［ブロックチェーン技術の最新動向］

① Bitcoin 技術のその後の動向

基般

佐古和恵 | NEC　古川　諒 | NEC　中川紗菜美 | NEC

サトシの Bitcoin

動作概要

2008 年にサトシナカモトが考案した Bitcoin [1] の基本的な構成要素として，各アカウントが保有する「Bitcoin」を別のアカウントに移転するトランザクションと，このトランザクションデータを記録する分散台帳（ブロックチェーン）がある．各アカウントは公開鍵に紐づく．A アカウントから B アカウントに X Bitcoin（以下，資産の単位の場合は BTC と表記する）を移転するトランザクションには A アカウントの署名データが付与される．このトランザクションが発生したときに，A アカウントに X BTC が存在すると台帳に記載されており☆1，なおかつ署名データが正しければ，このトランザクションデータは分散台帳に記録され，A のアカウントの残高が X BTC 減額され，B のアカウントの残高は X BTC 増額される．分散台帳に記録される際に，トランザクションはブロック単位でまとめられ，Proof of Work（PoW）と呼ばれる処理時間がかかるコンセンサスアルゴリズムが実行される [2]．このため，多くても 1 秒にせいぜい数十トランザクションしか分散台帳に記録されない．

課題

Bitcoin 方式に関しては，多くの課題が存在する．ここですべてリストできるものではないが，主なものを挙げると，

1. PoW 方式に起因して，多くの計算機資源が必要になること
2. 単位時間に取り扱えるトランザクションの数が少ないこと
3. トランザクションが将来にわたって確定するという保証はないこと
4. すべてのトランザクションが公開されているため，いつどの口座間でどのくらいの取引があったかが周知になってしまうこと
5. 支払条件を限定的にしか表現できないこと

がある．それぞれについて，いくつかの代替案が提案されている．たとえば，計算資源の課題 1 には PoW 方式の代わりになる PoS（Proof of Stake）方式などが提案されている．課題 2 に関しては，1 つのブロックにより多くのトランザクションが含められるように，ブロックのサイズを大きくする案や，ブロックに記載するトランザクションのサイズを小さくする Segwit（Segregated Witness）☆2 などの案が提案され，一部実用化されている．課題 3 はファイナリティの問題と呼ばれ，PoW 方式に起因する．そこで，誰もがノードになれる現行の Permissionless 方式ではなく，ノードになる人が事前に決められている Permissioned の方式が考えられている．課題 4 のプライバシに関しては Zcash☆3 など，新たな方式が提案されている．課題 5 に関しては，支払条件をスマートコントラクトとして表現できる Ethereum などの方式が開発されている．

☆1 具体的には，台帳には，A アカウントが使える残高を表現する UTXO と呼ばれる複数のパラメータで表現されている．

☆2 https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki

☆3 https://z.cash/

本稿の概要

本稿では，上記の課題 2 に対して，PoW 方式はそのままで，Bitcoin の外部においたメカニズムによって，多くのトランザクションが高速に実施できる Payment Channel 手法について紹介する．また課題 5 の対策として Bitcoin の機能を拡張するため，ほかのブロックチェーンに資産移転する Side-chain と呼ばれる方式を紹介する．さらに Bitcoin に使われている署名方式を Schnorr 署名に変更することで同じく課題 5 に対応する方式も述べる．

Payment Channel

前述したとおり，Bitcoin は単位時間あたりに処理できる取引量が小さい．これを解決するために提案されたのが Payment Channel と呼ばれる技術である．

通常，Bitcoin で取引をする際には取引ごとにトランザクションを作成し，それぞれが台帳に記載される必要がある．これに対し，Payment Channel はトランザクションを Bitcoin の外側（off-chain）で一時的に保持するようにし，あるタイミングで複数の取引による最終結果のみをまとめて台帳に反映する技術である（図 -1）．

これにより台帳に記載するトランザクション数が減少するため単位時間で可能な取引数を向上できる．ここでは，特定の 2 者間で単方向および双方向に支払いが可能な Micro Payment Channel と，それを用いて任意の 2 者間での取引を実現する Lightning Network について述べる．

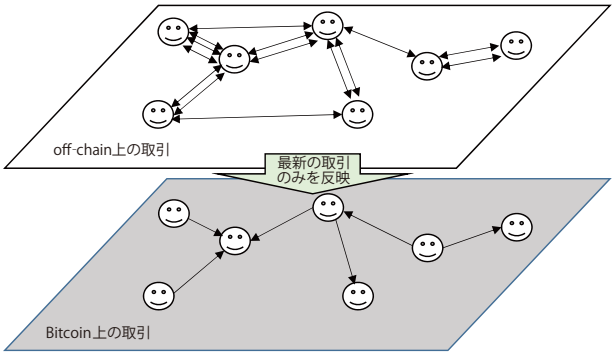

図 -1　Payment Channel の概要

単方向の Micro Payment Channel

まずは 2 者間で単方向の支払いが可能（A から B へは可能だが B から A は支払い不可）な Micro Payment Channel 技術について述べる．ここでは，たとえば，ビデオの毎分視聴するごとに料金を Bitcoin で支払う場合を考えてみよう．

複数の取引をまとめて処理をする場合，途中の取引内容は台帳に記載されないため，正しく取引が処理されずに一方が損をする可能性がある．このため，Micro Payment Channel では相手に不正をされても他方が損をしないような工夫をしている．その工夫の 1 つは，Bitcoin に実装されているマルチシグ口座を活用する．A と B のマルチシグ口座とは，A と B の両方が署名をしないと，この口座からの Bitcoin を移動できないというものである．

A から B への支払いを頻繁に行う場合，まず A が保有する Bitcoin の一部を，A と B のマルチシグ口座に移動するトランザクションを発行する．これが Bitcoin の分散台帳に記載されると，この額がマルチシグ口座にデポジットされたことになる．以後，この講座をデポジット口座と呼ぶ．

今後，A から B に Bitcoin で支払う場合には，A はこのデポジット口座から B に支払うトランザクションデータを作成して，署名して B に渡す．たとえば 1BTC をデポジットしていて 0.1BTC を支払う場合には，「デポジット口座から B に 0.1BTC，A に 0.9BTC を移転する」というトランザクションに A の署名を付与して B に送るのである．このトランザクションが B に署名され，Bitcoin の台帳に記載されれば B は 0.1BTC を入手できる．しかし，これでは通常の Bitcoin より高速にならない．

そこで，B は次の A の支払いまで待つのである．A は次の 1 分を視聴したら，今度は「デポジット

口座からBに0.2BTC，Aに0.8BTCを移転する」というトランザクションデータに署名してBに送る．このように，視聴時間が経過するたびに累計したトランザクションデータを作成してBに送る．視聴が一段落した，あるいはデポジットが空になった段階で，Bは最後のトランザクションデータに署名する．このトランザクションがBitcoinの台帳に記載されることで取引の総計が反映される（図-2）．

ただし，Bがトランザクションを Bitcoin の台帳に記載しなかった場合，デポジット口座は塩漬けになってしまう．この問題を解決するために，「一定時間たてばAの署名だけでもデポジット口座からの移動を可能とする」[☆4]，という条件を付けてデポジット口座を作成する．これによりBが一定時間の間にトランザクションを記載しなければ，Aはデポジット口座から資金を回収できるのである（図-3）．

なお，一定時間後に資金移動が有効になる機能（script）はBitcoinのTime Lock Contract（TLC）と呼ばれる．のちに紹介するLightning Networkでは，これに加えてハッシュ値の有無により，口座に振り込まれるかどうかを決定するHash Time Lock Contract（HTLC）scriptも登場する．

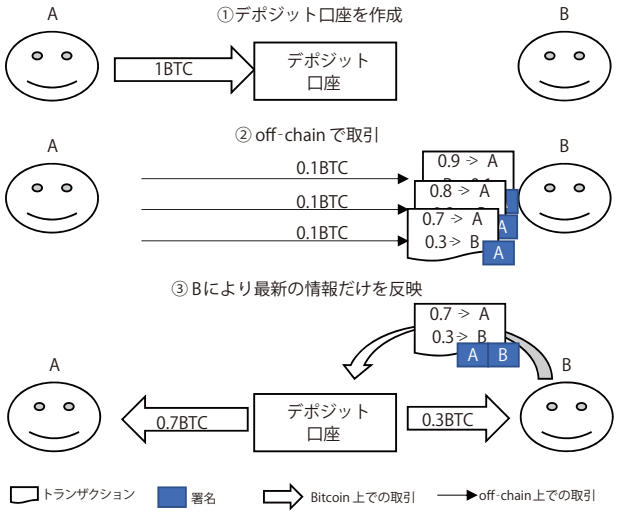

図-2　単方向の Micro Payment Channel

☆4　あるブロックの高さ以上でなければ口座から移動するトランザクションは有効にならない，と指定することにより実現される．

双方向の Micro Payment Channel

双方向の Micro Payment Channel では取引を行う両者がそれぞれ資金を出し合ってデポジット口座を作成し，お互いが最新取引を反映したトランザクションを持ち合い，どちらかが最新のトランザクションを Bitcoin の台帳に記載することで複数の取引をまとめて反映する（図-4）．

Bitcoin の移動が単方向の場合には取引を重ねるたびに受け取り側が受け取る金額が高くなるため，最新のトランザクションを Bitcoin の台帳に記載するのは当然であったのに対し，双方向の場合はチャネルに参加する両者にとって最新の取引を反映したトランザクションを送信するのが最良であるとは限らない．このため，最新でないトランザクションを送信した場合にペナルティを与える仕組みが必要である．

たとえば，AとBが同額をデポジットしたAとBのデポジット口座から「Bに0.8BTC, Aに0.2BTCを移転する」という取引1（図-4の②における2番

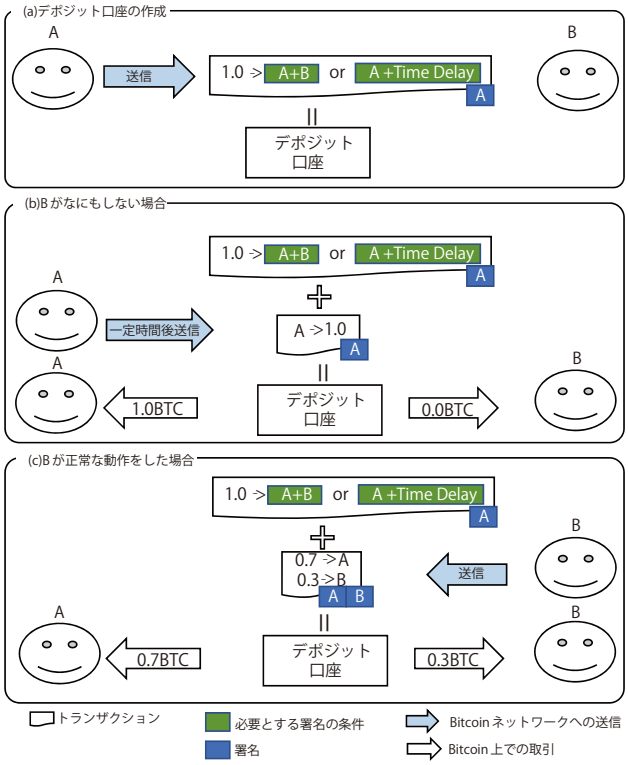

図-3　デポジット口座の塩漬けの防止

目の取引を指す）のあと，「A から B に 0.1BTC を移転する」という取引（図 -4 の②における 3 番目の取引）を行う場合を考える．この場合，2 つの取引の総和として「デポジット口座から，B に 0.9BTC，A に 0.1BTC を移転する」という取引 2 に更新されるべきである．しかし A にとって取引 2 でなく取引 1 を有効にした方が得になるので，B は A が取引 1 を有効にすることを防がないといけない．

そこで B は「B に 0.8BTC，A に 0.2BTC を移転する」という取引 1 において TLC を用いて「A への 0.2BTC は一定時間がたたないと移動できず，さらにこの 0.2BTC は A と B のデポジット口座へはいつでも移動できる」という条件を指定した取引にする．さらに，取引 2 の直前に，取引 1 が台帳に記載された場合に，「A への 0.2BTC は A と B のデポジット口座を経て B に移転する」というトランザクション（BRT, Breach Remedy Transaction）に A の署名を得る．これをもらっておいてから，取引 2 を実行すれば，A は不正をするインセンティブがなくなる．なぜならば，取引 1 に A の署名を追加して取引 1 を確定させても，自分の口座にいれるには時間がかかり，さらにその間に B が BRT を使って B の口座に移転させてしまう恐れがあるからである．

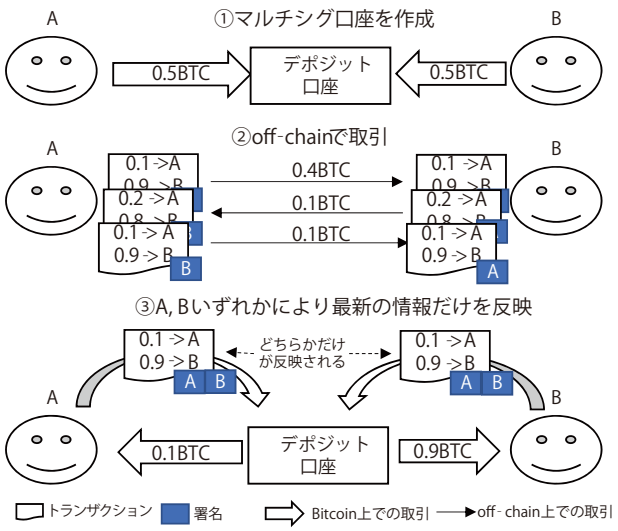

図 -4　双方向の Micro Payment Channel

上記は，A の不正を防止する観点で B が作成する取引について述べたが，同様に A も，B の不正を警戒して対称的に同じトランザクションを実施する．このような複雑な仕組みにより両者が最新の取引を Bitcoin に反映するように誘導し，双方向に支払いが可能となるのである．

Lightening Network

Micro Payment Channel では取引する 2 者ごとにデポジット口座を作成する必要があった．このため，ある利用者が 100 人の相手と Payment Channel による取引をしたければ，大量の Bitcoin をデポジットしなければならず，現実的には不可能である．

Lightning Network[3] はこの問題を解決するために生まれた技術である．Lightning Network では 2 者間の取引をリレーすることで，たとえばユーザ A とユーザ B，ユーザ B とユーザ C の間に Micro Payment Channel が存在すれば，A と C の間でも取引が可能である（図 -5）．これにより A と C の間のデポジット口座なしに A と C の間で取引ができる．

Micro Payment Channel をリレーする上で最も問題となるのは仲介するユーザによる持ち逃げである．たとえば A から B を経由して C に支払いを行う場合に，A から B に支払われた段階で B が C に支払わなければ B による持ち逃げが可能となる．

このような仲介者による持ち逃げを回避するための方法として，次のような手続きが提案されている．まず，最終的な支払先（A から C に支払う場

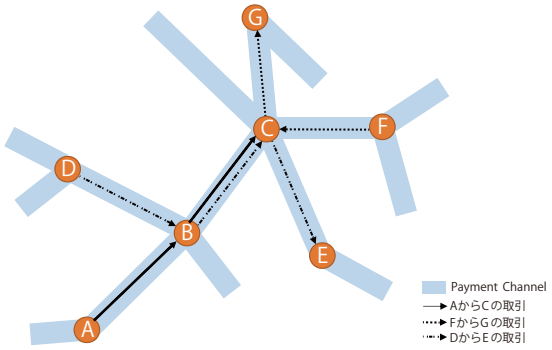

図 -5　Lightning Network のイメージ

合にはC）がランダムな数Rを作成し，これをハッシュ化したH＝h(R)を支払元（この場合はA）に送付する．Aは仲介をするBに対して支払いを行うが，このときにBがある期間内にH＝h(R')となるR'を送らないと，自分の口座にはいらないような条件を指定する．（これをHTLC（Hashed Time Lock Contract）と呼ぶ）．このようにすることで，R'を知らないBは自由に資産を入手することはできなくなる．同様にBからCへの支払いについても，R'をCが知っている場合にのみCへ資産移転が可能になる．CはR'の1つであるRを知っているため，資産を入手可能である．Cは自分への支払いを有効化する際にRを公開するため，ほかの仲介者もそれぞれ支払われたコインを使用できるようになる．図-6ではAとBの間にそれぞれ0.5BTCずつ拠出しているデポジット口座，BとCの間に同様に0.5BTCずつ拠出している口座がある場合に，HTLCを使用してAからCに0.1BTCの取引をBを経由して行う例を示している．このような仕組みにより，仲介者はリスクなく支払いを仲介できる．

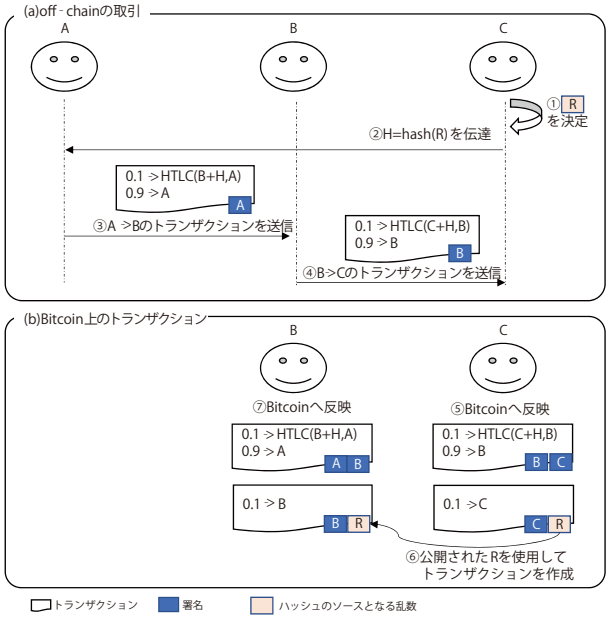

図-6　HTLCを使用したリレーの仕組み☆5

☆5　図-6中のHTLC（B+H,A）はBの署名+R′または一定時間後にAの署名で使用可能な口座を表す．

ブロックチェーン間の移転

Side-chainの概要

Side-chainとは複数のブロックチェーン間で資産を移動する技術を用いて，Bitcoinではサポートされていない機能の追加を別のブロックチェーンで行うことを可能としている．Side-chainの主な目的はこのようにBitcoinの機能拡張ではあるが，Bitcoinに記載されるトランザクションの量が減ったり，高速なSide-chainを採用することで結果的にスループットの向上にも寄与する．先に記述した課題2や5への対策になる．

Side-chainの実現例

では，Side-chainではどうやって複数のブロックチェーンの間で資産を移動するのであろうか．

資産移動の方法の1つとしてFederated pegと呼ばれる方法が用いられている．Federated pegでは信頼できる複数の運営者からなる運営チームの存在を仮定している．Bitcoin上でAから運営チームのマルチシグ口座へのトランザクションが台帳に記載されると，Side-chain上では運営チームのマルチシグ口座からSide-chain上のAの口座宛のトランザクションが送信され，AはSide-chain上での取引が可能となる．Side-chainでの取引後にAの資産をAのBitcoin上の口座に戻す際には，運営チームの仲介を経て逆の工程が実行されることになる．

しかしながらFederated pegでは仲介の運営チームやSide-chain上のルールを信頼する必要があり，Bitcoinのトラストモデルが変わってしまうとの指摘もある．

仲介不要のAtomic Swap

2つのブロックチェーン間の資産を仲介者なく交換することを目的にAtomic swapと呼ばれる手法の研究も近年さかんになっている．これを応用して，Side-chainを実現する方法もある．ここでは，

HTLC 機能がある 2 つのブロックチェーン間の資産を交換する手法を紹介する．

たとえば，A は Bitcoin の暗号資産を持ち，B は Litecoin と呼ばれる別の暗号資産を持っているとする．お互い，すでに合意した交換レートで A は B に Bitcoin を，B は A に Litecoin を渡すとする．A, B それぞれ Bitcoin ならびに Litecoin に口座を持っている前提である．

まず，HTLC に使うハッシュ値 H＝h(R) を A が決め，A は，Bitcoin 上で，H＝h(R') となる R' を示した場合にのみ，B に資産移転ができるようにするトランザクションを作成し，台帳に記載する．これを確認した B は，同様に Litecoin 上で，R を示した場合にのみ，A に資産移転ができるようにするトランザクションを作成し，台帳に記載する．そして，A は Litecoin 上で R を公開して，この資産を入手する．このとき，公開された R を使って，B も Bitcoin を獲得でき，中間者を介さずに資産交換ができることになる．

Schnorr 署名の活用

本章では，Schnorr 署名と呼ばれる Schnorr が提案した準同型性のある署名アルゴリズムを使って，Bitcoin の機能を拡張する方式を紹介する．ここで準同型性とは，同じメッセージ m に対して，公開鍵 Y1 と Y2 の署名から，公開鍵 (Y1＋Y2) の署名を合成できることを指す．

Schnorr 署名

現在，Bitcoin で活用されている署名は ECDSA 署名（Elliptic Curve Digital Signature Algorithm）である．ECDSA 署名も Schnorr 署名も，楕円曲線 E 上の点 G を用いて位数 q の巡回群を構成し，秘密鍵 x に対して Y＝xG が公開鍵となる．どちらの方式も，hash 関数 h とメッセージ m に対して署名時に乱数 r を生成する．ECDSA の署名要素 s は s＝(h(m)＋[rG]x))/r mod q であらわされる．なお，ここで [rG] は，点 rG の x 座標の値を示す．

Schnorr の署名要素 s は s＝r－h(m||[rG])x mod q となる．ここで, || は連結である．どちらも署名データを (s, rG) とする☆6．

ECDSA 署名の検証は，srG＝h(m)G＋[rG]Y であるかどうかを確認する．Schnorr 署名の検証は sG＝R－h(m||[R])Y を確認する．

Schnorr の方式の準同型性について述べる．秘密鍵 x1, x2 と対応する公開鍵 Y1, Y2 に対して，署名者 1 と署名者 2 がそれぞれ乱数 r1, r2 を生成し，お互いに R1＝r1G, R2＝r2G を送り合った上で，下記のデータを作成する．

s1' ＝ r1－h(m||[r1G＋r2G])x1 mod q

s2' ＝ r2－h(m||[r1G＋r2G])x2 mod q

このとき，Schnorr 署名の準同型性により署名 (s1'＋s2', r1G＋r2G) は，公開鍵 Y1＋Y2 に対する正しい署名になっている．

これは，Y1 と Y2 のマルチシグ口座に対する署名が 1 つの署名で構成できるということと，お互いに秘密鍵をもらさずに相互に正しく署名を生成していることが確認できることを意味している．

Discreet Log Contract

次に，この Schnorr 署名を応用して，従来の Bitcoin の script では表現できなかった支払条件を安全に付与することができる Discreet Log Contract[4] と名付けられた方式を紹介する．Bitcoin の script を使わずに支払条件を規定できるため scriptless script と呼ばれる方式の 1 つである．

実現する契約例

この方式で実現する支払条件は，A と B がたとえば，今週末の金曜日に Z 社が発表する価格で取引をする，ということをその週の月曜日に契約する方式である（図 -7）．

☆6 Schnorr の署名データを (s, c=h(m||[rG])) の対とする場合もある．

Z社の役割は，発表する価格Pに対して，Z社の署名を付けることと，そのとき利用する署名に使う乱数rについて，R＝rGをあらかじめ公開しておくことである．

準備

まず，Micro Payment Channelのように，AとBのマルチシグ口座を設立し，そこにAとBは同額（M BTC）をデポジットする．以後，この口座をデポジット口座と呼ぶ．週末の発表価格の可能性がP1からPnである場合，Aは各Piについてトランザクションを作り自分の署名を付与し，Bに送付する．すなわち，Aは，デポジット口座からBのPiマルチシグ口座にM＋Pi BTCを，残りのM-Piを自分の口座に移動させるトランザクションを作り自分の署名を付与してBに渡す．Bは受け取ったデータの正しさを確認する．

ここでBのPiマルチシグ口座とは，Bの公開鍵と価格Piから生成される口座であり，Z社がRを用いてPiの署名を発表したら[☆7]，Bがその口座から資金を移動できるものである．

Bの Pi マルチシグ口座の作り方

変形Schnorr署名を用いる場合，会社Zの公開鍵Yzに対して，Zの価格Piに対するRを用いた署名siというのは，siG＝R－h(Pi||[R])Yzを満たす．この右辺は，署名siを知らなくても作成できるものである．そこで，Piのマルチシグ口座は，Bの公開鍵YBに，この右辺の値，すなわちsiGを加えたものとする．これらは公開されているものなので，誰でも作成することができる．

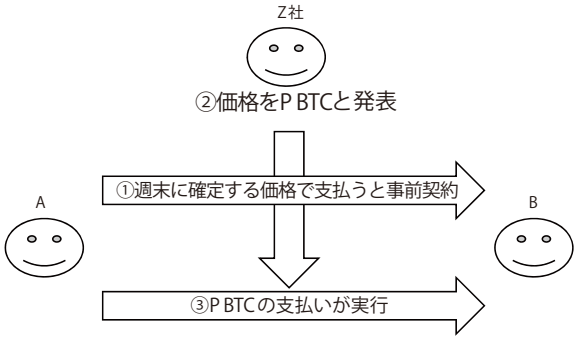

図-7　Discreet Log Contract による契約例

☆7 Z社はTrusted Third Partyであるという仮定を置いており，AまたはBと結託して不正な価格を発表することは想定していない．

価格 Pi の発表前の不正防止

Aは準備したトランザクションにAの署名をしてBに渡す．この準備の段階ではBにはメリットはない．これらのトランザクションにBが署名を付与して台帳に記載することは自由に行えるが，A,Bマルチシグ口座の残高を考慮すると，このうち1つの移転しか有効にならない．価格が発表される前に，どれか1つのトランザクションに自分の署名を追加して台帳に記載しても，その価格が発表されなければ，そのPiマルチシグ口座から自分の口座へと資金移動ができないので，塩漬けになってしまう．したがって，この時点ではBは何も不正を働くことはできないのである．

価格 Pi の発表後の手続き

会社Zが週末の金曜日に決定した価格Pkの署名skを，事前に公表したRを用いて発表すると，Bは，デポジット口座から自分のPkマルチシグ口座にM＋Pk入金されるトランザクションに署名し，台帳に記載する．さらに，この口座から自分の口座に戻すトランザクションを発行する必要がある．

この移動するためにはBのPkマルチシグ口座の公開鍵に対する秘密鍵が必要になる．BのPkマルチシグ口座の公開鍵とは，Bの公開鍵YBにskGを加えたものであり，その秘密鍵はBの秘密鍵にskを加えたものであるため，Bは容易に計算することができ，無事所定額を入手できることになる．AはBがこれらの処理を実行してくれたら，残高が自分の口座に振り込まれる．

持ち逃げ対策

ここで，どちらかが正しく処理を実行しなかった場合に備えた対策について述べる．まず，Aが正しく処理を実行しなかった場合に備えて，BもAと対称の同様のトランザクションを作成してAに送る．そして，A, Bともにトランザクションがそ

ろったところで，デポジット口座に入金をする．また，一方が間違ったトランザクションを発行してデポジット口座の金額を塩漬けしてしまった場合に備えて，一方の Pi マルチシグ口座からの引き落としが一定時間なく，タイムアウトした場合には，自分の口座に取り込めるようなトランザクションにする，という工夫が必要である．

技術と普及について

Bitcoin の希少性から，暗号資産として保有したり，投資目的で売買したりする人が増え，その人たち向けに，自身が保有する Bitcoin を円やドルと交換する交換業者がサービスを開始している．

2008 年にサトシナカモトが考案した Bitcoin は粗削りだがあまりに衝撃的であった．それ故，さまざまな改良を加えようとする技術の発展の流れと，粗削りなまま実際に使っていこうとする社会の流れがあった．技術が広く普及すると，その技術を改良することが難しくなる．そのため，Bitcoin の改良を試みるさまざまな亜種が出現する一方，元々の Bitcoin はそのままで，それにアドオンする形で課題を解決する方式が乱立している．必ずしも技術者が良いと思う方式が広がるわけではない中で，これからも Bitcoin 技術の改良は多産多死で続いていくのであろう．

参考文献

1) Nakamoto, S. : Bitcoin : A Peer-to-Peer Electronic Crash System (2008).
2) 佐古和恵：フィンテック：2. 透明性と公平性を実現するブロックチェーン技術，情報処理，Vol.57, No.9, pp.864-869（Sep. 2016）.
3) Poon, J. and Dryja, T. : The Bitcoin Lightning Network : Scalable Off-Chain Instant Payments, https://lightning.network/lightning-network-paper.pdf (2016).
4) Dryja, T. : Discreet Log Contracts, https://adiabat.github.io/dlc.pdf (2019).

（2019 年 10 月 8 日受付）

佐古和恵（正会員） KazueSako@nec.com
京都大学理学部（数学）卒業．セキュリティとプライバシを両立させる電子投票，電子抽選，匿名認証などの研究に従事．ISO TC307 ブロックチェーンと分散台帳技術　国際エキスパート．日本学術会議連携会員．博士（工学）．

古川　諒（正会員） rfurukawa@nec.com
2008 年東京工業大学総合理工学研究科博士前記課程修了．同年 NEC 入社．以来，アクセス制御，プライバシ保護，ブロックチェーンの研究に従事．

中川紗菜美 sanami-nakagawa@nec.com
筑波大学大学院システム情報工学研究科リスク工学専攻修了後，NEC にて，ブロックチェーン応用などの研究開発に従事．