ブロックチェーンの本質とは何か

セコム株式会社
IS 研究所
ｺﾐｭﾆｹｰｼｮﾝﾌﾟﾗｯﾄﾌｫｰﾑﾃﾞｨﾋﾞｼﾞｮﾝ
暗号・認証基盤グループ
主任研究員　佐藤　雅史　氏

【ブロックチェーンの仕組み編】

ブロックチェーンには、非中央集権的あるいは分散的にデータの真正性を担保しながらデータを共有するメカニズムというイメージがあると思うが、ブロックチェーンコミュニティにおいても人それぞれブロックチェーンを語る時の意味合いが微妙に異なり、ブロックチェーンを一言で定義するのは難しい。本講演では、敢えて Permissionless(Unpermissioned)と Permissioned と使い分ける。

■ビットコインの仕組みと特徴

ビットコインの正体は、A さんがいくら持っていて、それを B さんにいくら送るという記録を台帳に書きこんでいくものであるが、Permissionless の代表格であるビットコインが目指すのは中央機関、中央サーバーがなく、デバイス、コンピュータが相互につながる Peer-to-Peer ネットワークの中で取引を実現する世界である。

Peer-to-Peer ネットワークの中で取引情報の連鎖を実現するには、①アカウントを登録する中央機関、中央サーバーがない中でA，B といったユーザーをどう認証するか、②取引情報を顔も見えない相手のデバイスに投げていくため通信経路上で取引情報が改ざんされる恐れがある、③データが拡散、分散するので真実となるデータを確定しなければならない、という問題がある。

①、②の問題は公開鍵暗号方式を使った従来のデジタル署名で解決できるが、③の真実となるデータの確定が問題である。

たとえば、A から B に 10BTC 送るという命令を x1 として書き、同時に、A からCに 10BTC 送るという命令を x2 として書く。B に x1 の命令が速く届けば自分に 10BTC 送られたと考えるが、命令 x2 を早く受け取った人はこちらが真実のデータだと信じてしまう。Peer-to-Peer ネットワークの中では x1、x2 のどちらの取引を有効とするかジャッジしなければならない。そこでビットコインネットワークでは、取引情報は同時にいろいろな人に投げられるものの、真となる取引情報を皆が共有する台帳に書きこむことをブロックチェー

ンという。つまり、この台帳に書かれなかった取引は成立しないものとみなす。さきほどの例で x1 の命令が採用されたら x2 の命令は取引情報として認めない。

では台帳は誰が作るのか。台帳作成者を決めてしまうとそこが管理主体、中央集権的なサーバーとなってしまうため、台帳は皆で競争して作る。競争とは、よく proof of work と言われるやり方で、簡単に言うとサイコロを振って 3 より小さい値を出した人が勝ち、というルール下でサイコロを振るようなもの。ただ、サイコロは 6 面しかなく同時に 1 を出す人が多数いてなかなか勝負にならないのでサイコロの目を多くするためハッシュ関数を使ってサイコロ振りを実現しようとする。

つまり、参加者全員で共有するお題となる値を決め（自動的に生成されるターゲット値）、この値よりも小さなハッシュ値（10 分間に集まった取引情報とランダムな値を掛け合わせて算出したハッシュ値）を見つけた人を勝ちとする。ハッシュ値は暗号学的ハッシュ関数の性質からお題の値から逆算して推測することが難しいため、当たりを引くまでランダムな値を変更しながら大量に試行しないと勝負がつかない。お題よりも小さなハッシュ値を出した人が勝てば、その値を生み出す取引情報のセットとランダム値を含む各種パラメーターが新しい台帳の 1 頁（これをブロックという）として記述される。このハッシュ値を延々と掛けていくことで自然と台帳の 1 頁 1 頁が綴られ、10 分おきに 1 枚ずつ追加されていく。台帳が作れた人、つまり 10 分間に 1 回当たりを引いた人には、発掘したビットコインが報酬として新しく割り当てられるようになっている。ビットコインのブロックチェーンは、最初のサイコロ振りで当たりを引いた人が発掘したビットコインから始まっている。

■チェーンの分岐による問題

台帳は皆で厳密に共有しているわけではなく、皆が「勝った、作れた」という結果をネットワーク上に投げているのでネットワークの分断や遅延により「勝った」という通知がこない状態になることもある。そうすると、1〜3 までは皆同じチェーンを見ていたが、

・1→2→3→4a→5a→6a→7a（4a の通知が来た人たちのチェーン）

・1→2→3→4b→5b→6b（4a の通知が来なかった人たちのチェーンの流れ）

のように、4a の通知が来なかった人たちのチェーンという別のチェーンができてしまう。

ビットコインでは、長い方がより多くの人達が支持しているチェーンと考え、長い方を残し短い方を却下、つまり台帳として綴じずに消す。そのため、いくつかのブロック生成を待ち「さすがにこんなに長いものが覆されることはない」という段階で初めて取引成立状態になる。従ってビットコインのデフォルト設定では 1 ブロック生成に約 10 分かかるブロック 6 個生成、すなわち約 60 分で取引情報確定となる。

ブロックチェーンをねつ造するには、「長いチェーンを採用する」という性質を逆手に取り、あらかじめ長いチェーンを作っておきある時期に開放する。しかしそれには参加者の計算機能力を上回る能力を持っていなければならないのでねつ造よりもマイニングをした方が経済合理性が高く抑止効果があると考えられている。

■ビットコインからブロックチェーンへ

ビットコインの代表的なコンセプトは①中央集権的機関や運営主体がなくても動く暗号通貨（非中央集権性）、②機械的なルールにより実行されること、である。proof of work は「非中央主権性」の一方で、計算機能力に投資できる人が勝つモデルである。proof of work とは異なる台帳共有メカニズムの提案、あるいは取引情報閲覧へのアクセス制御といった、派生や拡張したメカニズムを考える動きもある。また、機械的なルールにより実行されるというメカニズムを契約行為などの機械的運用に使おうとする「スマートコントラクト」を実現しようという動きもある。

■Permissioned と Permissionless

Permissionless というのは、参加するのに申請や承認が不要なものである。Permissionless の代表例であるビットコインは、ソフトウェアを立ち上げてネットワークにつなぐだけで良く、辞めたければつながなければ良い。参加者がやることは「ウォレット」と呼ばれる自分の ID となるデジタル署名の鍵を作ることくらいである。従ってネットワークに誰が参加してくるかわからないという前提でも成立するメカニズム、さきほどのビットコインの例では proof of work といった競争原理を備える。

他方、Hyperledger Fabric、R3 Corda、Ripple などの Permissioned と呼ばれるものは、身元確認など承認のメカニズムを持つものである。

例えば、Hyperledger Fabric では電子認証局の役割を果たす MSP（Membership Service Provider）がユーザーや各ノードの参加可否を決めメンバーシップ登録を行う。また、取引情報や Chaincode の検証や仮実行（ポリシーのチェック、実行結果の取得）を行う「(endorsing) peer」、取引情報（Chaincode 含む）の順序の決定、取引情報を格納したブロックを生成する「orderer」がいて、役割が明確に割り当てられ、MSP から発行された証明書を持たなければ参加できない。アプリケーションが取引情報を生成すると peer に投げる。peer は皆同じ台帳を持っており、台帳に照らして取引情報をチェック（試行）する。取引情報に peer が試行した結果をエビデンスとして付けて orderer に投げると、orderer はブロックを時系列順に並べその結果をまた peer に返し台帳に反映する。

このように一口にブロックチェーンと言っても、ビットコインと Fabric では前提や仕組みがまるで異なる。

Permissioned と Permissionless の共通点、相違点は何かというと、皆で検証しているという点は同じかもしれないが、Permissionless は自由参加、Permissioned は承認を要するというところであり、前提が異なるために仕組みがまったく違うものになっている。

【問題整理編】

人はブロックチェーンに何を期待しているのか、それが人それぞれであることがブロックチェーンの定義を難しくしていると思う。

改ざん検知が可能なこと、データ（取引）の存在証明／事後検証であれば従来のタイムスタンプや電子署名技術でも可能であるし、ルール実行・検証の自動化や可用性の向上であれば従来技術の冗長性を高めれば良いのではないか。

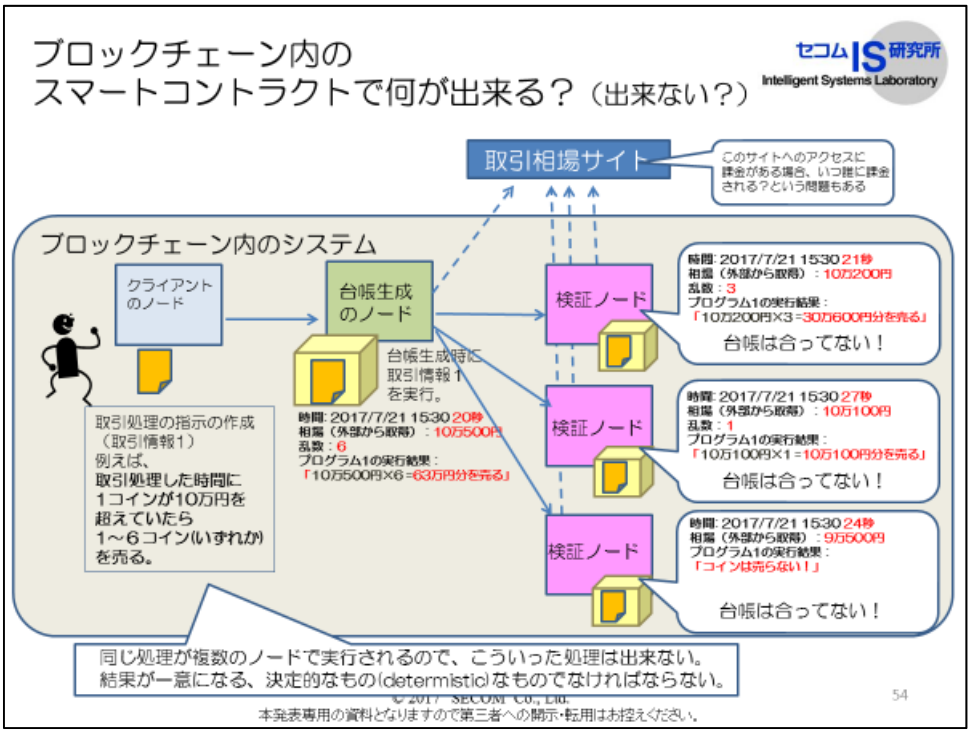
非中央集権性、End-to-end を実現するためには、情報に偏りがあってはならないので参加者皆が同じ情報を持ち対等に判断を下す必要がある。しかし全員が同じデータを持つように同期をとることは、これまでの分散処理の研究からさまざまな前提を置かなければ実現不可能であることがわかっている。

Permissionless はさらに困難で、誰がいつ入ってくるかわからない。そういう世界で特定時間内に同じデータの同期をとるのはかなり無茶である。そのため、ビットコインは皆が同じデータを持つわけではなく、ある意味「割り切って」ある瞬間にジャッジするとしている。全員が同じデータを持つことは前提とせず、同じデータを持つ多数の参加者がいればジャッジに偏りはなくなるであろうという期待に基づいたモデルのように見える。他方、Permissioned はある閉じられた世界で、参加するノードを限定し、一定の整合性やデータの一貫性を持たせようとしている。しかしその一方で、さまざまなことを確認し合うのでメッセージ数の増加、処理コスト増を招く可能性がある（従来の分散システムと同様の議論になりうる）。分散システムでは CAP 定理（一貫性（Consistency）、可用性（Availability）、分割耐性（Parition Tolerance）のうち 2 つしか同時に満たせない）というものが言われているが、様々なブロックチェーンのプロダクトにおいても、どの機能をどの程度実現するのか前提も目指す方向もまったく違うため、一括りに「ブロックチェーン」と語り比較することは無理があるのではないか。

■ブロックチェーン内のスマートコントラクトでできること、できないこと。

ブロックチェーンでうたわれるスマートコントラクトというのは、ブロックチェーンの中に配置されそのメカニズムの中で実行されるプログラム、つまり、ブロックチェーンが実行されるときに一緒に実行されるプログラムのことである。

ここに、ユーザーA が「ブロックチェーン外の取引相場サイトを見て、1 コイン 10 万円を超えていたら 1〜6 のうち乱数で決めた数だけ売る」という台帳生成の役割を果たすノードがあるとする。さらに、動作検証する検証ノードがブロックチェーン内のシステムにあるとする。台帳生成時に、ブロックチェーン外にある取引相場サイトで 1 コイン 10 万 500 円と 10 万円を超えており、乱数 6 が出たので 100,500 円×6＝630,000 円を売る。検証ノードが 1 秒遅れて取引相場サイトを見に行くと 1 コイン 100,200 円となり、しかも乱数 3 が出たので 3 コイン売るとし、結果 300,600 円売るとなり、台帳生成ノードと検証ノードが合わなくなる（図）。



よって、時間に基づき外部要因を参照することや、乱数を用いるというのはブロックチェーン上のスマートコントラクトの限界である。というのも、プログラムを相互にチェックする仕組みであり、「誰かの特定のノードを作りプログラムを実行して終了」ではなく、皆で実行しチェックするものなので、いつ誰が実行しても同じ結果にならないといけないような処理しかできない。よって、取引相場サイトの結果をブロックチェーン内に落とし込むノードがあり、それをブロックチェーン内の皆が見に行くようにすれば実現できる。そのためにはブロックチェーンの外と内をつなぐシステムが必要となる。当たり前のことであるが、ブロックチェーン内でトレースできる記録はあくまでもブロックチェーン内で起きた取引情報に関する履歴であってブロックチェーン外で起きた事象はトレースできないことには留意されたい。

■The DAO の ETH 流出

プログラムの実装世界と現実社会のつながりの問題の顕著な例として The DAO の流出事件（2016 年 6 月）がある。The DAO も Ethereum のブロックチェーン上に組まれており、自動でクラウドファンディングができ、なるべく人のオペレーションが入らないようにしたメカニズムである。しかし、Ethereum ではなく The DAO のプログラムにミスがあり、数十億円規模のイーサというツールが特定のアカウントに流出した。

ブロックチェーンの仕組みとは、改ざん不能で脈々と記録がつながり、巻き戻せない世界が理想形なので、何もしないことが理想を追求した形だと思うが、Ethereum の開発コミュ

ニティや The DAO 関係者で議論し、プログラムを上書きして時計の針を事件が起こる前に戻した。つまり、この事件はなかった、数十億円の流出はそもそもなかったというように台帳が書きかえられ、プログラム上できないとされていることが人の判断によって強制的にできてしまった。

それでは、ブロックチェーンがやりたかったことはそもそも何だったのか。原点に立ち返ると Satoshi Nakamoto の論文はビットコインだけで閉じた「ビットコイン世界」というものを描いていて、そこではビットコインだけが流通しビットコイン世界の中だけの経済合理性によって成り立つという前提に立っているのではないか。そこではプログラム化されたルールによって発行される通貨が勝手に動く暗号通貨を中心とした系だけで動く世界というのを描いていたと思う。

しかし実際は、ソフトウェアで動く世界もミドルウェア、OS、ネットワークがありネットワークで相互につながっている。ビットコイン世界にも miner がいてコイン交換所、ビットコインを決済に使う事業者もいる。Satoshi Nakamoto が前提としていた世界だけで動くのではなく、外界との関わりを持たざるを得ない。そのためビットコイン世界を前提として機能するように設計されたものに亀裂が入る可能性が出てくる。その一つの例が Segwit や Bitcoin Unlimited などの分裂騒動といえる。今後も、利用が拡大するにつれ参加者が増え、当初思い描いていた理想的な世界とのずれが大きくなっていくとしても致し方ないと思うが、コミュニティの自治によってどのように解決にしていくか着目したい。ブロックチェーン上の機械的なメカニズムがうまく作用するにも、それを生み出すコミュニティが健全に維持されることが前提となる。

■ブロックチェーンの本質とは何か

大切にしたい要素とは何か。それによって解決方法や課題へのアプローチ方法が異なると思う。重要なのは非中央集権性なのか、それとも、一部であれば一定の管理下に置いても許容できるのか。あるいは透明性の確保が重要なのか、何に対して共有や相互検証が必要なのか。

「多数のノードにより協調し相互に検証し、改ざんが困難なハッシュ値のチェーンを共有すること」は単に手段である。「ブロックチェーンの本質とは何か」という問いかけよりも何を目指したいのか？どうあるべきか？をよく考えることが重要だと思う。