ブロックチェーン概論

株式会社 Neo Breakthrough
Blockchain Lab Director
山科 優希
令和 5 年 3 月 23 日

0 はじめに

本教材は 2023 年 3 月 23 日に東北大学データ駆動科学・AI 教育知能センターが主催する、『ブロックチェーン初学者セミナー VOL.1 ハンズオン付き勉強会』にて使用するための教材である。本教材とセミナーでの発表資料を併用することにより、ブロックチェーン技術の体系的な学習を行うことができる。

本教材の読者に期待することは以下の通り。

・ブロックチェーン技術を用いたアプリケーション開発及び周辺技術開発において必要とされる基礎的な知識の習得
・暗号技術の歴史的背景を踏まえた体系的なブロックチェーン技術の学習
・Bitcoin 及び Ethereum の仕組み及び有用性の理解

以上が具体的な読者に期待することであるが、これに限らない。

2023 年現在、ブロックチェーン技術及びその周辺技術と、それらによって構築されるアプリケーションの集合体を総称する『web3』という概念への可能性が謳われている。『web3』に対する認識は多種多様であり、上記に示したものが定義ではなくあくまでも著者個人の認識に基づく説明である。『web3』が注目されると共に東北大学の所在する宮城県仙台市が web3 特区として活動することが決まり、本地域での『web3』に対する熱は 2022 年中旬から格段に伸びていると実感している。web3 特区にて関連事業を行う事業者の創出、及び東北大学の学生が立ち上げるテックスタートアップの一つの選択肢としてブロックチェーン技術が注目されることを望んでいる。同様に、本教材を通じてブロックチェーン技術に興味を持ち、研究対象の候補として捉えていただくことも望んでいる。

＊本教材に関する質問等は下記メールアドレスまで。

mail : yuki@neobreakthrough.com

1 Bitcoin Prehistory

本章では Bitcoin 以前の暗号技術についてまとめ、解説していく。Bitcoin の誕生がブロックチェーンの誕生と言えるが、その構成技術（デジタル署名・Proof-of-Work・Merkle Tree 等）は Bitcoin 以前から誕生している。暗号技術の歴史的背景を学習することで、ブロックチェーン技術の本質的な理解を促進する。そのため本章では Bitcoin 以前の暗号技術を体系的に学び、2 章以降の Bitcoin/Ethereum の理解に繋げる。

Diffie-Hellman 鍵共有 -Diffie and Martin Hellman（1976）

1976 年の論文”New Directions in Cryptography”[1] 内で提唱された『公開鍵暗号の駆け出し』となる技術。共通鍵暗号方式による鍵配送をセキュアに行うための理論。2 人が示したのはあくまでも原理のみであり、具体的関数の存在については示されていない。

RSA 暗号 -Ron Rivest, Adi Shamir, and Leonard Adleman（1977）

Diffie-Hellman の示した原理を満足した具体的関数の存在を証明。存在を示した Rivest/Shamir/Adleman の 3 名の名前に由来して RSA 暗号となった。素因数分解の困難性によって暗号の安全性が担保されている。

Merkle Trees -Ralph Merkle（1979）

Ralph Merkle によって発明された、大きなデータを要約して格納するツリー構造の一種。ハッシュ*チェーンのうち、二分木によるツリー構造を組成するものを示す。ハッシュを利用していることから、ハッシュ木と呼ばれることもある。

ハッシュ*
ハッシュ値。ハッシュ関数（任意のデータに対して固定長のデータを返す関数）を用いて得られた値のこと。

Blind Signatures for Untraceable Payments -David Chaum（1983）

デジタル署名の一種。『署名する人』と『署名するメッセージの作成者』が異なる場合に用いられる。応用具体例として、メッセージ送信者のプライバシーが必要な匿名システムなど。

Elliptic Curve Cryptography -Neal Koblitz and Victor S. Miller（1985）

楕円曲線を利用した暗号方式の総称。楕円曲線上の離散対数問題(EC-DLP)*の困難性が暗号の安全性を担保している。RSA 暗号よりも短い鍵で処理速度も速い状態で同レベルの安全性を保つことができる。一部の EC-DLP に対しては多項式時間アルゴリズムが見つかっている。

離散対数問題*
素数 p と定数 a が与えられ、a^x≡y(mod p)を満たしている。この y についての方程式の解 x を求める問題のこと。一般的に p が十分大きい値を取るとき、x から y を求めることは用意（a^x を p で割った時の余りを求めれば良い）であるが、y から x を計算することが非常に困難であり計算に膨大な時間がかかってしまう。

FLP Impossibility -Fischer, Lynch, and Paterson（1985）

1985 年 4 月に Fischer/Lynch/Patterson によって発表された論文”Imposebility of Distributed Consensus with One Faulty Process”において提唱。可用性のある非同期分散システムにおいて、有限時間内で分散合意を達成できるアルゴリズムが存在しない。

Time-stamp　How to Time-stamp a digital document”（1991）

Haber と Stornetta によって提唱された。ドキュメントをハッシュ化し、リンキングさせることによってドキュメントの存在と前後関係を示した。信頼できるタイムスタンプサービスが存在することで成立。

”The Crypto Anarchist Manifesto” -Timothy C.May（1992）

Timothy C May による暗号無政府主義宣言。88 年に提唱され、92 年にサイファーパンク運動設立総会にて読み上げられた。完全匿名・完全自由言論・完全自由市場を確固たるものとし、暗号無政府主義の基盤を固めた。

”Formalizing and Securing Relationships on Public Networks” -Nick Szabo（1997）

法学者であり暗号学者でもある Nick Szabo によって論文が発表され、『スマートコントラクト』という言葉を初めて提唱した。Nick Szabo は自動的であり暗号学的に安全なデジタル市場を構想し、スマートコントラクトを取引やビジネス機能が仲介者なしに信頼できる形で実行できるものとしていた。

Hashcash -Adam Back（1997）

Hashcash は 1997 年に Adam Back によって提唱された Proof-of-Work のシステム。電子メールのスパムやサービス拒否攻撃を制限するために使用される。送信のためにコストとして CPU を消費（ハッシュ値生成）することによって、コストを指標に送信元が信頼できるかどうかを検証することができる。

BitGold -Nick Szabo（1998）

Nick Szabo が分散型デジタル通貨の仕組みとして設計した。実装されることはなかったが、『Bitcoin アーキテクチャの直接的な先駆け』と呼ばれている。Szabo は『金の安全性と信頼性の特徴』を、サイバースペースで忠実に再現しようとしていた。

B-money -Wei Dai（1998）

Wei Dai が作成した『匿名分散型電子通貨システム』の初期提案。通貨を作る別の方法としてある程度の複雑さがわかっている計算問題の解答を参入者が入札するオークションによる方法が提案されている。

Napster　Modern P2P”（1999）

P2P 技術を用いた（主に音楽を共有することを目的とした）ファイル共有システム。米 Napster Inc.によって運営された。アカウントを中央サーバーで管理し、所有者から直接ダウンロードする方式でファイル共有を P2P ネットワーク上で行う”Hybrid P2P”であった。

Winny -金子 勇（2001）

元東京大学大学院情報理工学化研究助手の金子勇によって開発された、P2P 技術を応用したファイル共有ソフト。自身の接続している PC をファイル転送の中継点として使うため、実際のファイルが保存されている PC がどこであるかを知ることができない。Winny はピュア型 P2P であり、利用開始時に中央サーバーからアクセス権を受け取り、ネットワークに参加する。

BitTorrent -Bram Cohen（2001）

P2P 技術を利用したファイル転送用プロトコルのこと。従来のインターネットに反して『人気のあるソフトであればあるほどダウンロード速度が速くなる』という点が特徴であり、この点が Napster と異なる。

Reusable Proof-of-Work -Hal Finny（2004）

Nick Szabo の理論に基づいて Hal Finny が発明したデジタル通貨の原型とされるもの。RPOW はプロトタイプ以上にはならなかったが、もし普及していたら巨大なネットワークに対応することになっていたと言われている。高度なセキュリティモデル下での構築を条件として、二重支払い問題を解決した。

2　Bitcoin

”A Peer-to-peer Electronic Cash System” -Satoshi Nakamoto（2008）

Bitcoin は Whitepaper*が 2008 年 10 月 31 日に発表され、2009 年 1 月 3 日に始動した。Bitcoin のコミュニティ支援によって設立された Bitcoin.org にて日本語訳も公開され

ている。*勉強会での発表資料にて各章の補完をしている。また詳しく理解したい方には Bitcoin.org の Whitepaper を読むことをお勧めする。

Whitepaper*
Whitepaper はプロジェクトの説明書のことで、Bitcoin では大まかな設計について記載されている。

原文*
https://bitcoin.org/bitcoin.pdf

日本語訳
https://bitcoin.org/files/bitcoin-paper/bitcoin_jp.pdf

Bitcoin Wallet

Bitcoin Wallet は Bitcoin の管理・送受信を行うためのソフトウェア。タイプは以下の 3 種類。

1. デスクトップウォレット
   ユーザーがデスクトップにダウンロードして利用するウォレット。オンラインにアクセスすることがないため、秘密鍵をセキュアに管理することができる。
2. モバイルウォレット
   ユーザーがモバイルアプリをダウンロードして利用するウォレット。実店舗での Bitcoin 決済に便利。
3. ウェブウォレット
   ユーザーがウェブブラウザからアクセスして利用するウォレット。オンラインでアクセスできるためデスクトップに比べて利便性は高いが、秘密鍵が常にオンライン上で管理されているため、セキュリティリスクが高くなる場合がある。

Transaction（UTXO モデル）

Transaction について詳しく知りたい方は発表資料と Whitepaper を参照されることを薦める。本項ではトランザクション処理方式の UTXO（未使用トランザクション）モデルについて解説する。UTXO モデルを理解することで後に説明する Ethereum の Account モデルとの比較が容易になる。

UTXO は Unspent Transaction Output の略称。（TX は”Transaction”を表す。）日本語に訳すと『未使用トランザクションアウトプット』となる。トランザクションの Input は、以前のトランザクションの UTXO を使用して新しいトランザクションの Output を作成する。

UTXO モデルでは一度使用されたトランザクションの Output を再利用することが不可能であり、この特徴によって Bitcoin のセキュリティが担保されている。（二重支払いを防止することができる。）

Blockchain Architecture

発表資料を参照することで、ブロックチェーン技術の構造について概ね理解することができる。ここでは一度ブロックチェーン技術の構造についてまとめる。

”ブロック”はトランザクションを記録するためのデータ構造であり、分散型”台帳”と呼ばれる所以でもある。ブロックには Proof-of-Wotk に使用する”Nonce”やトランザクションを Merkle Tree でまとめた”Root Hash”などが格納されている。これらのデータをさらに”Block Header(Hash)”としてまとめる。

ブロック”チェーン”と呼ばれる所以でもあるが、各ブロックには一つ前のブロックの Block Header(Previous Hash)が記録されており、この値も踏まえて各ブロックの Block Header は作成される。これによって各ブロックのデータが数珠繋ぎのように記録され、不正・改ざんの困難性というブロックチェーンの特徴を実現している。

3　Ethereum

Ethereum Whitepaper

Ethereum Whitepaper は 2014 年に Vitalik Buterin によって発表され、2015 年に実装された。Ethereum は『ブロックチェーン上にチューリング完全なプログラミング言語の実行環境を構築することによってスマートコントラクトを実現する』というブロックチェーンであり、この特性から”World Computer”と表現されることがある。Ethereum について概念的に詳しく知るためには Whitepaper を読むのが良い。github 上に日本語訳も挙げられているため、そちらも参考にすると理解が促進される。また Ethereum には Yellowpaper というものもあり、こちらはより詳細な設計を記したもの。こちらを読むことによってトランザクション構造やスマートコントラクトについてより深い理解を促進する。

Ethereum Whitepaper 原本*
https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf

日本語訳
https://github.com/ethereum/wiki/wiki/%5BJapanese%5D-White-Paper

Yellowpaper

https://ethereum.github.io/yellowpaper/paper.pdf

Smart Contract

スマートコントラクトのアイディアは暗号学者・法学者の Nick Szabo によって提唱され、Ethereum ではスマートコントラクトが実装された。Ethereum のスマートコントラクトは『ブロックチェーン上で実行可能なコードであり、指定した条件が満たされた場合にプログラムが自動執行される』というものである。ブロックチェーン上で契約が可能であることから、TTP（Trusted Third Party）の存在が必要ない上に自身のアイデンティティとは独立して契約の締結が可能になる。

スマートコントラクトを実装した Ethereum というブロックチェーンの誕生により、人々は多様な Application をブロックチェーン上で開発することが可能になった。これらの Application は分散型ネットワーク上（Ethereum 上）で動いていることから、Decentralized Application（分散型アプリケーション）、略称して”DApps”と呼ばれることが多い。

Account モデル

Ethereum は Account モデルを採用している。Account モデルでは単一のアカウント（EOA/CA）を使用することによって UTXO モデルと比較して柔軟性を出している。各アカウントの説明については発表資料にまとめてあるため、参照されてほしい。

UTXO モデルではトランザクションが送信されるたびに UTXO が新しく作成され、各トランザクションは以前のトランザクションの Output を参照する必要があった。Account モデルでは単一アカウントの残高を更新するだけなので、直感的な理解が容易なモデル。

DeFi（Decentralized Finance）（分散型金融）

分散型金融はブロックチェーン上で構築された金融システム。スマートコントラクトによって既存金融では TTP として存在している機関なしで取引を実行できる。既存金融と同様に Lending のような仕組みや Derivative の提供も実現される。

これらの仕組みは AMM（Automated Market Maker）によって実現されている。これは直訳すると『自動マーケットメイカー』であり、既存金融でマーケットメイキングを行っている TTP の存在なしでスマートコントラクトによって自動執行している。AMM は各 DeFi Protocol において独自のアルゴリズムが設けられており、Ethereum を代表する DeFi Protocol の Uniswap は『X*Y=K(K : Const)』の式を基盤としている。

分散型金融は新たな金融領域として注目されており、金融領域に興味があり金融工学を専攻している方にとっては非常に興味深い領域である。研究対象としても面白い分野であるので、興味のある方は Uniswap の仕組みから学び始めてみると良い。

DAO（Decentralized Autonomous Organization）（分散型自律組織）

DAO の和訳は『分散型自律組織』であり、ブロックチェーンを基盤とした分散型の組織構造。組織の運営・決定がスマートコントラクトによって制御されており、組織参加者による意思決定の透明性が向上し、同時に不正・改ざんが難しくなる。DAO は多くの DApps において採用されており、DApps の分散性を主張しているように感じられる。

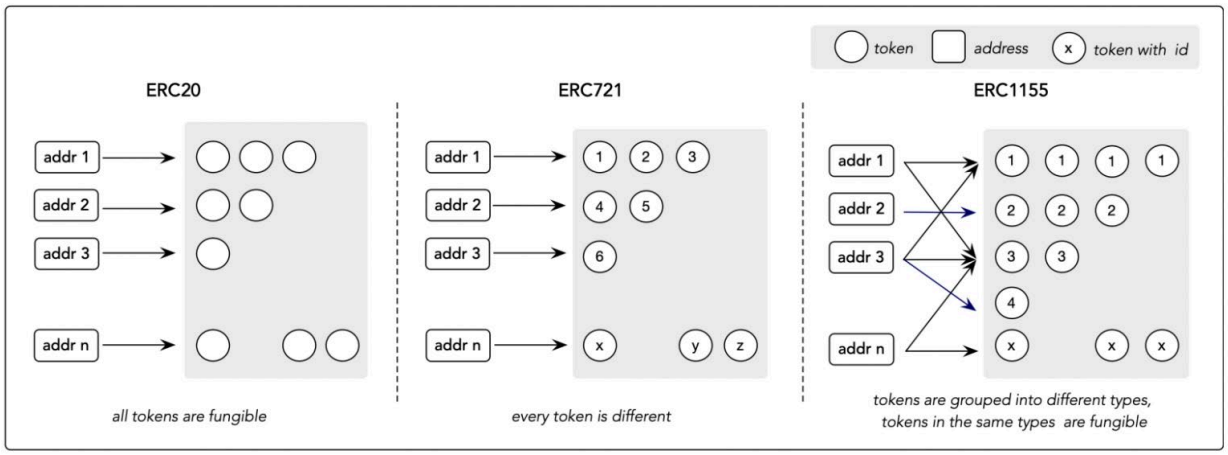

Fig. 2: NFT-related Token Standards

DAO は既存の組織構造の一つである『ティール組織』をブロックチェーン基盤で実現したものであるように考えられ、ブロックチェーンをここまで学んだ方であればティール組織を軽く理解することで DAO の雰囲気を掴むことができると思う。

組織構造や public goods（公共財）に興味のある方は DAO を学んでみると良い。必読の教材として、Vitalik Buterin のブログを記載しておく。

DAOs, DACs, DAs, and More: An Incomplete Terminology Guide
https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide

DAOs are not corporations: where decentralization in autonomous organizations matters
https://vitalik.ca/general/2022/09/20/daos.html

ERC721（Non-Fungible Token）

ERC20/ERC721 については発表資料にて詳しく説明しているのでそちらを参照いただきたい。ここでは ERC20 と ERC721 の Fungibility と Non-Fungibility の両方を兼ね備えている。ERC1155 は異なる種類のトークンを一つのコントラクト内に統合して効率的に管理できるように設計されている。複数のトークンを一つのトランザクションで同時に送信す

ることができ、ガス代を節約することができる。これは tokenID が同じである NFT を複数発行できる感覚であり、以下の図のように理解することができる。

出典：Non-Fungible Token(NFT): Overview, Evaluation, Opportunities and Challenges

4　ハンズオン資料

Metamask のインストール

Metamask のインストール方法については発表資料を参照していただきたい。ここでは Metamask の『秘密鍵』および『ニーモニック』について重要性を伝えておきたい。『ニーモニック』とは 12〜24 個の英単語であり、Metamask の設定から確認することができる。PC で作成したアカウントをモバイルのアプリに同期したい場合、秘密鍵のエクスポートでも、ニーモニックの入力でも同期することができる。『ニーモニック』はエントロピーを利用してランダムに生成されており、非常に安全であり私たち人間が覚えやすい形式である。このランダムに生成されたニーモニックを暗号化することによって秘密鍵が生成される。

ブロックチェーンにおいてアイデンティティとトランザクションが紐づいていないことは Bitcoin Whitepaper の 10 章”Privacy”の図を見れば明らかである。アイデンティティとトランザクションが結びついていないということは、Metamask に表示されている 16 進数のウォレットアドレスが私たちのマイナンバーカードの個人番号や自宅の住所などと結びついていないことを表す。つまり、ブロックチェーン上での私たちのアイデンティティはある意味 EOA であると捉えることができる。EOA の秘密鍵で署名することによって私たちはブロックチェーン上で契約を締結することが可能であるが、秘密鍵を盗まれてしまった場合私たちのブロックチェーン上でのアイデンティティは見知らぬ誰かに奪われてしまう。同様に、署名先の契約が不当なものであった場合でも取り消すことはできない。この点から、ブロックチェーン業界では”Don’t Trust, Verify”（信頼するな、検証しろ）という名言が生まれている。

【番外編 〜秘密鍵からウォレットアドレスを生成する方法】

秘密鍵からウォレットアドレスを生成する方法は以下の流れ。

1. ECDSA(Secp256k1)によって秘密鍵から 64byte の公開鍵を生成
2. 公開鍵をハッシュ関数 Keccka-256 に通して 32byte の文字列を得る
3. 最初の 12byte を削除して 20byte とする
4. prefix の 0x を先頭に追加することでウォレットアドレスを得る

Thirdweb を用いた NFT マーケットプレイス開発

今回のハンズオン勉強会では thirdweb のプレビューを使って簡易的に NFT マーケットプレイスのリスティング機能・購入機能を実装したが、少しコーディングを行うことによって web 上に公開できるレベルのマーケットプレイスも開発できる。意欲のある方はこちらの thirdweb 公式ブログを参照し、構築されてみてほしい。

・How to create an NFT Marketplace with Next.js and third web on Polygon
https://blog.thirdweb.com/guides/how-to-create-an-nft-marketplace-with-nextjs-and-thirdweb-on-polygon-network/

・Create Your Own NFT Marketplace with Typescript and Next.js
https://blog.thirdweb.com/guides/nft-marketplace-with-typescript-next/

・Marketplace With Next.js
https://github.com/thirdweb-example/marketplace