



透明性と公平性を実現する ブロックチェーン技術

佐古和恵 (NEC セキュリティ研究所)

ビットコインとブロックチェーン

ビットコインは、2008年に Satoshi Nakamoto を著者とする論文で提案された分散管理型の暗号通貨である。その中で使われる取引履歴を塊(ブロック)にしてチェーン状につなげる履歴管理の手法が、「ブロックチェーン」と呼ばれ着目されている。この手法により、だれか1人を信頼しなくても、正しいとされる履歴を複数のプレーヤで管理できるため、これを改良して別の暗号通貨を作ったり、資産管理や契約管理に活用する方式が提案されたりしている。ビットコインのシステムは、ブロックをチェーン状につなげる手法以外にも、取引の表現方法、矛盾の解消方法や、ブロックに暗号通貨としての価値を持たせる方法など、さまざまな発想を盛り込んで構成されている。したがって、どこを切り出すかによって、「ブロックチェーン」の見方が変わってくる。「ブロックチェーン」の明確な定義がないのは、ビットコインの仕組みに触発されたたくさんの技術者がさまざまな観点で改良案を出し、それらを「ブロックチェーン」と総称しているからである。その中には、必ずしも「ブロック」が「チェーン」状になっていないものもある。本稿では、筆者なりの観点で切り出したブロックチェーンのモデルを紹介する。

ブロックチェーンの概要

◆ モデル

ブロックチェーンは一言でいうと、分散型の履歴蓄積システムである。事前に定められたルールにそぐう履歴データのみ、蓄積することを目的としてい

る。履歴を蓄積したものを台帳と呼ぶ。ここで登場するプレーヤは、データを発生し、台帳に登録依頼を出す利用者と、ルールに従った履歴のみ台帳に登録する(複数の分散された)台帳管理者と、台帳を参照する参照者の3種類が存在する。履歴の蓄積を目的にしているので、基本的に追記のみを想定する(図-1)。

◆ 目指す姿

台帳の管理者が信頼できれば、これは従来のデータ管理システムと大きな変わりはない。しかし、信頼できない台帳の管理者であれば、ルールに従っていないデータを勝手に履歴として台帳に記載したり、一度記載した履歴データの内容を後日改ざんしたり、履歴データの順番を替えたり、あるいは登録依頼があったのに無視して登録しないという懸念がある。管理者が信頼できない場合、台帳の登録状況や保持状況に透明性がないことはさまざまな問題を発生させる。

ブロックチェーンでは、複数の台帳管理者を想定し、それぞれの台帳管理者がルールに従った履歴のみを正しく、一貫性を持って蓄積することを目的とする。すなわち、多くの管理者が結託しない限り

要件1) 登録依頼があったデータは、ルールに従っていれば、台帳に登録される

要件2) 登録依頼があったデータは、改ざんされることなく、台帳に登録される

要件3) 台帳に登録されたデータは、後日、改ざんされたり、消去されたり、順番を入れ替えられたりしない

ことを目的とする。1人の管理者を盲目的に信頼す

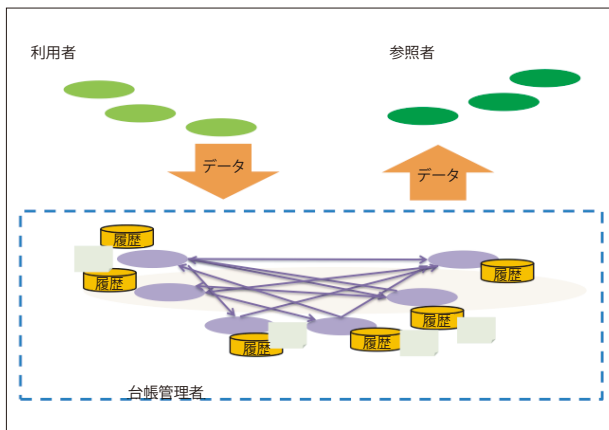


図-1 ブロックチェーンのプレーヤ

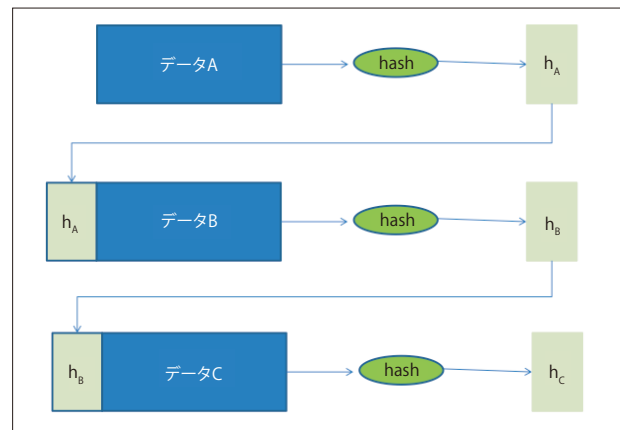


図-2 ハッシュチェーン

ることなく、複数の管理者が相互に確認し合うことにより、透明性と公平性のある履歴を蓄積するのである。

◆ ハッシュチェーン

登録順序の保証

前節に記載の要件2)は、登録依頼者のデジタル署名付きでデータを登録すれば台帳管理者であっても履歴データの改ざんはできない。しかし、1人の管理者が、1つ1つのデジタル署名付きの履歴データの内容は書き換えられなくても、これらの順番を後日入れ替えることはできる。本節では、データの順番入替を防止するハッシュチェーン技術を紹介する。この技術はビットコインの論文が発表される15年以上前に、Stuart HaberとW. Scott Stornettaにより提案されたもの¹⁾であり、一方向性ハッシュ関数を構成要素とする。

一方向性ハッシュ関数

暗号学的に安全な一方向性ハッシュ関数とは、任意の長さの入力 x に対して、一定長の長さの出力 $y = \text{hash}(x)$ を計算する関数で、 y を見ても x を推測することが困難で、さらに、 $\text{hash}(x_1) = \text{hash}(x_2)$ となる異なる2つの入力 x_1, x_2 を求めるのが難しい関数のことである。 y を x のハッシュ値と呼ぶ。

ハッシュチェーンの構成

ハッシュチェーンとは、ハッシュ値を連鎖させることにより、データ間の順番を確定する構成のこと

である。たとえば、ある時点で台帳にデータAが登録されており、データAのハッシュ値が h_A で、台帳の現在の状態として、周知されているとする。この次にデータBを登録するときは、 h_A と、登録するデータBを連結してハッシュをとったもの、すなわち $h_B = \text{hash}(h_A || \text{データB})$ を、 h_A に続くハッシュ値として周知させる。次のデータCを登録する場合には、同様に $h_C = \text{hash}(h_B || \text{データC})$ を計算し、周知させる(図-2)。

このハッシュチェーン h_A, h_B, h_C を周知させたあとに、台帳管理者が台帳の登録順をデータA、データC、データBと変更しても、すでに周知された h_B と、 h_A とデータCのハッシュ値が一致しないので、順番操作の不正を検出できる。データの欠損にも対応できる。これを活用することにより、前述の要件3)を満たすことができる。

ハッシュチェーンは集中管理

ハッシュチェーンだけを使って台帳を構成する場合、データを台帳に登録し、対応するハッシュチェーンを作成・周知させるプロセスが1人の管理者に集中するという問題がある。たとえば、その管理者が公平性を欠き、なんらかの理由により特定のデータAを登録したくない場合、台帳に登録せずに握りつぶすこともできてしまうのである。これでは前述の要件1)を担保できない。次節では、この問題を解消し、複数の台帳管理者で分散管理する方式を概説する。

◆ ハッシュチェーンからブロックチェーンへ

ビットコインでは、だれでもハッシュチェーンに履歴を組み込むことができ、その結果をお互いで共有できるように、ハッシュチェーンの構成方法を変更した。そのためには通信の在り方、チェーンをつなげられる条件、チェーンが分岐したときのルールを定めなくてはならない。本節ではそれぞれについて述べる。

通信の在り方

要件 1) を満たすために、台帳に登録すべき履歴候補データを一部の台帳管理者が、握りつぶすことを回避しなくてはいけない。そこで、ブロックチェーンでは、複数の台帳管理者が存在し、登録依頼を任意の複数の台帳管理者に送るようにする。複数に送ることにより、1人の管理者のマシンがダウンしている場合でも、ほかの管理者が処理してくれることが期待できる。ビットコインでは、これらの台帳管理者はP2Pネットワークでつながっていて、台帳管理者は自分が受け取った登録依頼をほかの台帳管理者にも転送する仕様になっている。

しかしこれでは、ネットワークの状態により、管理者がそれぞれ受け取る履歴候補データが異なっていたり、受信時刻が前後したりすることが想定される。それでも各台帳管理者が結果的に同じデータを同じ順番で履歴として管理できるように、次節で示すブロック単位の同期手法が採用されている。

履歴のチェーンへの組込み方

各台帳管理者は、履歴候補データを受信し、それがルールに従っていると判定できた場合は、それを台帳に登録し、チェーンに組み込む。しかし、上述の通り、各台帳管理者が保有するデータとその順序はそれぞれ異なっていることが想定される。したがって、ハッシュチェーンをそのまま利用すると、スタートは同じでも、それぞれ異なるハッシュチェーンを構成してしまう（これを分岐と呼ぶ）。そこで、どの履歴候補データをどの順番で履歴チェーンに組み込むかを、台帳管理者間で同期しないとイケない。ビットコインで採用されているブロックチェーンでは、大胆にもこれを「早いもの勝ち」というルール

にした。すなわち、いち早く、次のハッシュ値を計算して、みんなに知らせた人がチェーンに組み込む人であり、ほかの人はそのデータの順番に従う、ということである。しかし、容易に想像がつくように、P2Pネットワークでつながれている状態で「だれが一番早かったか」を確認するのは至難の業である。そこで、だれが一番かを分かりやすくするために、「次のハッシュ値」を計算するためにあえて時間がかかる「暗号パズル」を採用している。時間をかけている間にほかの台帳管理者同士が同期できる効果もある。なお、それでも分岐してしまったときのルールは別項で述べる。

暗号パズルの使い方を述べる。チェーンに組み込む「次のハッシュ値」はハッシュチェーンのように単純なデータのハッシュ値ではなく、暗号パズルを解いたハッシュ値とする。ビットコインでは、解くのに平均10分かかる暗号パズルをいち早く解いた管理者のみが、次のチェーンをつなげられるというルールになっている。暗号パズルの具体的な内容は次節に譲るが、暗号パズルが解けた管理者は、その答えと、チェーンに組み込む履歴候補データをほかの管理者にP2Pネットワーク（Peer to Peer Network）で同報する。

この「直前のハッシュ値、クイズの答え、履歴候補データ」のセットを「ブロック」と呼ぶ。このブロックが連鎖することが、ブロックチェーンと呼ばれるゆえんである。いち早く次のブロックを生成できた人のブロックが同報され、ほかの台帳管理者はこのブロックをブロックチェーンに組み込むのである。そのプロセスを具体的に述べる。

新しいブロックを受信したほかの管理者は、現在解こうとしているパズルを中断し、受け取ったブロックのクイズの答えの正しさと、履歴候補データがルールにのっとっていることを確認する。確認できたら、そのブロックを自分の台帳に組み込む。そして、そのブロックに含まれていない新しい履歴候補データを、次のチェーンに組み込むべく、新しい暗号パズルの問題にチャレンジするのである。

このようにすることによって、非同期に動作する

管理者同士が、ブロックの高さ（それまで何個のブロックがチェーンに存在するか）によって同期される。台帳に組み込まれたブロックに記載されているデータが、いわば、データベースに書かれた共有の履歴データとなる。ブロックチェーンが分岐しない限り、全部の管理者で同じ履歴データを同じ順序で共有することになる。

暗号パズル (proof of work)

暗号パズルで時間を浪費させる発想は 1992 年の Cynthia Dwork と Moni Naor の論文²⁾にさかのぼる。この論文では、スパムメールを排除するために、暗号パズルを解いて時間を浪費したことを示さないとメールを送れない仕組みを提案している。暗号パズルの良いところは、時間をかけないとパズルは解けないが、一方で、パズルを解けたことはだれでも簡単に確認できることである。このような仕組みがあれば、スパムメールを送るために時間や計算機パワーが必要になるので、むやみやたらに送られなくなるだろう。

ビットコインで使われている暗号パズルはハッシュ関数の一方向性を利用するものである。

暗号パズル：チェーンにつなげる「データ (D)」と、過去の正しいブロック B_m に対して、Hash (データ (D) || ブロック B_m || N) を計算すると、ハッシュ値の先頭に 0 が k 個以上続くハッシュ値になるような N を求めよ。

ハッシュ関数の一方向性より、これを解くには、 N をさまざま変えて試してみるしかない。地道に N をしらみつぶしで探索して、偶然 0 がたくさん続くハッシュ値になるものをみつけるまで、10 分くらいかかるように、 k の数が決まっている。

ちなみに、この暗号パズルに登場する「データ (D)」とは、それぞれの台帳管理者に P2P 通信で届いた履歴候補データで、いままで、この台帳管理者が保有するチェーン上に組み込まれていないものを連結したものである。ビットコインでは、ブロックサイズに 1MB の制限があり、平均 1,000 個程度の取引データが含まれている。

分岐したときのルール

ビットコインでは 10 分という時間を設けて、同期をとれるように設計してあるものの、世界のどこかのほぼ同じ時刻に暗号パズルを解けた人が複数いることは想定範囲内である。そこで分岐が起きたときのルールも決められている。分岐が起きたときは、どちらか長い方のチェーンが有効、というルールがある。各台帳管理者が、長い方のチェーンを選択していけば、結果的にそのチェーンに収束していくことが想定されている。

短い方のチェーンに含まれていた履歴候補データは、残念ながら、この形ではチェーンに組み込まれないことになる。ただし、この履歴候補データは、複数の台帳管理者間で P2P で共有されているので、ほかの台帳管理者によって長い方のチェーンにすでに組み込まれている確率が高い。もしもなければ、次のブロックに組み込まれることになる。

台帳管理者のインセンティブ

ブロックチェーンを設計するにあたって、台帳管理者に負担がかかることを認識する必要がある。台帳管理者のなり手がいなければ、少数の管理者の独占的な運用になりかねない。ビットコインでは、自分が生成したブロックが台帳に組み込まれると、報奨として、その台帳管理者にビットコインがもらえる、という設計が組み込まれている。また、履歴データの登録依頼をする利用者が登録手数料を設定すると、このデータを台帳に組み込んだ台帳管理者が、この手数料をビットコインでもらえるということになっている。このような設計により、暗号パズルを解くという莫大な量の計算を実施してくれる台帳管理者の存在を確保しているのである。

◆ ブロックチェーンのメリット

いままで詳細に見てきたとおり、ブロックチェーンは「権限の分散」「透明性」を求めて設計されてきた。

全幅の信頼のおける台帳管理者がいなくても、互いに信頼できない複数の台帳管理者が共同して台帳を管理することができる。その際、プロセスの透明性が担保されているので、ルールにのっとった履歴

データのみが台帳に登録され、それが改ざんされたり、後日変更されたりすることがない。

そしてこの台帳は一貫性が担保され、同じものを複数の台帳管理者が所持している。したがって、一部の管理者のマシンがダウンしたり、ウィルスに汚染されても、継続してデータを蓄積管理できる。すなわち、1カ所に台帳が存在する状況より「ロバスト性」というメリットも享受できる。

ブロックチェーンのバリエーション

◆ 台帳読み書きの観点

ブロックチェーンのバリエーションとして、だれが履歴候補データを台帳に書き込んで履歴データとできるか、と、だれが台帳を読み出せるかの観点から4つに分類することができる(表-1)。ビットコインのブロックチェーンは、だれでも台帳に追加でき、だれでも台帳を読み出せる方式になっている。一方で、台帳に追加できる人を事前に任命する方式もある。いわゆる、Permissionedと言われるものに該当する。

また、アプリケーションによっては、プライバシーの観点から台帳を読み出せる人を限定したい場合もあり、別途アクセス制御手段を設けることによって、可能になる。

なお、ブロックチェーンは集中管理ではない、との表現を目にすることがある。確かにビットコインのブロックチェーンは、だれでも台帳に読み書きができる点で、従来と異なるデータ管理がされている。一方で、読み書きの権限を一部の人に限定する場合は、その権限を管理する必要が発生する。だれに読み書きの権限を与えるか、はく奪するかを管理することが集中管理になる場合もある。ブロックチェーンを使うことで、あらゆる集中管理を回避できるわけではないことを、あらためて注意喚起したい。

◆ 台帳記載時のルール of 観点

ブロックチェーンのバリエーションとして、台帳に正しく記載されたと見なすルールによって分類

	台帳読み出し者限定	だれでも台帳読み出し
台帳書き込み者限定 (Permissioned)	台帳書き込み者も読み出し者も限定	台帳書き込み者は限定されているが、だれでも読み出し可能
だれでも台帳書き込み (Permissionless)	だれでも台帳に書き込めるが、読み出せる人は限定	だれでも台帳に読み書きできる。ビットコインなど

表-1 台帳読み書き権限による分類

することができる。ビットコインは前述の Proof of Work を採用している。この方式は、だれでも参加できるが、台帳に記載するための計算量が多く、計算資源の無駄遣いという批判がある。

Proof of Stake

台帳記載時の計算量を軽減する方式として、Proof of Stake と呼ばれる方式がある³⁾。この方式は、ブロックチェーンを活用する環境の資産 (stake) を定義し、その資産の多い人は少ない計算資源で暗号パズルが解けるようにするものである。この背景には、資産を多く持っている人は、不正をして、自分の持っている資産価値を減らすような行動はしないだろうから、少ない計算量でも安全性が保たれるだろう、という仮定がある。

BFT 解法

Proof of Work や Proof of Stake などは、Permissionless、すなわち、台帳に書き込める人に制約がなく、だれが書き込むか分からない場合にも適用できる画期的な方式である。あらかじめ台帳を書き込む人が決まっている場合は、上記の方式も活用できるが、過去から研究されてきたビザンチン・フォルト・トレランス (Byzantine Fault Tolerance : BFT) のあるプロトコルを適用し、効率化を図ることができる。

BFT は、分散システムにおいて、いくつかのノードにビザンチンフォルト (応答しなかったり、悪意をもって任意の応答をしたり、など) があつた場合にも、正常なノードの合意でシステムを正常に運営できる性質である。プロトコルが BFT になるためには、不正ノードの数の上限があることが知られている。

忠実な BFT の実現には、通信量が多く複雑になる場合が多いが、PBFT⁴⁾ と呼ばれる方式が存在し、

ノード数が多くない場合には現実的な時間で実現できる。あるいは、ビザンチンフォルトではなく、故障により応答しないクラッシュフォルトにのみ対応する、Paxos⁵⁾などの実用的な方式も知られている。ただし、この方式を活用してブロックチェーンを構成する場合には、台帳管理者が故障などで停止した場合には問題なくブロックチェーンが動作するが、ウィルスに感染するなどして、ビザンチンフォルトになった場合にも動作保証が必要な場合は、別の手段で担保しなくてはならない。

台帳管理者ノードのどのような不正に対応するか、台帳記載時にどんなコストをかけられるか、に応じて、適切な技術を選択して、ブロックチェーンを設計する必要がある。

スマートコントラクト

ブロックチェーンが複数のノードの関与で、改ざんできない、信頼できる履歴管理を担ってくれるのであれば、これを活用して、信頼できるプログラム実行が可能になる。それを実現する1つの方式がEthereum⁶⁾である。Ethereumでは、プログラム自身も、プログラムの入力も、そしてプログラムの出力もブロックチェーン上に登録される。ブロックチェーンが公平性と透明性を担保してくれるので、スマートコントラクトと呼ばれるプログラムの動作を特定することができ、さらに入力に対する出力の正しさをだれもが検証できる。このプログラムで「仮想通貨」を管理にしたり、資産を管理にしたり、フィンテックの範疇を超えたさまざまな応用が期待されている。

今後の課題

本稿で紹介した通り、ブロックチェーンは従来単独の管理者で運営していたハッシュチェーンを、いかに複数の管理者で公平に、透明性を持って実現で

きるかの課題にブレークスルーを与えた。従来の暗号プロトコルではあまり考えられていなかった「事前に定まっていなかった複数の管理者」による信頼性確保の観点をみせてくれた。また、通信レイヤの課題も合わせて、実際に動作するシステムが存在することは大きなインパクトがある。いままで机上の空論に終わりがちだった数々の暗号プロトコルの需要が喚起され、より公平で透明性のある社会ソリューション実現への足掛かりになることは確実だろう。

一方で、現在のブロックチェーンがすべてのアプリケーションに適するか、今後もビットコインが安全に活用されていくのか、は未知数である。さまざまな方式が提案されている中で、それぞれが暗黙の前提としている仮定を明確化し、性能面、性質面での公平な比較をはじめ、脆弱性の有無の確認や既知のセキュリティ対策との整合性など、学術的な検証が大きく待たれている。

本稿では、筆者なりの観点で切り出したブロックチェーンのモデルを紹介した。このモデルが、暗号通貨以外にも、分散管理された状態で、透明性や公平性が必要になるシステムを設計する際に参考になれば本望である。

参考文献

- 1) Haber, H. and Stornetta, W. S. : How to Time-Stamp a Digital Document, CRYPTO 1990., pp.437-455 (1990).
- 2) Dwork, C. and Naor, M. : Pricing via Processing or Combating Junk Mail, CRYPTO 1992, pp.139-147 (1992).
- 3) King, S. and Nadal, S. : PPCoin : Peer-to-Peer Cryptocurrency with Proof-of-Stake, <https://www.peercoin.net/assets/paper/peercoin-paper.pdf> (2012).
- 4) Castro, M. and Liskov, B. : Practical Byzantine Fault Tolerance, OSDI 1999, pp.173-186 (1999).
- 5) Lamport, L. : The Part-Time Parliament, ACM Trans. Comput. Syst., Vol.16(2), pp.133-169 (1998).
- 6) Vitalik, B. : White Paper : A Next-Generation Smart Contract and Decentralized Application Platform, <https://github.com/ethereum/wiki/wiki/White-Paper> (2014).

(2016年6月2日受付)

佐古和恵 (正会員) ■ k-sako@ab.jp.nec.com

NEC セキュリティ研究所技術主幹。暗号プロトコルの研究とプライバシー保護の研究に従事。工学博士。日本学術会議連携会員。