小特集
IEICE Communications Society Magazine
Special Issue

編集チームリーダ　大下裕一
Yuichi Ohsita

今だからこそ知りたいブロックチェーンの基礎

2008年にサトシ・ナカモトを名乗る人物によって投稿された「Bitcoin: A Peer-to-Peer Electronic Cash System」というタイトルの論文には，銀行のような中央管理者を介することなく，通貨の取引を行うための仕組みが記載されていました．その仕組みの中核を成す技術がブロックチェーンです．その後，ブロックチェーンに基づく，様々な暗号通貨が誕生しました．ブロックチェーンは，通貨の取引を行うための仕組みとして誕生しましたが，その技術はそれ以外の分野へも応用可能なものです．物流への応用など，暗号通貨以外の分野へのブロックチェーンの応用のニュースもよく耳にするようになったと思います．

本小特集では，このようなブロックチェーン技術について，データ構造と動作原理，ブロックチェーンの根幹を成す合意形成アルゴリズムやハッシュ，電子署名技術について，初学者にも分かりやすいような解説を，第一線で活躍されている研究者の方に執筆頂いています．更に，暗号資産についての現状も解説頂いています．

本小特集を読んで頂くことにより，ブロックチェーンの全体像をつかんで頂ければと思います．

最後に本小特集の発行にあたり，御執筆頂いた皆様，編集委員の皆様，校閲に御協力頂いた皆様に心から感謝致します．

小特集編集チーム
大下裕一，北尾光司郎，久保亮吾，小林亜樹，
佐藤陽一，橋本尚久，山口実靖，山本　嶺

ブロックチェーンのデータ構造と動作原理

小出俊夫　Toshio Koide　NEC セキュリティ研究所

1 はじめに

「ブロックチェーン」が何を意味しているかは，人により大きく異なるだろう．ブロックと称するデータの「かたまり」がチェーン状につながったデータ構造のことや，そのデータ構造を維持する分散システムのことかもしれない．はたまた，データを改ざんできない魔法のようなデータベースや，世の中を大変革してしまう技術のことかもしれない．ここではそのどれが正解であるかは追求しないが，少なくともブロックチェーンという用語を有名にした「ビットコイン」の目的を通して，その意味の輪郭をつかみ取ることはできるだろう．

ビットコインは，インターネット上の支払いシステムである．「サトシ・ナカモト」と称する人物が，ある暗号関係のメーリングリストで発表 [(1)] した論文 [(2)] のアイデアを元に有志が実装し，2009 年 1 月から現在まで動作し続けている．

ビットコインの目的は，第三者を介さずにインターネット上の支払いを可能にすることである．第三者とは例えば銀行に代表される金融機関が該当する．金融機関は不特定多数との取引を安心して行えるよう信用の維持と利用者の保護を図る必要があり，日本では金融庁や財務局の免許や許可の下でサービスを提供している．その安心と引換えに，ルール遵守や信用維持には少なくないコストが掛かっており，厳重な本人確認や送金額の制限，各種手数料の高さなどとなって現れてくる．

ビットコインの主な貢献は，皆が第三者を必要とせずにネットワーク全体を信用する状況を作り出す，トラストレス（trustless）の分散型信用基盤を実現したことにある．これによって，第三者の制約を受けずに誰でも（IoT 機器や AI など人間以外も）口座を持ち，世界をまたがり自由に安心して送金できる状況を作り出した．その一方で，例えばエネルギー効率やトランザクション性能は従来の分散データベースと比較にならないほど劣っている．

しかしその貢献には，性能の悪さをカバーするのに十分なインパクトがある．ビットコインやブロックチェーンが注目される理由は，信用の源泉が変化し様々な発展性と可能性をもたらす社会的インパクトと，それを実現させる技術的インパクトと，投機対象として魅力的価値を持つに至った経済的インパクトが背景にあるとみてよいだろう．

例えば，国籍不明のロボットがインターネット上で自律的にお金を稼ぎ，本人の証明ができず銀行口座を開けない難民の子供たちへ，直接，国際送金して寄付をすることも可能になる．更に，ビットコインでは送金を権利移転として表現するので，これを一般化し，特許，著作権，登記などの権利管理に広げれば，関連する団体や組織といった第三者をプログラムに置き換えて自動化・無国籍化する可能性もある．また，裏付けを持たない単なるビット列が価値を持ち，投機の対象となった事実も驚嘆に値する．

ビットコインの一般的な技術の解説は文献（3）に譲り，本解説記事では，ブロックチェーンの技術的インパクトの理解を深めることを目的とし，**2.** でビットコインのデータ構造と権利移転機能の応用の一つとしての送金の表現，**3.** で二重支払いや改ざんの問題に第三者を介さずに対処するビットコインの巧妙な動作原理をそれぞれ解説し，その知識を元に **4.** でブロックチェーン全般の今後の応用や課題について触れる．読者のブロックチェーンに対する理解を，ビットコインの構造と動作原理の理解を通して深めることができれば幸いである．

2 データ構造と表現

2.1 秘密鍵で作る口座

ビットコインには公開鍵のハッシュ値を文字列表現したビットコインアドレスというものが存在する．例え

ば，「1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa」は，ビットコイン上で初めて記録されたビットコインアドレスである．ビットコインアドレス A は次のように求まる．

$$K = \text{'0x04'},\mathrm{ECDSA}(k)$$
$$A = \mathrm{Base58Check}(\text{'0x00'},\mathrm{RIPEMD160}(\mathrm{SHA256}(K)))$$

ここで，k は 256bit の秘密鍵，K は secp256k1 をパラメータとする楕円曲線暗号を用いた署名アルゴリズム ECDSA の出力 512bit の前に 8bit のプレフィックスとして非圧縮を意味する 4 を追加した 520bit の公開鍵，Base58Check は入力にチェックサムを加えて人間が扱いやすい文字列にエンコードする関数，RIPEMD160 と SHA256 は暗号学的ハッシュ関数である．

ビットコインアドレスは口座にたとえられる．口座の基本的な要件である口座の開設は秘密鍵の生成，暗証番号や銀行印は秘密鍵，口座番号は秘密鍵に対応するビットコインアドレスに相当する．ビットコインアドレスからは秘密鍵を生成できないので，送金先として安心して公開できる．なお，秘密鍵は暗証番号や銀行印のような気軽さでは変更できないため，新たなビットコインアドレスを作って対応する．秘密鍵は誰でも自由に作成でき，誰が持っているかを記録することもないため，ビットコインアドレスの所有者の確認が困難である点は，銀行口座の要件とは大きく異なる点である．

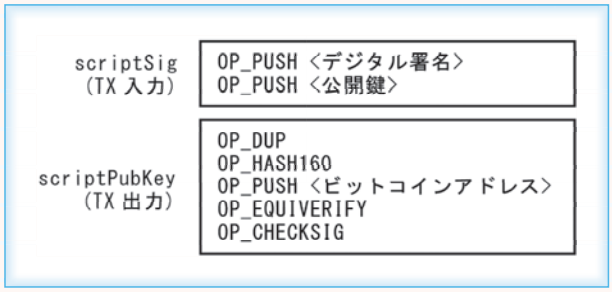

図 1　P2PKH 形式のスクリプト

2.2　UTXO による残高の表現

ビットコインの基礎となるデータ構造は，UTXO（Unspent transaction output）である．UTXO はビットコインの量と，scriptPubKey というフィールドを含んでいる．ビットコインの量は最小単位である satoshi で表現する．10^8 satoshi = 1 BTC である．scriptPubKey は，Locking Script とも呼ばれるスタックベースのスクリプトである．

2019 年 12 月現在，ビットコインのネットワーク上には有効な UTXO が約 1 億個存在し，誰でも読み取ることができるが，誰でも使用できるわけではない．そのビットコインを使用できるのは，scriptPubKey の前方に Unlocking Script とも呼ばれる scriptSig スクリプトを付け加えて実行した結果が True になるときに限る．

スクリプトには幾つかの慣例がある．P2PKH（Pay to Public Key Hash）と呼ばれる形式を図 1 に示す．動作手順の解説は文献（4）に譲るが，この二つのスクリプトを連結して実行した結果を True とするのは，scriptPubKey に指定されたビットコインアドレスに対応する秘密鍵によるディジタル署名と公開鍵を scriptSig に指定した場合のみである．すなわち，ビットコインアドレス A は対応する秘密鍵 k を持つ者への支払先のように利用でき，scriptPubKey 内に A を含む UTXO のビットコインの総和を A の残高と表現できる．つまり，全ての残高は全て公開されており，この特徴は銀行の残高の秘匿性の要件とは大きく異なるところである．

P2PKH 以外にも様々な表現が可能である．ビットコインのスクリプトはチューリング完全ではないが，チューリング完全なプログラム（スマートコントラクトと呼ぶ）を記述可能な Ethereum に代表されるプラットホームも存在する．

2.3　トランザクションによる送金の表現

scriptPubKey の書かれた UTXO と，その使用権を主

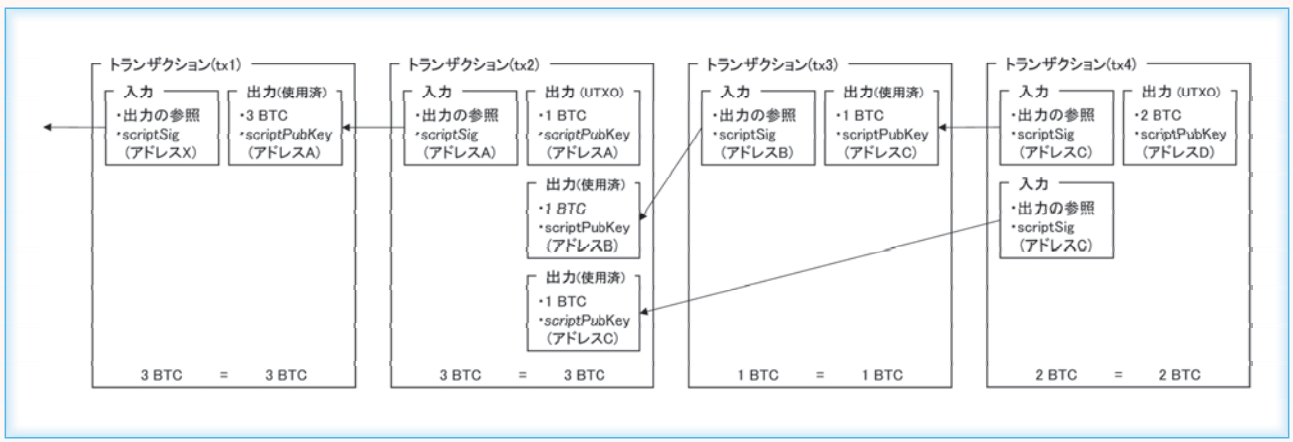

図 2　UTXO とトランザクションの例

張する scriptSig は，それぞれ個別のトランザクションというデータ構造に記述される．トランザクションは，一つ以上の入力と出力を含み，入力は UTXO への参照と scriptSig，出力は新規の UTXO である（図 2）．入力の scriptSig が正しければ，参照された出力は UTXO から使用済み出力へ変化する（図 2 の tx2 を参照，矢印は参照関係を示す）．

トランザクションの出力のビットコインの総量は，後述の例外を除き，入力で参照した他のトランザクションの出力のそれを超えてはならないというルールがある．一般的に送金は，送金元から引かれる額よりも送金先に到達する額が増えてはならない要件があるが，この特徴を用いれば，自分が使用できる UTXO のビットコインの総量の範囲内で，相手が使用できる UTXO を作ることでこの要件を満たし，ビットコインにおける送金を表現できる．

例えば，図 2 の tx1 の作成時には UTXO が一つあり，ビットコインアドレス *A* を含んだ scriptPubKey が 3 BTC とともに存在していた．このとき，*A* には 3 BTC の残高があるとみなされる．次に，*A* の秘密鍵を持つ者が tx2 を書いた．入力の scriptSig が正しければ，参照された tx1 の UTXO は使用済み出力となる．tx2 には UTXO が三つあり，それぞれビットコインアドレス *A*，*B*，*C* を含んだ scriptPubKey が 1 BTC とともに存在している．これは，*A* の 3 BTC を使って *B* と *C* へ 1 BTC ずつ送金し，お釣りを *A* に戻したことを意味し，*A*，*B*，*C* それぞれに 1 BTC の残高があるとみなされる．同様に tx3 は *B* から *C* へ 1 BTC 送金し，tx4 は *C* から *D* へ 2 BTC 送金したことを意味する．ここで，tx2 は一つの UTXO を分割し，tx4 は複数の UTXO を一つに統合した例であり，最終的に *A* に 1 BTC，*D* に 2 BTC の残高があるとみなされる．

2.4　ブロックチェーンによる履歴管理

トランザクションは，ブロックと呼ばれるデータ構造に書かれる．ブロックには，一つ以上のトランザクションと，一つ前のブロックへの参照が含まれている（図 3）．ブロックの参照には ID を用いるが，ブロック内には ID のフィールドはなく，ブロックのヘッダに対して暗号学的ハッシュ関数を 2 回適用した値を ID として使用する．

> ブロック ID = SHA256（SHA256（ブロックヘッダ））

図 3 右下のように，ブロックは参照によって過去に向かってつながった木構造となり得るが，最終的には後述の手法によってその中から 1 本のチェーンが選択されるような構造に収束することから，これをブロックチェーンと呼んでいる．

ブロック内のトランザクションを一部分でも書き換えると，ヘッダに含まれるマークル木のルートハッシュが変化し，定義に従いそのブロックの ID も変化する．すると，その次のブロックが参照するブロック ID と不一致になる．よって，ブロック ID さえ保持しておけば，そのブロックまでに含まれる全てのトランザクションに対して，後から改ざんされたかどうかを検証できる．送金や残高の履歴管理の一般的な要件として求められる改ざん検知はこのようにして実現されている．

また，全ての UTXO はブロック内のトランザクションの出力に含まれている．もう一つの要件として，それらの履歴と最終的な残高の整合が取れていることが挙げ

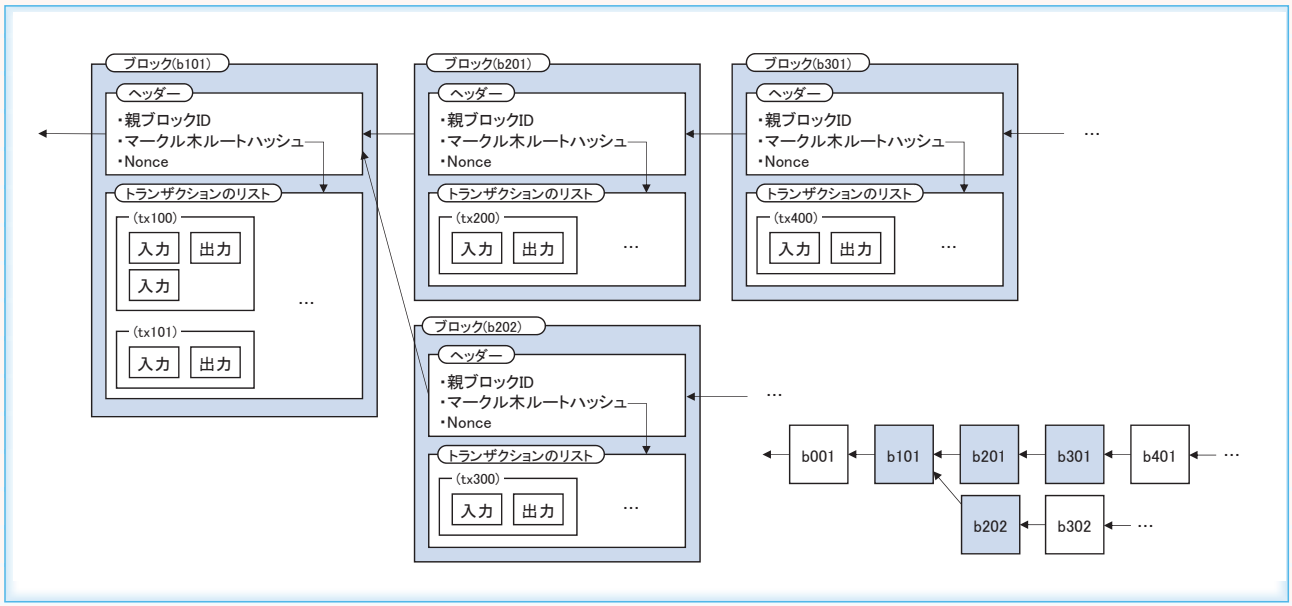

図 3　ブロックとブロックチェーンのデータ構造

られるが，アドレスの残高や送金の履歴は，ブロックチェーンを最初から順番に読み取ってトランザクション間の関係を追い掛けていくことで表現でき，誰にでも追跡可能であり，その計算結果が現在の残高として表現されるのでそもそも整合を取る必要がなく，要件は満たされている．

2019 年 12 月現在，ビットコインのブロックチェーンとして，60 万個以上のブロックが存在している．初めて作成されたブロックは genesis ブロックと呼び，2009 年 1 月に作られた．全てのブロックは，過去に遡ると必ず genesis ブロックにたどり着く．

3　第三者を介さない仕組み

3.1　利用者が提供者にもなるシステム

ビットコインには，公式の実装も，公式の Web サイトも存在しないし，そもそも発案者のサトシ・ナカモトが誰なのかも分かっていない．そして，ビットコインの仕組みそのものも，P2P（Peer-to-Peer）技術を応用し，徹底的に第三者に依存しないシステムになっている．P2P 技術は，第三者に相当する中央サーバやクラウドの代わりに，ノードが互いに同等な仲間（Peer）として接続し合うネットワーク構造を作り，全体として一つの目的を達成可能とする技術である．ノードは参加も脱退も自由だが，ネットワーク全体としては与えられたサービスを維持するように各ノードが協調動作する．

ビットコインはこの P2P 技術を応用し，サービスの利用者としてのノードが，同時にサービスを提供する側の一部に組み込まれるよう設計されている．あなたがビットコインのプログラムを起動すると，世界のどこかで誰かが動かしているビットコインのプログラムと接続する．そしてそのノードに他の接続先を聞いて，更に多くのノードと接続する．こうして，あなたのノードもビットコインのネットワークの一部として動作し始める．ブロックチェーンを構成するトランザクションなどのデータは，このネットワーク内でコピーされて維持される．2019 年 12 月現在，世界中で 1 万前後のノードによって，ビットコインネットワークが維持されている．

3.2　ビットコインが解決する重要な問題

ここで，そのコピーされるトランザクションが全て同一ではなく，悪意をもって本物とは異なったデータにしてコピーするノードがいたらどうなるだろうか．例えば図 4 にあるように，アドレス *A* からあなたが所有するアドレス *B* へ送金するトランザクションが入ったブロック Y を見て，あなたは納得して *A* の所有者へサービスを提供したとする．しかし *A* の所有者はブロックをわざと二つに分岐させており，あなた以外には，あなた宛に使われていたはずの UTXO を使って *C* 宛に送金するトランザクションを含めたブロック Z を送信していた．もしブロック Z が世界中に拡散するとそれが正しい履歴となり，ブロック Y の方は無効となり，あなたは無料でサービスを提供したことになってしまう．

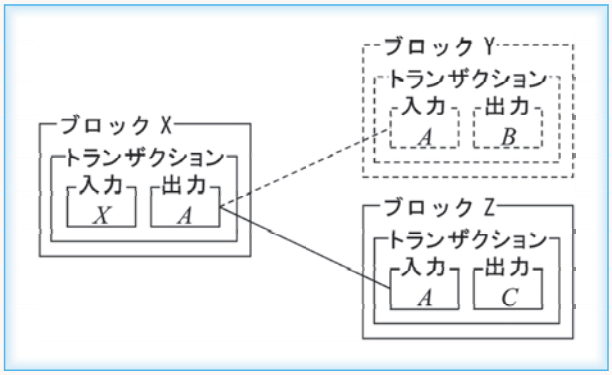

図 4　二重支払いの例

これは，ネットワークを分割してブロックの追加（ねつ造）と削除（隠滅）を行った例であるが，時間差を使ってもブロックの変更（改ざん）を行うことができる．例えば，あなたのノードが，ブロック Y をいち早く同期して手に入れたとする．その後，送金者はブロック Y をブロック Z に改ざんしたとする．あなたはその検知ができてもその行為を止めることはできない．同期が遅れて改ざん後のコピーを初めて受け取ったノードからすれば，ブロック Z が正当に見える．こんなシステムでは安心して使うことができない．

この問題は「二重支払い問題」と呼ばれるが，全てのノードが一本に収束したブロックチェーンを全ての瞬間で完全に同期して持っていれば解決する．だが，そんなことを実現する方法がビットコインには実装されているのだろうか．答えは現時点では「ノー」である．現在のビットコインは改ざんの定義を少し変えることで，この問題を事実上解決している．

3.3　ビットコインにおける悪意の定義

システムとしてはデータの変更にすぎない操作が改ざんと呼ばれるのは，そこに悪意が介在するからである．第三者に紛争解決を求めることを放棄するならば，システムが人間の「悪意」を客観的な数値で判定し自動的に解決しなければならないが，ビットコインは多数派に反することを悪意とみなすこととした．木構造状に分岐するブロックから多数派が 1 本のチェーンを選択するとき，それ以外のチェーンの作成や選択はこの定義のおかげで悪意と判定できる．

多数派を決めるために，ビットコインでは PoW (Proof of Work) [5], [6] を用いる．ビットコインにおけ

るPoWとは，ブロックIDが一定の範囲内に収まるときのブロックのヘッダのNonceのことである．Nonceとは，ブロックヘッダに含まれる4Byteの数値であり，値そのものには意味を持たないが，Nonceが少しでも変化するとブロックID（前述のブロックヘッダを入力として暗号学的ハッシュ関数を2回適用した値）が全く違う値に変化する．暗号学的ハッシュ関数の特徴である原像困難性（与えられたハッシュ値に対応する入力を見つけることが困難な性質）によって，条件に合うNonceは条件を満たすまでその値を変化させながらブロックIDの計算を繰り返すよりほかに見つける方法がない．

この性質により，そのブロックIDが一定の範囲内に入っているときのブロックヘッダのNonceは，その計算をし続けたことの証明となる．与えられたNonceが条件に一致しているかは素早く検証できるが，条件に一致するNonceの発見には時間が掛かる．2019年12月現在，ビットコインのネットワーク全体がたたき出すことが可能な計算回数は毎秒10^{20}回前後と推定されており，その力を持ってしても条件に合うNonceを発見するまでには平均10分ほど掛かる．（そうなるようにブロックIDの一定の範囲が自律的に調整される．）

ここで，一つの興味深い現象が発生する．チェーンの選択基準が同一なノード集合は，ノードのどれがNonceを発見しても同じ基準のチェーンを選択するから，結果的に一つのグループとしてまとまっていくのである．Nonceの発見は確率的なので同一グループのノードが同時に計算するとNonceの発見も早まる．そのため，最大の計算能力の総和を持つグループが最も素早くチェーンを伸ばしていくことになる．結果として，ビットコインネットワークにおける多数派は，同一のチェーン選択基準を持った，計算能力の総和が最も大きいノードグループということになる．よって，安定して信用できるビットコインネットワークを形成するには，チェーンの選択基準を収束させることが重要となる．

3.4　人間の欲望がチェーンを収束させる

ブロックには入力のないトランザクションを含められるというルールがある．出力量に一定の制限が存在するが，ノードはUTXOを自分のアドレス宛にした入力のないトランザクションをブロックに含めることで，そのノードはビットコインを発行して自分で所持できる．

実はこのルールによって，人間の欲望とチェーンの選択基準がつながる．大変な思いをして運良くNonceを発見し有効なブロックを作成したノードの運用者は，そのブロックによってビットコインを発行した事実をより強固にブロックチェーンに刻みたいと考えるのが自然である．そのためには自分の作ったブロックの後ろに多数のブロックが連なってほしいし，他のノードも同じ動機で動いているという推測が働くので，計算量が最も多く注ぎ込まれかつ有効なデータ構造を持った，最も覆りにくいチェーンを選択し自分もその作成に寄与することが基準となる．よって，少しでも多くの利益を得たい運用者には，その基準で動作するノードを運用する動機が生まれる．そして，その欲望は人間の大多数に共通しているため，その選択基準が多数派となる．

こうして，ブロックチェーンの分岐が一時的に発生したとしても，計算量が最も多く注ぎ込まれた有効なチェーンに収束し，それ以外のチェーンの選択は悪意であるという基準がネットワークに安定的に生まれ，ノード間の紛争が自律的に解決される．少数派がデータを改ざんするために過去に遡って矛盾なくブロックチェーンをつなごうとしても，それを上回るスピードで多数派によるブロックチェーンが成長するため，データの改ざんが事実上不可能となる．もちろん，多数派を牛耳れば過去のトランザクションの変更は可能だが，多数派の全てのノードを特定の運用者が牛耳ることは事実上困難であり，データの「変更」も事実上困難となる．

ブロックチェーンが正しく有効なブロックをつないでネットワーク全体で一つに収束することも，改ざんが事実上不可能となることも，ノードを運用する人間の利益を得たいという欲望によって結果的に起こっている振る舞いにすぎない．人間の欲望が改ざんできないブロックチェーンを安定的に維持する力となって今も動き続けているのであって，これは驚くべき事実である．

4　これからのブロックチェーン

4.1　技術的な課題

ブロックチェーンの技術的な課題として，スケーラビリティはよく話題に挙がる．例えば，ビットコインが処理できるトランザクションが実質的に1秒間に約4件で頭打ちとなる問題があるが，これはブロックの承認頻度が10分ごとでサイズ上限が1MByteであることに原因がある．Litecoin [7] のようにブロックのサイズや承認回数を増やせば処理性能は向上できるが，それと引換えに改ざん耐性の劣化やDDoS攻撃などのセキュリティリスクが高まる．そこで，チェーンの選択基準 [8] や，ネットワーク層の改善 [9] を通して，セキュリティリスクを緩和する方法や，トランザクションのサイズを圧縮する方法 [10], [11] が提案されている．

また，エネルギー消費に関する議論も盛んである．例えば，PoWのために全世界で大量のエネルギーが消費されている問題がある [12], [13]．現在のエネルギー消費

の推計を可視化するサイトもあり[14]，これによれば2019年12月現在のビットコインネットワーク全体の消費電力は約73TW･h，全世界の電力消費の約0.3%，日本の電力消費の約7.8%に相当する．この問題を解決するため，PoWではなくPoS（Proof of Stake）[15],[16]という通貨の保有量や年齢の証明，PoI（Proof of Importance）[17]という重要性の証明を活用してチェーンを収束させるブロックチェーンを構築する動きがあるが，これもセキュリティとのトレードオフの解決が課題となっている．

セキュリティそのものの研究もある．例えば，量子コンピュータの発展により暗号学的ハッシュ関数や楕円曲線暗号が危たい化し安全性が保たれなくなる問題への対処を目指した，量子コンピュータに対する耐性を持ったブロックチェーンの研究がある[18],[19]．また，ビットコインには管理者がいないのでどの改善手法を選択するかは利用者次第であり，互換性のない実装を持ったノードが同時に存在しつづけた結果，恒久的にチェーンが分岐するハードフォーク（Hard Fork）という現象が発生することもある．これは悪意の定義が異なるネットワークができたことを意味する．例えば，スケーラビリティの改善手法に関する対立によってビットコインキャッシュ[20]などの新たな暗号資産が誕生しており，これは技術の発展や多様化の一現象である．また，The DAOの資金流出事件への対応でEthereumの歴史を巻き戻すために起こったハードフォーク[21]は，第三者が強制的に介入した改ざんという見方も，善意の資金流出を悪意とみなす多数決によって善悪が変化したという見方もできる例である．

その他の話題も含めて最新情報を追うには，コミュニティの中でどのような議論がされているのかを知るのが一番である．ビットコインの改善提案を行う場としてBIP（Bitcoin Improvement Proposals）[22]や，開発者コミュニティの中でアイデアを議論できるメーリングリスト[23]などがインターネット上に存在する．また，ブロックチェーン関連の各種ワークショップ[24],[25]もある．オンライン・オフラインで技術的な議論や人脈を形成することも効果的な情報収集につながるだろう．

4.2　ビットコイン以外への応用

ビットコインのブロックチェーンとは，人間の欲望によってインターネットに浮かび上がったUTXOという価値の塊，すなわちビットコインの価値そのものと表現できるだろう．この価値をビットコイン以外へと一般化し，分散型の信用基盤としてブロックチェーンを成立させるには，少なくとも次の三つが注意深く設計されていることが望ましい．①は**2.**，②と③は**3.**で解説したビットコインが成立するための仕組みの本質に相当する．

①扱うべき価値とその表現方法
②サービス利用者の活用すべき動機
③①と②から継続性や改ざん耐性への変換方法

有望な応用の一つに特許，著作権，登記などの権利管理の自動化がある．権利はお金の価値と近く，人間には権利を欲するという共通の動機があるため，ビットコインのブロックチェーンと類似した設計で，分散型の信用基盤を構築できる可能性が高い．

ブロックチェーンを暗号資産以外に有効に応用できた例はまだ少ない．ブロックチェーン活用をうたう実証実験が散見されるが，新しい目的に適った分散型の信用の設計が不完全で第三者の信用を部分的に借りる必要があるか，そもそも目的が第三者の排除を前提としておらず，ブロックチェーンではなくクラウド上に実装した方が目的を効率的に達成できる場合が多い．

ブロックチェーンの良い応用例を生み出すには，ブロックチェーンの技術や性質をベースにアイデアを発想するよりは，第三者を介さずに信用が分散して存在する世界をベースに技術や常識抜きに発想したほうがよい．しかしそれは往々にして何らかの利権構造を破壊する結果につながるので，サトシ・ナカモトに倣ってその導入においても徹底的に第三者を排除する配慮が必要となるだろう．若しくは，最初は小さなコミュニティの中で楽しく使うだけでもよいかもしれない．その有用性に気付いたコミュニティの外の人たちよって肯定的に広く普及していく可能性もあるからだ．

■ 文献

(1) S. Nakamoto, “Bitcoin P2P e-cash paper,” Nov.2008, https://www.mail-archive.com/cryptography@metzdowd.com/msg09959.html
(2) S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” http://www.bitcoin.org/bitcoin.pdf
(3) A. M. Antonopoulos, “Mastering bitcoin: unlocking digital cryptocurrencies,” O'Reilly Media, Inc., Sebastopol, CA, 2014.
(4) https://en.bitcoinwiki.org/wiki/Pay-to-Pubkey_Hash
(5) A. Back, “Hashcash - a denial of service counter-measure,” Aug.2002, http://www.hashcash.org/hashcash.pdf.
(6) M. Jakobsson and A. Juels, “Proofs of work and bread pudding protocols,” in Secure Information Networks, pp.258-272, Springer, Boston, MA, 1999.
(7) “Litecoin, open source P2P digital currency,” https://litecoin.org
(8) Y. Sompolinsky and A. Zohar, “Secure high-rate transaction processing in bitcoin,” in International Conference on Financial Cryptography and Data

Security, pp.507-527, Springer, Berlin, Heidelberg, 2015.
(9) I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, “Bitcoin-ng: a scalable blockchain protocol,” 13th USENIX Symposium on Networked Systems Design and Implementation, pp.45-59, 2016.
(10) E. Lombrozo, J. Lau, and P. Wuille, “Segregated witness (consensus layer) ,” https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki
(11) J. Poon and D. Thaddeus, “The bitcoin lightning network: scalable off-chain instant payments,” https://www.bitcoinlightning.com/bitcoin-lightning-network-whitepaper/
(12) C. Stoll, L. Klaasen, and U. Gallersdorfer, “The carbon footprint of bitcoin,” Joule, vol.3, no.7, pp.1647-1661, July 2019.
(13) M. J. Krause and T. Tolaymat, “Quantification of energy and carbon costs for mining cryptocurrencies,” Nature Sustainability, vol.1 no.11, pp.711-718, Nov. 2018.
(14) “Cambridge bitcoin electricity consumption index,” https://www.cbeci.org
(15) “Nxt whitepaper,” https://nxtwiki.org/wiki/Whitepaper:Nxt
(16) S. King and S. Nadal, “PPCoin: peer-to-peer crypto-currency with proof-of-stake,” https://www.peercoin.net/whitepapers/peercoin-paper.pdf
(17) “NEM - Distributed ledger technology (Blockchain) - Harvesting & poI,” https://nem.io/xem/harvesting-and-poi/
(18) K. Ikeda, “qBitcoin: a peer-to-peer quantum cash system,” Science and Information Conference, pp.763-771, Springer, Cham, July 2018.
(19) E. O. Kiktenko, N. O. Pozhar, M. N. Anufriev, A. S. Trushechkin, R. R. Yunusov, Y. V. Kurochkin, A. I. Lvovsky, and A. K. Fedorov, “Quantum-secured blockchain,” Quantum Science and Technology, vol.3 no.3, 035004, July 2018.
(20) https://www.bitcoincash.org
(21) V. Buterin, “Hard fork completed,” https://blog.ethereum.org/2016/07/20/hard-fork-completed/, July 2016.
(22) “Bitcoin improvement proposals,” https://github.com/bitcoin/bips
(23) https://lists.linuxfoundation.org/mailman/listinfo/bitcoin-dev
(24) “Scaling bitcoin workshops,” https://scalingbitcoin.org/
(25) “Devcon,” https://devcon.org/

小出俊夫（正員）

2004 創価大大学院博士後期課程了．博士（工学）．同年 NEC 入社．OpenFlow 等の分散ネットワーク制御の研究，北米での OSS 開発を経て，帰国後は IoT やブロックチェーンの研究に従事．2002 年度 C&C 若手優秀論文賞，平 16 年度本会学術奨励賞，2013 本会 ICM 研究専門委員会 ICM 研究賞，第 64 回電気科学技術奨励賞各受賞．

ブロックチェーンの合意形成アルゴリズム

白石善明　Yoshiaki Shiraishi　神戸大学
掛井将平　Shohei Kakei　名古屋工業大学

1 はじめに

ビットコイン[(1)]は，銀行のような中央管理者を介することなく，匿名でディジタル通貨を直接取引するための仕組みとして考案された．通貨の取引情報はネットワークに接続する複数の端末の合意の下で処理され，中央管理者がいない分散環境において取引の記録が正しく保管されることが必要不可欠であった．この取引情報を分散管理する仕組みをディジタル通貨以外にも利用できるように，基盤技術として切り出したものがブロックチェーンと呼ばれている．

通貨の移転の管理は，送金元の口座から送金先の口座へどれだけの通貨を移すかが記された「台帳」で行われる．例えば，図1（a）のような銀行の決済システムを考えると，台帳の各行には，「送金元口座から送金先口座へ指定通貨量だけ移動せよ」という利用者が要求する処理（トランザクション）が記載されており，そのトランザクションどおりに口座間で通貨が移動される．この台帳が改ざんできると，個人の資産を不正に操作できてしまうので，当該システムを運用する銀行が台帳を保護する役割を担っており，利用者はその銀行を信頼するモデルとなっている．

ビットコインが目指したような電子決済システムの課題の一つとして，特定の機関に頼らない台帳の保護がある．ビットコインのアイデアは，図1（b）のように，トランザクションを複数個ごとにブロックと呼ばれるデータにまとめて，そのブロックの連鎖構造であるブロックチェーンを作り，これを各利用者で共有・管理するものである．しかし，誰もが好き勝手にブロックの連鎖構造を作れると，不正なトランザクションを挿入される恐れがあるので，適切なブロックかどうかを各利用者が相互に確認して合意する「合意形成」が行われる．

ブロックチェーンは，取引情報を安全に管理する「ブロックの連鎖構造」と「合意形成」に着目して，汎用的

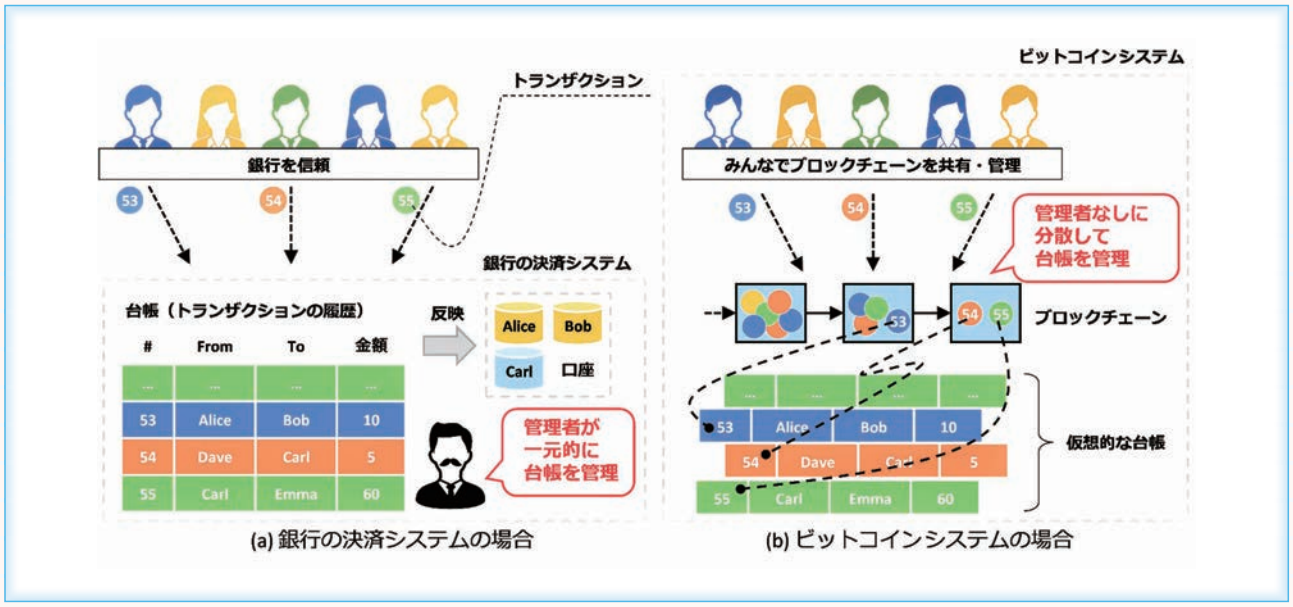

図1　銀行の決済システムとビットコインシステムにおける台帳の管理方法の違い

なデータ（ディジタルアセット）にも利用できるようにしたものである．台帳を汎用的に利用できるようになったことから，ディジタル通貨に限らない応用が期待されている．

2 ブロックチェーンの概要

ブロックチェーンはブロックの連鎖的なデータ構造を指す一方で，その周辺技術やシステムを含む用語でもある．ブロックチェーンシステムは，ブロックチェーンを共有するノード（計算機上のプロセス）の集まりから成るネットワークで構成される．図２に示すように，クライアントがトランザクションの実行を要求すると，それを受け付けたノードがその他のノードに対してトランザクションの提案を行う．各ノードはトランザクションの検証を行い，問題がなければ自身のブロックチェーンに取り込む．そして，一定量のトランザクションがたまったら，これらをブロックとしてまとめて，暗号学的ハッシュ関数を用いて直近のブロックに接続する．このハッシュ値の連鎖によりブロックの連鎖構造が形成される．

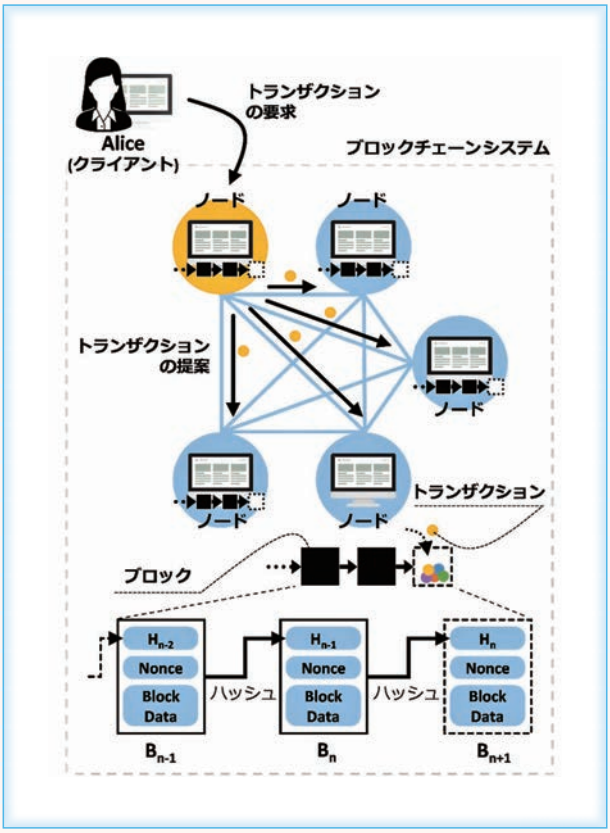

図２　ブロックチェーンの概要

3 ブロックチェーンにおける合意形成

3.1 ブロックチェーンの分岐

分散環境においては，異なるクライアントが同じタイミングでトランザクションを要求することが想定される．例えば，図３のように，Alice と Bob がそれぞれトランザクション TX_a と TX_b を要求したとする．各ノードがそれぞれの判断でブロックにトランザクションを取り込むと，ノードによって異なるトランザクションが保持され，トランザクションの履歴の分岐が発生する．

履歴が分岐すると，一方のチェーンに取り込まれたトランザクションが他方では取り込まれていないという状況が発生する．一貫性のある履歴を維持するには，どちらの分岐を正しい履歴とするかを，全ノードで合意できる仕組みが必要となる．

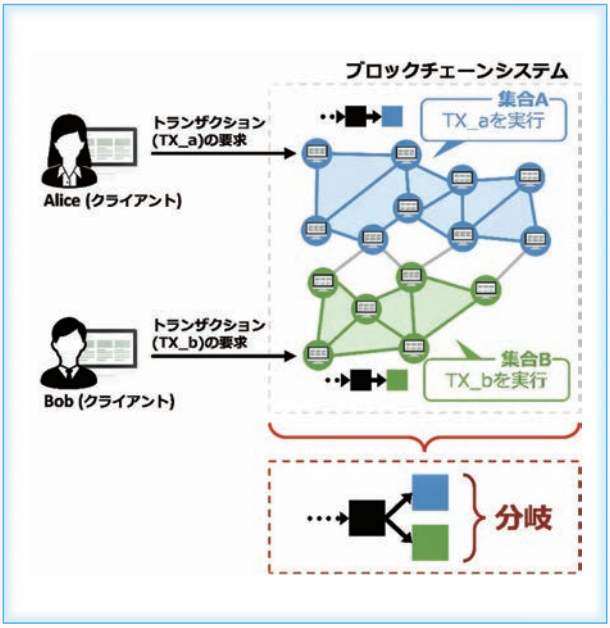

図３　トランザクション履歴の分岐

3.2 ビザンチン将軍問題

3.2.1 合意問題と分散システムにおける障害

ノード上で動作するプロセスは他のプロセスとメッセージを交換することで，合意形成に必要な情報を得る．しかし，任意のタイミングでプロセスが停止する「クラッシュ障害」や規定時間内に応答できない「タイミング障害」，任意のタイミングで任意の動作を引き起こす「ビザンチン障害」などの障害はメッセージの交換を阻害する要因となる．このような障害の中で，正常な全てのプロセスが同じ一つの値をどのように共有するかを考える問題を「合意形成問題」と呼ぶ．

ビットコインシステムのように，不特定のノードから構成されるネットワークにおいては，ビザンチン障害の発生が想定される．本障害を考慮した合意形成問題が数学者のレスリー・ランポートにより，ビザンチン将軍問題 [(2)] として形式化されている．

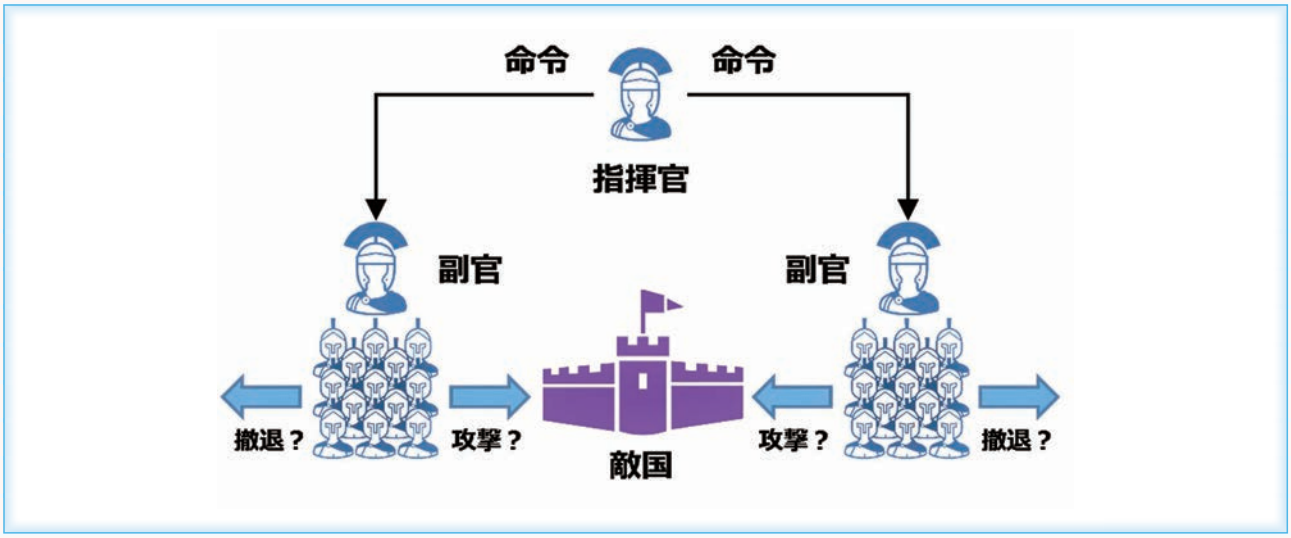

図４　ビザンチン将軍問題の状況

3.2.2　問題設定

本問題は，ビザンチン帝国（東ローマ帝国）における軍事作戦をメタファとした問題である．ビザンチン帝国軍には統率する“指揮官”と作戦遂行する“副官”の将軍たちがいて，副官がそれぞれの部隊を率いて敵国を包囲している状況（図４）を想定する．指揮官は全軍での一斉攻撃か，全軍撤退を行いたいものとする，すなわち指揮官が「攻撃」か「撤退」のどちらか一方の作戦を選択し，全ての副官に指示する状況を考える．

ここで，ビザンチン帝国軍の中には，敵国側に協力する裏切り者が存在し，作戦のかく乱を目論んでいるとする．指揮官が裏切り者であるかもしれない．指揮官が誠実であったとしても，指揮官からの命令を伝達するときに命令の不達や改ざんがあると指揮官が選択した命令を副官が実行できない．副官同士でその命令を伝え合えれば不達や改ざんを見破ることができる．しかし，裏切り者の副官が潜んでいれば，でたらめな命令が伝達されることがあり得る．

ビザンチン将軍問題では，このような状況において，以下の二つの条件を満たす体制を整える方法を考える．

条件１：全ての忠実な副官は同じ作戦を実行すること

条件２：指揮官が誠実である場合，忠実な副官は指揮官の命令を実行すること

3.2.3　多数決による解法

裏切り者にかく乱されない十分に必要な数の忠実な将軍が存在するときに，多数決によりビザンチン将軍問題を解決する方法が文献（2）に示されている．この解

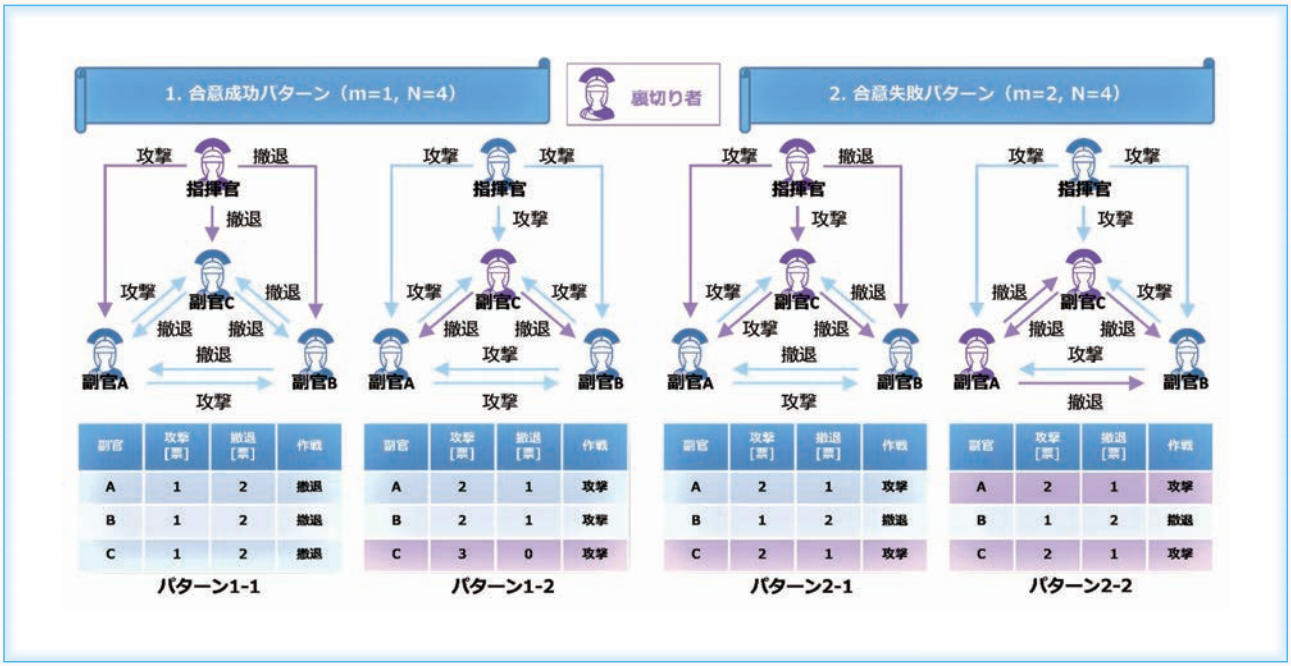

図５　多数決によるビザンチン将軍問題の解法

法では，裏切り者の人数をmとしたときに，指揮官を含む将軍の人数が$N=3m+1$だけ必要となる．

図5に合意が成功するパターン（$m=1,N=4$）と失敗するパターン（$m=2,N=4$）に関して，裏切り者に指揮官を含む場合と含まない場合の4種類を示す．合意成功パターンでは，どちらも忠実な副官は同じ作戦に合意できている．また，パターン1-2では，副官Aと副官Bが指揮官と同じ作戦に合意できている．一方で，合意失敗パターンでは，全ての忠実な副官が同じ作戦に合意できていない．また，パターン2-2では，副官Bは指揮官と異なる作戦を選択している．

3.3　分岐したブロックの選択

ビザンチン将軍問題の文脈でブロックチェーンの合意形成問題を考えると，忠実な副官がブロックチェーンのプロトコルに従う正常なノードであり，そして，裏切り者がブロックチェーンのプロトコルに従わない不正なノードとなる．これらのノードが存在する中で，指揮官が選択したブロックを正常なノードの間で合意することがブロックチェーンにおける合意形成である．公平な選択を行うには，忠実な将軍の数が重要となる．

ビザンチン将軍問題では，多数決をとるために将軍の数は既知でなければならない．しかし，ビットコインを代表とする「パブリック型」のブロックチェーンシステムは，利用者が任意に参加／離脱できる形態であるので，ノード数が既知でない．一方で，特定の組織から構成される「プライベート型」や複数の組織から構成される「コンソーシアム型」ではノード数は既知であるが，全員参加の合意形成の効率の向上が課題である．**4.** と **5.** では，それぞれのブロックチェーンシステムにおける合意形成アルゴリズムについて紹介する．

4　代表的な合意形成アルゴリズム ～パブリック型～

ビットコインシステムにおける合意形成アルゴリズムとしては，Proof of Work（PoW）と呼ばれる，ブロックの接続を確率的に成功させる仕組みが知られている．また，多くのパブリック型の合意形成ではPoWやPoWから派生したアルゴリズムが利用されている．本章では，PoWを説明した後に，PoWから派生したProof of Stake（PoS）とProof of Importance（PoI）を紹介する．

4.1　Proof of Work（PoW）

4.1.1　概要

PoW [(1)] はビットコインやイーサリアムなどのブロックチェーンシステムで採用されている合意形成アルゴリズムである．PoWは直訳すると「仕事の証明」であり，この「仕事」とは，一定の難易度の暗号パズルを解くことを指す．この暗号パズルの解がブロックを接続するための鍵となる．暗号パズルの解を発見した者には，報酬の獲得権が与えられる．そして，一般的に，最初の発見者が報酬を得られる．この権利のために暗号パズルを解くことが金の採掘になぞらえてマイニングと呼ばれる．また，このマイニングを行うノードは「マイナー」と呼ばれる．

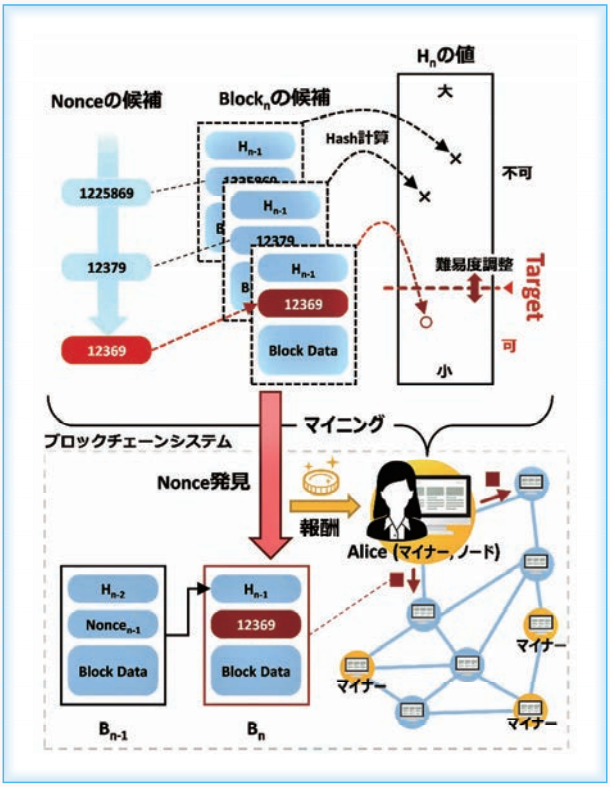

図6　Proof of Work (PoW) の概要

図6にPoWの概要を示す．各マイナーは，報酬の獲得権を得るために，B_nのマイニングを競争的に行う．B_{n-1}のハッシュ値H_{n-1}を含むB_nを作成するときに暗号パズルが解かれる．

PoWでの暗号パズルとは，ブロックのハッシュ値がTargetよりも小さくなるようなNonceを探すパズルである．安全な暗号学的ハッシュ関数が利用されていると，Targetより小さなハッシュ値を狙って計算することが困難である．つまり，マイナーはいろいろなNonceで実際にハッシュ計算を行い，Targetよりも小さなハッシュ値となるNonceを見つけ出すしかない．

4.1.2　PoWによる合意形成

PoWでは，マイニングの競争の勝者が発見したNonceを含むブロックが新たにブロックチェーンに接続される．各マイナーが競争的にブロックの接続を試み

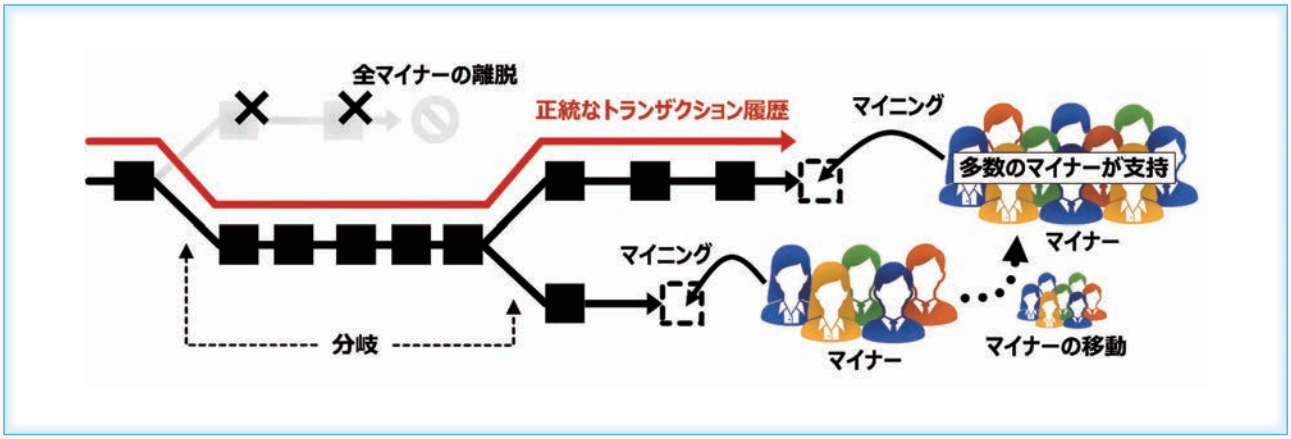

図7　PoW におけるトランザクション履歴の分岐と選択

るので，互いが知らないところで別々の Nonce が発見されることがある．これは，図7に示すようなトランザクションの履歴の分岐を意味するが，PoW では一時的な分岐は許容されている．分岐発生後も各マイナーは任意の分岐を選んでマイニングを継続する．結果的に，最も長くなったチェーンを正統なチェーンとして扱う．

長い方のチェーンを正統とする考え方は，マイニングの成功は確率的であり，ハッシュ計算の試行回数を増やすことで高まることに由来している．長く伸びたチェーンはそれだけ多くの試行が行われたと考えられ，結果的に多くのマイナーがそのチェーンに合意していたと考えられる．一方で，ハッシュ値の計算速度は CPU パワーに依存するので，特定のマイナーの CPU パワーがその他全てのマイナーの CPU パワーの過半数を超えると，大多数のマイナーの合意を無視して独自にチェーンを伸ばしていくことができる．その結果，トランザクションが二重にブロックチェーンに取り込まれることや，特定のトランザクションが意図的に無視されることが可能となる．そこで，全体の CPU パワーを増やすために，マイニングの勝者に経済的インセンティブとなる報酬を与えて，多数のノードがマイナーになる動機付けを行っている．

残念ながら正統なチェーンに含まれない短いチェーンに格納されていたトランザクションやそのときのマイニングの報酬はなかったこととされる．これにより，マイニングが徒労に終わることを避けたいマイナーが長いチェーンをマイニングするように誘引される．トランザクションが取り込まれたかどうかは，一定時間経過後にそのトランザクションを含むチェーンが最も長くになっていることを確認すればよい．例えば，ビットコインでは約1時間後となっている．

4.2　その他の合意形成アルゴリズム

PoW では，暗号パズルの難易度は全マイナーで同じであった．不正を行う動機の低いマイナーにはブロックの接続の難易度を下げ，計算資源の無駄遣いを改善するようなアルゴリズムが提案されている．

4.2.1　Proof of Stake（PoS）

PoS は直訳すると「賭け金の証明」であり，賭け金が多い者ほどブロックを接続しやすくなるような合意形成アルゴリズムである．Peercoin [3] で最初に実装されたとされ，Ethereum [4] の次期合意形成アルゴリズムとしても知られている．

PoS では，ディジタル通貨の量と保有期間に応じて算出される値（コイン年齢）を元に，ブロックの接続権の割当てが行われる．多くのディジタル通貨を持つ者ほど，ブロックの接続権を得られやすいが，そのような者は自らディジタル通貨の価値を下げるような不正を行う動機が低いという前提の上で成り立っている．

4.2.2　Proof of Importance（PoI）

PoI は直訳すると「重要度の証明」となる．ブロックチェーンシステムへの貢献度が高い者ほどブロックを接続しやすくなるような合意形成アルゴリズムであり，NEM [5] で採用されている．

PoS では，ディジタル通貨をため込むことでブロックの接続権を得られたが，このような仕組みは，富の集中を引き起こし，更には，ディジタル通貨の活発な流通を妨げる懸念がある．PoI では，ディジタル通貨の移動を活発に行っているノードがブロックを接続しやすくなっている．

5　代表的な合意形成アルゴリズム ～プライベート型／コンソーシアム型～

このタイプの特徴は，ノード数が既知であり，特定の参加者でネットワークが構成される点である．あらかじ

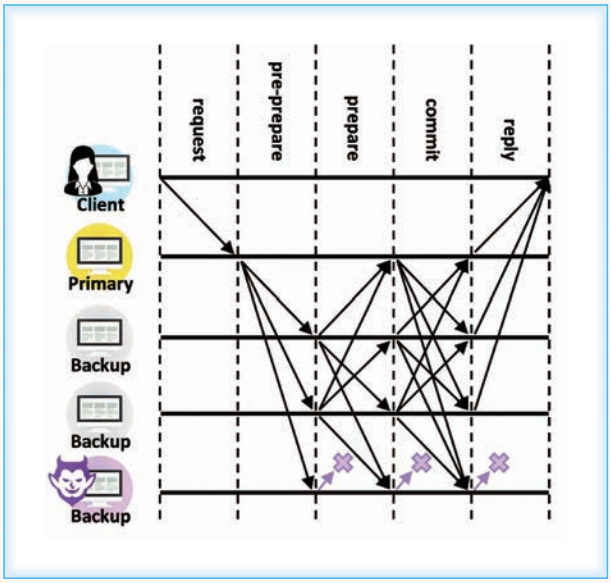

図８　Practical Byzantine Fault Tolerance（PBFT）の処理の概要（N=4,f=1）

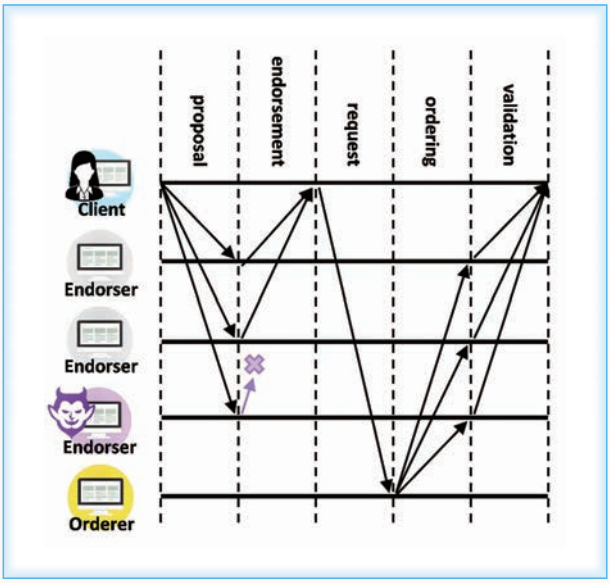

図９　Endorsement-Ordering-Validation の概要（最低承認数 =2）

めリーダを決めることで，合意形成の効率化を図る Practical Byzantine Fault Tolerance（PBFT）が知られている．本章では，PBFT を説明した後，多数決の参加者を限定する Endorsement-Ordering-Validation とノード間でリーダの選出を行う Raft を紹介する．

5.1　Practical Byzantine Fault Tolerance（PBFT）

5.1.1　概要

PBFT [6] は，1999 年に Castro らによって提案されたビザンチン障害に耐性のあるアルゴリズムである．ノード数が既知であることが前提であり，コンソーシアム型のブロックチェーンシステムである Hyperledger Fabric [7] の v0.6 で利用されていた．一つのプライマリノード（Primary）と複数のバックアップノード（Backup）で構成され，正常なノードの数を N，不正なノード数を f としたときに，N が $3f$+1 以上であればビザンチン障害に対応できるようになっている．

PBFT では，図 8 に示すように，クライアント（Client）が Primary にトランザクションを要求（request）し，Primary がその配布（pre-prepare）を行う．Backup は配布されたトランザクションの相互確認（prepare）を行い，他の Backup も Primary から同じトランザクションを受信したことを確認する．その後，他の Backup も prepare が完了したことを相互で確認した上でトランザクションの実行を行う（commit）．最後に，その実行結果をクライアントに返信（reply）する．

5.1.2　PBFT による合意形成

トランザクションの実行順序は，Primary により調整され，全ノードがトランザクションの内容に合意した上で実行されるので，パブリック型のようにブロックチェーンの分岐が発生せず，トランザクション履歴が覆ることもない．

しかし，合意形成はあらかじめ用意された既知のノードだけで行われることから，パブリック型のように誰でも合意形成に参加できるわけではない．PBFT を用いる場合，Primary と Backup を運用する主体への信頼が必要となる．また，ノード数の増加に伴い，通信量が増加するスケーラビリティの限界がある．

5.2　その他の合意形成アルゴリズム

5.2.1　Endorsement-Ordering-Validation

Endorsement-Ordering-Validation は，Hyperledger Fabric の v1.0 以降で採用されているアルゴリズムである．処理の流れは，図 9 に示すように，Client はトランザクションの提案を行う（proposal）．提案を受けたノード（Endorser）はトランザクションを実行し，その結果と結果に対する署名を Client に返信する（endorsement）．Client は承認に必要な数だけ返信を受けたら，トランザクションの順序付けを行うノード（Orderer）に Endorser の署名とともにトランザクションを要求する（ordering）．Orderer はトランザクションの順序を決定し，それをブロックに含めて Endorser に配信する．Endorser は，受け取ったブロックが最低承認数を満たすか検証し，自身のブロックチェーンに追加し，Client に応答を返す（validation）．トランザクションを実行するかどうかは，どれだけの Endorser から承認を受けるかのポリシー（図 9 における必要な承認数は 2）に依存し，ノードの全体数には無関係なため，

PBFT よりもスケーラビリティが高い．一方で，承認ポリシーの設定によっては必要な承認数を得ることが困難になることも考えられる．

5.2.2　Raft

Raft [8] は，Quorum [9] で採用されている合意形成アルゴリズムである．Raft において，ノードは Follower と Leader に分かれ，Leader の死活監視を全 Follower で行うことで，Leader が停止した際には Follower の中から新しい Leader が選出される．ブロックの作成は Leader が行い，Follower に配布される．Follower はブロックを検証し，過半数の Follower から承認が得られればブロックが確定する．

6　おわりに

パブリック型のブロックチェーンシステムでは，トランザクション履歴の分岐に対して，ブロックを接続できたときの「報酬の獲得権」という概念を作り，「最長のチェーンが正統なチェーン」というルールの下で競争する．競争が続いていく中で，ブロックチェーンの長さでどれだけのノードの合意が積み重なってきたかが示される．その結果，最長のチェーンは最も多くのノードが合意したチェーンとみなされる．

プライベート型のブロックチェーンシステムでは，合意形成に関わるノードが既知であり，トランザクションの順序を調整するような管理者の存在を前提とできる．そのため，管理者の号令の下，障害の発生していない正常なノードがトランザクションの検証を行い，その検証結果に応じてトランザクションを確定する．その結果，ブロックチェーンには全ての正常なノードが合意したトランザクションだけが含まれるので，ブロックチェーンの分岐が発生しない．

本稿では，ブロックチェーンシステムにおいて，パブリック型とプライベート型の観点から合意形成アルゴリズムをまとめた．合意する内容や合意形成の流れ，また，どのような前提の下で合意形成が可能かを知ることで，ブロックチェーン技術の理解の一助になれば幸いである．

■ 文献

(1) S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system, 2008,” https://bitcoin.org/bitcoin.pdf.
(2) L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” ACM Trans. Programming Languages and Systems (TOPLAS), vol.4,no.3,pp.382-401,July 1982.
(3) peercoin.net, “Peercoin,” https://www.peercoin.net/
(4) ethereum.org, “Ethereum,” https://ethereum.org/
(5) nem.io, “NEM,” https://nem.io/
(6) M. Castro and B. Liskov, “Practical byzantine fault tolerance,” Proc. Symposium on Operating System Design and Implementation (OSDI), USENIX, Feb.1999.
(7) hyperledger.org, “Hyperledger Fabric,” https://www.hyperledger.org/projects/fabric
(8) raft.github.io, “The raft consensus algorithm,” https://raft.github.io/
(9) goquorum.com, “Quorum,” https://www.goquorum.com/

白石善明（正員：シニア会員）

平 7 愛媛大・工・情報卒．平 9 同大大学院博士前期課程了．平 12 徳島大大学院博士後期課程了．平 14 近畿大・理工・情報・講師．平 18 名工大大学院助教授．平 25 から神戸大大学院准教授．博士（工学）．情報セキュリティ，コンピュータネットワーク，機械学習に基づくサイバー攻撃分析の研究・教育に従事．平 15 本会暗号と情報セキュリティシンポジウム（SCIS）20 周年記念賞，平 18SCIS 論文賞など各受賞．

掛井将平（正員）

平 23 岐阜大・工・応用情報卒．平 25 同大学院工学研究科応用情報学専攻博士前期課程了．平 31 神戸大大学院博士後期課程了．博士（工学）．同年名工大サイバーセキュリティセンター助教．情報セキュリティの研究に従事．

スマートコントラクト
――ブロックチェーンからなるプログラミングプラットホーム――

知念祐一郎　Yuichiro Chinen　大阪大学
芦澤奈実　Nami Ashizawa　大阪大学
矢内直人　Naoto Yanai　大阪大学
クルーズ ジェイソン ポール　Jason Paul Cruz　大阪大学

1 スマートコントラクトとは

2008年のビットコイン（Bitcoin）誕生から現在に至るまで多くのブロックチェーンが出現しているが，ブロックチェーンと聞くと暗号通貨を思い浮かべる人も多いだろう．しかし，その機能は単なる通貨の送受信以上の領域にまで広がっている．その最たるものがスマートコントラクトだ．スマートコントラクトは契約の自動執行と説明されることもあり，その様はしばしば自動販売機に例えられる[(1)]．自動販売機にお金を入れて商品を選ぶボタンを押せば，その場で売買契約が成立する．この流れを抽象化すると，送金額と商品情報という入力に対して，商品とお釣りが出力される．これをブロックチェーン上でプログラムとして実行するのがスマートコントラクトである．大雑把には，単なる契約の執行だけではなく，一連の処理を自動化することが可能であり，既存の中央集権型[(用語)]のアプリケーションを置き換えるような高いポテンシャルを秘めている．ブロックチェーンの基本的な概念は他の記事を読んで頂くとして，本稿ではスマートコントラクトの基礎から最先端の研究動向までを解説する．

2 スマートコントラクトの基礎知識

ブロックチェーンが2008年に提案され，スマートコントラクトを導入したイーサリアム（Ethereum）が登場して以降，スマートコントラクトの概念は発展を続けている．「スマートコントラクト」という言葉は，よく“契約をスムーズに行う技術”と思われがちである．これはブロックチェーンが登場するよりもはるか前の1996年にNick Szaboが執筆した「Smart Contracts: Building Blocks for Digital Markets」における定義によるところが大きい．

しかしながら，実はスマートコントラクトのデファクトスタンダードであるEthereumが契約（contract）をスマートに行える技術でも，あるいは法的に保証されるような正式な取引に限る技術でもないことを考えると，ブロックチェーンが広まった現在では，このような意味におけるスマートコントラクトの用法は誤解である．Ethereumの開発者であるGavin Woodは，スマートコントラクトは「Ethereumネットワークプロトコルの一部として，EVM（Ethereum Virtual Machine）の文脈として確定的に実行される，変更不可能なコンピュータプログラム」と定義している．詳細は後述するが，この定義からも現在のスマートコントラクトは明らかに契約に特化した技術ではないことが分かる．

表1　スマートコントラクトと暗号通貨の違い

	ブロックチェーン上の主な情報	トランザクションの内容
暗号通貨	・通貨残高	・送金
スマートコントラクト	・コード ・変数	・コード実行 ・コードの書込みと変数の初期化

スマートコントラクトはブロックチェーン上に展開されるコードであることを考えると，実は単にコンピュータプログラムと考えることができる．ブロックチェーン上に公開されるデータが改ざんできないことと同様，スマートコントラクトのコードは一度展開されると，（変更を反映するような新たなコードのインスタンスを展開することはできるが）コード自体を変更することはできない．また，スマートコントラクトの動作は実行を行うノードの時刻や環境変数に左右されない決定的なものである．つまり，コードや関数の実行は誰が実行したとしても同じ結果になるのである．このため，ネットワーク内のどのノードが検証をしたとしても同じ結果が得られることが保証できている．この事実により，誰かがスマートコントラクトを実行したという事実をブロックチェーン上で一意に共有できる（図1）．

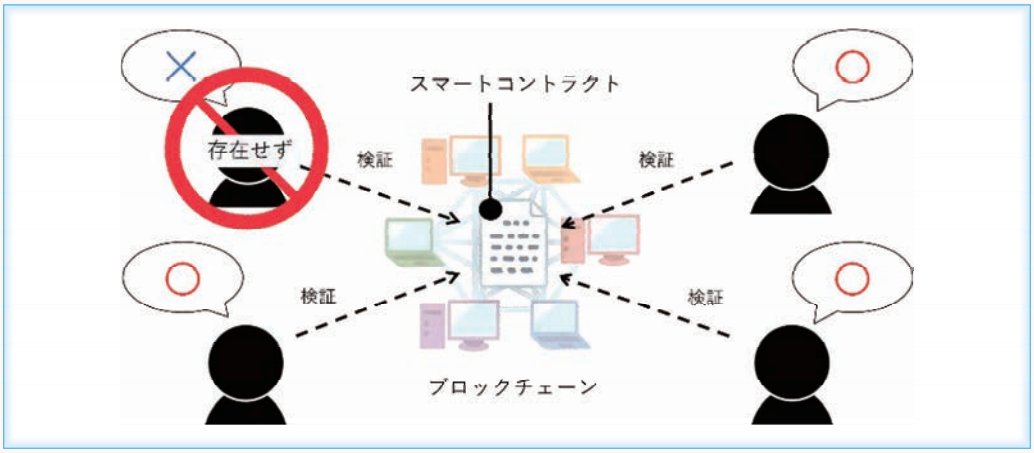

図 1　スマートコントラクトの基本原理

スマートコントラクトは Solidity のような高級言語で記述されることが多い．これらのソースコードは EVM (用語) という実行環境で動作するバイトコードにコンパイルされる．コンパイルを経てブロックチェーン上に展開されるとコントラクトはアドレスという識別子を与えられる．このアドレスを参照することで実行することができる．

3　スマートコントラクトの機能

昨今のブロックチェーンは，スマートコントラクトを暗号通貨と併せて標準的に備えている．暗号通貨の詳細な説明は他の記事で行われるため本稿では割愛するが，大雑把には暗号通貨はブロックチェーン上に通貨残高を保持し，取引処理を表すトランザクション (用語) によって送信者の残高を減らし受信者の残高を増やすことで送金を実現する．これに対しスマートコントラクトは，ブロックチェーン上にプログラムの実行コードと変数を保持する（表 1）．この実行コードと変数を，スマートコントラクトでは「コントラクト」と呼ぶ．コントラクトがスマートコントラクトにおけるプログラムの単位となる．トランザクションを発行してコントラクトを実行することで，変数の書換えや送金を行う．またブロックチェーン上へのコントラクトのデプロイ (用語) もトランザクションにより行われる．

スマートコントラクトは分散性や改ざん耐性に優れるというブロックチェーンの性質を受け継いでおり，信頼性や透明性 (用語) の高いシステムが構築できる．例えば，写真や不動産の権利といった価値をトークン (用語) として定義し送受信することで，中間者を排除しつつ不正の行えない取引基盤を構築できる [(2), (3)]．

4　スマートコントラクトの問題点

Ethereum では既に公開ネットワーク上で金融やゲームなど様々な分野のスマートコントラクトが稼動しているが，同時に問題点も露呈している．Ethereum で最も存在感のある実例は，ICO（Initial Coin Offering）だろう．ICO は新規事業を発足する際などに，株式のように独自のトークンを発行して暗号通貨の投資を募る資金調達法である．第三者を介することなく，広く早く資金を集められるため，Ethereum 上で多くの ICO が行われている．しかし，詐欺 ICO の頻発や法律上の扱いが問題視されたことから，現在は STO（Security Token Offering）という有価証券としてトークンを発行することで厳密に法規制を満足する形に移行している．

また，システム自体の問題点として，セキュリティやプライバシーを担保する難しさがある（図 2）．例えば Ethereum ではプログラム実行時には gas と呼ばれる「プログラム実行用の燃料」が存在することに加え，実行時の挙動や安全性の性質が従来のプログラミング言語と大きく異なる．このため，「開発者目線」では従来のプログラミング言語の知見が使いにくい．更に，ブロックチェーンの透明性によりプログラムのバイトコードが誰でも閲覧可能となってしまうため，プログラムそのものの解析が攻撃者にとって実施しやすい．このような特殊性に加え，元がブロックチェーンという金銭的価値のあるデータを取り扱うことが多いため，悪意のあるユーザに攻撃されるなどサイバー犯罪も起きやすい．例えば 2016 年 6 月に起きた The DAO 事件ではスマートコントラクトのぜい弱性を踏み台にすることで 60 億円相当，2017 年のパリティ多重署名ウォレット攻撃では 150 億円相当の暗号通貨 Ether がそれぞれ盗まれている．また，従来のプログラミング言語との重大な違いとして，

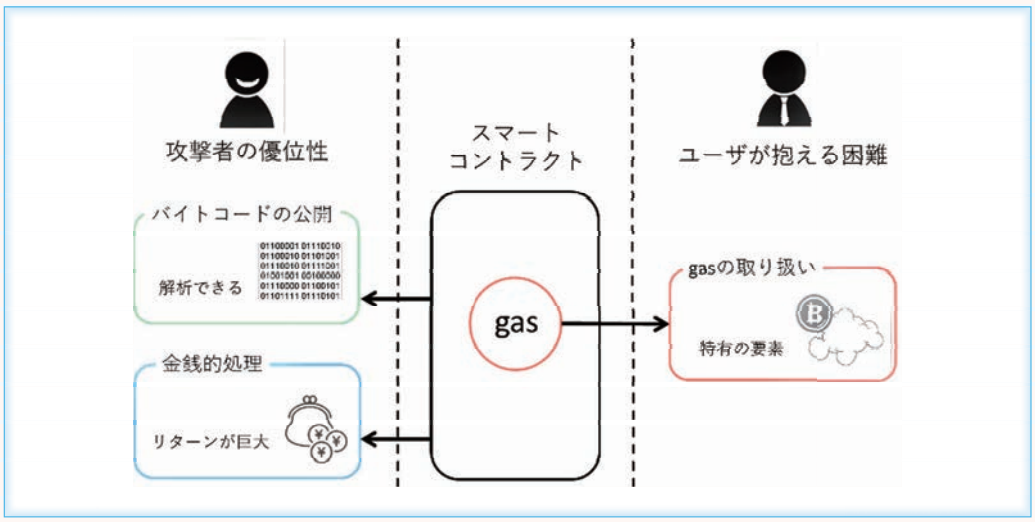

図２　スマートコントラクトのセキュリティ上の問題

一度利用を開始したスマートコントラクトは変更不可能となる点がある．このため，ぜい弱性を持つコードを利用してしまった場合は，長期的に踏み台にされるなど取返しがつかない事態も誘発しかねない．

4.1　リエントランシー

The DAO 事件で悪用されたリエントランシーというぜい弱性を，文献（4）の内容を基に紹介する．これはfallback 関数と呼ばれる Solidity 特有の無名関数が通貨の送金等により実行される性質を通じて，プログラムの実行中に不正に再度そのプログラムが呼び出されるというものである．ソースコード１（図３）に被害者となるリエントランシーを含むコントラクト Fund を，ソースコード２（図４）に攻撃者であるコントラクト Attacker を示す．Fund は銀行のような機能を想定しており，通貨の預入れや引出しができる．コントラクト Attacker が Fund に対し関数 attack で自身が持つ通貨を送信する（ソースコード２の４行目に相当）と，Fund が持つ key-value ストアである shares（ソースコード１の２行目に相当）に Attacker のアドレスに対応する預金額が記録される．その後 Attacker が自身の預けた通貨を引き出す（ソースコード２の５行目に相当）と，Fund の関数 withdraw で shares に記録された額が Attacker に送信される（ソースコード１の４行目に相当）．ここで Attacker の fallback 関数が実行され，再び Fund の withdraw 関数が呼ばれる（ソースコード２の８行目に相当）．Fund は通貨の送信後に預金額の記録をリセットする（ソースコード１の６行目に相当）が，再び withdraw 関数が呼ばれる時点では実行されていないため，初めと同じ額の通貨が送信されてしまう．これにより，Fund が保有する通貨が全て Attacker に送金されるか，トランザクションが消費する gas 量が制限に達するまでこの「再帰呼出し」は継続される．

```
1  contract Fund {
2    mapping(address => uint) shares;
3    function withdraw() public {
4        (bool success,) = msg.sender.call.value(
             shares[msg.sender])("");
5        if (success)
6            shares[msg.sender] = 0;
7    }
8  }
```

図３　ソースコード１　被害者コードの一部

```
1  contract Attacker {
2    function attack(address target) public
         payable{
3        fund = Fund(target);
4        fund.deposit.value(address(this).balance
             )();
5        fund.withdraw();
6    }
7    function() external payable{
8        fund.withdraw();
9    }
10 }
```

図４　ソースコード２　攻撃コードの一部
図３、図４とも文献（4）から引用

最終的に The DAO 事件では不可逆であるはずのブロックチェーンを巻き戻し，攻撃をなかったことにするという非中央集権性を揺るがす決定が下された．

5　スマートコントラクトの応用技術

スマートコントラクトの利用は，取引の非中央集権化，支払い機能の自動化，公開した取引データの不変

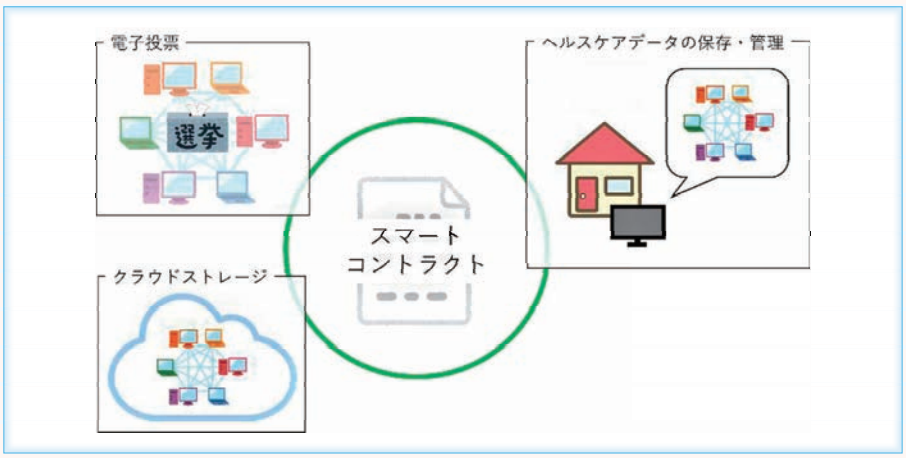

図５　スマートコントラクトの応用イメージ

性，セキュリティとプライバシー，取引の透明性など様々な利点を提供することができる．例えばスマートコントラクトは Internet of Things（IoT），クラウドストレージ，ヘルスケア，電子投票やディジタルコンテンツの著作権管理など，多数の応用が注目されている（図5）．以下にこれらについて紹介する．

(1) IoT

IoT 機器の利用においては，機器が生成するデータのセキュリティとプライバシーは保護されなければならない．スマートコントラクトを使うことにより，センサからのデータ収集や決済機能の提供，IoT 機器自体へのアクセス制御など様々な機能を提供できる．例えば，スマートホームなどで宅内に設置された IoT 機器内の全データをブロックチェーンに保存・呼び出すことで，サービス機能の改善や自動集金が可能となる．

(2) クラウドストレージ

Dropbox や Google Drive，iCloud のような従来のストレージサービスはしばしば高価である一方，データの漏えいや単一障害点などの懸念もある．このとき，スマートコントラクトの利用により，分散化されたストレージへのデータの保存とストレージからのデータの取得を便利に行えるようになる．これにより，障害への耐性向上が期待できる．つまり，ストレージサービスにスマートコントラクトを適用することで，安全性や可用性を改善することができる．

(3) ヘルスケアサービス

ヘルスケアサービスの利用においてもスマートコントラクトは有益である．一般にヘルスケアサービスにはプライバシー保護への懸念や，機器・サービス間の相互運用，データの整合性などの問題がある．スマートコントラクトを利用することで，ヘルスケアデータをブロックチェーン上に暗号化した状態で保存することができるようになる．このとき，データへのアクセス権は正当な権利を持つ利用者に制限されるとともに，従来のシステムでは自動化が難しかったような様々な制御が行えるようになる．例えば処方箋の発行，検査結果の保管や保険の管理などである．

(4) 電子投票

従来の紙媒体を用いる物理的な投票システムでは投票内容ののぞき見や，無効票の投票，投票内容の強要や結果の改ざんなど，多くのぜい弱性が考えられる．これに対し，スマートコントラクトを用いることで，投票者の情報及び投票内容を安全かつ有効にすることができるようになる．更に，投票結果の集計は即時に自動的に行われる．このため，不正が行われるような可能性も抑えることができる．

(5) ディジタルコンテンツの著作権管理

ディジタルコンテンツではコンテンツの複製や偽造への対策が必要である．スマートコントラクトを使うことで，コンテンツの所有者と消費者両方の観点からコンテンツの正しさを保証することが可能となる．まず，所有者においてはコンテンツが売買・共有されるときの条件を設定でき，再配布や非合法な利用を防ぐことができる．また，消費者の観点からは，コンテンツが正当な手続きで購入したものであることが保証される．加えて，特許や知的財産などの電子文書もブロックチェーン上に保存することで，正当な権利の所有者だけがその所有権を明らかにできるようになる．これにより海賊版など権利の侵害を防ぐことが期待できる．

6　スマートコントラクトのプラットホーム

スマートコントラクトのプラットホームとしては，前述したとおり Bitcoin に次ぐ時価総額を誇る Ethereum がデファクトスタンダードである．また，近年では

Hyperledger Fabric が急速に広まっている．以下にそれぞれについて説明する．

(1) Ethereum

Ethereum は非中央集権的なコミュニティによって開発されるブロックチェーンである．不特定多数の参加者からなる P2P ネットワーク上で構築され，ユーザが誰にも管理されずに自身のデータを保有し公平に利用できるインターネット基盤を目指している．Ethereum におけるコントラクトはクラスのように関数と変数を持ち，Ethereum ネットワークを構築するためのクライアントソフトを用いてブロックチェーン上にデプロイされる．デプロイされたコントラクトに対しトランザクションを発行することでコントラクトを実行し，通貨の送金や変数の書込みを行う．プログラム実行には gas と呼ばれる暗号通貨で購入する燃料が必要であり，これがスマートコントラクトの取引手数料となる（図 6）．Ethereum では誰もが自由にトークンを発行し，ウォレット(用語)や取引所に流通させることができる．このように様々な価値をトークンとして表現し送受信することで形成される経済圏をトークンエコノミーという．

(2) Hyperledger Fabric

Ethereum が公開されたコミュニティに支えられているのに対し，特定の組織内など比較的閉じた環境での運用を目的としてよく利用されるのが Hyperledger Fabric である．これは Linux Foundation 傘下の Hyperledger Project が推進するブロックチェーンである．IBM を筆頭に多くの企業が参画しており，エンタープライズ向けのブロックチェーン基盤の構築を目指している．Hyperledger Fabric におけるスマートコントラクトはチェーンコードと呼ばれ，Go, Node.js, Java で記述することができる．公開環境である Ethereum と比べて，特定の組織間でネットワークが構成され参加も許可制となる Hyperledger Fabric は厳格なアカウント管理やアクセス制御が可能であり，処理可能なトランザクション量も多い．

(3) Ethereum と Hyperledger Fabric の相互立ち位置

一見すると Ethereum と Hyperledger Fabric の両者は設計思想も機能も異なるように思えるが，少しずつその距離は縮んできている．Ethereum は EEA（Enterprise Ethereum Alliance）という企業連合の下，Quorum という特定組織向けのブロックチェーンの開発を進めている．一方，Hyperledger Fabric においても Hyperledger Project の一つである Hyperledger Burrow は，EVM インタプリタを実装し，Hyperledger Fabric 上で Ethereum スマートコントラクトの実行を可能にしている．更に EEA と Hyperledger Project はメンバーシップ提携を結んでおり，両者は協力してビジネス向けブロックチェーン技術の推進を目指している．

とはいえ現状は Ethereum の方がインターネット上にある情報が圧倒的に多く，初めてスマートコントラクトに触れる場合はこちらを利用することになるだろう．Ethereum の公開ネットワークは取引手数料が高く気軽にスマートコントラクトをデプロイできるとは言い難いが，幾つかあるテストネットの利用や，ローカル環境にネットワークを構築する，あるいはブラウザ用の Solidity の統合開発環境を用いることで容易にスマートコントラクトを体験できる．

7 研究動向の最前線

スマートコントラクトの研究領域は，大きく二つある．まずはスマートコントラクトそのもののセキュリティに関する研究である．これは The Dao 事件といったサイバー攻撃による不正送金などを防ぐ狙いがある．次にスマートコントラクトにおけるプライバシーに関する研究である．スマートコントラクトの実行内容は全

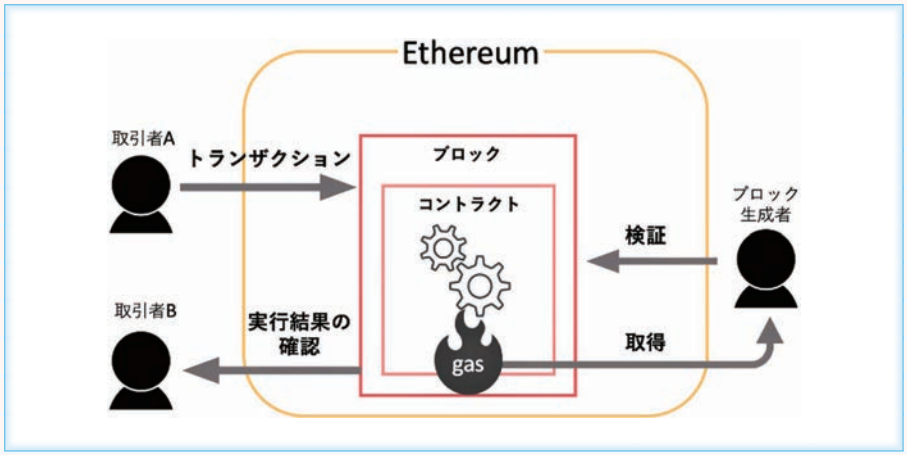

図 6　Ethereum における動作原理の直観

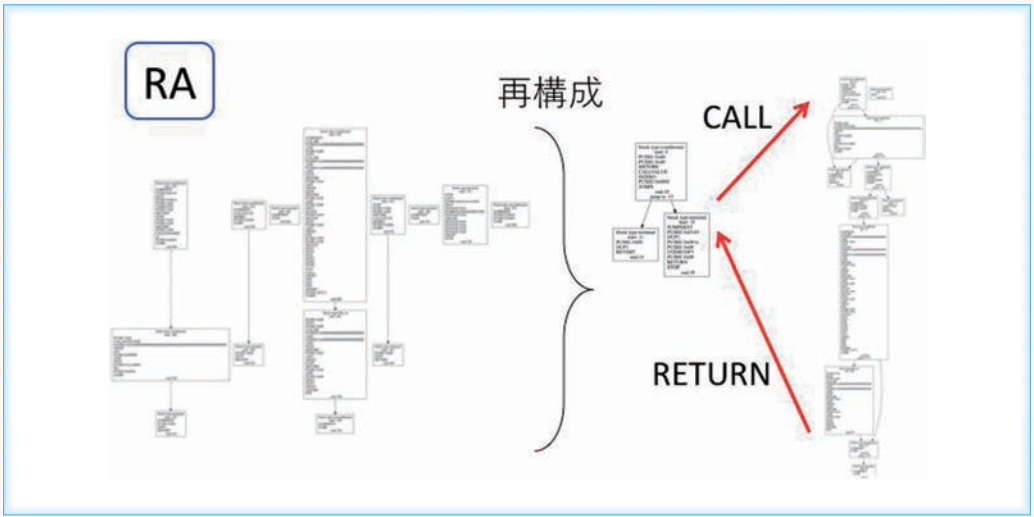

図 7　RA [4] における解析イメージ

ノードに公開されるため，医療データなどを扱う際は注意が必要となる．以下にそれぞれ説明する．

(1) セキュリティとしての研究

4. でも述べたとおり，コードにぜい弱性があった場合，大規模な被害が長期にわたって引き起こされる可能性がある．このため，コードのぜい弱性をいかに解析できるかという研究が盛んである．なお，筆者の知る限り大多数の研究は Ethereum を対象に行われている．

一般にソースコードからプログラムを解析する方法として，コード内の命令からプログラムの制御の流れを抽出するシンボリック実行と，コードが仕様を満たすか述語論理から確認する形式検証がある．いずれの内容においても外部関数の呼出しを通じて解析を回避する攻撃 [5] が 2019 年に発見され，その攻撃をいかに解析できるかが最新の動向となっている．シンボリック実行では Oyente [6] が有名であり，Ethereum 上で多くのぜい弱性を発見した実績を持つ．ソースコードが公開されており，最新の研究にも Oyente の拡張に注力しているものが多い．筆者らの知る限り，本記事を執筆している 2019 年 11 月時点では，文献 (5) の攻撃を解析できる可能性を持つシンボリック実行ツールは，筆者らが 2019 年 10 月に発表した RA [4] か，Fraunhofer AISEC の研究者らが 2019 年 9 月に発表した Annotary [7] のみである．前者は EVM のバイトコードで，後者はコンパイル前の Solidity で解析を行う．今後はこれらの更なる研究開発が期待される．

参考として，図 7 に RA による解析のイメージ図を添付する．従来の解析ツールでは，複雑な攻撃に対し，図中・左側の断片化された結果しか表示されない．これに対し，RA では攻撃の内容を一連のグラフとして表現することで，攻撃の正確なプロセスや前後の条件を解析することが可能となる．

次に，形式検証の代表的な成果として EVM を形式化した KEVM [8] があるが，具体的な安全性解析は行えていない．それ以外の形式検証による成果も文献 (5) の攻撃に関する解析は行えておらず，シンボリック実行ほど研究が進んでいない．

(2) プライバシーとしての研究

ユーザ観点におけるスマートコントラクトのもう一つの重要な課題はプライバシーである．ブロックチェーンでは各ノードによってトランザクションが検証可能な形で処理されることから，例えば医療データや投票システムにブロックチェーンを適用した際に，どの利用者がどのようなデータを提示したのか明らかとなってしまう．

データの利用状況を秘匿する機構として，ブロックチェーンのプラットホームを拡張する研究が盛んである．代表的なアプローチとしてはプログラムの正しさ及びデータの正しさと秘匿性を保証する Trusted Execution Environment (TEE) を用いる方法 [9] と，入力を隠したまま出力を計算する秘密計算を用いる方法 [10] がある．前者の研究では任意のコードを実行できるが，TEE のぜい弱性に安全性が起因してしまう．一方，後者の研究ではループ処理など無制限の実行状態が議論されていない．

8　スマートコントラクトのこれから

スマートコントラクトは新しいプログラミングの領域であり，筆者らの所感としてはインパクトの大きいサービスを模索しているのが現状である．また，セキュリティ及びプライバシーの問題も多く，本稿で述べたような技術的な問題を一つずつ潰していくことがまずは重

要である．

また，今後ブロックチェーン自体のパフォーマンスや利便性の向上に従い，スマートコントラクトもより便利な基盤となっていくことが予想される．その最たる例がFacebook による Libra である．これは Facebook がけん引し多くの企業や銀行が参画する Libra 協会が作る予定のブロックチェーンであり，スマートコントラクト記述言語として Move を備えている．Move は関数型言語を採用しており，データとロジックを分離することでより安全な機構を目指すことができる．従来のプラットホームである Ethereum も，シャーディングというトランザクションの検証を並列化する技術により，トランザクションの処理量の増加と，それに伴う取引手数料の低下が期待できる．また，複雑な EVM の代替として，eWASM というブラウザ上で実行可能な言語である WebAssembly への移行が予定されており，プログラム実行のパフォーマンス向上や開発が容易になることも期待できる．

本稿の結びとして，ブロックチェーン及びスマートコントラクトは高い可能性を秘めた技術であることを述べたい．はっきり言えることとして，既存のいつ終了するか分からないサービスと違い，世界中でどこか一つでもノードが稼動していればブロックチェーンが維持されることが大きい．このため，本稿で述べたような技術的課題が解決した暁には，新しい情報技術やインターネットの世界が開けていくと期待している．

■用語解説

・中央集権型

クライアントサーバモデルのように，特定の特権を持つ管理者が存在する仕組み．中央集権型のサービスの利用者は，管理者が不正を行わないことや正常にサービスの提供を維持することを信頼する必要がある．これに対して非中央集権型若しくは分散型のサービスでは，利用者が特定の管理者を信頼することなくサービスを享受できる．

・EVM

Ethereum Virtual Machine の略で，Ethereum スマートコントラクトの実行環境のこと．実行されるバイトコードは機械語に近いが，命令によってはブロックチェーン上の情報を参照したり，トランザクションを発行するような複雑な動作があり，安全性解析を難しくする要因の一つとなっている．

・トランザクション

ブロックチェーンへの書込み処理．トランザクションがブロック生成者によって検証され，ブロックに取り込まれることによって初めてその書込みが不可逆で改ざん困難なものとなる．暗号通貨の文脈では取引情報とも説明されるが，それは送金したという事実をブロックチェーンへ書き込むということを意味する．スマートコントラクトでは，送金したという事実，あるいはコントラクトを実行して変数を操作したという事実をブロックチェーンへ書き込むことを意味する．

・デプロイ

コントラクトの実体をブロックチェーン上に生成すること．同時にコンストラクタによってコントラクトが持つ変数が初期化される．

・透明性

情報が扱われる過程が確認しやすいことを意味する指標．ブロックチェーンの参加者であればブロックチェーン上の全てのデータを過去に遡って閲覧可能であり，特にスマートコントラクトでは誰がどれだけの資産を保有しているか，どのようなロジックにより処理されるかが全て公開情報となるため公平で平等なやり取りが可能となる．

・トークン

スマートコントラクトによって定義される電子的な資産．用途に応じた規格が存在し，規格に準じたトークンを発行することで，ウォレットソフトなどで気軽に送受信することができる．最も広く用いられる規格である ERC20 は代替可能な価値をやり取りするために定められた．例えば，ERC20 に準拠したトークンを記述することで，誰もが自分の通貨やポイントを発行することができる．ほかにも，ERC20 の問題点を解決した ERC223，知的財産のように代替不可能な価値を表現する ERC721，電子的な証券であるセキュリティトークンを発行するための ERC1400 等がある．

・ウォレット

ここでは，秘密鍵を使って暗号通貨を送受信するためのアプリケーションのこと．秘密鍵を保管する場所を指す場合もある．著名なウォレットアプリである MetaMask は，トークンの規格である ERC20 や ERC721 に対応しており，Chrome の拡張機能やスマートフォンアプリとして利用できる．

■ 文献

(1) スマートコントラクトとは【bitFlyer（ビットフライヤー）】, https://bitflyer.com/ja-jp/glossary/smartcotract
(2) KODAKOne, https://www.kodakone.com/
(3) Harbor, https://harbor.com/
(4) 知念祐一郎，矢内直人，ジェイソン ポール クルーズ，岡村真吾，“RA: スマートコントラクトの安全性解析にむけたシンボリック実行ツール，” 情報処理学会 CSS2019 予稿集，pp.569-576, Oct.2019.
(5) M. Rodler, W. Li, G. O. Karame, and L. Davi, “Sereum: Protecting existing smart contracts against re-entrancy attacks,” Proc. NDSS 2019, Internet Society, Feb.2019.
(6) L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter,” Proc. ACM CCS 2016, pp.254-269, Oct.2016.
(7) K. Weiss and J. Schutte, “Annotary: A concolic

execution system for developing secure smart contracts," Proc. ESORICS 2019, LNCS 11735, pp.747-766, Sept.2019.
(8) E. Hildenbrandt, M. Saxena, N. Rodrigues, X. Zhu, P. Daian, D. Guth, B. Moore, D. Park, Y. Zhang, A. Stefanesc,and G. Roşu, "KEVM: A complete formal semantics of the ethereum virtual machine," Proc. IEEE CSF 2018, pp.204-217, July 2018.
(9) R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller,and D. Song, "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contract execution," Proc. IEEE EuroS&P 2019, pp.185-200, June 2019.
(10) S. Bowe, A. Chiesa, M. Green, I. Miers, P. Mishra,and H. Wu, "Zexe: enabling decentralized private computation," 2018. https://eprint.iacr.org/2018/962.

知念祐一郎

平 30 阪大・基礎工を同大学院・情報科学入学のため中途退学 . 現在は同大学院情報科学研究科・博士前期課程に在学 . 専門はブロックチェーン . 令元マルウェア解析コンテスト MWSCup 2019 にチーム F.SE として優勝 , 及びスマートコントラクトの安全性解析に関する論文で CSS 奨励賞受賞 .

芦澤奈実

平 31 阪大・基礎工を同大学院・情報科学入学のため中途退学 . 現在は同大学院情報科学研究科・博士前期課程に在学 . 専門はプログラム解析 .

矢内直人（正員）

平 21 一関高専・専攻科・生産工学専攻卒 . 平 26 筑波大大学院・システム情報工学・博士後期課程了 . 博士（工学）. 同年から阪大・大学院情報科学・助教 . 専門は情報セキュリティ . 平 27 からプログラムの安全性解析に関する研究として JST 事業 ACT-I に採択 . 情報セキュリティに関する実践的教育にも従事 . 平 26 SCIS 論文賞 , 平 27CSS 論文賞 , 平 29 大阪大学賞 , 平 30CSS 奨励賞 , 令元 CSS 奨励賞など各受賞 .

クルーズ ジェイソン ポール

平 21 Ateneo de Manila University, Philippines・Bachelor of Science in Electronics and Communications Engineering 卒 . 平 23 同大学院・Master of Science in Electronics Engineering 了 . 平 29 奈良先端大・情報科学研究科・博士後期課程了 . 博士（工学）. 同年から阪大・大学院情報科学・特任助教 . 専門は情報セキュリティ . 令元スマートコントラクトの安全性解析及びブロックチェーンのアルゴリズムに関する論文で CSS 奨励賞受賞 .

暗号学的ハッシュ関数と電子署名

花岡悟一郎　Goichiro Hanaoka　産業技術総合研究所

1 はじめに

ビットコインなどのブロックチェーンに基づく仮想通貨は，暗号資産（crypto assets）と総称されている．一般的に暗号技術とは，狭義には，機密データを秘匿するための手段を指すことが多いが，暗号資産においては，それら以外の広義の暗号技術に含まれる暗号学的ハッシュ関数や電子署名が重要な役割を担っている．本稿では，これらの技術の機能や使われ方についての簡単な解説を行うものとする．まず，**2.** において広義の暗号技術を俯瞰し，**3.** において暗号学的ハッシュ関数について，**4.** において電子署名についてそれぞれ解説する．また，発展的な内容として，**5.** において暗号資産への適用が検討されている電子署名の拡張技術について紹介する．

2 広義の暗号技術

上述のとおり，我々の社会生活において暗号技術とは，機密データを秘匿化し，盗聴者に対して安全に情報の通信や保管を行うための技術を指すことが多い．しかしながら，1970 年代以降，そのような狭義の暗号技術の根底にある概念や機能に基づき，機密データの秘匿化の範疇を超えた他の有用な技術が創出され続けている．このような一連の技術が広義の暗号技術に当たる．

「狭義の暗号技術」に基づく，機密データの秘匿を主たる目的とはしていない「広義の暗号技術」の簡単な例として，例えば次のようなものが挙げられる：

送信者 S と受信者 R が安全でない通信路を介してデータの通信を行う場合を考える．送信者 S は，データを盗聴されても構わないが，他者になりすましやデータの改ざんをされることを防ぎたいものする．このような場合，送信者 S は受信者 R との間に，あらかじめ「狭義の暗号技術」における鍵情報 K を共有しておき，送信者 S は鍵情報 K を用いて送信データ M の暗号化を行い，得られた暗号文 C を送信データ M とともに，受信者 R に送信する．C と M の組を受け取った受信者 R は暗号文 C に対して鍵情報 K を用いて復号し，復号結果が M であれば，送信データ M の送信者は確かに送信者 S であり，また，通信路上で改ざんがなかったものと納得する．この例から分かるように，狭義の暗号技術に基づき，データの秘匿化以外の機能（なりすまし・改ざんの防止）が提供できている．

重要な注意：上記の「広義の暗号技術」の例を安全に実行するためには，基盤となる「狭義の暗号技術」について頑強性（non-malleability）などの性質を要求する必要がある．ただし，本稿は暗号技術になじみのない一般的な読者の理解を促すことを主眼においたものであるため，厳密な議論よりも直観的な説明を優先する．したがって，本稿で紹介されている一連の技術を実際に利用することを考えている読者については，本稿はあくまで導入としての活用にとどめ，より厳密な記述がなされた専門書を参考とすることを強く推奨する．

3 暗号学的ハッシュ関数

前章においては，狭義の暗号技術に基づき，データの秘匿化以外の機能を提供可能な広義の暗号技術を構成可能であることを紹介した．ただし，広義の暗号技術を構成する際に，必ずしも狭義の暗号技術を構成要素として明示的に使用するとは限らない．現代暗号理論の研究の過程において技術が洗練され，狭義の暗号技術の設計理念を活用しつつも，これらを陽には用いることなく，目的とする機能を直接的に提供可能な基本技術の設計が数多くなされている．

そのような広義の暗号技術のうち代表的なものの一つが暗号学的ハッシュ関数 [1] である．暗号学的ハッシュ関数とは，おおむね次のような性質を満たすもので

ある：

①任意長の入力データを（短い）固定長の出力データ（ハッシュ値）に変換する．

②与えられたハッシュ値から，それに対応する入力データを導出することができない．

③同じハッシュ値に変換されるような複数の入力データの組を見つけることができない．

暗号学的ハッシュ関数を用いることで，例えば，前章において取り上げたなりすまし・改ざん防止機能を，より洗練された形で実現することができる：

送信者 S は受信者 R との間に，あらかじめ乱数 K を共有しておき，送信者 S は乱数 K と送信データ M を連結したものを入力データとして暗号学的ハッシュ関数に入力し，得られたハッシュ値 H を送信データ M とともに，受信者 R に送信する．H と M の組を受け取った受信者 R は乱数 K と受信データ M から上記と同様にしてハッシュ値を導出し，これが受信した H と一致すれば，送信データ M の送信者は確かに送信者 S であり，また，通信路上で改ざんがなかったものと納得する．

暗号学的ハッシュ関数には，上記のほかに幅広い用途があり，より上位の様々な（広義の）暗号技術の要素技術となっている．特に，ブロックチェーン関連技術においては，過去の一連のデータ履歴に改ざんがないことを保証する中核的な役割を果たしている．ブロックチェーン上においてデータはブロックと呼ばれる単位に分割され，これらのブロックが時系列に沿って記録される．ここで，ブロックの順序の入れ替えや，過去のブロックの内容の改ざんを防ぐ目的で，暗号学的ハッシュ関数がおおむね以下のように利用される：

最初のブロック B_0 に関しては，そのままそれを入力として得られるハッシュ値を求め，これを H_0 としてブロックチェーン上に記録する．n 番目（$n \geq 1$）のブロック B_n に対しては，H_{n-1} と B_n を連結した値を入力として得られるハッシュ値 H_n を求めて，同様に，ブロックチェーン上に記録する．このとき，ハッシュ値 H_n の正当性について合意がなされていれば，過去のすべてのブロックの正当性についても自動的に合意がなされたことになる．すなわち，暗号学的ハッシュ関数の前述の③の性質により，H_{n-1} と B_n を連結した値以外に，H_n をハッシュ値とする入力を見つけることはできないことが保証されている．そのため，H_n の正当性について合意がなされているのであれば，H_{n-1} と B_n の正当性についても合意がなされているものとみなすことができる．H_{n-1} は，H_{n-2} と B_{n-1} を連結した値を入力としたハッシュ値であり，また，それ以外に H_{n-1} をハッシュ値としてとる入力を見つけることは困難であることから，H_{n-1} の正当性についての合意がなされているのであれば H_{n-2} と B_{n-1} の正当性についても合意がなされているものとみなすことができる．このようにして，ブロック B_n とハッシュ値 H_n の正当性さえ確認できれば，過去の全てのブロックの正当性についても確認が可能となる．また，その際，それらのブロックの順序（履歴）についても，入れ替えがないことが保証できる．このように，暗号学的ハッシュ関数を用いて一連のブロックを安全に連結する手法をブロックチェーンという．

暗号学的ハッシュ関数の用途は広く，後述の電子署名においても重要な役割を果たしている．

4 電子署名

暗号資産とは，大雑把には，ブロックチェーン上に全利用者の全ての取引情報を記載していくことで，各利用者の資産情報を全利用者で安全に共有することを可能とする技術や，それにより各利用者が保持する電子的な資産を指す．上記のような取引情報は，支払人が受取人と支払金額を指定することで生成がなされる．しかしながら，単に，支払人，受取人，支払金額のみが記載される場合，この情報が支払人の意思によるものか確認することができないため，他者の暗号資産を不正に侵害することが可能となってしまう．そのため，取引情報には支払人，受取人，支払金額のほかに，当該取引情報が確かに支払人の意思により生成されたものであることが確認できるようにするための付加的な情報も必要となる．そのような要求に応える技術が電子署名である．

電子署名とは，文字どおり，手書きによる署名の電子的なアナロジーであり，電子データに対して署名者にしか生成できない付加的なデータを加える技術や，そのようなデータそのものを指す．通常の手書き署名は，物理的な文書上に署名者が署名を施し，これを受け取った受信者は当該署名者の過去の署名との同一性を照合することで，当該文書に偽造やすり替えがないことを確認する．このような仕組みが可能となるのは物理的な文書と手書き署名が不可分だからであり，このような性質を電子的に実現することは必ずしも自明ではない．電子署名の最も基本的な構成方法の例を次に示す．

まず，一方向性関数 $F(x)$ を準備する．一方向性関数とは，ランダムに選ばれた入力 X について，$F(X)$ から X を復元することが困難な関数を指す．例えば，前章で紹介した暗号学的ハッシュ関数を用いてもよい．次に署名者は事前に二つの乱数 X_0 と X_1 を選び，また，$F(X_0)$ と $F(X_1)$ を計算する．署名者は X_0 と X_1 の組を安全に保管し，一方，$F(X_0)$ と $F(X_1)$ の組を公開する．手書

き署名の場合は，過去の署名と照合することにより署名の正当性の確認を行ったが，電子署名では $F(X_0)$ と $F(X_1)$ を用いて照合を行うことにより正当性の確認がなされる．これらの情報を用いて，以下のように電子署名の生成がなされる．

簡単のため，署名の対象となる電子データは 1 ビットの情報，すなわち，0 か 1 とする．署名者は，データ 0 に署名を施す場合，0 の電子署名として X_0 を発行する．検証者は，電子署名 X_0 を一方向性関数 $F(x)$ に入力して得られた値を，公開されている $F(X_0)$ と照合し，これらが一致した場合は X_0 を署名者のデータ 0 に対する正しい署名として受理する．公開されている $F(X_0)$ と $F(X_1)$ の組が確かに署名者のものであると確認されていることを前提にすると，これらの値の逆像のいずれか，すなわち，X_0 若しくは X_1 を発行することができるのは署名者のみであることが分かる．また，上記の場合では，X_0 が発行されていることから，署名者が特にデータ 0 に対して意思を表示していることも確認できる．したがって，X_0 は署名者のデータ 0 に対する電子的な署名になっていることが分かる．

上記の電子署名の構成方法の例では，1 ビットのデータに対する署名しか発行ができないため，実際の暗号資産においては ECDSA 方式 [2] などのより洗練された電子署名方式が用いられている．なお，いずれの電子署名方式においても一般的に，署名者のみが知る秘密情報（上記の例では X_0 と X_1 の組）を署名鍵，若しくは，秘密鍵と呼び，署名の照合に用いる公開情報（上記の例では $F(X_0)$ と $F(X_1)$ の組）を検証鍵，若しくは，公開鍵と呼ぶ．

5　暗号資産のための拡張電子署名

本章では，暗号資産関連分野において，一部で利用，若しくは，利用が検討されている電子署名の拡張技術について紹介する．

5.1　マルチ署名

前章で述べたとおり，暗号資産関連技術においては，電子署名を用いることで，ブロックチェーン上に記載された取引情報が支払者の意思によるものであることを保証する．その際，電子署名の発行には署名鍵が必要となるが，この署名鍵を誰がどのように管理するのかについては慎重な判断が求められる．特に，署名鍵の漏えいは，当該署名鍵にひも付けられた資産の流出を直ちに意味する．そのような署名鍵の漏えいを防止するための手法としてマルチ署名（マルチシグ）が知られている．

マルチ署名とは，複数の署名鍵を用いて，若しくは，単一の署名鍵を幾つかに分割して，電子署名を生成する手法の総称である．これらの技術を用いることで署名鍵の漏えいリスクの分散が可能となり，したがって，暗号資産の流出の可能性を軽減することができる．マルチ署名を構成するためのアプローチは幾つかに分類される．最も素朴なアプローチは，単純に一般的な電子署名の署名鍵を複数用意し，これら全てを用いて複数の電子署名を発行するものである．このとき，発行された全ての電子署名の正当性が確認されない限り，当該取引情報が支払者の意思によるものとみなさないことで，署名鍵の漏えいリスクを分散することができる．しかし，この場合，電子署名のサイズは利用した署名鍵の個数に比例して大きくなるため，効率的とはならない．そのため，より効率的なアプローチとして以下のものがある：

①狭義のマルチ署名

前述のとおり，複数の署名鍵を用いて単純にマルチ署名を構成しようとすると，署名サイズが署名鍵の数に応じて大きくなってしまう．しかし，準同形性を持つゼロ知識証明等の特殊な暗号技術を応用することで，署名鍵のうちの一つを用いて生成された電子署名に対して，更に別の署名鍵を用いて電子署名を発行したとしても，署名サイズが増えないような方式の構成が可能となる．このような電子署名技術が狭義でのマルチ署名 [3] であり，これまでにも活発に研究がなされている．

②集約署名

集約署名（アグリゲート署名）[4] は，狭義のマルチ署名の一種と捉えることができる．集約署名においては，前述の素朴なアプローチのように，複数の署名鍵を用いて個別に複数の電子署名を発行した後，これらの電子署名を単一の電子署名に集約し，サイズを大幅に削減することが可能である．もちろん，このような性質は一般的な電子署名において提供されていないため，集約署名の構成は，だ円曲線上の双線形写像（いわゆる，ペアリング演算）等の特殊な代数的構造を用いて設計がなされている．

③しきい値署名

しきい値署名 [5] は，上記の二つのアプローチと異なり，単一の署名鍵を幾つかに分割し，これらを個別に用いて電子署名の分散片を独立に生成し，このように生成された分散片を合成することで電子署名の発行を行う技術である．この際，署名鍵が一度も復元されないまま，当該署名鍵により電子署名が最終的に発行される点に注意されたい．ECDSA 方式のような標準的な電子署名方式を効率的に閾値署名化する手法も知られており，従来技術との親和性も高い．

5.2　リング署名

電子商取引においては，利用者のプライバシー保護も重要な観点となる．一般的な暗号資産関連技術においては，署名鍵（に対する検証鍵）と暗号資産のひも付けがなされているため，取引の際に直ちに支払者に関する氏名などの個人情報が漏えいするわけではないが，同一人物による支払い履歴の追跡などが可能となるおそれがある．そのため，暗号資産における利用者の匿名性を高めるための手法についても活発に議論がなされている．リング署名[6]はそのような匿名性についての向上がなされた電子署名である．リング署名においては，署名者は他の無関係な複数の署名者の検証鍵を選び，自らの署名鍵のほかに，これらの検証鍵も用いて電子署名の生成を行う．こうして生成された電子署名の検証においては，電子署名の生成に使用された全ての検証鍵のほかに，署名者自身の検証鍵が用いられる．このとき，検証者は，検証に用いられる多数の検証鍵のうち，いずれに対応した署名鍵が当該電子署名の生成に使用されたのかを知ることができない．そのため，リング署名を用いることで複数の取引情報に関して，これらが同一支払者によるものであるかについて判断が困難となるため，利用者のプライバシー保護が可能となる．

■ 文献

(1)　SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, FIPS PUB 202, 2015.
(2)　Digital Signature Standard (DSS), FIPS PUB 186-4, 2013.
(3)　K.Ohta and T.Okamoto, “A digital multisignature scheme based on the fiat-shamir scheme,” Proc. ASIACRYPT 1991, pp.139-148, Nov.1991.
(4)　D. Boneh, C. Gentry, and B. Lynn, “Hovav shacham: aggregate and verifiably encrypted signatures from bilinear maps,” Proc. EUROCRYPT 2003, pp.416-432, May 2003.
(5)　Y. Desmedt and Y. Frankel, “Shared generation of authenticators and signatures (extended abstract),” Proc. CRYPTO 1991, pp.457-469, Aug.1991.
(6)　R. L. Rivest, A. Shamir,and Y. Tauman, “How to leak a secret,” Proc. ASIACRYPT 2001, pp.552-565, Dec.2001.

花岡悟一郎（正員）

産業技術総合研究所サイバーフィジカルセキュリティ研究センター高機能暗号研究チーム長．1997 東大・工卒．2002 同大学院工学系研究科博士課程了．博士（工学）．日本学術振興会特別研究員（PD）を経て，2005 から産業技術総合研究所所属．高機能暗号の設計，安全性評価，応用技術の研究開発に従事．平 30 年度科学技術分野の文部科学大臣表彰（科学技術賞），第 15 回ドコモ・モバイル・サイエンス賞（先端技術部門），平 19 及び 30 年度本会論文賞，The Wilkes Award（2007・British Computer Society）等各受賞．

ブロックチェーン・暗号資産の光と影

面　和成　Kazumasa Omote　筑波大学

1 まえがき

ブロックチェーンのユースケースとして，最も成功したと言われているものに暗号資産*1がある．ブロックチェーンは，耐改ざん性と高可用性を有する新しい技術であり，非中央集権的な環境に適しており，近年非常に注目を浴びている．しかしながら，ブロックチェーン・暗号資産は依然としてセキュリティ的に解決しないといけない課題がある．本稿では，ブロックチェーンの説明から始め，その代表的なユースケースである２種類の暗号資産（ビットコイン，イーサリアム）を解説し，それらに潜むセキュリティリスクについて言及する．本稿が，ブロックチェーンや暗号資産に関心がある人，またそれらの研究・開発等を始めたいと思っている人に少しでも役立てられれば幸いである．

2 暗号資産の特徴について

暗号資産は，銀行を介さない個人間送金が国境を越えて行えるものである．銀行に頼る必要がないため，銀行口座を持たないユーザもそのような送金が可能となる．このような銀行に頼らないグローバルな送金は，現時点で暗号資産でしか実現できない．一方で国内に閉じた話に限って言えば，銀行を介さない個人間送金は暗号資産でなくても行える．近年流行っている「○○ペイ」を利用すればその実現は容易である．したがって，暗号資産は個人間送金の世界統一化に向けた一つの解決策と言えるかもしれない．

図１は，暗号資産においてどのように送金が行われるのかを示した図である．アリス，ボブ，キャロル*2の３人の参加者*3が金融取引を行っており，アリスが5,000円を持っているとする．このとき，「アリスがボブに1,000円支払う」というトランザクション（取引データ）をアリスがブロードキャストすると，各参加者で送金の履歴が更新される．図１ではブロックチェーンのイメージに近づけるため，ジェネシスブロック（最初のブロック）から各ブロックを積み上げて送金等の履

＊１　暗号通貨や仮想通貨とも呼ばれる

＊２　システム上では仮名となる

＊３　システムとして考える場合はノードと呼ぶ

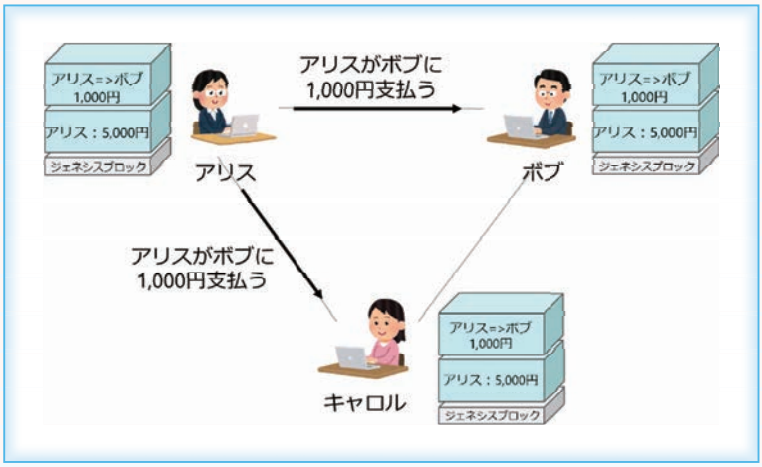

図１　暗号資産における送金

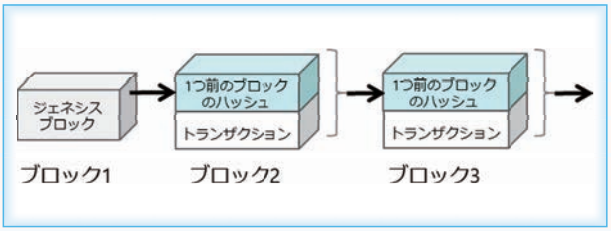

図２　ブロックチェーンの基本的なデータ構造

歴を更新しており，アリスが最初に 5,000 円持っていたこともブロックに記載されている．これにより，ある人が別の人に 1,000 円を支払う場合，1,000 円札が元の所有者から新しい所有者のもとへ物理的に移動するような不換紙幣の取引に近いことが電子的に実現できることが分かる．

3　ブロックチェーン（Blockchain）

3.1　ブロックチェーンとは

暗号資産にはブロックチェーンが使われている．ブロックチェーンは，暗号技術と P2P（Peer-to-Peer）ネットワークの両面を併せ持つ技術であり，P2P で価値を取引するためのセキュアな分散台帳システムであると言える．銀行などの信頼できる第三者機関は必要ない．またブロックチェーンは，ブロックチェーンネットワーク上のノード間で共有されているトランザクションの台帳である．この台帳は，追加のみが可能なデータベースであり，変更や改ざんができない．

図２にブロックチェーンのデータ構造を示す．各ブロックがその前のブロックのハッシュ値を含んでいるため，一つ前のブロックのトランザクションの改ざんが困難となる．最後のブロックが分かれば，それ以前の全てのブロックにアクセスすることが可能となってジェネシスブロックにたどり着く．

ブロックチェーンにおいて，どのようにトランザクションが発行され，どのようにブロックチェーンがネットワーク上で更新されるのかの大まかな流れは図１を参照されたい．トランザクションがブロードキャストされると，それがチェックされてからブロックの中に格納され，そのブロックが共有された後に各参加者がそのブロックを追加するという流れである．各参加者は基本的に同じブロックチェーンを持つことになる．

3.2　ブロックチェーンの性質

ブロックチェーンの主な性質としては以下がある．

3.2.1　耐改ざん性

トランザクションがブロックチェーンに記録されると，それを変更することは事実上不可能である．あるブロックが変更されると，後続のブロックのハッシュ値が全て変更されるため，あるブロックの改ざんを成功させるにはそれ以降の全てのブロックを作り直さないといけない．そのため，後続のブロックが追加されるほど，そのブロックの耐改ざん性が増す．この性質は暗号技術で実現されている．

3.2.2　高可用性

ブロックチェーンネットワークは，複数のノードによって支えられており，一時的なノードの故障，時折発生する一部の計算ノードの利用不能，ネットワーク遅延やパケット消失などに耐えるように設計されている．

3.2.3　二重使用耐性

暗号技術では二重使用を防ぐことはできない．それは，通常のトランザクションと二重使用のトランザクションは，いずれも正当なものだからである．したがって，ノードが過去に起きた全てのトランザクションを認識することにより，あるトランザクションが二重使用の試みなのかを判別する．

4　ブロックチェーンのための暗号技術

ブロックチェーンで用いられる基本的な暗号技術は，次の二つである．暗号という名称が付くが，暗号化されているわけではないことに注意する．

4.1　暗号学的ハッシュ関数

暗号学的ハッシュ関数は，任意長の入力データを変換し，固定長の出力を生成する一方向性関数である．ハッシュ値は，ある任意の入力メッセージに対して効率的に計算可能であり，同じハッシュ関数に同じ入力を与えれば，毎回同じハッシュ値を生成する．暗号学的ハッシュ関数 $H(\ )$ は，次のセキュリティの性質を満たす．

衝突困難性：$H(X) = H(Y)$ を満たす X と Y を求めることが事実上不可能となる性質．

原像計算困難性：出力 $H(X)$ から入力 X を求めることが事実上不可能となる性質．

第二原像計算困難性：ある入力 X とそのハッシュ $H(X)$ が与えられたとき，$H(X) = H(Y)$ となる Y を求めることが事実上不可能となる性質．

4.2　ディジタル署名

送信者はメッセージをハッシュ化したものに署名し，生成された署名データをメッセージに付加して受信者に送る．例えば，ビットコインではだ円曲線をベースとし

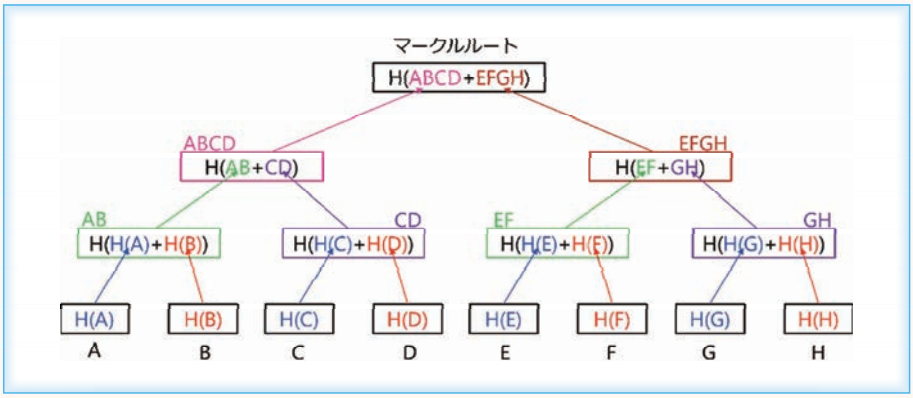

図３　マークルツリー

た ECDSA 署名が採用されている，受信者は，この署名を検証することでメッセージの真正性をチェックできる．ディジタル署名は，以下のセキュリティの性質を満たす．

データの完全性：データが変更されていないことを保証する．データが変更されていれば，メッセージのハッシュ値が一致しないため検証が通らない．

否認不可：送信者がディジタル署名したことを後で否定できない．

4.3　マークルツリー（Merkle Tree）

マークルツリーは，暗号学的なハッシュポインタを用いた二分木である．これは，ペアのデータをハッシュ化し，そのハッシュ出力を更にハッシュ化するといったことを，マークルルートと呼ばれるルートノードまで繰り返す（図 3）．例えばビットコインでは，葉はブロックチェーンにおけるブロック内のトランザクションに対応する．（図 3 では，A〜H が各トランザクションに対応．）マークルツリーは，ブロックチェーンのデータ構造と同様に耐改ざん性を有する．

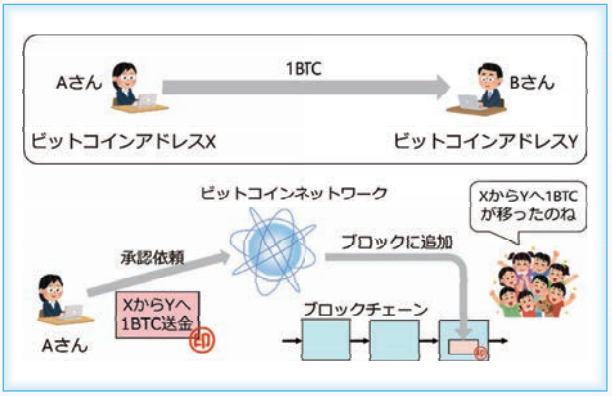

図４　ビットコインの送金の概念図

5　ビットコイン（Bitcoin）

ビットコインは，いかなる国にも限定されないグローバルな資産を目指す非中央集権的な暗号資産である [(1)]．ここでは，**3.** のブロックチェーンの内容を踏まえてビットコインについて詳しく説明する．

5.1　ビットコインの送金

ビットコインの送金について，図 4 の概念図を用いて説明する．ここでは，A さんが B さんに 1BTC（BTC はビットコインの単位）を送金することを考える．このとき，A さんのウォレット（ビットコインアドレス X）から B さんのウォレット（ビットコインアドレス Y）にビットコイン（1BTC）が送金される．ただし，何か電子的なお金が送られるわけではないことに注意する．送金手順としては，まず A さんが「X から Y へ 1BTC 送金」というトランザクションを A さんのディジタル署名付きでビットコインネットワークにブロードキャストして承認依頼を行う．承認がなされると，そのトランザクションがブロックに追加され，そのブロックがブロードキャストされる．各参加者はそのブロックを検証し，問題なければそのブロックをブロックチェーンに追加する．ブロックチェーンは誰もがその内容を確認できるので，参加者はこの記録を参照することで X から Y へ 1BTC 移ったことに合意する．ビットコインアドレスの生成方法については，最初にディジタル署名の秘密鍵となるランダムなビット列を生成し，それを変換して公開鍵にする．更に公開鍵は 2 回ハッシュされてビットコインアドレスに変換される．

5.2　ビットコインのブロックチェーン

ビットコインのブロックチェーンは図 5 のようになっている．各ブロックはヘッダ部とボディ部を持つ．ヘッダ部は基本的に，ハッシュ値，タイムスタンプ，マークルルート，ナンスの四つの値がある．マークルルートは

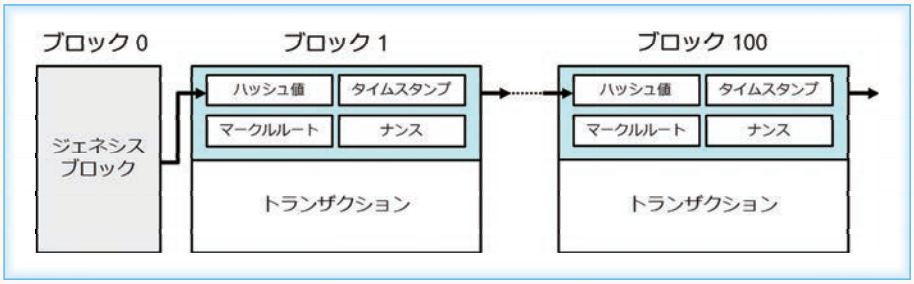

図 5　ビットコインのブロックチェーン

4. で説明したように，ボディ部にあるトランザクション（サイズの上限は 1MByte）で構成されるマークルツリーのマークルルートが格納されている．そのため，ブロックのハッシュ値はブロックのヘッダ部のハッシュ値で十分となる．各ブロックには，実際一つ前のブロックのヘッダ部のハッシュ値が格納されている．2020 年 1 月現在では，ブロックチェーンのサイズは約 260GByte である＊4．

5.3　探索パズルと PoW（Proof of Work）

ブロック内のハッシュ値が有効となるためには，そのハッシュ値が特定の値以下という条件を満たさなければならない．具体的には，ハッシュ値の先頭ビット列が一定数ゼロとなることで特定の値以下となる．また，ブロック 1 のナンス（nonce）＊5 を調整することで，ブロック 2 のヘッダに格納されるブロック 1 のハッシュ値が変化する．ナンスは，上記の条件を満たすようハッシュ値を調整するための値である．いち早く有効なハッシュ値を見つけようとするこのパズルは探索パズルと呼ばれる．図 6 では，ナンスの値をナンス *a*，ナンス *b*，ナンス *c* と変化させていき，ナンス c で探索パズルが解けたことを意味する．探索パズルが解ける時間は，条件となっている先頭ビット列のゼロの個数で調整可能であり，約 10 分になるように設定されている．

PoW は日本語訳で「演算（量）の証明」と訳される．一つのブロックを追加するには一つの探索パズルを解かなければならない．この探索パズルを解く効率的な方法が存在しないため，誰がやっても約 10 分の時間が掛かることから，一つのブロックを生成する演算量が平均で 10 分ということになる．つまり，ブロックチェーンの長さがそのまま演算量の証明になっている．探索パズルが解けるとマイニングが成功となる．

＊4　https://www.blockchain.com/ja/charts

＊5　number used once の略で一度だけ使用される使い捨ての数を表す．

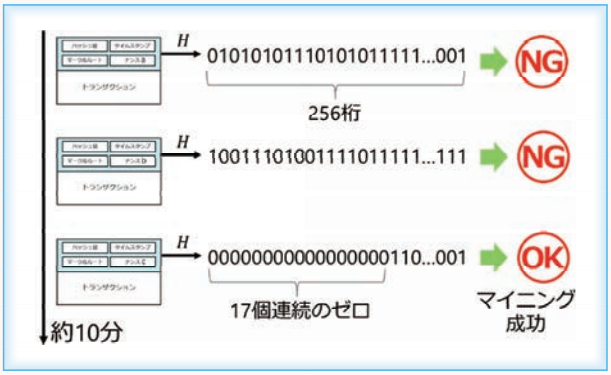

図 6　探索パズルとマイニング成功

5.4　マイニング

マイニングとは，ビットコインにおいて二つの意味を持ち，このことがビットコインを少し分かりにくくしている．一つは文字どおりビットコインの採掘であり，もう一つはトランザクションの承認行為である．計算能力を有する者であれば誰でもマイニングに参加して，新たなビットコインを採掘できる．採掘されたビットコインは報酬として，一番早く探索パズルを解いたマイナー（採掘者）にブロックチェーンから自動的に支払われる．（このときトランザクション手数料も合わせて支払われる．）これがマイニングを行うインセンティブとなり，マイニングに競争原理が働くことによって承認行為が強力に実行されるという仕組みである．

5.5　コンセンサスとフォーク

ビットコインのブロックチェーンは世界でただ一本しか存在しない．ただし，各ノードが独立してマイニングを行うので，当然異なるブロックが生成され，ブロックの分岐が発生する．（これをソフトフォークと呼ぶ．）最長のチェーンが有効なブロックチェーンであることがノード間で合意されているので，各ノードは最長のチェーンに対して構築を続けようと試みる．マイナーは，報酬のために無駄なことはせず，より長いブロックチェーンにつながるブロックをマイニングしようとするため，結果としてブロックチェーンが一本に収束していく．

5.6　ビットコインネットワーク

ビットコインネットワークは，中央サーバが存在しないP2Pネットワークであり，各ノードが等しく扱われる．ビットコインネットワークは，ノードが自由にネットワークに参加／離脱できて，それでもなおシステムが機能する．オンライン／オフラインを合わせた全世界のトータルのノード数は不明であるが，オンラインのノード数は約1万台である[*6]．また，非同期であり，ネットワーク遅延やパケット消失がありながらも，システムは非常に堅ろうである．ビットコインネットワークは，システム的な単一障害点がないだけでなく，管理主体もない，非中央集権的なネットワークであると言える．

ただし，ノードがビットコインネットワークに参加する場合，ノードのIPアドレスを管理している「DNS seed」と呼ばれるDNSサーバ群に頼る必要がある．このサーバ群は信頼されており，中央集権的な側面を持つことに注意する．

5.7　トランザクション

ビットコインの各所有者は，ビットコインを受け取った以前のトランザクションのハッシュ値と受取人の公開鍵を合わせてディジタル署名を施すことで，コインをほかの誰かに譲渡できる．受取人は，支払人の公開鍵を既に持っているので，トランザクションを検証することができる．

図7は，ビットコインのホワイトペーパー[(1)]に載っているトランザクションの仕組みを表した図である．図の中央部分（所有者1＝Owner1＝が発行するトランザクション）に着目する．所有者1がトランザクションを発行する際，所有者1は次の二つの値（A），（B）（図の（A），（B）に対応）を合わせたもののハッシュ値を計算し，それに対して自身の秘密鍵（Owner 1's Private Key）を使ってディジタル署名を施す．

（A）所有者1がその額を受け取った以前のトランザクション

（B）所有者2の公開鍵

正当なトランザクションであることを保証するために，このディジタル署名は「（C）所有者1の公開鍵」を使って検証できる．同様に，所有者2が所有者3への譲渡を行うとき，所有者2は自身の秘密鍵を使って，以前のトランザクション（所有者1から受け取ったもの）と所有者3の公開鍵を合わせてハッシュしてからディジタル署名を施す．このようなトランザクションは，ネットワークに参加している者であれば誰でも公開鍵を使って検証できる．

なお，本稿では誌面の都合上UTXO（Unspent Transaction Output）の説明を割愛した．

6　イーサリアム（Ethereum）

イーサリアムはイーサーと呼ばれる暗号資産を扱い，更に非金融データも扱うことができる．本章では，紙面の都合上，イーサリアムの説明は最小限とする．

6.1　非金融データの取り扱い

ブロックチェーンは，暗号資産だけでなくどんな価値の取引や記録のためにも使える．例えば，「存在証明」はそのようなユースケースの一つである．これは，ある文書が特定の時点で存在していたことを後から誰でも検証できるように，その文書のハッシュ値をブロックチェーンに格納するものである．イーサリアムのブロックチェーンプラットホームでは，お金だけでなく，株，土地，ディジタルコンテンツなど，何らかの本質的価値を持つ多くのものの取引を容易にする．また，イーサリ

＊6　https://bitnodes.earn.com/

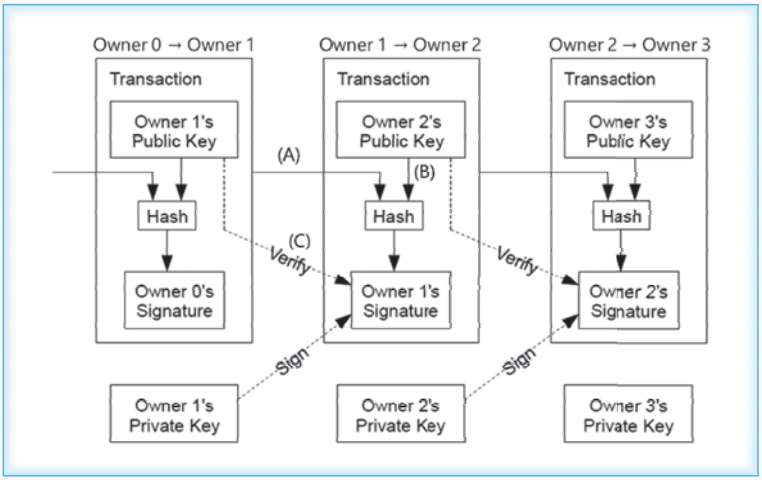

図7　ビットコインのトランザクション[(1)]

アムでは，異なるアプリケーションのトランザクションが，全てのイーサリアムノードで実行できる特徴を持つ．

6.2　スマートコントラクト

スマートコントラクトとは，ブロックチェーン上で契約を自動的に実行する仕組みのことである．これは自動販売機によく例えられ，利用者が硬貨を投入し，ボタンを押すと契約が成立するのに似ている．実際には，トランザクションを発行することによって，コントラクト（プログラム）がブロックチェーン上に配置されたり，コントラクトが実行されたりする．コントラクトはブロックチェーンを持つ全てのノードが実行する．

ビットコインとは異なり，イーサリアムはチューリング完全な言語をサポートしているので，スマートコントラクトはプログラミング的な観点で言えば，コンピュータ上でできることは何でも実行できることになる．ノードがスマートコントラクトを実行するにはイーサリアム仮想マシン（EVM）が必要である．

6.3　イーサリアムのブロックチェーン

イーサリアムブロックチェーンのデータ構造は，ビットコインによく似ている．ただし，堅ろう性を高め，状態を適切に保持するのを助けるため，ヘッダには多くの情報が含まれている．ビットコインでは，ブロックにある全てのトランザクションに関して，ブロックヘッダにマークルルートが一つあるだけであったが，イーサリアムでは，ステートルート，トランザクションルート，レシートルートの合計で三つが存在する．

6.4　イーサリアムの手数料

イーサリアムのトランザクションは，イーサリアムにおける計算の基本単位である「ガス」（gas）に基づいて動作する．マイナーに支払われるこのトランザクション手数料は，ノードの稼動とネットワーク全体の維持に対する報酬となる．

7　暗号資産におけるセキュリティリスク

暗号資産におけるセキュリティリスクについて，様々な攻撃が指摘されている．例えば，51％攻撃やセルフィッシュ・マイニング攻撃，二重支払い攻撃などは，ブロックチェーン特有の攻撃として有名なものである．本稿では，これら有名な攻撃の解説はほかに譲るとして，ここでは我々が研究している暗号資産のセキュリティリスクについて解説する．具体的には，①ブロックチェーン汚染攻撃，及び②ブロックチェーンネットワークサービスへの攻撃について述べる．

7.1　ブロックチェーン汚染攻撃

暗号資産で用いられるブロックチェーンにおいても，非金融データの格納が可能である．それはビットコインでも同様である．Matzutt ら [2] は，ビットコインを対象として，ブロックチェーンに格納される非金融データに着目し，悪意あるユーザにより違法なコンテンツがブロックに格納されるブロックチェーン汚染攻撃の問題を指摘している．更に我々 [3] は，より多くの非金融データがブロックに格納可能なイーサリアムを対象として，ブロックに対する汚染攻撃の実態について分析した．図8は，イーサリアムブロックチェーンへの汚染データに関する我々の分析結果である．横軸は埋め込まれていたファイルの種類，縦軸は抽出できた数（一つのトランザクジョンから一つのファイルを抽出）である．イーサリアムにおけるジェネシスブロック（2015年7月30日）から2018年11月30日までの全ブロックのデータ領域を分析対象とした．データ領域は，スマートコントラクトに使われる領域である．

我々の分析結果から，イーサリアムのデータ領域には，違法な画像やマルウェアが含まれていることが明らかになった．ノードはブロックチェーンを全て持つので，この分析結果は，世界中のノードが違法な画像やマルウェアを強制的に持たされてしまうことを意味する．違法な画像の所持は国によっては逮捕される可能性があり，またマルウェアはサイバー攻撃に利用されるリスクがある．このような違法コンテンツは，ブロックチェーンの耐改ざん性により，各ユーザが容易に削除できるものではない．したがって，ブロックチェーン汚染攻撃はブロックチェーンシステムの根幹を脅かす攻撃となり得るものである．

7.2　ブロックチェーンネットワークサービスへの攻撃

イーサリアムネットワークは，柔軟性と運用性の点から，JSON-RPC と呼ばれる遠隔から制御できる機能が実装されている．これにより，ノードの管理者は遠隔から

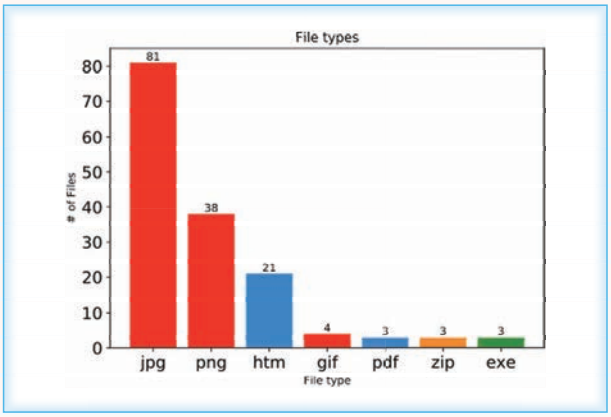

図8　イーサリアムブロックチェーンへの汚染データ [3]

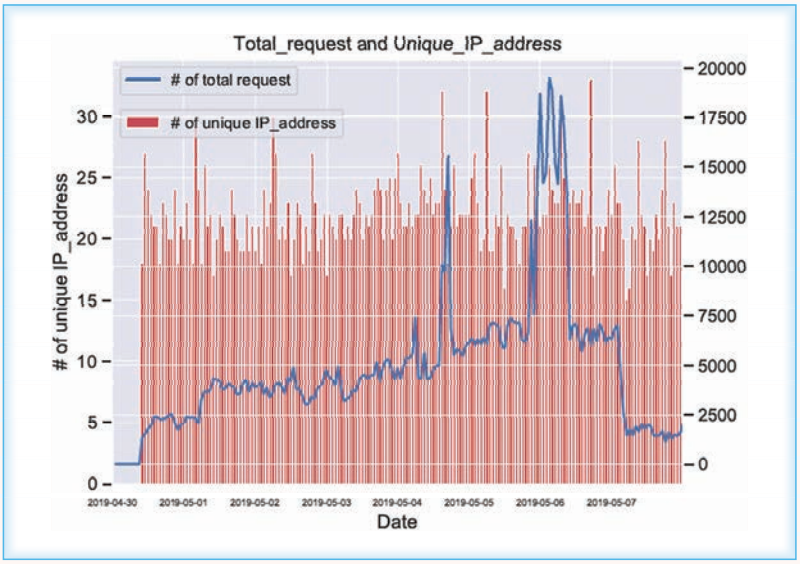

図９　イーサリアムネットワークへの攻撃パケット [4]

アクセスしてノードの様々な制御ができるようになる．

我々 [4] は，イーサリアムネットワークを対象として，全世界９か国に「おとりノード」を設置し，それらにアクセスしてくる不正な通信を観測した．図９は，2019 年４月 30 日から５月７日までの８日間における，我々が設置したイーサリアムノードへの攻撃パケットを時系列で示したものである．赤色の棒グラフはアクセスしてくるユニークな接続元（送信元 IP アドレス）の数を示しており，青色の折れ線グラフはアクセスしてくる全リクエスト数を示している．

我々の分析結果より，接続元の数がほぼ一定であるのに対して，特定のマシンからの接続が急増していることが読み取れる．この不審な JSON-RPC による遠隔からのアクセスを詳細に分析したところ，暗号資産が幾らあるのか，パスワードはぜい弱なのかといったことを試みる通信が世界中からなされていることが明らかになった．そのため，これらの通信は悪意ある振舞いであると判断できる．本来なら，JSON-RPC でのアクセスは管理者である我々しか行わないはずである．したがって，アクセス制御を実施していない JSON-RPC の通信ポートが，常にインターネットからの攻撃にさらされているということを肝に銘じておく必要がある．

8　むすび

ブロックチェーンは暗号技術が利用されているため，技術的に安全であると言われているが，ブロックチェーンをネットワークシステムとして捉えてみると，多くのセキュリティリスクが存在していることがうかがえる．本稿では，ブロックチェーン・暗号資産の良い面だけでなく，それらに潜むセキュリティリスクというマイナス面についても解説してきた．今後は，これらのセキュリティリスクを踏まえ，真に安全なブロックチェーンシステム・暗号資産の実現に向けて，より一層の研究に取り組んでいくことが重要である．

■ 文献

(1) S. Nakamoto, “Bitcoin: a peer-to-peer elec tronic cash system,” 2009, http://www.bitcoin.org/bitcoin.pdf
(2) R. Matzutt,J. Hiller,M. Henze,J. H. Ziegeldorf,D. Müllmann,O. Hohlfeld,and K. Wehrle, “A quantitative analysis of the impact of arbitrary blockchain content on bitcoin,” Proc. FC2018,pp.420-438,Feb.2018.
(3) T. Sato, M. Imamura, and K. Omote, “Threat analysis of poisoning attack against Ethereum Blockchain,” Proc.WISTP2019,pp.139-154,Dec. 2019.
(4) 原　和希，佐藤哲平，今村光良，面　和成，“ブロックチェーンネットワークにおけるハニーポット設置に向けた悪意あるユーザのプロファイリング”，信学技報，ISEC2019-14，pp.15-22，July 2019.

面　和成（正員）

筑波大・システム情報系・准教授．2002-03 北陸先端大情報科学研究科博士後期課程了．博士（情報科学）．同年４月富士通研究所入所，セキュアコンピューティング研究部に配属．2011 年北陸先端大情報科学研究科准教授，2019 から情報通信研究機構招聘専門員兼任．2016 から現職．