[http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html#more](http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html)

## 2013年7月14日

### ビットコイン内部の仕組み

注意：図形や数式が多くなります…

[](http://4.bp.blogspot.com/-7JUxMQEVGhw/UklNXRrpezI/AAAAAAAABUw/LxrnAGPfw5U/s1600/Bitcoin.jpg)  
 **はじめに**

本ビデオでは、ビットコインを所有、送金、あるいは「採掘」するとは一体どういうことなのかについてより深く理解していただくために、ビットコイン内部の仕組みを解説します。

お知らせ：[トルコ語版](http://www.bitkoyun.com/viewnews.php?id=11&title=bitcoin-nasil-calisir) （翻訳者：[bitkoyun.com](http://bitkoyun.com/)）ができました。

**全体として、ビットコインは一体何なのでしょう？**

まず、ビットコインの概要を簡単に理解しておきましょう。

ビットコインはその核心において、台帳のように口座を記録して金銭を管理するためのデジタルファイルに過ぎません。このファイルのコピーは、ビットコインネットワーク上に存在するすべてのコンピューターに置かれています。ファイル内の数字は、現実世界では何も意味していません。数字に価値があるのは、皆が自分の口座内の数字と引き換えに実際のモノやサービスを取引し、他の人々もそうするだろうと期待しているからです。数字に力があるのは、それに価値があると私達が信じているからです。[不換紙幣と同じです](http://en.wikipedia.org/wiki/Fiat_money)。

送金する場合は、自身の口座内の金額が減り、受け取り手の口座内の金額が対応して増えることをネットワークに対して配信します。ビットコインネットワーク内のノード（またはコンピューター）はその取引情報を自らが所有する台帳に書き込み、他のノードへも回します。数式に基づいたセキュリティ機能を追加すれば、これで全て出来上がりです。台帳を管理するコンピューター同士で構成されたシステム、となります。

銀行が台帳を管理する方法と似ていますが、台帳が単一の組織ではなく、集団によって管理される点が、重要な違いです。例えば、銀行では自分の取引しか照会できませんが、ビットコインでは全員が全員の取引内容を照会できます。

また、銀行は基本的に信用できるか、少なくとも間違いがあれば訴えることができます。ビットコインでは、赤の他人とやりとりを行うので、誰も信用できません。ビットコインのシステムは非常によくできており、信用が必要ないのです。特殊な数式が、システムの全てを保護しています。

本記事の後半では、ビットコインにおいて赤の他人同士がお互いの財政取引の管理が成立する理由を解説します。

**ビットコインにおける送金方法**

基本的な処理内容として、アリスがお金をボブに送るとき、彼女は口座と金額の情報を入れたメッセージを送信するだけです：

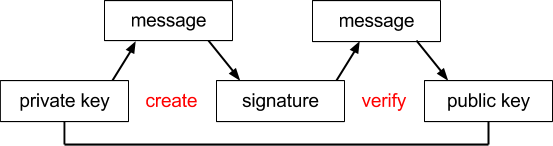
「アリスからボブへ5.0 BTCを送金。」

この情報を受け取ったノードは全て、自らが保有する台帳に取引情報を書き込み、並行して取引メッセージを他のノードへ回します。しかし、取引情報が本物であり、当事者がメッセージを送信した、とノードはどうやって判断するのでしょう？

ビットコイン規定に基づき、資金を開放して消費する場合にはパスワードが要求されます。このパスワードは、「デジタル署名」と呼ばれます。実際に手で書かれた署名と同様に、デジタル署名はメッセージが本物であることを証明します。デジタル領域で複製や偽造を防止できる数式アルゴリズムに従って証明するのが、手書きと異なる点です。

固定されたパスワードと異なり、各取引毎に全く異なるデジタル署名が要求されます。ビットコインでは、赤の他人と取引を行っていることを念頭に置いてください。他の誰かに利用される恐れのあるパスワードは公開したくないでしょう。

デジタル署名は、2種類の異なる（でも関連性のある）鍵を利用します。「秘密鍵」は署名を行い、「公開鍵」は署名を検証するのに用いられます。



秘密鍵はパスワードのようなもので、公開鍵はパスワードを隠したままでパスワードの検証を行う仲介者、と考えられます。

公開鍵はまた、ビットコインにおける「送金先」のアドレスでもあります。つまり、誰かに送金を行う際は、その誰かの公開鍵に送金を行っている、とも言えます。

https://lh6.googleusercontent.com/x85FVrfd81ggjSTXbppsk4OC-4070a90Kdl9G4R78UTEA0FrEf8WzJuESASajt1VOQ2yg8y7AL_HMxYCcYpB_RjQb92zxZ0F3z3Oh8PLJmOQCtERfv_CY7DXHsjdeIjRoA

お金を受け取るには、自分が送金先の公開鍵アドレスの正当な所有者である、と証明しなければなりません。これは、取引メッセージと自分の秘密鍵を組み合わせてデジタル署名を行えば、可能です。

署名 = f(メッセージ、秘密鍵)

ネットワークにおける他のノードはこの署名を別の関数に当てはめ、公開鍵と署名が一致するかどうかを検証できます。

1 =? v(メッセージ、公開鍵、署名)

デジタル署名の演算処理を行うことで、秘密鍵そのものを見ることなく、送金者が秘密鍵を所持していたことを確認できます。

署名はメッセージと組み合わせて行われるのも重要な点です。これにより、署名が各取引で異なるものになり、他の取引で誰かが利用することができなくなります。メッセージに署名が依存するということは、メッセージに対する変更は署名を無効にするため、ネットワーク内でメッセージを回す際に変更を加えられる心配がないということでもあります。

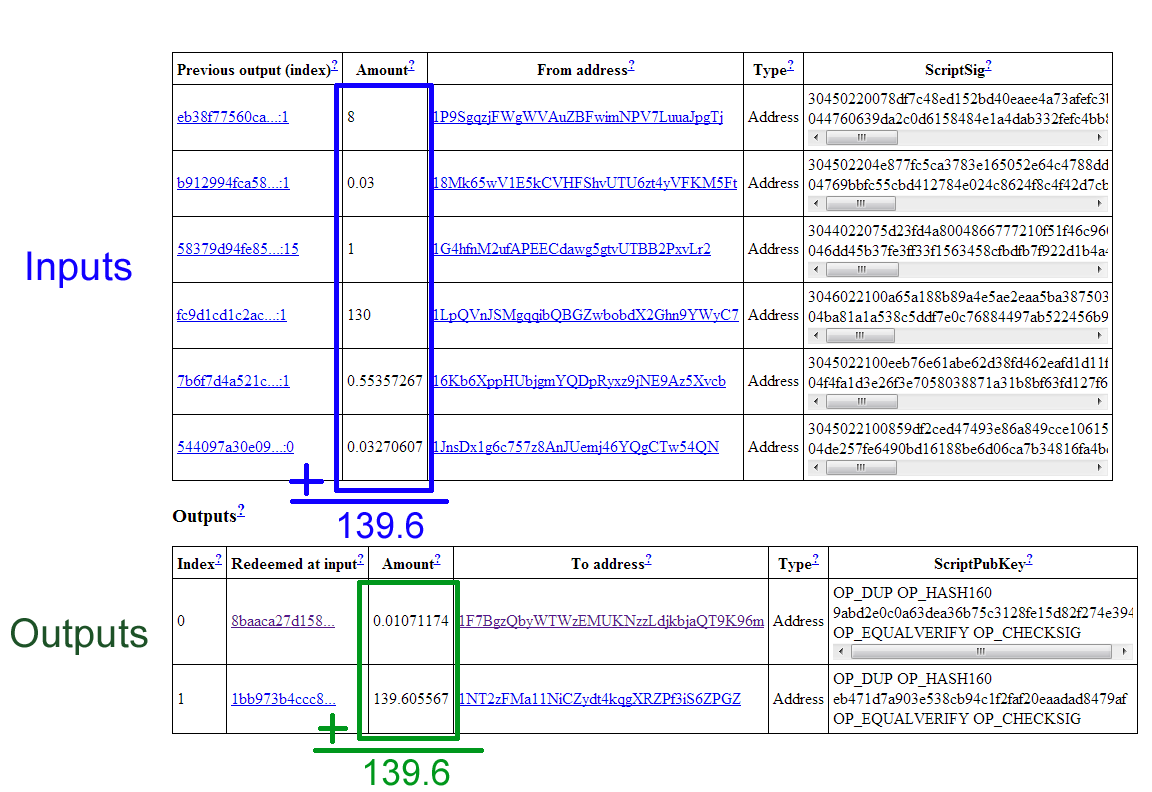
舞台裏で行われている数式は複雑であり、ここでは全てを解説しません。まずは、グーグル検索で得られる情報を列挙しておきます。[ECDSA](http://kakaroto.homelinux.net/2012/01/how-the-ecdsa-algorithm-works/) と数式処理を用いた署名ロックについて。ビデオの最後でも解説します。

**ビットコイン取引と台帳の詳細**

これまで、デジタル取引が取引の認証に使われている点を見てきました。しかし、口座残高の変化を正確に管理する上で、ネットワーク内のノードが果たす役割は、ここまで述べてきたほど簡単なものではありません。実際、口座残高の変遷は全く記録されないのです。誰がどれだけお金を持っているか記録せずに、どうやって送金する原資が存在する、とわかるのでしょうか？

資金を実際に所有しているかどうかの証拠は、残高記録ではなく、以前の取引記録へのリンクから得られます。以下、仕組みを解説します。

アリスがボブへ5.0 BTCを送るためには、彼女は自身の過去取引を参照し、5.0 BTC（あるいはそれ以上）を受け取った取引を指定しなければなりません。こうして指定された取引は、「入力」と呼ばれます。この取引を検証するノードは、こうした入力をチェックすることで、アリスが実際にその金額を過去に受け取っており、入力の合計額が5.0 BTCあるいはそれ以上であることを検証できます。

わかりやすくするために、実際の取引を見てみましょう。

<http://blockexplorer.com/tx/a117c441aa5bd3fcb442e3c47a180c584420bcd9f93c68dab9feddd1d26b767e>

この取引においては、6つの入力が合計139.6を参照しています。出力側において、線が2本あることに注目してください。最初の線は送金者にフィードバックされ、取引終了後の差額が戻されます。規定を簡素化するために、各入力の金額は、取引内で全て消費しなければなりません。つまり、入力の合計にぴったり一致しない（満たない）金額を送金したい場合は、後に差額を自分に戻す必要があります。

こうした入力参照のリンクを通じて、ビットコインの所有権は鎖状に回され、各取引が有効であるかどうかは、それ以前の取引の有効性に依存します。しかし、以前の取引が有効であると、どうしてわかるのでしょうか？わからない、が正解です。従って、以前の取引の入力もまた、検証しなければなりません。実際、ビットコイン財布ソフトウェアをインストールすると、まずそれまでの取引記録が全てダウンロードされ、最初の取引に至るまで全ての有効性が検証されます。赤の他人とやりとりするのだ、という点を思い出してください。ですので、自ら取引を全て検証するのは重要な点です。この作業には24時間以上かかることもありますが、1回きりで済みます。

ある取引が行われると、その取引は消費されたと見なされ、もう一度取引を行うことはできません。さもなければ、誰かがそれを他の取引で参照し、二重使用に使う恐れがあります。つまり、取引を検証する場合は、ノードは様々な確認内容に加えて、取引の入力がこれまでに消費されていないかどうかも確認しなければなりません。正確に言うと、ノードこれまで行われた取引記録を全てチェックし、今回の取引に用いられている入力がこれまでに一度も使われたことがないかどうか、確認します。これまでの取引回数は全体で2000万以上あるため、時間がかかる作業のように思えます。しかし、未使用の取引一覧をチェックすることで時間を短縮することはできます。

つまり、ビットコインのノードは台帳の残高ではなく、取引一覧の記録を管理していることになります。ビットコインを所有するということは、この一覧内に自分の名前が記載された取引があり、それはまだ使用されていない（つまり、他の取引に対する入力としてまだ利用されたことがない）ことを意味します。

所有権がこのように管理されているため、興味深い副作用として、自分の残高を調べようとすると、過去に発生した全ての取引を参照して、まだ使用されていない自分の入力を全て足し合わせることになります。

取引についてもう一つ興味深いのは、システムの設計上、誰か個人に送金するだけの単純な作業よりもはるかに複雑な取引が行える点です。前述の出力段において、意味不明な文字列を発見されたかと思います。

OP\_DUP OP\_HASH160 9abd2e0c0a63dea36b75c3128fe15d82f274e394 OP\_EQUALVERIFY OP\_CHECKSIG

<http://blockexplorer.com/rawtx/a117c441aa5bd3fcb442e3c47a180c584420bcd9f93c68dab9feddd1d26b767e>

つまり、出力は「送金先アドレス」というより、謎解きの課題のように見えます。ビットコインでの送金行為は、メール送信というよりは、公共のロッカーにお金を入れ、数学上のパズルを解くことで鍵を空けられるようにする、に近いものです。パズルは特殊なスクリプト言語で構成されており、通常は公開鍵の所有者一人だけが謎を解けるようにしていますが、もっと複雑な空け方を設定することもできます。例えば、3人のうち2人の署名が必要、とすることで、エスクロー取引が可能になります。他の例で言えば、ビットコイン取引が初めて行われたとき、誰にでも解けるパズルとして発行されました。

ほとんどの財布ソフトウェアでは、このスクリプト層は隠されています。ご自身でソフトウェアを作成し、条件を設定することもできますが、リスクが大きい行為です。アドレス設定が間違っていたため、[2600 BTC 以上がたった一度の取引で失われました](https://bitcointalk.org/index.php?topic=50206.0)。

これはビットコインにおいて重要な点を示唆しています。いざとなったら助けを求めるための銀行やクレジットカード会社が存在しないため、どんな「ユーザーによる過失」であれ、ビットコインは永久に、口座内だけでなく、ビットコイン経済全体からも失われます。秘密鍵を失えば、対応する公開鍵に結び付けられた資金は恒久的に失われます。ハードディスクのクラッシュや、バックアップ不足などから秘密鍵を失う人もいるため、ビットコイン通貨はいずれ[デフレ](http://en.wikipedia.org/wiki/Deflation)化するでしょう。

**匿名性**

ビットコインを確保するための最後のポイント（「採掘」）を解説する前に、ビットコインの匿名性について少し述べてみたいと思います。

IPアドレスを隠蔽できる[TOR](http://h/)ネットワークを通じてビットコインにアクセスすると、公開鍵以外は全て隠したまま、ビットコインを使えます。誰かに自分の取引記録を全て見られたくない場合は（過去の取引記録は全てのコンピューターに収められています）、各入金取引に対して新しい公開鍵を毎回作成することもできます。

ただし、公開鍵同士が不注意にも結び付けられてしまうこともあります。前述の取引において、6つの「入力」取引がソースとして用いられています。それらは全て異なるアドレスへ送られたにも関わらず、その取引のおかげで全部、関連付けられてしまいます。送金者は、全ての入力をアンロックできるデジタル署名を提供するため、自分が全てのアドレスの所有者だと、暴露してしまいました。実際、研究者がビットコインユーザーの行動を調べる場合、こうした結びつきを利用します。Dorit RonとAdi ShamirSeeによる[ビットコイン取引全グラフの量的分析](http://eprint.iacr.org/2012/584.pdf)を参照してください。

公開鍵を用いて「受け取りアドレス」を作成すると、自分の身元を追跡されるのではないか、と心配する人もいます。しかし、この部分ですら匿名で行え、かつ驚くべきことに、ネットワークに接続していなくても行えます。財布ソフトウェアのボタンをクリックするだけで、新しい秘密鍵と公開鍵の組合せがランダムに生成されます。可能なアドレスの組合せは無数にあるため、誰かがその鍵を既に所有しているかどうかをチェックする必要すらありません（メールアドレスを申請する場合には、たいてい誰かに取られているのと対照的です）。しかし、実際に誰かの鍵と一致すれば、彼または彼女の資金にアクセスできることになります。

可能なビットコインアドレスの組合せはこれだけあります：

1461501637330902918203684832716283019655932542976 (1.46 x 10^48 または 2^160)

この巨大な組合せの数は、ビットコインシステムを何重にも保護しています。実際、どれだけ大きいのかを見てみましょう。世界中にある砂浜の砂粒の総数は、7.5 x 10^18th、または 7,500,000,000,000,000,000と考えられています。砂粒一つ一つが、地球全体の砂粒の総数を代表しているとしましょう。それですら、ビットコインアドレスの可能数よりもずっと小さいのです。

(正確に言えば、ユーザーが増えるにつれて、アドレスが一致する可能性は急速に増えていきます。これは [誕生日のパラドックス](http://en.wikipedia.org/wiki/Birthday_problem)と呼ばれていますが、それでも2.9 x 10^39、あるいは10億アドレスの範囲内に収まります)。

http://www.hawaii.edu/suremath/jsand.html

**ビットコインの二重使用**

ビットコインのセキュリティについてまとめてみましょう。デジタル署名を検証すれば、取引メッセージを作成し得た人間はたった一人であることがわかります。その送金者が実際に資金を所有しているかどうかは、取引の入力をチェックして、まだ使用されていないかどうかを見ればわかります。それでもなお、システムには「未使用の小切手」の信頼性を損なう大きなセキュリティ上の欠陥があります。それは、取引の**順番**に関わってきます。

取引はネットワークのノード間を通過するため、受け手が取引の作成日時順に受け取るかどうかは、保証されていません。取引作成日時は簡単に偽造できるため、作成日時の記録に頼ることもできません。（ペイパルのような中央集権的システムとは異なり、取引の順番を記録する中央コンピューターは存在しません。）

それゆえ、取引の前後関係を確認する手法は存在せず、これが詐欺の原因となります。アリスに悪意があるとすれば、ボブと取引を行って送金を行い、ボブが商品を出荷した時点で、同じ「入力」を利用して取引を彼女自身へ向けて送信できます。取引が配信される時間にばらつきがあるため、ネットワークのノードの中には、ボブへ送られる取引よりも先に、2回目の「二重使用」取引を受け取る所も出てきます。ボブが取引を受け取ると、一部のノードは、それは同じ入力を再利用しようとしているために無効だ、と見なします。ボブは出荷済みの商品と、代金の両方を失うわけです。ボブとアリスのどちらがお金を受け取っているかについては、取引が先に作成された証拠がないために、ネットワーク上で合意が得られなくなります。

この点に関しては、ネットワーク上で取引の作成日時について合意できる方法がないために、分散型システムにとっての大きな課題となっています。ビットコインは、一種の数学的競争を用いて、巧妙に作成順序を見つけ出し、間違いが起こらないようにしています。

**ブロックチェーン：取引順序の決定**

ビットコインシステムは取引をグループにまとめ、ブロックという単位の中におきます。このブロックを連結して、**ブロックチェーン**と呼ばれるものを形成します。これは、先ほど述べた取引チェーンとは別の話ですので注意してください。ブロックチェーンは取引を順序付け、取引チェーンは所有権がどう変遷するかを追跡します。

各ブロックは、直前のブロックへの参照ポイントを持っています。ここにおいて、ブロックの作成順序が決定されます。参照をたどることで、初めて行われた取引にいたるまで、全てたどることができます。同じブロック内にまとめられた取引は同時に行ったと見なされ、ブロックに収まっていない取引は「未確認」または未注文と見なされます。

どんなノードであれ、未確認の取引をブロックに集めてネットワークに配信し、チェーンに次回追加されるブロックを提案できます。複数の人間が同時にブロックを作成する可能性もあるため、複数の中から選ばなければなりません。ネットワークはどうやって選択を行っているのでしょうか？ブロックが到着する順番は信頼できません。前述したとおり、ネットワーク上の場所によって順番が異なる場合があるからです。

ビットコインは、ブロックが有効と見なされるためには、特殊な数式を解かなければならない、という解決法を見出しました。各コンピューターはブロック内の記述を全て行った後、**暗号化ハッシュ演算**とよばれるランダムな推定演算を行い、回答を指定された基準以内に収めなければなりません。

ハッシュ関数は任意の長さのテキストから、短い固定長の要約版を作成します。ビットコインの場合、長さは32バイトです。以下は、ビットコインが指定するハッシュ関数、SHA256の実行結果例です：

SHA256(「short sentence」)

0x 0acdf28f4e8b00b399d89ca51f07fef34708e729ae15e85429c5b0f403295cc9

SHA256(「The quick brown fox jumps over the lazy **dog**」)  
0x d7a8fbb307d7809469ca9abcb0082e4f8d5651e46d3cdb762d02d0bf37c9e592  
SHA256(「The quick brown fox jumps over the lazy **dog.**」) **(ピリオド追加)**  
0x ef537f25c895bfa782526529a9b63d97aa631564d5d789c2b765448c8635fb6c

第三の例において、文末にピリオドを追加しただけで結果ががらりと変わることに注目してください。出力は絶対に予測できません。つまり、特定の値を出力したければ、当てずっぽうを繰り返すしかありません。鍵の暗証番号を推定する作業に似ています。最初の推定が当たればラッキーですが、実際は何回も試行錯誤が続きます。実際、通常のコンピューターであれば、このブロック問題を解くには何年も推定を繰り返さなければなりません。

ビットコインネットワーク全体で全てのコンピューターが数字を推定したとすると、誰かが解決法を見出すまで平均10分かかります。

数式を解いた最初の人物が自分のブロックを配信し、チェーンに新たに追加される取引の一群をまとめたブロックとして認められます。数式の処理にランダム性が加わることで、2人の人間が同時に回答をえる可能性はほぼなくなります。時折、同時に複数のブロックが回答を得る場合があり、**分岐**が生まれる原因となります。

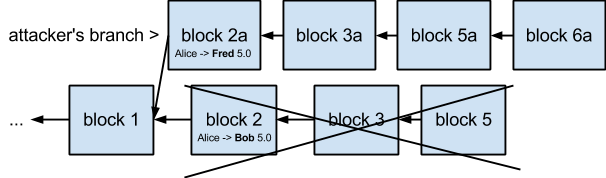
この場合、最初に自分が受け取ったブロックを採用します。他の人々は異なるブロックを先に受け取っている可能性があり、彼らは別のブロックを採用します。

均衡が破られるのは、他の誰かが新たにブロックを確認できた場合です。基本ルールは、より長い分岐を常に選択する、というものです。ブロックが同時に確認される事態は、数式のおかげでほとんど防がれています。その上、分岐ルールによって、同時確認が連続して起こる事態が防がれます。結果として、ブロックチェーンは急速に安定化します。つまり、ブロックチェーンの先端2-3箇所のみの範囲で、全員がブロックの順序について合意にいたることができます。

**ブロックチェーンにおける二重使用の問題**

チェーンの先端において不明瞭さがありうるという事実は、取引のセキュリティに関して重要な事項を示唆しています。例えば、自分の取引が短い分岐に入っていると、ブロックチェーン内から消されることになります。通常は、取引は未確認のグループに戻され、後にブロックに組み込まれます。しかし、この取引が消される可能性は、システムを安定化させるために防いできた二重使用を復活させてしまうのです。

前述の環境にて、二重使用を用いた攻撃の仕組みについて見てみましょう。アリス（詐欺師）が、ボブに送金します。ボブはブロックチェーン内で取引が「確認」されるのを待ち、商品を出荷します。



（ボブが出荷を行うタイミングを覚えておいてください）

さて、ノードが常に長いブロックチェーン分岐を選択するおかげで、ボブが確認を行った分岐よりも長い分岐をアリスが作成することができれば、ボブが受け取ったお金は実質的に消失します。ボブが受け取るべきお金は、未確認取引のグループに一旦戻されます。しかし、アリスは既にそのお金を別の取引の入力として使ってしまっており、ノードはボブが受け取るお金は無効だと判断します（既に使われてしまっているからです）。

**二重使用の防止**

さて、商品注文を行うシステムのレベルで、アリスの詐欺行為を防止する方法はあるのでしょうか？アリスはブロックチェーンをあらかじめ計算しておき、タイミングを見計らってネットワークに挿入することができるように見えます。しかし、各ブロック内で行う演算処理は、こうしたトリックを防止できるように作られています。前述した暗号化ハッシュ演算をもう少し踏み込んで見てみれば、防止の仕組みが理解できます。

ブロック内の演算を行うということは、ブロックの暗号化ハッシュ演算の結果が一定の値より小さくなるまで、ブロックの終端でランダムに異なる値を試すということです。解答が見つかれば、ハッシュ演算出力は、ブロックを一意で認定可能な指紋のように機能します。ブロック内でたった一文字でも変更されれば、前述の例で示したとおり、ブロックのハッシュ演算出力は全く変わったものになってしまいます。

実際に、ハッシュ演算出力（または指紋）を用いて、「前回のブロック」が参照されます。その結果、ブロックチェーンの中間でブロックを交換することが不可能になります。入れ替えたブロックのハッシュ値が以前と異なるため、次のブロックから参照できなくなってしまうからです。細かい点ですが、それ以上に重要なのは、直前のブロックのハッシュ演算が正しく行われないと、次のブロックもまた、ハッシュ演算が行えないという点です。直前のブロックへは、ハッシュ関数内でも利用されるテキスト列を用いて参照されます。したがって、参照が変更されれば、演算をやり直す必要があります。

アリスの話に戻ると、以上が、ブロックチェーン分岐をあらかじめ彼女が計算しておくことが不可能な利用です。彼女は、入れ替えたいブロックの直前ブロックがハッシュ演算を完了させないと、ブロックチェーン分岐を作成できません。つまり、彼女はボブが商品を出荷するまでの短い時間の間に、ネットワーク全体を相手にして、それより長い分岐を作成しなければなりません。もしボブが彼女の二重使用ブロックを見つければ出荷を停止しますから、彼女は隠れて作業を行う必要があります。

さて、もしアリスが超高速なコンピューターや大量のコンピューターを所有していれば、ネットワーク全体よりも素早く分岐を作成できるのでしょうか？しかし、たとえ彼女が何千台ものコンピューターを所持していても、無理でしょう。彼女が競争している相手は1台のコンピューターではなく、ネットワーク全体だからです。宝くじを引き当てる努力に似ています。彼女は何千台ものコンピューターを使用して何千枚もの宝くじを購入できますが、それでも、他の誰かが当たりくじを引き当てる場合がほとんどでしょう。彼女がネットワーク全体のコンピューター処理能力のうち、過半数を掌握すれば、他の誰かよりも早くブロックのハッシュ演算を終える確率は50%を超えます。詐欺を何度も連続して行おうと思えば、掌握しなければならない量はずっと多くなります。

つまり、ブロックチェーン内の取引は数式に基づいた競争にさらされています。攻撃を試みる者は、ネットワーク全体を敵に回すのです。ブロックがお互いに連結することで、ブロックチェーンの初期に近づけば近づくほど、取引は確固たるものになります。攻撃者は二重使用を行ってブロックを入れ替えるまでのかなり長い時間、ネットワークに先んじる必要があります。従って、システム全体で二重使用の危険にさらされる可能性はブロックチェーンの終端付近に限られています。だからこそ、数ブロック分待ってから、お金を確かに受け取ったと見なすことが推奨されているのです。

ビットコインシステムにおける最後の疑問に答える前に、ブロックチェーンについてもう一言だけ言っておきます。驚くべきことに、これまで語った仕組みには、信用が一切関与していません。ビットコインネットワークにて赤の他人から情報を受け取った際でも、ブロックのハッシュ演算結果が正しいかどうか、自分で確かめることができます。数式処理があまりにも難しいため、ハッシュ演算の解答を攻撃者自身が故意に得る方法など存在しない、と理解できます。解答は、ネットワーク全体の演算能力が用いられた得られた証拠となります。

**採掘とプール**

お金がデジタル署名と取引チェーンを通じて送金され、取引の成立順序がブロックチェーン内で守られている方法について見てみました。ここからは、最後の疑問：ビットコインがどうやって生成されるか、について見ていきましょう。送金するためには、原資を受け取った取引へ参照を行います。しかし、最初はどうやってブロックチェーンに組み込まれるのでしょうか？

ビットコインをゆっくりと、しかしランダムに生成して配布するために、ブロックのハッシュ演算を完了させた人物には「報酬」が与えられます。ブロックのハッシュ演算が採掘と呼ばれる所以です。演算の真の目的は取引を検証してブロックチェーンを守ることではありますが。4年ごとに、ブロック検証の報酬は半減されるため、いずれはコインの発行は停止します。合計で、2100万ビットコインが生成されます。送金単位は1億分の1ビットコイン(.00000001)まで分割できるため、発行上限がビットコインの利用を制限することはありません。

ブロック検証の報酬が止まると、採掘者が取引を行う報酬はどこから出るのでしょうか？ブロック検証の報酬のほかに、取引にオプションで手数料を組み込むことで、採掘者に対して取引処理の報酬を支払うことができます。現時点では、採掘者の目的はブロック検証の報酬であるため、彼らは取引手数料抜きでブロックのハッシュ演算を行っています。将来においては、付加された手数料に応じて取引が行われ、手数料の付加していない取引は無視されるでしょう。つまり、ビットコインの送金は有料化されます。ただ、それでも現在のクレジットカード手数料より安く済むでしょう。

**採掘プール**

前述したように、通常のコンピューターがブロックのハッシュ演算を行おうとすると、数年かかります。ネットワーク全体がハッシュ演算を行うのは、毎回平均10分程度で済みます。個人がそれに対抗できる可能性は、ほぼ皆無でしょう。報酬を得る可能性を増やすために、多くの人は採掘プールと呼ばれるグループに参加し、共同でブロックのハッシュ演算を行うことで、貢献度に応じて利益を得ています。これは、同僚同士で宝くじを大量に購入する行為に似ています。異なる点は、こうしたプールの中には巨大なものもあり、ネットワーク全体の処理量の20%以上をも占めます。

巨大プールが出現し始めたことで、セキュリティ面で深刻な懸念が出ています。前述の通り、個人の攻撃者がネットワーク全体よりも素早く、数個分のブロックを素早く処理することは、ほぼあり得ません。しかし、可能ではあります。攻撃者の能力がネットワーク全体に占める割合が増えるにつれ、可能性は増していきます。実際、採掘プールの一つであるBTC Guildは、自分達だけで[6ブロックを連続して検証しており](http://www.reddit.com/r/Bitcoin/comments/1cy4a4/this_could_have_been_a_successful_51_attack_from/)、[自発的にメンバー数を限定することで](http://mineforeman.com/2013/04/06/btc-guild-voluntarily-limits-their-hash-rate/)ビットコインネットワーク全体からの不信を払拭しようとしています。（パイチャート表示）

コンピューターの処理能力が大量にあっても、ブロックチェーンの初期に向かえば向かうほど、攻撃者が変更を試みるのは難しくなります。取引が生成されてから商品が出荷されるまでの間、ネットワーク全体よりも素早く処理を行わなければならないからです。

現在推奨されているのは、取引を完了と見なす前に、最低一つのブロック（または確認作業）を待つということです。大規模な取引であれば、最低6ブロック待つようにと推奨されています。BTC Guildが6ブロック分先んじて処理することができた今、もっと長く待つのが賢明でしょう。

**確認時間**

設計上、各ブロックの確認処理は10分かかります。6ブロック待つということは、約1時間かかります。クレジットカード取引が数秒で終わることを考えると、この長い確認時間は重たく見えます。しかし、クレジットカードのユーザーは取引終了後数ヶ月の間、支払キャンセル（支払拒否）を行えることを考慮すれば、業者の観点からするとビットコインはずっと素早く処理を行っています。

なぜブロックごとに10分かかるのでしょう？10分という時間は任意に設定されたものです。しかし、時間を短くするとシステムが不安定になり、長くすると確認時間をとられすぎです。ネットワーク上のコンピューターが増え、採掘専用のハードウェアが開発されている状況から、ブロックの確認時間はもっと短縮されるでしょう。これを補償するため、2週間に一度、全てのビットコインソフトは自己校正を行って、数式処理の難易度を上げることで10分かかるように調整します。対照的に、似たようなデジタル通貨である[Litecoin](http://en.wikipedia.org/wiki/Litecoin)はブロック処理の時間を2.5分で運営しています。\*

\*M. Rosenfeldが著した Aハッシュレートをベースにした二重使用の分析 によると、セキュリティの強度はブロック数に依存しており、各ブロックの処理にかかる時間ではない、となります。しかし、この仮説は、攻撃者にとって時間が関係ない場合に限られます（例えば、ネットワークの処理能力を数分間だろうが数日間だろうが凌駕できる場合）

\*各ブロックにとられる時間と、システム全体の効率性との関連について、サトシがフォーラムで述べている内容にも注目してください。<https://bitcointalk.org/index.php?topic=130222.60>

**ビットコインの仕組みのまとめ**

要約すると、ビットコインはピアツーピアネットワークにより運営され、数式演算で保護されたデジタル通貨、となります。デジタル署名が個々の取引を認証し、所有権は取引チェーンを通じて移管されます。こうした取引の順序は、ブロックチェーンの構造を通じて守られています。各ブロックについて複雑な演算をこなさなければならないので、攻撃を試みる者はネットワーク全体に対して演算処理の競争を挑まねばならず、勝ち目はありません。

ビットコインは他にも興味深いアイデアを示唆しています。例えば、政府による干渉の防止、匿名性、手数料の削減などです。同時に、多くの課題も残されています。現時点では、ビットコインを他の通貨に交換するのは困難です。また、犯罪行為や脱税の温床とも見なされており、政府が禁止に動いています。また、ブロックチェーンを保護する演算処理には、多大な電力が消費されます。

## 付録：デジタル署名と暗号化ハッシュ演算の説明

（完全な解説というよりは、入門編のようなものです）

### ブロックチェーン演算：暗号化ハッシュ

（さらに続きます！）

これまで見てきたように、ブロックを処理するのは宝くじを引き当てるようなものであり、くじは演算処理のサイクル数をもって購入されています。宝くじにはビットコインネットワーク内の全てのコンピューターが参加するため、地球上で最速のコンピューターをもってしても、連続して当たりくじを引き当てる可能性はほぼ皆無です。結局は、これが二重使用攻撃を防いでいます。

### デジタル署名の演算

前述の通り、デジタル署名を行うということは、秘密鍵を所有しており、秘密鍵を使用してあるメッセージに署名できるということです。しかも、秘密鍵の内容は秘密にしたままです。以下に、真のデジタル署名アルゴリズムの構造（不安定なものではありますが）を紹介します。[RSA](http://en.wikipedia.org/wiki/RSA_(algorithm)).

次の変数を用意します：

p = 公開鍵

q = 秘密鍵

p\*q = N （公開されています）

m = メッセージ

s = q\*m の公式を用いて、秘密鍵とメッセージを用いてデジタル署名（s）を作成します。すると、第三者は s\*p = m\*N の公式に当てはめることで、署名が正式なものかどうかを検証できます。なぜなら、

m\*q\*p = m\*N、 N = q\*p だからです。

この方法で、第三者は、全て公開済みの情報である公開鍵、N、メッセージ、署名のみを入手するだけで済みます。この方法の唯一の欠点は、除算ができれば q = N / p を用いて秘密鍵を割り出すことが可能な点です。本当の演算処理は抽象代数学にてモジュラー演算を用いて行われており、そこでは一般人は巨大な数字を扱うことは不可能である、という要素も利用されています。 N = q\*p において、 q と p が巨大な素数だとしましょう（300桁以上）。Nしか判明していない場合は、q と p を探すには途方もない量の推測と検証作業が必要です。しかし、事前に q が判明していれば、p を割り出すのは簡単です。この因数分解[の罠](http://en.wikipedia.org/wiki/Trapdoor_function)が、RSAアルゴリズムの要です。ビットコインで用いられている公開鍵/秘密鍵のアルゴリズムは[楕円曲線DSA（ECDSA）](http://en.wikipedia.org/wiki/Elliptic_Curve_DSA)と呼ばれており、離散対数を見つける困難さを利用しています。ECDSAは、同じセキュリティ能力でもキーサイズがずっと小さいという点で、RSAに比べて優れています。

著者： [Scott Driscoll](https://plus.google.com/113939521475745155585) [8:14 PM](http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html)