IMES DISCUSSION PAPER SERIES

ブロックチェーンを利用した暗号資産の安全性と匿名性：原理と限界

大塚（おおつか） 玲（あきら）

Discussion Paper No. 2021-J-4

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

日本銀行金融研究所

〒103-8660 東京都中央区日本橋本石町 2-1-1

日本銀行金融研究所が刊行している論文等はホームページからダウンロードできます。
https://www.imes.boj.or.jp

IMES Discussion Paper Series 2021-J-4
2021 年 6 月

ブロックチェーンを利用した暗号資産の安全性と匿名性：原理と限界

大塚（おおつか） 玲（あきら）*

要　　旨

本論文では暗号資産に用いられる暗号プロトコルのセキュリティ特性（安全性、匿名性）について、原理と技術的な限界の解説を試みる。特に産業イノベーションが先行している自由参加型ブロックチェーンについて、最近の学術研究に基づいた結果を改ざん不可能性の観点から評価する。匿名性については、既存の暗号資産技術をベースに開発された Mixing による手法と、暗号資産技術自体を再構築した zk-SNARKs 等の暗号技術を用いた手法について、それらの利点と限界を議論する。また、一般取引向けのデジタル通貨への適用可能性の観点から、現状の技術の整理・評価を試みる。

キーワード：暗号資産、セキュリティ、ビットコイン、ブロックチェーン、匿名性、改ざん不可能性、暗号プロトコル
JEL classification: E42、O33、O36

* 情報セキュリティ大学院大学教授（E-mail: otsuka@iisec.ac.jp）

本稿は、筆者が日本銀行金融研究所客員研究員の期間に行った研究をまとめたものである。本稿の作成に当たっては、宮地充子教授（大阪大学）から有益なコメントを頂戴した。また、全編に渡り日本銀行金融研究所の菅 和聖氏より非常に熱心なご校閲を頂いた。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者個人に属し、日本銀行の公式見解を示すものではない。また、ありうべき誤りはすべて筆者個人に属する。本稿は、2021 年 2 月 1 日時点の情報を元に執筆された。

目 次

1　はじめに

本稿ではブロックチェーン[1]で実現される暗号資産の安全性および匿名性について、その原理と技術を解説し、それぞれの技術について原理的に示唆される限界を議論する。

日本銀行より先に公表された資料[2]でも、中央銀行デジタル通貨 (CBDC: Central Bank Digital Currency) を「安心して使える」ものとするためには偽造抵抗力を確保し、各種不正を排除する必要があることが指摘されている。他にも，暗号資産交換所における暗号資産の窃取や、DAO 事件に代表されるスマートコントラクトの脆弱性に係るセキュリティリスクも懸念されているが、それらの話題は別稿に譲り、本稿ではブロックチェーンそのものに十分な偽造抵抗力があり、各種不正を排除する能力を備えているかを確認することを目的とする。

よく知られているように、ブロックチェーンには、大きく分けて許可型ブロックチェーン (Permissioned Blockchain) と自由参加型ブロックチェーン (Permissionless Blockchain) がある。いずれもブロックチェーンを冠しているが、それらの技術を構成している原理は大きく異なる（図 1 を参照）。

許可型ブロックチェーンは、ビザンチン合意[3](Lamport, Shostak, and Pease [1982]) から PBFT[4](Castro and Liskov [1999], Castro [2001]) に至る一連の研究を源流とする技術群であり、メンバーの総数を固定し、不正ノード (攻撃者) が一定数以下に抑えられることを前提とする。すなわち、攻撃者の数が予め定められた数を 1 人でも上回ると合意形成が保証されない。許可型ブロックチェーンの原理的な特徴は、ノードの中から忠実なリーダーを選出する過程に力点が置かれることにある。具体的には、ノードが持ち回りでリーダーになり、最新の取引のリストを含めたブロックを生成する。不正ノードを除く圧倒的多数の正しいノードがこのブロックを投票により支持すれば、同じリーダーが引き続き次のブロックを生成する。否決されれば、リーダーは罷免され、次のノードがリーダーに就任し、ブロックの生成を継続する。このように、許可型ブロックチェーンではリーダーの選出に投票が用いられることから、議決に必要な賛成票数を事前にノード間で合意しておく必要がある。

これに対して、自由参加型ブロックチェーンは、ビットコイン (Nakamoto [2008]) を起源とする新しい合意形成アルゴリズムであり、原理的な特徴は、正しいブロックの選別に力点が置かれることにある。自由参加型ブロックチェーンのブロックは計算パズルの解 (Proof of Work:PoW) になっており、解 (PoW) を見つけるにはブロック内の変数であるノンス（Nonce）をランダムに変化させて、当該ブロックのハッシュ値が条件を満たすまで偶然に頼って探すしかない。これを採掘 (マイニング) という。ビットコインでは平均 10 分でマイニングに成功するように計算パズルの難度が設定されており、マイニングに成功すれば高額の報酬 (執筆時の 2021 年 2 月 1 日時点では 6.25BTC ≈2200 万円) が得られる。誰でもマイニングに参加できるため、インターネット上に計算パズルの解 (=ブロック) が次々に公開される。さらに、各ブロックは取引の集合と先行するブロックへの参照を含めなければならないことになっており、公開されたブロックは、原始ブロックを根とし、最新のブロックを葉とする木構造を構成する。自由参加型ブロックチェーンでは、原始ブロック (根) から最新ブロック (葉) までに最も多くのブロックを含む最長のパス (path) だけが有効とされ、それ以外のパスは無効と

[1]ブロックチェーンは DLT(Distributed Ledger Technology) と呼称されることもあるが、本稿では両者を区別せずブロックチェーンという用語で統一する。

[2]日本銀行、「中央銀行デジタル通貨に関する日本銀行の取り組み方針」、2020 年 10 月 9 日公表。

[3]ビザンチン合意は、ビザンチン将軍問題 (補論 A を参照) における合意形成プロトコルを指す。攻撃者が存在する困難な状況下で、分散システムが合意を形成する問題のモデルとして広く参照されている。

[4]PBFT(Practical Byzantine Fault Tolerance) は、ビザンチン合意を達成する代表的なプロトコル (補論 B) である。

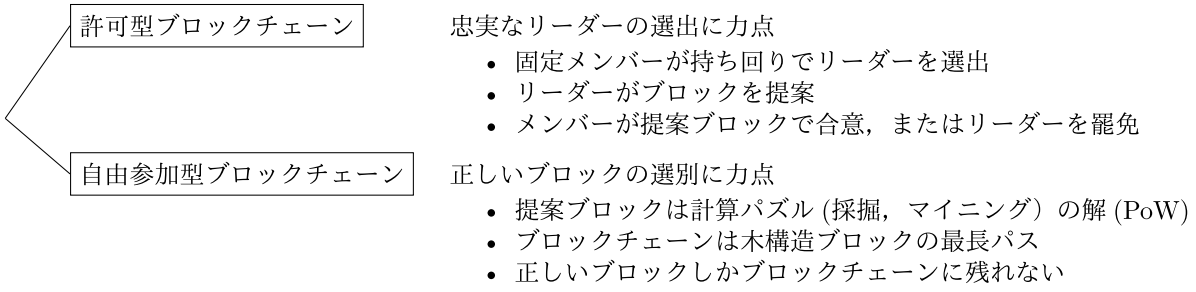


図 1: ブロックチェーンの種類と相違点

するルールを設けている。この結果、ただ 1 つのパスだけが選別され、原始ブロックから最新ブロックに至る一直線のブロックチェーンが選ばれる[5]。このことから、マイニングしたブロックがブロックチェーンに残る確率を上げるためには、最長パスに続くブロックをマイニングしなければならず、これがインセンティブとなって唯一の合意されたブロックチェーンが形成される。このように、自由参加型ブロックチェーンは計算量的なハードルと適切なインセンティブ設計により、柔軟で強靱なシステムを形成している。

現時点で、許可型ブロックチェーンは銀行間での資金決済や、企業間取引など、複数の組織が共同ネットワークを構築する領域への適用が検討されることが多い。他方、自由参加型ブロックチェーンは暗号資産やスマートコントラクトなどのインターネット上での不特定多数の一般利用者間の取引に適用されている。本稿では、一般利用型のデジタル通貨への展開を考える上で選択肢の一つになると考えられる自由参加型ブロックチェーンを中心に考察することにしたい。

以下では、まずブロックチェーンの基本的な原理を簡単に解説し、偽造抵抗力や各種不正への耐性を議論する。次に、匿名性について概念の整理と仕組みを解説し、それらの技術的な制約から来る原理的な限界を考察する。最後に、ブロックチェーンに基づくデジタル通貨に匿名性を付加する際に想定される課題を考察する。

2　ブロックチェーンの仕組み

本節では安全性の議論に入る前の準備として、ブロックチェーンの仕組みをプレイヤーの構成、ブロックチェーンの構造、Proof of Work の 3 点から解説する。

(1)　プレイヤーの構成

ビットコイン型のブロックチェーンでは、取引、ブロック、ブロックチェーンの 3 種類のデータと、利用者、ノード、採掘者の 3 種類のプレイヤーが存在する。利用者間の支払いの取引は、送金者が取引を作成して公開することで開始される。取引は、簡略化[6]すると以下の構造

[5]ブロックチェーンは一般には木（ツリー）構造をとり、ブロックチェーン内に最長のパスが複数存在する状況も想定される。この状況は Fork と呼ばれる。Fork は、後で述べる木構造上の最長パスを選択するアルゴリズムを適用することにより、一定時間後に解消され鎖（チェーン）構造になることが証明されている (Garay, Kiayias, and Leonardos [2015])。

[6]実際のビットコインでは、支払条件が独自のスクリプト (プログラム) で与えられており、最も単純なスクリプトには、送金者のアドレスに対応する公開鍵とデジタル署名を求める条件が記載されている。また、一般には送金者のアドレス、受領者のアドレスは複数個指定されるが、ここでは当分の間、説明を単純にするた

を持つ。

$$\text{取引} = \big(\text{送金者のアドレス}, \text{受領者のアドレス}, \text{金額}\big) + \text{送金者の署名}$$

ここでアドレスとは、公開鍵のハッシュ値を指す。受領者は、取引の都度、受取用の公開鍵と秘密鍵のペアを生成し、事前に受取用のアドレスを送金者に伝えておくことで、送金者が上記の取引を作成して公開する。取引の公開とは、ノード[7]が構成する P2P ネットワークを用いて全ノードに取引を行き渡らせることで行われる。利用者間で発生した全ての取引は各ノードに蓄えられ、採掘者に供給される。

採掘者の役割は、ノードに蓄えられた取引をまとめてブロックを作成し、後述のマイニングを行うことでブロックチェーンを形成することである。取引がブロックに含められ、ブロックチェーンに登録されたことを、取引が「確定する (confirmed)[8]」という。ブロックは取引の実行順序を決定し、ブロックチェーンはブロックの実行順序を決定することから、ブロックチェーンにより原始状態から現在に至る全ての取引の実行順序が確定する。すなわち、唯一のブロックチェーンに全採掘者が合意することは、確定した取引の集合とその実行順序で合意することと同義である。全てのアドレスの資産高が 0 の状態を原始状態とすると、原始状態からブロックチェーンを辿ることにより、全てのアドレスの資産高が確定する。

ビットコイン等の暗号資産では新しいブロックが生成される度に一定額のコインが新規発行され、マイニングに成功した採掘者にその全額が与えられる。このマイニング報酬がブロックチェーンを維持する根源的なインセンティブである。

(2) ブロックチェーンの構造

ビットコインにおけるブロックは、下式に示す 3 項組である。

$$B_i = \langle \mathrm{H}(B_{i-1}), H_i, N_i \rangle$$

i 番目のブロック B_i は、先行ブロックのハッシュ値 $\mathrm{H}(B_{i-1})$、登録する取引の順序付リストのハッシュ値 H_i、ノンス N_i の 3 項組で定義される。実装上は、さらにタイムスタンプなどの補助情報が追加されるが、本節の範囲では不要であるため省略する。各ブロックは図 2 に示すように、前ブロックのハッシュ値を次のブロックに含めることで、鎖構造 (チェーン) をとる。ここで、H は暗号学的ハッシュ関数を表し、256bit のビット列を出力する SHA-256 もしくは SHA-3 を想定している[9]。H_i は、一定数の取引の集合を (実用上) 一意に表すハッシュ値であり、ビットコインではマークル木 (Merkle Tree, Merkle [1982]) が用いられている。マークル木は図 2 に示すようにトーナメント方式でハッシュ値を取ることにより、複数の取引の順序付リストについてのハッシュ値を計算するアルゴリズムである。

図 2 の例では、まず取引 $\tau_1, \ldots, \tau_8$ の各組のハッシュ値 $h_{12} = \mathrm{H}(\tau_1, \tau_2)$、$h_{34} = \mathrm{H}(\tau_3, \tau_4)$、$h_{56} = \mathrm{H}(\tau_5, \tau_6)$、$h_{78} = \mathrm{H}(\tau_7, \tau_8)$ を用いて、マークル木は以下の式で計算される。

$$H_i = \mathrm{H}\left(\mathrm{H}(h_{12}, h_{34}), \mathrm{H}(h_{56}, h_{78})\right)$$

マークル木の根にあたるハッシュ値をマークルルート (根) と呼ぶ。この手続きを再帰的に繰り返すことにより、任意の数の順序付リストに対するマークルルートを計算することができる。

めに 1 入力 1 出力の取引のみを考える。

[7] ビットコインでは 10,000 台以上のノードが存在していると言われる。

[8] ブロックチェーンに登録されたばかりのブロックは Fork 等により不安定な状態にある。このため、取引が真に確定するまでには、当該ブロックに続き、さらに数ブロックが登録されるまで待つ必要がある。

[9] 暗号学的ハッシュ関数については補論 C を参照されたい。

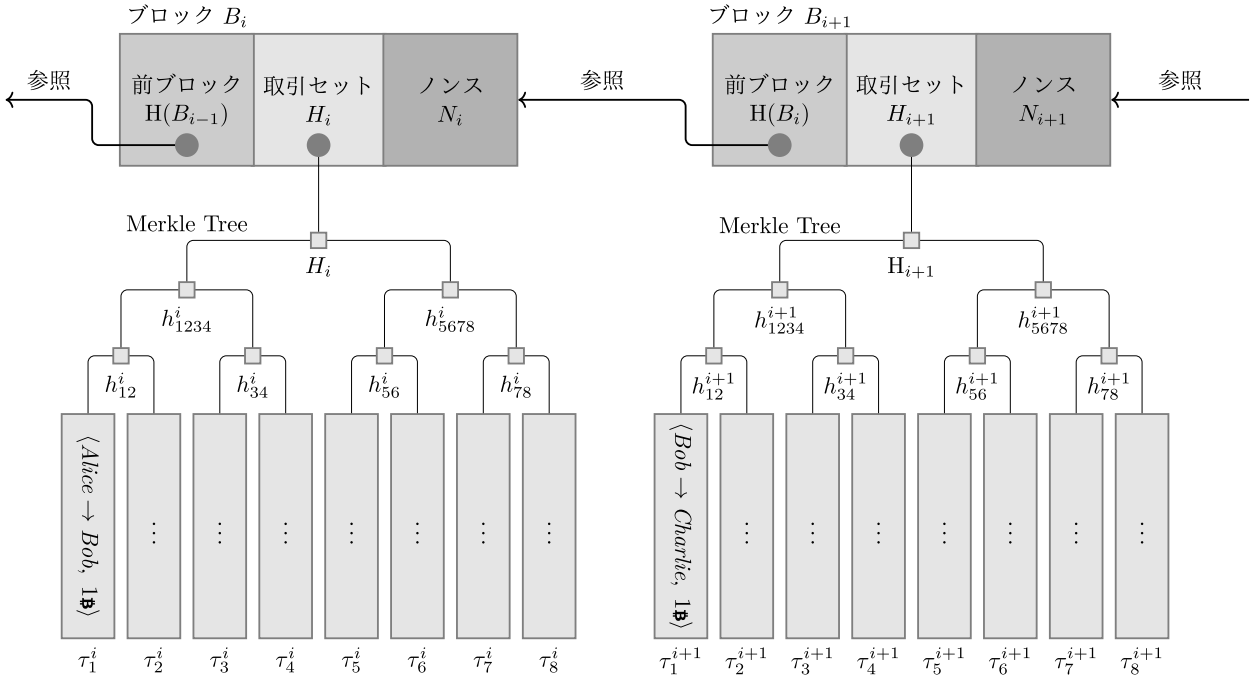


図 2: ブロックチェーンの模式図。各ブロックは $B_i = \langle \mathrm{H}(B_{i-1}), H_i, N_i \rangle$ の形をしている。各ブロックのハッシュ値は PoW により一定値以下のものだけが有効とされる。ブロックには取引のマークルルートが含められ、大量の取引の集合を 1 つのハッシュ値で指示している。

(3) Proof of Work

Proof of Work は、先行するブロックのハッシュ値 B_{i-1} と、マークルルート H_i が与えられたときに、その時点での難度 (D) に応じて、

$$\mathrm{H}(B_i) = \mathrm{H}(\langle \mathrm{H}(B_{i-1}), H_i, N_i \rangle) < \frac{2^{256}}{D} \tag{1}$$

を満たす N_i を見つける仕事である。ここで、H は任意長のビット列を入力とし、区間 $[0, 2^{256})$ のランダムな整数値を出力するハッシュ関数を表す。

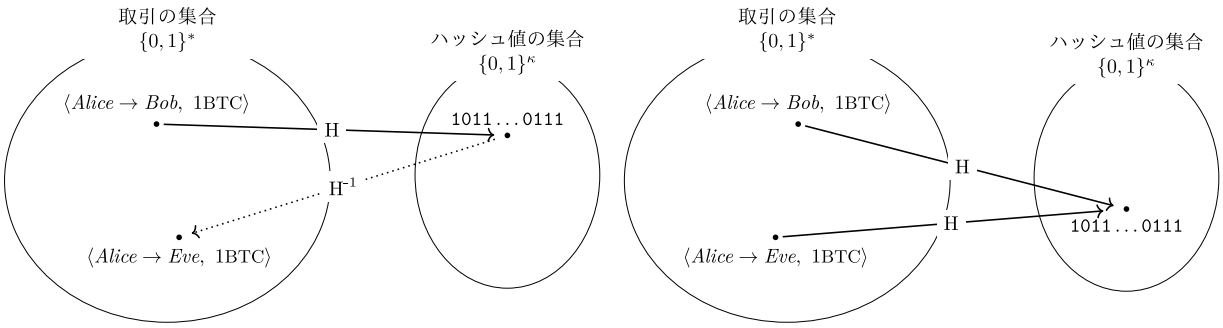
暗号資産で用いられる SHA-256 等の一方向性ハッシュ関数の出力の各ビットは一様ランダムと信じられており、この式を満たす N_i を見つけるには網羅的に探索する以外に効率的な方法は知られていない。ビットコインでは約 10 分間隔でマイニングに成功するように難度 D が制御されており、現在の難度は $D \approx 10^{22.9}$ に設定されている。したがって、1 つのノンスを用いた試行で成功する確率 p は

$$p = \frac{1}{D} \approx \frac{1}{10^{22.9}}$$

で表される。

他方、現在の世界のハッシュレート[10]は 1 垓回 $(10^{20}$ 回$)$/秒と観測されており、10 分間に平均して 600×10^{20} 回のハッシュ関数が実行されている。したがって、10 分程度でマイニン

[10] 全採掘者が 1 秒間に試行するハッシュ計算の回数。



(a) ハッシュ関数の逆関数による取引の改ざん。改ざんの目標となる取引を確定し、その取引と同じハッシュ値を持つ別の取引を探索する。

(b) ハッシュ関数の衝突による取引の改ざん。予め同じハッシュ値をとる複数の取引を用意し、一方をブロックチェーンに登録し、後に差し替える。

図 3: マークル木に含まれる取引の改ざん

グに成功する計算になる。現在、全世界で原発 7 基分に相当する 7.46GW[11]の電力がこの計算に投入されている。それほどのパワーを投入しなければ発見できないことから、仕事量の証明として PoW を利用できる。

正しい採掘者は矛盾[12]を含まない最長のブロックチェーンを独立に見つけて、正しいブロックチェーンと見做す。このルールにより、常に最長のブロックチェーンに後続するブロックの採掘にマイニングパワーが集中するため、たとえ複数の同じ長さのブロックチェーン（Fork）が存在しても、長さに違いが生じた時点で長い方のブロックチェーン 1 本に収斂する。

3 ブロックチェーンの安全性

ブロックチェーンが安全であるとは、ブロックチェーンおよび各ブロックに含まれる取引の改ざんが不可能であり、かつ、ブロックチェーン内に含まれる全てのブロックの内容と順序について全採掘者で合意が成立することである。前者は、次節で述べるようにデジタル署名や暗号学的ハッシュ関数などの暗号技術で達成されるが、後者はブロックチェーンの合意を乱す Fork が最大の脅威となる[13]ことから、ブロックチェーンの各種パラメータ値から Fork 長についての理論限界を与えることが重要となる。

したがって、本節では、まずブロックや取引の改ざんが暗号技術でいかに守られているかを解説する。続く節では、まず Fork 長の限界に迫る攻撃として知られる利己的マイニング (Selfish Mining RHorning [2010], Eyal and Sirer [2014], Eyal and Sirer [2018]) を解説し、そのような攻撃の存在も踏まえて、Fork 長の理論限界を与える共通プレフィックス定理 (Common Prefix Theorem, Garay, Kiayias, and Leonardos [2015]) の解説に進む。

[11] ビットコインエネルギー消費指標：https://digiconomist.net/bitcoin-energy-consumption/

[12] ブロックチェーンが矛盾を含まないとは、各ブロックに含まれる全ての取引のデジタル署名等の出力条件が満たされ、かつ取引を先頭から順に実行した際に残高が負の値をとるアドレスを 1 つも含まないこと、さらに各ブロックのハッシュ値が Proof of Work になっていることである。

[13] ブロックチェーンにおける Fork は、同じ長さの最長ブロックチェーンが複数存在する状況を指し、Fork が存在すると最終的にいずれのブロックチェーンが残るかに応じて、取引の実行順序が変化する不安定な状況が生じる。

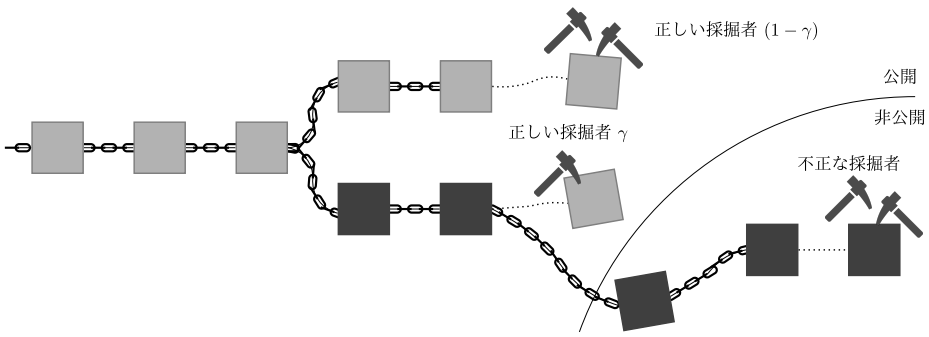


図 4: 利己的マイニング戦略の概要。利己的マイニングは、新しく発見したブロックの公開を意図的に遅らせることで、所有する計算能力で得られる以上の利益を得ようとする行為である。図では黒いブロックが不正な採掘者により発掘されたブロックを示している。正しい採掘者は両者のブロックを区別できないため力が分散してしまい、確率 γ で利己的マイニングにより採掘されたブロックに続くブロックを採掘する。

(1) ブロックの改ざん不可能性

マークル木を改ざんして別の取引に差し替える攻撃の可能性を考える。マークルルート H_i は既にブロックに記載されており、この値を改ざんできないと仮定すれば、マークル木の葉に含まれる取引を差し替えることにより、同一の H_i で異なる取引セットを指示できれば攻撃は成功である。

図 3 に示すように、2 つの方法が考えられる。(a) 暗号学的ハッシュ関数の逆関数による取引の改ざんが考えられる。この攻撃は、先に改ざんの目標となる取引を確定し、その取引と同じハッシュ値を持つ取引に差し替えるというものである。この攻撃については、対象となるハッシュ関数が安全であり一方向性を有していれば、同じハッシュ値を持つ取引を発見することは計算量的に困難であるため、実行することが困難である。また、(b) ハッシュ関数の衝突による取引の改ざんも考えられる。この攻撃は、予め同じハッシュ値を持つ複数の取引を準備しておき、一方の取引をブロックチェーンに登録しておいて、後に他方の取引に差し替えるというものである。この攻撃についても、暗号学的ハッシュ関数の衝突困難性により、同一ハッシュ値を出力するような複数の取引を発見することは計算量的に困難であるため、実行することが困難である。これらの議論は、マークル木の各節点の差し替えについても同様に適用できる。

以上のような議論から、ブロックに H_i が固定され、実装されたハッシュ関数が安全である限り、取引セットを改ざんすることは、計算量的に困難であると考えられる。したがって実用的には、H_i が取引セットをその順序も含めて一意に指定すると考えてよい。このことから、ブロックチェーンの改ざん不可能性は、ブロックチェーンに含まれるブロックの差し替えが可能か否かに帰着される。

(2) 利己的マイニング

イ 利己的マイニングの戦略

Fork が生じると、図 4 に示すように、正しい採掘者のマイニング能力が分散する。この状況を悪用した攻撃に、利己的マイニングが知られている。以下では、正しい採掘者の採掘したブロックを正しいブロックと呼び、利己的マイニング戦略を採る不正な採掘者が採掘したブ

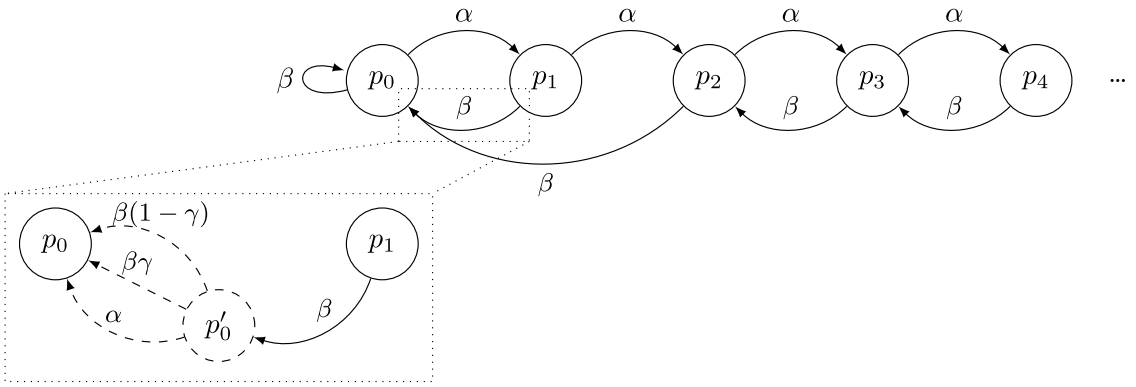


図 5: 利己的マイニング戦略のマルコフ過程モデル。不正な採掘者の隠し持っているブロックが 3 個以上の状態 $(p_3, p_4, \ldots)$ において、(確率 β で) 正しい採掘者による新しいブロックの採掘が成功した場合には、同ブロックが公開されると同時に、不正な採掘者は隠し持ったブロックの 1 つを公開する。これにより状態 $(p_2, p_3, \ldots)$ に遷移し、Fork 状態が維持される。また、不正な採掘者が隠し持っているブロックが 2 個の状態 (p_2) で、(確率 β で) 正しい採掘者がマイニングしたブロックを公開した場合には、不正な採掘者は隠し持っている 2 個のブロックを一気に公開して状態 p_0 に遷移することで勝利を確定し、その間の全てのマイニング報酬を獲得する。不正な採掘者が隠し持っているブロックが 1 個の状態 (p_1) で、(確率 β で) 正しい採掘者がマイニングしたブロックを公開した場合には、左下の破線枠内に示すように 3 通りの遷移を経て、不正な採掘者が隠し持っているブロックが 0 個の状態 (p_0) に遷移する。3 通りの遷移のいずれを通るかにより、不正な採掘者が獲得するマイニング報酬は異なる。

ロックを不正なブロックと呼ぶ。

利己的マイニングは、新しく発見したブロックの公開を意図的に遅らせることで、所有する計算能力で得られる以上の利益を得ようとする行為である。具体的には、以下の手順で攻撃を実施する。

1. マイニングに成功してもブロックを公開せずに保持しておき、マイニングしたブロックに続く後続ブロックのマイニングに取りかかる。
2. 保持しているブロックが 2 個以上になった場合も、続けて後続ブロックをマイニングする。
3. 正しい採掘者がマイニングに成功し、新しいブロックの拡散を開始した時に、保持していたブロックを公開し、Fork を維持する。
4. 手元に保持しているブロックが残り 1 つになった場合は、直ちにそれを公開する。

利己的マイニングの採掘者は、確率の揺らぎにより一時的に連続してマイニングに成功した時にそれらのブロックを隠し持っておき、Fork を長く維持する。最後に隠し持っていたブロックを一気に公開することで、最終的にマイニング競争での勝利を確定し、Fork が生じている間の全てのマイニング報酬を得ることができる。

ロ　マルコフ過程モデルによる利己的マイニングの解析

定常状態の確率　この節では、利己的マイニング戦略をとる不正な採掘者により採掘されたブロックが、採掘されたブロック全体のうち、どの程度の割合を占めるかを、図 5 に示すマ

ルコフ過程モデル[14]を用いて定量的に解析する。$\alpha + \beta = 1$ とし、不正な採掘者は確率 α で次ブロックのマイニングに成功、正しい採掘者は確率 β で次ブロックのマイニングに成功する。状態 p_i は、不正な採掘者が i 個のブロックを隠し持っている状態を表し、同時に p_i でその状態の存在確率を表す。

定常状態における各状態の存在確率 $p_0, p_1, \ldots$ は、

$$\sum_{i=0}^{\infty} p_i = 1, \quad p_1 = \alpha p_0, \quad \beta p_{i+1} = \alpha p_i \quad (i \geq 1)$$

で表される。$\alpha + \beta = 1$ として、これを解くと一般項

$$p_0 = \frac{2\alpha - 1}{\alpha^2 + \alpha - 1}, \quad p_1 = \frac{\alpha(2\alpha - 1)}{\alpha^2 + \alpha - 1}, \quad p_i = p_1 (\frac{\alpha}{1 - \alpha})^{i-1} \quad (i \geq 2) \tag{2}$$

が得られる。

状態間の遷移で生じるブロックの個数　次に、状態間の遷移において、不正なブロックと正しいブロックが生成される個数を考える。このとき、状態 p_1 での遷移は、やや複雑である。すなわち、状態 p_1 において、確率 β で正しい採掘者がマイニングに成功した場合に生じる状態遷移（β-遷移）では、不正な採掘者が同時に隠し持っている 1 つのブロックを公開し、隠し持っているブロックが 0 個の状態に遷移する。しかし、この状況では Fork が存在している。Fork が存在しない本来の p_0 と区別するために、この状態を仮に p_0' と呼ぶ。図 5 の破線枠内に示すように、次のブロックが採掘された時点で、どの採掘者が採掘に成功したかによらず Fork が解消して状態 p_0 に遷移する。この遷移は次の 3 通りに分かれる。

1. 不正な採掘者がマイニングに成功して p_0 に遷移する状況。この状況は確率 α で生じ、不正な採掘者の採掘した 2 ブロックが最長のチェーンに追加される。

2. 正しい採掘者がマイニングに成功して p_0 に遷移し、Fork 状況にあるブロックのうち、正しい採掘者の採掘した方のブロックに続けてさらにもう 1 つの正しいブロックのマイニングに成功する状況。このマイニングに正しい採掘者のうち $(1 - \gamma)$ が参加しているとすると、この状況は確率 $\beta(1 - \gamma)$ で生じる。この際、2 ブロックが追加されるが、不正な採掘者のブロックは最長のチェーンに含まれない。

3. 正しい採掘者がマイニングに成功して p_0 に遷移し、Fork 状況にあるブロックのうち、不正なブロックに続けてブロックのマイニングに成功する状況。このマイニングに正しい採掘者のうち γ が参加しているとすると、この状況は確率 $\beta\gamma$ で生じる。この際、正しいブロックと不正なブロックがそれぞれ 1 個ずつ、合計 2 ブロックが最長チェーンに追加される。

利己的マイニングによる利得　ブロックの追加は β-遷移でしか起こらず、状態 p_1 と状態 p_2 における β-遷移では 2 個のブロックが生成されることから、1 回の β-遷移で生成されるブロック数の期待値は $1 + p_1 + p_2$ で表される。他方、不正な採掘者が生成するブロック数は、

[14] マルコフ過程モデルは、状態間に遷移確率が定義され、初期状態から確率に従って状態遷移を繰り返す確率過程のことをいう。特に、次の状態への遷移確率が現在の状態だけから決まり、過去の状態と無関係に定まる（マルコフ性）という特徴を持つ。

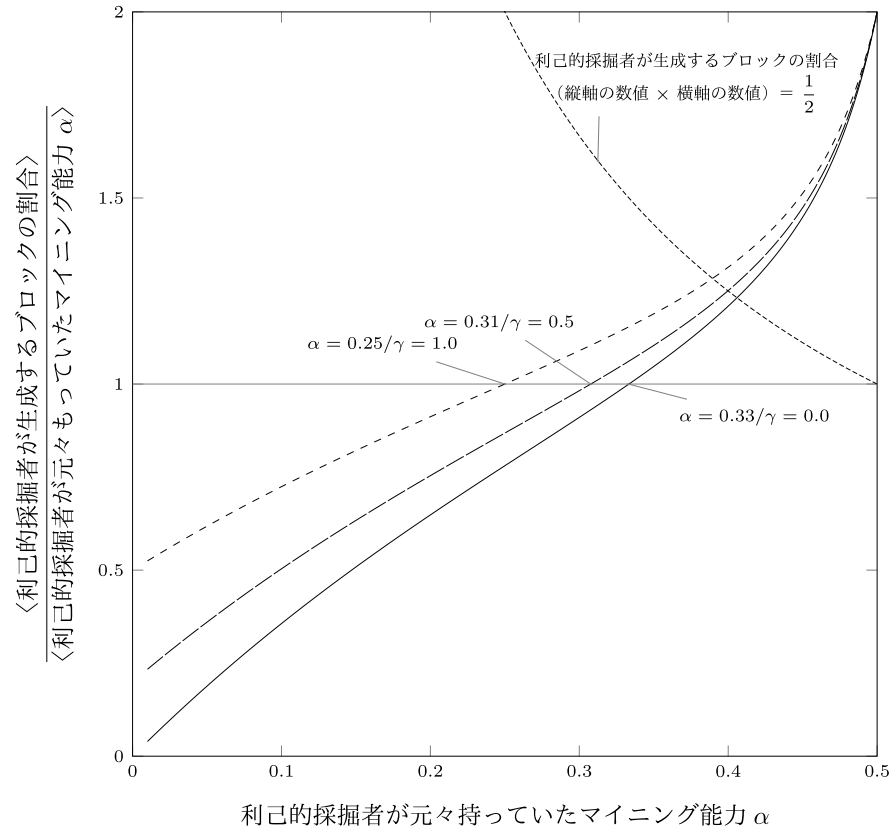


図 6: 利己的マイニング戦略による利得。縦軸は、不正な採掘者が利己的マイニング戦略を採ることで得られる利得が、同戦略を採らない場合に比べて何倍であるかを表す。これが 1 倍を超えた領域では、不正な採掘者は利己的マイニング戦略を採ることにより追加的な利得が得られる。また、不正な採掘者が生成するブロックが 1/2 となる曲線の右上の領域では、不正な採掘者は過半のマイニング能力を獲得できる。利己的マイニング戦略を採ることによる追加的な利得は、正しい採掘者の計算能力が全体のパラメータ γ のばらつきを映じて 25—33% を越えたところから生じる。40% 程度では、いずれの γ の値の場合でも、概ね利得が 1.25 倍を越えるため、不正な採掘者は全マイニング能力の過半を獲得できる。

p_0 での β-遷移では 0 個、p_2 での β-遷移では 2 個、p_1 での β-遷移では前述のように $0 \sim 2$ 個に分かれ、それぞれの確率は $\beta(1-\gamma)$、$\beta\gamma$、α である。その他の状態では 1 個である。したがって、その期待値は $1 - p_0 + p_2 + \alpha p_1 - \beta(1-\gamma)p_1$ で求められる。以上をまとめると、生成されるブロック全体のうち不正な採掘者が生成するブロックの割合は、各定常状態の存在確率 (2) 式を用いて、以下で求められる。

$$\frac{1 - p_0 + p_2 + \alpha p_1 - \beta(1-\gamma)p_1}{1 + p_1 + p_2} = \frac{\alpha(1-\alpha)^2(4\alpha + \gamma(1-2\alpha)) - \alpha^3}{1 - \alpha(1 + (2-\alpha)\alpha)} \tag{3}$$

ハ　不正な採掘者の計算能力と利得

(3) 式に表される利己的マイニング戦略を採る場合に得られるブロックの割合が、同戦略を採らずに採掘した場合のブロックの割合（＝利己的採掘者が元々持っていたマイニング能力）

を上回れば、利己的マイニング戦略を採ることによる追加的な利得があったことになる。この様子をグラフに描くと、図 6 に示すように $\gamma = 0.5$ [15]の時、不正な採掘者のマイニング能力 α が全体の 31% 以上を占めていれば利己的マイニングによる利得が得られる。この利得の効果により、不正な採掘者のマイニング能力は 40% を少し上回ったところで、利得が 1.25 倍に達し、過半のマイニング能力（$= 40\% \times 1.25$）を獲得できる。

このように、マイニングによる合意形成手法では最長のブロックチェーン以外は無視されるため、ブロックチェーンの先頭部分を改ざんすることは極めて困難である。しかし、過半には及ばずとも全体の 31% 以上の採掘能力を持つ攻撃者が利己的マイニング等の戦略を取れば、長い Fork を維持することで最長のブロックチェーンを複数存在させることが可能であることを見てきた。続く節では、長い Fork が存在する確率を分析する。

(3) 共通プレフィックス定理

ブロックチェーンが Fork している状態は、最長ブロックチェーンについての合意が達成されていない不安定な状態である。Fork 状態では、一方のブロックチェーンでは確定している取引が他方のブロックチェーンでは確定していないといった状況が生じ、この状況が悪用されれば、二重支払い[16]などの不正取引が横行するリスクがある。安全性が確認されている自由参加型ブロックチェーン[17]に対する最も有効な攻撃は、利己的マイニングなどを活用して Fork 状態をできる限り長く維持し、不正取引の成功確率を高めることである。

そこで、正しい採掘者のマイニング能力が不正な採掘者のマイニング能力を一定以上の割合で上回る場合に、取引が事実上確定したと考えてよい範囲を明らかにするために、Fork 長の上限を見積もることが重要となる。これにより、2 人の正しい採掘者が保持するブロックチェーンは、末尾（葉）の k ブロックを取り除くと、互いに他方の保持するブロックチェーンの共通プレフィックスになっている。この k が、事実上取引が覆らないと考えてよいブロックの深さを表す。以下では、k と確率の関係を具体的に見積もる方法を共通プラフィックス定理 (Garay, Kiayias, and Leonardos [2015]) に沿って解説する。ここで、Fork しているブロックチェーンのうち、合意が達成されている部分（共通プレフィックス）を除いた末尾部分の長さ k の最小値を Fork 長と呼ぶ。

ビットコインを初めとするブロックチェーンシステムでは、各ノードが P2P で接続されたネットワークを形成し、このネットワーク上で情報を拡散させることで、全ノードが同じ情報を共有するシステムを採用している。拡散遅延に起因する過渡状態の解析を省くために、この拡散時間よりも十分長い時間をラウンドと呼ぶ離散的な時間単位で考える。全てのノードは採掘者の役割を担い、メッセージの伝達とブロックの生成に寄与する。

イ ラウンド毎の解析

まず、1 つのラウンドにおける不正な採掘者と正しい採掘者による採掘の成功確率について考える。採掘者の総数を n 人とし、このうち t 人が不正であるとする。大多数は正しい（善

[15] $\gamma = 0.5$ は、正しい採掘者が Fork 状態にあるブロックのいずれが不正な採掘者のものが区別できない状況に対応している。この状況を作り出すために、不正な採掘者は正しい採掘者がマイニングしたブロックの拡散を始めたと同時に、隠し持っていたブロックの拡散を始める。

[16] 二重支払い (double spending): 同一資産を複数の取引に使用し、保有資産額以上の利益を得る攻撃。

[17] より正確には、ビットコイン等の Proof of Work 型のブロックチェーンを対象としている。Proof of Stake(QuantumMechanic [2011]) 等の合意形成アルゴリズム基づくブロックチェーンについては、個別の検討が必要である。

良である）と仮定 (Honest Majority Assumption) する。すなわち、

$$t < (1-\delta)(n-t)$$

を仮定する。ここで、δ は正しい採掘者の優位度を表すパラメータである。例えば、$\delta = 0.5$ の時、不正な採掘者の数は全体の 1/3 未満であることを示す。今、最悪時の Fork 長の上限を求めるために、不正な採掘者は全て結託して 1 人の攻撃者として振る舞い、Fork 長をできるだけ長く維持することを目標としていると考える。

i 番目のラウンド r_i で正しい採掘者のいずれかがマイニングに成功する事象を $X_i = 1$ で表し、失敗する事象を $X_i = 0$ で表す。さらに、ラウンド r_i で正しい採掘者が 1 人だけマイニングに成功する事象を $Y_i = 1$ で表す。したがって、常に $Y_i = 1 \implies X_i = 1$ が成り立つ。さらに、不正な採掘者がラウンド i でマイニングに成功する事象[18]を $Z_i = 1$ で表す[19]。これらの事象は、単純にマイニング試行[20]の回数に比例した確率で生じるものとする。p は 1 回のマイニング試行が成功する確率を表し、いずれの採掘者も 1 ラウンド当たり最大 q 回のマイニング試行ができるとする。このとき、正しい採掘者が 1 つのラウンド内でマイニングに成功する確率 f は、

$$(1-f)pq(n-t) < E[X_i] = f < pq(n-t) \tag{4}$$

の範囲におさえられる[21]。

正しい採掘者がちょうど 1 回だけマイニングに成功する確率 $E[Y_i]$ は、

$$f(1-f) < E[Y_i] \tag{5}$$

で下限が与えられる[22]。他方、不正な採掘者が 1 ラウンドでマイニングに成功する確率は、

$$E[Z_i] < \frac{t}{n-t} \cdot \frac{f}{1-f} < (1-\delta)\frac{f}{1-f} \tag{6}$$

で上限が与えられる[23]。

ロ　ラウンド列の解析

次に、λ 個の連続したラウンドの区間 S において維持されうる Fork 長の上限について考える。区間 S で、正しい採掘者がマイニングに成功するブロックの数 $X(S)$ は、その期待値 $E[X(S)] = E[\sum_{i \in S} X_i]$ を中心に揺らぐが、S を十分に長くとれば、中心極限定理により期

[18] 不正な採掘者は、利己的なマイニング戦略等をとる可能性があるため、必ずしもマイニングに成功したラウンドで新しいブロックをネットワークに公開するとは限らない。

[19] この節では、Fork 長が最も長くなり得る最悪ケースについて考察するため、不正な採掘者は互いに協力して実質的に 1 人の採掘者として振る舞う場合を想定している。複数の不正な採掘者が採掘に成功した場合も、互いに協力し合わない場合の Fork 長は、協力して統一行動をとる場合に比べて短くなるため考慮しない。

[20] マイニング試行とは、ランダムに発生させたノンスの候補 N_i が不等式 (1) を満たすか否かを判定することを指し、不等式 (1) が成立する場合に成功とする。

[21] $f = 1-(1-p)^{q(n-t)} < pq(n-t)$, および $\dfrac{f}{1-f} = (1-p)^{-q(n-t)} - 1 > (1+p)^{q(n-t)} - 1 > pq(n-t)$ より導かれる。

[22] $E[Y_i] = \binom{q(n-t)}{1} p(1-p)^{q(n-t)-1}$ および $f < pq(n-t)$ より導かれる。

[23] $E[Z_i] = 1-(1-p)^{qt} < pqt = \dfrac{t}{n-t} pq(n-t) < \dfrac{t}{n-t} \cdot \dfrac{f}{1-f}$ より導かれる。

待値のごく近くの範囲に収まり、適当な $\epsilon > 0$ をとることで、高確率で以下の不等式が成立する。このように、正しい採掘者と不正な採掘者のマイニングの結果が、中心的な期待値から大きく外れない試行を典型実行（typical execution）と呼ぶ。典型実行でない事象が生じる確率は、十分に長い区間 S をとれば無視できるほど小さいとみなせる。

$$(1-\epsilon)E[X(S)] < X(S) < (1+\epsilon)E[X(S)] \tag{7}$$
$$(1-\epsilon)E[Y(S)] < Y(S) \tag{8}$$
$$Z(S) < (1+\epsilon)E[Z(S)] \tag{9}$$

大多数は正しい（善良である）とする仮定（Honest Majority Assumption) とパラメータが適切な条件 $3\epsilon + 3f < \delta$ を満たす仮定の下では、$E[Z(S)] < E[Y(S)] < E[X(S)]$ が成り立つ[24]。したがって、十分に長い区間 S をとれば、

$$Z(S) < Y(S) < X(S) \tag{10}$$

が高い確率で成り立つ (補論 D)。

以上で導かれた典型実行の帰結を踏まえて、ラウンド列において Fork が維持される条件について考察する。正しいブロック採掘のみ（$X_i = 1, Y_i = 1, Z_i = 0$）が、不正な採掘者による隠しブロックがない状態で生じると、Fork が直ちに解消し、最長ブロックチェーンが唯一に定まる。したがって、不正な採掘者は、$Y_i = 1$ の事象が生じる前にマイニングに成功し ($i' \leq i$ に対して $Z_{i'} = 1$)、隠し持っていたブロックをネットワークに公開することにより、Fork の解消を阻止しなければならない。これが、十分に長い区間 S で成功しつづけているとき、不正な採掘者は $Y(S)$ と同数以上のブロックのマイニングに成功していなければならないことになる。すなわち、

$$Z(S) \geq Y(S)$$

が成立することになる。しかし、これは Honest Majority Assumption の下での典型実行の条件式 (10) と矛盾する。したがって、Fork の延長は、典型実行が仮定できず、揺らぎの影響が大きく出る、S の短い区間に限られる。このもとで、典型実行と見なすことができるような十分な長さの S をとり、$k = E[X(S)]$ と定義する。このとき、k ブロック以上の Fork が生じる確率は、ある一定以上の k については、無視できるほど小さいと評価できる（詳細は補論 D を参照)。したがって、最長ブロックチェーンの末尾から k ブロックを除けば、正しい採掘者は同一のブロックチェーンで事実上合意しているとみなしてよいこととなる（合意していない確率は無視できるほど小さい)。

図 7 に Fork の発生と解消の様子を図示する。図 7 では、「○」で、当該ラウンド r_i での正しい採掘者による単一ブロックのマイニング成功（$X_i = 1$、かつ $Y_i = 1$）を表し、「∞」で正しい採掘者による複数ブロックのマイニング成功（$X_i = 1$、かつ $Y_i = 0$）を表し、「●」で不正な採掘者によるブロックのマイニング成功（$Z_i = 1$）を表している。また、棒グラフは Fork の長さ k を表し、ケース (a) またケース (b) の成立により Fork が延長する事象を“Fork”と表記している。

図 7 のシミュレーション結果では、ラウンド 1 で、不正なブロックと正しい採掘者のブロックが 1 つずつ生成されて Fork が発生したあと、ラウンド 2 で正しい採掘者のブロック

[24] $E[Y(S)] < E[X(S)]$ は $f > 0$ より $(1-f)f < f$ が成り立つことから導かれる。$E[Z(S)] < E[Y(S)]$ は、$(1-\delta)\frac{f}{1-f} < f(1-f)$ を示せば十分。これは $\delta > 2f - f^2$ と同値であり、仮定 $\delta > 3\epsilon + 3f$ から導かれる (補論 D 脚注 45)。

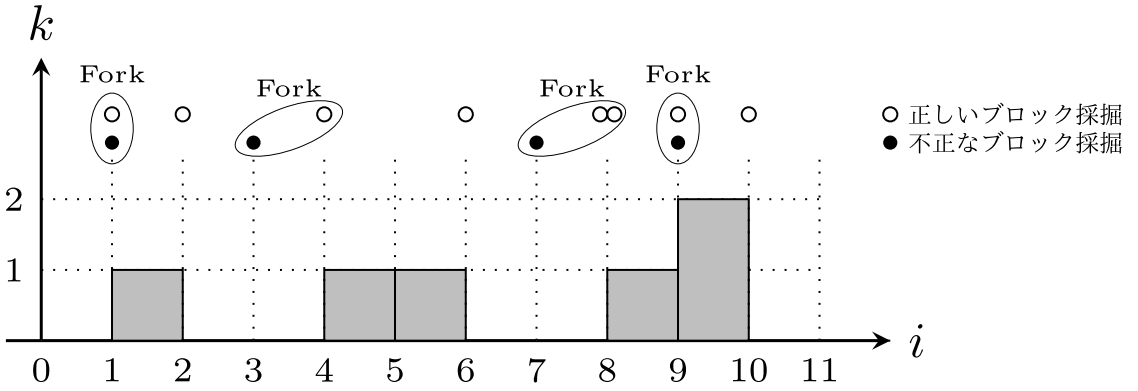


図 7: Fork の発生と解消に関するシミュレーション。横軸はラウンドを表し、時間の経過とともに右に向かって新しいラウンドに進む。縦軸は、各ラウンドでの Fork の長さを表す。各ラウンドで、不正な採掘者は意図的に長い Fork を作ろうと試みるが、正しい採掘者のブロック生成速度が上回るために典型実行では Fork は短く保たれる。

が生成されて解消している。ラウンド 3 と 7 では、不正なブロックが採掘されるが、不正な採掘者により隠匿されている。ラウンド 4 と 8 で正しいブロックが採掘されると、これらの不正なブロックが公開されて Fork が生じる。また、ラウンド 8 の Fork は、3 又となっているが、ラウンド 9 で 2 又となっている。

ラウンドあたりの正しい採掘者によるマイニング成功確率を f $(= E[X_i = 1])$ とすると、Fork 長 k が $2f|S|$ を超える確率は、典型実行の要請から、対応する区間長 $|S|$ の指数関数で上からおさえられる。

$$\Pr\left[k > 2f|S|\right] < e^{-\Omega(\epsilon^2|S|)} \tag{11}$$

ここで $\Omega(x)$ はランダウの Ω-記号で、$f(x) = \Omega\left(g(x)\right)$ のとき、十分大きな任意の x に対して、ある定数 $k > 0$ が存在して、$f(x) > k \cdot g(x)$ が成り立つことを表す。上記の Fork 長の限界式 (11) を満たす k を選ぶ限り、最新 k ブロックを除いた残りのブロックチェーン（共通プレフィックス）は正しい採掘者の間で完全に一致するとみなせる（一致しない確率が無視できるほど小さい）ことが理論的に保証される。しかし、共通プレフィックス定理により、Fork 長の上限はビットコインの場合、不正な採掘者が少ない (10% 以下) 環境下では実用的に 10 ブロック程度に抑えられると考えて良い (補論 D)。

(4) ブロックチェーンの安全性

以上をまとめると、ブロックチェーンの安全性は以下のようにまとめられる。

- **取引の偽造耐性（暗号資産の不正利用耐性）**

  取引を偽造するためには、(1) デジタル署名を偽造する、(2) 秘密鍵を盗取する、のいずれかが必要である。ビットコインやイーサリアムといった一般的な暗号資産では、暗号学的に安全なデジタル署名 (ビットコインでは ECDSA 等) が用いられることから、(1) の攻撃は現在のところ困難と考えられる。他方、(2) についてはコールドウォレット等の鍵管理技術で利用者自身で自衛策を講じる必要がある。

- **ブロックチェーンの改ざん耐性**

  ブロックチェーンに含まれるブロックや取引セットを改ざんするには、少なくとも実装されている SHA-256 や SHA-3 等の暗号学的ハッシュ関数の一方向性や衝突困難性を破る必要があり、現在のところ困難と考えられる。

- **ブロックチェーン合意形成**

  Fork 状態が続くと、ブロックチェーンの合意形成を阻害され、不正取引を受け入れる余地が生じる。取引の完了性（ファイナリティ）が特に求められる取引については、当該取引が登録されたブロックが数ブロック以上の深さに到達するまで待つことで、合意が覆される確率を十分小さくできる。

4 匿名性と取引内容の秘匿

ビットコインやイーサリアム等のブロックチェーンでは、取引に用いられる口座には、一時的なアドレスが用いられるため、一定の匿名性は確保される。他方、取引内容については、平文でブロックチェーンに記録されるため、その内容は秘匿されない。こうした状況のもとでは、取引情報に含まれる個人情報や、仮名化された取引情報の統計分析を利用した攻撃により、個人を特定できる可能性がある。

こうしたことを可能にする攻撃手法として以下のようなものがある。まず、取引グラフ分析は、時間の経過とともに変化するいくつかの全体的な取引の特徴（例えば、毎日の取引高、為替レート、または取引パターン）を発見することに焦点を当てる。

例えば、ビットコインではビットコインネットワーク内の 4 つの特徴的なトランザクショングラフ[25]のパターンを検出できることが示されている (Ron and Shamir [2013])。こうした取引グラフ分析と匿名データの追跡手法と併用することで、個人の金融取引履歴を発見できる可能性がある。次に、P2P ネットワーク分析では、ブロックチェーンの P2P ネットワークに属するノードに次々と接続し、他の接続先ノードの IP アドレスのリストを要求することで、P2P ネットワークのサイズ、構造、分布に関する情報を収集することができる。フェルドら (Feld, Schönfeld, and Werner [2014]) は、クラスターのサイズとその分布を分析し、全ノードの 30 %以上が 10 のクラスターに属するなどの特徴を明らかにしている。

したがって、取引の統計情報ももらさないことが、取引の匿名性を確保するうえで重要である。これを実現する手法として、(1) Mixing による手法（CoinSwap、CoinJoin、StealthAddress 等）、(2) 準同型暗号とゼロ知識証明（zk-SNARKs[26]等）による手法（Zerocash 等）が知られている。ただし、それぞれの手法で達成される匿名性の定義はそれぞれ異なっているほか、匿名性に対する理論的な保証が必ずしも与えられていない場合がある点には留意が必要である。以下では、これらの手法がそのような匿名性をどのような技術で達成しているか解説する。

[25] ビットコインではトランザクション (取引) で受領したコインを集約して総額（残高）で管理するのではなく、個々のトランザクションのアウトプットアドレス (UTXO) の集合で管理している。資金移動は、自分が秘密鍵を管理する UTXO を複数束ねて別のアドレス宛の支払いに充てることで実現される。従って、取引を介してアドレス (UTXO) 間に依存関係が生じるため、有向グラフで表現することができる。この有向グラフをトランザクショングラフと呼ぶ。

[26] zk-SNARKs（Zero-Knowledge Succinct Non-Interactive Argument of Knowledge）

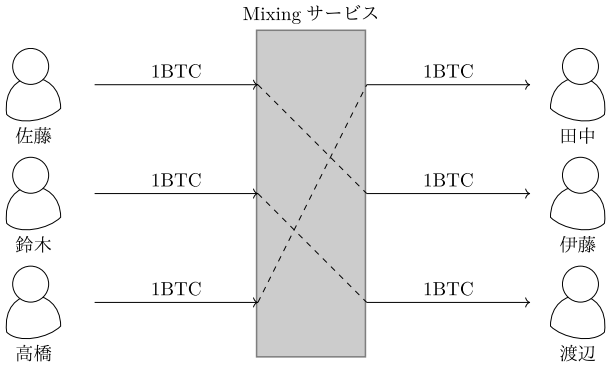


図 8: 集中型 Mixing の概要。図では佐藤 → 伊藤, 鈴木 → 渡辺, 高橋 → 田中などの取引関係が、Mixing サービスを中継することでブロックチェーン上での追跡が困難になる。

(1) Mixing による匿名化

Mixing とは、複数の取引をまとめることで、個々の取引における資金の受け手と送り手を特定できないようにする技術である。複数の取引をまとめる際には、取引を仲介する仲介事業者（Mixing 事業者）を介する集中型 Mixing と、こうした Mixing 事業者を要しない分散型 Mixing がある。

イ　集中型 Mixing

集中型 Mixing は、図 8 に示すように、Mixing 事業者の Web サイト[27]等で複数の取引を集約し、取引を 1 つに纏めることで取引当事者間の取引関係を秘匿する技術である。素朴な集中型 Mixing には、次の 2 つのリスクがある。

a) Mixing 事業者による資産の盗取 (Meiklejohn et al. [2013])

b) Mixing 事業者による取引に関係した口座情報の漏洩

CoinSwap(Maxwell [2013b]) は、ビットコインの主要な開発者の 1 人であるグレゴリー・マックスウェル氏がビットコインフォーラムで提案した手法で、Mixing 事業者を仲介させることで送金者と受領者の関係を分断している。これにより取引の匿名性を確保している。その際、期限付きエスクローとハッシュロック[28]を用いて、送金者-Mixing 事業者間と Mixing 事業者-受領者の間に発生する 2 つの取引を同時に実行する巧妙な仕組みを構築し、a) Mixing 事業者が資産を盗取するリスクを排除している。ただし、依然として b) Mixing 事業者による取引に関係した口座の漏洩リスクは残る。

TumbleBit (Heilman et al. [2017]) は、a) と b) のリスクを解消し、Mixing 事業者による盗取回避とプライバシー漏洩対策を両立した最初の方式である。プロトコル中で用いるパズルは図 9 に示すランダム自己帰着性を利用している。ブラインド署名 (Chaum [1983]) をはじめ、ランダム自己帰着性は匿名技術に古くから用いられている。プロトコルは 3 つのフェー

[27] `http://bitcoinfog.com`、`https://bitblender.io`、`http://app.bitlaundry.com` 等がある。

[28] エスクローアドレスに資金が確保されており、期限が到来すれば事前に定められたアドレスに返金し、期限までに指定されたハッシュ値の入力値 (秘密) が提示されれば即時に資金移動が許可される仕組み。秘密を Mixing 事業者に開示せず、別チャネルで送金先に直接開示することで盗取リスクを軽減した資金移動が可能になる。

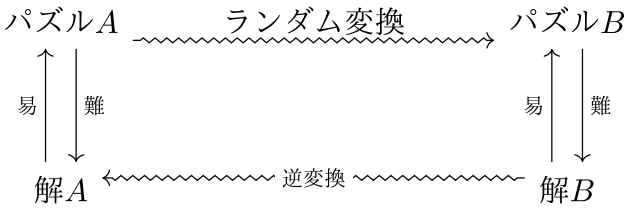
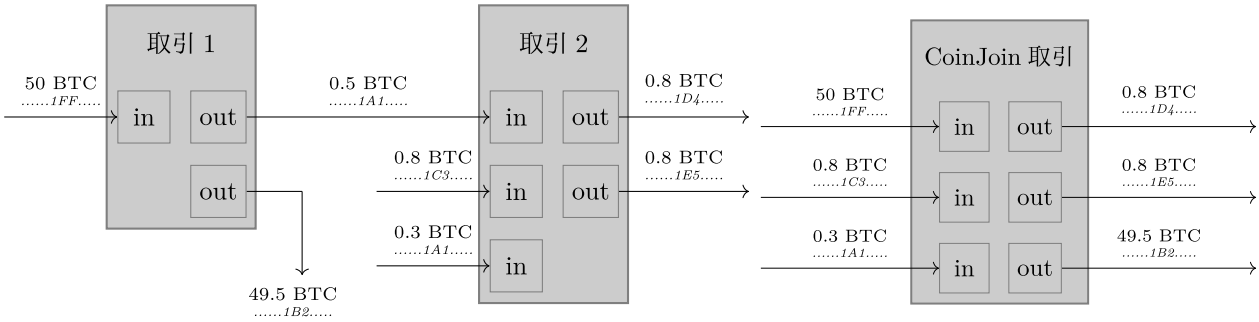


図 9: TumbleBit で用いられるパズルの数理構造 (ランダム自己帰着性)。与えられたパズル A からランダム変換によりパズル B を生成できる。さらに、解 B が与えられれば、逆変換を用いて解 A を効率良く求められる。



(a) CoinJoin 前の複数取引

(b) CoinJoin 後の取引

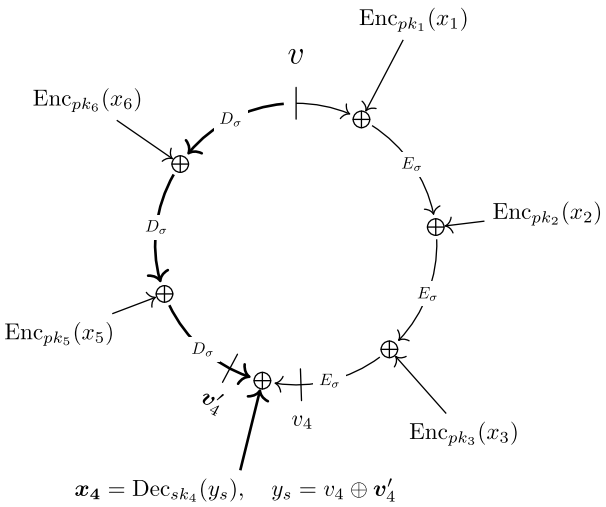
図 10: CoinJoin による取引の匿名化の概要

ズで構成されている。まず、Tumbler（TumbleBit プロトコルにおける Mixing 事業者）から受領者への送金取引と期限付エスクロー取引を設定する。受領者が資金を受領するためには、Tumbler から入手したパズル A を期限内に解かなければならない。受領者は送金者にパズル A をランダムに変換したパズル B を送付する。次に、送金者は Tumbler からパズル B の解 B を購入し、解 B を受領者に送信する。最後に、受領者は解 B から解 A を求め、解 A で冒頭の Tumbler から受領者への送金取引をブロックチェーンに登録して資金を受領する。TumbleBit では、Mixing 事業者である Tumbler でさえも送金者と受領者を特定できず、取引に関係した口座の漏洩対策が実現されている。TumbleBit では Tumbler の取引不正を検知する仕組みを暗号プロトコルで実現しており、偽ったパズルを出題したり、誤った解を送金者に販売する等の不正行為を未然に検知できる。

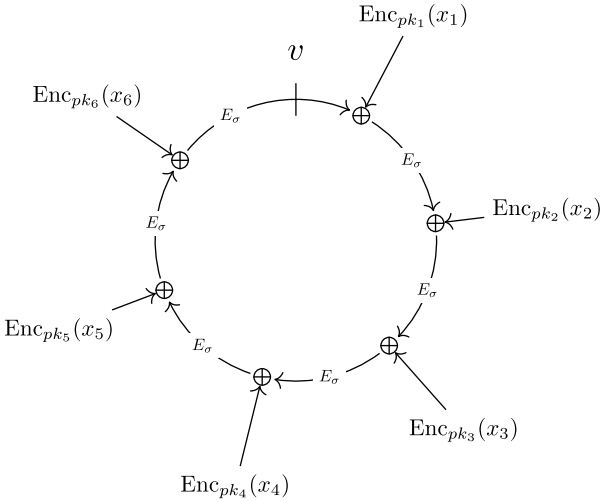
ロ　分散型 Mixing

分散型 Mixing は Mixing 事業者を介さず、利用者間のプロトコルで Mixing を実現する技術であり、CoinJoin(Maxwell [2013a]) 等が知られている。

CoinJoin も前述のマックスウェル氏がビットコインフォーラムで提案した手法であり、図 10 に示すように、「複数の取引を同一トランザクションにまとめて実行する」ことにより、入出力の対応関係を複雑にし、匿名性を高めている。方式のシンプルさから CoinJoin には多くの亜種が存在する。図 10(b) の取引では、50BTC の入力 (...1FF...) と 49.5BTC の出力 (...1B2...) の対応は明らかだが、その他の送金者と受領者の対応（入出力の対応）は不透明である。取引額を一定額に統一できれば匿名性はさらに高まる。入出力の対応を CoinJoin の参加当事者からも秘匿するために、各当事者が匿名性を確保したルーティングプロトコルである Mix network(Chaum [1981]) で出力を隠蔽する手法、各当事者が取引にブラインド署名



(a) リング署名の生成。秘密鍵 sk_4 を知っている署名者は、$x_4 = \mathrm{Dec}_{sk_4}(v_4 \oplus \boldsymbol{v}'_4)$ で逆算することで、リング署名を生成できる。



(b) リング署名の検証。v から始めて、順に暗号文 $\mathrm{Enc}_{pk_i}(x_i)$ を加算して暗号化 E_σ を繰り返すことにより v に戻ることを確認する。

図 11: リング署名

を使用するなどの方法が提案されている。

Mixing の匿名性のまとめると、以下の通りである。

- Mixing 技術は、既存の暗号資産の仕様を変更せずに匿名性を高められる利点がある。
- Mixing 技術の匿名化は、原理的に Mixing に関わる当事者の範囲に留まる。ただし、ブロックチェーンに登録された情報では Mixing 取引と一般取引を区別できず、Mixing による匿名化の範囲を特定することは一般に困難である。取引時刻や取引額の分布から、送金者と受領者の対応を絞り込めることもある。

(2) リング署名

リング署名は、複数のユーザがグループとして発行するデジタル署名技術であり、真の署名者をグループの中に隠蔽できるという意味で匿名性を確保できる (Rivest, Shamir, and Tauman [2001])。この方式の名前は、署名が持つリング状の構造に由来している（図 11 を参照）。リング署名の検証は、グループに属するユーザが公開した公開鍵の集合を使って行う。リング署名の発行は、グループに属するユーザが保有する秘密鍵のうち任意の 1 つがあれば可能である。このことから、グループに属するユーザのうちの誰かがリング署名を発行したことを証明できるが、誰が署名を発行したか（どの秘密鍵が使われたか）を特定することはできないため、匿名性が達成される。

r 人のグループによるリング署名を発行するには、まず、署名を発行するユーザ（以下、署名者と呼称）が、n 人のユーザ全体の中から $r-1$ 人を任意に選定し、自身を含めた r 人でグループ（匿名セット）を構成する。これは、ユーザ全体の公開鍵の集合鍵 $(pk_0, \cdots, pk_n)$

の中から $r-1$ 個を任意に選択することに相当する。次に、署名者は、先ほど選定した $r-1$ 個の公開鍵と自身の秘密鍵（1 個）を使って、サイズ r のリング署名を生成する。

したがって、前節の Mixing では取引当事者に匿名化の範囲が限定されていたが、リング署名では任意に匿名化の範囲を指定でき、より強力な匿名性を達成できる。

イ　署名／検証アルゴリズム

この節では、リング署名の署名アルゴリズムと検証アルゴリズムの概要を説明する。以下において、署名者は、任意に選んだ s 番目の位置に自身の署名を埋め込むものとする。グループの公開鍵の集合を $(pk_1, \ldots, pk_r)$ とする。この s 番目の pk_s は、署名者のものである。メッセージ m は、暗号資産の取引内容（受領者、金額など）の情報である。H は暗号学的ハッシュ関数を表す。E_σ および D_σ は、σ を鍵として利用した共通鍵暗号の暗号化関数および復号関数である。Enc_{pk_i} は、pk_i を鍵として利用する公開鍵暗号の暗号化関数である。また、Dec_{sk_s} は、sk_s を鍵として利用する公開鍵暗号の復号関数である。$\oplus$ はビット単位の排他的論理和演算である。$[1, r]$ は、1 から r までの自然数の集合を表す。こうした設定のもとで、署名アルゴリズムと検証アルゴリズムは、それぞれ次のように構成される。

署名アルゴリズム　リング署名は以下のように構成できる。

a) リング署名対象のメッセージ m のハッシュ値 $\sigma = \mathrm{H}(m)$ を求める。この σ は共通鍵暗号の鍵となる。

b) 乱数 $(v, x_1, \ldots, x_r)$ を選択する。

c) s を除く各 $i \in [1, r]$ について $y_i = \mathrm{Enc}_{pk_i}(x_i)$ を計算する。

d) $v = \mathrm{E}_\sigma(y_r \oplus \mathrm{E}_\sigma(y_{r-1} \oplus \cdots \mathrm{E}_\sigma(y_s \oplus \cdots \mathrm{E}_\sigma(y_2 \oplus \mathrm{E}_\sigma(y_1 \oplus v)) \cdots) \cdots))$
を満たす y_s を求める[29]。

e) $x_s = \mathrm{Dec}_{sk_s}(y_s)$ を計算し、m に対するリング署名 $(v, x_1, \ldots, x_r)$ を得る。

検証アルゴリズム　メッセージ m とリング署名 $(v, x_1, \ldots, x_r)$、公開鍵の集合 $(pk_0, \cdots, pk_r)$ が与えられたとき、リング署名は以下の手順で検証される。

a) 共通鍵暗号の鍵を $\sigma = \mathrm{H}(m)$ とする。

b) 各 $i \in [1, r]$ について $y_i = \mathrm{Enc}_{pk_i}(x_i)$ を計算する。

c) $w = \mathrm{E}_\sigma(y_r \oplus \cdots \mathrm{E}_\sigma(y_2 \oplus \mathrm{E}_\sigma(y_1 \oplus v)) \cdots)$ を計算する。

d) $v = w$ ならば 1(検証成功)、$v \neq w$ ならば 0(検証失敗) を出力する。

[29] y_s は次のようにして求められる。最初に、v_s を求める。そのために、まず $v_2 = \mathrm{E}_\sigma(y_1 \oplus v)$ とする。続いて、すべての $i \in [2, s-1]$ について番号の小さいほうから順に、$v_{i+1} = \mathrm{E}_\sigma(y_i \oplus v_i)$ を求めていく。ここで、$y_i = \mathrm{Enc}_{pk_i}(x_i)$ である。次に、v'_s を求める。まず、$v'_r = \mathrm{D}_\sigma(v)$ を計算しておく。続いて、すべての $i \in [s+1, r]$ について、番号の大きいほうから順に $v'_{i-1} = \mathrm{D}_\sigma(y_i \oplus v'_i)$ を求めていく。最後に、$y_s = v_s \oplus v'_s$ とする。

ロ　リング署名による取引の匿名化

リング署名の匿名性と追跡可能性を応用したブロックチェーンプロトコルが開発されている (例えば、Saberhagen [2013], Noether and Mackenzie [2016], Sun et al. [2017])。CryptoNote (Saberhagen [2013]) は、追跡可能リング署名[30](Fujisaki and Suzuki [2007], Fujisaki [2011]) を拡張し、秘密鍵が取引に署名できる回数を 1 回に制限した one-time 秘密鍵のハッシュ値をタグとして用い、二重支払いの検出に用いている。one-time 秘密鍵を複数回使用しない限り、署名者は匿名セットに含まれる他のユーザと区別できない。

具体的には、真の送金者の公開鍵を P とするとき、図 12(a) に示すように、P に対応する鍵イメージ $I = x\,\mathrm{H}_p(P)$ をトランザクションに含め、このトランザクション全体をリング署名で送金者 P に署名させる。ここで x は P の秘密鍵であり、H_p は楕円曲線上の点を別の点に写像する特殊な一方向性ハッシュ関数である。I と $\mathrm{H}_p(P)$ から x を求めることは楕円曲線上の離散対数問題[31]に相当するため困難である。G を楕円曲線上のベースポイント[32]とするとき、秘密鍵 x を決めれば、$P = xG$, $I = x\,\mathrm{H}_p(P)$ によって P と I は一意に定まるが、I を公開しても、秘密鍵 x を知らない限り P と I の対応はとれない (計算量的に困難)。

CryptoNote では送金者に鍵イメージ I をトランザクションに含めてリング署名をさせるため、同じ P を別のトランザクションで別の匿名セットに埋め込もうとしても、そこでも同じ I をトランザクションに含めざるを得ない。こうしたもとで、匿名性を悪用して同一アドレス P で二重支払いを実行すると、必ず同じ I を双方のトランザクションに含めざるを得ないため、同じ I に対応する 2 つのリング署名が存在することになる。詳細は割愛するが、1 つの I に対応する 2 つの (リング) 署名があれば、I から P を求めることができ、二重支払いを実施したアドレス P を特定できるようになっている。

さらに、受領者の匿名性は次のようにして実現されている。図 12(b) に示すように、受領者が公開している 2 つの公開鍵 $A(= aG)$, $B(= bG)$ を利用して、一時アドレスに送金して、匿名で受領者に送金するテクニックが使われている。ここで a, b は受領者の秘密鍵である。これは、送金者が作成した乱数 r を用いて、A と B を合成した P と、乱数 r へのコミットメント $R(= rG)$ を生成し、P を送金先のアドレス、R は付加情報としてトランザクションに含め、トランザクション全体に前述のリング署名を施す。P と R は乱数にしか見えず、受領者の公開鍵 A, B との関係は暗号学的に秘匿されている。受領者は、ブロックチェーンに公開されたトランザクションのリストから、自分宛の支払いを $P = \mathrm{H}(aR)G + bG$ が成立するか否かで判定する。成立すれば、自分宛の支払であり、当該資金の使用時に必要な秘密鍵 x は、$x = \mathrm{H}(aR) + b$ で容易に求められる。

これらの技術により、CryptoNote では送金者、受領者双方の匿名性を確保すると共に、匿名性を悪用した二重支払いを防止している。さらに、CryptoNote の改良版にあたる暗号通貨のプロトコルとして、ネーターら (Noether and Mackenzie [2016]) は、Ring Confidential Transaction（RingCT）を提案している。RingCT は、CryptoNote と比較して、取引の金額も隠匿できる点で優れているため、匿名性に加えて、取引内容の秘匿性も同時に達成できる。RingCT は、2014 年にリリースされた Monero[33]に実装されている。

リング署名を用いた匿名化技術の特徴をまとめると、以下の通りである。

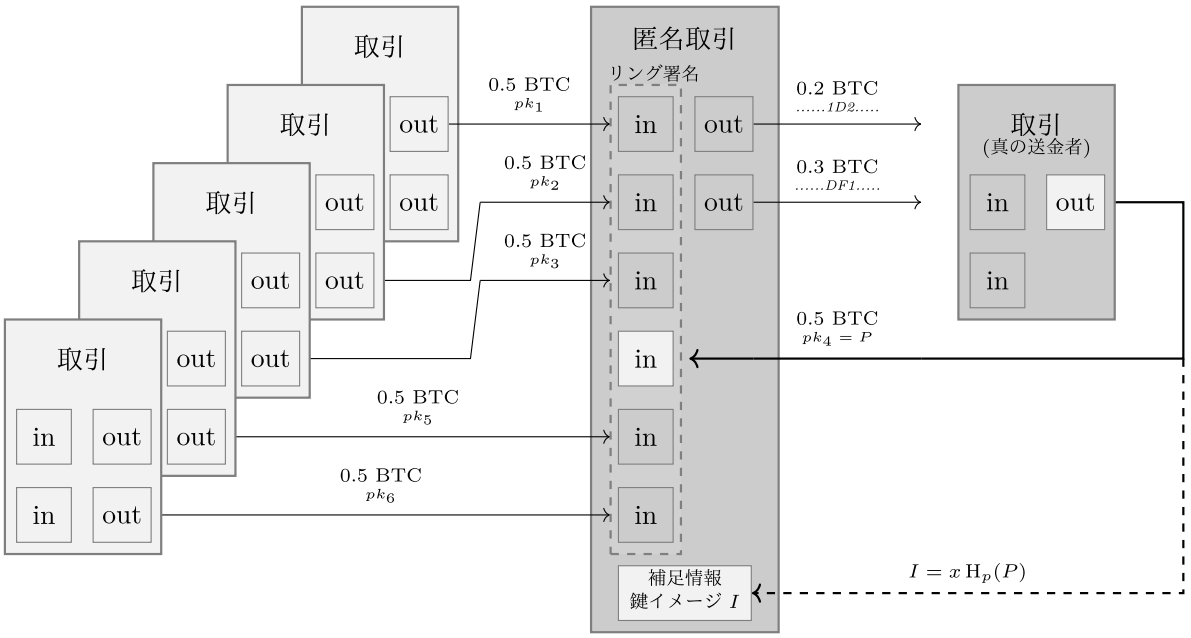
---

[30]追跡可能リング署名は、通常はリング署名と同じく匿名性を有するが、2 つの異なるメッセージに署名した場合には匿名性がなくなり、署名者が特定されるという特徴を持つ。

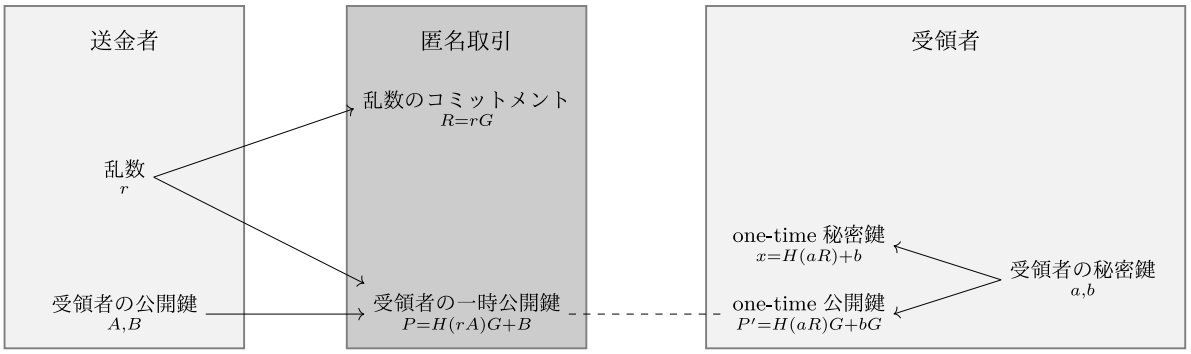
[31]ビットコインをはじめとする多くの暗号資産は楕円曲線暗号に基づいており、CryptoNote も楕円曲線上の演算で構成されている。楕円曲線暗号および離散対数問題については、補論 E に簡単な解説を加えた。

[32]ベースポイント G のスカラー倍で表される楕円曲線上の点が群を構成する。

[33]`https://getmonero.org/`



(a) One-time リング署名による送金者の匿名性。送金者の公開鍵 P を匿名セット $\{pk_1, \ldots, pk_r\}$ に紛れさせ、P と一意に対応する鍵イメージ I を含めて、取引データ全体をリング署名する。この結果、匿名セットに含まれるどのアドレスから実際に支払われたかは分からない。鍵イメージ I は、匿名性を悪用した二重支払いを防止するために用いられる。



(b) 一時アドレスによる受領者の匿名性。受領者は自身の公開鍵 (アドレス)A,B を 2 つ公開する。送金者は、受領者を決定すると、A,B からランダムに一時アドレス P とコミットメント R を生成し、宛先アドレスとして取引に埋め込む。秘密鍵 a, b を知る受領者だけが、当該取引が自身への支払いであることを検知でき、P に対応する秘密鍵 x を求めて、資金を受領できる。

図 12: CryptoNote における匿名性の仕組み

- Mixing では実際に発生した取引を混ぜる必要があったが、リング署名では任意に選択したユーザー集合（匿名セット）の中に署名者を隠蔽できる。
- リング署名では、匿名セットのサイズが大きいほど匿名性は高まるが、同時に取引データのサイズも増大する。取引あたりのデータ量が数キロバイトに及ぶこともあり、データ容量や署名の検証等に要する計算コストの面でブロックチェーンの負荷を増大する。

(3) ゼロ知識証明

ゼロ知識証明 (Zero-Knowledge Proof) は、ある人（証明者）が、他の人（検証者）に対して、ある事実（数学的な命題）が正しいことを、命題が真であること以外の一切の情報を与えずに証明する技術である。特に、証明者から検証者への一方的なメッセージ送信のみでゼロ知識証明を行う技術を非対話ゼロ知識証明（NIZK: Non-Interactive Zero-Knowledge Proof) と呼ぶ（詳しくは補論 F を参照)。特に、決済取引においては、送金者と受領者が双方向にコミュニケーションを取ることが通常では想定されないため、1 度の一方向的な情報の送信で取引を検証できる非対話（non-interactive）のプロトコルであることが重要である。

NIZK 証明は、暗号資産を使った決済取引では、匿名性や取引内容の秘匿性を維持したまま、資産（コイン）が偽造されていないことや、マネーロンダリングなど特定の不正取引への不関与などの条件が満たされていることの証明に用いられる。典型的な利用の方法には、例えば、取引金額が暗号化された状態で、取引の前後で利用者全体の総資産が増減していないことを、取引金額を明かさずにゼロ知識で証明することが挙げられる。今、$\boldsymbol{x}^t = (x_1^t, \ldots, x_n^t)$ を時刻 t における全利用者の資産残高とし、複数の決済取引が実行された後、時刻 t' に全利用者の資産残高が $\boldsymbol{x}^{t'} = (x_1^{t'}, \ldots, x_n^{t'})$ に変化したとする。この間、総資産に変化はないとすると、以下の条件式が成り立つ必要がある。

$$\sum_{i=1}^{n} \left(x_i^t - x_i^{t'} \right) = 0$$

暗号資産の取引に関する条件式ではより高次の多項式が必要になることもあるが、多くの場合、必要な条件式を多項式 $= 0$ の形で表すことができる。

以下では, 暗号化された資産残高情報や取引情報などが, 多項式 $= 0$ の形の条件式を満たすことを NIZK で示すための方法として注目されている zk-SNARKs (zero-knowledge Succinct Non-interactive ARgument of Knowledge) の概要を解説し、zk-SNARKs を用いて匿名性と取引の秘匿を同時に実現する暗号プロトコル Zerocoin およびその実装である Zerocash について解説する。

イ zk-SNARKs

概要 zk-SNARKs(Gennaro et al. [2013]) は、図 13 に示すように、暗号文による操作だけで、背後にある平文が任意の条件式（多項式 $= 0$）を満たすことをゼロ知識証明によって示すことを可能にする技術の一つである。暗号文に対する操作だけで、秘匿されている平文に

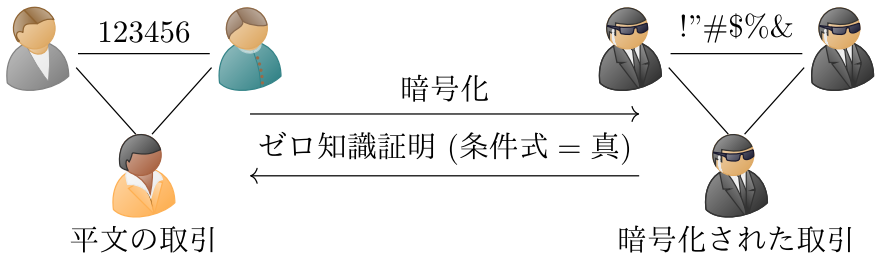


図 13: ゼロ知識証明による匿名化と取引内容の秘匿の仕組み。暗号化された取引がルールに従って資金移動がなされていること（条件式=真）を取引内容を秘匿化したまま証明する。

対して演算を実行できる機能を持つ暗号を準同型暗号と呼び、ゼロ知識証明でも準同型暗号の性質を利用する。

最初に、zk-SNARKs に概要を紹介しておく。zk-SNARKs は以下の 3 つのステップで構成される。

（ステップ 1 ：証明したい条件式の変換） 今、入力 x がある条件を満たすことを、$P(x) = 0$ という多項式で表すことにする。$P(x)$ は多項式なので、加算演算と積算演算をゲートとする計算グラフで表すことができる。各ゲートの入力と出力はそれぞれの演算で定められた関係を満たす必要があり、後に述べる QAP(Quadratic Arithmetic Program) の理論を用いて、計算グラフの各ゲート i に対応する多項式 $v_i(z), w_i(z), y_i(z)$ とワイヤ値 $C_1, \ldots, C_m$ が満たすべき関係を表す等式を、$t(z)h(z) = v(z)w(z) - y(z)$ に変換する。ここで、$v(z) = \sum_{i=1}^{n} C_i v_i(z)$、$w(z) = \sum_{i=1}^{n} C_i v_i(z)$、$y(z) = \sum_{i=1}^{n} C_i y_i(z)$ はワイヤ値を係数とする多項式の線形結合である。このとき、最初の条件式 $P(x) = 0$ の入力 x は $n(\leq m)$ 本のワイヤ値 $(C_1, \ldots, C_n)$ で表されている。

（ステップ 2 ：暗号文での条件式の生成） 次に、多項式に関する等式の成立を暗号文を使って検証できるゼロ知識証明を構成する。ここで証明したい事実は、入力値 x に対応するワイヤの $C_1, \ldots, C_n$ が与えられたときに、残りのワイヤ $C_{n+1}, \ldots, C_m$ が多項式間の等号関係

$$t(z)h(z) = v(z)w(z) - y(z) \tag{12}$$

を満たすことを示すことである。計算グラフの入力値 $C_1, \ldots, C_n$ が決まると、残りのワイヤ値 $C_{n+1}, \ldots, C_m$ は自動的に決まる[34]。

まず、ある秘密の値 s がランダムに選ばれ、その暗号文 $g, g^s, g^{s^2}, \ldots, g^{s^m}$ を公開されているものとする。s の値は暗号文を解読しない限り、誰も知らないという設定を想定している。この公開された s の暗号文とワイヤ値 $C_1, \ldots, C_m$ を使って、平文の条件式 (12) から以下の暗号文の条件式 (13) を満たす暗号文 $T(s), H(s), V(s), W(s), Y(s)$ を計算する。これらの値が NIZK 証明 π である。

$$e(T(s), H(s)) = \frac{e(V(s), W(s))}{e(g, Y(s))} \tag{13}$$

[34]条件を満たさない入力値に対して、残りのワイヤ値を調整して多項式間の等号関係を満たすことは、$P(x) = 0$ との同値関係から不可能である。

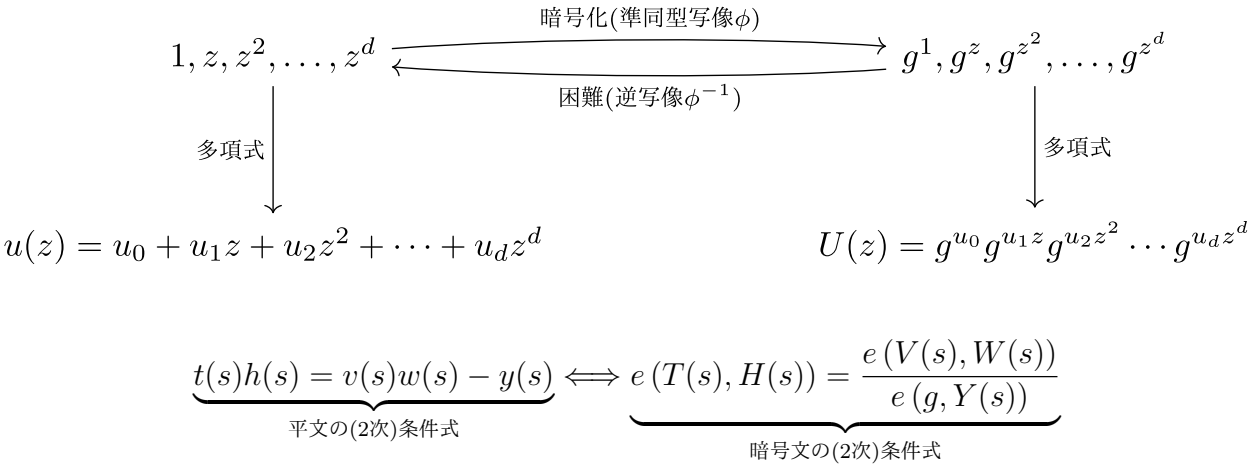


図 14: zk-SNARKs における（平文の）条件式と暗号文での検証に用いられる条件式の関係

(ステップ 3:ランダム化によるゼロ知識性の達成)　さらに、ゼロ知識性を達成するために、3 つの乱数 α, β, γ を導入して、3 つの値 $V(s), W(s), Y(s)$ をランダム化し、対応する $H^*(s)$ を求めることで、NIZK 証明 $\pi^* = (T(s), H^*(s), V^*(s), W^*(s), Y^*(s))$ の基本要素が作られる。

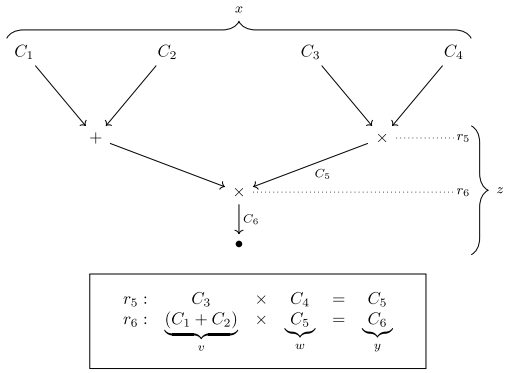
$$
\begin{aligned}
V^*(s) &\leftarrow V(s)T(s)^{\alpha} \\
W^*(s) &\leftarrow W(s)T(s)^{\beta} \\
Y^*(s) &\leftarrow Y(s)T(s)^{\gamma}
\end{aligned}
$$

この NIZK 証明 π^* は π と同様に上の検証式を満たすため、π^* を受信した人 (検証者) は、入力 x が $P(x) = 0$ の条件式を満たすことを納得する。しかし、ワイヤの値 $C_{n+1}, \ldots, C_m$ を知ることはできない。ここまでに用いた平文の積は、$t(s)h(s)$ と $v(s)w(s)$ の 2 つの 2 次式のみであり、残りの計算は平文上の加減算である。従って、任意回の加減演算と 1 回の乗算が行える準同型暗号を利用すれば、NIZK を構成できる。

zk-SNARKs の原理　非対話ゼロ知識証明と一方向性を持つ準同型写像があれば、暗号資産の決済取引で、匿名性と取引内容の秘匿性を達成するプロトコルを実現される。代表的な準同型暗号には、楕円曲線暗号ベースの双線形写像に基づく準同型暗号と、近年盛んに研究されている格子暗号ベースの完全準同型暗号 (Gentry, Halevi, and Smart [2012]) がある。先に発見された楕円曲線暗号ベースの双線形写像に基づく準同型暗号は、既に多くの研究がなされており暗号資産への応用も盛んだが、これまでに知られている範囲では平文の 2 次式で表せる条件式[35]までしか扱えないという制約がある (Boneh, Goh, and Nissim [2005])。そこで、zk-SNARKs は QAP(Quadratic Arithmetic Program) を利用して、平文の高々 2 次式で任意の条件式を扱えるようにすることで、暗号文上で条件式を判定することを可能にした理論である。Quadratic Span Program(QSP, もしくは QAP[36]) は、$\{0, 1\}$ を出力する任意の条

[35]乗法群に基づく暗号では、平文の和が暗号文の積に対応し、平文の積が暗号文の双線形写像に対応する。楕円曲線の双線形写像に基づく準同型暗号では、3 つ以上の平文の積を扱う方式は知られていない。

[36]QSP は 2 値回路 (Boolean 回路) の充足性を多項式の剰余が 0 になるという関係に帰着させる理論であり、QAP は整数値の演算回路の結果がある値になるという条件を、同様に多項式の剰余が 0 になるという関係に



(a) 条件式の計算グラフ。各演算子の入力 v, w と出力 y の間の関係式を列挙する。全ての関係式の成立を示すことで、正しく計算したことを証明できる。

$z=$	r_5	r_6
$v_1(z)$	0	1
$v_2(z)$	0	1
$v_3(z)$	1	0
$v_4(z)$	0	0
$v_5(z)$	0	0
$v_6(z)$	0	0

左入力

$z=$	r_5	r_6
$w_1(z)$	0	0
$w_2(z)$	0	0
$w_3(z)$	0	0
$w_4(z)$	1	0
$w_5(z)$	0	1
$w_6(z)$	0	0

右入力

$z=$	r_5	r_6
$y_1(z)$	0	0
$y_2(z)$	0	0
$y_3(z)$	0	0
$y_4(z)$	0	0
$y_5(z)$	1	0
$y_6(z)$	0	1

出力

(b) 各 $v_i(z), w_i(z), y_i(z)$ が $z = r_5, r_6$ で表の値をとることで、$v(z) = \sum_{i=1}^{6} C_i v_i(z)$, $w(z) = \sum_{i=1}^{6} C_i w_i(z)$, $y(z) = \sum_{i=1}^{6} C_i y_i(z)$ とし、$F(z) = v(z)w(z) - y(z)$ とおくとき、$F(r_5) = F(r_6) = 0$ を満たす。多項式 $v_1(z) = \frac{1}{r_6 - r_5}(z - r_5)$, $w_1(z) = (z - r_5)(z - r_6)$, $y_1(z) = (z - r_5)(z - r_6)$ などは上の表の値をとる。

図 15: QAP:Quadratic Arithmetic Program の仕組み

件式 F(演算回路)[37]を、回路の機能から定まる 3 つの多項式 $v(z), w(z), y(z)$ に分解する。このとき、以下が成り立つ。

$$F(z) = 0 \Longleftrightarrow v(z)w(z) - y(z) = 0$$

図 14 に示すように、zk-SNARKs は, QAP を利用して高次多項式 $= 0$ の形の条件式を、2 次多項式 $= 0$ の形の条件式に変換してゼロ知識証明を実現した。

Quadratic Arithmetic Program　zk-SNARKs では QAP を用いて、条件式に用いられる高次多項式を、双線形写像[38]の準同型性で扱うことができる 2 次多項式で表現する。図 15(a) に示すように、条件式を $+$ と $\times$ の計算グラフで表したとき、各演算子の入出力は演算子固有の関係を満たす必要がある。全ての演算子について、これらの関係を合わせたものは、図 15(a) の下部に示すように、入力ワイヤ C_1, C_2, C_3, C_4 と中間ワイヤ C_5, C_6 の関係として表すことができる。

図 15(b) に示すように、各中間ワイヤに対応する適当な値 $z = r_5, r_6$ を定め[39]、計算グラフにおける演算子の左側（一方）の入力を $v_i(z)$、右側（他方）の入力を $w_i(z)$、出力を $y_i(z)$ とする。ここで i はワイヤ C_i の関係を表し、$v_i(r_5) = \alpha$ は r_5 の演算子に αC_i が入力されていることを表す。例えば、$v_3(r_5) = 1$ で r_5 の関係式に対応する演算子の左側の入力にワイヤの値 C_3 が割り当てられていることを表す。このとき、$z = r_5, r_6$ で図 15(b) の値をとるように

帰着させる理論である。両者の理論的な差異はあまりないため、本論文では理解が比較的容易な QAP をベースに解説する。

[37]ここでは z としてワイヤの値を想定している。図 15(a) のワイヤを入力ワイヤ $u = (C_1, C_2, C_3, C_4)$ と、中間ワイヤ $w = (C_5, C_6)$ に分割すると、$F(z) = 0 \Leftrightarrow F(u, w) = 0$ を表す。これは、さらに後述の関係 R について $(u, w) \in R$ と対応している。

[38]ここでは乗法群上の双線形写像 $\phi : G_1 \times G_2 \to G_T$ を考える。すなわち、$\phi(xy, z) = \phi(x, z)\phi(y, z)$ かつ $\phi(x, yz) = \phi(x, y)\phi(x, z)$ を満たし、非退化 $(x, y \neq 1_{G_1}, 1_{G_2}$ に対して $\phi(x, y) \neq 1_{G_T})$ の写像として定義する。したがって、$\phi(x^a, y^b) = \phi(x, y)^{ab}$ が導かれる。

[39]r_5 と r_6 は相異なる値であれば、任意の値で良い。

多項式 $v_i(z), w_i(z), y_i(z)$ を各 i について定め、

$$v(z) = \sum_i C_i v_i(z), \quad w(z) = \sum_i C_i w_i(z), \quad y(z) = \sum_i C_i y_i(z)$$

とし、さらに、$F(z) = v(z)w(z) - y(z)$ と書くと、$z = r_5, r_6$ に対応する条件式は、

$$F(r_5) = 0, \quad F(r_6) = 0$$

と書ける。すなわち、条件式が入力値から出力値を正しく計算するとき、多項式 $F(z)$ は各演算子に対応する値 r_5, r_6 を解に持つ。

条件式が満たすべき値の全てを解 $\mathcal{R}$ $(= \{r_5, r_6\})$ に持つ多項式 $t(z) = \prod_{r_i \in \mathcal{R}}(z - r_i)$ を考えると、

$$\forall r_i \in \mathcal{R}: \quad t(r_i) = 0 \Rightarrow F(r_i) = 0$$

が成り立ち、多項式 $t(z)$ は多項式 $F(z)$ を多項式として割り切る。すなわち、ある多項式 $h(z)$ が存在して、全ての $z \in \mathbb{Z}$ について、

$$t(z)h(z) = F(z) = v(z)w(z) - y(z)$$

が成り立たなければならない。この条件式は、ランダムに選んだ特定の値 $s \in \mathbb{Z}/q\mathbb{Z}$ を z に代入しても成り立つから、

$$t(s)h(s) = F(s) = v(s)w(s) - y(s) \tag{14}$$

も成り立つ。ここで s は定数なので、$t(s), h(s), v(s), w(s), y(s)$ も多項式ではなく値 (平文) になっていることに注意する。これらの平文を準同型写像 ϕ で対応させて暗号文に変換する。

$$t(s) \xrightarrow{\phi} T(s) = g^{t(s)}$$
$$h(s) \xrightarrow{\phi} H(s) = g^{h(s)}$$
$$v(s) \xrightarrow{\phi} V(s) = g^{v(s)}$$
$$w(s) \xrightarrow{\phi} W(s) = g^{w(s)}$$
$$y(s) \xrightarrow{\phi} Y(s) = g^{y(s)}$$

式 (14) で表される条件式が 2 つの平文の積を含むことから、暗号文の演算によって条件式が満たされることを確認できるようにするために、同じ条件式を暗号文上では双線形写像を用いて表す必要がある。平文の和 (差) は暗号文の積 (除) で、平文の積は暗号文の双線形写像 $e : G \times G \to G_T$ で表すことにすると、式 (14) の平文の条件式は、次のような暗号文の条件式に対応づけられる。

$$e(T(s), H(s)) = \frac{e(V(s), W(s))}{e(g, Y(s))} \tag{15}$$

念のために確かめると、双線形性 $e(g^a, g^b) = e(g, g)^{ab}$ から、式 (15) は、

$$e(T(s), H(s)) = e(g^{t(s)}, g^{h(s)}) = e(g, g)^{t(s)h(s)}$$
$$e(V(s), W(s)) = e(g^{v(s)}, g^{w(s)}) = e(g, g)^{v(s)w(s)}$$
$$e(g, Y(s)) = e(g, g^{y(s)}) = e(g, g)^{y(s)}$$

を用いて

$$e(g,g)^{t(s)h(s)} = e(g,g)^{v(s)w(s)}/e(g,g)^{y(s)} = e(g,g)^{v(s)w(s)-y(s)}$$

と変形できる。式 (14) が成立すれば、$e(g,g)$ のべきが等しいことが分かる。暗号文上の条件式 (15) ではランダムに選んだ s が暗号文の形でしか現れないため、適切に構成すれば、証明者にも s が秘匿されたまま、条件式 (14) が検証されることになる。

これで、QAP と楕円曲線の双線形写像を用いて、多項式で表される平文上の条件式を、暗号文上の条件式で表す準備ができた。以下では、この関係を用いてゼロ知識証明を構成する。

非対話ゼロ知識証明 (NIZK)　ゼロ知識証明は証明者と検証者の 2 者間のプロトコルである。証明者は、対象 (ステートメント)x が、ある性質 (言語 $\mathcal{L}$) を満たす ($x \in \mathcal{L}$) ことを検証者に納得させることが目的である。ここで、$x \in \mathcal{L}$ か $x \notin \mathcal{L}$ の判定は、特別な情報を持つ者以外にとっては困難であることが求められる。このような言語 $\mathcal{L}$ の例として、論理式の充足可能性問題 (SAT:SATisfiability Problem) を例に考える。SAT は論理式の値を真にする入力が存在するような論理式の集合である。例えば、以下の論理式 x は、

$$x = (x_3 \vee x_4 \vee \bar{x}_1 \vee x_5) \wedge (\bar{x}_3 \vee x_4 \vee x_5) \wedge (x_3 \vee \bar{x}_4 \vee \bar{x}_1) \wedge (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_5)$$

$x_1 = $ 真$, x_2 = $ 偽$, x_3 = $ 真$, x_4 = $ 真$, x_5 = $ 偽 の時に、真 となるから、$x \in$ SAT である。SAT は NP 完全問題の一つであり、変数の数が増えると急速に難しくなり、総当たり法よりも良い解法は知られていない。しかし、この例のように、解の 1 つが与えられれば、非常に効率良く $x \in$ SAT を判定できることが分かる。このような役割を果たす補助情報のことを、証拠 (witness) という。ゼロ知識証明では、witness を知っている証明者が、witness を知らない検証者に対して、witness の情報を一切与えずに $x \in \mathcal{L}$ を納得させることができる。また、このようなステートメント x が witness w を用いて $x \in \mathcal{L}$ を判定できるとき、「x と w は関係 R を満たす。」と言い、$(x,w) \in R$ と書く。

zk-SNARKs においては、ゼロ知識証明が以下のように用いられている。まず、任意の条件式の成立を $F(z) = 0$ とし、対応する QAP の各ワイヤの値を $C_1, \ldots, C_m$ とする。このとき、$x = (C_1, \ldots, C_n)$ を入力ワイヤ、残りのワイヤ $w = (C_{n+1}, \ldots, C_m)$ を witness とすると、言語 $\mathcal{L}$ は以下で定義される。

$$\begin{aligned}\mathcal{L} = \{ \langle C_1, \ldots, C_n \rangle \mid & \ t(z) \text{が} (v(z)w(z) - y(z)) \text{を割り切る。すなわち、} \\ & \text{ある} C_{n+1}, \ldots, C_m \text{と多項式} h(z) \text{が存在して,} \\ & t(z)h(z) = \left(\sum_i C_i v_i(z)\right)\left(\sum_i C_i w_i(z)\right) - \left(\sum_i C_i y_i(z)\right) \}\end{aligned}$$

初期設定　証明者と検証者の双方に、条件式 F に依存した共通の情報 (crs_F: Common Reference String) を入力する。

$$\begin{aligned}\mathrm{crs}_F = & (t(z), \{v_1(z), \ldots, v_m(z)\}, \{w_1(z), \ldots, w_m(z)\}, \{y_1(z), \ldots, y_m(z)\}, \\ & g, g^s, g^{s^2}, \ldots, g^{s^m}, \\ & g^{t(s)}, \{g^{v_1(s)}, \ldots, g^{v_m(s)}\}, \{g^{w_1(s)}, \ldots, g^{w_m(s)}\}, \{g^{y_1(s)}, \ldots, g^{y_m(s)}\})\end{aligned}$$

証明 π の生成　witness $w = (C_{n+1}, \ldots, C_m)$ を知っている証明者は、証明 $\pi = (V(s), W(s), Y(s), H(s))$ を次のようにして計算する。

$$V(s) = \prod_i \left(g^{v_i(s)}\right)^{C_i}, \quad W(s) = \prod_i \left(g^{w_i(s)}\right)^{C_i}, \quad Y(s) = \prod_i \left(g^{y_i(s)}\right)^{C_i}$$

$$h(z) = \frac{\left(\sum_i C_i v_i(z)\right)\left(\sum_i C_i w_i(z)\right) - \left(\sum_i C_i y_i(z)\right)}{t(z)} = \sum_i h_i z^i \text{ とおいて、}$$

$$H(s) = \prod_i \left(g^{s^i}\right)^{h_i} = g^{h(s)}$$

証明 π の検証　証明 $\pi = (V(s), W(s), Y(s), H(s))$ を以下の式で検証する。

$$T(s) = \prod_i \left(g^{s^i}\right)^{t_i} = g^{t(s)} \text{ を求め、}$$

$$e(T(s), H(s)) = \frac{e(V(s), W(s))}{e(g, Y(s))} \text{が成立するか否か。}$$

zk-SNARKs は、以下のとおり、完全性、健全性、ゼロ知識性を満たすゼロ知識証明 (補論 F を参照) である。

完全性　完全性は、ゼロ知識証明において witness を知っている証明者が作成した証明 π が、常に検証者に受理される性質を言う。証明者は、witness $(C_{n+1}, \ldots, C_m)$ を用い、証明 π の生成手順に従って

$$\pi \leftarrow (V(s), W(s), Y(s), H(s))$$

を計算し、以下の等号の成立を確認することで $x = (C_1, \ldots, C_n) \in \mathcal{L}$ を検証できる。

$$e(T(s), H(s)) \stackrel{?}{=} \frac{e(V(s), W(s))}{e(g, Y(s))}$$

等号は

$$t(s)h(s) = v(s)w(s) - y(s)$$

のとき、s の値によらず常に成立する。

健全性　健全性は、ゼロ知識証明において witness を知らない攻撃者が作成した証明 $\hat{\pi}$ を、検証者が誤って受理する確率が無視できるほど小さいという性質である。

攻撃者が、$\hat{\pi} = (\widehat{V}(s), \widehat{W}(s), \widehat{Y}(s), \widehat{H}(s))$ を生成した時に、与えられた $T(s)$ に対して、以下の検証式を満たす必要がある。

$$e(T(s), \widehat{H}(s)) \stackrel{?}{=} \frac{e(\widehat{V}(s), \widehat{W}(s))}{e(g, \widehat{Y}(s))}$$

この検証式を満たすのは計算量的に困難であることが示される[40]。

[40]厳密には $V(s), W(s), Y(s), H(s)$ が witness から正しく作られたことの証明を、証明 π の生成手順に付加することで、関連する全ての検証式を満たす $\hat{\pi}$ の生成が困難であることから示されるが、見通しが悪くなるため割愛した。

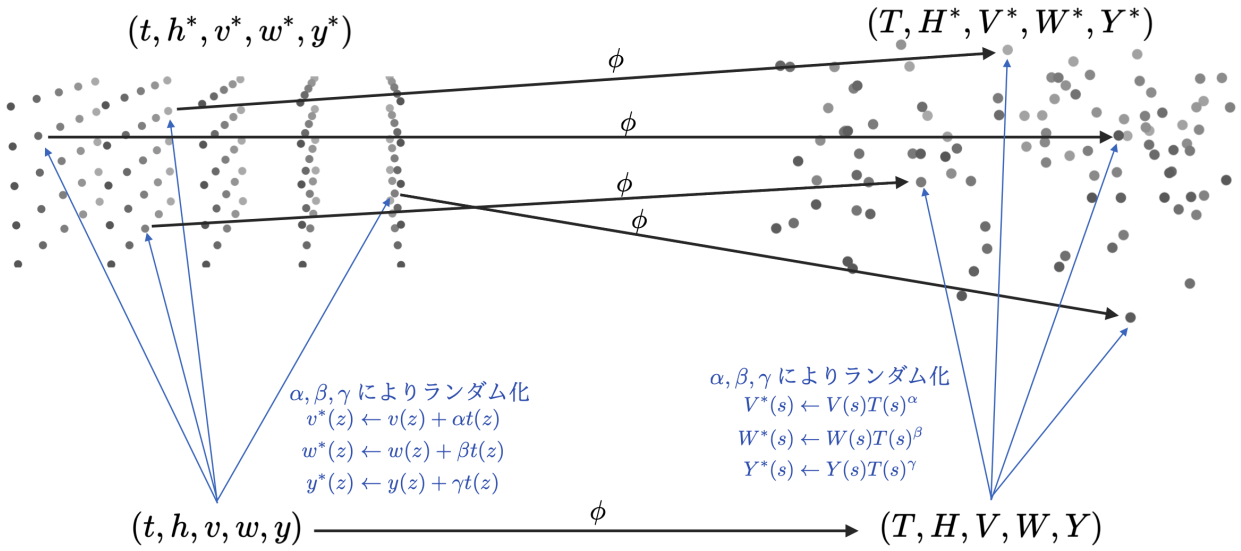


図 16: zk-SNARKs におけるゼロ知識性の原理。$t(z)$ が $v(z)w(z)-y(z)$ を割り切るという関係は $(v, w, y) \to (v^*, w^*, y^*)$ のランダム化後も不変。同様に暗号文 $(V, W, Y) \to (V^*, W^*, Y^*)$ のランダム化でも不変である。ランダム化後の証明 π^* が条件式 $e\left(T(s), H^*(s)\right) = \frac{e(V^*(s), W^*(s))}{e(g, Y^*(s))}$ を満たすことは確認できるが、$\pi^* = (V^*, W^*, Y^*, H^*)$ は一様乱数と区別がつかず、witness に関する情報は一切得られない。

ゼロ知識性　ゼロ知識性は、witness を知っている証明者が作成した π が、一様乱数の組と区別できない性質である。この性質は、$t(z)$ が $v(z)w(z) - y(z)$ を割り切るという条件が、$v(z), w(z), y(z)$ に $t(z)$ を加えても成立するという性質を利用し、証明 π をランダム化した証明 π^* を生成することで達成される[41]。この性質を利用して、以下のランダム化を考える。

$$
\begin{aligned}
v^*(z) &\leftarrow v(z) + \alpha t(z) \\
w^*(z) &\leftarrow w(z) + \beta t(z) \\
y^*(z) &\leftarrow y(z) + \gamma t(z)
\end{aligned}
$$

ランダム化後の $v^*(z)w^*(z) - y^*(z)$ を $t(z)$ が割り切るという条件は、元の条件と同値である。

$$
\begin{aligned}
t(z) \mid v(z)w(z) - y(z) &\iff t(z) \mid v^*(z)w^*(z) - y^*(z) \\
&\iff t(z) \mid v(z)w(z) - y(z) + t(z)\left(\beta v(z) + \alpha w(z) + \alpha\beta t(z) - \gamma\right)
\end{aligned}
$$

従って、$z \leftarrow s$ を代入して準同型暗号 ϕ で写像した暗号文についても、$V(s), W(s), Y(s), T(s)$ に対する以下のランダム化手順で求められる。

$$
\begin{aligned}
V^*(s) &\leftarrow V(s)T(s)^\alpha \\
W^*(s) &\leftarrow W(s)T(s)^\beta \\
Y^*(s) &\leftarrow Y(s)T(s)^\gamma
\end{aligned}
$$

[41] π^* は完全性と健全性も満たす。本稿では、順序立てて平易な解説をする目的で、ランダム化を当初から導入することを避けた。

同様に、$H^*(s)$ も計算できて[42]、以下の関係を満たすランダム化された証明 $\pi^* = (V^*, W^*, Y^*, H^*)$ が生成される。

$$e\left(T(s), H^*(s)\right) = \frac{e\left(V^*(s), W^*(s)\right)}{e\left(g, Y^*(s)\right)}$$

$\pi^* = (V^*, W^*, Y^*, H^*)$ が G 上の元 (スカラー値) であることを考慮すると、得られている変数の数と未知数 s, α, β, γ の数が等しいため、未知数の自由度から π^* は G 上の一様分布 (U_G, U_G, U_G, U_G) と区別がつかず、witness に関する情報を一切得ることができない。

ロ　Zerocoin と Zerocash

Zerocoin (Miers et al. [2013]) は NIZK を利用して匿名性を達成する。Zerocoin では固定金額の UTXO を匿名コインとしてブロックチェーンに登録し、匿名コインで支払うことにより、送金者と受領者の関係を秘匿する。例えば、固定額（例えば$1）のコインを生成する場合、送金者はランダムなシリアル番号 S と乱数 r を生成して、そのコミットメント C を公開し、同時に同額の支払いを実行する。正しく C が作られ、同額の支払いを伴っていることを、誰でも確認できるとき、C は有効であると定義される。C を換金する際には、シリアル番号 S を指定し、有効なコミットメントのリスト $(C_1, \ldots, C_N)$ を集め、(1) C が $(C_1, \ldots, C_N)$ に含まれていること、(2) C に対応する乱数 r を知っていることを NIZK π で証明する。すなわち、(S, π) を匿名コインとして利用できる。

ただし、Zerocoin の匿名性には以下の主要な制限がある。

a) 送金者はコイン C のシリアル番号 S を知っているので、受領者が償還する前に先に償還する危険がある。また、コインが譲渡された場合、S を追跡することにより、受領者を特定できる可能性がある。

b) 固定額のコインしか発行できない。

Zerocash (Ben Sasson et al. [2014]) は、zk-SNARKs を利用して匿名性と取引の秘匿性を同時に達成している。Zerocash では新たに分散台帳に基づくデジタル通貨における匿名性の概念として“台帳識別不能性”(Ledger-Indistinguishability) を導入して、匿名性と取引の秘匿性を包括的に取り扱うことを可能にしている。台帳識別不能性は、2 つの台帳 1 と台帳 2 に対して、攻撃者が同じ種類の取引（アドレス生成、匿名コインの発行 (mint)、交換 (pour)、償還 (receive) 等）をする機会が与えられても、取引の前後で台帳がシャッフルされると、攻撃者は 2 つの台帳のどちらが台帳 1 であったか区別できないという性質である。この性質を実現するためには、台帳に記録された取引が暗号文であることが求められ、暗号化された台帳でコインの偽造等が行われていない等の安全性条件を確認する必要がある。そのような複雑な条件の確認に前述の zk-SNARKs が用いられている。まとめると、

a) Zerocash は、zk-SNARKs とコミットメントを利用して、送金者のアドレスを秘匿し、任意の金額のコインを発行可能としている。

b) Zerocash は、コインが送金されると、受領者の新しい秘密鍵を使って新しいシリアルナンバー S のコインが発行される。このメカニズムにより、S を知っていてもコインを追跡できない。

[42] 証明者は、$H^*(s) = H(s)V(s)^{\beta}W(s)^{\alpha}T(s)^{\alpha\beta}g^{-\gamma}$ で求められる。

表 1: プライバシー保護ブロックチェーン技術の比較

方式	匿名性とその限界	取引内容の秘匿性	主なスキーム
Mixing	送金者と受領者の関係を秘匿。ただし、取引当事者内に限定。	なし	CoinSwap, CoinJoin 等
リング署名	支払い者の匿名性のみ。任意の集団に送金者を秘匿できるが、取引サイズが集団サイズに比例して増大する。	なし	Monero
zk-SNARKs	完全な匿名性	あり	Zerocoin, Zerocash 等
完全準同型暗号	完全な匿名性	あり	Mitani and Otsuka [2020a,b] 等

c) 受領者の公開鍵を用いて取引内容を秘匿できる。

Zerocash の欠点の一つに、多くの計算を要することが挙げられる。zk-SNARKs は秘匿した取引（暗号文）に関して高次の多項式で表される条件式を QAP(Gennaro et al. [2013]) を用いて暗号文の 2 次式に還元して NIZK を構成する必要があることに起因する。

(4) 各プロトコルの匿名性と計算コスト

表 1 に、プライバシー保護ブロックチェーン技術の特性を比較した。集中型 Mixing と分散型 Mixing は、Mixing サービスで送金者アドレスと受領者アドレスの取引関係を秘匿することを目的としている。両者の違いは、前者がこのタスクを行うために Mixing 事業者を必要とするのに対し、後者は Mixing 事業者を仲介させず、参加者同士が協力して分散的に Mixing を行う点である。両者に共通するデメリットとしては、Mixing 処理に伴う取引の遅延や取引内容を秘匿できないことである。このうち、取引内容については、Mixing では取引額を固定された金額に揃えることが要求されるため、取引に関連する情報を完全に秘匿することができない。これ以外の、デメリットとしては、集中型 Mixing では、Mixing サービスの利用者がサービス利用料を負担する必要があること、Mixing 事業者が単一故障点であるという意味でサービスを支える仕組みに脆弱性があること、Mixing の参加者がプロトコルの実行を拒否して Mixing の実行を阻害するリスクなどがある。さらに、分散型 Mixing における参加者間の頻繁な通信は、ネットワークに高い伝送オーバーヘッドを引き起こすことになり、Mixing サービスのスケーラビリティを制限する可能性がある。

リング署名は送金者（署名者）の匿名性を達成できる。加えて、CryptoNote は、リング署名の他に、one-time 支払い、Confidential Transaction を利用して、送金者（署名者）の匿名性、受領者の匿名性をそれぞれ達成している。これらの方式のデメリットは、署名サイズがグループの参加者数（公開鍵の数）に比例することであり、公開鍵が増えるとブロックチェーンを維持するためのストレージや通信コストを圧迫する。

リング署名は典型的な匿名署名[43]であるため、グループの参加者の 1 人が署名したことは検証できるが、署名者は特定できないという意味で、署名者の匿名性を達成できる。CryptoNote は、リング署名の他に、one-time ペイメント、Confidential Transaction といった技術を利用して、取引内容だけでなく、送金者（署名者）の匿名性、受取者の匿名性をそれぞれ達成している。この方式のデメリットは、署名サイズがグループのメンバー数（公開鍵の数）に

[43]署名者の匿名性を実現する署名では、リング署名、ブラインド署名、グループ署名などが知られている。

比例することであり、公開鍵が増えるとストレージや通信回線の容量を圧迫する。署名サイズを小さくするために公開鍵の数を抑えた場合には、匿名性が弱まり、取引グラフ分析等 (4節) により送金者が特定されるリスクが高まる。

NIZK 証明をコミットメントと共に用いることで、Zerocoin や Zerocash 等で匿名性と取引の秘匿性の達成を可能にしている。NIZK 証明は、匿名かつ追跡不可能な方法で、コインの所有していることの証明を可能にする。Zerocash 等は、NIZK 証明とコミットメントという 2 つの技術を共に用いることで、匿名性と取引の秘匿性の達成している。コミットメントは取引内容を秘匿しつつ、取引内容を固定し、後に取引内容を変更ができない状態で内容を確認することができる。これら 2 つの技術を用いることで、Zerocash は、匿名性と取引の秘匿性を達成しつつ、さらに取引サイズを一定に保つことができる。他方、Zerocash で使用されている zk-SNARKs の NIZK プロトコルは、高い計算負荷が発生する。

本稿では詳しく取り上げなかったが、取引内容を秘匿したまま計算を実行できる準同型暗号に基づくシステム（Homomorphic Encryption System）も重要かつ優れたプライバシー保護技術である。準同型暗号を利用した暗号資産のプロトコルとして、Monero で採用されている準同型コミットメントを応用した Confidential Transaction（CT）が提案されている。さらに、最近では暗号文に対する加法演算と乗法演算の両方を利用できる完全準同型暗号を用いて匿名性、取引の秘匿を同時に達成し、監査可能性を実現する方式も提案されている (Mitani and Otsuka [2020a], Mitani and Otsuka [2020b])。

5　まとめとデジタル通貨へのインプリケーション

2008 年のサトシ・ナカモト (Nakamoto [2008]) によるビットコインの提案以降、ブロックチェーンは産業イノベーションが猛烈な勢いで進んだため、学術的な研究が遅れ気味だったが、近年ではメジャーな暗号および情報セキュリティの国際会議でもブロックチェーンのセッションが設けられるまでになっている。

現在も新しいコンセンサスアルゴリズムに基づくブロックチェーンが提案されており、学術的な安全性評価が追いついていない状況にある。ビットコインの安全性については相対的に最も研究が進んでおり、本稿で述べた安全性評価はその成果の一部を解説したものである。ビットコインは 2008 年の提案以降、マイニングに伴う消費電力量の問題や貨幣価値の不安定性など、さまざまな問題を抱えながらも、大きなセキュリティ上の危殆化はなく最大の発行高を誇る暗号資産の地位を維持している。改ざん不可能性に加えて、利用者が存在する限り根絶しない強靱性もブロックチェーンの大きな魅力である。

また、匿名性はデジタル通貨を一般の消費者取引に普及させる上で不可欠な性質である。現在のプライバシー保護技術の多くは、利用者がコストを掛けて意図的に匿名性を付与しなければならない状況にある。この結果、匿名サービスの多くは高付加価値の商業取引か不正取引の隠蔽に利用され、多くの正当な小口取引では匿名性を高めるサービスを利用できないでいると推察される。こうした状況に対して、zk-SNARKs や完全準同型暗号によるプライバシー保護技術が提案されており、これらは安価かつ効率的な匿名サービスの提供を可能にすると期待される。もっとも、既存のブロックチェーンにプライバシー保護技術を取り入れる変更には非常に大きな困難を伴うことが予想される。また、マネーロンダリングや犯罪防止のための監査可能性をどのような条件で加えるべきかといった論点も定まっていない。こうした、匿名性と監査可能性の両立については、さらなる研究提案や社会的コンセンサスの醸成を待つ必要がある。今後、関連する研究が引き続き活発化すると予想されることから、利用者の負担の軽減と匿名性の向上を両立させるような技術の今後の提案や改善に期待したい。

参考文献

Ben Sasson, Eli, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza, “Zerocash: Decentralized Anonymous Payments from Bitcoin (Extended Version),” Proceedings of IEEE Symposium on Security and Privacy 2014, IEEE, 2014, pp. 459–474.

Blum, Manuel, Paul Feldman, and Silvio Micali “Non-Interactive Zeroknowledge and its Applications,” Proceedings of Annual ACM Symposium on Theory of Computing `88, Association for Computing Machinery, 1988, pp. 103–112.

Boneh, Dan, Eu-Jin Goh, and Kobbi Nissim, “Evaluating 2-DNF Formulas on Ciphertexts,” Proceedings of Theory of Cryptography Conference (TCC) 2005, Lecture Notes in Computer Science, 3378, Springer, 2005, pp. 325–341.

Castro, Miguel, “Practical Byzantine Fault Tolerance,” Ph.D. Dissertation, Massachusetts Institute of Technology, 2001.

———, and Barbara Liskov, “Practical Byzantine Fault Tolerance” Proceedings of Symposium on Operating Systems Design and Implementation `99, Association for Computing Machinery, 1999, pp. 173–186.

Chaum, David, “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms,” *Communications of the ACM*, 24(2), Association for Computing Machinery, 1981, pp. 84–90.

———, “Blind Signatures for Untraceable Payments,” Proceedings of CRYPTO `82, Lecture Notes in Computer Sciences, 1440, Springer, 1983, pp. 199–203.

Dobbertin, Hans, Antoon Bosselaers, and Bart Preneel, “RIPEMD-160: A Strengthened Version of RIPEMD,” Proceedings of International Workshop on Fast Software Encryption `96, Lecture Notes in Computer Sciences, 1039, Springer, 1996, pp. 71–82.

Eyal, Ittay, and Emin Gün Sirer, “Majority Is Not Enough: Bitcoin Mining Is Vulnerable,” Proceedings of International Conference on Financial Cryptography and Data Security 2014, Lecture Notes in Computer Sciences, 8437, Springer, 2014, pp. 436–454.

———, “Majority Is Not Enough: Bitcoin Mining Is Vulnerable,” *Communications of the ACM*, 61(7), Association for Computing Machinery, 2018, pp. 95–102.

Feld, Sebastian, Mirco Schönfeld, and Martin Werner, “Analyzing the Deployment of Bitcoin’s P2P Network under an AS-Level Perspective,” *Procedia Computer Science*, 32, Elservier, 2014, pp. 1121–1126.

Eiichiro, Fujisaki, “Sub-Linear Size Traceable Ring Signatures without Random Oracles,” Proceedings of Cryptographers’ Track at the RSA Conference 2011, Lecture Notes in Computer Science, 6558, Springer, 2011, pp. 393–415.

———, and Koutarou Suzuki, “Traceable Ring Signature,” Proceedings of International Workshop on Public Key Cryptography 2007, Lecture Notes in Computer Science, 4450, Springer, 2007, pp. 181–200.

Galbraith, Steven D., and Pierrick Gaudry, “Recent Progress on the Elliptic Curve Discrete Logarithm Problem,” *Designs, Codes and Cryptography*, 78(1), Springer, 2016, pp. 51–72.

Garay, Juan, Aggelos Kiayias, and Nikos Leonardos, “The Bitcoin Backbone Protocol: Analysis and Applications,” Proceedings of EUROCRYPT 2015, Lecture Notes in Computer Science, 9057, Springer, 2015, pp. 281–310.

Gennaro, Rosario, Craig Gentry, Bryan Parno, and Mariana Raykova, “Quadratic Span Programs and Succinct NIZKs without PCPs,” Proceedings of EUROCRYPT 2013, Lecture Notes in Computer Science, 7881, Springer, 2013 pp. 626–645.

Gentry, Craig, Shai Halevi, and Nigel P. Smart, “Fully Homomorphic Encryption with Polylog Overhead,” Proceedings of EUROCRYPT 2012, Lecture Notes in Computer Science, 7237, Springer, 2012, pp. 465–482.

Heilman, Ethan, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg, “TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub,” Proceedings of Network and Distributed System Security Symposium 2017, Internet Society, 2017 (available at https://www.ndss-symposium.org/wp-content/uploads/2017/09/ndss201701-3HeilmanPaper.pdf、2021 年 6 月 8 日).

Koblitz, Neal, “Elliptic Curve Cryptosystems,” *Mathematics of computation*, 48(177), American Mathematical Society, 1987, pp. 203–209.

Lamport, Leslie, Robert E. Shostak, and Marshall C. Pease, “The Byzantine Generals Problem,” *ACM Transactions on Programming Languages and Systems*, 4(3), Association for Computing Machinery, 1982, pp. 382–401.

Maxwell, Gregory, “Coinjoin: Bitcoin Privacy for the Real World,” 2013a (available at https://bitcointalk.org/index.php?topic=279249、2021 年 6 月 8 日).

———, “CoinSwap: Transaction Graph Disjoint Trustless Trading,” 2013b (available at https://bitcointalk.org/index.php?topic=321228.0、2021 年 6 月 8 日).

Meiklejohn, Sarah, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage, “A Fistful of Bitcoins: Characterizing Payments Among Men with No Names,” Proceedings of Conference on Internet Measurement Conference 2013, Association for Computing Machinery, 2013, pp. 127–140.

Merkle, Ralph C., “Method of Providing Digital Signatures,” Google Patents, 1982 (available at https://www.google.com/patents/US4309569、2021 年 6 月 8 日).

Miers, Ian, Christina Garman, Matthew Green, and Aviel D. Rubin, “Zerocoin: Anonymous Distributed E-Cash from Bitcoin,” Proceedings of IEEE Symposium on Security and Privacy

2013, IEEE, 2013, pp. 397–411.

Mitani, Tatsuo, and Akira Otsuka, “Confidential and Auditable Payments,” Proceedings of International Conference on Financial Cryptography and Data Security, Lecture Notes in Computer Science, 12063, Springer, 2020a, pp. 466–480.

———, and ———, “Traceability in Permissioned Blockchain,” *IEEE Access*, 8, IEEE, 2020b, pp. 21573– 21588 (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8970301、2021 年 6 月 8 日).

Nakamoto, Satoshi, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008 (available at https://bitcoin.org/bitcoin.pdf、2021 年 6 月 8 日).

Noether, Shen, and Adam Mackenzie, “Ring Confidential Transactions,” *Ledger*, 1, University of Pittsburgh, 2016, pp. 1–18.

Pollard, John M., “Monte Carlo Methods for Index Computation (mod *p*),” *Mathematics of Computation*, 32(143), American Mathematical Society, 1978, pp. 918–924.

QuantumMechanic, “Proof of Stake Instead of Proof of Work,” 2011 (available at https://bitcointalk.org/index.php?topic=27787.20、2021 年 6 月 8 日).

RHorning, “Mining Cartel Attack,” 2010 (available at https://bitcointalk.org/index.php?topic=2227.0、2021 年 6 月 8 日).

Rivest, Ronald L., Adi Shamir, and Yael Tauman, “How to Leak a Secret,” Proceedings of ASIACRYPT 2001, Lecture Notes in Computer Science, 2248, Springer, 2001, pp. 552–565.

Roetteler, Martin, Michael Naehrig, Krysta M. Svore, and Kristin Lauter, “Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms,” Proceedings of ASIACRYPT 2017, Lecture Notes in Computer Science, 10625, Springer, 2017, pp. 241–270.

Ron, Dorit, and Adi Shamir, “Quantitative Analysis of the Full Bitcoin Transaction Graph,” Proceedings of International Conference on Financial Cryptography and Data Security. Lecture Notes in Computer Science, 7859, Springer, 2013, pp. 6–24.

van Saberhagen, Nicolas, “CryptoNote v2.0,” 2013 (available at https://bytecoin.org/old/whitepaper.pdf、2021 年 6 月 8 日).

Shor, Peter W., “Algorithms for Quantum Computation: Discrete Logarithms and Factoring,” Proceedings of Annual Symposium on Foundations of Computer Science 1994, IEEE, 1994, pp. 124–134.

Sun, Shi-Feng, Man Ho Au, Joseph K. Liu, Tsz Hon Yuen, and Dawu Gu, “RingCT 2.0: A Compact Accumulator-Based (Linkable Ring Signature) Protocol for Blockchain Cryptocurrency Monero,” Proceedings of European Symposium on Research in Computer Security 2017, Lecture Notes in Computer Science, 10493, Springer, 2017, pp. 456–474.

A　ビザンチン将軍問題

ビザンチン軍の複数の師団が敵城の外に陣を張り、各将軍がそれぞれの師団の指揮をしている状況を考える。伝令を使って互いに連絡を取り合い、将軍らは各師団の行動を決定しなければならない。しかし、将軍の何人かは敵に寝返っており、ビザンチン軍の統一行動[44]を阻止しようとしている。どのような作戦を実行すれば以下の目標を達成できるか？

A 忠実な将軍の師団は全て統一行動を取る。

B 将軍の寝返りが少数であれば、統一行動は乱されない。

仮に師団が 3 つで、将軍の 1 人が寝返っているとしよう。この場合、残った 2 人の将軍が「攻撃」か「撤退」のいずれかの統一行動をとることが目標である。しかし、寝返った将軍は、軍の統率を混乱させるために、「攻撃」と「撤退」の矛盾するメッセージを伝令を使ってそれぞれの将軍に伝える。2 人の忠実な将軍のうち少なくとも 1 人は、「攻撃」と「撤退」の 2 つのメッセージを受け取って困惑し、寝返った将軍の思惑通り統率が乱されることになる。3 人の場合は 1 人でも寝返ると目標は達成出来ない。ビザンチン将軍問題は、一般に寝返った将軍の数が全将軍の数の 1/3 未満でなければ合意に到達できないことが知られている (Lamport, Shostak, and Pease [1982])。

将軍をノード、伝令をインターネット上で送信される非同期メッセージと捉えると、分散システムにおける合意形成も同じ問題をはらんでいる。分かりやすい例に、二重支払い（Double-spending）の問題が挙げられる。ビットコインではシステムを効率化するために、暗号資産を受領した金額を分割使用させず、全額を次の支払いに使用しなければならないルールを設けている。未使用の暗号資産は UTXO（Unspent Transaction Output）と呼ばれる単位で管理され、UTXO は一度だけ他の支払いに充てることができることになっている。話しを戻し、同じ UTXO を原資とする 2 つの支払い取引 A と B が存在したとする。A と B のいずれか一方を承認すると、他方は二重支払い取引なので無効としなければならない。すなわち、A と B は排他的であり、どちらを承認するかをシステム全体で合意しなければならない。しかし、これは簡単ではなく、冒頭のビザンチン将軍問題と本質的に同じ問題になっている。未使用残高を管理できるタイプの暗号資産でも、超過払い（Over-spending）問題として本質的に同じ問題が存在する。

B　PBFT 合意プロトコル

合意形成アルゴリズムの目的は、分散システムが状態レプリケーションを達成することである。

定義 B.1 (状態レプリケーション/State Replication)**.** 全ノードが同一の（無限長）コマンド列 $c_1, c_2, c_3, \ldots$ を同一順序で実行する時、そのノード集合は**状態レプリケーション** *(state replication)* を達成するという。

ブロックチェーンにおける状態レプリケーションは、単に定義中の“コマンド”を“取引”と読み替えればよい。素朴に考えれば、単一ノードをプライマリに選定し、プライマリが決定したコマンド実行順序に従って他のノードがコマンドを同一順序で実行すれば、状態レプ

[44]例えば、総攻撃か一斉撤退かなど。総攻撃なら敵城を落とせるが、師団の一部が攻撃しているにも関わらず残りの師団が撤退すると、ビザンチン軍は壊滅的なダメージを受ける状況を想定している。

リケーションは達成される。このようなプライマリノードのことをシリアライザ（Serializer）と呼ぶ。ただし、シリアライザが次に述べるビザンチンの場合、状態レプリケーションは必ずしも達成されない。

定義 B.2 (ビザンチン故障)**.** 任意の動作するノードを**ビザンチン**（byzantine）あるいはビザンチンノードと呼ぶ。ビザンチンには可能な動作の全てが許される。

例えば、全くメッセージを送信しない、想定外のメッセージを想定外のノードに送信する、または入力値を偽るなどの動作を許される。ビザンチンの動作には結託も含まれ、全てのビザンチンノードが 1 人の攻撃者に操られている状況も想定しなければならない。ビザンチンは信頼性工学に由来する用語であり、暗号理論や情報セキュリティにおける攻撃者（adversary）と同義である。ここでは両者を区別せずに用いる。

全ての通信は他のノードに仲介されることなく、2 つのノード間で直接行われると仮定し、送信元アドレスは詐称できないと仮定する。この仮定は、単一のビザンチンノードが全てのノードになりすますことが出来ないようにするための条件である。

PBFT（Practical Byzantine Fault-Tolerance）は、定義 B.1 の状態レプリケーションを実現する最初のプロトコル (Castro and Liskov [1999], Castro [2001]) である。全体で n 個のノードがあり、この中に最大 f 個のビザンチンノードが含まれると仮定する。ただし、ビザンチンは全体の 1/3 未満で、$n = 3f + 1$ とする。外部のクライアントから、分散システム内のノードに取引が次々と送られてくる状況を想定する。PBFT 合意プロトコルの目的は、全てのノードが同じ順序で取引を処理すること（状態レプリケーションの達成）である。

PBFT の概要は次の通りである。全てのメッセージには送信元のデジタル署名が付されており、メッセージ送信元のノードを正しく識別できる。各ノードはそれぞれ識別番号 $i = 0, \ldots, n-1$ を有しており、i 番目のノードを n_i と書く。各ノード n_i で個別に管理される非負の整数 v を**ビュー**（*view*）と呼び、ノード n_i は $v \bmod n$ で指定されるノードをプライマリと認識する。このとき**ノード n_i はビュー v にある**と言う。全てのノードは、それぞれ 1 つのノードを**プライマリ**と見なし、他のノードを**バックアップ**と見なす。ノードは自身がプライマリの間、シリアライザとして取引の実行順序（取引のシーケンス番号）を決定し、バックアップに取引の実行順（シーケンス番号）を提案する。バックアップは、プライマリの障害を検知すると、自身のビューを更新 $v \leftarrow v + 1$ し、持ち回り順で次のノードをプライマリと見做す。これを**ビューの変更**と呼ぶ。ビューの変更が合意されると、新しいプライマリノードがシリアライザを担当する。

全ノードはビュー $v = 0$ から開始する。ビューは各ノードが独自に判断して v の値を更新するため、ノード毎に異なるビューにある可能性を考慮する必要がある。また、各ノードは、受信したメッセージが定められたフォーマットに従っており、かつ同じビュー v に属している時に限り、当該メッセージを受理すると仮定する。以上を考慮すると、PBFT 合意プロトコルは、任意のシーケンス番号 s（実行順序）について、正しいノードは複数の取引 t, t' を実行しないことを保証できる。すなわち、s 番目に実行する取引 t は、正しいノード全体で同一であることを保証できる。

これは、次の補題から証明できる。

補題 B.3 (コーラム共通集合)**.** $|S_1| \geq 2f + 1$、$|S_2| \geq 2f + 1$ を満たすノード集合 S_1, S_2 について、共通集合 $S_1 \cap S_2$ に正しいノードが存在する。

Proof. 全体のノード数は $n = 3f + 1$ であるから、鳩の巣原理により、共通集合 $S_1 \cap S_2$ は少なくとも $f + 1$ のノードを含む。一方、故障ノードは高々 f なので、$S_1 \cap S_2$ は少なくとも 1 つの正しいノードを含む。 □

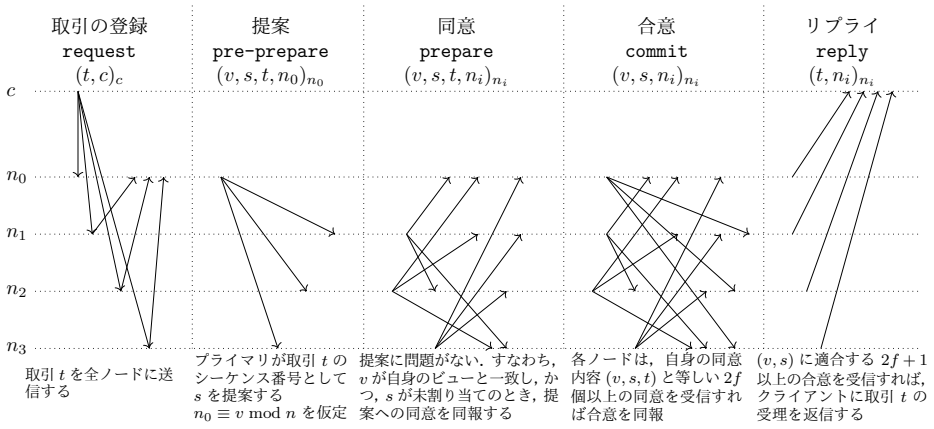


図 17: PBFT 合意プロトコル。例は 4 つのノードからなり、ノード n_0 は現在のビュー v のプライマリである。非同期メッセージのため、メッセージの到着順序と発信順序は異なる。クライアント c は $f+1$ 以上の reply を受信することで、取引の実行順序が合意されたことを確認する。

pre-prepare$(v, s, t, \cdot)$ と $2f$ 個の prepare$(v, s, t, \cdot)$ を合わせて、(v, s, t) の**合意証明**と呼ぶ。ただし、$\cdot$ は任意のノード番号を表す。

補題 B.4 (シーケンス番号の一意性)**.** あるノードが (v, s, t) の**合意証明**を得たとき、どのノードも $(v, s, t' \neq t)$ の**合意証明**を得ることはできない。

Proof. 2 つの（必ずしも別個ではない）ノードが異なる合意証明 $(v, s, t), (v, s, t')$ を得たと仮定する。合意証明は $2f+1$ 個のメッセージを含むので、補題 B.3 によって、少なくとも 1 つの正しいノードが (v, s, t) と (v, s, t') のそれぞれについて pre-prepare または prepare を送信したことになる。正しいプライマリは、各 (v, s) に対して 1 つの pre-prepare のみを送信し、正しいバックアップは各 (v, s) に対して 1 つの prepare のみを送信するので、これは矛盾である。したがって、$t' = t$。 □

この補題により、正しいノードが同じシーケンス番号 s で 2 つの異なる取引 t, t' を実行する可能性は排除された。しかし、合意証明を得るまでに各ノードは少なくとも $2f+1$ 個の正しいメッセージの受信を待たなければならず、プライマリがビザンチンの場合、プロトコルは永久に停止する可能性が残っている。そこで、PBFT では故障判定タイマーを導入して、タイマーが作動すると図 18 のアルゴリズムを起動してビューの変更を提案し、プライマリを罷免する動作を行う。

まず、ノード n_i の故障判定タイマーが作動すると、n_i はビュー v での pre-prepare/prepare/commit メッセージの受理を停止し、以下のメッセージで $v+1$ へのビュー変更を提案する。

$$\texttt{view-change}(v+1, \mathcal{P}_i, n_i)_{n_i}$$

ここで、$\mathcal{P}_i$ は n_i がビュー v の間に収集した合意証明の集合である。

$2f+1$ 個のバックアップノードの故障判定タイマーが作動し、ノード n_{v+1} が $2f+1$ 個の view-change を受信すると、新しくプライマリに任命されたことを確信し、ビューの変更を実施する。

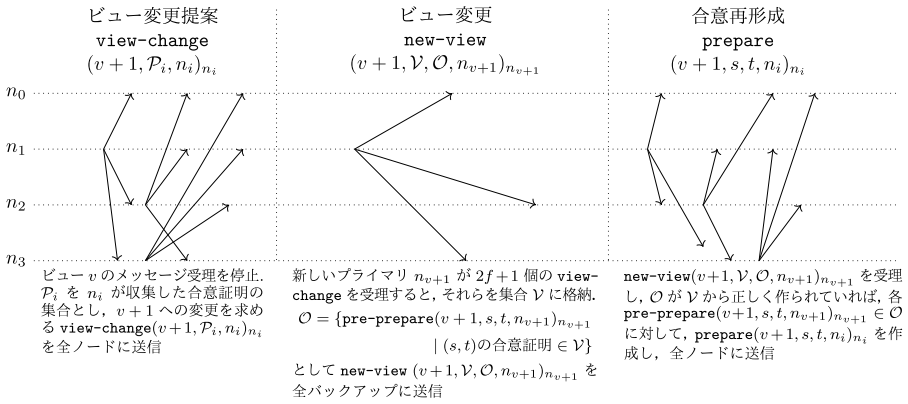


図 18: PBFT におけるビューの更新プロトコル:故障判定タイマーが作動すると、プライマリ n_v を罷免し、新しいプライマリ n_{v+1} が全ての合意証明を再構成する。

$$\mathcal{V} = \{\mathcal{P}_i \mid \text{受信した } 2f+1 \text{ 個の} \texttt{view-change}(v+1, \mathcal{P}_i, n_i)_{n_i}\}$$

$\mathcal{V}$ にはこれまでに発行された合意証明が全て含まれている。合意証明に含まれる各 (s,t) は、途中に抜けがある可能性がある。そこで、合意証明に含まれるシーケンス番号 s の最大値を $s^{\mathcal{V}}_{max}$ とし、抜けたシーケンス番号を null-取引で埋め、0 から $s^{\mathcal{V}}_{max}$ までの全ての値が揃った pre-prepare の集合として $\mathcal{O}$ を構成する。

$$\begin{cases} \texttt{pre-prepare}(v+1, s, t, n_{v+1})_{n_{v+1}} \in \mathcal{O} & \text{if } (s,t)\text{の合意証明} \in \mathcal{V} \\ \texttt{pre-prepare}(v+1, s, \texttt{null}, n_{v+1})_{n_{v+1}} \in \mathcal{O} & otherwise \end{cases}$$

バックアップは null-取引を「何もしない」と解釈する。この操作により、以後の取引に割り当てられるシーケンス番号は $s^{\mathcal{V}}_{max}$ より大きい値となり、後から受理される取引が前方に割り込んで実行順序が入れ替わるのを防ぐ。

一方、各バックアップノード n_i は、new-view(を) 受理すると、新しいビュー $(v+1)$ における合意再形成アルゴリズムに従って、new-view の $\mathcal{O}$ に含まれる全べての pre-prepare について冒頭の合意アルゴリズムを再実行する。これにより、$s^{\mathcal{V}}_{max}$ 以下の全てのシーケンス番号について、取引の実行順序 (取引とシーケンス番号の対応) に関する合意を新しいビュー $v+1$ で再形成する。

定理 B.5 (全てのビューにおけるシーケンス番号の唯一性)**.** PBFT は、正しいノードが (v,s,t) を実行する場合、以降のビュー $v' \geq v$ において正しいノードが $(v', s, t' \neq t)$ を実行することがないことを保証する。

すなわち、あるシーケンス番号 s に取引 t が一旦割り当てられると、この関係はどのビューにおいても不変であることが示される。言い換えると、PBFT では、プライマリーの罷免によりビューが更新されても、それまでに合意された取引の実行順序は変わることがないことが示される。

Proof. $v' = v$ の場合は補題 B.4 で証明済み。従ってまず、$v' > v$ かつ $n_{v'}$ が正しいプライマリになる最小の v' の場合を考える。

正しいノードが (v, s, t) を実行した場合、v' における正しいプライマリ $n_{v'}$ が発行する new-view$(v', \mathcal{V}, \mathcal{O}, n_{v'})$ の $\mathcal{O}$ には (v', s, t) にマッチする pre-prepare が含まれる。これにより、正しいノードが $(v', s, t' \neq t)$ についての合意証明を取得することがないことが保証される。

次に、$\mathcal{V}$ について考える。正しいノードが (v, s, t) を実行した場合、PBFT 合意プロトコルにおいて、少なくとも $2f+1$ 個のノードが commit$(v, s, \cdot)$ を送信していなければならない。このノードの集合を R_1 とすると、R_1 に含まれる正しいノードはすべて、事前に (v, s, t) の合意証明を取得している。一方、$\mathcal{V}$ には、$2f+1$ 個のノード集合 R_2 からの view-change が含まれる。したがって、補題 B.3 により、少なくとも 1 つの正しいノード $c_r \in R_1 \cap R_2$ が存在し、$\mathcal{V}$ に含まれる合意証明の中に、ノード c_r から受信し、かつ (s, t) に一致するものが含まれる。したがって、ビュー v において正しいノードがシーケンス番号 s で取引 t を実行した場合、$v' > v$ における正しいプライマリ $n_{v'}$ は、v' へのビュー更新の際に new-view$(v', \mathcal{V}, \mathcal{O}, n_{v'})$ を送信し、この $\mathcal{O}$ に pre-prepare$(v', s, t, n_{v'})$ を含む。

他方、正しいバックアップは new-view$(v', \mathcal{V}, \mathcal{O}, n_{v'})$ に有効な新規ビュー証明 $\mathcal{V}$ が含まれ、かつ $\mathcal{O}$ が $\mathcal{V}$ から正しく構成されている時に限り、v' に移行し、$\mathcal{O}$ に含まれる pre-prepare に応答して prepare を送信する。これにより、v' においては、$\mathcal{O}$ に含まれる全てのシーケンス番号 s について、バックアップは $\mathcal{O}$ に含まれる (v', s, t) の形の合意証明しか取得できない。

以上より、正しいノードが v でシーケンス番号 s で取引 t を実行した場合、どのノードも $(v', s, t' \neq t)$ の組み合わせを持つ合意証明を取得できないことが証明された。数学的帰納法により、以降のビュー $v' \geq v$ においても、正しいノードはシーケンス番号 s で t' を実行しない。 □

C 暗号学的ハッシュ関数

暗号学的ハッシュ関数は、長いビット列を固定長のビット列に変換する関数で、データベースの高速検索技術などに用いられる関数である。暗号理論では、さらに**衝突困難性**と呼ばれる性質を付加したハッシュ関数が用いられる。

定義 C.1 (暗号学的ハッシュ関数)**.** ハッシュ関数 $\mathrm{H} : \{0,1\}^* \to \{0,1\}^\kappa$ について、$\mathrm{H}(x) = \mathrm{H}(y)$ を満たす $x \neq y \in \{0,1\}^*$ を見つけることが計算量的に困難なとき、H は衝突困難 (collision resistant) であるという。また、衝突困難なハッシュ関数を暗号学的ハッシュ関数という。

暗号学的ハッシュ関数の別の定義に、一方向性ハッシュ関数が用いられることもある。一方向性ハッシュ関数は、y が与えられたときに、$y = \mathrm{H}(x)$ を満たす入力 x を求めるのが計算量的に困難なハッシュ関数と定義される。衝突困難なハッシュ関数は一方向性ハッシュ関数であるが、一方向性ハッシュ関数であるからと言って必ずしも衝突困難とはならないことには注意が必要である。この意味で、衝突困難なハッシュ関数は暗号理論的により強いハッシュ関数である。暗号資産では、SHA-256(**standard4federal**), SHA-3(Keccak)(**Fips202**), RIPEMD-160(Dobbertin, Bosselaers, and Preneel [1996]) などが使われている。

D 共通プレフィックス定理

定義から $Y(S) \leq X(S)$ なので、$|S|$ を大きくすれば、$Y(S) < X(S)$ が成り立つ確率は指数関数的に 1 に近づく。したがって、$Z(S) < Y(S)$ が成り立つ条件を考えれば良い。正しい

採掘者があるラウンドでマイニングに成功する確率を f、典型実行の定義にある揺らぎ率を $\epsilon \in [0,1]$ として、正しい採掘者の優位度を表すパラメータ $\delta \in [0,1]$ との関係を考える。

$$E[Y_i] = \binom{q(n-t)}{1} p(1-p)^{q(n-t)-1} > f(1-f)$$
$$E[Z_i] = 1-(1-p)^{qt} < pqt = \frac{t}{n-t} pq(n-t) < \frac{t}{n-t} \cdot \frac{f}{1-f}$$

典型実行においては、$(1+\epsilon)E[Z(S)] < (1-\epsilon)E[Y(S)]$ が成り立たなければならないが、これは

$$(1+\epsilon)\frac{t}{n-t} \cdot \frac{f}{1-f}|S| < (1+\epsilon)(1-\delta) \cdot \frac{f}{1-f}|S| < (1-\epsilon)f(1-f)|S|$$

を満たせば十分であることが分かる。ここで、$\dfrac{t}{n-t} < (1-\delta)$ を用いた。右の不等式を展開して整理すると、

$$\frac{2\epsilon + (1-\epsilon)2f - f^2(1-\epsilon)}{1+\epsilon} \leq 2\epsilon + 2f < \delta$$

が導かれる。左の等号は $\epsilon = f = 0$ のとき成立。したがって、前述の 3 つのパラメータ f, ϵ, δ が $2\epsilon + 2f < \delta$ を満たせば[45] 、典型実行の条件である $(1+\epsilon)E[Z(S)] < (1-\epsilon)E[Y(S)]$ が成り立つ。

ところで、$Y(S)$, $Z(S)$ の分布は十分大きな $|S|$ に対して正規分布 $\mathcal{N}(\mu, \sigma^2)$ で近似できる。これを用いると $Y(S) < Z(S)$ が生じる確率を $|S|$ の関数として見積もることができる。$\mu_y = f(1-f)$, $\mu_z = (1-\delta)\frac{f}{1-f}$ とおき、$\mathcal{N}(\mu, \sigma^2)$ を平均 μ, 分散 σ^2 の正規分布とすると、それぞれの確率変数は $Y(S) \sim \mathcal{N}(\mu_y|S|, \mu_y(1-\mu_y)|S|)$, $Z(S) \sim \mathcal{N}(\mu_z|S|, \mu_z(1-\mu_z)|S|)$ のように近似できる。ランダムに選んだ区間 S において、$Y(S) < Z(S)$ が生じる確率は、

$$\begin{aligned}
\Pr[Y(S) < Z(S)] &< \int_{-\infty}^{\infty} \Pr[Z(S) = x] \cdot \Pr[Y(S) < Z(S) \mid Z(S) = x]\, dx \\
&= \int_{-\infty}^{\infty} \Pr[Z(S) = x] \cdot \Pr[Y(S) < x]\, dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\mu_z(1-\mu_z)|S|}} \exp\left(\frac{-(x-\mu_z|S|)^2}{2\mu_z(1-\mu_z)|S|}\right) \cdot \frac{1}{2}\left\{1 + \operatorname{erf}\left(\frac{x-\mu_y|S|}{\sqrt{2\mu_y(1-\mu_y)|S|}}\right)\right\} dx
\end{aligned}$$

と書ける。これを解くと、

$$\Pr[Y(S) < Z(S)] < \frac{1}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{(\mu_y - \mu_z)}{\sqrt{\mu_z(1-\mu_z) + \mu_y(1-\mu_y)}}\sqrt{\frac{|S|}{2}}\right)$$

[45] 論文 (Garay, Kiayias, and Leonardos [2015]) では、Honest Majority Assumption に $3\epsilon + 3f < \delta$ を加えることで、すなわち、δ をここで導いた条件の 3/2 倍に大きく取ることで、$(1+\epsilon)Z(S) < (1-\epsilon)Y(S)$ の不等式の分離を大きくし、典型実行の安全性を高めている。

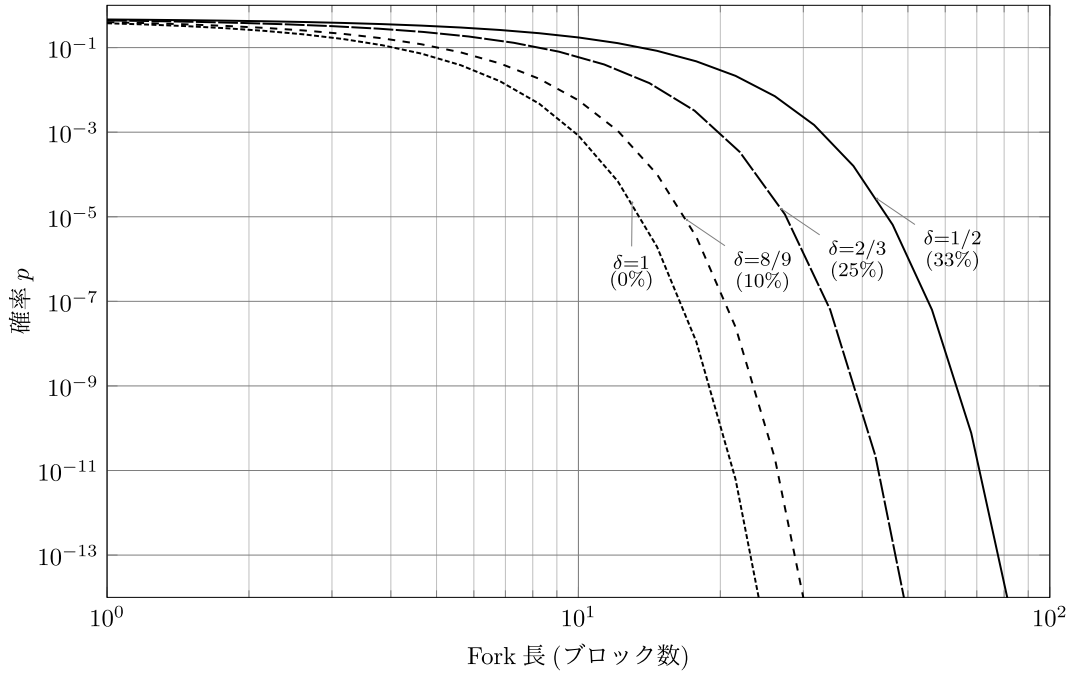


図 19: 長い Fork が生じる確率。不正な採掘者が Selfish Mining 等により長い Fork を起こすことができる確率を、不正な採掘者の割合を $\delta = 1/2(33\%)$、$\delta = 2/3(25\%)$、$\delta = 8/9(10\%)$、$\delta = 1(0\%)$ としてプロットしたもの。横軸は Fork 長、縦軸は確率で表示している。

が得られる[46]。

採掘が 10 分に 1 回の割合で成功するように調整されているビットコインを例に考えると、1 ラウンドは 10 秒、$f = 1/60$ と想定できる[47]。$\mu_y = f(1-f)|S|$, $\mu_z = \frac{f}{1-f}(1-\delta)$ を代入して、上の式をグラフで表示すると、図 19 のようになる。図は不正な採掘者が Selfish Mining 等により長い Fork を意図的に起こすことができる確率を、不正な採掘者の割合を 33%,25%,10%, 0% としてプロットしたものである。横軸は Fork 長、縦軸は確率で表示している。グラフから不正な採掘者が 10% 以下の場合には、取引がブロックに取り込まれてから、数個の後続ブロックが積み上がれば, 当該取引が Selfish Mining 等で覆される確率は 1/100 程度に抑えられることが分かる。更にブロックが積み上がれば、Fork に起因するブロックチェーンの不安定性は急速に小さくなる。

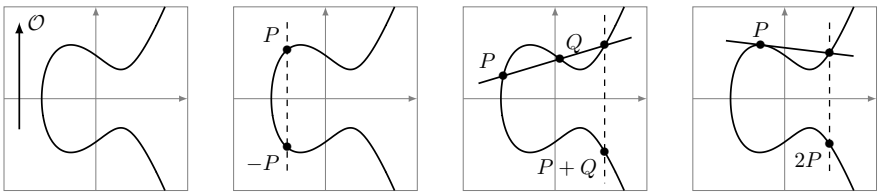
E　楕円曲線暗号

楕円曲線暗号 (Koblitz [1987]) は、公開鍵暗号の一つで暗号化と復号を高速に計算できる方式として知られている。楕円曲線とは、$a, b \in \boldsymbol{K}$(体) に対して、

$$y^2 = x^3 + ax + b$$

[46] $\int_{-\infty}^{\infty} \exp\left(-y^2\right) \mathrm{erf}(b(y-c))dy = -\sqrt{\pi}\,\mathrm{erf}\left(\frac{bc}{\sqrt{1+b^2}}\right)$ を用いた。

[47] 10 分 = 600 より、1 ラウンドあたりの採掘成功確率 $f = 1/60$ と推定した。



零点 (無限遠点) $\mathcal{O}$　　逆元 $-P$　　加法 $P+Q$　　2 倍点 $P+P$

図 20: 楕円曲線上の有理点の加群

で定まる曲線であり、多くの暗号通貨では secp256k1[48] と呼ばれる $a=0,\ b=7$ の曲線が用いられている。楕円曲線の $\boldsymbol{K}$ 上の有理点の集合に無限遠点 $\mathcal{O}=(\infty,\infty)$ を加えた集合を、

$$E(K)=\{(x,y)\in\boldsymbol{K}^2\mid y^2=x^3+ax+b\}\cup\{\mathcal{O}\}$$

と書く。すると、図 20 のように、$\mathcal{O}$ を零元とする加法が定義できる。すなわち、楕円曲線上の 2 点 $P=(x_1,y_1), Q=(x_2,y_2)$ とすると、$P+Q=(x_3,y_3)$ は

$$x_3=\left(\frac{y_2-y_1}{x_2-x_1}\right)^2-x_1-x_2,\quad y_3=\frac{y_2-y_1}{x_2-x_1}\left(x_1-x_3\right)-y_1$$

で計算できる。また、$2P=(x_4,y_4)$ は、

$$x_4=\left(\frac{3x_1^2+a}{2y_1}\right)^2-2x_1,\quad y_4=\frac{3x_1^2+a}{2y_1}\left(x_1-x_4\right)-y_1$$

で計算できる。このように楕円曲線では有理点の幾何学的演算により次々と有理点を求めることができ、曲線上の有理点に関して加群が形成される。

$\boldsymbol{K}=GF(q)$ として、$G,Y\in\boldsymbol{E}(\boldsymbol{K})$ に対して、

$$Y=sG$$

を満たす s を求める問題を楕円曲線上の離散対数問題 (ECDLP[49]) と呼ぶ。ここで、q は素数冪であり、p を素数, r を自然数とするとき $q=p^r$ と書ける。暗号通貨で用いられる secp256k1 では、q として素数の

$$q=2^{256}-2^{32}-2^9-2^8-2^7-2^6-2^4-1$$

が使用されている。すなわち、$p=q, r=1$。

楕円曲線上の離散対数問題は計算量的に困難な問題として知られている。ECDLP を解く最善のアルゴリズム (Galbraith and Gaudry [2016]) は、任意の群に適用可能な Pollard の ρ-法 (Pollard [1978]) であり、位数 n の群に対して $\sqrt{\pi n/2}$ の計算量を要し、鍵長に対して指数時間アルゴリズムとなる。一方、Shor の量子アルゴリズム (Shor [1994]) では $\lceil 9\log_2 n+2\log_2\log_2 n\rceil+10$ 量子ビットが必要 (Roetteler et al. [2017]) とされ、現在の量子コンピュータでは secp256k1 の解読は困難と考えられている。

[48] Standards for Efficient Cryptography (SEC), Certicom Research, `http://www.secg.org/sec2-v2.pdf`.
[49] Elliptic Curve Discrete Logarithm Problem

F　非対話ゼロ知識証明（NIZK）

NIZK 証明システム (Blum, Feldman, and Micali [1988]) は以下のように定義される。確率的多項式時間アルゴリズム (P,V) を証明者と検証者とする。κ をセキュリティパラメータとし、言語 $\mathcal{L} \subseteq NP$ に対して、(P,V) が以下の性質を満たすとき、言語 $\mathcal{L}$ に対する NIZK 証明系と呼ぶ。

a) 完全性: 任意の入力 $x \in \mathcal{L}$ に対して、証拠（witness）w が存在して、任意の多項式 $p(\cdot)$ に対して下式が成立する。

$$\Pr[V(R,x,\pi)=1 \mid \pi \leftarrow P(R,x,w)] \geq 1-\frac{1}{p(|x|)}$$

を満たす。ここで、R は全参加者が参照可能な文字列（common reference string）とする。

b) 健全性: 任意の入力 $x \notin \mathcal{L}$, 任意の確率的多項式時間アルゴリズム P^* , 任意の多項式 $p(\cdot)$ に対して、下式が成立する。

$$\Pr\left[V\left(R,x,\pi^*\right)=1 \mid \pi^* \leftarrow P^*(R,x)\right] < \frac{1}{p(|x|)}$$

c) ゼロ知識性: 任意の入力 $x \in \mathcal{L}$ に対してある証拠 w と 確率的多項式時間シミュレータ S が存在して、下の 2 つの分布は計算量的に識別不能である。

$$\{R,x,\pi | \pi \leftarrow P(R,x,w)\} \approx \{R,x,\pi^*\} \leftarrow S(x)$$