

# ブロックチェーン・暗号通貨の 数理

安永憲司

金沢大学

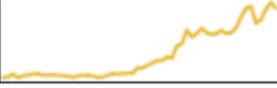
金沢大学暗号理論勉強会  
2017.6.15-16

# 暗号通貨の歴史

- 1980年代：David Chaum の電子現金
  - 銀行発行の現金を電子的に実現
- 2008年：Satoshi Nakamoto の Bitcoin
- 2011-2013年：シルクロード（闇サイト）事件
- 2013年：Bitcoin への注目

# 様々な暗号通貨

## ■ Crypto-Currency Market Capitalizations

#	Name	Market Cap	Price	Circulating Supply	Volume (24h)	% Change (24h)	Price Graph (7d)
1	 Bitcoin	\$44,222,564,865	\$2698.46	16,388,075 BTC	\$2,713,170,000	-9.14%	
2	 Ethereum	\$36,226,580,229	\$391.90	92,439,270 ETH	\$3,134,760,000	10.20%	
3	 Ripple	\$9,799,864,062	\$0.255695	38,326,381,283 XRP *	\$130,502,000	-7.81%	
4	 Ethereum Classic	\$1,873,091,624	\$20.24	92,549,997 ETC	\$301,845,000	-5.91%	
5	 NEM	\$1,840,122,000	\$0.204458	8,999,999,999 XEM *	\$15,461,000	-13.29%	
6	 Litecoin	\$1,522,938,522	\$29.56	51,524,082 LTC	\$365,955,000	-8.47%	
7	 Dash	\$1,317,550,159	\$179.03	7,359,341 DASH	\$85,760,900	-7.72%	
8	 BitShares	\$984,101,445	\$0.379075	2,596,060,000 BTS *	\$294,081,000	2.33%	
9	 Stratis	\$812,481,407	\$8.26	98,422,348 STRAT *	\$16,999,500	-5.78%	
10	 Monero	\$751,743,961	\$51.42	14,618,316 XMR	\$25,592,600	-10.83%	

# ビットコイン (Bitcoin)



- Satoshi Nakamoto (2008) が提案
- 信頼できる第三者を置かずに実現可能な暗号通貨
  - 非中央集権的に実現
- 基礎となる技術はブロックチェーン（公開台帳・分散台帳）などと呼ばれる

# ブロックチェーン・公開台帳・分散台帳

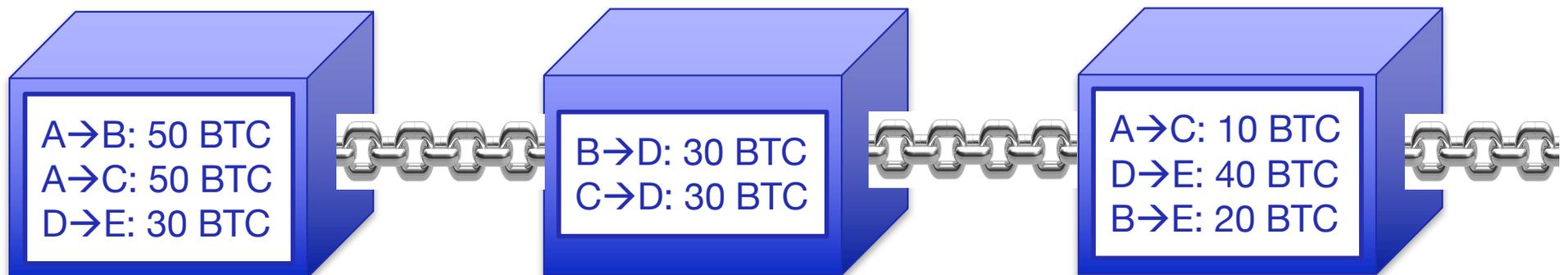
## ■ 非中央集権的に台帳を管理

- 台帳：追記専用のログ。情報に順序があり、記録後は内容・順序の変更不可
- 公開・分散型：誰でも書き込み・読み取り可能
  - 非許可型 (permissionless) と呼ばれることも



# Bitcoin の実現方法

- 公開台帳にすべての取引内容を記載
  - 追記の際に、過去の取引を見て、二重支払い等の不正をチェック
  - 送金者の電子署名が必要なため、送金偽造は不可
    - 電子署名の公開鍵（検証鍵） = Bitcoin 上の ID



# ブロックチェーンの実現方法

- チェーンにブロックを接続するためにパズルを解くことを必要とする
  - Proof-of-Work と呼ばれる
- チェーンを少しずつ伸ばすことにより、全員が同じ台帳を共有できる

# Proof-of-Work (PoW)

- 仕事の証明 [Dwork, Naor 1992]
  - 解くために少し時間の掛かるパズル  
(答えの正当性は簡単に確認できる)
  - Bitcoin では、PoW に成功すれば報酬としてコインを受け取れるため採掘 (mining) とも呼ばれる
    - 実際は、ハッシュ関数を使った探索

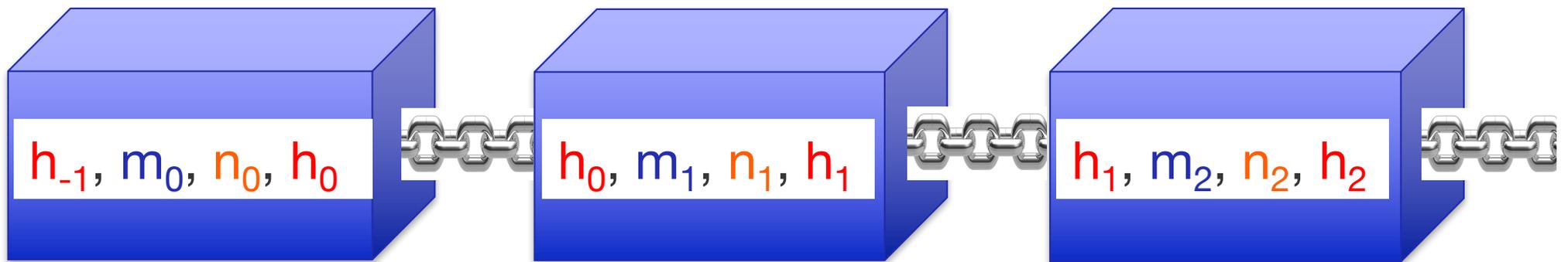
Find  $n$  s.t.  $H(h, m, n) < D$

The diagram shows the equation  $H(h, m, n) < D$  with four callout boxes pointing to the variables:

- $n$ : ナンス (Nonce)
- $h$ : 前ブロックのハッシュ値 (Previous block's hash value)
- $m$ : 記載内容の系列 (Sequence of content)
- $D$ : 困難性パラメータ (Difficulty parameter)

# ナカモトプロトコル [Nakamoto 2008]

## ■ [Pass, Seeman, shelat 2017] によるモデル化

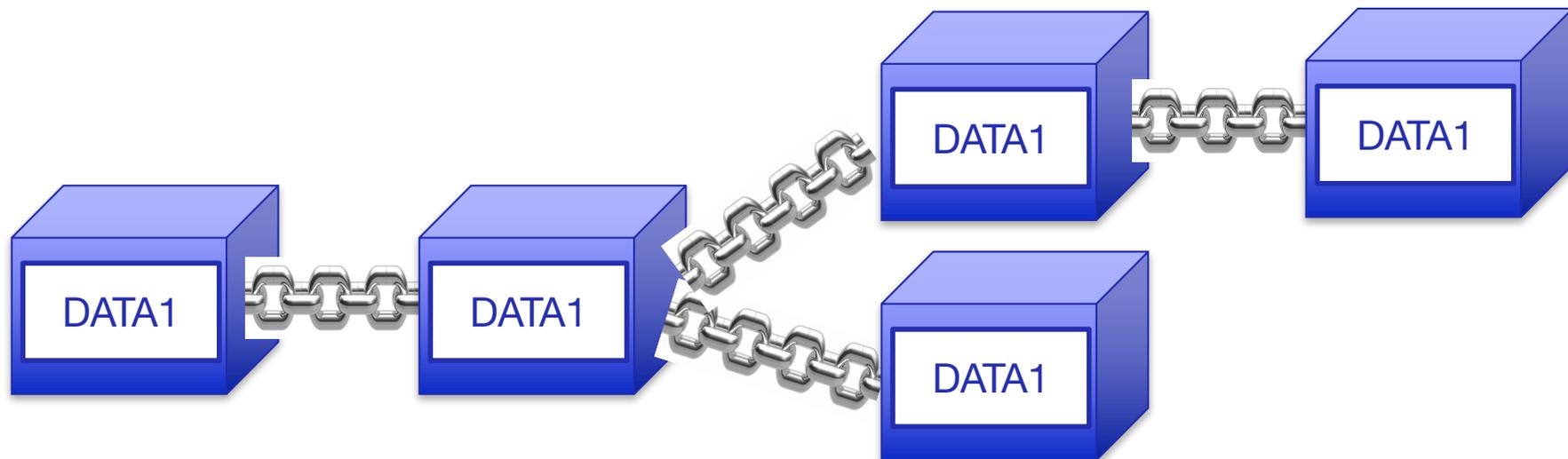


$$\forall i = 1, 2, \dots, h_i = H(h_{i-1}, m_i, n_i) < D$$
$$h_{-1} = H(0, 0, \perp)$$

- H はランダムオラクルとしてモデル化
- $\forall (h, m), \Pr_n[H(h, m, n) < D] = p$

# チェーンの枝分かれ

- 枝分かれは存在しうるが、  
「長いチェーンが正当なもの」というルール
  - 過半数が正しく実行するとき、  
一定時間経てば書き換えはほぼ不可能
- Bitcoin では深さ 6 で確定とみなすことが多い



# ビットコインにおける調整・報酬

- システム全体で PoW が 10 分に 1 回しか成功しないよう困難性パラメータ D を調整
  - 2016 ブロック（約 2 週間）毎に再調整
- PoW 実行誘因として成功者に**ブロック報酬**を付与
  - インフレーション対策として 210000 ブロック（約 4 年）毎に報酬は半減
- 取引をブロックへ取り込む誘因として PoW 成功者に**取引報酬**を付与
  - 取引の当事者から支払われる

# ブロックチェーンの応用

- 「非中央集権的に維持できる台帳」と考えれば  
応用範囲は広い
  - 分散管理のため、安定したシステムが実現
  - 中央組織における情報集約が不要
  - 中央組織を介さずに情報共有可能

# ブロックチェーンの活用例

## ■ ブロックチェーン技術活用のユースケース

<p><b>金融系</b></p> <p>決済 (SETL、FactoryBanking)</p> <p>為替・送金・貯蓄等 (Ripple、Stellar)</p> <p>証券取引 (Overstock、Symbiont、BitShares、Mirror、Hedgy)</p> <p>bitcoin取引 (itbit、Coinffeine)</p> <p>ソーシャルバンキング (ROSCA)</p> <p>移民向け送金 (Toast)</p> <p>新興国向け送金 (Bitpesa)</p> <p>イスラム向け送金/シャリア遵法 (Abra、Blossoms)</p>	<p><b>ポイント/リワード</b></p> <p>ギフトカード交換 (GyftBlock)</p> <p>アーティスト向けリワード (PopChest)</p> <p>プリペイドカード (BuyAnyCoin)</p> <p>リワードトークン (Ribbit Rewards)</p>	<p><b>資産管理</b></p> <p>bitcoinによる資産管理 (Uphold(旧Bitreserve))</p> <p>土地登記等の公証 (Factom)</p>	<p><b>商流管理</b></p> <p>サプライチェーン (Skuchain)</p> <p>トラッキング管理 (Provenance)</p> <p>マーケットプレイス (OpenBazaar)</p> <p>金保管 (Bitgold)</p> <p>ダイヤモンドの所有権 (Everledger)</p> <p>デジタルアセット管理・移転 (Colu)</p>	<p><b>公共</b></p> <p>市政予算の可視化 (Mayors Chain)</p> <p>投票 (Neutral Voting Bloc)</p> <p>バーチャル国家/宇宙開発 (BitNation/Spacechain)</p> <p>ベーシックインカム (GroupCurrency)</p>
	<p><b>資金調達</b></p> <p>アーティストエクイティ取引 (PeerTracks)</p> <p>クラウドファンディング (Swarm)</p>	<p><b>ストレージ</b></p> <p>データの保管 (Stroj、BigchainDB)</p>	<p><b>コンテンツ</b></p> <p>ストリーミング (Streamium)</p> <p>ゲーム (Spells of Genesis、Voxelnauts)</p>	<p><b>医療</b></p> <p>医療情報 (BitHealth)</p>
	<p><b>コミュニケーション</b></p> <p>SNS (Synereo、Reveal)</p> <p>メッセージ、取引 (Getgems、Sendchat)</p>	<p><b>認証</b></p> <p>デジタルID (ShoCard、OneName)</p> <p>アート作品所有権/真贋証明 (Ascribe/VeriSart)</p> <p>薬品の真贋証明 (Block Verify)</p>	<p><b>将来予測</b></p> <p>未来予測、市場予測 (Augur)</p>	<p><b>IoT</b></p> <p>IoT (Adept、Filament)</p> <p>マイニング電球 (BitFury)</p> <p>マイニングチップ (21 Inc、)</p>
		<p><b>シェアリング</b></p> <p>ライドシェアリング (La'ZooZ)</p>		

出典：経済産業省 商務情報政策局 情報経済課「平成27年度 我が国経済社会の情報化・サービス化に係る基盤整備 (ブロックチェーン技術を利用したサービスに関する国内外動向調査)報告書概要資料」

# ブロックチェーン技術の展開が有望な事例とその市場規模

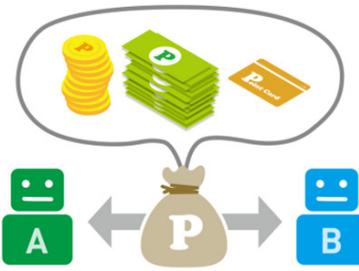


## ブロックチェーン技術による 社会変革の可能性



※記載金額は、ブロックチェーン技術が影響を及ぼす可能性のある市場規模

### 01 価値の流通・ポイント化 プラットフォームのインフラ化

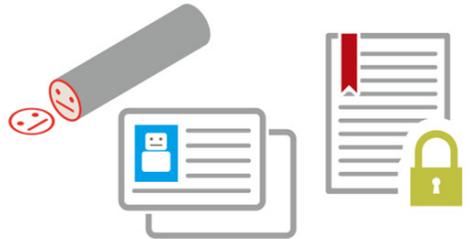


地域通貨    電子クーポン    ポイントサービス

自治体等が発行する地域通貨を、ブロックチェーン上で流通・管理

市場規模 **1兆円**

### 02 権利証明行為の 非中央集権化の実現

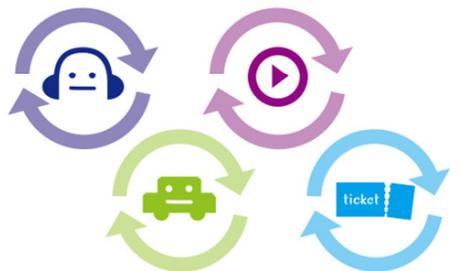


土地登記    電子カルテ    各種登録  
(出生・婚姻・転居)

土地の物理的現況や権利関係の情報を、ブロックチェーン上で登録・公示・管理

市場規模 **1兆円**

### 03 遊休資産ゼロ・ 高効率シェアリングの実現



デジタルコンテンツ    チケットサービス    C2Cオークション

資産等の利用権移転情報、提供者/利用者の評価情報をブロックチェーン上に記録

市場規模 **13兆円**

### 04 オープン・高効率・高信頼な サプライチェーンの実現

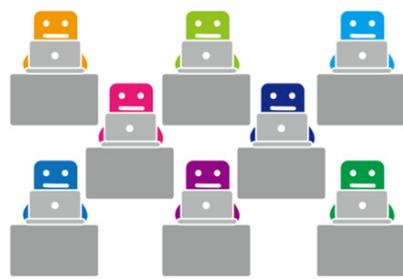


小売り    貴金属管理    美術品等真贋認証

製品の原材料からの製造過程と流通・販売までを、ブロックチェーン上で追跡

市場規模 **32兆円**

### 05 プロセス・取引の全自動化・ 効率化の実現



遺言    IoT    電力サービス

契約条件、履行内容、将来発生するプロセス等をブロックチェーン上に記録

市場規模 **20兆円**

出典：経済産業省 商務情報政策局 情報経済課「平成27年度 我が国経済社会の情報化・サービス化に係る基盤整備 (ブロックチェーン技術を利用したサービスに関する国内外動向調査)報告書概要資料」

# Hyperledger プロジェクト

- Linux Foundation がオープンソースソフトウェアによるブロックチェーン技術の整備を目指したもの
  - IBM, Intel, Fujitsu, Hitachi, NTT Data, NEC 等参加
- 現在 3 つのフレームワーク
  - Fabric (IBM), Swatooth Lake (Intel), Iroha (ソラミツ)
- 3 つともプライベート・コンソーシアム型ブロックチェーンであり、パブリック型でない
  - Byzantine fault-tolerant プロトコルベース  
(非許可型ではない分散計算プロトコル)
- 企業受けがいいので、これらが利用されるかも

## ビットコイン・ブロックチェーンの課題 (1/3)

- 51%攻撃（過半数正直者ハッシュパワーが必要）
  - 計算資源の半数を不正者が占めると破綻の可能性
  - 不正者に都合のよい分岐が正しいチェーンとなる
- マイニングの専門化（専用ハードウェア等）
- マイニングのためのエネルギー消費が膨大
  - Proof of Useful Work
  - 代替パズル：Proof of Stake, Proof of Space 等

## ビットコイン・ブロックチェーンの課題 (2/3)

- マイニングプールの構成
  - 単独マイニングでは報酬を獲得しにくいいため
  - プール管理者が力を持ち非中央集権化に逆行
- 取り引きの最終が確率的であり、時間がかかる
  - 分岐が正しくなる可能性が常に残る
- 匿名性の確保
  - ビットコインは取引内容をすべて公開・共有
  - 匿名性の高い暗号通貨：ZeroCoin, Zerocash

## ビットコイン・ブロックチェーンの課題 (3/3)

- インセンティブ設計
  - ビットコインでは、ブロック報酬と取引報酬
  - 報酬の設定方法・妥当性は？
  - 暗号通貨以外で利用するときの報酬は？
- 様々な暗号通貨をどのように選択すべきか
  - 800以上存在
  - 機能性・安全性の指標

# 暗号周辺における研究動向

## 以降で紹介する内容

- 暗号技術としての Nakamoto プロトコルの分析
  - Garay, Kiayias, Leonardos (Eurocrypt 2015)
  - Pass, Seeman, shelat (Eurocrypt 2017)
  - Bentov, Pass, Shi (ePrint 2016)
- 公開台帳の安全性モデル・不可能性
  - Pass, Shi (ePrint 2016) のモデル
- 望ましい性質をもつプロトコルの提案
  - 反応性 (responsiveness) : Pass, Shi (ePrint 2016)
  - 公平性 (fairness) : Pass, Shi (PODC 2017)
- マイニングプールの報酬関数の考察
  - Schrijvers, Bonneau, Boneh, Roughgarden (FC'16)

# Garay, Kiayias, Leonardos (Eurocrypt 2015)

- The Bitcoin Backbone Protocol: Analysis and Applications
- ブロックチェーンと公開台帳の機能を定式化
  - ブロックチェーン：(1) common prefix (2) chain quality
  - 公開台帳：(1) persistency (2) liveness
- Nakamoto プロトコルが、ブロックチェーンの機能を満たし、公開台帳を実現できることを証明
  - 敵対者のハッシュパワー  $p < 1/2$  のとき
  - 通信モデル：同期ネットワーク

# Pass, Seeman, shelat (Eurocrypt 2017)

- Analysis of Blockchain Protocol in Asynchronous Networks
- 部分的同期ネットワークにおける、Nakamoto の分析
  - 通信遅延の上限が所与
- 通信遅延の上限がない場合の Nakamoto への攻撃
  - ハッシュパワー  $\rho$  に関して安全な領域との差が緊密
- ブロックチェーンの機能に対する別の定式化
  - ブロックチェーン：(1) consistency (2) chain quality (3) chain growth
  - (1),(2),(3) を満たせば公開台帳を満たす

# Bentov, Pass, Shi (ePrint 2016)

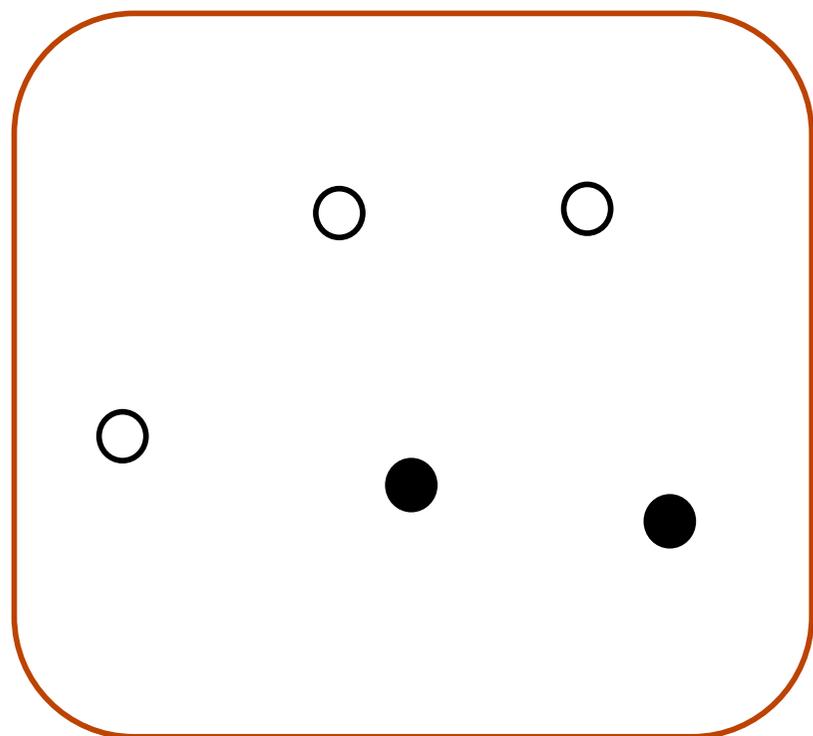
- The Sleepy Model of Concensus
- 正直・不正ノード以外に offline ノードがいる合意問題
  - 既存プロトコルでは「offline = 不正」扱い
- Sleepy モデルにおける合意プロトコルの提案
  - 公開台帳の機能 = state machine replication 問題
  - 設定・仮定
    - online の過半数は正直 (ただし permissioned 設定)
    - 部分的同期ネットワーク
    - weakly-synchronized clock, PKI, CRS (common reference string), CRH (collision-resistant hash)
  - アプローチ：Nakamoto の PoW を暗号技術で

# 公開台帳の安全性モデル

- Pass, Shi (ePrint 2016) によるモデル化
  - Proof-of-Work ベースを対象
- 公開台帳は以下の性質を満たす
  - 一貫性 (consistency) : 正直ノードは同じ台帳を管理
  - 生存性 (liveness) : 正直ノードは台帳に記録可能
- 設定
  - 環境  $Z$  と敵対者  $A$  が攻撃を実行
  - 部分的同期ネットワーク
  - Proof-of-Work をランダムオラクルを利用してモデル化
  - 正直ノードを敵対者  $A$  が制御するまでに遅延発生 (delayed corruption)

# 安全性モデル [PS16]

Z



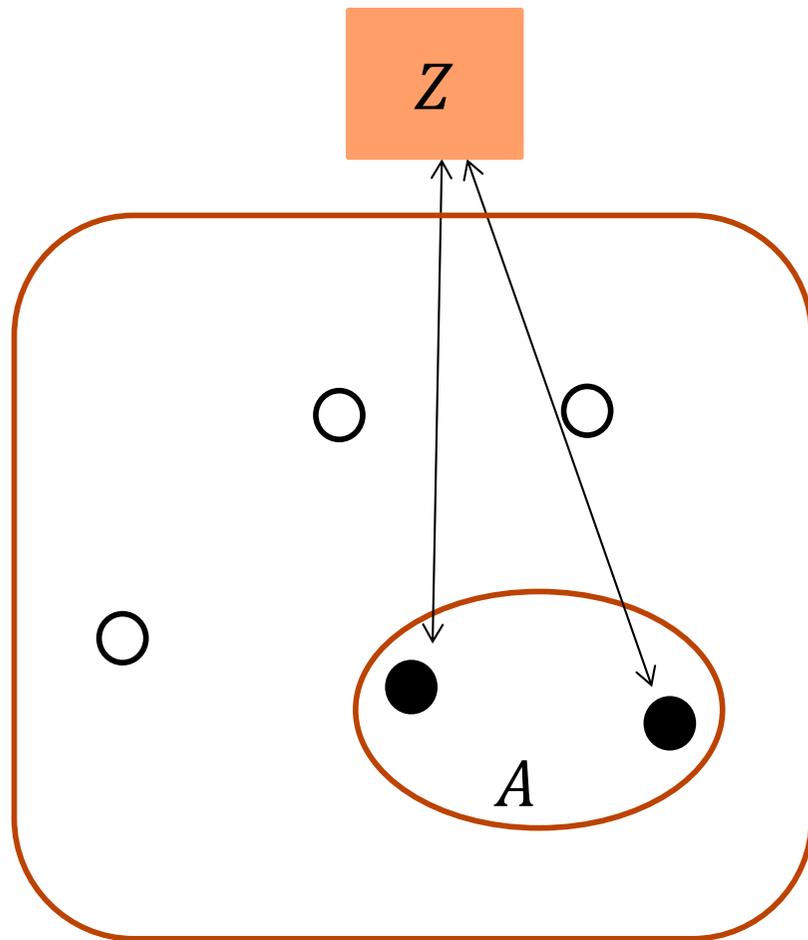
○ : 正直ノード

● : 不正ノード

- 環境 Z は、自由にノードを生成
  - 不正ノード割合は常に  $\rho$  以下
- 正直ノードは、プロトコルに従う
- 各タイムステップで正直ノード  $i$  は
  - 1度だけランダムオラクルを利用
  - $\text{LOG}_i$  を出力

$\text{LOG}_i$  : ノード  $i$  が承認した取引の系列

# 安全性モデル [PS16]



○ : 正直ノード

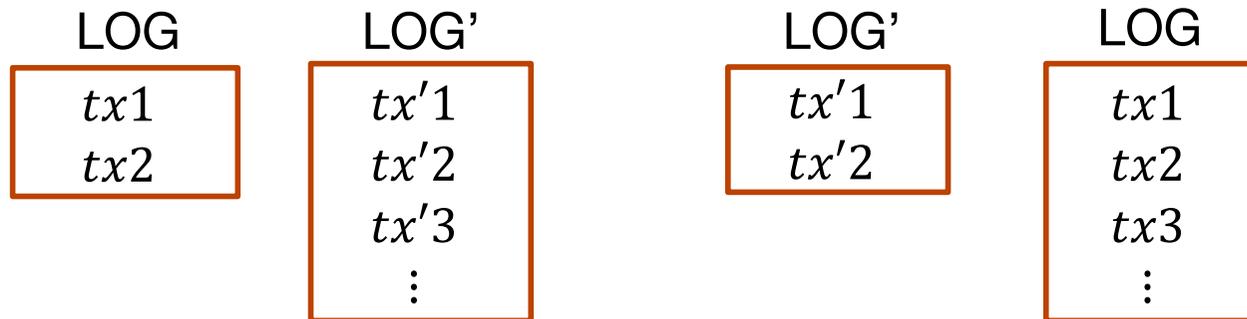
● : 不正ノード

- 環境  $Z$  は、不正ノードといつでも通信可能
- 不正ノードは敵対者  $A$  に制御され、任意の振る舞いが可能
- ノード間のメッセージ伝達は  $A$  が行う
  - 遅延を発生可能
  - メッセージに  $id$  はなく、遅延上限  $\Delta$  以内にすべての正直ノードへ送付

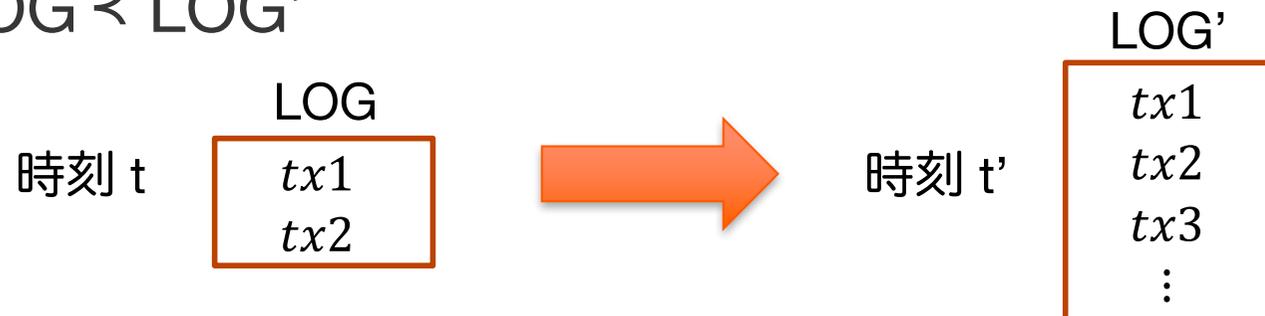
# 公開台帳の性質

## ■ 一貫性 (consistency)

- 共通の語頭 (common prefix) :  
正直ノード  $i$  と  $j$  が、時刻  $t, t'$  で LOG, LOG' を出力  
→ LOG < LOG' または LOG' < LOG



- 自己一貫性 (self-consistency) :  
正直ノード  $i$  が、時刻  $t, t'$  ( $t < t'$ ) で LOG, LOG' を出力  
→ LOG < LOG'

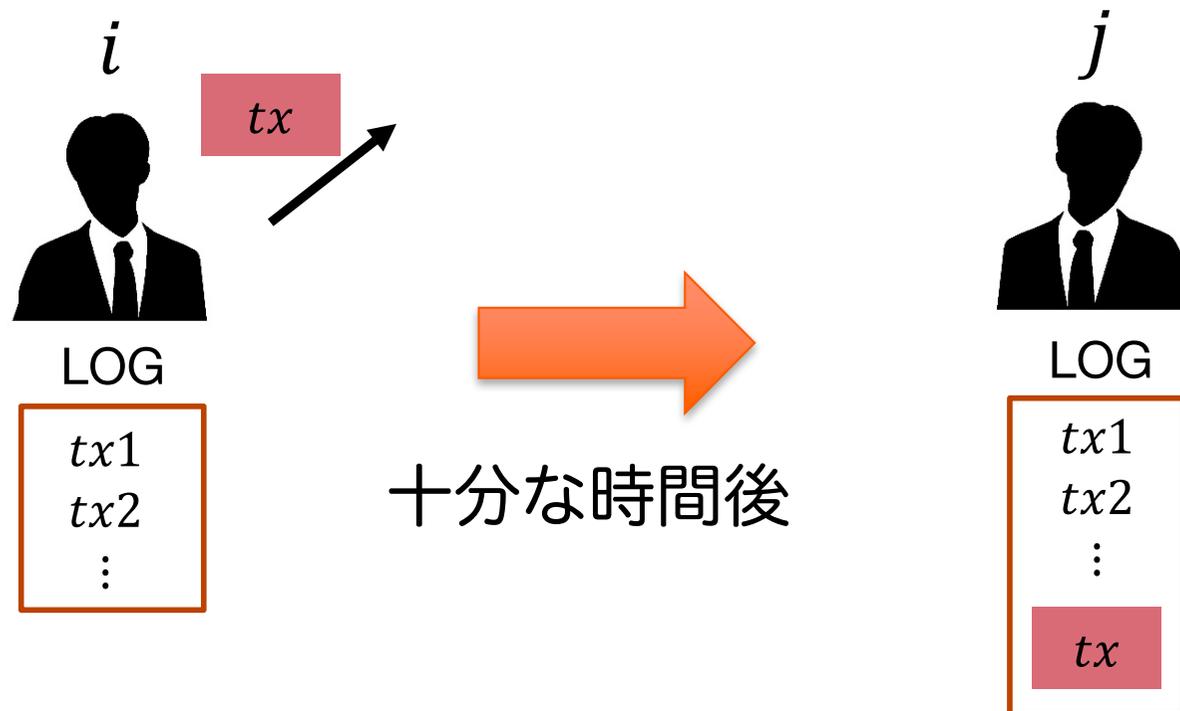


# 公開台帳の性質

- 生存性 (liveness) :

時刻  $t \geq T_{\text{warmup}}$  に正直なノードが  $tx$  を入力

→ 時刻  $t' \geq t + T_{\text{confirm}}$  の正直ノードは  $tx$  を LOG に含む



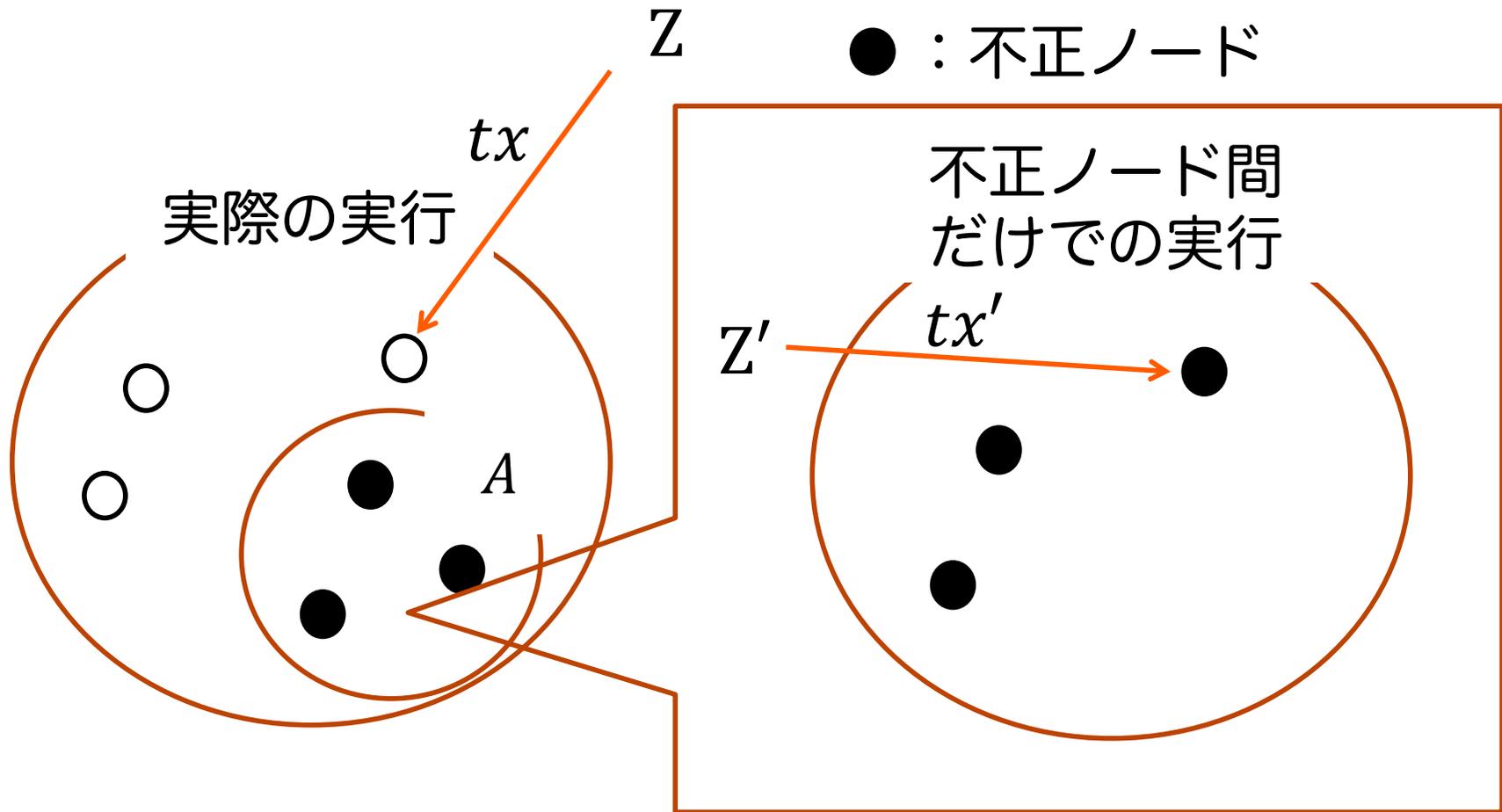
# 公開台帳に関する不可能性

- 敵対者ハッシュパワー  $\rho > 1/2$  では構成不可能
  - [稲澤, 越中谷, 安永, 満保 (2017)]
- 正直ノードは計算を止めることができない
  - [Pass, Shi (2016)]
- 敵対者ハッシュパワー  $\rho > 1/3$  のとき  
反応的プロトコルは構成不可能
  - 反応的 (responsive) :  $T_{\text{confirm}}$  が実際の遅延に依存
  - [Pass, Shi (2016)]

# 敵対者ハッシュパワー > 1/2 での構成不可能

○ : 正直ノード

● : 不正ノード

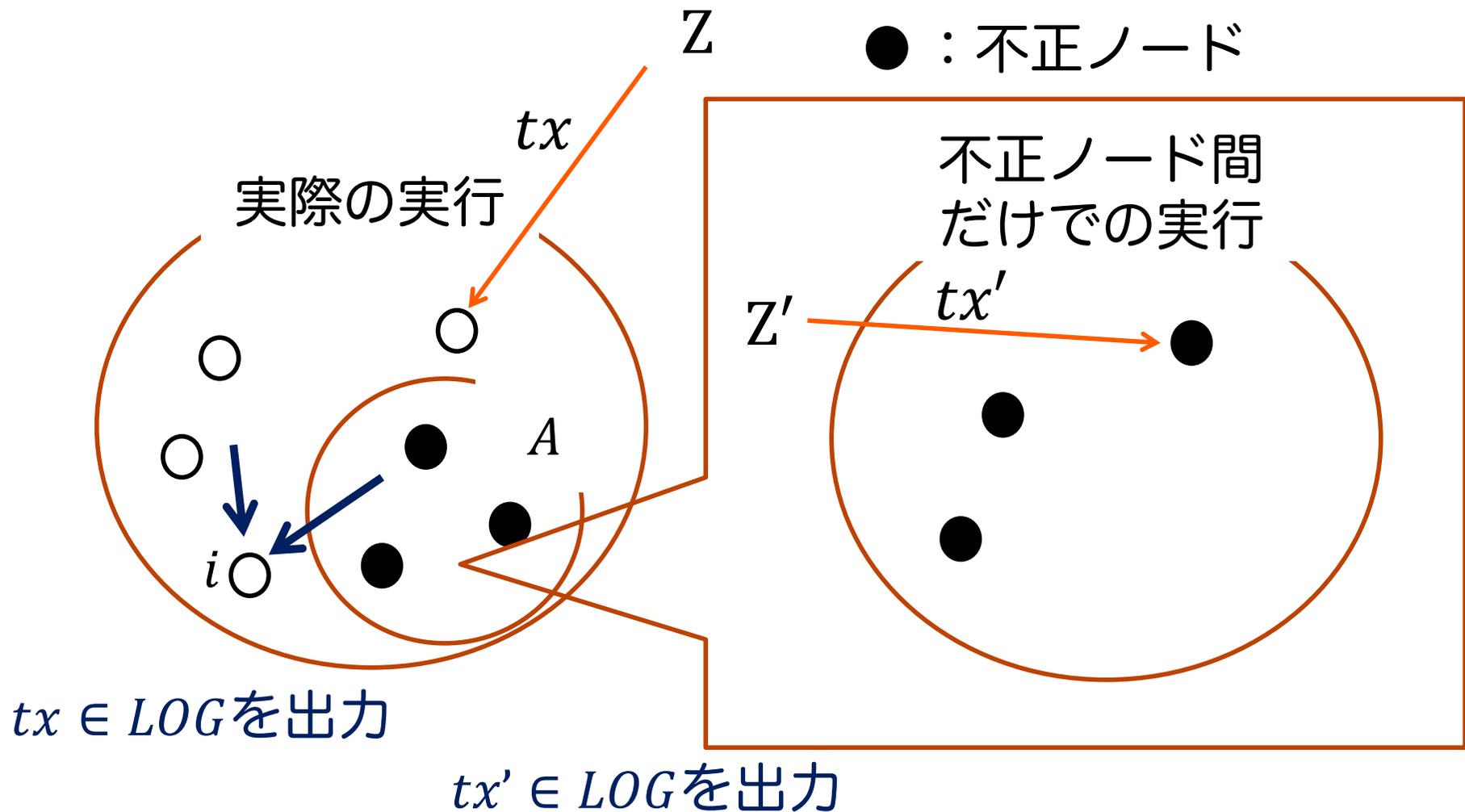


- Z は正直ノードにランダム tx を入力
- 敵対者 A は半分のノードを制御し、不正ノード間だけでランダム tx' を入力

# 敵対者ハッシュパワー > 1/2 での構成不可能

○ : 正直ノード

● : 不正ノード



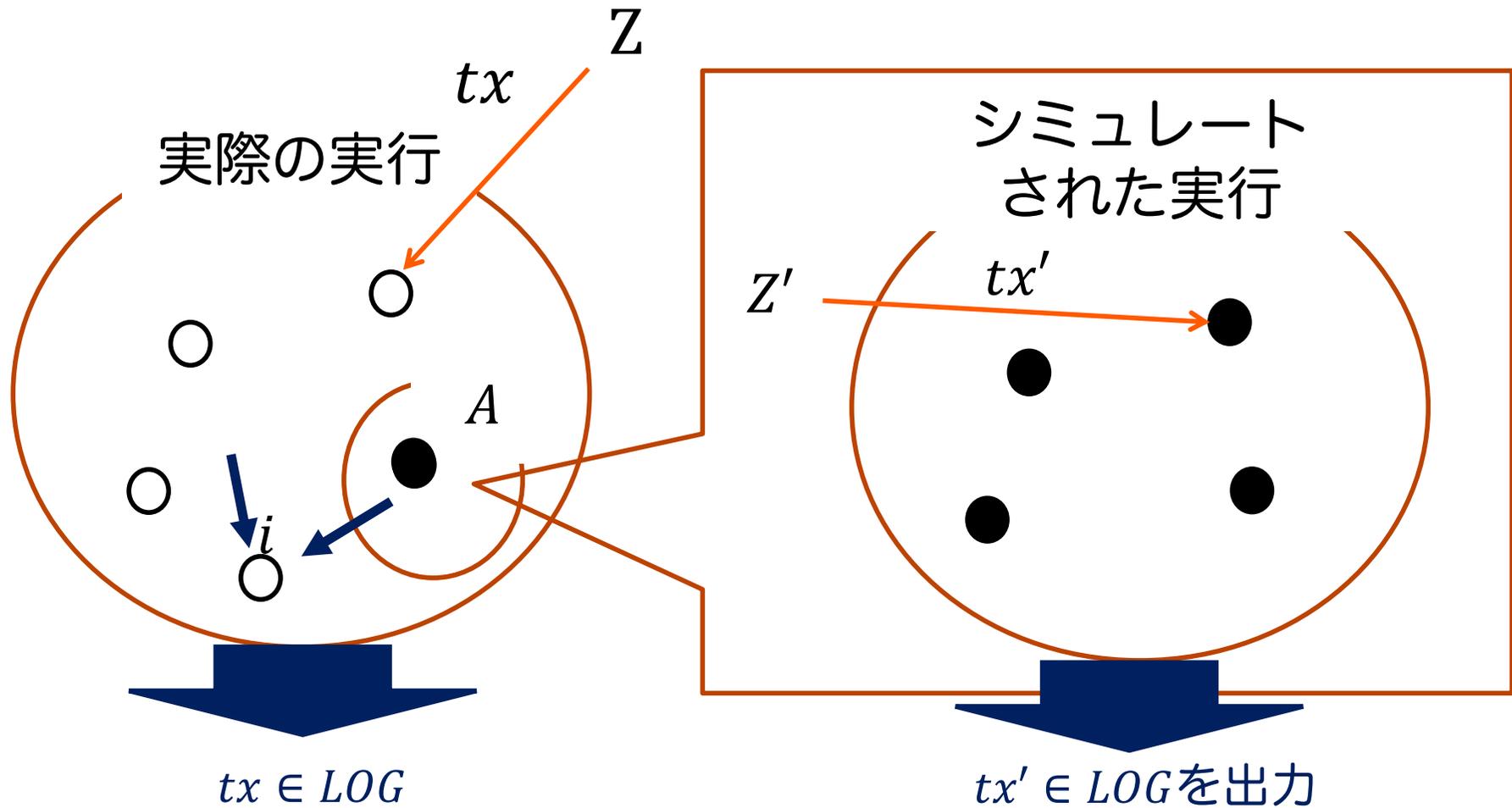
- 新規ノードを生成し、不正ノードは正直に振る舞う
- 新規ノードはどちらが正しいか区別できない

# 敵対者ハッシュパワー $> 1/2$ での構成不可能

## ■ 証明概要

- $n/2$  正直ノードがランダムな tx を入力
- 同時に  $n/2$  不正ノードだけでランダム tx' を入力
  - 正直ノードの通信は無視し、不正ノード間だけで実行
- 新規正直ノードを生成
- 不正ノードも正直に振る舞う
- 十分時間が経つと、新規ノードのログの先頭には tx/tx' のいずれか
  - 新規ノードはどちらが正しいか区別できない
- 確率  $1/2$  で tx は先頭になく、一貫性を満たさない

# 正直ノードは計算を止めることができない



- 正直ノードは  $tx$  入力後、計算をやめる (と仮定)
- $A$  は正直に振る舞い、頭の中で、同様の実行をシミュレート
- 新規ノード生成後、シミュレート実行の通信を行うと、新規ノードはどちらが正しいか区別できない

# 正直ノードは計算を止めることができない

## ■ 証明概要

- 正直ノードがランダム tx を入力
  - その後、新規入力がないため計算を止めてもよい
- 敵対者 A は、頭の中で、正直ノードの実行をシミュレートし、ランダム tx' を入力
  - 外とは通信を行わない
- 新規正直ノードを生成
- A は不正ノードを利用して、シミュレートしているノードの通信を全体へ行う
- 十分時間が経つと、  
新規ノードのログの先頭に tx/tx' のいずれか  
→ 確率 1/2 で一貫性を満たさない

# 敵対者ハッシュパワー > 1/3 での 対応的プロトコルの構成不可能

- プロトコルが**反応的 (responsive)**  
⇔  $T_{\text{confirm}}$  が遅延上限  $\Delta$  でなく実際の遅延  $\delta$  に依存
- **証明概要**
  - $n/3$  人ずつのグループ A, B, C に分割
    - A は corrupt され, B, C は honest
    - 1/3-ハッシュパワー耐性 → 2グループ実行で合意可能
  - A は、B に対して正直に振る舞い、  
C に対して時間遅れで正直に振る舞う
    - PoW のため、A は B, C と同時に対応できない
  - B-C 間は遅延が発生して通信できない
  - A-B 間で tx, A-C 間で tx' を合意 → 矛盾

# Pass, Shi (ePrint 2016)

- Hybrid consensus: Efficient Consensus in the Permissionless Model
- 公開台帳の定式化・反応性 (responsiveness) の導入
- 反応的公開台帳をハッシュパワー  $\rho < 1/3$  で構成
  - ブロックチェーン + byzantine fault tolerance (BFT) で実現
    - ブロックチェーンで委員を選び、そのメンバで BFT
  - Nakamoto だと 3/4-honest、Fruitchain [PS2017] だと 2/3-honest が必要
- 不可能性に関する考察

# 利己的マイニング (selfish-mining)

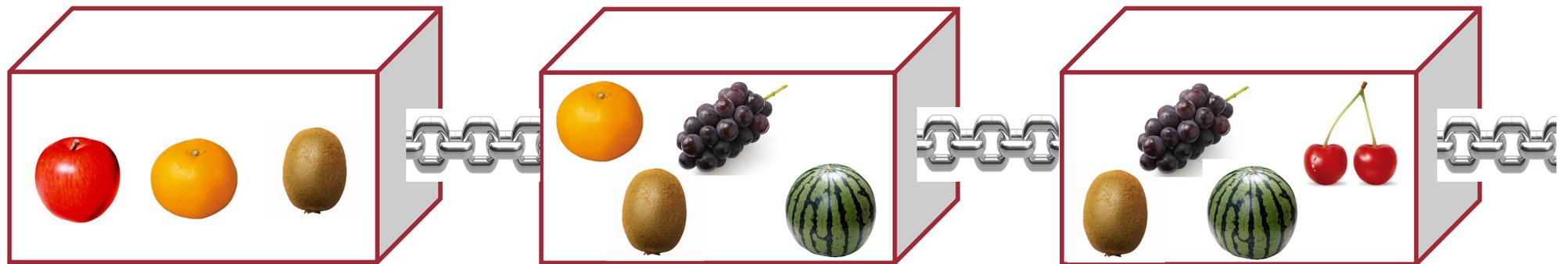
- Nakamoto に対する利己的マイニング攻撃
  - 新しいブロックを採掘 → 出さずに取っておく
  - 他の正直プレイヤーが採掘 → それを出す
- 正直プレイヤーの計算を無駄にできる
  - 敵対者が  $\rho$  ハッシュパワーを持つとき、 $T$  ブロックのうち、期待値は  $\rho T$  個だが、成長割合が  $(1 - \rho)T$  であるため  $\rho/(1 - \rho)$  貢献可能
    - $\rho$  が  $1/2$  に近い → ほとんどすべての割合貢献

# Pass, Shi (PODC 2017)

- Fruitchains: A Fair Blockchain
- 公平性をもつブロックチェーンの提案
  - 公平性： $\beta$  割合ハッシュパワー  $\rightarrow$   $\beta$  割合ブロック貢献
  - 利己的マイニングへの対策
- アイディア：情報一貫性のためのマイニングと同時に、データを保存するフルーツマイニングを実行
- 「取引手数料報酬」を利得と考えるとき  $n/2$ -結託耐性ナッシュ均衡を実現
  - 結託しても期待報酬が増えないため
- 報酬の分散の低減化（マイニングプール対策）
  - 難しさの異なる2種類のマイニングを利用できる

# Fruitchain の概要

- データはフルーツマイニングで入れる (→フルーツ)
  - Nakamoto ブロックチェーンにはフルーツを入れる
    - 2-for-1 trick [GKL15] で同時に実行可能
    - 受け取った古くないフルーツをすべて入れる
- ブロック貢献 = フルーツ貢献 ≈ ハッシュパワー



# マイニングプール

- ビットコインマイニングは報酬は高いが難しい
  - 25 BTC = 6,000 USD, 数年に一回成功
    - 専用ハードウェアでも3ヶ月に一回程度
  - 無記憶過程であり、1年費やしても成功率は不変
- 多くのマイナーはマイニングプールに参加して安定した報酬を受け取れることを望む
  - 参加者はブロック（=解）とともにシェア（=解に近いもの）を提出し、その内容を元に報酬分割
- 報酬の分け方はプール毎に様々
  - 報酬の分け方は「誘因両立」であるか？

# Schrijvers, Bonneau, Boneh, Roughgarden (FC'16)

- Incentive Compatibility of Bitcoin Mining Pool Reward Functions
- マイニングプール内の報酬関数のためのゲーム理論的モデル
  - プールは1つだけ
    - 他のプール・単独マイニングへの変更は考えない
  - 誘因両立性 (Incentive Compatibility) を定義
  - 比例報酬は誘因両立でない,  
誘因両立性を満たす新しい関数を導入,  
pay-per-last-N-shares (PPLNS) は 誘因両立

# 設定

- 採掘者  $n$  人で固定
- 採掘者  $i$  の採掘力  $\alpha_i$ ,  $\sum_{i=1}^n \alpha_i = 1$
- 採掘者がシェアを見つけるまでの時間  
= パラメータ  $\alpha_i$  の指数分布
  - 期待値  $1/\alpha_i$
  - 各シェアは確率  $1/D$  でブロック (解)

# 報酬関数、履歴、採掘者戦略

- 報酬関数  $R: H \rightarrow [0,1]^n$ 
  - 履歴から割り当て  $\{a_i\}_{i=1}^n, \sum_{i=1}^n a_i = 1$  を決定
- 順序なし履歴  $\vec{b} = (b_1, \dots, b_n) \in \mathbb{N}^n$  を利用
  - 当該ラウンドで採掘者  $i$  は  $b_i$  個シェアを報告
  - 実際には、報告順や過去ラウンドの報告などを利用する場合も
- 採掘者戦略  $\sigma$ 
  - 報酬関数  $R$  に対し、採掘者は戦略  $\sigma(R)$  を選択
  - $$\sigma(R) = \max_{\sigma} \lim_{t \rightarrow \infty} \frac{\sum_{j=1}^T R_i(\vec{b}_j)}{t}$$
    - $t$ : 採掘者  $i$  の採掘時間,  $T$ : 時間  $t$  までのラウンド数,  
 $\vec{b}_j$ : ラウンド  $j$  における採掘者の提出シェア数

# 報酬関数に求められる性質

- 性質 1 : 報酬関数  $R$  が誘因両立  
⇔ 採掘者の最適反応戦略  $\sigma(R)$  は解をすぐに報告
  - 厳密な定義は後で
- 性質 2 : 報酬関数  $R$  が比例支払い  
⇔ 各採掘者  $i$  に対し  $\mathbb{E}_b[R_i(\vec{b})] = \alpha_i$ 
  - 現実には、小さい割合  $f$  の手数料徴収なども
- 性質 3 : 報酬関数  $R$  が  $(\gamma, \delta)$ -予算均衡  
⇔  $\forall \vec{b}, \gamma \leq \sum_{i=1}^n R_i(\vec{b}) \leq \delta$ 
  - $\gamma < 1$  のとき分配されない額が存在

# 誘因両立性

- 採掘者は解を見つけたとき、すぐに報告するか、さらに  $d$  個シェアが見つかるまで待つか
- 時刻  $t$  に採掘者  $i$  が解を見つけ、それまでに  $\vec{b}_t$  のシェアが報告されていたとき

- $d$  個分待ったときの期待報酬 (1)

$$\mathbb{E}_{\vec{b} \text{ s.t. } \|\vec{b}\|_1 = d} [R_i(\vec{b}_t + \vec{b})] = \sum_{\vec{b} \text{ s.t. } \|\vec{b}\|_1 = d} \Pr[\vec{b}] R_i(\vec{b}_t + \vec{b})$$

- すぐに報告したときの期待報酬 (2) (すぐに報告し、次のラウンド開始により、 $d$  シェア分の報酬獲得)

$$R_i(\vec{b}_t) + d \frac{\mathbb{E}_{\vec{b}} [R_i(\vec{b})]}{\mathbb{E}_{\vec{b}} [\|\vec{b}\|_1]} = R_i(\vec{b}_t) + \frac{d}{D} \mathbb{E}_{\vec{b}} [R_i(\vec{b})]$$

- (1)  $\leq$  (2) のとき、 $R$  は誘因両立

# 既存の報酬関数

- 比例報酬 (proportional)  $R_i^{(prop)}(\vec{b}) = \frac{b_i}{\|\vec{b}\|_1}$ 
  - 誘因両立でない。  
期待値よりも少ないシェアしか見つからない場合、解が見つかったとしても、期待値通りのシェアが見つかるまで待つ
- シェア毎支払い (pay-per-share)  $R_i^{(pps)}(\vec{b}) = \frac{b_i}{D}$ 
  - 誘因両立であるが  $(1/D, \infty)$ -予算均衡。  
解を出しても出さなくても同額なので出したほうが良い。一方、見つかるシェア数に制限がない

# 新しい報酬関数

- シェア数だけでなく解の発見者を考慮した関数

- $$R_i^{(ic)}(\vec{b}, s) = \frac{b_i}{\max\{\|\vec{b}\|_1, D\}} + I\{i = s\} \left( 1 - \frac{\|\vec{b}\|_1}{\max\{\|\vec{b}\|_1, D\}} \right)$$

- $\|\vec{b}\|_1 \geq D$  のとき、第2項は 0 で、比例報酬に一致
- $\|\vec{b}\|_1 < D$  のとき、各シェアに  $\frac{b_i}{D}$  割り当て、残りを解の発見者に
- 比例支払い、誘因両立、(1,1)-予算均衡である
- 報酬 63% ( $\approx 1 - e^{-1}$ ) はシェア提出者へ支払われる
- 分散は単独採掘と同等だが、目標額を得るための時間は比例報酬の定数倍程度 ←分散だけでは見えない

# より多くの情報を必要とする報酬関数

- 最終 N シェア毎支払い (pay-per-last-N-shares)

- $R_i^{(pplns)}(\vec{s}) = \frac{\#\{s_j: s_j \in \vec{s} \wedge s_j=i\}}{N}$

- すぐに解を提出するか、次のシェアが見つかるまで待つか、を比較 (先ほどより弱い)
- $\alpha_i < 1 - \frac{D}{N}$  のとき、すぐに解を提出
- $N \geq D$  のとき、すぐに解を提出

# その他の研究動向

- ブロック報酬がなくなったときの問題
  - Carlsten, Kalodner, Weinberg, Narayanan, “On the Instability of Bitcoin Without the Block Reward” (CCS 2016)
- Proof of Stake ベースのプロトコル
  - Daian, Pass, Shi, “Snow white: Provably secure proofs of stake” (ePrint 2016)
  - Kiayias, Russell, David, Oliynykov, “Ouroboros: A provably secure proof-of-stake blockchain protocol” (Crypto 2017)
- Proof of Work の困難性変化を考慮した分析
  - Garay, Kiayias, Leonardos, “The Bitcoin Backbone Protocol with Chains of Variable Difficulty” (Crypto 2017)
- 汎用的結合可能なブロックチェーン
  - Badertscher, Maurer, Tschudi, Zikas, “Bitcoin as a Transaction Ledger: A Composable Treatment” (Crypto 2017)
- Proof of Useful Work 関係
  - Ball, Rosen, Sabin, Vasudevan, “Proofs of Useful Work” (ePrint 2017)
  - Ball, Rosen, Sabin, Vasudevan, “Average-Case Fine-Grained Hardness” (STOC 2017)

# Carlsten, Kalodner, Weinberg, Narayanan (CCS 2016)

## ■ On the Instability of Bitcoin Without the Block Reward

### ■ ブロック報酬と取引報酬

- 初期はブロック報酬がメインだが4年で半減
- 「取引の差額 = 取引手数料」としてマイナーが入手

### ■ 取引報酬だけになったときの問題点

- 取引手数料 100 BTC 分がマイニングされ、5 BTC 分の取引が残っているとき、
  - (1) 最長チェーンをマイニングして 5 BTC を受け取り、残り 0 BTC とするか
  - (2) 1 つ前に戻って枝分かれマイニングして 105 BTC のうち 55 BTC を受け取り、残り 50 BTC とするか
- マイナーにとって様々な戦略が存在
- 取引手数料が時間により変わることが原因