には、紙幣や硬貨といった紙や金属の移動が伴い、手元から通貨がなくなるので、同じ通貨を二度使うことはできません。これに対して、オンラインでの通貨の取引では通貨の移転に関するデータが記録されるだけであり、実際に物質が移動するわけではないので、不正なデータを作ることで二重支払いが行われてしまうのです。

ブロックチェーンではこの二重支払いの問題は完全に解決したということはできません。**行われる可能性は極めて低いですが、二重支払いを行うための攻撃方法は存在**しているためです。

まずはブロックチェーンにおける二重支払いの問題の理解に必要な、ブロックチェーンの分岐について説明します。

6.4.1 ブロックチェーンの分岐

ブロックチェーンは基本的に、1 つのブロックに対して 1 つのブロックが繋がっています。しかし、稀に 1 つのブロックを親として 2 つ以上の子ブロックが繋がり、ブロックチェーンに分岐が発生することがあります。ブロックチェーンの分岐が発生するタイミングとしては考えることができるのは、マイニングの際に**同時にブロックが作成**された時、または悪意のあるマイナーが**不正目的であえてブロックを分岐**させた時などが考えられます。

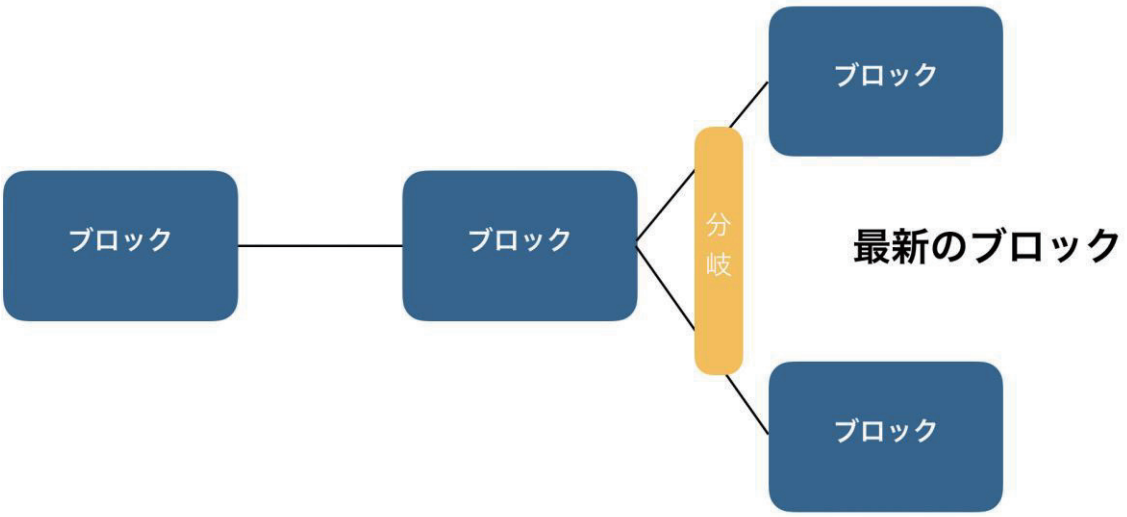


図 6.3 ブロックチェーンの分岐

ブロックチェーンは通常**一列に並んでいるブロックに含まれるトランザクションのみを、正式な記録として採用**します。つまり、分岐が発生すると、２つの異なる台帳が存在する状態になります。

ブロックチェーンではこのような状態を解決するために、ブロックチェーンが**自然１つに収束するような仕組み**作りがされています。多くのブロックチェーンでは分岐が発生した際には、**より信頼のあるブロックチェーンを正規のブロックチェーンとして採用**します。信頼のあるブロックチェーンの定義は様々ありますが、その中の１つとして**最も長く伸びたブロックチェーンを採用する**方法があります。この方式は Bitcoin に用いられています。

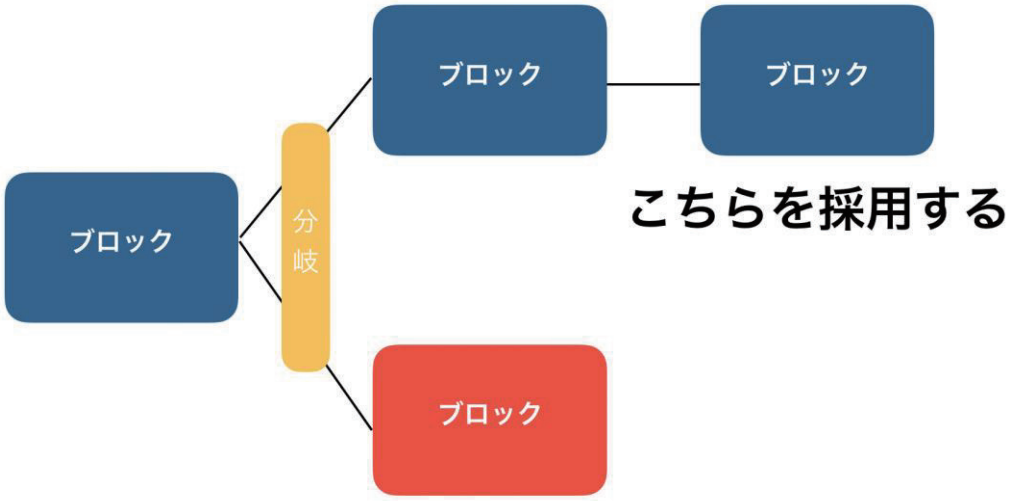


図 6.4 メインチェーンの採用

ブロックの承認数

ブロックがブロックチェーンに繋げられて時間が経つと、そのブロックの後ろに次々とブロックが繋がります。

あるトランザクションに注目した時、そのトランザクションがまだブロックに取り込まれていない場合、そのトランザクションは 0 承認であると言います。その後、時間の経過とともに、トランザクションがブロックに取り込まれると、トランザクションが 1 承認された状態であると言います。時間の経過とともに、そのトランザクションが含まれているブロックにブロックが 2 つ 3 つと繋がると、2 承認、3 承認と承認数は増えていきます。この承認とは**トランザクションやブロックに対する信頼**であると考えることができます。ブロックの後ろに、ブロックが繋がると信頼が増えると考えることができ、この承認数の仕組みが二重支払いやブロックチェーンの改ざん耐性の高さについて理解する上で重要になります。

6.4.2 二重支払い

ここからは実際に二重支払いの仕組みを説明します。ブロックチェーンでの二重支払いとは**最も信頼のあるメインのブロックチェーンが分岐により発生した他のブロックチェーンに切り替わった際に発生**します。

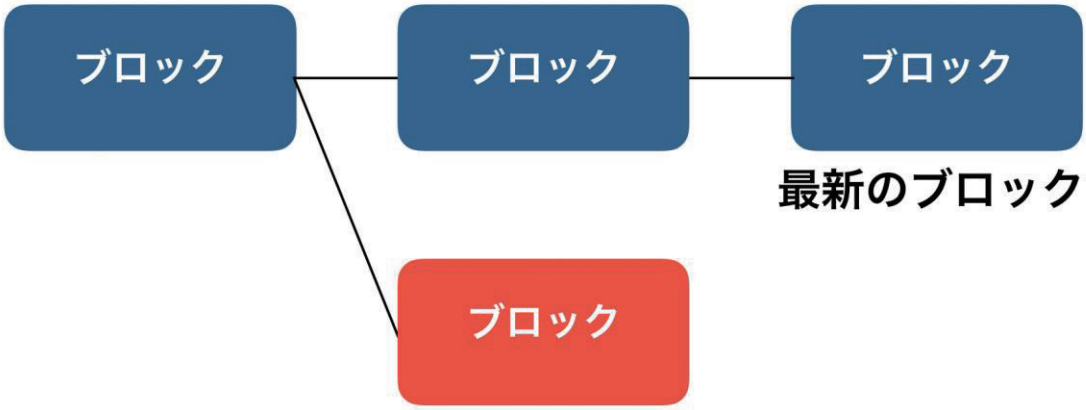


図 6.5 意図的なブロックの分岐

図 6.5 の上側のブロックチェーンが現在メインのブロックチェーンであるとします。ここで最新のブロックの、2 つ前のブロックから分岐を発生させます。分岐が発生したとしても、この状態では上のブロックチェーンの方が長いため、分岐して作成されたブロックチェーンは採用されず、メインのブロックチェーンには影響はありません。

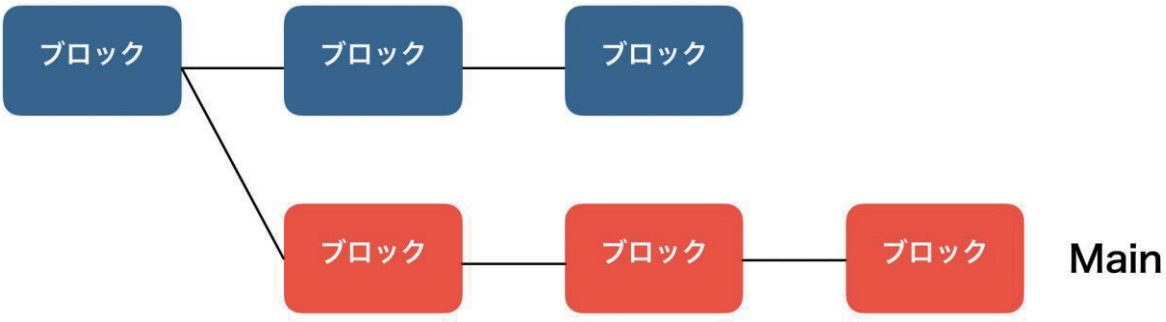


図 6.6 二重支払いの成立

次に、分岐したブロックチェーンにブロックが繋げられた場合を考えます。図 6.5 ではメインのブロックチェーンである上のブロックチェーンよりも分岐した下のブロックチェーンの方が長く伸びた状態になっています。この場合、**下のブロックチェーンがメインのブロックチェーンとして採用**されることになり、このブロッ

クチェーンに記録されたトランザクションがネットワーク全体として正式なものとして見なされます。

このとき、元々ブロックチェーンを構成していたブロックは分解され、ブロックに含まれているトランザクションは再度マイナーにより下のブロックチェーンを構成するブロックに取り込まれます。元々のブロックチェーンでは正当なものとして見なされていたトランザクションも、新たなブロックに含める際に既にブロックチェーンに記録されているトランザクションと矛盾があれば、そのトランザクションは無効なものとして見なされ、ブロックには取り込まれず棄却されます。

実際に、二重支払いについて具体例を用いながら説明します。

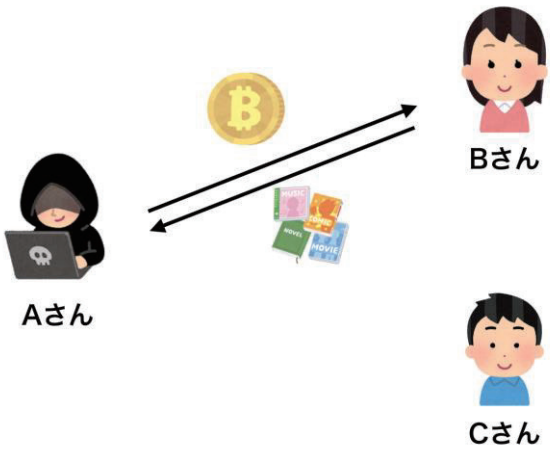


図 6.7 二重支払いのイメージ

A さんは B さんに仮想通貨を支払い、B さんは A さんに商品を渡す取引を行います。

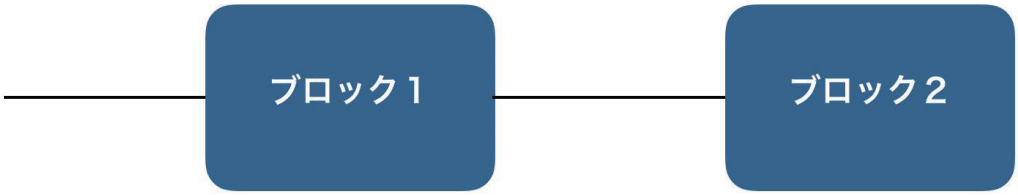


図 6.8 二重支払いの際のブロック

A さんから B さんへの仮想通貨の送金のトランザクションはブロック 2 に記録されました。

ここで A さんが B さんへの送金をなかったことにし、さらに B さんに送った通貨を再度用いて C さんと取引をしようとします。

そのために A さんは C さんへの送金のためのトランザクションを作成し、それを含めた不正ブロックを作成します。その後、不正ブロックを含んだブロックチェーンを伸ばし、B さんへの送金が記録されているブロックチェーンよりも長くなった瞬間に、分岐したブロックチェーンがメインのブロックチェーンとして採用されます。そうなると、A さんから B さんへの送金は取り消されてしまい、A さんから C さんへの送金が正しい記録としてみなされます。これにより B さんは商品を A さんに渡したのにも関わらず代金を受け取ることができません。また、A さんは同じ通貨を 2 度使うことで通貨価値の 2 倍の商品を得ることができました。

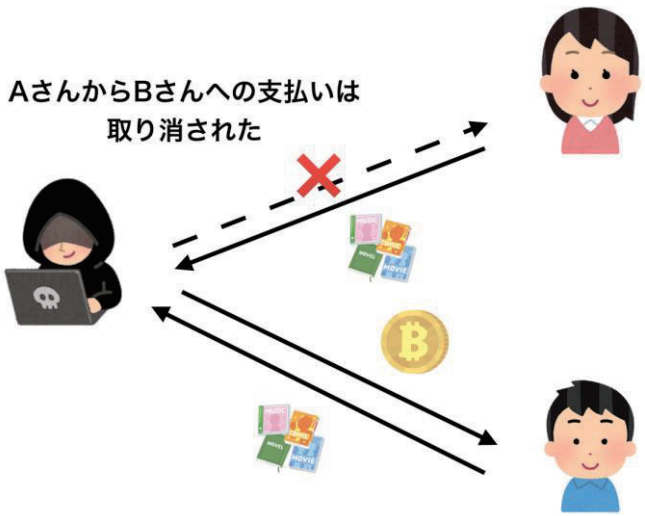
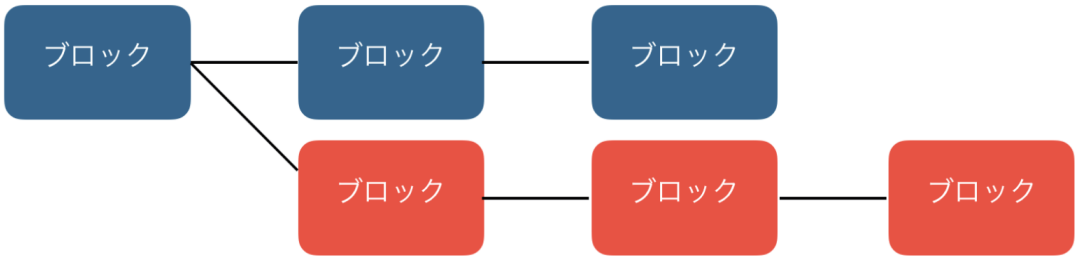


図 6.9 二重支払い

このような攻撃を防ぐ唯一の方法は、**ブロックの承認数を増やすこと**です。メインのブロックチェーンではマイニング報酬を求めて、世界中の多くのマイナーが膨大な計算資源を投じて計算を行っています。そのため、メインのブロックチェーンより短い状態からより長くブロックチェーンを伸ばすためには、メインのブロックチェーンに費やされる計算力に匹敵する計算力をブロックの作成に費やさなければ

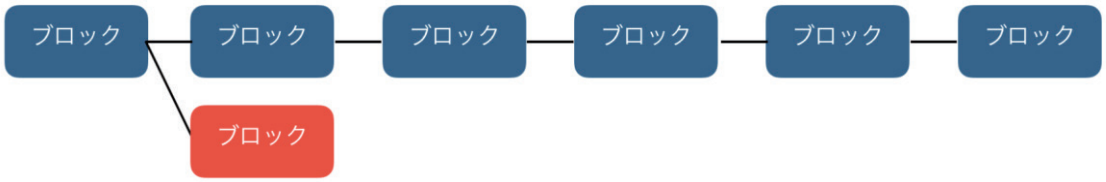
なりません。これには膨大なコストがかかり、Bitcoin や Ethereum などの参加者の多いブロックチェーンでは現実的ではありません。

しかし、メインのブロックチェーンに劣る計算力でも、より早くブロックを作ることは確率的には可能です。しかし、連続してそれを行うことができる確率はかなり小さくなります。そのため特定のブロックの後ろに繋がるブロック数を増やし、分岐が発生した際に、分岐したブロックチェーンがメインのブロックチェーンより長くなる可能性を低くすることで二重支払いを防ぐことができます。



最新のブロックから近い場所での分岐の際にはメインのチェーンの追いつくためのブロック数は少なくてよい

図 6.10 浅い場所からのブロックチェーンの切り替わり



最新のブロックから遠い場所で分岐が発生するとメインのチェーンに追いつくためのブロック数が多く、追いつくことが難しい

図 6.11 深い場所からのブロックチェーンの切り替わり

6.4.3 ブロックチェーンへの攻撃

51%攻撃

大量の計算資源を投じて、ネットワーク内の計算力の多くを占有し、**メインのブロックチェーンよりも早く不正なチェーンを伸ばす**ことにより、メインのブロックチェーンを切り替える攻撃です。「51%」と名付けられた攻撃ですが、実際にはネットワーク全体の計算力の 51%を占有しなくても、確率的にメインのブロックチェーンよりも早くブロックチェーンを伸ばし、メインのブロックチェーンを切り替えることは可能です。

正当なチェーンに費やされる計算力と、不正なチェーンに費やされる計算力、分岐が発生する深さからメインのブロックチェーンが切り替わる確率の計算法は、「Satoshi Nakamoto」の論文で紹介されています。

51%攻撃では以下を行うことができます。

- 自身が保有していたことのある通貨を再度使用すること(二重支払い)
- マイニングを独占すること

51%攻撃では**他人の通貨を盗むこと**や、**通貨を無尽蔵に発行すること**などはできません。

この攻撃を行うためは、最低でもネットワーク内の計算力の 4 割程度を占有することが必要となると言われています。ネットワークへ費やされる計算資源の多い Bitcoin や Ethereum では個人でネットワーク内の計算力の大きなシェアを持つことは現実的にはほとんど不可能です。仮に、実際に計算資源を準備して攻撃を行ったとしても、不正に得られる通貨は電力代やコンピュータの準備にかかるコストを下回ると考えられ、経済的に合理性がある行動ではありません。しかし、Bitcoin などに比べて参加者が少ないブロックチェーンでは、大きな計算力を持っているマイナーが参加した時には計算力の寡占化が進み、攻撃が行われる可能性は十分にあります。

7. ブロックチェーンの課題

ブロックチェーンは大きな技術革新ではありますが、利点ばかりではなく実用化に向けた課題も多く存在しています。

7.1 ブロックチェーンの課題

7.1.1 スケーラビリティ問題

スケーラビリティ問題とは、**ブロックチェーンで一定時間に処理することのできるトランザクション数には制限があり、その数が実用的ではない**という問題です。

具体的には、Bitcoin では 1 秒間に 7 個、Ethereum では 15 個のトランザクションしか処理することができません。これに対して、大手のクレジットカード会社は 1 秒間に数千件のトランザクションを処理することができます。

ここからはブロックチェーンのスケーラビリティ問題が発生する原因について説明します。

スケーラビリティ問題の原因

ブロックチェーンでは**ブロックのデータサイズ**と、**一定時間内に作成されるブロック数**があらかじめ仕様で定められています。これにより、一定時間に処理することのできるトランザクション数が決まります。この数はブロックチェーンの仕様を変えない限り増やすことができません。

ネットワークの処理能力を超える大量のトランザクションが発行された場合には、トランザクションはすぐにブロックに取り込まれず、処理の待ち時間が発生します。処理の待ち時間が長くなると、トランザクションの発行者は待ち時間を短くするために、高い手数料を設定するようになり、**トランザクション手数料が高騰するなどの副次的な問題が発生**します。

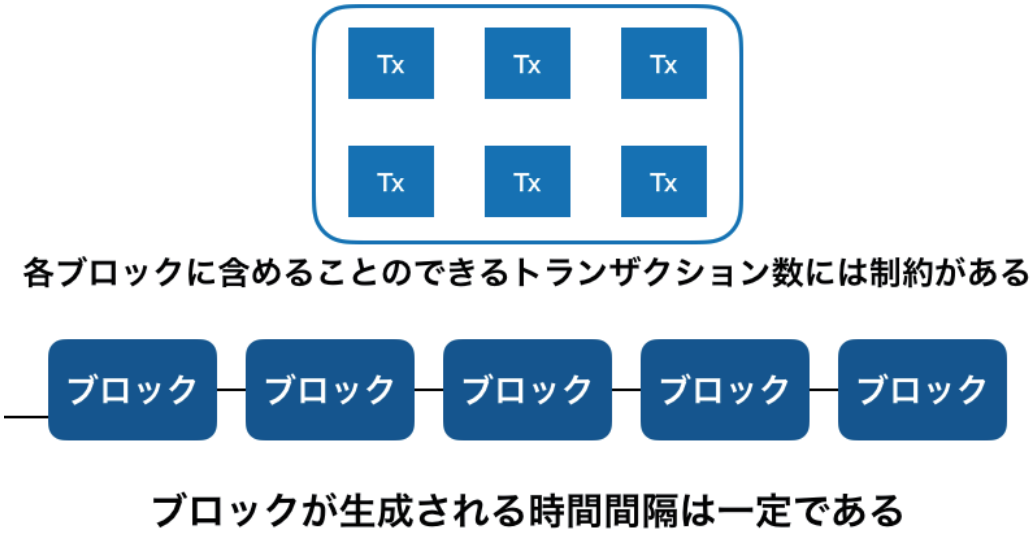


図 7.1 スケーラビリティ問題の原因

Let's TRY

7章演習

ブロックやトランザクションの仕様を変更することで、スケーラビリティ問題を解決することができます。ブロックやトランザクションに対してどのような変更を加えるとスケーラビリティ問題が解決するか図 7.1 を参考にして 4〜5 人のグループで考えてください。

グループで話し合った結果を代表者は発表してください。

回答例

- ブロックの〜〜〜を、〜〜〜に変更するとスケーラビリティ問題は解決する。
- トランザクションの〜〜〜を、〜〜〜に変更するとスケーラビリティ問題は解決する。

7.1.2 マイクロペイメント問題

ブロックチェーンのネットワーク上で処理を行う際には、トランザクション手数料が必要になります。多くのブロックチェーンでは、**手数料は送金額ではなく、トランザクションのデータサイズや処理の複雑さ**を指標としてユーザーが任意に設定します。つまり、送金を行うためのトランザクションのサイズは送金額に依存せず、ほとんど一定であるため、送金額が多いほど多くの手数料が発生し、送金額が少ないほど手数料も少なくていいというわけではありません。このような仕組みのため、**送金額が小さいほど送金額に対して、トランザクション手数料の割合が大きく**なり、ある一定額以下の送金の際にはトランザクション手数料が送金額を上回ることになります。このような理由から、**少額の送金は行われにくい**傾向があります。この少額決済に関連する問題を「マイクロペイメント問題」と言います。

特にブロックチェーンと親和性が高く、これからブロックチェーンとともに発展が期待される IoT 分野では、機械間で大量の少額決済が行われることが考えられ、その際に必要になる高額の手数料が問題になることが予測されています。

7.1.3 処理速度の問題

誰もが参加することのできるパブリックブロックチェーンでは、コンセンサスアルゴリズムに、Proof of Work や Proof of Stake が多く用いられています。これらのコンセンサスアルゴリズムが用いられているブロックチェーンではトランザクションがブロックに取り込まれ、**時間の経過によりブロックの承認数が増えることで、安全に取引を行うことができる**ようになります。時間がかかっても良いので、高い安全性を担保して欲しい際には長時間待ち、少額の決済を素早く行いたい場合には短い時間で取引を完結させることもできます。

つまり、ブロックチェーンにおいて取引の安全性と、即時性はトレードオフの関係になっており、時間というパラメータが非常に重要になっています。

このような性質から、処理に即時性と安全性が同時に高い水準で求められるサービスにはブロックチェーンを用いることはできません。実際に、Bitcoin や Ethereum などのパブリックブロックチェーン上で行われる取引では、取引の即時性

を求めて承認数が少ない状態で取引が完結したと見なすことや、トランザクションがブロックに取り込まれる前に取引を完結させることはあります。しかし、このような取引には、二重支払いが発生する可能性などがあり、取引の安全性には大きな問題があります。

7.2 課題解決のために期待されている技術

現在、ブロックチェーンの課題解決のために様々な技術が研究、開発されています。

7.2.1 Lightning Network(ライトニングネットワーク)

ライトニングネットワークとはBitcoinで用いられる技術であり、Bitcoinの「スケーラビリティ問題」、「マイクロペイメント問題」、「処理速度」の問題の3つを解決することが期待されています。ライトニングネットワークは、ブロックチェーンのネットワークの上に、通貨の取引のための別のネットワークを作り、そこで送金処理を行います。この際に**ブロックチェーンに全ての取引は記録されず、最終的な取引結果のみがブロックチェーンに記録**されます。

従来はAさんがBさんに1BTCの送金を5回行う際に、取引毎に5つのトランザクションを発行し、ブロックチェーンに記録する必要がありました。ライトニングネットワークを用いると、個別の5回の送金はブロックチェーンに記録されず、最終的にAさんがBさんに5BTC送金したことのみがブロックチェーンに記録されるようになります。ライトニングネットワークのように、ブロックチェーンの外で取引を行う技術は「**オフチェーン技術**」と呼ばれます。

ライトニングネットワークの特徴

即時決済

Bitcoinの決済では、仕様上安全に取引を行うためには時間を必要としました。これに対して、ライトニングネットワークは**送金に使用されるネットワークが、Bitcoinのネットワークから独立**しており、またブロックチェーンを用いていません。これにより即時決済を行うことが可能になります。

大きな処理能力

ライトニングネットワークではブロックチェーンを用いておらず、Proof of Workのような取引の確定までに時間のかかるコンセンサスアルゴリズムは必要としませ

ん。そのため**一定時間に処理することのできるトランザクション数はブロックのサイズや、ブロックの生成間隔などよる制限はされません**。そのため、ライトニングネットワークでは大きな処理力が生まれます。

低手数料

ブロックチェーンで送金を行う際には、ブロックを作成するマイナーにトランザクション手数料を支払う必要があります。ライトニングネットワークでも、ブロックチェーンのネットワークと同様にP2Pネットワークが利用されており、送金を行う際に相手のノードまでトランザクションを届ける際には、**経由するノードに対して手数料を支払わなければなりません**。手数料は経由するノード自身が定めることができ、その額や相手のノードまでの距離を考慮してトランザクションの発行者が、トランザクションを伝達する経路を定めることができます。この際にかかる手数料はトランザクション手数料に比べて安くなると考えられています。

7.2.2 Plasma(プラズマ)

プラズマは**Ethereumで用いられるスケーラビリティ問題を解決するための技術**です。これまでは、Ethereumのネットワーク内で発行されたトランザクションは、Ethereumのネットワーク内で処理が行われ、Ethereumのブロックチェーンに記録が行われていました。しかし、プラズマでは**Ethereumのブロックチェーンを親ブロックチェーンとして、プラズマチェーンと呼ばれる階層的な子ブロックチェーンを作り、最終的なブロックの状態のみを親チェーンに記録**します。プラズマのように複数のブロックチェーンを用いる技術は「サイドチェーン技術」と呼ばれます。

以下の図はプラズマのようにブロックチェーンが階層構造を形成するサイドチェーン技術のイメージです。

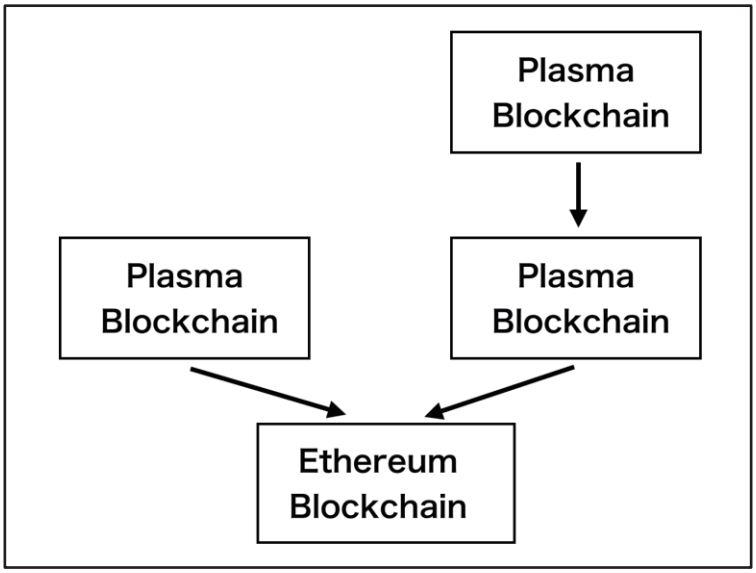


図 7.2 階層的な複数のブロックチェーンを作成する

プラズマの特徴

ブロックチェーンに記録されるトランザクション数の減少

子ブロックチェーンのネットワークの中で発行されたトランザクションは、子ブロックチェーンのみに記録されるため、**全てのトランザクションを Ethereum のブロックチェーンに記録する必要がありません**。子ブロックチェーンに記録された内容の詳細は親ブロックチェーンには記録されませんが、子ブロックチェーンの個々のブロックのハッシュ値は親ブロックチェーンへ記録されます。そのため、子ブロックチェーンで改ざんが発生した際には、親ブロックチェーンでその検知ができます。

このような仕組みにより、Ethereum のブロックチェーンのネットワーク内で処理伝達されるトランザクション数を抑えることができ、スケーラビリティ問題の解決に繋がります。

データサイズの大きなトランザクションもスムーズに実行することができる

Ethereum のネットワーク内では、Bitcoin のような通貨の送金だけでなく、「スマートコントラクト」と呼ばれる複雑なプログラムによる契約を行うためのトランザクションも発行されています。これらのトランザクションは、単純な送金を行うためのトランザクションと比べてデータサイズが大きく、また処理が複雑であるためにトランザクションの検証と伝達には時間がかかります。プラズマでは、トラン

ザクションをEthereumのネットワーク内のコンピュータだけでなく、**プラズマチェーンのネットワークに分割して処理を行う**ことで処理速度を上げることができます。

ブロックチェーンは大きな技術革新ですが、まだまだ多くの課題を抱えているのも事実です。様々な解決法が考えられていますが、最後に紹介したオフチェーン技術とサイドチェーン技術は、研究開発が盛んであり、これらの課題の解決に特に大きな期待が集められています。

8. スマートコントラクトの概要

この章では、ブロックチェーンの利用方法として期待されているスマートコントラクトと DApps について説明します。

8.1 スマートコントラクト

スマートコントラクトには様々な解釈があり、その中に以下のようなものがあります。

- 賢い契約
- 契約の自動化
- プログラム化された契約
- 自動執行権のある契約
- スマートコントラクト・プラットフォーム上で動く契約

上の解釈からわかるようにスマートコントラクトには厳密な定義はありません。そのためここでは、**人が介在しない商取引**のことをスマートコントラクトと呼ぶことにします。

スマートコントラクトの身近な例として、自動販売機、オンライン決済、自動改札などが上げられます。これらの例からわかるようにスマートコントラクトは決して新しい技術ではなく、**私たちが普段から利用している技術**です。つまり、スマートコントラクトはブロックチェーンにより作られた技術ではなく、ブロックチェーンが誕生するより前に生まれた技術です。

このようなスマートコントラクトの利点としては、人間が介在しないことで、**人件費を削減**することができることや、**ヒューマンエラーの排除**することができる点が挙げられます。

8.1.1 スマートコントラクトの仕組み

スマートコントラクトが作成されてから契約が終了するまでの流れは 4 つに分けることができます。

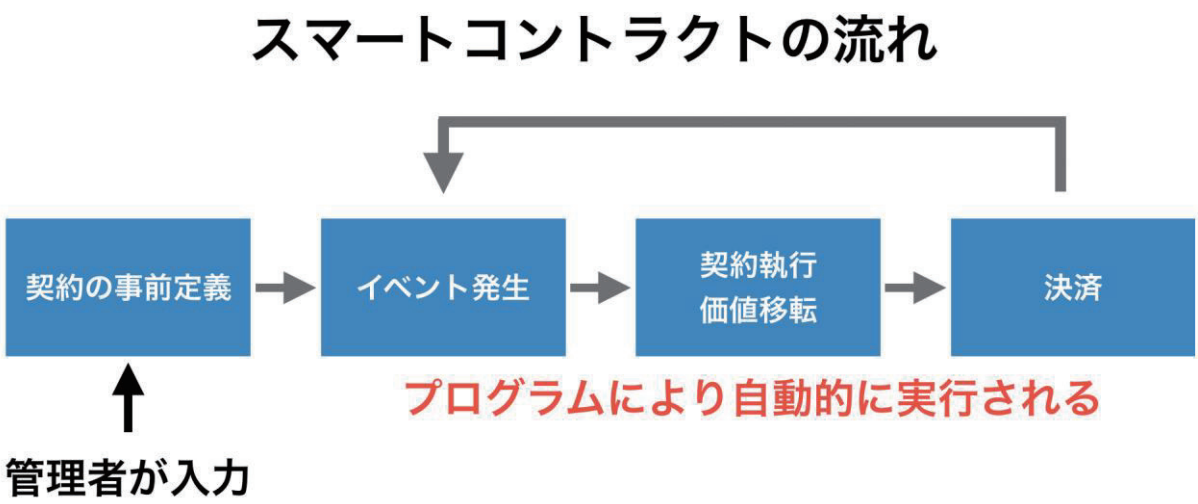


図 8.1 スマートコントラクトの処理の流れ

契約の事前定義

スマートコントラクトにより行われる契約の内容を決定し、それをプログラムに記述します。

イベントの待機

実行可能な状態になったスマートコントラクトは、発動のトリガーとなるイベントを待ちます。

契約実行　価値移転

あらかじめ定められたイベントが発動すると、定められたプログラムに従って契約を行うための処理が実行されます。

決済

契約内容に基づいて資産の移転が行われます。

8.1.2 スマートコントラクトの特徴

現在、スマートコントラクトが大きく注目されている理由には、ブロックチェーン技術の誕生が大きく関係しています。従来のスマートコントラクトは、イベント発生、契約執行・価値移転、決済部分の処理はコンピュータによって行われます。このためスマートコントラクト機能にはこのコンピュータの管理者が必要でした。また、この管理者は契約の仲介者としても機能しています。

このコンピュータを用いて処理を行っていた部分にブロックチェーンを用いることで、**管理者を必要とせずにスマートコントラクトが実行できる**ようになります。この点が現在スマートコントラクトに大きな注目が集まっている理由です。

ここからはブロックチェーンを用いたスマートコントラクトの特徴について説明します。

トラストレス

これまでのスマートコントラクトに仲介者が必要であった理由は、オンライン上で信頼ない二者間だけでは安全に取引を行うことができなかったためであり、**信頼点として企業など第三者が必要**でした。

しかし、ブロックチェーンを用いると、**管理者を必要とせずに、安全に取引を行うことができる**ようになります。これにより、仲介者に依存した取引から脱却することができることや、ユーザーとしては**仲介手数料を安く抑えることができる**などの利点があります。

改ざん耐性

ブロックチェーンを利用したスマートコントラクトでは、**契約を記述したコードや、契約結果がブロックチェーンに記録**されます。ブロックチェーンの改ざん耐性の高さにより、契約を改ざんされることなく安全に保存することができるようになります。

透明性の高さ

ブロックチェーンを利用したスマートコントラクトでは、契約を記述したコードや、契約結果はブロックチェーンに記録されます。利用しているブロックチェーンがパブリックブロックチェーンであった場合には、**契約を記述したコードや、契約結果は誰でも閲覧することができる**ため、透明性の高い取引を行うことができます。

ここで問題になるのが秘匿性です。契約の内容を第三者に見られたくない場合もあると思います。この解決策の１つとして、**暗号化してブロックチェーンに書き込む**という方法があります。これにより、暗号文は公開されているものの、鍵を持っている人しか、平文は読むことができなくなります。しかし、この際に注意しなければならないのが、個人情報などの扱いです。一度、ブロックチェーンに書き込んだことは消すことができません。現在安全であると考えられている暗号技術で個人情報を暗号化してブロックチェーンに乗せたとしても、時間の経過とともに暗号技術が破られた際には個人情報を誰でも見ることができる状態になってしまいます。このような点から**秘匿性の高い取引に関してはパブリックブロックチェーンを使うのではなく、コンソーシアムブロックチェーンやプライベートブロックチェーンなどを使う**ことがよいとされています。

8.2 スマートコントラクトの利用例

スマートコントラクトの利用はあらゆる分野で考えられていますが、特に以下の分野での利用が具体的に考えられています。

8.2.1 証券取引

証券、株式、不動産などの取引には直接的に金銭が関与するため、法律により様々な契約方法が厳格に定められています。そのため、契約をプログラミングコードとして記述することが比較的容易であり、スマートコントラクトとの相性が良いということが特徴です。

Corda

Corda(コルダ)は金融機関向けのスマートコントラクトのプラットフォームとして、債権や株式の発行、取引などを行うことができます。また、いくつかの銀行ではCordaを活用するためのデモが始まっています。

8.2.2 シェアリングエコノミー

シェアリンエコノミーは、スマートコントラクトとの親和性が非常に高く、注目されています。仲介者を必要とせずに、個人間で実現するためには、取引を行う相手がどのような人物であり、信頼することができるのか知る必要があります。このような情報をブロックチェーンを用いて管理することにより、契約を結ぶ前に取引相手が信頼のできる人物であるかを確かめることができます。

Golem

Golem(ゴーレム)は個人のコンピュータリソースを世界中でシェアすることのできるサービスです。登録者のコンピュータがアイドル状態の時に、他の登録者がそのCPUを利用することができます。CPUを使用した対価は仮想通貨により、CPUを貸し出したユーザーに対して直接支払われます。

8.2.3 電力取引

太陽光発電などにより、個人宅で作られた電力を電力会社を介することなく、個人間で取引を行う仕組みがスマートコントラクトにより実現されようとしています。スマートコントラクトを用いることで、これまでの中央集権的な電力システムから脱却し、自律的に電力を受給できるようになります。

関西電力

国内でも関西電力がオーストラリアの企業と共同でブロックチェーンを用いた電力 P2P 取引システムの実証研究を 2018 年春からはじめています。

8.2.4 内容証明

スマートコントラクトの例として、「Proof of Existence」と呼ばれる契約書や所有権の書類の存在証明が挙げられます。データのハッシュ値をブロックチェーンに記録することで、その時刻に、そのデータが存在していたことを永久に証明することができます。これらは特に、政府に対する信頼が薄く、管理を委託することができない発展途上国での活用が期待されています。

Factom

Factom(ファクトム)は記録の存在の証明を行う分散型の公証プラットフォームです。2015 年から始まったサービスであり、Factom 独自のブロックチェーンと Bitcoin のブロックチェーンを併用することで、改ざん耐性を高めています。

Mijin

Mijin(ミジン)は取引説明書や仕様書など、複数のユーザーにより書類の作成を行う場合に、更新の履歴をブロックチェーンで管理することのできるサービスです。

8.3 DApps

8.3.1 DApps とは

DApps とは、Decentralized Application の略であり、日本語では分散自律型アプリケーションと呼ばれます。これに対して、従来のアプリケーションは Centralized Application と呼ばれます。従来のアプリケーションでは、そのアプリケーションを提供している企業が管理者となり、システムやデータの管理を行っています。これに対して、DApps には**特定の管理者は存在せず、システムの維持は DApps が稼働するブロックチェーンのネットワークの参加者により行われ**、アプリケーションの**コードや、アプリケーションで保持するデータはブロックチェーンに記録**されます。

8.3.2 DApps の利点と課題点

DApps はブロックチェーンの応用例であり、その**利点や課題点はそれぞれの性質を引き継ぎます**。

具体的には、スケーラビリティ問題があることから大量の処理を一度に行うことができないため、大量のトランザクションが発行されるようなアプリケーションは作ることができない点や、即時性が求められる処理には向いていないなど DApps を作成する際にはブロックチェーンの特性をよく理解する必要があります。

8.3.3 Ethereum(イーサリアム)

2018 年現在、DApps を作成する上で最も一般的なブロックチェーンが Ethereum です。Ethereum は「Ethereum Foundation」が中心となって開発を行っているブロックチェーンであり、**分散型アプリケーションを作るための基盤**です。

Ethereum 上でプログラムを動かすためには、あらかじめプログラムのまとまりである**コントラクト**をトランザクションにより登録し、そのコントラクトをトランザクションで呼び出すことでプログラムを実行することができます。

分散アプリケーションのためのプラットフォームには Ethereum 以外にも、NEO や EOS といったブロックチェーンがあります。

8.4 DApps の利用例

8.4.1 分散型取引所

一般的な仮想通貨取引所では、ユーザーは通貨を使用するための秘密鍵を取引所に預ける必要があります。そのため**サイバー攻撃などにより取引所から秘密鍵が盗まれ、それにより通貨が流出する危険性**があります。実際に鍵の盗難により通貨が流出する事件は、これまでに何度も発生しています。これに対して、分散型取引所では、秘密鍵を取引所に預けるのではなく、自身で鍵を管理します。

Bancor

Bancor(バンコール)では、トークン同士の交換を仲介するためのスマートトークンと呼ばれるトークンを発行します。このスマートトークンによりあらゆるトークン同士の交換をスムーズに行うことができるようになります。

KyberNetwork

現在、多くのトークンが発行され続けているものの、これらのトークンは特定のアプリケーショ内のみでの使用に限られていることがほとんどです。
KyberNetwork(カイバーネットワーク)では、特定のアプリケーションのみで使用されるトークンを他のトークンと瞬時に交換することのできる仕組みを提供します。

8.4.2 ゲーム

DApps のコードはブロックチェーンに記録されています。これにより、**ゲームの結果が意図的に操作させていないことをユーザーが確認**することができ、ゲームの透明性が高くなります。

CryptoKitties

CryptoKitties(クリプトキティーズ)は子猫の育成ゲームです。子猫を育成し、交配することで新たな子猫を生み出します。このゲームの大きな特徴として、子猫を

Ether を用いて売却することができる点が挙げられ、実際にゲーム内の子猫が 1000 万円ほどで売却された事例もあります。

ETH.TOWN

ETH.TOWN(イーサタウン)は自身が不動産オーナーになることのできるシミュレーションゲームです。その中でキャラクターを育て、それらを売買することができます。

8.4.3 市場予測

株価や、サッカーの試合結果などの未来の出来事を予測する市場です。DApps を用いることにより、**管理者が必要なくなり手数料を削減**することができ、なおかつ**透明性の高い取引**を行うことができます。

Gnosis

Gnosis(グノシス、ノーシス)は市場予測のアプリケーションのためのプラットフォームです。これを用いることで簡単に市場予測に関連するアプリケーションを作成することができます。

8.4.4 身分証明

ブロックチェーンの仕組みを用いて個人情報を管理することで、**第三者に個人情報を開示することなく**、身分の証明ができるようになります。

uPort

uPort(ユーポート)では、個人の身分証明を始め、企業やデバイスなどのあらゆるものの ID を管理することを目標にしています。個人情報がブロックチェーンに記録されていることにより、スマートフォンなどで uPort を利用することで簡単に身分証明を行うことができます。

Let's TRY

巻末演習

現在、私たちが利用しているサービスの大半は管理者が存在する中央集権的な仕組みが利用されています。これは、ブロックチェーンが誕生する以前は、非中央集権的な仕組みを実現することができなかったためです。

ブロックチェーンが誕生した現在において、以下のサービスをあなたが新たに立ち上げる際に、中央集権的な仕組みと、非中央集権な仕組みのどちらを採用しますか。それぞれのサービスの特徴を踏まえて 4〜5 人のグループで考えてください。

- SNS
- グルメサイト
- オンラインフリーマーケット

回答例

- SNS は中央集権的に管理者が存在する仕組みとして運営した方がいい。理由は〜〜〜なためです。
- グルメサイトの〜〜〜点には非中央集権的の特性である、〜〜〜を活かすことができる。
- オンラインサイトの〜〜〜の部分は中央集権的に、〜〜〜な部分は非中央集権的に運営した方がいい。

付録

ブロックチェーンゲームのカード

3 章の演習で利用します。コピーして利用してください。

Genesis Block

Genesis Block

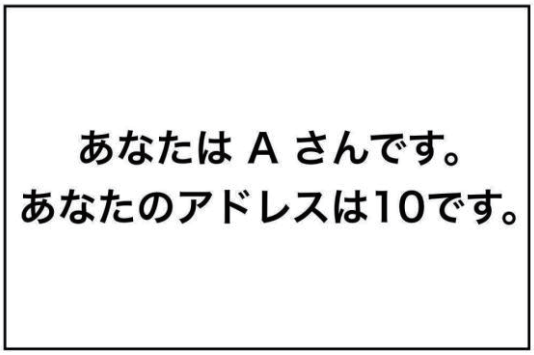
このブロックの合計値	131 ①(②~⑮の和)	2,248,091 ①の3乗
前のブロックの合計値	0	②
ナンス	21	③
送り主	宛先	送金額(BTC)
新規発行通貨	10 ④	100 ⑤
⑥	⑦	⑧
⑨	⑪	⑫
⑬	⑭	⑮
計算スペース		

Block

このブロックの合計値	①(②~⑮の和)	①の3乗
前のブロックの合計値		②
ナンス		③
送り主	宛先	送金額(BTC)
新規発行通貨	④	100 ⑤
⑥	⑦	⑧
⑨	⑪	⑫
⑬	⑭	⑮
計算スペース		

トランザクション

アドレス



あなたは A さんです。
あなたのアドレスは10です。

あなたは B さんです。
あなたのアドレスは15です。

あなたは C さんです。
あなたのアドレスは25です。

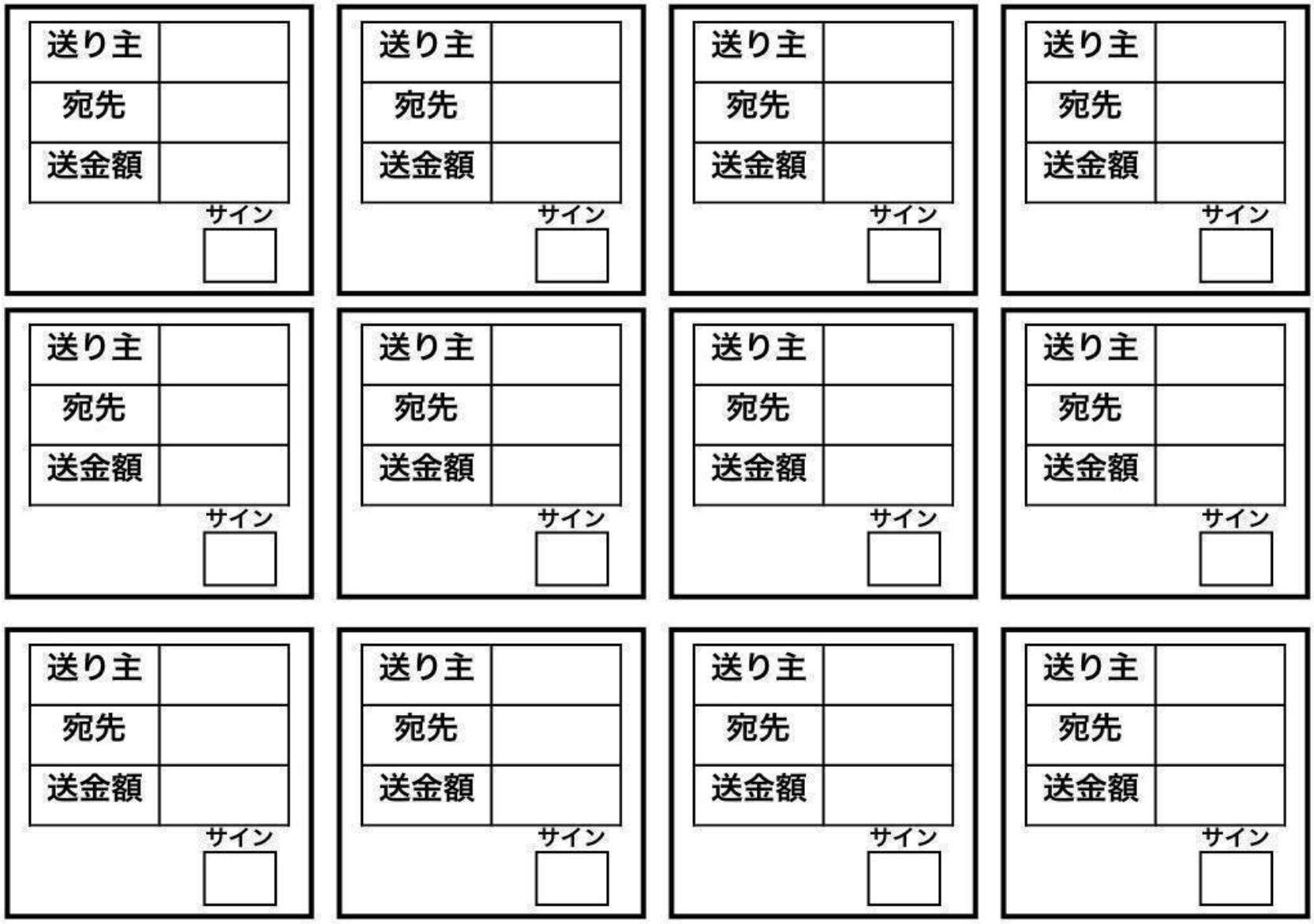
あなたは D さんです。
あなたのアドレスは5です。

あなたは E さんです。
あなたのアドレスは20です。

あなたはFさんです。
あなたのアドレスは30です。

あなたはGさんです。
あなたのアドレスは45です。

あなたはHさんです。
あなたのアドレスは35です。


送り主
宛先
送金額
サイン

このブロックの合計値	①	①の3乗
前のブロックの合計値	②	
ナンス	③	
送り主	**宛先**	**送金額(BTC)**
新規発行通貨	④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭
計算スペース		

このブロックの合計値	①	①の3乗
前のブロックの合計値	②	
ナンス	③	
送り主	**宛先**	**送金額(BTC)**
新規発行通貨	④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭
計算スペース		

このブロックの合計値	①	①の3乗
前のブロックの合計値	②	
ナンス	③	
送り主	**宛先**	**送金額(BTC)**
新規発行通貨	④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭
計算スペース		

このブロックの合計値	①	①の3乗
前のブロックの合計値	②	
ナンス	③	
送り主	**宛先**	**送金額(BTC)**
新規発行通貨	④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭
計算スペース		

Genesis Block

このブロックの合計値	131 ①	2,248,091 ①の3乗
前のブロックの合計値	0 ②	
ナンス	21 ③	

送り主	宛先	送金額(BTC)
新規発行通貨	10 ④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭

計算スペース

Genesis Block

このブロックの合計値	131 ①	2,248,091 ①の3乗
前のブロックの合計値	0 ②	
ナンス	21 ③	

送り主	宛先	送金額(BTC)
新規発行通貨	10 ④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭

計算スペース

Genesis Block

このブロックの合計値	131 ①	2,248,091 ①の3乗
前のブロックの合計値	0 ②	
ナンス	21 ③	

送り主	宛先	送金額(BTC)
新規発行通貨	10 ④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭

計算スペース

Genesis Block

このブロックの合計値	131 ①	2,248,091 ①の3乗
前のブロックの合計値	0 ②	
ナンス	21 ③	

送り主	宛先	送金額(BTC)
新規発行通貨	10 ④	100 ⑤
⑥	⑦	⑧
⑨	⑩	⑪
⑫	⑬	⑭

計算スペース

令和元年度「専修学校による地域産業中核的人材養成事業」
スマートコントラクトを使用したシステム開発人材の育成

ブロックチェーン概論

令和２年２月
学校法人　麻生塾　麻生情報ビジネス専門学校
〒812-0016　福岡県福岡市博多区博多駅南２丁目１２-３２