社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

「ブロックチェーン」を学んで，使ってみよう!

Let's learn use blockchain!

藤原　明広†

† 千葉工業大学
E-mail: †akihiro.fujihara@p.chibakoudai.jp

あらまし　2008 年にビットコインが登場して以降，「ブロックチェーン」という言葉が様々な場面で使われるようになった．一方，ブロックチェーンの定義は明確に定まっていないと聞くこともあり，一般的には理解が難しい技術であるような印象も持たれているかもしれない．しかし，その技術の本質は決して難しくなく，（暗号技術の詳細を除けば）難しい数学や物理学の知識は全く不要である．従って，その本質は中高生にも十分理解可能である．本解説記事では，「ブロックチェーン」と呼ばれている技術の核心について，できる限り分かりやすい解説を試みる．またブロックチェーンの語源はビットコインにあるが，ビットコインによって提案された革新的な技術は，インターネット規模の公開参加型合意形成にあることを明らかにする．またブロックチェーン技術の応用の一つに，電子データの永続証明がある．Bloxberg を用いた永続証明の発行と検証を行う方法についても解説する．

キーワード　ブロックチェーン，ビットコイン，公開参加型合意形成，永続証明，Bloxberg

†

†
E-mail: †akihiro.fujihara@p.chibakoudai.jp

1. ブロックチェーンについて(注1)

ブロックチェーンという言葉が誕生した由来をさかのぼると，ビットコイン [1] が発表された暗号学メーリングリスト(注2)にたどり着く．2008 年 11 月頃，ビットコインの発明者である Satoshi Nakamoto と，暗号技術者として活躍し，最初期のビットコイン開発にも関わった Hal Finney が，このメーリングリストで対話していた．このメールの中に “block chain” という言葉を見つけることができる [2]．また 2009 年 1 月にリリースされたビットコインのバージョン 0.1 のソースコード中にも，同じ言葉が登場する(注3)．おそらくこれらが「ブロックチェーン」という言葉の起源であると推察される．

ブロックチェーンとは，図 1 に示すように，ブロックと呼ばれる日時情報（タイムスタンプともいう）が刻印されたデータ構造が時系列順につながったデータベースである．ただし各ブロックは，その一つ前に作成されたブロックの一部 (ブロック

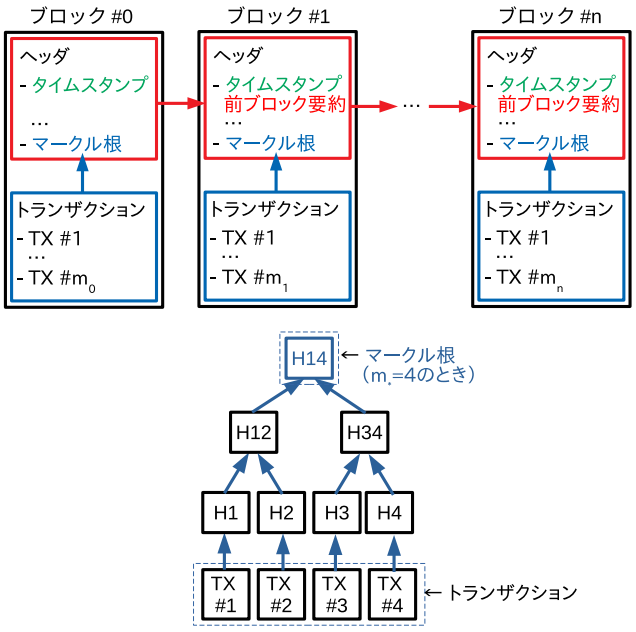

図 1　ブロックチェーンの構造とトランザクションのマークル木．

(注1)：独善的にならないよう，客観的な証拠を交えて説明することを心がけているが，一般的なブロックチェーンの説明と異なる部分があるかもしれない．内容を分かりやすくする為に，敢えてそのようにした．
(注2)：cryptography – The Cryptography and Cryptography Policy Mailing List https://www.metzdowd.com/mailman/listinfo/cryptography
(注3)：バージョン 0.1 の興味深い特徴としては，Windows OS のみに対応している点である．https://github.com/Dan-McG/bitcoin-0.1.0

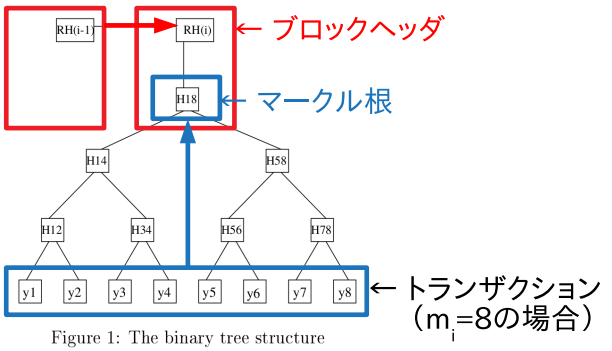

Figure 1: The binary tree structure

図 2　Massias らの論文の Figure 1 (を，本解説記事の図 1 と比較しやすいように筆者が加筆したもの)

ヘッダと言う) の要約(注4)を含む．この要約によってブロック同士が鎖（チェーン）状につながっていることから，ブロックチェーンと呼ばれる．

またブロックの中には，一般にトランザクションと呼ばれるデータが複数，格納されている．このトランザクションを集めた要約(注5)もブロックヘッダに含まれる．全トランザクションはブロックヘッダを介して時系列順につながっているため，ブロックチェーンを確認することで，その前後関係が明確になる．

ブロックチェーンという言葉がビットコインに由来する為，ブロックチェーンはビットコインの登場と共に誕生した新技術と勘違いされることが多い．実はブロックチェーンと同等の技術は，ビットコインの登場以前より存在していた．ビットコイン白書 [1] の中には，Massias らが 1999 年に発表した論文が引用されている [3]．図 2 に示した通り，Massias らの論文の Figure 1 に，本解説記事の図 1 とほぼ同じ内容を確認できる．以上のことから，ブロックチェーンと同等の概念は少なくともビットコインの誕生より約 10 年前から知られていた(注6)．

ブロックチェーンと同等の概念が生まれたとされる 1990 年代前半は，まだインターネットが十分に普及していなかった．従って，当時の環境ではコンピュータやネットワークの性能が十分でなく，実用的な利用は困難だったであろう．この意味では，理論は先行していたが，その存在が忘れられてしまった最近になってやっと，その実用化が可能な環境が整ってきたと言える．このような背景から，ビットコインやそれ以降の仮想通貨や暗号資産，最近では非代替性トークン (Non-Fungible Token, NFT) のブームが起こっていると理解すると，ブロックチェーンが新技術と誤解されている一方で，その応用が次々と模索されている最近の事情を自然に理解することができる(注7)．

ブロックチェーンが新技術ではないことを説明したが，実はビットコインの提案の中には新技術があったと聞くと驚かれるかもしれない．ビットコインが提案した新技術は，仲基合意(注8)(Nakamoto Consensus) と呼ばれる合意形成アルゴリズムである．この合意形成アルゴリズムについて説明する前に，ブロックチェーン技術において本質的に重要な役割を果たしているネットワークの話をしなければならない．ビットコインのシステムは，インターネット上に構築されたピア・ツー・ピア (Peer-to-Peer, P2P) ネットワークに参加している不特定多数のコンピュータ（ノードとも呼ばれる）によって維持されている．ビットコインのノードは世界中に散在している(注9) にもかかわらず，全ノードは同じブロックチェーンを保持している．

沢山のノードが同じものを持ち合うというのは，無駄なのではないかと思うかもしれないが，実はこれがブロックチェーンと呼ばれる技術の本質であり，大きな力となる．例えるなら，空気中の分子がそれぞれ異なる方向にランダムに動いていれば無風だが，分子を一斉に同じ方向に移動させることができれば，一瞬にして空気中に台風を作り出すことができる．これと似た類推で，全てのノードが同じブロックチェーンを持つことができれば，ブロックチェーン上のトランザクションを確認することで，誰がいくらのビットコインを所持しているか，特定のノードに依存することなく，全ノードが一つの共同体のように共通理解を提示できる．これが信用された第三者機関 (Trusted Third Party, TTP) を仮定しない，つまり，どのノードも単体では信じないが，全ノードが同じブロックチェーンを持っているという現実的に非常に稀な状態を信じることで，仮想通貨や暗号資産と呼ばれる貨幣を発行・取引する力となる．

ネットワークに参加しているノードが，合意の下に同じ状態を持ち合う技術は，ビットコイン以前から複数提案されていたが，仲基合意はそれらと本質的に異なる方法を提案しているという意味で新しかった．

2.　仲基合意＝国際要約オリンピック（笑）

ブロックチェーンはブロックヘッダがつながって構成されていることを説明したが，そもそも個々のブロックはどのように生成されるのであろうか？ここで，国際要約オリンピックという仮想的な競技を比喩にして，ブロックの生成方法を説明する．ビットコインでは，ある一定値以下の短いブロックヘッダの要約が作れたノードがブロックを作成できるルールになっており，この競技時間は平均 10 分と短い．文章の要約作業で例えるな

(注4)：ここでの要約とは，厳密にはブロックヘッダを入力にした時のハッシュ関数の出力のことである．ハッシュ関数については紙数の都合で説明できない．国語の授業等で学習した文章の要約と同じく，要約は一般に入力（元の文章）より小さくなり，入力と要約の対応関係は明確である．

(注5)：この要約はトランザクションのマークル根として作成される．図 1 ではトランザクション数 $m = 4$ の，図 2 では $m = 8$ のマークル根をそれぞれ示しているが，紙数の都合で詳細は説明できない．

(注6)：実はブロックチェーンと同等の概念の存在は，1992 年頃にまでさかのぼることができるという話もある [4]．

(注7)：最近の人工知能ブームは深層学習にあるが，その基礎技術の一つとして知られる階層型ニューラルネットワークは，今から 40 年以上前に理論が提案されている [5]．最近になってコンピュータの処理能力の向上やビックデータの収集が可能になった背景から，深層学習で意味のある計算が可能になったことがブームとなった事情と似ている．

(注8)：ビットコインを発明した Satoshi Nakamoto が書いたとされる日本語の記録が一つも見つからないことから，どうやら名前通りの日本人ではなさそうである．従って，仮名であると推察されるが，それが誰なのか，またはチーム名なのか，詳しい事は分かっていない．一説では，Satoshi はポケモンのサトシから，Nakamoto は江戸時代の町人学者として知られる富永仲基 [6] から，それぞれ取られたという話がある [7]．

(注9)：BITNODES `https://bitnodes.io/nodes/live-map/`

ら，国語が苦手な筆者は，その作業の難しさをよく分かっている．本質的な内容を含みつつ，本質が分かりやすいように，短時間で元の文章を (非可逆) 圧縮した要約を作成することは一般にとても大変である．(注10)．従って，筆者のように苦手な者はこの競技から脱落するが，腕に覚えのある文章のプロなら参戦するだろう．

ビットコインでは，短い要約を最初に作成した優秀なノードは，金メダルではなく，賞金として（法定通貨に交換可能な）ビットコインがもらえる．また銀や銅メダルに対応する賞はないため，通常のオリンピックよりも激しい競争となる(注11)．

この競技は平均 10 分おきに随時開催されており，今この瞬間にも世界中のノードが競技者として戦っている．良い要約ができたと判断したノードは，要約が確認できるようにしてブロックヘッダをその他のネットワーク上のノードに送信する．良い要約であれば，他のノードは，その素晴らしさを受け入れて，競争が終了する．競争が終了したと同時に，次のラウンドの競争が始まる．そして，上記のプロセスが永遠と継続していく中で，最も長いブロックチェーンが正統なものとして生き残る(注12)．このようにすることで，良い要約をつづった唯一のブロックチェーンをネットワーク上の全ノードが保持するようになる．

つまり，仲基合意とは「国際要約オリンピック」で熾烈な競争(注13)を行うことで，一つの正統なブロックチェーンを全ノード間で合意して保持し合う仕組みである．またブロックチェーン内のデータは暗号化されることなく，全て包み隠さず公開されている為，誰でも不正（他の多数のノードと異なるブロックチェーンを保持していること）を直ちに発見することができる．したがって万が一，不正を働いたノードがあっても，それを排除することで，システムが要求する誠実な方向に修正できる．

3. 既存の分散システムの合意形成アルゴリズムと仲基合意の違い

合意形成アルゴリズム自体はビットコインが誕生する以前から，分散システムの研究領域において様々なものが提案されてきた．既存の合意形成アルゴリズムの典型例として，Practical Byzantine Fault Tolerant (PBFT) [8] を取り上げて，このアルゴリズムが仲基合意と比較して根本的に異なる特徴をまとめた

(注10)：実際にはブロックヘッダの中のナンス (Nonce, Number used at ONCE) を調整して，目標値以下のハッシュ値を最も早く計算できたノードが，ブロックを生成できる．この作業のことを作業証明 (Proof of Work) という．

(注11)：オリンピックは開催時に体力のある若い競技者のみが対象となるが，お金が関わると世界中の老若男女が競技者となりうるという意味もある．

(注12)：「長いものには巻かれろ」という諺があるが，仲基合意では正にこの原理が採用されている．つまり，強者に従って妥協することを「合意」としている．実際には合意というよりは，「泣き寝入り」と言う方が適当かもしれない．

(注13)：ビットコイン白書 [1] の中に “once-CPU-one-vote” という文言が存在する．しかし，CPU はマシン毎にスペックが異なるので，それぞれの CPU で要約を作成する速度は異なるはずである．また仲基合意では最も長いブロックチェーンが正統なものになることから，その本質はオリンピックのような世界中の強者間での競争にある．従って，各ノードは平等な「投票」を行っているのではなく，その本質は格差のあるノード間での「競争」である．この点について Nakamoto 自身も提案内容の本質を誤解していた可能性がある．

表 1　PBFT と仲基合意の比較

合意形成アルゴリズム	PBFT	仲基合意
ノード数 N	少数 ($N < 1000$)	多数も可
ノードの計算性能	性能格差無し (平等)	性能格差有り (不平等)
ノード間の接続	特定 (既知ノードのみ) 密な結合	不特定 (未知ノード有) 疎な結合
合意形成の方法	投票による多数決	競争によるリーダ (1 位) の選出と その他多数の追従
合意形成に要する通信回数	ノード数が多数で 非実用的 ($O(N^2)$)	ノード数が多数でも 実用的 ($O(N)$)

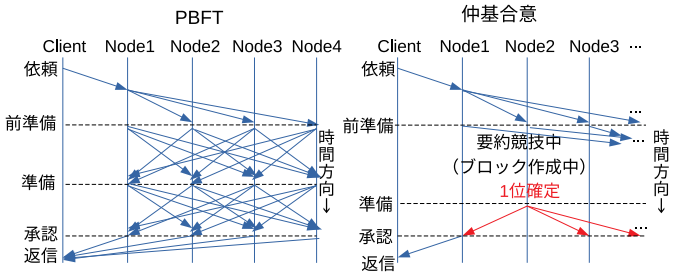

図 3　PBFT と仲基合意の時間-空間ダイアグラム

ものを表 1 に示す．また合意形成を行う上での通信プロトコルを説明した時間-空間ダイアグラムを図 3 に示す．全ノードに承認してもらいたいトランザクションをクライアント (Client) が送信する．その後，ノード間で前準備，準備，承認の過程を経て，最終的に承認されたことをクライアントに返信する．また図 3 の中の矢印はクライアントやノードの間での通信パターンを表す．これらの図表から，仲基合意と PBFT は全く異なることが分かる．また仲基合意はノード数が多数でも実用的である為，インターネット規模の多数のノードでも合意形成が可能である．

4. 公開参加型合意形成

既存の合意形成アルゴリズムも仲基合意も，結局は，ネットワークを介して独立したノード群からトランザクションの承認を得る点では全く同じであり，この過程こそが重要である．また，よく考えてみると，この過程の中にブロックをチェーンでつないだデータベース，つまりブロックチェーンは必須ではない．

また「ブロックチェーン」や「分散台帳技術」のように，データベースに視点を置いた名称が，ブロックチェーン関連技術の総称として使われることが多い．しかし，これまで述べた通り，技術の本質は合意形成の方にある為，これらの名称が関連技術の正しい理解を妨げている可能性がある．トランザクションを広く公開することで全ノードに検証してもらい，問題がなければ承認される．一般に秘密情報は公開できない為，トランザクションを暗号化してブロックチェーン上に公開したとしても，全ノードがその情報を検証して承認したとは判断できない．以上の視点より，「ブロックチェーン」と呼ばれているものが目

指す内容を説明する言葉として，例えば，公開参加型合意形成 (Open-Participation-Style Consensus) といった名称の方が適切なように思われる(注14)．

5. ブロックチェーンの応用と課題

ブロックチェーンを使ってビットコインを作成できることから，まず思い浮かぶ応用としては電子貨幣 (Electronic Cash) があるだろう．お金は紙幣やコインといった現金のように物理空間に実態を持つ形態があるが，実はこの実態は必須ではなく，完全に仮想世界（サイバー空間）に属するものであることは，電子マネーが普及する現在では理解できる方も多いだろう．お金の本質は，お金を持っている人が，その価値に見合った物やサービスを受け取ることができる一種の証明であるとも言える．この証明を，周囲の大多数の人達と信じ合える（つまり上記の内容について合意が取れる）状態自体がお金であると言える．従って「お金」とは，信者同士で信じ合っている「宗教」と同じと言える(注15)．合意形成アルゴリズムがブロックチェーンの本質であることから，ネットワーク上のクライアントやノードの数が増えて，正統なブロックチェーンの信者が増えるほど，便利なお金になっていく．

またお金は一度使用したものを二重に使用することができない．これはお金を所持していることも使用したこともブロックチェーン上の記録として残り，これを過去にさかのぼって書き換えることができなければ実現できる．ビットコインでは，世界中のノードが競争してブロックを作成していることと，ノードが同じブロックチェーンを持っていることの二つの仕組みによって，過去のトランザクションを改ざんすることが事実上困難になっている．

以上より，お金は過去にさかのぼって帳消しにすることができない永続的に残る証明とも言える．このような証明のことを永続証明 (Persistent Proof) という．永続証明はブロックチェーン技術のキラーアプリである．仮想通貨や暗号資産といった代替可能な貨幣のみでなく，NFT も永続証明である．またスマートコントラクトと呼ばれる，ブロックチェーン・システム上で実行可能なプログラムは，プログラムとその状態や遷移の存在を証明する永続証明である．

ビットコインのような TTP を排除した電子貨幣システムにおいては，取引手数料を従来より少額にすることが可能になってくる．これにより，世界中の人達と 1 円未満の少額のお金を送り合うことが可能となり，新しい経済システムを作る潜在能力を持った技術とも言える．この応用のことを超少額決済 (Micropayment) という．しかし，現状の技術では超少額決済は事実上困難になっている．その理由は，ネットワークの通信速度の制約やブロックのサイズに上限が存在する為，取引処理能力に限界が生じてしまうことがある．この為，トランザクション承認の需要に対して，供給が追いついていない．従って，ビットコインの利用者が増えるほど，トランザクション承認手数料が値上がりし，超少額決済を行う動機がない状態に追いやられている．

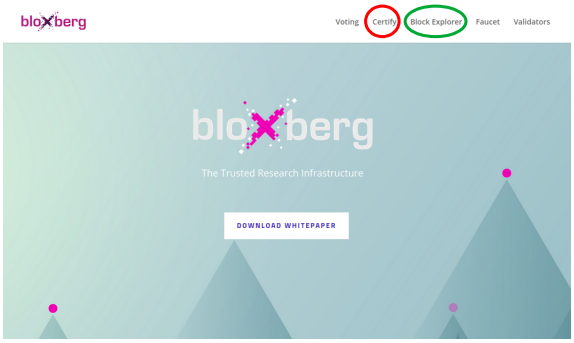

図 4　Bloxberg のウェブページ (`https://bloxberg.org`)

Bitcoin Core の取引処理能力は，1 秒間に 7 トランザクション承認程度が上限であることが知られているが，クレジットカードの VISA は最大で 1 秒間に 56,000 取引承認が可能と言われている．普段の買い物などで実用されているクレジットカードの性能に到底追いついていない．この技術課題のことをビットコインのスケーラビリティ問題という．この問題の根本原因は，仲基合意が競争原理に基づいているからである．グーグルの検索エンジンは驚くほど早く検索結果を返してくる．これはグーグル社内にある沢山のコンピュータが世界中から来る検索クエリを分担し，協力し合って作業しているからである．このような「協力」がない限り，スケーラビリティ問題の改善はない．スケーラビリティ問題はブロックチェーン技術の主要な技術課題であり，沢山の研究者が，その解決に向けて研究している [9]．ちなみに筆者の研究室では大学院生と共同で，競争と協力をバランスさせたブロックチェーン・システムを提案し，研究開発を進めている [10]．

6. ブロックチェーンを使ってみよう

本節では Bloxberg [11] という，ブロックチェーンを用いて信頼できる研究基盤を構築するプロジェクトを紹介し，電子ファイルの永続証明を発行・検証する方法について解説する．このプロジェクトは，ドイツの学術団体として知られる Max Planck Digital Library (MPDL) が発足し，世界 30 以上の大学等の学術機関が参加しているプロジェクトである．各参加機関が 1 台ずつノードを管理することで運営されている．合意形成アルゴリズムは，仲基合意とは異なり，権威証明 (Proof of Authority, PoA) [12] を採用しており，ノードがブロックを交代で承認していく仕組みになっている．Bloxberg のウェブページを図 4 に示す．

このブロックチェーン基盤は，主に科学技術に関する研究データや資料等の証明を国際的に共有する目的で運営されており，誰でも使用できる．この証明は，元ファイルの要約，タイムスタンプ，文字情報が含まれており，これらの要約が Bloxberg

(注14)：「ブロックチェーン」という言葉を関連技術の総称として呼ぶのは止めた方が，「ブロックチェーン」の定義がしやすくなるのではないだろうか．
(注15)：お金も宗教も信じる者は救われるし，それらの知識がないと人生をうまく生きることができないという共通点がある．また，それらの知識を利用して（本来の目的から逸脱して）強者が弱者を侵略する論理として利用されてきた過去の歴史においても同じか．

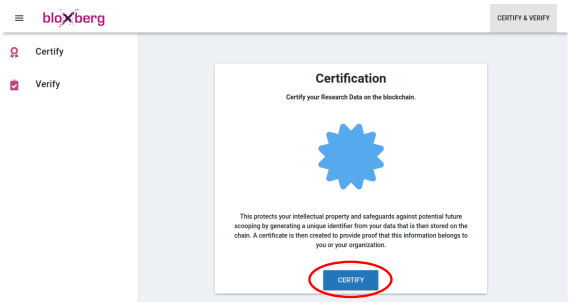

図 5　永続証明発行サイトの画面

のブロックチェーンに記録される．このサービスを利用することで，科学技術研究における成果やデータの盗用を防ぐことができ，その先取権を証明できる．

以降では，本解説記事の原稿[注16]の永続証明を発行・検証する方法を説明する．原稿は PDF ファイルだが，テキストファイルや画像ファイルなど，基本的にはどのような電子データでも永続証明を発行可能である．また発行した永続証明を検証する方法も説明する．Ubuntu GNU/Linux と Python 言語の初歩的な知識を前提とする[注17]．

6.1　永続証明の発行

まず上記の原稿ファイル（ファイル名は 20211201IEICE-Fujihara.pdf）の要約を計算する．端末を起動し，以下のコマンドを実行する．すると 16 進数列の要約[注18]が得られる．

```
$ sha256sum 20211201IEICE-Fujihara.pdf
df5bd89f1c136606b4ee3b6263f3b689149f3b75c5d4dfb
f1fe4ce435f254545  20211201IEICE-Fujihara.pdf
```

この要約を含む証明書 (Certificate) を作成する．まず図 4 の中の赤丸で囲まれている “Certify” タブをクリックする．すると図 5 のようなページが現れる．次に図 5 の赤丸で囲まれている “CERTIFY” ボタンをクリックする．次に “How would you like to generate the hash?” と書かれたポップアップ窓が現れるので，“Generate from File(s)” を選択し，永続証明を作成したい電子ファイルを選択する．ファイルを指定したら右下の “NEXT” ボタンを押す．

次に図 6 の赤い四角で囲まれた四つのテキストボックス (Author or Group Name, Bloxberg Address, Title or Brief Description of Research, Email Address) に，必要に応じて文字情報を入力する．全てのテキストボックスが空でも証明は作成できる．入力後に右下の “NEXT” ボタンを押す．すると図 7 のような画面になるので，中央の “CERTIFY ON THE BLOCKCHAIN” ボタンを押す．少し待つと，ボタンの下に “Transaction Confirmed! Select Finish to create your certificate.” と表示されるので，右下の “FINISH” ボタンを押す．すると，BloxbergDataCertificates.zip という名前の圧縮ファイルがダウンロードできる．これを解凍して中の PDF ファイルを開くと，図 8 のような証明書になっている．また，図 8 の赤枠内に四つの情報（Cryptographic Identifier, Transaction ID, Timestamp, Merkle Root）が記録されている．Cryptographic Identifier は，元データの要約に一致する．Transaction ID は証明書の内容を記録したトランザクションの ID 番号であり，後に検証する時に使用する．Timestamp は証明書が発行された日時（協定世界時）を表す．Merkle Root は証明書の要約で，ブロックチェーンに取り込まれたトランザクションにも tokenHash として記録されている．

図 6　文字情報の入力画面

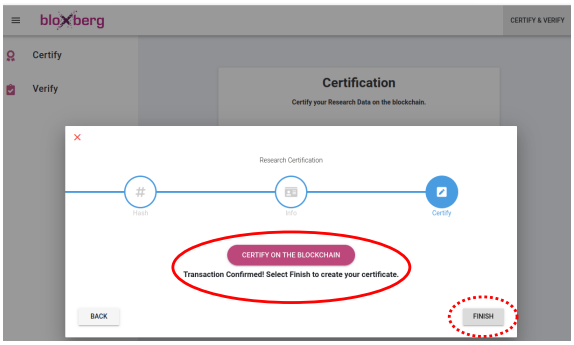

図 7　証明書の発行

図 8　PDF ファイルの証明書の中身

6.2　永続証明の検証

端末を起動し，apt コマンドで poppler-utils と jq をインストールする．また pip コマンドで pyld もインストールする．

(注16)：https://github.com/fujihalab/IEICE_BC_Article を参照せよ．
(注17)：紙数の制限より，細かい環境設定は説明できない．
(注18)：今回の場合，“df5bd89f1c136606b4ee3b6263f3b689149f3b75c5d4dfbf1fe4ce435f254545” となる．

```
$ sudo apt install poppler-utils jq
$ sudo pip3 install pyld
```

次に pdfdetach コマンドを使って，証明書[注19]に埋め込まれた JSON 形式のテキストファイルを抽出する．

```
$ pdfdetach -saveall 90706396-52a6-11ec-9203-024
2ac140011.pdf
```

上記のコマンドを実行すると，bloxbergJSONCertificate という名前のファイルが出力される．jq コマンドを使って内容を確認すると以下のようになる．

```
$ cat bloxbergJSONCertificate|jq
{
  "id": "https://bloxberg.org",
  (中略)
  "crid": "df5bd89f1c136606b4ee3b6263f3b68914
  9f3b75c5d4dfbf1fe4ce435f254545",
  "cridType": "sha2-256",
  "metadataJson": "{\"authorName\": \"Akihiro
  Fujihara\", \"researchTitle\": \"Let's learn
  use blockchain!\", \"emailAddress\": \"akihi
  ro.fujihara@p.chibakoudai.jp\"}",
  "proof": {
    "type": "MerkleProof2019",
    "created": "2021-12-01T12:59:45.381102",
  (以下略)
```

“crid” の値は要約になっている．“metadataJson” の値は，証明作成時に入力した文字情報になっている．“proof” の値の中にある，“created” の値が証明書の発行日時になっている．

この JSON ファイルを用いて，Python プログラム (check_merkle_root.py) を実行する．

```
$ python3 check_merkle_root.py bloxbergJSONCert
ificate
Merkle root: bd338896e74a500a00e86486b97720a2ea
02607f81c1b01286ead1e84625a2b1
```

得られたマークル根は証明内のものと一致する．

(注19)：今回の場合は，90706396-52a6-11ec-9203-0242ac140011.pdf というファイル名であったが，一般に証明毎に異なる．

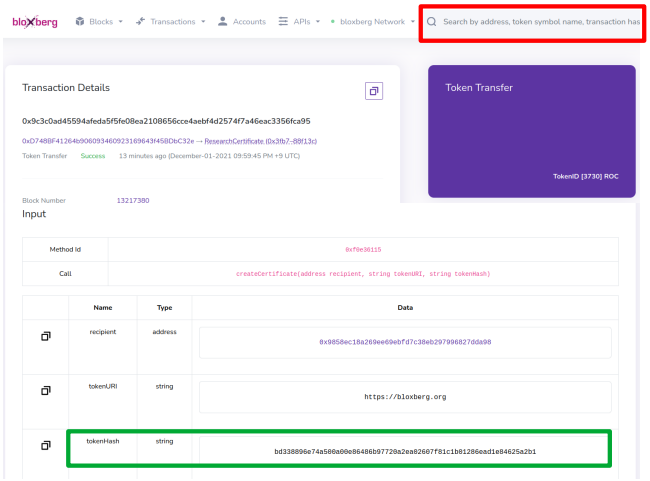

図 9　Block Explorer 上でトランザクションを確認した時の画面

```
1  # check_merkle_root.py のプログラム
2  import sys, json, hashlib, pyld
3
4  # read certificate
5  if len(sys.argv) != 2:
6    print('usage: $ python3 check_merkle_root
7          .py <certificate (json file)>')
8    sys.exit(0)
9  infilename = sys.argv[1]
10 with open(infilename, 'r') as json_file:
11   json_proof = json.load(json_file)
12
13 # canonicalize the json
14 json_proof.pop('proof')
15 normalized_proof = pyld.jsonld.normalize(
16   json_proof, {'algorithm': 'URDNA2015',
17   'format': 'application/nquads'})
18
19 # calculate tokenHash in the transaction
20 merkle_root = hashlib.sha256(
21   str.encode(normalized_proof)).hexdigest()
22 print('Merkle root:', merkle_root)
```

また図 4 の緑丸に囲まれた “Block Explorer” タブをクリックすると，ブロックチェーンに取り込まれたトランザクションを検索できるサイトに移動する．図 9 の右上の赤枠の部分に，証明書に記載されている Transaction ID を入力して Enter キーを押すと，対応するトランザクションの情報が参照できる．図 9 の緑枠の部分に tokenHash の項目があるが，これが上記の Python プログラムで計算したマークル根と一致する．ブロックチェーンは改ざんが困難である為，この証明は永続証明となる．以上より，元ファイルと証明書とブロックチェーンの間の対応関係が検証できた．

7. ま と め

本記事では，ブロックチェーン技術の核心が公開参加型合意形成にあることを解説した．またブロックチェーン技術の主な応用としてデータの永続証明があることも解説した．Bloxbergを使って永続証明の発行と検証を行う方法も解説した．

この記事を機会に，ブロックチェーンに興味を持っていただけたら幸甚である．

文　　献

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System” (White paper), 2008 (2021 年 2 月 9 日閲覧確認).

[2] 最初に “block chain” という言葉が使われた記録: https://www.metzdowd.com/pipermail/cryptography/2008-November/014827.html https://satoshi.nakamotoinstitute.org/emails/cryptography/6/#selection-37.36-37.47

[3] H. Massias, X. S. Avila, and J.-J. Quisquater, “Design of a secure timestamping service with minimal trust requirement,” 20th Symposium on Information Theory in Benelux (1999)

[4] 岩下直行, 「暗号資産の現在と将来」情報処理 Vol. 62, No. 11, 特別解説 (2021).

[5] 麻生英樹, 「多層ニューラルネットワークによる深層表現の学習」人工知能学会誌 28 巻 4 号 (2013).

[6] 釈徹宗, 「天才 富永仲基 独創の町人学者」新潮社 (2020).

[7] C. Wright, “Satoshi’s Vision: The Art of Bitcoin” Howson Books (2019).

[8] M. Castro and B. Liskov, “Practical Byzantine Fault Telerance,” The Proceedings of the Third Symposium on Operating Systems Design and Implementation (1999). https://pmg.csail.mit.edu/papers/osdi99.pdf

[9] Q. Zhou, *et al.*, “Solutions to Scalability of Blockchain: A Survey,” IEEE Access, Vol. 8, pp. 16440-16455 (2020).

[10] T. Yanagihara and A. Fujihara, “Cross-Referencing Method for Scalable Public Blockchain,” Internet of Things, Vol. 15, 100419 (2021).

[11] Bloxberg Whitepaper, The Trusted Research Infrastructure, Whitepaper 1.0 (2019). https://bloxberg.org/wp-content/uploads/2019/07/bloxberg_whitepaper.pdf

[12] S. D. Angelis, *et al.*, PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain, The Proceedings of the Second Italian Conference on Cyber Security (ITASEC 2018). http://ceur-ws.org/Vol-2058/paper-06.pdf