SPECTRE:

プルーフ・オブ・ワークのシリアライズ: 再帰的選択を通じたトランザクションの確認

Recursive Elections

Yonatan Sompolinsky, Yoad Lewenberg及び Aviv Zohar
ヘブライ大学工学・コンピュータサイエンス学部、イスラエル
{yoni sompo,yoadlew,avivz}@cs.huji.ac.il

要旨

ビットコインでは誰もが匿名でプロトコルに参加可能な無許可環境での一貫したトランザクションセットに関する合意を達成するためにナカモトコンセンサスを活用している。その台頭以来、その他多くの無許可コンセンサスプロトコルが提案されている。

我々は高スループットや迅速な確認時間下においてもセキュアな状態を維持する暗号通貨のコンセンサスコアの新しいプロトコルであるSPECTREを発表する。SPECTREはどのようなスループットでも計算能力の最大50%の攻撃者に対して耐性を持つ（ネットワークや渋滞や帯域幅誓約によって定義する限度に到達）。SPECTREは高いブロック作成レートを持ち、トランザクションをたった数秒で確認できる（主にネットワークのラウンドトリップタイムによって制限を受ける）。

SPECTREの基礎となるモデルは部分的同期ネットワークのカテゴリーに分類される。そのセキュリティは誠実な参加者間のメッセージの送信時間の限界の存在に依存するが、プロトコル自体にはこの限界に依存するパラメータは含まれない。そのため、そのようなパラメータのエンコードを行うその他のプロトコルは極端な安全マージンを持って運用する必要があるが、SPECTREは実際のネットワーク遅延に従って収束する。

SPECTREの成果は従来のコンセンサスの要件よりも弱い特性を満たしているという点である。従来のパラダイムでは、全ての破損していないノードが2つのトランザクション間の順位を決定して合意する必要があった。対照的にSPECTREは誠実なユーザが実施するトランザクションに関してのみこの要件を満たす。通貨に関して同時に発表される2つの相反する支払いは不誠実なユーザによってのみ発生しうると判断するため、システムのユーザビリティを損なうことなくそのようなトランザクションの受諾を遅延することができる。我々の枠組みでは暗号通貨の分散型元帳においてこの弱い要件セットを正式化している。その後、SPECTREがこれらの要件を満たしているという正式な証拠を提供する。

1. 概要

ビットコインはサトシ・ナカモトが発明及び導入した新しい暗号通貨システムであり付随するプロトコルである[13]。通貨トランザクションは公開管理された元帳であるブロックチェーン内で整理される。チェーン内の各ブロックは通貨のユーザが発表したトランザクションのバッチである。ブロックチェーンを延長する新規ブロックでは整合性を維持する必要があるためブロックチェーンには整合性のとれたトランザクションだけが含まれる。

残念ながら、ナカモトコンセンサスには重大なスケーラビリティの制約がある [5], [18], [14]。より大きいブロック、もしくはより頻繁にブロックを作成することで高いトランザクションスループットをサポートするためにプロトコルを調整するには基礎となるネットワーク上でより強い前提が必要となるため、安全マージンが小さくなる。

本論文において我々は高いスケーラビリティを達成する新プロトコルである SPECTREを提案する。SPECTRE内のトランザクションは数秒で確認可能であり、ビットコインと比較してスループットを桁違いに改善できる。SPECTREはネットワークインフラと容量によってのみ制限を受ける。

そのため、本プロトコルではナカモトコンセンサスによって発生するセキュリティとスケーラビリティの二律背反の関係を緩和することができる。

SPECTREでは全てのブロックをカウントして元帳に統合する。技術的な話をすると、SPECTREはナカモトのブロックチェーンをブロックDAGと呼ばれる有向非巡回グラフに一般化する。ブロックのDAG全てを維持することによってSPECTREはマイナーが同時に、そしてより頻繁にブロックを作成することを可能にする。このデザインはノードがブロック作成時に選択したチェーンのIDに関する見解の相違を調整する必要を回避することを意図している。

SPECTREのコンセンサス特性の論法では新しい正式な枠組みが必要となる。

実際、ナカモトコンセンサスの堅牢性について形式化した以前の取り組み [7], [15]では元帳内のブロックの堅牢性に集中している。SPECTREでは全てのブロックがDAGに統合されるが、DAGに組み込まれる個別のトランザクションが衝突によって拒否される可能性があるためにこれをすぐにSPECTREのトランザクションの堅牢性にまで拡大することはできない。

そのため、本論文では次の2つの内容について論じている: (1) 内在的にスケーラブルなプロトコル、SPECTRE (2) 元帳の表現において必ずしもブロックのチェーンを使用しない暗号通貨支払いプロトコルの正式な枠組み (この点に関して我々の枠組みは以前に提案された枠組みとは異なる)。我々はこれをSPECTREに対して適用し、SPECTREの堅牢性について詳細に分析する。

SPECTREの基本となる主要な技術はDAG内の各ブロックペア間の順序に関する投票アルゴリズムである。投票を行うのはブロック（マイナーではない）である。各ブロックの投票をDAGの位置に従ってアルゴリズムによって解釈する（そしてインタラクティブに提供しない）。我々は過半数による投票は迅速に不可逆的なものとなることを把握しており、この過半数による投票を用いてトランザクションの整合性のとれたセットを抽出する。基本的にビットコインの最長チェーンルールも投票メカニズムとみなすことができる。各ブロックがそのブロックの含まれる全てのチェーンに1票を追加する。スコアの最も高いチェーンは最長チェーンでもある。ただし、以下で示すようにビットコインによる「1つの勝者チェーン」の選択によってビットコインは本質的にスケーラビリティの問題を抱えることになっている。

最近になって公開ブロックチェーンシステムの新プロトコルに関する複数のプロジェ

クトが発表されている。これらにはBitcoin-NG [6], Byzcoin [9], Deckerらによるプロジェクト[4]、Hybrid Consensus [16], Solidus [1]そしてAlgorand [11]が含まれる。これらのプロジェクト及びその他の関連プロジェクトについてはセクション6で説明する。

2. 問題の公式声明

このセクションでは暗号通貨プロトコルのセキュリティ及びスケーラビリティ特性に関する論法の一般的な枠組みについて説明する。一般的に、我々の枠組みでは暗号通貨プロトコルでブロックの作成とブロック元帳の作成に関するマイニングプロトコルと元帳を解釈してそこから有効なトランザクションの整合性のとれたサブセットを抽出するためのTxOプロトコルの2つのプロトコルを指定している。プロトコル内のトランザクションは時間の経過と共に高い確率で受け入れられるため、ユーザは追加でロバストTxOプロトコルを実行して受け入れられたトランザクションの堅牢性を数量化する。悪意のある攻撃者がこれを試みた場合にトランザクションが無効になるという確率に対する限界（例えば、ビットコイントランザクションでは攻撃者が自身の存在していないより長い代替チェーンの作成に成功した場合にはトランザクションが無効になる。この事象は時間の経過と共に発生する確率が低下する）。次に、出来るだけ一般性を維持するために我々の枠組みを抽象的な意味において説明する。セクション4では要件を満たし、要件を満たすためにブロックDAGを使用するプロトコルについて説明する。それまで解決策及びマイニングプロトコルの詳細については説明しない。我々の枠組みを用いてSPECTREがビットコインの苦しむセキュリティとスケーラビリティの両立の問題を回避する方法について明確に説明する。

トランザクション

トランザクションは基本的にtxと表記される。$inputs\,(tx)$ はtxを受諾する前に受諾する必要のあるトランザクションセットである。これらはtxにおいて費やす貨幣を提供するトランザクションである。2つの異なるトランザクションtx_1 と tx_2 は共通のインプットを共有する場合には衝突する。これら2つのトランザクションは同じ通貨を消費するため、$tx_2\ 2\ conflict\ (tx_1)$と書く(これは対称関係である)。

マイニングプロトコル

N はノード（マイナー）のセットである。マイナーはマイニングプロトコルに従って元帳にトランザクションを追加してメッセージを伝播することで元帳の維持と拡大を行う。

サイズB KBのメッセージをシステム内の全ノードに伝播するために要する時間は$D = D(B)$ 秒以下と想定される。ここでは、マイニングプロトコルをマイナーが従う必要のあるルールの抽象的なセットとみなす。*honest* とはプロトコルの指示に常に従うノード、*malicious* とは悪意のあるノードを意味する。

我々が集中するプロトコル群では、マイナーが計算能力を保有し、プルーフ・オブ・ワーク（PoW）を実行する。α は攻撃者の相対的な計算能力を意味する。正式には、システム内の次のPoWの作成者が*malicious*である確率があり、これについては適切に定義されている。PoW作成はメモリなしのプロセスとしてモデリングされている [13], [18], [17]。

元帳の作成

マイニングプロトコルの結果はトランザクションを含む元帳とも呼ばれる（抽象的）

公開データ構造Gである(我々のソリューション提案ではブロックDAGとして事例を挙げて説明する)。ノードはローカルで元帳を複製する。ノードは元帳に関する認識が若干異なる(ブロックを全てのノードに伝播するには時間がかかるため)ために、tの時点でノードvが観察する元帳の状態をG_{vt}とする。ローカル環境が重要でない場合にはG_tと記載する。

TxOプロトコル

公開元帳Gの場合、*TxOプロトコル* では$TxO(G)$と記載されるGから整合性のあるトランザクションのサブセットを抽出する。このセット内の全てのトランザクションでもインプットが必要であり、セット内の別のトランザクションと衝突することはできません。

ロバストTxOプロトコル

システムのユーザは自身の支払いに関する保証を得る必要がある。
基本的に、全てのユーザがトランザクションを受け入れ、永久的にトランザクションが受け入れられた状態を維持することを保証したい。G_tの場合、*ロバストTxOプロトコル* では$RobustTxO(G_t)$と表記される$TxO(G_t)$のサブセットを指定し、誤り確率ϵまでトランザクションが受け入れられた状態を維持することが保証される受諾済みトランザクションセットを表現する。 *RobustTxO* はインプットとしてG_{vt}（vのローカルレプリカ）、と$D,\lambda,\alpha,$ 及びϵの値を取る。[1] (*Robust*) *TxO* の*tx*は*(robustly) accepted*とみなされる。

望ましい特性

そのため、暗号通貨プロトコルには以下の特性が必須となる。

特性 1 (整合性)

受諾されるセットの整合性が取れている: 元帳Gに関して

1) $tx \in TxO(G)$ と $tx_2 \in inputs\,(tx)$ の場合は$tx_2 \in TxO(G)$

2) $tx \in TxO(G)$と $tx_2 \in conflict\,(tx)$ の場合は $tx_2 \notin TxO(G)$

特性 2 (安全性)

あるノードがトランザクションを確実に受諾する場合、高い確率で全てのノードによって永久的に確実に受諾され、このイベントの待機時間は一定であることが期待される。

正式に、$tx \in RobustTxO(G_t^v, D, \lambda, \alpha, \epsilon)$ の場合には$\forall \epsilon > 0, \forall v \in \mathcal{N}$ であり、確率は少なくとも$1-\epsilon$である。$\tau \geq t$ の場合には $\forall u \in \mathcal{N}, \forall s \geq \tau : tx \in RobustTxO(G_s^u, D, \lambda, \alpha, \epsilon)$ である。この事象が発生する場合、$\tau - t$ で期待される値は一定である。

特性 3 (生存性の弱さ)

元帳内でトランザクションが公開される場合、しばらく後に全てのノードによって確実に受諾される。ただし、そのインプットが確実に受諾され、衝突するトランザクションが公開されていないことが条件である。正式には $v \in \mathcal{N},\ tx \in G_t^v$ 及び $\epsilon > 0$ 。

$\psi(t, D, \lambda, \alpha, \epsilon) := \min\{s \geq t : tx \in RobustTxO(G_s^v, D, \lambda, \alpha, \epsilon)\}$ はvによる確実な受諾の待機時間を定義する。その後、

$\mathbb{E}\left[\psi - t \mid inputs(tx) \subseteq TxO(G_\psi^v) \wedge conflict(tx) \cap G_\psi^v = \emptyset\right]$ は一定である。

定義1

暗号通貨レベルのセキュリティ閾値は最大αによって定義される(攻撃者の相対計算力) 。これに対して特性 1-3は条件を満たしている。

特性2と3で明記される$\tau - t$と$\psi - t$の値は指定されたプロトコル内のトランザクションの確認のための待機時間を定義する。

生存性特性の「弱さ」は短時間に連続して衝突するトランザクションが公開された場合には解決を保証しない（攻撃者が防止するには難しいが）という事実に対応している。これを、全ての衝突について限られた時間内に決定する必要のある従来のコンセンサスプロトコルと比較して考えると、この特性は生存性と呼ばれる。

ただし、システムの誠実なユーザは決して衝突するトランザクションを公開することはなく、元の資金（インプット）を自身で確実に受諾した後にのみ通貨の移動を行う。そのため、誠実なユーザによる支払いは弱い生存性において正式化された条件を満たしていることが保証され、確実に受諾される。一方で、詐欺を試みる攻撃者は衝突を公開する前に被害者が受諾するまで自身の攻撃の秘密性を維持する必要がある。その場合、被害者は高い確率で自身のトランザクションが無効とならないことを保証される。そのため、これら2つの特性によって誠実なユーザの支払いが一定の予期される時間内に確実に受諾され、永久的に受諾された状態を維持することが保証される。

[1]明確に説明すると、これらの値を既知の値とみなす。ただし、SPECTREではノードがこれらのパラメータの価格値を把握している、もしくは合意しているという推定は行わないことを強調する。セクション 3を参照。

本プロジェクトでは我々は高いスループットをサポートし、迅速な確認時間を達成しながら、高いセキュリティを維持するプロトコルのデザインを目指している。

3. SPECTRE vs ビットコイン – 概要

SPECTREではビットコインのソリューションの多くの特徴を採用している。特に、マイナーはトランザクションのバッチであるブロックを作成する。有効なブロックにはPoWパズルに対する解決策が含まれている必要がある(例えば、ビットコインでは部分的なSHA256衝突に基づくPoWを使用する)。λと記載されるブロック作成率はPoWの難易度を場合によって再調整することでプロトコルによって一定に維持されている。付属文書DではSPECTREのこのメカニズムについて詳細に説明している。ブロックサイズはBKBによる制限を受ける。

ビットコインのスループットはブロックサイズ限度（これによってDが増加する）及び/もしくはブロック作成率λを増加させることによって増加させることができる。そのため、セキュリティ閾値もしくはナカモトコンセンサスは$D \cdot \lambda$の増加によって低下することは立証されている。

理論2 *[ビットコインのスケーラビリティがない] $D \cdot \lambda$の増加に従い、ビットコインプロトコルのセキュリティ閾値は0になる。*

この理論の証拠はこれまでに様々な形で提供されている。[18], [15], [7]を参照されたい。高いセキュリティ閾値を維持するために、ビットコインではλを1秒あたり1/600ブロックと低く維持している。λ（及びB）はプロトコル開始時点で完全に決定されてしまうためにこの大きな安全マージンが必要となる。結果的に、ネットワークの状態が健全であり、Dが低い場合でもビットコインは1秒あたり3~7トランザクションという低いスループットと数十分という遅い確認時間に苦しむことになる。対照的に、SPECTREのスループットはセキュリティ閾値を低下させることなく増加させることが可能である。

理論3 *[SPECTREにスケーラビリティがある] あらゆる$D \cdot \lambda$ で SPECTREのセキュリティ閾値は50%である。*

そのため、分散アルゴリズムにおいてSPECTREはDのあらゆる値でセキュアな状態を維持するために部分同期化設定に分類される。付属文書Eでは理論3で証明している。もちろん、λを無限に増加させるとネットワークにメッセージ（ブロック）が殺到し、渋滞してしまう。理論3は理論上の枠組み（セクション2で指定）内で生きており、ノードの帯域幅やネットワーク容量の制限のモデルを作成していない。実務的にはこのような障壁によって$\lambda = 10$や$b = 100$を設定することで1秒あたり数千トランザクションのスループットが可能になる。より詳細な説明に関しては、付属文書BとDをご参照ください。

漸進的にSPECTREの確認時間は$\mathcal{O}\left(\frac{\ln(1/\epsilon)}{\lambda(1-2\alpha)} + \frac{D}{1-2\alpha}\right)$である。実務上、これによって通常のネットワーク条件下でわずか数秒という確認時間が可能になる。*RobustTxO*を実行する場合、SPECTREの各ノードでネットワーク内の最近のDに対して独自の上限を使用する。この限度はそのノード自身の運用にのみ影響を与える。Dを理解することによってトランザクションの早急な受諾が発生する可能性があり、過大評価を行うことで受諾が必要以上に遅延する可能性がある（時間差）。重要なことは、ネットワーク障害やネットワーク遅延が発生した場合、ノードは他のノードとの調整を行うことなくローカルクライアント内でDに対するより控え目な限度値に切り替えることができる、という点である。

4. SPECTREプロトコル

A. ブロックDAGの生成

ビットコイン同様に参加するノード (マイナーと呼ばれる)はPoWパズルを解決することでトランザクションのブロックを作成する。ブロックはヘッダー内で前のブロックのIDを参照することでその前のブロックを指定する (ヘッダーに衝突耐性のあるハッシュを適用することでブロックのIDを入手する)。次のサブセクションではこのような前のブロックの選定方法について説明する。これによって一般的に$G = (C, E)$と表記されるブロックの有向非巡回グラフ（DAG）構造が発生する（ブロックは自身の前に作成されたブロックしか参照できない）。ここでは、Cはブロックを表し、Eはハッシュリファレンスを表す。頻繁に$z \in C$の代わりに$z \in G$を使用する。$past(z, G) \subset C$はzから到達可能なブロックのサブセットを表し、同様に

$future(z, G) \subset C$ はz に到達可能なブロックのサブセットを表す。これらのブロックは恐らくzの前後に作成されたブロックである。DAG内のエッジは新しいブロックから以前に作成されたブロックと過去にさかのぼる点に注意が必要である。ノードは過去のセットを全て受領するまでブロックを有効とみなさない。$cone(z, G)$ はz: $cone(z, G) := past(z, G) \cup \{z\} \cup future(z, G)$ に関してDAGが直接命令するブロックのセットであり、$anticone(z)$ は$cone(z, G)$を捕捉するものである。セット$past(b, G)$ はbの作成時に完全に固定される（DAGにブロックが後から追加された時点で成長する $future(z, G)$ と$anticone(z, G)$とは対照的である）。そのため、状況を考慮することなく単純に$past(b)$ を使用できる。
固有のブロック*genesis* はシステム開始時に作成されるブロックであり、全ての有効なブロックには過去のセットが必要である。さらに、仮想ブロック*virtual*(*G*)も使用する。このブロックは$past(virtual(G)) = G$.の条件を満たす。その役割は単純に方法論的なものであるが、*virtual*(*G*)は現在観察されるDAGが*G*であるノードが作成を試みる次のブロックを表すと考えることも可能である。
G_t^v はtの時点でノード$v \in \mathcal{N}$ が観察するブロックDAGを表す。このDAGではそのノードが受領した全ての（有効な）ブロックメッセージ履歴を表し、セクション2で想定する抽象的データ構造を具体的に示している。

B. マイニングプロトコル

SPECTREのマイナーに対する指示は極めてシンプルである。

1) ブロックを作成もしくは受領する場合には全てのピアにブロックを送信する。

2) ブロックを作成する場合、そのヘッダーにローカルで観察されるDAGの全てのリーフブロック（0°内のブロック）のハッシュを含むリストを含める。

このような指示によってマイナーはブロックの内容の衝突に関係なく独立して動作することが可能となる。

C. TxOプロトコル

概要

ブロックDAGには衝突するトランザクションが含まれる可能性があるため、ノードがDAGを解釈し、そこから受諾されたトランザクションのセットを抽出するための方法を提供する必要がある。全てのノード（最終的）が同意する方法でこれを実施することがSPECTREの主要な課題である。
その方法について説明する。
ブロックDAG *G* のトポロジーにはブロックに対する自然な優先順位関係が含まれている。*y*から*x*に到達できる場合(例：$x \in past(y)$)、*x* は *y*より前に作成されている可能性が高いため、*x*は*y*に対して優先される。
SPECTREではこの関係を≺と表記される*G*のブロックの関係全体に適用している。 この順位は*G*:のトランザクションの順位に変換できる。 tx_1を含むブロックがtx_2を含むブロックより前に来る場合にはtx_1がtx_2に対して優先される。この関係には受諾されるト

ランザクションの自然のサブセットが含まれる。*tx*は*G*内の全ての衝突するトランザクションに対して優先される場合に受諾される。関係≺は全てのブロックのペアで独立して発生するペア投票手順によって発生する。このレイヤの運用については次のサブセクションで説明する。

時として≺がブロックの順位を決定するように言及する場合があるかもしれないが、≺は必ずしも推移関係ではないことを強調しておく。お互いに対して循環的に優先される一連のブロックを持つことは可能である[2]。SPECTREではブロックに対する完全な直線的順位決定を行わないことでこの枠組みの弱いコンセンサス要件を活用している。線形順序はコンセンサス問題の解決に相当する[3]。

ブロックのペアによる順序決定

SPECTREの基本レイヤではブロックDAGのペア順序決定を行う。2つのブロック $x, y \in G$ を固定する。$x \prec y$ もしくは $y \prec x$ であるかどうかを判断する場合、DAGの構造が抽象的投票を表すものとして解釈する。全てのブロック $z \in G$ はペア (x, y) に関してボーターとみなされ、その投票はDAGの構造から推測される。$\{-1, 0, +1\}$ の番号で投票を表現し、z の全てのペア上での投票プロフィールを $vote(z, G)$ で表す。$vote_{x,y}(z, G) = -1$ は x が y $(x \prec y)$ に対して優先されること、$vote_{x,y}(z, G) = +1$ は y が x に対して優先されること、そして $vote_{x,y}(z, G) = 0$ はタイであることを意味する。重要なことは、$vote(z, G)$ が非対称関係であるという点である：$vote_{y,x}(z, G) = -vote_{x,y}(z, G)$。

表記を簡素化するために、投票に*virtual* (*G*) も関連付けている。*G*の仮想ブロックは*past* (*virtual* (*G*)) = *G*の条件を満たす仮想ブロックである。*virtual* (*G*) の投票はDAGブロック全体の合計票数を表す。

$z \in G \cup \{virtual(G)\}$ の z の投票の基本的なルールは以下の通りである。

1) $z \in G$ が*future* (*x*) 内にはあるが、*future* (*y*) 内にはない場合、x に好意的に投票する（例：$x \prec y$）。

2) $z \in G$ が $future(x) \cap future(y)$ 内にある場合、z の投票は過去のDAGに従って再帰的に決定される（例：*virtual* (*past* (*z*)と同じ投票）。この投票結果がタイである場合、*z* の投票によって決定される[3]。

3) $z \in G$ がいずれのブロックの未来にもない場合、自身の未来のブロックの過半数の投票と同じ投票を行う。

4) z が*G*の仮想ブロックである場合、*G*内のブロックの過半数と同じ投票を行う。

5) 最後に(*z*が*x*もしくは*y*に等しい場合)、*z* は 自身に投票して*past* (*z*)内の全てのブロックの後に入り、*past* (*z*)外の全てのブロックの前に入る。

[2]これは社会的選択におけるコンドルセのパラドックスに関係する [2]。

[3]*z*のヘッダーでエンコードされる情報（例：タイブレーカー用の明確な手順）を使用する、もしくはタイ状態のブロックの（ハッシュの）辞書式順序を使用することが可能である。

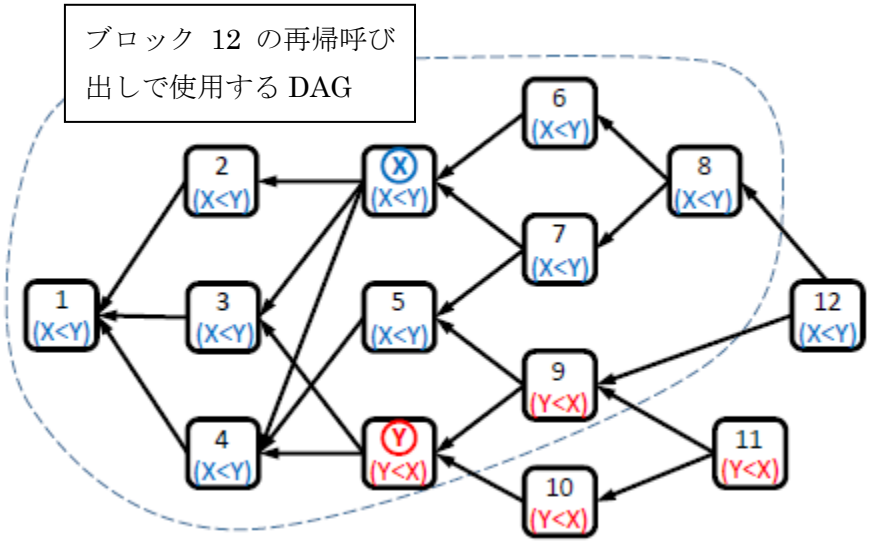

図1: シンプルなDAGの投票手順の例

ブロックx とブロック 6-8 は過去にxがあるがyがないためvote $x \prec y$ と投票する。同様に、ブロックy とブロック9-11は$y \prec x$と投票する。ブロック12はブロック10,11,12を含まないDAG上の再帰呼び出しに従って投票する。1-5のあらゆるブロックは未来において$y \prec x$よりも$x \prec y$投票のほうが多いため $x \prec y$と投票する。

直観的に、最初のルールでは誠実に公開されたブロックは秘密とされていたブロックよりも多くの投票を獲得することを定めている。これは誠実なノードは未来のセットに対して新しいブロックを追加し続けるためである。

2番目そして4番目のルールでは新規ブロックによって以前の決定に従い、以前の決定を強化する投票を追加するために過半数の増幅が保証される。3番目のルールが最も繊細なルールである。このルールでは($future\,(x)$に加えて)$past\,(x)$のブロックに対してyが長期間保留されていた場合には自身に投票とすることを許可している。これはマイニング前の攻撃スキームに対抗するために必要である。これについては今後のセクションで説明する。全ての投票はDAGのトポロジーを尊重している。yからxに到達できる場合、全てのブロックが満場一致で$x \prec y$と投票する。

図1では1つのブロックペア(x,y)に関する投票手順を説明している。付属文書Aでは追加の例とこのキーアルゴリズムに関する情報を提供している。

以下のアルゴリズム1では投票手順を実施している。このアルゴリズムでは$n < 0$ の場合は$\widetilde{sgn}\,(n) = -1$、$n > 0$ の場合は$\widetilde{sgn}\,(n) = +1$、そして$\widetilde{sgn}\,(0) = 0$である。4行目の再帰呼び出しに関して、$G$($past\,(z) \subsetneq G$ であるため)より小さいDAGをインプットとして採用しているため、最終的にベースケース$G = \varnothing$ にたどり着いた後に戻っていく。読みやすさのためにアルゴリズムを天然型で表記しており、ランタイムは$\mathcal{O}(|G|^3)$である。この手順のより洗練された実装形式もデザインしており、この

場合のランタイムは$\mathcal{O}(d \cdot \lambda)$である。コードのフルバージョンをオンラインで公開する。
SPECTREのペア順序決定には以下のような貴重な特性がある。

特性4

ブロックが公開されると、高い確率で、ペア順序でその前の順位であるブロックのセットが最初に閉じられる。直前もしくは直後に公開されたブロックによってのみ構成される。

このトランザクションのセキュリティに対する保証は少なくとも直観レベルにおいて即時的な効果を持つ。公開済みのブロック*x*でトランザクションが組み込まれたユーザは*x*'の公開後から受諾するまでしばらく待機することによってその安全性を保証できる。このユーザは後に衝突するトランザクションを含む可能性のある後に公開されるあらゆるブロックに対して*x*が優先されるため、彼のトランザクションの受諾が及び課されることはない。セクション5ではこの保証を達成する方法について説明する。

アルゴリズム1 *CalcV otes*

インプット: *G* – ブロックDAG

アウトプット: *vote* (*virtual* (*G*)) – *G*内のブロックのペア順序決定

1: *G* = ∅の場合
2: 空白の順序を返す
3: 全ての$z \in G$
4: *vote* (*z; past* (*z*)) *CalcV otes* (*past* (*z*)) そして任意にタイブレークを行う
5: 一部のトポロジー順序内の全ての$z \in G$ （リーフからルート）
6: 全ての$x, y \in G \ (x \neq y)$
7: $\left(x \in \overline{past}\,(z) \wedge y \notin past\,(z)\right) \vee \left(x \in past\,(z), y = z\right)$ の場合、
8: $vote_{x,y}\,(z, G) \leftarrow -1$
9: $\left(y \in \overline{past}\,(z) \wedge x \notin past\,(z)\right) \vee \left(y \in past\,(z), x = z\right)$ の場合、
10: $vote_{x,y}\,(z, G) \leftarrow +1$
11: $x, y \in past\,(z)$ の場合、
12: $vote_{x,y}\,(z, G) \leftarrow vote_{x,y}\,(z, past\,(z))$
13: $x, y \notin past\,(z)$ の場合、
14: $vote_{x,y}\,(z, G) \leftarrow \widetilde{sgn}\left(\sum_{z' \in future(z,G)} vote_{x,y}\,(z', G)\right)$
15: $vote\,(virtual\,(G), G) \leftarrow \widetilde{sgn}\left(\sum_{z \in G} vote\,(z, G)\right)$
16: $vote\,(virtual\,(G), G)$ を返す

トランザクションの受け入れ

ブロックに対するペア関係を実現した我々は次に受け入れられるトランザクションのセットの構築に目を向けた。整合性を維持するために、我々は以下の3つの条件が全て満たされた場合にトランザクションに受け入れられたものとしてマーキングを行って

いる。
1) 全てのインプットが受け入れられている。
2) トポグラフ的に関係のないアンチコーンセットの全ての衝突するトランザクションが当該トランザクションを含むブロックの後にあるブロック内に集中している。
3) 過去のセットの全ての衝突するトランザクション（トポグラフ的にDAG内で前に存在するトランザクション）が拒絶されている。
アルゴリズム2ではこれらのルールを実施しており、受け入れられたトランザクションセットを出力している。再帰演算を行い、最初に$TxO(G,G)$によって呼び出すべきである（今後はシンプルに$TxO(G)$と表記する）。このアルゴリズム内では$Z_G(tx)$は*G*内でtxを含む全てのブロックを意味する。
DAG内に同じトランザクションの複数のコピーが発生する可能性があるため複雑な事態が発生する。$[tx]$はtxの全てのコピーを含む同等のクラスを意味する。
SPECTREプロトコルの3つ目の構成要素である*RobustTxO*手順については付属文書Cで説明する。

5. プルーフの高レベル概要

次にSPECTREの手順によってトランザクションが安全に受け入れられる、そして誠実なユーザの全てのトランザクションが迅速に受け入れられる理由について説明する。特性4の証明を目指す。上述のように、この特性はトランザクションの必要なセキュリティ特性に変換しやすい（付属文書Eでこれを正式に行っている）。

アルゴリズム2 *TxO*

インプット: *G* – ブロックDAG, *subG* – （仮想）ブロックの過去である*G*のサブDAG
アウトプット: *Tx* – *G*内の有効なトランザクションのハイパーセット

1: *vote* (*virtual* (*G*)) ← *CalcVotes*(*G*)
2: $Tx \leftarrow \emptyset$
3: 全ての $z_1 \in subG$ に対して
4: 全ての $tx \in z_1$ に対して
5: 全ての $tx_2 \in G \cap conflict\,(tx)$ に対して
6: 全ての $z_2 \in Z_G(tx_2) \cap anticone\,(z_1, G)$ に対して
7: $vote_{z_1,z_2}\,(virtual\,(G)) \geq 0$ の場合、
8: 改行 (4行目に改行して次の*tx*をピックアップ)
9: $[tx_2] \cap TxO(G, past\,(z_1)) \neq \emptyset$ の場合、
10: 改行 (4行目に改行して次の*tx*をピックアップ)
11: 全ての $[tx_3] \in inputs\,(tx)$ に対して
12: $[tx_3] \cap TxO\,(G, past\,(z_1)) = \emptyset$ の場合、

13: 改行 (4行目に改行して次の*tx*をピックアップ)

14: *Tx*に*tx*を追加

15: *Tx*を返す

具体的に、我々は以下の命題の証明を目指している (提案内では $G_r^{pub} := \cup_{u \in honest} G_r^u$)

提案

$t_{pub}\,(x \in G_{t_{pub}}^{pub})$ *の時点でブロックxが公開され、yが* $t_{acc}\,(y \notin G_{t_{acc}}^{pub})$ *以前に公開されていないものと想定する。*[4]

$T = t_{acc} - t_{pub}$ *である。この場合、xが必ずしもyに対して優先されない確率* $(\Pr\left(\exists u \in honest, \exists s \geq t_{acc} : vote_{x,y}\left(virtual\left(G_s^u\right)\right) \geq 0\right)$ *）はTにおいて急激に低下する。*

プルーフの概要

将来の何らかのDAGにおいて*y*が*x*の前に来る場合を想定する。 s は何らかのノードでこの事象が発生する最速のタイミングとする。*y*は*x*の過去、もしくは未来に存在することはできない（そうでなければ順序がトポロジーによって決定され、これを戻すことができない）。そのため、 $y \in anticone\,(x)$ と仮定する。

***x*の公開後のブロックレース**

最初にブロックの公開後に作成されたブロック*x*の投票を考慮する。

- t_{pub} と t_{acc} の間で作成された（ほぼ）全ての誠実なブロックは過去に*x*があるが*y*はないため永久的に $x \prec y$ に好意的に投票する。 n_1 はそのようなブロックの数である。
- t_{acc} と s の間で作成される全ての誠実なブロックも*s*の選択によって $x \prec y$ に好意的に投票する。 n_2 はそのようなブロックの数である。
- m_1 と m_2 は n_1 と n_2 に対応する時間間隔で攻撃者が作成したブロックの数である。誠実なノードは計算能力の $1 - \alpha > \alpha$ を保有する。

[4] t_{acc} はノードがブロック*x*内に現れるトランザクションを受け入れた時間を表す。

結果として、あらゆる正の定数*C*で n_1 と共に $m_1 + m_2 + \quad C - (n_1 + n_2) \geq 0$ という関係が満たされる可能性が急速に減少する。これは*C*で始まる整数上でのバイアスのかかったランダムウォークが起点に帰ってくる確率として分析される ([13], [17], [18]を参照)。

$m_1 + m_2 - (n_1 + n_2)$ は*x*の公開後に作成されたブロックのみを考慮した*x*と*y*の間

の投票総数を意味する。攻撃者が「プレマイニング」の準備段階で*x*の公開前に事前に作成したブロックが一部の一定の優位性以上の利点を与えていないことを示す（この優位性は上の*C*において計算されている）。

プレマイニング段階

*x*の公開前に作成された誠実なブロックは通常は過去にあり（一部の小さなブロックのセットは除く）、未来における投票の過半数によって投票を決定される（アルゴリズム1に従う）。そのため、これらの投票はDAGの成長や攻撃者のブロックの公開によって変化する可能性がある。

そのため、*x*の過去の全てのブロック*z*に関してそのブロックの上で*x*に好意的に投票するブロック数、そして*x*に反対投票を行うブロック数を考慮する必要がある。X_zはt_{pub}時点までの*z*の未来における攻撃者ブロック数と誠実なブロック数の差である。補題24では最悪の場合の差 X_z(全てのブロック $z \in past(x)$) を起点に向かって偏った負でない整数のランダムウォークを反映するものとしてモデリングできることを示した。結果として、攻撃者が $past(x)$ 内のブロックで秘密裏に取得できる最高の差は指数関数的に減少し、高い確率で特に定数によって制限される。

まとめると、$t_{acc} - t_{pub}$ が増加するにつれて、投票数 n_1 、あるいは*x*が受領する「確認」は線形的に増加し、攻撃者が $z \in past(x)$ の未来の $y \prec x$ 投票数を増やすに十分なブロックを公開できる確率は n_1 と共に急激に低下する。これは全ての $z \in past(x)$ に等しくあてはまるため、特にブロック*genesis*は未来において$y \prec x$投票よりも$x \prec y$投票が多くなることを示唆している (非常に発生確率の低い事象が発生した場合は除く)。仮想ブロックの投票は*genesis*ブロックの投票によって決定され（これは簡単に確認可能であり、補題13で証明済みである）、引数を完成させる。上記の提案は補題 14と15の要旨である。上記スケッチでは追加の詳細説明を行っている。例えば、t_{pub}, t_{acc}もしくは*s*の*D*秒前後に作成された誠実なブロックは*x*に好意的に投票していない可能性がある。我々の正式な分析（付属文書E）において、最悪のケースを想定してこれらを攻撃者のブロックとしてカウントし、上述の定数*C*に追加している。さらに、攻撃者が受諾を遅らせるためにブロックを公開する場合にユーザが n_1 を正確に測定する方法についても説明している。

6. 関連する研究

以前の研究ではセキュリティとスケーラビリティの両立の課題の解決を試みたプロトコルを複数提案しているが、全てのプロトコルでブロックに対する全順序を提案している。

GHOSTは1つのチェーンに収束するまでブロックのツリーを漸進的に選択する代替チェーン選択ルールである [18]。[8]で示すようにGHOSTの生存性特性が攻撃される可能性を示している。Inclusive [10]ではブロックDAGの使用が提案されており、チェーン外ブロックを元帳に統合することでスループットを増加させている。

Inclusiveはチェーンに依存するために、セキュリティとスケーラビリティの問題を緩和はしているが回避はしていない。さらに、Inclusive論文ではノードが（調整能力なしに）自身のブロックに異なるトランザクションを組み込むインセンティブのゲーム理論分析を行っている。[5]

Bitcoin-NG [6] ではPoWを要求するがトランザクションを含まないキーブロックとPoWを要求しないが、トランザクションを含むミニブロックという2つのタイプのブロックで構成されるクレバーなチェーン構造を採用している。Bitcoin-NGではスケーラビリティを大幅に向上させているが、キーブロックの生成はまだ遅く、依然として確認時間が長くかかっている。PoWを独自で行うことで従来のBFTプロトコルを実行するために後で使用する委員会のインスタンスを生成している

この研究の例としてはByzcoin [9]、Deckerらによる研究[4]、Hybrid Consensus [16]そして最近発表されたSolidus [1]が挙げられる。この方法で構築したプロトコルはコンセンサスプロトコルに基づいているためにスケーラビリティが高いが、ビットコインが実現する一部の特性が欠如している。

これらでは大規模な委員会を必要とし、委員会メンバーは長期間オンライン状態を維持する必要があり、ネットワーク隔離やDoS攻撃を受けやすくなる[9], [4],[1]。 また、委員会の大多数が悪意のある集団で構成される場合には失敗し、回復の手段がない(一方でビットコインには自己安定性がある)。さらに、前方秘匿性も必要である。過去のある時点の委員会の十分な割合の暗号鍵が危険にさらされた場合、攻撃者は同様に受け入れられる可能性のあるバージョンの事象を作成できる。

Algorandプロトコル[11]は通貨の所有自体を用いてスケーラブルなコンセンサスプロトコルを実現するプルーフオブステークに基づくアルゴリズムである。追加のテクニック（VRFに基づく）を活用してコンセンサスプロトコル内で発生する委員会メンバーを隠す。対照的に、SPECTREのマイナーは明示的なコンセンサスプロトコルに直接的に関与することはない上に他のノードの同期化状態を気にすることなく運用を行うことができる。Honey Badger [12] はネットワークパラメータに依存せず、様々なネットワーク条件下でのチューニングを必要としない原子ブロードキャストプロトコルである（SPECTREと同様）。参加者のIDが判明している従来の許可環境を用いている。

7. 結論

本論文において我々は本質的にスケーラビリティの高い新しい暗号通貨プロトコルであるSPECTREを紹介する。

ビットコイン及びその派生品と異なり、SPECTREはスループットが上昇し、電波遅延が無視できない水準になった場合にも計算能力50%以下の攻撃者に対してセキュアな環境を維持する。我々の結果では、SPECTREは特にナカモトコンセンサスと比較して非常に短い確認時間を達成している。我々が導き出した受諾ポリシーを改善し厳格化するための追加の研究によってさらに確認時間を短縮できる可能性がある。SPECTREの実績の鍵は明確に二重支払いが行われているトランザクションに関する決定を遅延できることである。そのため、従来のコンセンサスプロトコルよりも弱り問題を解決できる。また、この事実はトランザクションの全順序を必要とするEthereumのようなシステムにとってはSPECTREが不適切であることも示している。

SPECTREの中心的なアルゴリズムであるペア投票手順（アルゴリズム1）は重要である。その運用の説明に関しては付属文書Aを参照していただきたい。

[5]我々はこの議論に基づき、ノードがトランザクションの「衝突」を回避し、自身のブロック内に独自の内容を組み込むことで利益を最大化しようとする、という前提に立っている。

参考文献

[1] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. *arXiv preprint arXiv:1612.02916*, 2016.
[2] Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. *Handbook of Social Choice & Welfare*, volume 2. Elsevier, 2010.
[3] Miguel Correia, Nuno Ferreira Neves, and Paulo Ver´ıssimo. From consensus to atomic broadcast: Time-free byzantine-resistant protocols without signatures. *The Computer Journal*, 49(1):82–96, 2006.
[4] Christian Decker, Jochen Seidel, and Roger Wattenhofer. Bitcoin meets strong consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, page 13. ACM, 2016.
[5] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In *13th IEEE International Conference on Peer-to-Peer Computing (P2P), Trento, Italy*, September 2013.
[6] Ittay Eyal, Adem Efe Gencer, Emin G¨un Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 45–59, 2016.
[7] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.
[8] Aggelos Kiayias and Giorgos Panagiotakos. On trees, chains and fast transactions in the blockchain. Cryptology ePrint Archive, Report 2016/545, 2016.
[9] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 279–296, 2016.
[10] Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar. Inclusive block chain protocols. In *International Conference on Financial Cryptography and Data Security*, pages 528–547. Springer, 2015.
[11] Silvio Micali. Algorand: the efficient and democratic ledger. *arXiv preprint arXiv:1607.01341*, 2016.
[12] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 31–42. ACM, 2016.
[13] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
[14] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. *IACR Cryptology ePrint Archive*, 2016:454, 2016.
[15] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. *IACR Cryptology ePrint Archive*, 2016:454, 2016.
[16] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. Cryptology ePrint Archive, Report 2016/917, 2016.
[17] Meni Rosenfeld. Analysis of hashrate-based double spending. *arXiv preprint arXiv:1402.2009*, 2014.
[18] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527. Springer, 2015.

付属文書A

直観的知識及び具体例

このセクションではSPECTREの運用に関する基本的な説明と直感的知識を提供する。本プロトコルの中核をなすアルゴリズム1の基本となる考え方について主に説明する。その後、耐久性の実現方法を説明する目的でシンプルな攻撃を例に挙げて説明する。

直観的知識 1 (可視ブロックに好意的に投票)

誠実な参加者がブロック*x* を知っている場合、彼らのブロックの過去にこのブロックが含まれる。ブロックは（未知のブロックよりも）自身の過去のブロックに好意的に投票し、誠実なノードはすぐにブロックを公開するため、隠された攻撃者のブロックは投票に敗北する。

直観的知識 2 (過半数増幅)

衝突の可能性があるブロック*x*、*y* が存在する場合、これらの公開後に誠実な参加者が生成するブロックはDAG内でこれら両方を参照する。アルゴリズム1に従い、これらの新しいブロックは*x*と*y*に関して過去のサブDAGの投票を採用する。そのため、ブロック*x*がブロック*y*の前にある場合、この決定を支持する投票が追加され、攻撃者が投票を無効にすることが困難になる。

直観的知識 3 (最近のブロックの参照が効果的)

過去のブロックは未来に従って投票する。そのため、攻撃者が最近のブロックを参照しないブロックを作成する場合、最近のブロックを参照する他のブロックと比較して不利になる（参照しない最近のブロックの投票が得られず、「説得力」がない）。

直観的知識 4 (過去からの投票によってプレマイニング攻撃に対抗)

ブロック*y*を作成し、これを隠し、長期にわたってこのブロック上に多くのブロックを作成する攻撃者を例に挙げて考えてみる。

長期間経過後にネットワーク上に衝突するトランザクションが公開され、ブロック*x*内に追加される。ブロック*y*にはこれを参照する多くのブロックがある（攻撃者が作成）。そのため、未来からの投票だけをカウントすると、*x*が一定数の投票を獲得できるとしてもブロック*y*が勝利する。SPECTREでは、*y*が隠されている間に誠実なノードが作成したブロックは未来からの投票に期待する。これらは通常は*x*に好意的に投票し、*y*が隠されている間に攻撃者が作成したブロックの数を通常は上回る（図3ではプレマイニングの例を示している）。

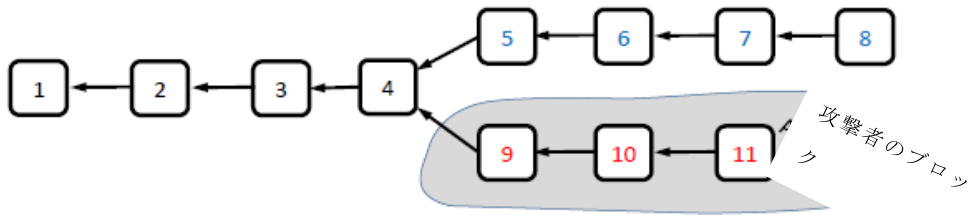

図2: SPECTREは「シンプルな」ブロックのチェーンに適用される最長チェーンルールに一致している。図で示すDAGではブロック8で終了するチェーンの方が長いため、最長チェーンプロトコル内で選択される。SPECTREでは each one of the blocks ブロッ

ク5,6,7,8のそれぞれがブロック9,10,11に対して先行する。ブロック6と10が関係するペア投票の例について考えてみる。ブロック6-8は過去にブロック6が存在するが、10は存在しないため、強く 6 ≺ 10と投票する。ブロック5は過去にブロック6と10がない弱い投票者であり、その未来の過半数と同じように投票する (そのため、6 ≺ 10 と投票)。同様の理由からブロック 9-11も全て 10 ≺ 6と投票する。2つのチェーンの分岐点にあるブロック4も過去に6も10も存在しない弱い投票者であり、未来のブロックの過半数に従って投票する。ブロック4では4票が 6 ≺ 10、3票が10 ≺ 6に投票するため、 6 ≺ 10に投票する。同様に、ブロック1-3も未来のブロックに従って投票し、 6 ≺ 10に多数が投票され、結果に自身の投票を追加する。そのため、最終結果は 6が 10に対して優先される。

A. 最長チェーンとの同等性

2本のチェーン間の「シンプルな」分岐においてSPECTREがビットコインの最長チェーンルールとどのように一致しているのかを説明する。図2で示すDAGを考えてみよう。ビットコインではより長いチェーンが選択される。同様に、SPECTREのペア順序では最長チェーン5,6,7,8内の各ブロックが 短いチェーン内の各ブロック9,10,11に対して優先される。

これがあてはまる理由については図の注釈を参照していただきたい。

次に我々が二重支払いと検閲と呼ぶ2つの異なる攻撃シナリオを検証する。我々のマイナープロトコルの要件を思い出してみよう: 各マイナーは *(i)* 最近のブロックを参照し、*(ii)* 自身のブロックをすぐに公開する必要がある。各攻撃は基本的にこれらいずれかの要件の違反である。二重支払い攻撃では攻撃者はブロックのセット（衝突するトランザクションを含む）の公開を遅らせ、検閲攻撃ではブロックを公開するが、特定のブロックとその内部のトランザクションを「無視」し、ノードに対して十分な投票を獲得できなかったために受け入れられなかったと思わせようとする。

B. 二重支払い攻撃の例

図3では（失敗した）二重支払い攻撃を示している。この攻撃は以下の3つのフェーズで構成されます。

フェーズI: プレマイニング

フェーズIでは攻撃者はブロックの作成を開始し、作成したブロックをネットワークから秘匿している。作成した最初のブロック (ブロック *y*) には誠実なノードに送信されたトランザクションと後で衝突するトランザクションが含まれている。攻撃者が作成したブロックはチェーンを構成し、SPECTREの投票ルールのために全て *y* ≺ *x* と投票する (ブロック *y*,13,14)。

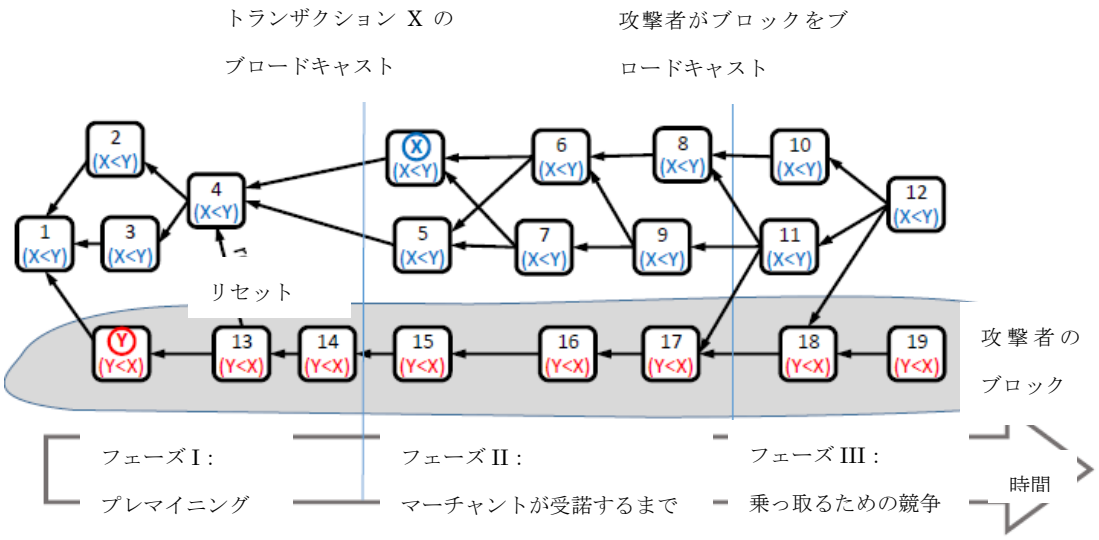

図3: DAGの投票手順で二重支 敗）

ブロックx とブロック6-8は過去にxだけがありyがないため、強く x ≺ y と投票する。同様に、ブロックy とブロック13-19 は強く y ≺ xと投票する。ブロック11の過去であるDAG ではブロック1-5のそれぞれで未来においてy ≺ x投票よりもx ≺ y投票が多くなるため、それぞれがx ≺ yと投票する。ブロック 11 は（過去投票の仮想ブロックとして）過去の過半数に従って投票するため、同様にs x ≺ yと投票する。11と12でも同様の議論が成り立つ。最後に、DAGの全ての投票を集計することでxが多くの投票を獲得し、x ≺ yとなる。

誠実なノードが作成したブロックは*y*(そしてまだ作成されていない*x*)を把握しておらず、未来の投票の過半数に従って最終的に投票する。このフェーズでは攻撃者ブロックは作成済みの誠実なブロックを参照する (後に*y* ≺ *x*と投票させるため)。一定時間経過後、攻撃者はトランザクションをネットワークに送信し、フェーズIIに進む。フェーズI終了時点で攻撃者は誠実なノードよりもブロック4上に多くのブロックを持っており、有利なスタートとなっている。ブロック4をより簡単に*y* ≺ *x*に投票させることができる(誠実なノードは基本的に攻撃者よりも迅速にブロックを作成するためにこの利点は後でなくなる)。

フェーズII: 受諾を待つ

攻撃者は秘密でブロックの作成を継続する。攻撃者がブロックを公開すると、衝突するトランザクションを誰もが見ることができるようになり、二重支払いが発覚してしまう。そのため、攻撃者はブロック*x* が十分な重量（このブロック上に作成されたブロックという形）を持ち、*x*のトランザクションの受領者がこれを受け入れ、攻撃者にサービスもしくは製品を提供するまで待つ。このフェーズ中に攻撃者は作成済みの攻撃者ブロック (ブロック15-17)に対して注意深く秘密のチェーンのみを参照させ、ブロック*x*を決して間接的に参照させないため、作成済みのブロックは*y* ≺ *x*と投票する。このフェーズ中に作成された誠実なブロックは*y*が隠されているために基本的に*x* ≺ *y*と投票する。

一部の少数のブロック (*x*がネットワーク全体に拡散される前に作成したブロック
– この例ではブロック5) は*x*を参照しないため、未来の投票の結果に従って投票する。

フェーズ III: 乗っ取るための競争

被害者が*x* を受け入れると、攻撃者は*y*内の衝突するトランザクションが*x*に対して優先されることを狙って秘密のブロックを公開する。この場合、*x*のトランザクションは拒絶されたものとみなされ、支払いはキャンセルされる（攻撃者は支払いを行っていない品目を入手する）。攻撃者は秘密のチェーン（この時点から誠実なノードもこのチェーンを参照する）を公開し、この上で作成を継続する。彼が作成するブロックはここでも*x*を参照しないため、$y \prec x$と投票し彼の目的を支援する。新しい誠実なノードは初めて衝突するトランザクション*y*を把握するため、過去のサブDAGの結果に従って投票する。

攻撃が失敗する理由

最初に、上記の例の攻撃者は各フェーズで誠実なノードよりも作成しているブロック数が少ない。攻撃者は全ての誠実なノードよりも計算能力が低いためにこれがあてはまる場合が多い。 攻撃者がブロック作成において「ポアゾンバースト」を行うことでネットワークを乗っ取ることは可能であるが、攻撃を長期化継続する場合にはこの可能性は低い。防御者はトランザクション受け入れの待機時間を長くし、このようなバーストの確率を下げることでフェーズIIの長さを制御することができる。

フェーズIIが十分長くなると、この期間の*x* の投票数は*y*よりも多くなる。*x*の過去の弱いブロックはこの過半数に従って*x*に投票する。未来に依存するブロックはカスケードを開始する。過去の各ブロックが未来のブロックの過半数と同意する

投票を追加することで決定を強化する。フェーズIIで得られた過半数が大きければ大きいほど、攻撃者がフェーズIIIで追いつける可能性が低くなる。そのため、攻撃は*x*の直前に作成されたブロックの投票の獲得に成功できるかどうかに大きく依存する(例：ブロック4)。

誠実なネットワークが予想するよりも多くのブロックを作成する攻撃者はこの攻撃に成功する、という点が重要である。$y \prec x$と投票するブロックは反対の投票を行うブロックよりも数が多くなる。そのため、理論3の50%基準が必要となる。

C. 検閲攻撃の例

図4では（失敗した）検閲攻撃を示している。この攻撃では1つのメインのフェーズ中に攻撃者がブロックを作成し、これをすぐに公開するが、誠実なネットワークが作成する最近のブロックを無視している（そしてこれを参照しない）。図（左のステージI）ではブロックチェーン（この時点で全てのブロックが公開されている）の最新状態を示している。

ネットワークを観察し、ブロック*x*のトランザクションがセキュアであるかどうあｋを確認することを希望する誠実な参加者は*x*を参照しない多数のブロックを確認できる。これらのブロックは*x*に投票することを保証されていない。攻撃者はその後、衝突するトランザクション*y*を挿入してその上にブロックを追加する可能性がある（この予想される攻撃について図の右側で説明している）。これによって以前に作成した攻撃者ブ

ロックに*x*に反対投票を行うよう説得できる可能性がある。

検閲攻撃の主要なリスクは攻撃者のブロックを見たマーチャントがブロック*x*のトランザクションが十分セキュアでないと判断する可能性である。これによってトランザクションの受諾が永久的に遅延する可能性がある。我々のSPECTREの分析ではこの場合にもマーチャントはトランザクションを迅速に（そしてセキュアに）受諾していることが証明されている。

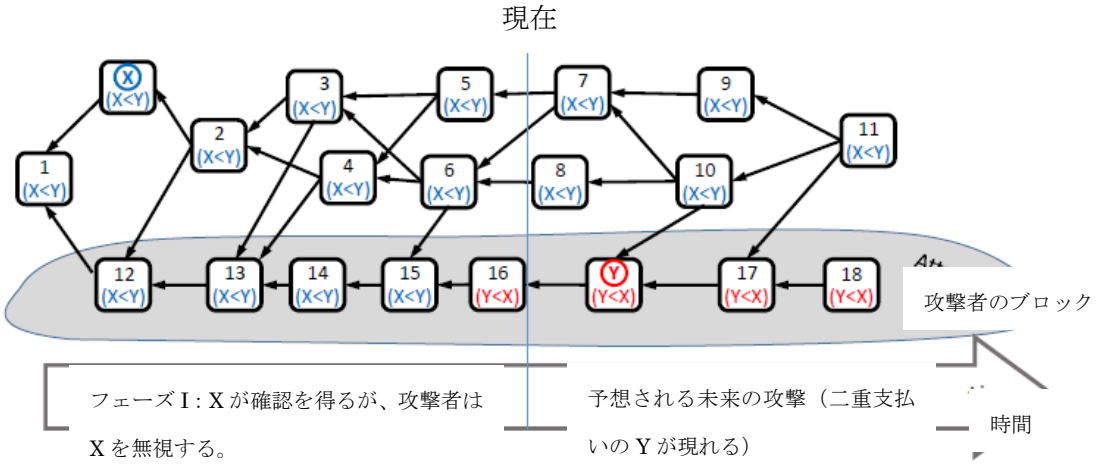

図4: DAGの投票手順において検閲攻撃が失敗した例

図の左側ではブロックDAGの現状を示している。右側では可能性の高い未来の展開を示している。ブロック12-16はxに強い投票を追加しない。これらに対してブロックyが出現した場合にブロックyに対して投票するよう説得することは可能か？過去のその他のブロックをさらに説得するのか？この予想される未来の各ブロックについて記載する。ブロック2-9は過去にx（ただし、yはない）が存在するためxに強く投票する。ブロック17-18も同様にyに投票する。ブロック16の未来の多くのブロックがxよりもyに多く投票するためyに投票する。ブロック1, 12-15はxに投票する。これらのブロックは未来においてyよりもyに対して多く投票している。ブロック10-11は再帰呼び出しを行う場合に過去に多くのx < y投票者を確認する。

付属文書B

シミュレーション結果

Python においてSPECTREプロトコルを実装し、ネットワーク力学の事象主導型シミュレーターを実装した。各実験で20ノードのエルデシューレーニイランダムネットワークトポロジーを作成した。各ノードが5つの発信リンクを構成する。各リンクの遅延は均一に分散され、後に直線的に縮尺を調整することでグラフの直径が*D*となるようにする（指定の*D*）。各ポイントは少なくとも500回の実験の平均的な結果を表している。

SPECTREの主な利点は確認の速さである。我々の正式な分析から算出した漸近的待ち時間は$\mathcal{O}\left(\frac{\ln(1/\epsilon)}{\lambda(1-2\alpha)}+\frac{D}{1-2\alpha}\right)$である。実際の待ち時間を測定するために、アルゴリズム7から導き出したオンライン受諾ポリシーを活用した。付属文書Cの最後で説明した通り、マーチャントは過去*D*秒に二重支払いがリリースされていないことを確認す

るために追加で*D*秒待つ必要がある、という点を強調する。

遅延直径は受諾時間にどのような影響を与えるか？

ブロック作成率が高い場合、受諾待ち時間のほとんどはブロック伝播遅延となる。図5では様々な遅延直径値*D*、そして様々なセキュリティ閾値ϵにおけるSPECTREのトランザクション受諾時間を示している。ナカモトコンセンサスと異なり、*D* はトランザクションの受諾時間に影響を与えているが、セキュリティには影響を与えていない。

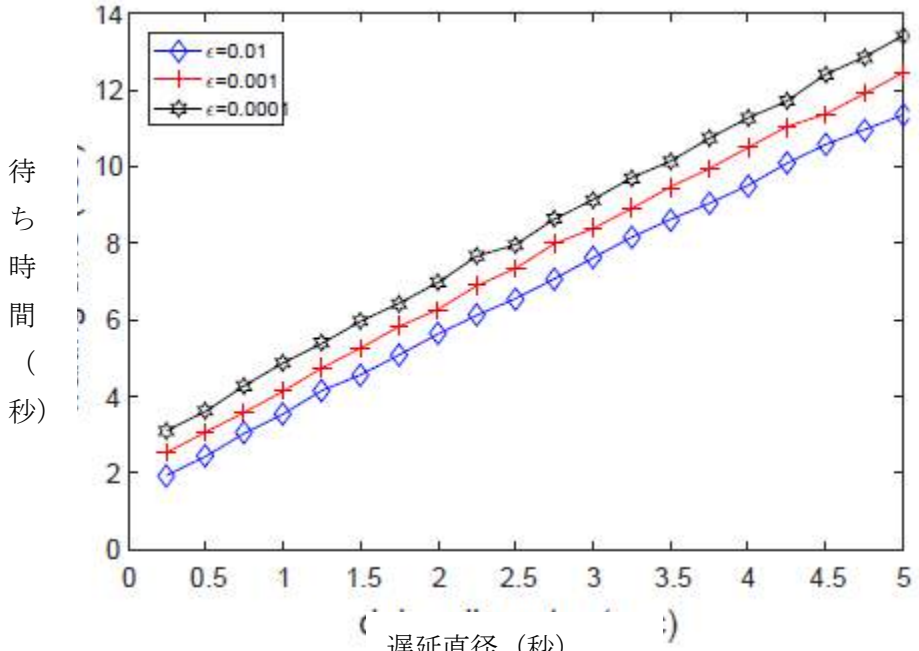

図5:トランザクションが *Robu*[illegible] 均時間。目に見える二重支払いがない前提。λは1秒10ブロック、α は0.25である。

ブロック作成率が受諾時間にどのような影響を与えるか？

図6では一定の遅延*d* = 5秒の条件下で、ブロック作成率 λ の様々な値における受諾時間を示している。グラフは我々の漸近的拘束におけるλの役割を再確認している。ブロック作成プロセスを加速することで受諾時間が短縮されている。比較してみると、ビットコインのブロック作成率である$1/600$ は桁違いに長い待ち時間を示唆している（グラフには記載していない）。

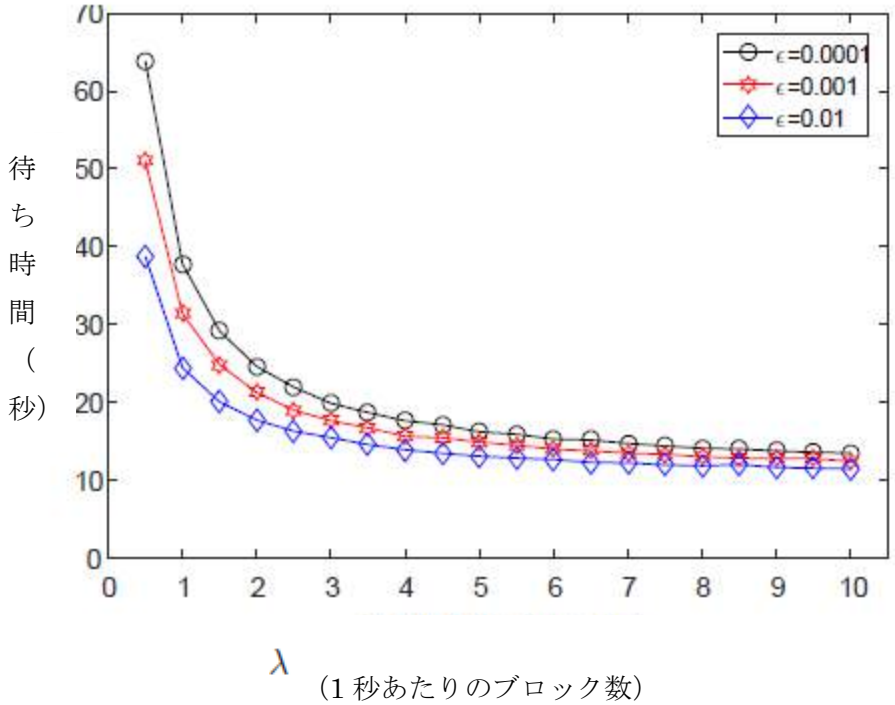

図6: トランザクションが*RobustTxO*に入るまでの平均時間。目に見える二重支払いがなく、d = 5秒でα = 0.25という条件

攻撃者が受諾を遅らせることができるか？

一部の不誠実なノードが他のマイナーのブロックを参照しないブロックを公開する検閲攻撃の効果を証明する。

SPECTREの生存性の弱さ特性 (提案3) では検閲攻撃が存在する場合にも二重支払いではないトランザクションを迅速に受諾することが保証されている。

ただし、そのような攻撃によってトランザクションの受諾に多少の遅延が発生する可能性はあるが、小規模な攻撃者にとってこの遅延は些細なものである。図7では「平和な日々」の受諾時間と検閲攻撃下の受諾時間を比較することでこの影響を数量化している。ここで使用しているパラメータはd = 5秒、$\lambda =$ 1秒あたり10ブロック、そして$\epsilon = 0.01$である。結果は攻撃の微細な影響を示しているが、5〜10秒以上トランザクション受諾を遅延させるために攻撃者はネットワーク内の計算能力の大きな割合を占める必要があることを示している。

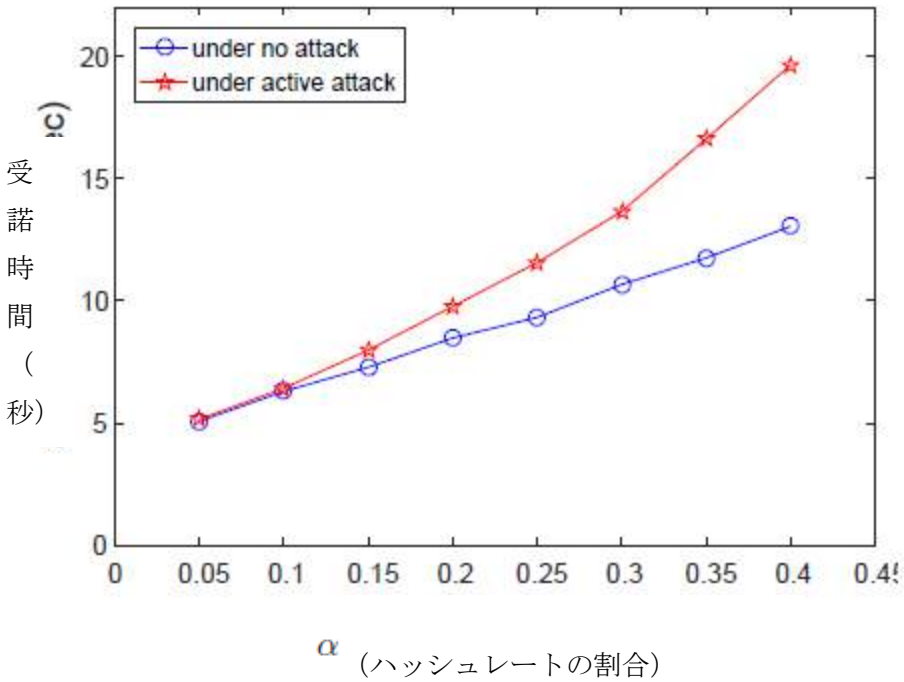

図: トランザクションが*RobustTxO*に入るまでの平均時間。検閲攻撃が発生しており、目に見える二重支払いがなく、*d* = 5秒、$\lambda =$ 1秒あたり10ブロック、そして $\epsilon = 0.01$ という条件

ϵ は攻撃者の規模によってどのように減少するか？

誠実なノードϵ-がトランザクションを受諾した場合でも最終的にトランザクションが拒絶される小さなリスクがある。非常に有能な攻撃者の場合でもこの事象が発生する可能性はすぐになくなることを証明している（例：ハッシュレート $\alpha = 0.4$ の場合）。図8ではこれを示している。*d* = 5秒、$\lambda =$ 1秒あたり10ブロックという前提である（y軸は対数目盛で表示されている）。

我々のセキュリティ分析の厳格性はどの程度のものか？

アルゴリズム3が依存する分析では攻撃成功の可能性を制限するために攻撃者が一切の遅延なく全てのノードにブロックをブロードキャストし、全てのノードからブロックを受領できる、などの複数の最悪の事態を想定している（付属文書Eの補題14と20を参照）。従って、分析は厳格なものではなく、実際には攻撃が成功する可能性はより低い。図9では分析の限界と2つの異なる実証的シミュレーション間の比較を行っている。これらのシミュレーションでは攻撃者用にブロックを作成し、最適な二重支払い攻撃のシミュレーションを行っている。グラフ内の各ポイントで実験を10,000回反復し、実証的な成功率を測定した。

シミュレーションでは遅延なしにブロックの送受信を行うことのできる最悪の場合の攻撃者と遅延を持つその他のノードに接続されるより現実的な攻撃者という2種類の攻撃者を想定している。我々はこれらの条件下での攻撃成功率をSPECTREのポリシーが

計算する分析リスクと比較した（アルゴリズム7）。
結果、SPECTREの*RiskTxAccept*が考慮するリスクは実際のリスクの上限であり、トランザクションは我々が公式に保証するよりもさらに安全であることを証明している。

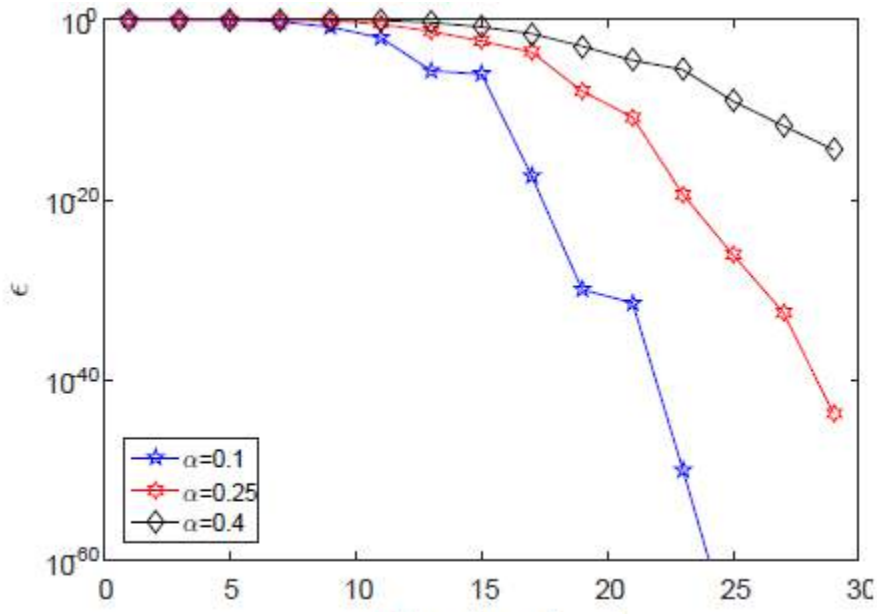

図8: d = 5秒、$\lambda =$ 1秒あたり10ブロック、$\alpha =$ 0.1、0.25及び0.4の条件下での二重支払い攻撃の成功確率と受諾までの待ち時間の関係。ここでの確率はアルゴリズム3で実施する計算の結果である。

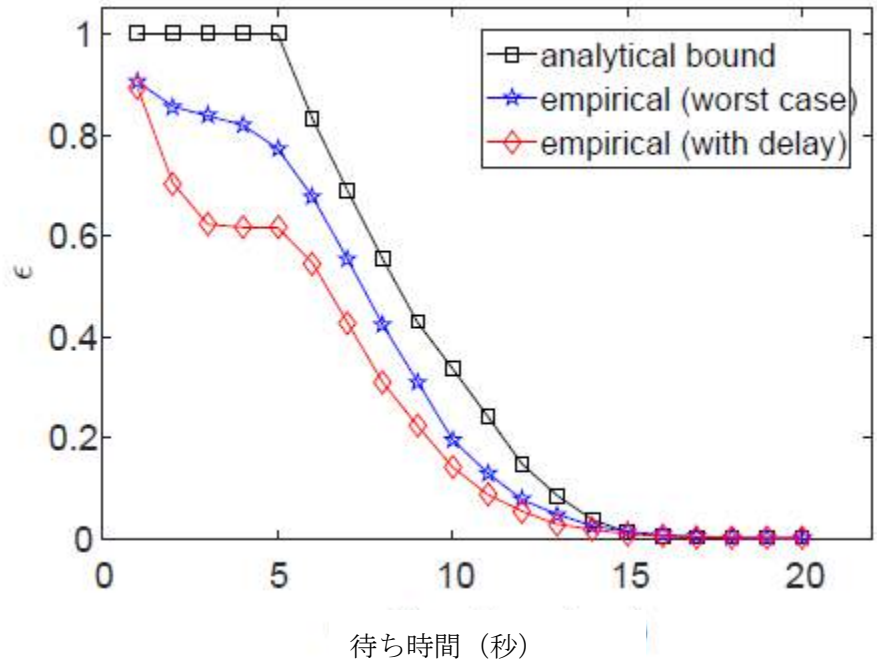

図9: d = 5秒、λ =10、α =0.25の条件下での二重支払い攻撃成功の分析確率と現実的確率及び受諾までの待ち時間の関係

付属文書C

追加のアルゴリズム

セクション4ではSPACTREによるブロックのペア順序の決定方法、そしてこの順序を用いた受諾済みトランザクションのサブセットの構築方法について説明した。このセクションではこの順序の堅牢性を測定し、これを受諾済みトランザクションの堅牢性に変換するためのSPECTREの第二層の手順について説明する。

A. ブロックのペア順序決定の堅牢性

現在観察可能なDAGの順序ではブロックxがyに対して優先されると仮定する。この関係が永久的に継続する可能性を測定する方法が必要である。アルゴリズム3では攻撃者が$x \prec y$という関係を反転させることのできる確率に対する上限をアウトプットする。引数yを指定しない場合、アルゴリズムのアウトプットではxの見えていないブロック（攻撃者が隠している、もしくはまだ作成されていない）に対する堅牢性を解釈する。アルゴリズム内の $gap(b, G)$ はセット $\{z \in anticone(b, G) : vote_{z,b}(virtual(G)) \geq \check{0}\}$ のサイズを示している。以下のパラグラフでは $\langle G, z, K \rangle$ について説明する。

以下のアルゴリズムでは簡潔さのためにユーザが自身で設定する必要のある次のパラメータを省略している: α – 攻撃者の最大サイズ、 d – Dの上限D (ネットワーク内の最近の遅延直径), λ – ブロック作成率

アルゴリズム 3 *Risk* (オフライン)

インプット: $G^v_{time\_now}$ – ノードvの現在のブロックDAG, x – Gのブロック、y (任意) – $anticone(x, G)$ 内のブロック

アウトプット: *risk* – 未来のある時点でxがyに対して優先されない確率に対する上限 $(\Pr(\exists u \in honest, \exists s \geq time\_now : vote_{x,y}(virtual(G^u_s)) \geq 0))$

1: $time\_now < publication(x) + 2 \cdot d$ の場合、

2: 1を返す

3: $K \leftarrow \lceil \sqrt{|future(x, G)|} \rceil$

4: $NULL = y$ の場合、

5: $g \leftarrow |\overline{future}(x, G)|$

6: $M \leftarrow 0$

7: その他の場合

8: $g \leftarrow \sum_{z' \in \overline{future}(x,G)} vote_{y,x}(z', G)$

9:

$M \leftarrow \left|\left\{z \in future(x, G) : vote_{x,y}(z, G) = +1 \wedge gap(z, \langle G, z, K \rangle) = 0\right\}\right|$

10: $n_x \leftarrow \left|\overline{future}\,(x, G)\right| - M$
11: $j \leftarrow gap\,(x, G) + K$
12: $l \leftarrow K$
13:
$risk \leftarrow f_{pre\_mine}(l) + f_{pre\_pub}(K) + f_{post\_pub}\,(M) + f_{post\_mine}\,(n_x, g, j, l, M)$
14: *risk* を返す

13行目で本アルゴリズムはいくつかの関数を使用しているが、これらの詳細な定義については後のセクションで説明する。(54)では f_{pre_pub} の明確な公式を示している。f_{post_pub} は(50)と(52) 、f_{post_mine} は(5)で指定している。 補題 24に先立って、f_{pre_mine} の数値を計算する方法を説明する。

関数 f_{pre_mine} は攻撃者がプレマイニングフェーズよりも大きい/プレマイニングフェーズ中（*x*の作成まで）利得を得ている確率の上限である。関数 f_{post_mine} は攻撃者が$x \prec y$の関係を逆転させられるだけのブロックを作成できる確率の上限である。基本的に f_{post_mine} は[17]の公式を改良したものである。我々のバージョンの公式では、の間隔 $[time(x), t_{acc}]$ （t_{acc} は現在時間を表す）中に誠実なノードが n 個のブロックを作成した場合、攻撃者がこの間隔中に*m*個のブロックを作成している確率は $\binom{n-1+m}{m} \cdot \alpha^m \cdot (1-\alpha)^n$ である。*g*で $future\,(x, G)$ 内のブロックの全ての投票を集計する場合、攻撃者が過半数の投票を逆転させることのできる可能性は約 $\left(\frac{\alpha}{1-\alpha}\right)^{\max\{g-m,0\}}$ である。
数式の組み合わせによって攻撃の成功確率の上限を算出する。[6]
ここでの主な課題は*n*を正確に測定することである。アルゴリズム3では構造情報のみを使用し（が少なくとも $2 \cdot d$ 秒公開されていることの確認は除く）、ブロックのタイミングの測定に依存しないためにこれは困難なタスクである。$publication(x)$ の後に作成されたブロックの上限を設定するために $n \approx |future\,(x, G)|$ を使用する手段はあるが、次の2点の課題がある。
・不誠実なノードがブロック*x* を作成し、隠している可能性がある。この場合、作成と公開の間に長い時間が経過しており、 $|future\,(x, G)|$ だけではnをはるかに下回る可能性が示唆される。*n*の過小評価を回避するために、変数*j*によって $anticone\,(x, G)$内の誠実なブロック数の上限を設定し、これを我々のカウントに追加している (加算は$f_{post_mine}\,(n_x, g, j, l, M)$内で行っている)。
関数f_{pre_pub}は我々が*j*を過小評価する確率の上限を設定している。

・攻撃者は攻撃ブロックを公開することで$future\,(x, G)$ のサイズを増加させ、我々に*n*を過大評価させる。これによって攻撃の成功確率の上限が厳格でなくなり、大きな α の値を持つ攻撃者が受諾を無限に遅延させることができる。*Risk*では攻撃者のブロックを認識し、*n*のカウントから除外することでこの問題を解決する。これは以下の方法で行う。

*G*はブロックDAG、*b*は*G*内のブロック、*K*は整数である。z_1から*b*までエッジを接続し、全てのエッジ$(z, b) \in G$ を$(z, z_K) \in G$と置き換えることで*K*仮想ブロックの新しいチェーン$z_1, \ldots, z_K$ を作成することで$\mathrm{DAG}\ \langle G, b, K \rangle$ を算出する。これによって、DAG *K*に$y \notin past\,(x, G)$ に対抗して強く$x \prec y$と投票する人工投票者が追加される。

9行目ではアルゴリズムは$gap\,(z, \langle G, z, K \rangle) = 0$かどうか、つまり、$anticone\,(z)$ 内で修正された$\mathrm{DAG}\ \langle G, z, K \rangle$内の*z*に対して優先されるブロックが存在するかどうかをチェックする。存在しない場合、*z*を*n*にカウントする（10行目）。

以下の特性ではこの手順の有効性について説明している: *誠実なブロックのためにKの投票者を追加する場合、一部の小さなKではペア順序内で他に優先されるブロックが存在しない（過去セットは除く）*。これは補題29で正式に明言され、証明されている。

6後に説明するように我々が使用する計算値はより複雑である。セクション5では*g*内において、DAG全体の投票ではなく$future\,(x, G)$ のみの投票を集計する理由について説明している。基本的に投票者は未来の事象の発生に応じて投票を変更する可能性があるために*past* (*x*)内のブロックの投票を含めた全ての投票をカウントすることに意味はない。*past* (*x*)内の投票者が$x \prec y$をどれだけ強く支持したかを測定することが重要である。

関数f_{post_pub} は$future\,(x, G)$ 内の誠実なブロックの数を過少評価する確率の上限である。

B. トランザクション受諾の堅牢性

次の手順ではブロックの堅牢性（*Risk*が計算）をトランザクションの堅牢性に変換する。この移行はブロックの（堅牢でない）順序（アルゴリズム1）から受諾済みの（堅牢でない）トランザクションのセットへの移行（アルゴリズム2）と同様に実施する。*RiskTxAccept* 手順 (アルゴリズム4) では*G*と*tx*（そして追加の引数）をインプットとして一部の誠実なノードが*tx*を受諾しない確率の上限を返す。*RiskTxAccept*の主なタスクは*Risk*が誘因するエラー限界を適切に考慮に入れて集計することである。簡単に認識できるように、*RiskTxAccept* と *RiskTxReject* (アルゴリズム5)はお互いの鏡像である。*RiskTxAccept* では特定のトランザクションが受諾済みのトランザクションサブセットから削除される確率の上限を設定する一方で、*RiskTxReject* では特定のト

ランザクションがこのサブセットに含まれる確率の上限を設定する。これは$tx_2 \in y$と$tx_1 \in x \in future(y)$のように2つの衝突するトランザクションがトポロジー的に関連しているが、tx_2が（以前の何らかの衝突により）受諾済みセット内に存在しない場合に特に重要である。この場合、tx_2を含むブロックはtx_1を含むブロックに対して優先されるが、我々はtx_1を受諾する。*RiskTxReject*.が計算するtx_2の拒絶状態に堅牢性があるかどうかも受諾される。

アルゴリズム 4 *RiskTxAccept*
インプット: $G = G^v_{time\_now}$ – ブロックDAG アウトプット: *risk* – 未来のある時点で誠実なノードが$[tx] \cap subG$のトランザクションを受諾しない確率の上限 $(\Pr(\exists u \in honest, \exists s \geq time\_now, [tx] \cap subG \cap RobustTxOG^u_s = \emptyset))$ 1: $minrisk \leftarrow 1$ 2: 全ての$z_1 \in Z_G([tx]) \cap subG$ 3: $risk \leftarrow Risk(G, z_1, \emptyset)$ 4全ての$tx_2 \in G \cap conflict(tx)$ 5: 全ての$z_2 \in Z_G(tx_2) \cap anticone(z_1, G)$ 6: $risk \leftarrow risk + Risk(G, z_1, z_2)$ 7: $risk \leftarrow risk + RiskTxReject(G, [tx_2], past(z_1))$ 8: 全ての$[tx_3] \in inputs(tx) \cap past(z_1)$ 9: $risk \leftarrow risk + RiskTxAccept(G, [tx_3], past(z_1))$ 10: $minrisk \leftarrow \min\{minrisk, risk\}$ 11: $risk \leftarrow minrisk$ 12: *risk* を返す

これらの手順に基づいてSPECTREの*RobustTxO*手順について紹介する。ユーザは彼が現在観察するDAG全体*G*を、そして許容可能な最大誤差確率ϵをインプットとして提供すべきである。上述のように、ユーザはα, d, λパラメータも設定すべきである。これらは上述の*RobustTxO*の補助手順において使用する。

アルゴリズム5 *RiskTxReject*
インプット: *G* – ブロックDAG、*subG* – （仮想の可能性のある）ブロックの過去である*G*のサブDAG、*tx* – 保護するトランザクションのコピー アウトプット: *risk* – 未来のある時点で誠実なノードが$[tx] \cap subG$内のトランザクションを受諾する確率の上限

1: $risk \leftarrow 0$
2: 全ての$z_1 \in Z_G([tx]) \cap subG$
3: $minrisk \leftarrow 1$
4: 全ての$tx_2 \in G \cap conflict\,(tx)$
5: 全ての$z_2 \in Z_G(tx_2) \cap anticone\,(z, G)$
6: $minrisk \leftarrow \min\{minrisk, Risk\,(G, z_2, z_1)\}$
7: $minrisk \leftarrow \min\{minrisk, RiskTxAccept\,(G, [tx_2], past\,(z_1))\}$
8: 全ての$[tx_3] \in inputs\,(tx)$
9: $minrisk \leftarrow \min\{minrisk, RiskTxReject\,(G, [tx_3], past\,(z_1))\}$
10: $risk \leftarrow risk + minrisk$
11: *risk*を返す

アルゴリズム6 *RobustTxO*

インプット: $G = G^v_{time\_now}$ – アルゴリズムを実行するノードが観察する現在のDAGを表すブロックDAG、 ϵ – ユーザが許容する最大リスク、α – 攻撃者の最大規模、*d* – ネットワークの遅延直径の上限、λ – ブロック作成率
アウトプット: 特性2が定義するように受諾されたままであることが保証されたトランザクションのセット

1: $RobustTx \leftarrow \emptyset$
2: 全ての$z \in G$
3: 全ての$tx \in z$
4: の場合、$RiskTxAccept\,(G, [tx] \cap G) <$
5: *tx*を*RobustTx*に追加

C. オンラインポリシー

ユーザのブロックが確認を得た時点でユーザがオンライン状態であることを要求する*Risk*の代替実施方法について紹介する。この前提は絶え間ない顧客の行列に対応するレジのような多くの現実的なシナリオに通用する。オンラインバージョンの主な利点はより厳格な分析に依存するために、トランザクション受諾が若干速くなるという点である。ここでは目に見える二重支払いが存在しないケースに限定して考える(例：*y* = *NULL*).

ユーザがオンライン状態にあるという事実を次の2種類の方法で活用できる: 最初に、ユーザが$received^v(b) + 2 \cdot d$後に受信し、$future\,(x)$に帰属しないあらゆるブロックは攻撃者のブロックとしてマーキング可能である。[7] 第2に、ユーザは*x*の作成から経過した時間を測定することで隠れた攻撃者のブロック数を概算できる。

7 $received^v b$ はノード*v*がノード*b*を受信した時間である。以下では、全ての r, G_r^{pub} は $G_r^{pub} := \cup_{u \in \mathcal{N}} G_r^u$ と定義される。

以下では*Risk*のオンラインバージョンについて説明している。アルゴリズムではノード*v*のDAG、保護対象のブロック*x*をインプットとしてブロック $y \in G_\infty^{pub} \setminus G_t^{pub}$ がこれに対して優先される確率の上限を返す。

アルゴリズム 7 *Risk* (オンライン)
インプット: G_t^v – *t*の時点で*v*が観察するブロックDAG、*x* – G_t^v 内のブロック
アウトプット: *risk* – $y \in G_\infty^{pub} \setminus G_t^{pub}$ でブロックが未来のある時点で*y*に対して優先されない確率の上限
1: $time\_now < publication(x) + d$ の場合、
2: 1を返す
3: $T \leftarrow time\_now - received^v(x)$
4: $G_x \leftarrow G^v_{received^v(x)+2\cdot d} \cup future\,(x, G_x)$
5: $g \leftarrow \min_{x' \in \overline{anticone}(x, G_x)} \lvert future\,(x', G_x)\rvert$
6: $risk \leftarrow risk\_hidden(T, g)$
7: $risk < \epsilon$ の場合
8: *ACCEPT*を返す
9: それ以外の場合
10: *WAIT* を返す

(45)-(46)では $risk\_hidden$ の定義について説明している。実務では、ノード*v*には $G_\infty^{pub} \setminus G_t^{pub}$ が部分的に見えているために、アルゴリズム7を使用するためにユーザは追加で*d*秒待って $conflict\,(tx) \cap G^v_{t+d} = \emptyset$ を確認する必要がある。つまり、攻撃者が間隔 $[t-d, t]$ 内で二重支払いを公開していないことを確認する必要がある。オンラインポリシー修正の正確性は結果 27において証明している。

付属文書D
実施詳細

ミンティング

SPECTREではターゲットが以下で定義する必要値*TARGET*を満たすあらゆるブロックは同じミンティング報酬を受領する。その目標値が多くとも $(1+\delta)$ の係数だけ*TARGET*よりも高い場合（解決がより簡単）、その報酬は同じ係数分だけ減少する。パラメータ δ は陳腐化した難易度によってマイニングを行ったブロックに対するプロトコルの耐性を表している。そのため、選択した δ が2に等しい場合、目標値

$2 \cdot TARGET$ もしくは$3 \cdot TARGET$ のブロックが有効であり、これらのミンティング報酬はそれぞれ2もしくは3の係数だけ減少する。目標値が$3 \cdot TARGET$ より高いブロックは無効であり破棄される。
*TARGET*の定義及び再調整方法について説明する。
再計算
ビットコインやその他のPoWに基づくシステムと同様に、*TARGET*で表現するブロック作成難易度（サブセクション4.1）を状況によって変更する必要がある。様々なネットワーク条件やシステムに投資する計算資源の量の変化によってネットワーク混雑を回避するために1秒あたりに作成するブロック数を制限する必要がある。ビットコインでは次のような方法でこれを実行する。ブロック2016個ごとに次のブロック（リファレンスブロック）を調整した難易度に従ってマイニングする。新しい難易度は前のリファレンスブロックからの経過時間（各ブロック内のタイムスタンプを使用）を再計算式に接続することで計算する。

この数式のアウトプットは新しいリファレンスブロックのマイニングを行うための*TARGET*の新しい値である。我々はこのスキームをSPECTREのために改良している。
x_{n-1} は前のリファレンスブロックである。
$\left|past\left(x_n\right) \cap \overline{future}\left(x_{n-1}\right)\right| = 2016$ の特性を持つ全ての新規ブロック x_n [8] は新しいリファレンスブロック候補である。追加の候補が存在する場合、$dist\_gap$ が最小の候補を選択し、随意的な体ブレーキングを行う。この場合、
$dist\_gap(b, G) := \min_{K \in \mathbb{N}} gap\left(b, \langle G, b, K \rangle\right) = 0$ である。変数
$dist\_gap(b, G)$ は最小の*K*を表し、*b*のために*K*投票を追加すると*gap* が0に等しくなる。これによってx_n 候補（上記特性を満たす）のセット内でx_{n-1}を引き継ぐ唯一のブロックがリファレンスブロックとして選択される。特に、付属文書Aで説明しているように、一定期間隠されていた攻撃者ブロックでは$dist_gap$ が大きくなるため、リファレンスブロックとなる資格はない。さらに、x_{n-1} の前にマイニングされた攻撃者ブロックは$future\left(x_{n-1}\right)$に帰属することができないために次の再計算に影響を与えることはない。

新しいリファレンスブロックをマイニングする新しい難易度を x_{n-1} とx_n の間の経過時間を用いて*TARGET*を更新する数式を通じて再度計算する。
数式ではノードが1秒あたり1MBのような十分な帯域幅を持つと考えられる事前定義されたλ を目指すべきである。この難易度は $antipast\left(x_n\right) \setminus anticone\left(x_{n+1}\right)$内の全てのブロックの難易度を決定する。 x_{n+1} は次のリファレンスブロックである。このセット内の全てのブロックはx_nと同じ難易度に従ってマイニングすべきである。
ブロック$b \in antipast\left(x_n\right) \setminus \overline{anticone}\left(x_{n+1}\right)$ をリファレンスブロックx_nが指定する難易度よりも簡単な難易度で解決する場合、陳腐化した目標が多くとも x_n の目標の $(1+\delta)$ であることを条件（例：多くとも $(1+\delta)$だけ難易度が簡単）として

b は有効とみなされる。 パラメータ δ はプロトコルの許容度閾値である。*b*のマイニング報酬は上記で説明するように対応する係数だけ減じられる。目標値が必要値を $(1+\delta)$ を上回る係数だけ超過するブロックは無視され破棄される。

ブロックヘッダー

全てのブロックをDAGに組み込むために、全てのブロックはそのヘッダーに前のブロックのハッシュに対するポインターを組み込んでいる。冗長化は許容されないため、*b* のヘッダーでは*past* (*b*)のリーフブロックだけをポイントすべきである。この結果、ブロックのヘッダーが $\approx 50 + d \cdot \lambda \cdot 32$ バイトとなる。そのため、ブロックサイズの削減とブロック作成率の上昇範囲には限度があり、ブロックヘッダーのオーバーヘッドは含まれるトランザクション数に応じて大きくなる。さらに、現在観察可能なDAGにリーフが多すぎる場合（稀に発生するブロックのバースト、もしくは攻撃者が多くの陳腐化したブロックを公開する場合）、次のブロック作成者は指し示すリーフブロックの数を制限することができます。未来のブロックのヘッダーには利用可能なスペースがあるためこのブロックによって除外されたブロックをポイントして含めることで後からこれらをDAGに統合することができる。

効果的な実装

我々の最新のSPECTREの実装では効率の低い場合の多いシンプルな計算値を用いており、*genesis*ブロックまで投票のカスケードを行う。

[8]これらのパラグラフではいくつかの新しい表記を行っている: $\overline{future}(x) := future(x) \cup \{x\}$ 、そして同様に $\overline{past}(x)$ 及び $\overline{anticone}(x)$。さらに、$antipast(x) = future(x) \cup anticone(x)$、及び同様に $antifuture(x)$

複数の効率的な実装が可能である。このような効率的な実装を設計するには攻撃者が陳腐化したブロックの特異な構造を曝露することで他のノードに対して広範囲に及ぶ計算を実施することを強いるCPU攻撃に対する注意が必要である。これらの攻撃は攻撃者にとって大きなコストを伴うものである。 $\mathcal{O}(|G|^3)$ によるシンプルな実装と比較して予想される $\mathcal{O}(d \cdot \lambda)$ において機能するSPECTREの手順を実装している。その詳細な仕様とCPU攻撃のコスト証明については将来の論文にて説明する。

トランザクション料金

トランザクション本文では支払人から支払い受取人へと移動する金額を指定する。トランザクション料金では支払人からそのブロックにトランザクションが含まれるマイナーへの支払いを指定する。我々は以下の意味においてこれらの2つの要素を個別のトランザクションとみなしている。 $tx \in x$ と仮定し、 $fee(tx, x)$ はブロック*x*の作成者への料金支払い tx を表す。2つの異なるブロック x, y に tx の2つのコピーが含まれるものと仮定する。この場合、本文は同じトランザクションのコピーとみなされ（セクション2の $[tx]$ を参照）、トランザクション $fee(tx, x)$ と $fee(tx, y)$ は衝突、つまり二重払いとみなされる。従って、SPECTREの通常スキームと同様に、料金をブロック $x \text{ iff } tx \in TxO(G)$ の作成者に与え、*x*は*tx*を含むその他全てのブロックに対して優先される。

*x*と*y*の関係に堅牢性がない特殊な状況ではこのルールによってマイナーが害を被る可能性がある (SPECTREではこれらのブロックが近い時間に公開され、攻撃が発生中の場合には堅牢性を保証しない)。我々は決済トランザクションを導入することでこの問題に対応する。決済トランザクションは*x*と*y* の作成者が自身のブロックが衝突することを観察した後に署名する任意のトランザクションである。これを $settlement(x,y)$ と表記する。$settlement(x,y)$ では $x \cap y$ のトランザクションの全て（もしくは関係当事者が選択する場合には一部）の料金を*x*と*y*の間で均等に配分するものと解釈する。$settlement(x,y)$ では基本的に $fee(tx,x)$ と $fee(tx,y)$ をオーバーライドする。DAG内のブロック*z*に $settlement(x,y)$ が含まれる場合、受諾されたものとみなされ（$TxO(G)$ のメンバー）、$x,y \in past(z)$ で*z*はトランザクション $spending\ fee(tx,x)$ もしくは $fee(tx,y)$ を含む全てのブロックに対して優先される。そのため、一当事者が自身に帰属する前に料金を使った場合、高い確率で後から決済することはできない。そのため、マイナーはトランザクション料金報酬が確定する、もしくは決済が開始するまで待ってから報酬を使用することを推奨される。
このスキームを用いて複数当事者のブロック間の衝突を同時に解決する。さらに、決済スキームは料金に関する衝突だけでなく、あらゆる二重支払いに適用できる。

付属文書 E
理論3の完全な証明

理論3. *あらゆる* $D \cdot \lambda$ *に関して、SPECTREのセキュリティ閾値は50% である。*

A. 追加表記

– $node(b) \in \mathcal{N}$ – ブロック*b*を作成したノード、*time*(*b*) – 作成時間、*publication*(*b*) – *node*(*b*)が他の誠実なノードに*b*の送信を開始した時間、$received^v(b)$ – ノード*v*が*b*を受領した時間

. $\overline{future}(x) := future(x) \cup \{x\}$、そして同様に $\overline{past}(x)$ 及び $\overline{anticone}(x)$。さらに $antipast(x) = future(x) \cup anticone(x)$、*そして同様に* $antifuture(x)$

. $\hat{\mathcal{E}}_s^u(x,y) := vote_{y,x}(virtual(G_s^u)) = +1$ の事象

. $\mathcal{E}_s^u(x,y,\epsilon) := Risk(G_s^u,x,y) < \epsilon$ の事象

. $\mathcal{A}_s^u(tx) := tx \in TxO(G_s^u)$ の事象

. $\mathcal{A}_t^u(tx,\epsilon) := TxO \in RobustTxO(G_t^u,\epsilon)$ の事象

. $\mathcal{E}_{t\to\infty}^{all}(x,y,\epsilon) := \cap_{u\in honest} \cap_{s\in(t,\infty)} \mathcal{E}_s^u(x,y,\epsilon)$ の事象、そして同様に

$\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)$、$\mathcal{A}^{all}_{t\to\infty}(tx,\epsilon)$ 及び $\widehat{\mathcal{A}}^{all}_{t\to\infty}(tx)$

. $past_h(z,G)$:= $past(z,G)\cap honest$、そして同様に未来及び過去セット

. $V_{x\prec y}(G)$:= $\{z\in G|z$ は(x,y) に関して強力な投票者である。

$vote_{x,y}(z)=-1\}$（$V_{x\prec y}(G)$ はx,yに依存する。）

. $\mathcal{P}_{oiss}(\delta,j):=e^{-\delta}\cdot\frac{\delta^j}{j!}$

B. 正式な主張

理論3を分解して安全性、進捗性、生存性の弱さそして整合性という各セキュリティ特性について個別に提案を行う。理論を証明するめに以下の提案を証明する必要がある。

提案 4 (整合性)

受諾されるセットの整合性がとれている: *履歴Gに関して*

1) $tx\in TxO(G)$ *及び* $tx_2\in inputs(tx)$ *の場合、* $tx_2\in TxO(G)$

2) $tx\in TxO(G)$ *及び* $tx_2\in conflict(tx)$ *の場合、* $tx_2\notin TxO(G)$

提案 5 (安全性)

$v\in honest$ 及び時間 t に関して、$tx\in RobustTxO(\epsilon,G^v_t,d^v,\alpha)$ で確率が少なくとも $1-\epsilon$ の場合、$a\ \tau\geq t$ が存在し、$\forall u\in honest,\forall s\geq\tau:$ $RiskTxAccept(tx,G^u_s,d^u,\alpha)<\epsilon$ となり、期待される$\tau-t$が有限となる。

提案6 (弱い生存性)

*t*を現在時間とし、$tx\in G^{pub}_t$ と仮定する。 $\psi\geq t$ は誠実なノードϵ- が txを受諾する*t*後の最も速い時間とする。その場合、$conflict(tx)\cap G^{pub}_{\psi}=\emptyset$ の事象と、全ての$tx_2\in inputs(tx)$でtx_2 が誠実なノードに永久的に受諾された状態を維持する事象を条件に、期待される$\psi-t$が有限となる。

特定のϵ でトランザクションを確実に受諾した後にこのトランザクション（確率が少なくとも$1-\epsilon$）は全ての$\epsilon'<\epsilon$でも同様に受諾されるという別の提案を追加及び証明する。

提案 7 (進捗)

あらゆる$v\in honest$ と時間*t*において、$tx\in RobustTxO(\epsilon,G^v_t,d^v,\alpha)$ で確率が少なくとも $1-\epsilon$ の場合、あらゆる ϵ' でϕ が存在し、$\forall s\geq\phi:$ $RiskTxAccept(tx,G^v_s,d^v,\alpha)<\epsilon$ となり、期待される $\phi-t$ が有限となる。

上記3つの各提案に関してブロックの堅牢性（トランザクションの堅牢性ではない）に関して一致する提案を行っている。

提案 8 (安全性 (ブロック))

あらゆる $v \in honest$ において、$Risk\left(x, y, G_t^v\right) < \epsilon$ で確率が少なくとも $1-\epsilon$ の場合、τ が存在し、$\forall u \in honest, \forall s \geq \tau : Risk\left(x, y, G_s^u\right) < \epsilon$, 及び $\mathbb{E}[\tau - t] < \infty$ となる。

提案9 (進捗 (ブロック))

あらゆる $v \in honest$ において、$Risk\left(x, y, G_t^v\right) < \epsilon$ で確率が少なくとも $1-\epsilon$ の場合、あらゆる $\epsilon' < \epsilon$ で ϕ が存在し、$\forall s \geq \phi : Risk\left(x, y, G_s^u\right) < \epsilon$ 及び $\mathbb{E}[\phi - t] < \infty$ となる。

提案10 (生存性の弱さ (ブロック))

*t*を現在時刻とし、$x \in G_t^{pub}$ と仮定する。

ψ *を誠実なノードv*で$Risk\left(x, y, G_s^v\right) < \epsilon$ *となる最初のタイミングsとする。その場合、* $y \notin G_\psi^{pub}$ *の事象を条件に、期待される* $\psi - t$ *は有限となる。*

以下のサブセクションにおいてこれらの提案の正確性を証明している。整合性については後のサブセクションで証明する。最初に3つのシンプルな補題から開始する。

C. 基本的な特性

以下の2つの補題はアルゴリズム1の7-14行目のものである。

補題11

トポロジー関係について満場一致の合意がある: $G = (C, E)$ *がブロックDAG であり、* $(y, x) \in E$ *の場合、* $\forall z \in G : vote_{x,y}\left(z, G\right) = -1$ *である。*

補題12

あるブロックがその過去のブロックに関して行う投票はその過去にのみ依存するため、永久的に固定される。 G_1 *と* G_2 *は2つのブロックDAG であり、* $x, y, z \in G_1 \cap G_2$ *と仮定する。* $\{x, y\} \cap \overline{past}\left(z\right) \neq \emptyset$ *の場合、* $vote_{x,y}\left(z, G_1\right) = vote_{x,y}\left(z, G_2\right)$ *となる。*

従って、*z*は$z \in \overline{future}\left(x\right) \cup \overline{future}\left(y\right)$ の場合、ペア (x, y) の強力な投票者であり、それ以外の場合は弱い投票者である。

以下の補題では*genesis*の投票が仮想ブロックの投票と一致していることを証明している。直観的に、*genesis*はDAG内の自身を除く過半数の投票に従って投票し、この過半数を増幅し、仮想ブロックの投票を決定する。

補題13

genesisの投票が最終投票である: $vote\left(virtual\left(G\right)\right) = vote\left(genesis, G\right)$

証明

$vote_{x,y}\left(genesis, G\right) \geq 0$ の場合、$vote_{x,y}\left(virtual\left(G\right)\right) \geq 0$ であることを証明するだけで十分である。(x, y) がトポロジー的に関連している場合、補題11によって全ての投票がこれらの順序に対して満場一致で合意し、特に $vote_{x,y}\left(virtual\left(G\right)\right) = vote_{x,y}\left(genesis, G\right) \geq 0$ となる。それ以外の場合、x もしくは y がブロックとなることはないため、*genesis*は弱い投票者となり、14行目によって以下の証明が得られる。

$$vote_{x,y}\left(virtual\left(G\right)\right) = \widetilde{sgn}\left(\sum_{z \in G} vote_{x,y}\left(z, G\right)\right) = \tag{1}$$

$$\widetilde{sgn}\left(vote_{x,y}\left(genesis, G\right) + \sum_{z \in future(genesis, G)} vote_{x,y}\left(z, G\right)\right) \geq \tag{2}$$

$$\widetilde{sgn}\left(\sum_{z \in future(genesis, G)} vote_{x,y}\left(z, G\right)\right) = vote_{x,y}\left(genesis, G\right) \geq 0, \tag{3}$$

そのため、$vote_{x,y}\left(virtual\left(G\right)\right) \geq 0$ である。

D. 安全性 (ブロック)の証明の概要

提案8では誠実なノードが観察するDAG内で十分な堅牢性を持っている場合には全てのノードは順序$x \prec y$の堅牢性に関して永久的に合意すると主張している。これは本証明の中核をなす、最も複雑な部分である。残りの提案はこれに従うものであり、これらの証明は自明である。証明が複雑かつ技術的であるため、その構造の概要から説明する。

分析を簡素化するために、攻撃者の行動に関して最悪の場合の想定を行う必要がある。補題20ではこれらが最悪の場合を想定したものであり、最適な攻撃を表すものであると証明している。我々が修正する各ブロックの投票は $p\_vote\left()$ と表記される。これに関してはサブセクションE.6で詳細に説明する。

次の補題では、$future\left(x\right)$ の合計投票が$x \prec y$に対して十分に好意的に偏っている場合、*genesis*ブロック、つまり仮想ブロック（補題13）は$x \prec y$と投票する。これは、大まかに言って最近の弱い投票者の投票はDAG内でカスケードが発生し、より古い弱いブロックを説得し、*genesis*の投票が形成されることを証明する。特定の弱い投票者z_{late}（*x*が誠実なブロックである場合、$z_{late} = x$）を選択し、その投票が(i)逆転することはなく、(ii) *genesis*までカスケードが発生することを保証するだけの十分な堅牢性があることを確認することでこれを証明する。結果として、攻撃が成功（誠実なノードが観察するDAG内の$x \prec y$の逆転）するために、攻撃者はある間隔内で誠実な

ネットワークが追加するよりも多くのブロックを $future\left(z_{late}\right)$ に追加する必要がある。
以下の補題ではこれらの観察を公式化している。誰も詳細を十分に把握していない一部のパラメータ（*h, j*等）を使用している。実務においてノードがこれらのパラメータにアクセスすることなくどのようにブロック関係の堅牢性を推測するかを説明する。
補題 14

$t \geq publication(x) + 2 \cdot d$である。$z_{late}$ は$\overline{past_h}\left(x\right)$の最新ブロックである。

- $h := \left|anticone_h\left(z_{late}, G_t^{oracle}\right)\right|$
- $j := \left|future_h\left(z_{late}, G_t^{oracle}\right) \setminus future_h\left(x, G_t^v\right)\right|$
- $m := \left|future_a\left(z_{late}, G_t^{oracle}\right) \setminus future_a\left(x, G_t^v\right)\right|$
- $k_1 := \left|G_{[t-d,t]}^{oracle} \cap honest\right|$
- $l := \max_{z \in G_t^{oracle} \cap honest} \left\{\left|future_a\left(z, G_{time(z_{late})}^u\right)\right| - \left|future_h\left(z, G_{time(z_{late})}^u\right)\right|\right\}$
- $g := \sum_{z \in \overline{future}(x, G_t^v)} vote_{y,x}\left(z, G_t^v\right)$

その場合、

$$\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)^{\complement} \subseteq \left\{\exists s \geq t, \exists u \in honest \text{ s.t. } \left|G_{[t,s]}^u \cap malicious\right| \geq \left|G_{[t,s]}^u \cap honest\right| + g - 2 \cdot h - j - k_1 - l - m\right\}.$$

前の補題の結果を考慮し、順序$x \prec y$が逆転する確率の上限を設定できる。おの結果はビットコインのセキュリティの従来の分析に類似している。

- x （そしてSPECTREでは$x \prec y$に投票）を現在ポイントするブロック数が多いほど、攻撃者がブロックカウント競争に勝利して決定を覆す可能性は低い。

補題15
補題14のパラメータを踏まえて

$$\Pr\left(\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)^{\complement}\right) \leq \sum_{h'=0}^{\infty} \mathcal{P}_{oiss}(d \cdot (1-\alpha) \cdot \lambda, h') \cdot \left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-j-k_1-l-m-h')^+}.$$

通常のノードは補題14で前提とするパラメータの値を把握していない。以下の結論ではこれらのパラメータを適切な限界で置き換えることによってこの補題の結果（及び以下の補題の結果）があてはまることを証明している。また、ノードがこのような限界を算出する方法についても説明する。

結論16

以下の場合

- $j \geq \left|anticone_h\left(x, G_t^{oracle}\right)\right|$
- $l \geq \max_{z \in G_t^{oracle} \cap honest} \left\{\left|future_a\left(z, G^u_{time(z_{late})}\right)\right| - \left|future_h\left(z, G^u_{time(z_{late})}\right)\right|\right\}$
- $n_x \geq future_h\left(x, G_t^{oracle}\right)$
- $g \leq \sum_{z \in \overline{future}(x, G_t^v)} vote_{y,x}\left(z, G_t^v\right)$.

以下の数式が導き出される。

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)\right) \leq \tag{4}$$
$$\sum_{k=0}^{\infty} \mathcal{P}_{oiss}((2-\alpha)\cdot d\cdot\lambda, k)\cdot \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d\cdot(1-\alpha)\cdot\lambda, h)\cdot$$
$$\sum_{m=0}^{\infty} \binom{n_x + j + h + m - 1}{m} \cdot (1-\alpha)^{n_x+j+h} \cdot \alpha^m \cdot$$
$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-k-j-l-m)^+}$$

$future\left(x, G\right)$ 内の一部のブロックが攻撃者に帰属することが判明している場合にあわせて上記の結果を調整する。この情報は理論上の預言者によって与えられたものと仮定する。後のセクションではアルゴリズム3が攻撃者のブロックを認識する方法について説明する。

結論17

結論16の前提に加えて $M \leq \left|future_a\left(x, G_t^v\right)\right|$ *と仮定する場合*、

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y) \mid \left|future_a\left(x, G_t^v\right)\right| \geq M\right) \leq$$
$$\sum_{k=0}^{\infty} \mathcal{P}_{oiss}((2-\alpha)\cdot d\cdot\lambda, k)\cdot \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d\cdot(1-\alpha)\cdot\lambda, h)\cdot \tag{5}$$
$$\left(\sum_{m'=M}^{\infty} \binom{n_x + j + h + m' - 1}{m'} \cdot (1-\alpha)^{n_x+j+h} \cdot \alpha^{m'}\right)^{-1} \cdot$$
$$\sum_{m=M}^{\infty} \binom{n_x + j + h + m - 1}{m} \cdot (1-\alpha)^{n_x+j+h} \cdot \alpha^m \cdot$$
$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-k-j-l-(m-M))^+} .$$

この不等号のRHSを $f_{post\_mine}\left(n_x, g, j, l, M\right)$ と表記する。

ここまで、我々の分析では補題14のパラメータに対する適切な限界が得られているという前提に立っている。補題24, 29及び31ではこれらのパラメータの適切な限界の設定

方法を示している。これらの各パラメータに関して、個別の誤差関数を定義し、正確な限界の役割を果たさない確率の上限を設定する。これらの誤差関数は補題25, 30及び32によって急速に劣化する。アルゴリズム3ではアウトプットする合計リスクにこれらの誤差関数を集計する。
パラメータ:

・l – *x*の公開前に攻撃者が得ているマイニング前のリード、誤差関数は $f_{pre\_mine}(l(G_t^v))$、サブセクションE.6.1で数値を計算。

・n_x – $future\,(x, G_t^v)$ 内の誠実なブロック数、誤差関数は $f_{post\_pub}\,(|future\,(x, G_t^v)|)$、結論29（不等号（52））内で定義

・j – *time*(*x*)後に作成される誠実なブロック数、誤差関数は$f_{pre_pub}\,(n_j(G_t^v))$、補題31（不等号（54））で定義

n_xは全ての誠実なブロックを適切にカウントしていることを証明したが、ここではn_xが全ての攻撃ブロックの排除に成功していることを証明している。このような保証がない場合、弱い攻撃者がブロックを公開して受諾を無限に遅延させることができてしまう。

補題18

事象$\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)$ *を条件とする。*$\tau \in [t,\infty)$ *という時間が存在し、*τ *後に残る*m^* *で*

$$\forall s \geq \tau\colon M(oracle^u, s) \geq \left|future_a\left(x, G_s^{oracle^u}\right) \cap G^{oracle}_{[t,s]} \setminus V_{x\prec y}(G_s^{oracle^u})\right| - m^*$$

となる（そして時間tまでの事象によって決定される$\mathbb{E}[m^*]$ *）。*

上記の分析（特に補題14）によって攻撃者が$x \prec y$の関係を逆転させられる確率の上限を設定している。順序が$x \prec y$のままであることを条件に、誤差関数f_{post_mine}（この順序が逆転する確率の上限を設定する）も消滅し、これらの順序が全てのノードによって堅牢であるとみなされることを証明できる。

補題19

$\psi \in [t,\infty)$が存在し、$\Pr\left(\mathcal{E}^{all}_{t\to\infty}(x,y,\epsilon)^{\complement} \mid \mathcal{E}^v_t(x,y,\epsilon)\right) < \epsilon$ となる。さらに、$\mathbb{E}[\psi - t] < \epsilon$である。

誠実なノードによるアルゴリズム3のアウトプットがϵよりも小さい場合、少なくとも$1-\epsilon$の確率でアルゴリズム3（一定期間経過後）を実行するあらゆる誠実なノードでϵよりも小さい結果が出る[9]。これによって安全性ブロックの証明が完了する。

E. 整合性の証明

証明

パートI: 最初にDAG *G*と$tx_1, tx_2 \in T\colon$において、$tx_2 \in inputs\,(tx_1)$ と

$[tx_1] \cap TxO(G) \neq \emptyset$ の場合、$[tx_2] \cap TxO(G) \neq \emptyset$ であることを証明する。
$tx_2 \in inputs\,(tx_1)$ 及び $[tx_1] \cap TxO(G,G) \neq \emptyset$ と仮定し、
$tx_1 \in [tx_1] \cap TxO(G,G)$ とする。
2つ目のループ（4行目）の反復が $tx = tx_1$ 上で発生している場合を考える。
$tx \in TxO(G,G)$ であるため、この反復中にアルゴリズムが14行目に到達している。これはあらゆる $[tx_3] \in inputs\,(tx_1)$ で、13行目に到達していないことを意味する。特に $[tx_3] = [tx_2]$ の場合、条件 $[tx_2] \cap TxO\,(G, past\,(z_1)) = \emptyset$ が失敗している。つまり、$[tx_2] \cap TxO\,(G, past\,(z_1)) \neq \emptyset$ である。
$TxO\,(G, past\,(z_1)) \subseteq TxO\,(G,G)$ が（i）アルゴリズム実行中に TX からトランザクションが削除されないこと、そして（ii）$z_1 \in G \cap subG$ において、$TxO\,(G, subG)$ と $TxO\,(G,G)$ の運用（4-14行）が同一であることを遵守しているため、$TxO\,(G, subG)$ の14行目のトランザクションの追加は $TxO\,(G,G)$ でも発生している。特に、$[tx_2] \cap TxO\,(G,G) \neq \emptyset$ である。
パートII: DAG G と $tx_1, tx_2 \in T$ において、$tx_2 \in conflict\,(tx_1)$ 及び $[tx_1] \cap TxO(G,G) \neq \emptyset$ の場合、$[tx_2] \cap TxO(G) = \emptyset$ である。
$tx_2 \in conflict\,(tx_1)$ 及び $[tx_1] \cap TxO(G,G) \neq \emptyset$ と仮定し、tx_1 は後者の共通部分内の要素とする。否定によって $tx_2 \in [tx_2] \cap TxO(G,G)$ が存在すると仮定する。z_1 の z_1^1 上での最初のループ（3行目）の反復中に $tx_1 \in z_1^1$ となり、tx_1 上の2回目のループ（4行目）の反復中にアルゴリズムが14行目に到達している。特に、10行目に到達していないために $z_1^2 \notin past\,(z_1^1)$ となる。
対称引数の場合、$z_1^1 \notin past\,(z_1^2)$ であり、$z_1^2 \in anticone\,(z_1^1, G)$ （及び $z_1^2 \in anticone\,(z_1^1, G)$）であることを示唆している。従って、$vote_{z_1^1, z_1^2}\,(virtual\,(G)) \geq 0$ もしくは $vote_{z_1^2, z_1^1}\,(virtual\,(G)) \geq 0$ である。いずれの場合も、tx_1 上、もしくは tx_2 上で8行目に達しており、両方で14行目に達したという前提に矛盾している。

F. 安全性の証明 (ブロック)

分析を簡素化するために、攻撃者の行動に関して最悪の場合の想定を行う必要がある。つまり、攻撃者は時間 t 直後にブロックを全て公開し（誠実なノードがトランザクションを受諾した時間を表す）、$time(x)$ 前に攻撃者のブロックが利用可能な全てのブロックをポイントすると仮定する必要がある。これらの仮定によってDAGが修正される（攻撃者が最適な攻撃を実施しない場合）。これらの修正が最悪のケースを表現していることを証明する必要がある。このために、我々は疑似投票という概念を用いる。疑似投票は我々が初回疑似投票者と呼ぶ一部のブロックの疑似投票を明確に定

義して固定することによって開始される。
9ϵ はここでは単純に $f_{pre\ mine} + f_{pre\ pub} + f_{post\ pub} + f_{post\ mine}$ よりも大きい値を表す。

その後、アルゴリズム1と同様に残りのブロックの疑似投票を定義する。より詳細に説明すると、アルゴリズム1では $vote()$ を $p\_vote()$ で置き換え、アルゴリズムが初回疑似投票者 c の $p\_vote(c)$ を参照する場合はその固定された事前決定値を参照する。そのため、初回疑似投票者の疑似投票によってその他のブロックの疑似投票が変化する可能性がある。
疑似命題20

$x, y \in G = (C, E)$ であり $G_t^v \subseteq G$ とする。 $G' = (C, E')$ は以下のエッジをEに追加することによって発生するDAGとする。
1)
$\forall z_1 \in G \cap before(time(x)) \cap malicious,\ \forall z_2 \in G \cap before(time(z_1)) \setminus \{z_1\}:$
(z_1, z_2) をEに追加する。.
2) $\forall z_1 \in G \cap malicious \setminus G_t^v,\ \forall z_2 \in G^{oracle}_{[publication(z_1),\infty)} \cap honest:$ 、
(z_2, z_1) をEに追加する。

以下の疑似投票者（そして彼らの投票）を指定することで $p\_vote()$ を定義する。
3)
$\forall z \in (G \cap malicious \setminus G_t^v) \cup G \cap before(time(x)) \cap malicious:\ p\_vote_{x,y}(z, G) = +1$
その場合、 $vote_{x,y}(virtual(G), G) \leq p\_vote_{x,y}(virtual(G'), G')$.[10]

重要なことは、ここでは $G \cap malicious$ 内のブロックは $y \prec x$ に対して好意的にタイブレークを行うと仮定している点である。
証明

パート *I:* 否定によって $vote_{x,y}(virtual(G), G) = +1$ でありながら、 $p\_vote_{x,y}(virtual(G), G') = -1$ と仮定する（仮想投票は0の値をとることができないため、この場合にのみ主張が失敗する可能性がある）。
b は $\overline{future}(x, G) \cup \{virtual(G)\}$ 内のブロックであり、
$p\_vote_{x,y}(b, G') = -1$ とする。b は $G \setminus G_t^v$ や $G \cap before(time(x)) \cap malicious$ に帰属することはできない。なぜなら、これらのセット内のブロックの疑似投票は+1だからである。z は $past(b, G)$ 内のブロックである。 $b \notin (G \setminus G_t^v) \cup (G \cap before(time(x)) \cap malicious)$ であるため、G' 内にエッジ (z_2, z_1) を通過する b から z の経路が存在し、G の2つ目の修正に対する条件が満たされ、エッジ (z_1', z_2') を通過する場合には1回目の修正の条件

が満たされる。特に、$time(z_2) \geq publication(z_1) \geq t - d$であり
$time(x) \geq time(z_2') \geq time(z_1')$である。*b*と*z*はこの経路の頂点であるため、
$time(b) \geq time(z_2) \geq t-d \geq publication(x)+d \geq time(x)+d \geq time(z_1')+d \geq time(z)+d$
である。
*z*2は誠実であるため、$z \in past\,(z_2, G)$であり、$z \in past\,(b, G)$である。$E \subset E'$
と組み合わせることで $past\,(b, G') = past\,(b, G)$が導き出される。
パートII:
*b*は$\overline{future}\,(x, G) \cup \{virtual\,(G)\}$ *内の最初のブロックであり、*
$vote_{x,y}\,(b, G) = +1$ *であるが*$p\_vote_{x,y}\,(b, G') = -1$ *であり、zは*
$antifuture\,(x, G')$ 内の最後のブロックであり、
$vote_{x,y}\,(z, past\,(b, G)) > p_vote_{x,y}\,(z, past\,(b, G'))$である。そのような*z*が
存在する場合、前のパートと同じように、$past\,(z, G') = past\,(z, G);$であることが判明する。これによって*z*が*G*と*G'* の両方で弱い投票者であることが証明されるため、その疑似投票は未来における疑似投票の合計を示すものである。[11]
そのような*z*が存在することを確認するために、*genesis*が次の条件を満たしていることを確認する。
補題13
$vote_{x,y}\,(b, G) = vote_{x,y}\,(virtual\,(past\,(b, G))) = +1$ *は*
$vote_{x,y}\,(genesis, past\,(b, G)) \geq 0$ *を示唆し、同様に*$p\_vote_{x,y}\,(b, G') = p\_vote_{x,y}\,(virtual\,(past\,(b, G'))) = -1$ *は*
$p\_vote_{x,y}\,(genesis, past\,(b, G')) = -1.$[12] *を示唆する。*

[10]同じ頂点セットを共有するために$virtual\,(G) = virtual\,(G')$である点に注意が必要である。
[11]*z*は初回疑似投票者ではなく（疑似投票は-1であるため）、過去に*y*は存在するが*x*は存在しないため疑似投票手順では疑似投票を+1に割り当てる、そのため$y \in past\,(z, G')$ ではありえない。.

パートIII:
zの選択によって、$z' \in future\,(z, past\,(b, G'))$ *が*(x, y) *に関して弱い場合、*
$vote_{x,y}\,(z', past\,(b, G)) \leq p\_vote_{x,y}\,(z', past\,(b, G'))$*となる。さらに、bの選択によって、*$(x, y)$ *に関して*$z' \in past\,(b, G')$ *が強く、*
$vote_{x,y}\,(z', past\,(b, G)) = +1$ *である場合、*
$p\_vote_{x,y}\,(z', past\,(b, G')) = +1$*となる。全ての*

$z' \in future\left(z, past\left(b, G'\right)\right)$ *に関して、*
$vote_{x,y}\left(z', past\left(b, G\right)\right) \leq p\_vote_{x,y}\left(z', past\left(b, G'\right)\right)$ *である。*
従って

$$\sum_{z' \in future(z, past(b,G))} vote_{x,y}\left(z', past\left(b, G\right)\right) \leq \tag{6}$$

$$\sum_{z' \in future(z, past(b,G))} p\_vote_{x,y}\left(z', past\left(b, G'\right)\right) \leq \tag{7}$$

$$\sum_{z' \in future(z, past(b,G))} p\_vote_{x,y}\left(z', past\left(b, G'\right)\right) +$$

$$\sum_{z' \in future(z, past(b,G') \setminus past(b,G))} p\_vote_{x,y}\left(z', past\left(b, G'\right)\right) = \tag{8}$$

$$\sum_{z' \in future(z, past(b,G'))} p\_vote_{x,y}\left(z', past\left(b, G'\right)\right). \tag{9}$$

最後の等号は$future\left(z, past\left(b, G\right)\right) \subseteq future\left(z, past\left(b, G'\right)\right)$から派生し、$E \subseteq E'$であるために成立する。（*G*を*G′* に変換することにより）誠実なブロックの未来に*z′* が追加される場合、*z′* は攻撃者に帰属するため、(7)の不等号は成立する。そのため、 $p\_vote_{x,y}\left(z', past\left(b, G'\right)\right) = +1 > 0.$ となる。
パートIV:

結果として、*z*は (x, y)に関して弱い投票者であるため、(6)-(9) は
$vote_{x,y}\left(z, past\left(b, G\right)\right) \leq p\_vote_{x,y}\left(z, past\left(b, G'\right)\right)$ を示唆し、*z*の選択と矛盾する。
補題 14

$t \geq publication(x) + 2 \cdot d$とする。$z_{late}$を$\overline{past_h}\left(x\right)$内の最新のブロックとする。

- $h := \left|anticone_h\left(z_{late}, G_t^{oracle}\right)\right|$
- $j := \left|future_h\left(z_{late}, G_t^{oracle}\right) \setminus future_h\left(x, G_t^v\right)\right|$
- $m := \left|future_a\left(z_{late}, G_t^{oracle}\right) \setminus future_a\left(x, G_t^v\right)\right|$
- $k_1 := \left|G_{[t-d,t]}^{oracle} \cap honest\right|$
- $l := \max_{z \in G_t^{oracle} \cap honest} \left\{\left|future_a\left(z, G_{time(z_{late})}^u\right)\right| - \left|future_h\left(z, G_{time(z_{late})}^u\right)\right|\right\}$
- $g := \sum_{z \in \overline{future}(x, G_t^v)} vote_{y,x}\left(z, G_t^v\right)$

[12] *b*は (x, y) に関して強い投票者である、もしくは仮想投票者であるために
$p\_vote_{x,y}\left(b, G'\right) = p\_vote_{x,y}\left(virtual\left(past\left(b, G'\right)\right)\right)$ という等号が成り立つ。

その場合、

$$\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement} \subseteq \Big\{\exists s \geq t, \exists u \in honest \ s.t. \ \left|G^u_{[t,s]} \cap malicious\right| \geq \left|G^u_{[t,s]} \cap honest\right| + g - 2\cdot h - j - k_1 - l - m\Big\}. \quad (10)$$

証明

パートI:

以下の証明では次のような仮定を行っている。$time(x)$ 前に作成した攻撃者ブロック z は常に（アルゴリズム1に従い別の投票を行うことになっている場合にも）$y \prec x$ に好意的に投票する。さらに、そのような*z*は $\overline{past}\,(z) = G^{oracle}_{time(z)}$ を満たしているものと仮定する。つまり、作成時点で利用可能な全てのブロックをポイントする。最後に、攻撃者が*t*以降の時点で $honest \backslash \{v\}$ 内の全てのノードに全てのブロックを公開すると仮定する。前の補題ではこれらが最悪の事態を想定したものであることを示唆している。*G*を*Gus*と考える。その場合、補題は $p\_vote_{x,y}\,(virtual\,(G^u_s)) = -1$ である限り、常に $vote_{x,y}\,(virtual\,(G^u_s)) = -1$ であることを証明している（最悪の事態の想定では常に*y*に対して有利なタイブレークが行われる）[13]。前の補題（特に(3)）で公式化しているように以下の分析が $p\_vote\,()$ に対して適用される。
上記の内容に関わらず、引数が正式に作成されたため、今後はこの表記を省略する。

パートII: 以下の論理包含の連鎖を見てみよう。

$$vote_{x,y}\,(virtual\,(G^u_s)) \geq 0 \Rightarrow vote_{x,y}\,(genesis, G^u_s) \geq 0 \Rightarrow \sum_{z' \in future(genesis, G^u_s)} vote_{x,y}\,(z', G^u_s) \geq 0 \quad (11)$$

最初の論理包含は補題13に従う。2番目は*genesis*の投票の定義に従う[14]。そのため、

$$\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement} = \cup_{u\in honest, s\in[t,\infty)} \widehat{\mathcal{E}}^u_s(x,y)^{\complement} = \{\exists u \in honest, \exists s \geq t : vote_{x,y}\,(virtual\,(G^u_s)) \geq 0\}$$

となる。

ただし、後者の事象が要求するような*s*が存在する場合、そのような最初の*s*を確認できる。これに関して、*t*と*s*の間で全ての誠実な投票は*x* ≺ *y*に対して好意的である。これは $time(z') \in [t,s)$ 及び $past\,(z') = G^{node(z')}_{time(z')}$ のあらゆる誠実なブロック*z′*において、$vote\,(z') = vote\left(virtual\left(G^{node(z')}_{time(z')}\right)\right)$ であり、誠実なノードのDAGの仮想ブロックが*y* ≼ *x*に対して好意的に投票する最初のタイミングとして*s*を選択することで、$vote_{x,y}\,(z', G^u_s) = -1$ であることを把握できる。

パートIII: 以下では、 $G^u_{[t_1,t_2]}$ *は* $G^u_s \cap before\,(t_2) \setminus before\,(t_1)$ *を意味する。*

[13]実際、我々はこの補題に若干の修正を加えた上で使用している。2回目の修正は指定された条件を満たす全ての (z_2, z_1) ではなく、そのサブセットに適用される。これはノード*v*と*t*の間及び*t* + *d*の間で作成さ

れたブロックは $G \setminus G_s^v$ 内の全ての攻撃者ブロックをポイントする必要がないからである。ただし、補題の証明に影響がないことは明らかである（そして、$(G \cap malicious \setminus G_t^v) \times (G^{oracle}_{[publication(z_1),\infty)} \cap honest))$ のサブセットに2回目の修正を適用した場合にもこの事は変わらない）。

14ここでは、*x*と*y*がトポロジー的に関連していないと仮定しているため、$x = genesis$ もしくは $y = genesis$ という選択肢が除外さる。そのため、*genesis*は (x, y) に関して弱い投票者である。*x*と*y*がトポロジー的に関連している場合、全ての投票は永久的に同じ方向性で行われるために結果は無関係である (補題11)。

全ての $z \in \overline{past_h}(x)$ に関して

$$vote_{x,y}(z, G_s^u) \leq \widetilde{sgn}\left(\left|G^u_{[t,s]} \cap malicious\right| - \left|G^u_{[t,s]} \cap honest\right| + 2 \cdot h + l + k_1 + j + m - g\right).$$

$D(z) := \left|future\left(z, \overline{past_h}(x)\right)\right|$ に対する数学的帰納法によってこの主張を証明する。

D(*z*) < *D*である*z*の主張を証明していると仮定する。これを*D*(*z*) = *D*の*z*で証明する。

$z = x$ の場合、$vote_{x,y}(z, G_s^u) = -1$ であるため上記の不等号が証明される。

それ以外の場合、*z*は弱い投票者であり、未来の投票の合計によって $vote_{x,y}(z, G_s^u)$ が成立する。我々はこのような投票者を次の3つのサブセットに分解する: $future\left(z, G^u_{time(z_{late})}\right)$ のメンバー、 $future\left(z, G^u_{[time(z_{late}),t]}\right)$ のメンバー及び $future\left(z, G^u_{[t,s]}\right)$ のメンバー

1) $future\left(z, G^u_{time(z_{late})}\right)$ のメンバー：帰納法の仮定によって、$future_h\left(z, \overline{past_h}(z_{late})\right)$ 内の全てのブロックが*x* ≺ *y*に投票し、z_{late} の選択によって $future_h\left(z, G^u_{time(z_{late})}\right) \setminus \overline{past}(z_{late}) = anticone_h\left(z, G^u_{time(z_{late})}\right)$ となる。

そのため、$\sum_{z' \in future_h\left(z, G^u_{time(z_{late})}\right)} vote_{x,y}(z', G_s^u) \leq 2 \cdot \left|anticone_h\left(z_{late}, G^u_{time(z_{late})}\right)\right| - \left|future_h\left(z, G^u_{time(z_{late})}\right)\right|$ である。

以下の結果が導き出される。

$$\sum_{z' \in future\left(z, G^u_{time(z_{late})}\right)} vote_{x,y}\left(z', G^u_s\right) \leq$$
$$2 \cdot \left|anticone_h\left(z_{late}, G^u_{time(z_{late})}\right)\right|$$
$$- \left|future_h\left(z, G^u_{time(z_{late})}\right)\right| + \left|future_a\left(z, G^u_{time(z_{late})}\right)\right|$$

2) $future\left(z, G^u_{[time(z_{late}),t]}\right)$ のメンバー：

a)誠実なブロック: パート I では $future_a\left(z, G^u_{[time(z_{late}),t]}\right) \setminus future_a\left(z_{late}, G^u_{[time(z_{late}),t]}\right) = \emptyset$ を証明する。これは、$anticone_h\left(z_{late}, G^u_{[time(z_{late}),time(z_{late})+d]}\right) \supseteq future\left(z, G^u_{[time(z_{late}),t]}\right) \setminus future\left(z_{late}, G^u_t\right)$ であることを示唆する。

以下が導き出される。

$$\sum_{z' \in future_h\left(z, G^u_{[time(z_{late}),t]}\right)} vote_{x,y}\left(z', G^u_s\right) \leq$$
$$\sum_{z' \in future_h(z_{late}, G^u_t)} vote_{x,y}\left(z', G^u_s\right) +$$
$$\left|anticone_h\left(z_{late}, G^u_{[time(z_{late}),time(z_{late})+d]}\right)\right| \leq$$
$$\sum_{z' \in future_h(z_{late}, G^v_t)} vote_{x,y}\left(z', G^u_s\right) +$$
$$\left|anticone_h\left(z_{late}, G^u_{[time(z_{late}),time(z_{late})+d]}\right)\right| +$$
$$\left|G^{oracle}_{[t-d,t]} \cap honest\right| \leq$$
$$\sum_{z' \in future_h(x, G^v_t)} vote_{x,y}\left(z', G^u_s\right) +$$
$$\left|anticone_h\left(z_{late}, G^u_{[time(z_{late}),time(z_{late})+d]}\right)\right| +$$
$$\left|G^{oracle}_{[t-d,t]} \cap honest\right| +$$
$$\left|future_h\left(z_{late}, G^v_t\right) \setminus future_h\left(x, G^v_t\right)\right|.$$

b) 攻撃者のブロック: パートIで説明した最悪の場合を想定することで以下を導き出す。

$$\sum_{z' \in future_a\left(z, G^u_{[time(z_{late}),t]}\right)} vote_{x,y}\left(z', G^u_s\right) =$$

$$\sum_{z' \in future_a\left(z_{late}, G^u_{[time(z_{late}),t]}\right)} vote_{x,y}\left(z', G^u_s\right) \leq$$

$$\sum_{z' \in future_a(x, G^v_t)} vote_{x,y}\left(z', G^u_s\right) +$$

$$\left|future_a\left(z_{late}, G^u_t\right) \setminus future_a\left(x, G^v_t\right)\right|.$$

c) 全てのブロック： $future\left(z, G^u_{[time(z_{late}),t]}\right)$ 内の誠実なブロックと攻撃者のブロックを組み合わせることで以下を導き出す。

$$\sum_{z' \in future\left(z, G^u_{[time(z_{late}),t]}\right)} vote_{x,y}\left(z', G^u_s\right) \leq$$

$$\sum_{z' \in future_h(x, G^v_t)} vote_{x,y}\left(z', G^u_s\right) +$$

$$\left|anticone_h\left(z_{late}, G^u_{[time(z_{late}),time(z_{late})+d]}\right)\right| +$$

$$\left|G^{oracle}_{[t-d,t]} \cap honest\right| + \left|future_h\left(z_{late}, G^v_t\right) \setminus future_h\left(x, G^v_t\right)\right|$$

$$+ \sum_{z' \in future_a(x, G^v_t)} vote_{x,y}\left(z', G^u_s\right) +$$

$$\left|future_a\left(z_{late}, G^u_t\right) \setminus future_a\left(x, G^v_t\right)\right| = \quad (12)$$

$$g + \left|anticone_h\left(z_{late}, G^u_{[time(z_{late}),time(z_{late})+d]}\right)\right| + \quad (13)$$

$$\left|G^{oracle}_{[t-d,t]} \cap honest\right| + \left|future_h\left(z_{late}, G^v_t\right) \setminus future_h\left(x, G^v_t\right)\right|$$

$$+ \left|future_a\left(z_{late}, G^u_t\right) \setminus future_a\left(x, G^v_t\right)\right|.$$

3) $future\left(z, G^u_{[t,s]}\right)$ のメンバー：最後に、sを選択することでtとsの間で作成された全ての誠実なブロックが$x \prec y$に投票するため、以下が導き出される。

$$\sum_{z' \in future\left(z, G^u_{[t,s]}\right)} vote_{x,y}\left(z', G^u_s\right) \leq$$
$$- \left|future_h\left(z, G^u_{[t,s]}\right)\right| + \left|future_a\left(z, G^u_{[t,s]}\right)\right| \leq$$
$$- \left|G^u_{[t,s]} \cap honest\right| + \left|G^u_{[t,s]} \cap malicious\right|,$$

ここでも $t \geq publication(x) + d \geq publication(z) + d$ であるという事実を利用している。

4) 上記全ての結果を組み合わせることで以下を導き出す。

$$\sum_{z' \in future(z, G^u_s)} vote_{x,y}\left(z', G^u_s\right) \leq \tag{14}$$
$$2 \cdot \left|anticone_h\left(z_{late}, G^u_{time(z_{late})}\right)\right|$$
$$- \left|future_h\left(z, G^u_{time(z_{late})}\right)\right| + \left|future_a\left(z, G^u_{time(z_{late})}\right)\right| \tag{15}$$
$$+ g + \left|anticone_h\left(z_{late}, G^u_{[time(z_{late}), time(z_{late})+d]}\right)\right| +$$
$$\left|G^{oracle}_{[t-d,t]} \cap honest\right| +$$
$$\left|future_h\left(z_{late}, G^v_t\right) \setminus future_h\left(x, G^v_t\right)\right| +$$
$$\left|future_a\left(z_{late}, G^u_t\right) \setminus future_a\left(x, G^v_t\right)\right| + \tag{16}$$
$$- \left|G^u_{[t,s]} \cap honest\right| + \left|G^u_{[t,s]} \cap malicious\right| \leq \tag{17}$$

$$\tag{18}$$
$$2 \cdot h + l + k_1 - g + \left|G^u_{[t,s]} \cap malicious\right| - \left|G^u_{[t,s]} \cap honest\right|$$
$$+ \left|future_h\left(z_{late}, G^v_t\right) \setminus future_h\left(x, G^v_t\right)\right| +$$
$$\left|future_a\left(z_{late}, G^u_t\right) \setminus future_a\left(x, G^v_t\right)\right| \leq$$
$$2 \cdot h + l + k_1 - g + \left|G^u_{[t,s]} \cap malicious\right| - \left|G^u_{[t,s]} \cap honest\right|$$
$$+ \left|future_h\left(z_{late}, G^{oracle}_t\right) \setminus future_h\left(x, G^v_t\right)\right| + \tag{19}$$
$$\left|future_a\left(z_{late}, G^{oracle}_t\right) \setminus future_a\left(x, G^v_t\right)\right| = \tag{20}$$
$$2 \cdot h + l + k_1 - g + j + m + \left|G^u_{[t,s]} \cap malicious\right| \tag{21}$$
$$- \left|G^u_{[t,s]} \cap honest\right|.$$

$$vote_{x,y}\left(z, G^u_s\right) \leq \widetilde{sgn}\left(\left|G^u_{[t,s]} \cap malicious\right| - \left|G^u_{[t,s]} \cap honest\right| + 2 \cdot h + l + k_1 + g + j + m\right)$$

*z*は弱い投票者であるため、と結論付ける。

パート*IV:* 特に、*z* = *genesis*の場合、 $vote_{x,y}\left(genesis, G_s^u\right) \geq 0$ の事象は $\left|G_{[t,s]}^u \cap malicious\right| \geq \left|G_{[t,s]}^u \cap honest\right| - 2 \cdot h - l - k_1 - g - j - m$ の事象内に含まれる。(11)までに、この事象には全ての $u \in honest$ と $s \geq t$ の $\widehat{\mathcal{E}}_u^s(x,y)^{\complement}$ も含まれるため、$\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)^{\complement}$ も含まれる。

補題15

補題14のパラメータを前提とする。

$$\Pr\left(\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)^{\complement}\right) \leq \sum_{h'=0}^{\infty} \mathcal{P}_{oiss}(d\cdot\alpha\cdot\lambda, h')\cdot \left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-j-k_1-l-m-h')^+} \tag{22}$$

証明

全てのノード $u \in honest$ が遅くとも*d*秒の遅延で誠実なブロックを受信するため、

$\left|G_{[t,s]}^u \cap malicious\right| - \left|G_{[t,s]}^u \cap honest\right| \leq \left|future_a\left(z_{late}, G_{[t,s]}^{oracle}\right)\right| - \left|future_h\left(z_{late}, G_{[t,\max\{s-d,t\}]}^{oracle}\right)\right|$ となる。さらに、$\left|future_a\left(x, G_{[s-d,s]}^{oracle}\right)\right|$ によって $\left|future_a\left(x, G_{[s',s]}^{oracle}\right)\right|$ の上限を設定し、後者がパラメータ $\alpha\cdot d\cdot\lambda$ でポアソン分布に従うことを観察する。この変数を h' と表記する。 h' のあらゆる値で変数 $\left|future_a\left(x, G_{s'}^{oracle} \setminus G_t^{oracle}\right)\right| - \left|future_h\left(x, G_{s'}^{oracle} \setminus G_t^{oracle}\right)\right| + h'$ をランダムウォーク*Xi*（*i*番目のステップが時間*t*経過後の*i*番目のブロックの作成時間）としてモデリングし、α は正の無限大に流れていく。 $g > h + j + k_1 + l + m + h'$ の場合に*X*が間隔 $[-h-j-k_1-h'-l-m+g, +\infty)$ に到達する確率は $\left(\frac{\alpha}{1-\alpha}\right)^{g-2\cdot h-j-k_1-l-m-h'}$ であり、それ以外の場合は1である ([18], [17]を参照)。

補題16

- $j \geq \left|anticone_h\left(x, G_t^{oracle}\right)\right|$
- $l \geq \max_{z \in G_t^{oracle} \cap honest}\left\{A^z_{time(x)} - H^z_{time(x)}\right\}$
- $n_x \geq future_h\left(x, G_t^{oracle}\right)$
- $g \leq \sum_{z \in \overline{future}(x, G_t^v)} vote_{y,x}\left(z, G_t^v\right).$

上記の場合

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)\right) \leq \tag{23}$$

$$\sum_{k=0}^{\infty}\mathcal{P}_{oiss}((2-\alpha)\cdot d\cdot\lambda, k)\cdot\sum_{h=0}^{\infty}\mathcal{P}_{oiss}(d\cdot(1-\alpha)\cdot\lambda, h)\cdot$$

$$\sum_{m=0}^{\infty}\binom{n_x+j+h+m-1}{m}\cdot(1-\alpha)^{n_x+j+h}\cdot\alpha^m\cdot$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-k-j-l-m)^+}$$

証明

前の補題の結果を基礎としている。これらの結果は対応するパラメータが限界の役割を果たす場合には影響を受けないままである。(14)-(21)を参照。

変数k_1, $\left|anticone_h\left(z_{late}, G^u_{[time(z_{late}), time(z_{late})+d]}\right)\right|$と$h'$は独立したポアソン過程の合計である。最初の2つのパラメータは$d\cdot(1-\alpha)\cdot\lambda$と$h'$ is $d\cdot\alpha\cdot\lambda$である。そのため、これらの合計は新しいポアソン変数*k*で、パラメータは$(2\cdot(1-\alpha)+\alpha)\cdot d\cdot\lambda = (2-\alpha)\cdot d\cdot\lambda$である。変数$\left|anticone_h\left(z_{late}, G^u_{time(z_{late})}\right)\right|$は追加のポアソン変数でパラメータは$d\cdot(1-\alpha)\cdot\lambda$である。我々はこれを$h$と表記する（これによって補題14の元の意味を置き換える）。

補題14では上限$\left|future_a\left(z_{late}, G_t^{oracle}\right)\right|$の変数$m = \left|future_a\left(z_{late}, G_t^{oracle}\right) \setminus future_a\left(x, G_t^v\right)\right|$を使用している。誠実なネットワークが$z_{late}$の作成以来*n*個のブロックを作成している場合には、同時に攻撃者が作成するブロック数は負の二項分布に従う（[17]を参照）。つまり、値は*m*で、確率は$\binom{n+m-1}{m}\cdot(1-\alpha)^n\cdot\alpha^m$である。最悪のケースではこれら全てのブロックが$future\left(z_{late}, G_t^{oracle}\right)$に属する。ここでも、パラメータ*n*を増加させることによって確率的にオリジナルを支配する*m*上での分布が発生するために*n*の上限を設定するだけで十分である。$time(z_{late})$後（*t*まで）に作成されるブロック数はブロックが

$past_h\left(z_{late}\right)$ 内にあるため $antipast_h\left(z_{late}, G_t^{oracle}\right)$ によって上限を設定される。
そのため、
$n \leq |anticone_h\left(z_{late}, G_t^v\right)| + |future_h\left(z_{late}, G_t^v\right) \setminus future_h\left(x, G_t^v\right)| + |future_h\left(x, G_t^v\right)| \leq h + j + n_x$ となる。
最後に、*l*と*j*は補題14の対応する変数の上限、*g*は下限であるため、証明内（そして補題15の証明内）の全ての等号を“$\leq$”に変換することが可能であり、証明は影響を受けない。
以下ではが誠実なブロックであることが判明している場合、そして個別のブロックではなくブロックのグループを保護する必要がある場合に関して以前の結果を再検証している。
補題21
$node(x) \in honest$ *及び* $publication(y) \geq publication(x) + d.$ *と仮定する。* z_{late} *は* $\overline{anticone_h}\left(x, G_t^v\right)$ *内の最新のブロックであり、* z_{early} *は* $\overline{anticone_h}\left(x, G_t^v\right)$ *内の最初のブロックである。*
さらに、以下の内容を仮定する。

- $l := \max_{z \in G_t^{oracle} \cap honest} \left\{ \left| future_a\left(z, G^u_{time(z_{early})}\right) \right| - \left| future_h\left(z, G^u_{time(z_{early})}\right) \right| \right\}$
- $n_x \geq \max_{x' \in \overline{anticone_h}(x)} \left\{ \left| future_h\left(x', G_t^{oracle}\right) \right| \right\}$
- $g \leq \max_{x' \in \overline{anticone_h}(x, G_t^v)} \left\{ z \in \overline{future}\left(x', G_t^v\right) : vote_{y,x}\left(z, G_t^v\right) = -1 \right\} - \min_{x' \in \overline{anticone_h}(x, G_t^v)} \left\{ z \in \overline{future}\left(x', G_t^v\right) : vote_{y,x}\left(z, G_t^v\right) = +1 \right\}.$

この場合、

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t \to \infty}(x,y)^{\complement}\right) \leq \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d \cdot \lambda, h) \cdot \tag{24}$$

$$\sum_{m=0}^{\infty} \binom{n_x + m - 1}{m} \cdot (1-\alpha)^{n_x} \cdot \alpha^m. \tag{25}$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-h-l-m)^+}.$$

証明
$k_1 \geq \left| G^{oracle}_{[t-d,t]} \cap honest \right|$ 及び
$m := \left| future_a\left(z_{early}, G_t^{oracle}\right) \setminus future_a\left(z_{late}, G_t^v\right) \right|$ とする。補題14の証明の分析内容を調整する。全ての $z \in \overline{anticone_h}\left(x, G_t^v\right)$ に対して以下を主張する。

$$vote_{x,y}\left(z, G_s^u\right) \leq \widetilde{sgn}\left(\left|G_{[t,s]}^u \cap malicious\right| - \left|G_{[t,s]}^u \cap honest\right| \right.$$
$$\left. l + k_1 + m - g\right).$$

$D(z) := \left|future\left(z, \overline{anticone_h}\left(x\right)\right)\right|$ の数学的帰納法によって主張を証明する。$D(z) < D$のzにおいて主張を証明しているものと仮定する。$D(z) = D$のzでも主張を証明する。 $z = x$ の場合、$vote_{x,y}\left(z, G_s^u\right) = -1$であるため、上記の不等号が満たされる。それ以外の場合、$publication(y)$の仮定によって$y \notin \overline{past}\left(z\right)$ が成り立つためにzは弱い投票者である。そのため、$vote_{x,y}\left(z, G_s^u\right)$ は未来の投票の合計によって決定される。我々はこれらの投票者を以下の3つのサブセットに分解する。
$future\left(z, G_{time(z_{late})}^u\right)$のメンバー、$future\left(z, G_{[time(z_{late}),t]}^u\right)$のメンバー及び
$future\left(z, G_{[t,s]}^u\right)$のメンバー

1) $future\left(z, G_{time(z_{late})}^u\right)$のメンバー： 帰納法の仮定によって
$future_h\left(z, \overline{anticone_h}\left(z_{late}\right)\right)$内の全ブロックが$x \prec y$に投票することが判明しているため、以下が導き出される。

$$\sum_{z' \in future\left(z, G_{time(z_{late})}^u\right)} vote_{x,y}\left(z', G_s^u\right) \leq$$
$$-\left|future_h\left(z, G_{time(z_{late})}^u\right)\right| + \left|future_a\left(z, G_{time(z_{late})}^u\right)\right|.$$

2) $future\left(z, G_{[time(z_{late}),t]}^u\right)$のメンバー：このセット内の全ての$z'$ は
$x' \in \overline{anticone_h}\left(x, G_t^v\right)$で$future\left(x'\right)$ に属する。そのため、gの定義により以下が導き出される。

$$\sum_{z' \in future\left(z, G_{[time(z_{late}),t]}^u\right)} vote_{x,y}\left(z', G_s^u\right) \leq$$
$$-g + \left|future_a\left(z_{early}, G_t^{oracle}\right) \setminus future_a\left(z_{late}, G_t^v\right)\right| +$$
$$\left|future_h\left(z, G_t^u\right) \setminus future_h\left(z, G_t^v\right)\right| \leq \qquad (26)$$
$$-g + \left|future_a\left(z_{early}, G_t^{oracle}\right) \setminus future_a\left(z_{late}, G_t^v\right)\right| + \left|G_{[t-d,t]}^{oracle} \cap honest\right| =$$
$$-g + m + k_1.$$

3) $future\left(z, G_{[t,s]}^u\right)$のメンバー：$s$の選択によって$t$と$s$の間に作成された全ての誠

実なブロックは$x \prec y$に投票する。そのため、以下が導き出される。

$$\sum_{z' \in future\left(z, G^u_{[t,s]}\right)} vote_{x,y}\left(z', G^u_s\right) \leq$$
$$-\left|future_h\left(z, G^u_{[t,s]}\right)\right| + \left|future_a\left(z, G^u_{[t,s]}\right)\right| \leq$$
$$-\left|G^u_{[t,s]} \cap honest\right| + \left|G^u_{[t,s]} \cap malicious\right|,$$

$t \geq publication(x) + d \geq publication(z) + d$である事実を利用する。

4) 全体図

$$\sum_{z' \in future(z, G^u_s)} vote_{x,y}\left(z', G^u_s\right) \leq \tag{27}$$
$$l + k_1 + m - g - \left|G^u_{[t,s]} \cap honest\right| + \left|G^u_{[t,s]} \cap malicious\right| \leq \tag{28}$$
$$l + k_1 + m - g - \left|future_h\left(z_{late}, G^{oracle}_{[t,\max\{s-d,t\}]}\right)\right| +$$
$$\left|future_a\left(z_{late}, G^{oracle}_{[t,s]}\right)\right| \tag{29}$$

そのため、sとuで$vote_{x,y}\left(virtual\left(G^u_s\right)\right) \geq 0$が(29)が負の数ではない事象に含まれる。補題15の証明の場合と同じように、後者の事象の確率は$\left(\frac{\alpha}{1-\alpha}\right)^{(l+k_1+h'+m-g)^+}$が上限となる。$h'$は$\left|future_a\left(x, G^{oracle}_{[s',s]}\right)\right|$に等しい。その後、$k_1$と$h'$をパラメータ$\alpha \cdot d \cdot \lambda + (1-\alpha) \cdot d \cdot \lambda = d \cdot \lambda$のポアソン変数に組み込むことで以下を算出する。

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) \leq \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d \cdot \lambda, h) \cdot \tag{30}$$
$$\sum_{m=0}^{\infty} \binom{n_x + m - 1}{m} \cdot (1-\alpha)^{n_x} \cdot \alpha^m \cdot \tag{31}$$
$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-h-l-m)^+}.$$

結論22

補題21の仮定に加えて、$publication(y) \geq t$であることが判明している場合、

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) \leq \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d\cdot\alpha\cdot\lambda,h)\cdot \tag{32}$$

$$\sum_{m=0}^{\infty}\binom{n_x+m-1}{m}\cdot(1-\alpha)^{n_x}\cdot\alpha^{m}\cdot \tag{33}$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-h-l-m)^{+}}.$$

証明

*y*が時間*t*まで公開されていないため、$future\left(x,G^{oracle}_{[t-d,t]}\right)$ 内の全ての誠実なブロックは*x*に投票することが判明している。そのため、(26)内の $k_1=\left|G^{oracle}_{[t-d,t]}\cap honest\right|$ の減少は表面的なものであり、そのため補題15からはh'とパラメータ$d\cdot\alpha\cdot\lambda$　のポアソン変数だけを減じるだけでよい。

結論23

$X\subseteq G^v_t\cap honest$ 及び$Y\subseteq G^{oracle}_t\setminus G^v_t$とする。さらに、*X*内の要素はトポロジー的にお互いに関係しないと仮定する（例：$\forall x_1,x_2\in X, x_1\in anticone\left(x_2,G^v_t\right)$）。$z_{late}$ は*X*内の最新のブロック、z_{early} は*X*内の最初のブロックである。
その場合、

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) \leq \sum_{h=0}^{\infty} \mathcal{P}_{oiss}((3-2\cdot\alpha)\cdot d\cdot\lambda,h)\cdot$$

$$\sum_{m=0}^{\infty}\binom{n_x+m-1}{m}\cdot(1-\alpha)^{n_x}\cdot\alpha^{m}\cdot\left(\frac{\alpha}{1-\alpha}\right)^{(n_x-h-l-m)^{+}}$$

証明

補題21の結果を調整する。主な修正点は*g*が全てのペア(x,y) に関連する必要があるという点である。
$g:=\max_{x_1,x_2\in X}\left\{z\in\overline{future}\left(x_1,G^v_t\right):vote_{y,x_2}\left(z,G^v_t\right)=-1\right\}-$
$\min_{x_1,x_2\in X}\left\{z\in\overline{future}\left(x_1,G^v_t\right):vote_{y,x_2}\left(z,G^v_t\right)=+1\right\}$を定義する。間隔 $[time(z_{early})+2\cdot d,t]$ において全ての誠実なブロックは $\cap_{x\in X}\overline{future}\left(x,G^v_t\right)$ に属している 。特に、$h':=n_x-g$とする場合、*h′* がパラメータ $2\cdot d\cdot\lambda$のポアソン変数によって上限を設定される。その後、補題21の証明

において行った分析を適用する。*s*は$(x,y) \in X \times Y$で
$vote_{x,y}\left(virtual\left(G_s^u\right)\right) \geq 0$となる成立する最初のタイミングである。この補題の結果とh'上の確率分布と組み合わせることで事象$\cup_{(x,y)\in X\times Y}\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)$が最大でも以下となるという結論に達する。

$$\Pr\left(\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)^{\complement}\right) \leq \sum_{h'=0}^{\infty}\mathcal{P}_{oiss}(2\cdot d\cdot(1-\alpha)\cdot\lambda,h')\cdot\sum_{h=0}^{\infty}\mathcal{P}_{oiss}(d\cdot\lambda,h)\cdot$$
$$\sum_{m=0}^{\infty}\binom{n_x+m-1}{m}\cdot(1-\alpha)^{n_x}\cdot\alpha^m\cdot\left(\frac{\alpha}{1-\alpha}\right)^{(n_x-h'-h-l-m)^+} =$$
$$\sum_{h=0}^{\infty}\mathcal{P}_{oiss}((3-2\cdot\alpha)\cdot d\cdot\lambda,h)\cdot$$
$$\sum_{m=0}^{\infty}\binom{n_x+m-1}{m}\cdot(1-\alpha)^{n_x}\cdot\alpha^m\cdot\left(\frac{\alpha}{1-\alpha}\right)^{(n_x-h-l-m)^+}.$$

結論17

結論16の仮定への追加において、$M \leq |future_a\left(x,G_t^v\right)|$と仮定する場合、以下が導き出される。

$$\Pr\left(\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)\mid|future_a\left(x,G_t^v\right)|\geq M\right)\leq \tag{34}$$
$$\sum_{k=0}^{\infty}\mathcal{P}_{oiss}((2-\alpha)\cdot d\cdot\lambda,k)\cdot\sum_{h=0}^{\infty}\mathcal{P}_{oiss}(d\cdot(1-\alpha)\cdot\lambda,h)\cdot \tag{35}$$
$$\left(\sum_{m'=M}^{\infty}\binom{n_x+j+h+m'-1}{m'}\cdot(1-\alpha)^{n_x+j+h}\cdot\alpha^{m'}\right)^{-1}\cdot$$
$$\sum_{m=M}^{\infty}\binom{n_x+j+h+m-1}{m}\cdot(1-\alpha)^{n_x+j+h}\cdot\alpha^m\cdot$$
$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-k-j-l-(m-M))^+}$$

この不等号のRHSを$f_{post_mine}\left(n_x,g,j,l,M\right)$と表記する。補題25, 32そして30に基づき、$f_{post_mine}$を計算するために、これらの合計を切り捨てることで誤差の発生率が極端に低くなる。

証明

*M*の仮定に基づいて$\left|future_a\left(z_{late},G_t^{oracle}\right)\setminus future_a\left(x,G_t^v\right)\right|$

$= \left|future_a\left(z_{late}, G_t^{oracle}\right)\right| - \left|future_a\left(x, G_t^v\right)\right| \geq \left|future_a\left(z_{late}, G_t^{oracle}\right)\right| - M$
となる。
その後、結論16の結果と(4)を調整することで上記で更新された*m*の定義を考慮に入れる。そのため、べき指数ではmの代わりに$m - M$を使用し、$\left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-j-k-l-(m-M))^+}$とする。次に、*M*以上の負の二項分布（結論16の証明において明記）の条件付けを行うことで$m - M$上の更新された確率分布を算出することができる。$future_a\left(x, G_t^v\right)$の*M*ブロックは$z_{late}$（そして$t$の前）と$future_a\left(x, G_t^v\right) \subseteq future_a\left(z_{late}, G_t^{oracle}\right)$の後に作成された。
その結果、$m - M$上の確率分布は以下のように計算される。

$$\Pr\left(m - M\right) = \left(\sum_{m'=M}^{\infty} \binom{n_x + j + h + m' - 1}{m'} \cdot (1-\alpha)^{n_x+j+h} \cdot \alpha^{m'}\right)^{-1} \cdot \binom{n_x + j + h + m - 1}{m} \cdot (1-\alpha)^{n_x+j+h} \cdot \alpha^{m},$$

そして希望する条件にたどり着く。結論16の証明における残りの引数は影響を受けない。

1) f_{pre_mine} *を計算するための数学的手法:*

・$\delta := \alpha \cdot \lambda \cdot d$*.とする。*$N \gg 1$ [15] *を選択して、マトリックス*$T \in \mathbb{R}_{N\times N}$ *を以下のように定義する。全ての*$1 \leq l < N-1,\ T_{l-1,l} = 1-\alpha,\ T_{l+1,l} = \alpha$ *及び*$l = N-1\text{:}\ T_{l-1,l} = 1-\alpha,\ T_{l,l} = \alpha$

[15]補題25では最大誤差$\widehat{\epsilon}$を達成しており、*N*を選択することで$\left(\frac{\alpha}{1-\alpha}\right)^{N-1} < \widehat{\epsilon}/2$及び$e^{-d\cdot\alpha\cdot\lambda} \cdot \frac{(d\cdot\alpha\cdot\lambda)^N}{N!} < \widehat{\epsilon}/2$となるために十分である。特に、*N*は$\widehat{\epsilon}$内の対数である。

マトリックスの最初の列は$1 < l < N-1\text{:}\ T_{l,0} = e^{-\delta} \cdot \frac{\delta^l}{l!}$と$l = N-1\text{:}\ T_{l,0} = 1 - \sum_{l=0}^{N-2} e^{-\delta} \cdot \frac{\delta^l}{l!}$において次のように定義される：

$T_{0,0} := (1-\alpha) \cdot e^{-\delta},\ T_{1,0} = e^{-\delta} \cdot \alpha + e^{-\delta} \cdot \delta$

・固有値1に対応する*T*の固有ベクトルを検索してπと表記する。
$\Pi(l) := \sum_{l'=0}^{l} \pi(l')$と定義し、最後に$f_{pre_mine}(l) := 1 - \Pi((l-1)^+)$と定義する。

マトリックス*T*は負ではない整数上の特別反射ランダムウォーク(X_k)の推移確率マト

リックスである：$T_{i,j} := \Pr\left(X_{k+1} = i \mid X_k = j\right)$
全ての位置（エッジ0と$N-1$以外）においてランダムウォークは$\text{w.p.}(1-\alpha)$ の場合は負の無限大、$\text{w.p.}\alpha$の場合は正の無限大に近づいていく。起点に到達すると、（修正済みの）ポアソン分布に従って$\{0,1,...,N-1\}$ 内の次の位置にジャンプする。このランダムウォークによってエルゴード的マルコフ連鎖が発生するため、π と表記する独自の固定分布を持つ。Π はπ の累積確率関数である。
補題24
全ての$\hat{r} \geq r$ と全ての$l \in \mathbb{N}$ ：

$$\Pr\left(\max_{z \in G_{\hat{r}}^{oracle} \cap honest} \left\{\left|future_a\left(z, G_r^{oracle}\right)\right| - \left|future_h\left(z, G_r^{oracle}\right)\right|\right\} > l\right) \leq f_{pre\_mine}(l). \quad (36)$$

証明
パート *I:* We全ての$z \in G_r^{oracle} \cap honest$ 上で最大値を取っていると仮定して結果を証明した。その後、全ての$z \in G_{\hat{r}}^{oracle} \cap honest$ 上で最大値を取っても結果は変わらない。これは変数$\left\{\left|future_a\left(z, G_r^{oracle}\right)\right| - \left|future_h\left(z, G_r^{oracle}\right)\right|\right\}$ が負の数ではなく（以下で示す）、$z \in G_{\hat{r}}^{oracle} \setminus G_r^{oracle}$ の値が0であるためである。変数$\max_{z \in G_s^{oracle} \cap honest} \left\{\left|future_a\left(z, G_s^{oracle}\right)\right| - \left|future_h\left(z, G_s^{oracle}\right)\right|\right\}$ が反射ランダムウォークとしてモデリングできることを示している（誠実なネットワークの内部遅延*d*のためにウォークが起点に到達した時点で特殊な動作が発生する）。)。
新しい誠実なブロック*b*が作成されるたびに、過去の全ての*z*において$future_h\left(z, G_{time(b)}^{oracle}\right)$ が1増加する。*b*自体のこの変数の値は0である。そのため、$\max_{z \in G_s^{oracle} \cap honest} \left\{future_a\left(z, G_s^{oracle}\right) - future_h\left(z, G_s^{oracle}\right)\right\}$ の値の下限は0である。
一方で、新しい攻撃ブロックが作成される場合には、その時点で利用できる全ての誠実なブロックの$future_a\left(z, G_s^{oracle}\right)$ の値が1増加する（補題14、パートIで指定する最悪の事態に従う）。そのため、誠実なネットワーク上での攻撃者の最大の進捗は反射ランダムウォークとしてモデリングすることが可能である。誠実なブロック*b*の作成によって$before\left(time(b)\right)$ の適切なサブセットである可能性がある（$d > 0$の場合）$past_h(b)$ 内のブロックでのみ$future_h\left(z, G_s^{oracle}\right)$ が増加するため、誠実な

ブロックが攻撃ブロックに対して反抗して
$\max\left\{\left|future_a\left(z, G_s^{oracle}\right)\right| - \left|future_h\left(z, G_s^{oracle}\right)\right|\right\}$ の値を低下させない状況が発生する可能性がある。起点に訪問するたびにウォークの動作をゆがめることでこれを考慮に入れる（そしてその他全ての状態で誠実なネットワークの内部遅延による影響がないことを証明する）。

以下の分析では、我々は最悪のシナリオを想定している。つまり、 z_1 及び z_2 が誠実なブロックである場合、 $|time(z_1) - time(z_2)| < d$ であり、
$z_1 \in anticone\left(z_2\right)$ である。これは最悪のケースであり誠実なブロック間の一部のエッジを省略することで $future_h\left(z, G_s^{oracle}\right)$ の値が低下し、
$\left|future_a\left(z, G_s^{oracle}\right)\right| - \left|future_h\left(z, G_s^{oracle}\right)\right|$ の値が上昇する。

攻撃者が秘密にブロックを作成する場合、作成する全ての新しいブロックに関して新しいブロックがどのブロックをポイントすべきかに関して決定する必要がある。次の戦略を考える: *time*(*b*)の時点で作成された攻撃者の新しいブロック *b*が $G_{time(b)}^{oracle}$ をポイントする（もちろん自身は除外）。補題14、パートIにおいてこれが最悪のケースを想定したものであることは既に説明しているが、この戦略によって
$\max_{z \in G_s^{oracle} \cap honest}\left\{\left|future_a\left(z, G_s^{oracle}\right)\right| - \left|future_h\left(z, G_s^{oracle}\right)\right|\right\}$ を最大化できることは簡単に確認できる。

パート II

t_iはG_r^{oracle}内の i番目のブロックの作成時間である。*zs*は変数
$\arg\max_{z \in G_s^{oracle} \cap honest}\left\{\left|future_a\left(z, G_s^{oracle}\right)\right| - \left|future_h\left(z, G_s^{oracle}\right)\right|\right\}$
である。

さらに、$A_s^z := \left|future_a\left(z, G_s^{oracle}\right)\right|$ 及び $H_s^z := \left|future_h\left(z, G_s^{oracle}\right)\right|$ と定義する。

省略記号 $A_s := \left|future_a\left(z_s, G_s^{oracle}\right)\right|$ 及び
$H_s := \left|future_h\left(z_s, G_s^{oracle}\right)\right|$

サブシリーズ $(s_k) \subseteq (t_i)$ を再帰的に定義する。 $s_0 = 0$ 及び全ての
$k > 0$: $s_{k+1} = \min_i\{t_i : \ t_i \geq time\left(z_{s_k}\right) + d\}$
$(A_{s_k} - H_{s_k})$ は*Xk*と同じ確率分布であると主張する。

この主張があてはまると仮定すると、 s_k は最初の s_k で $s_k \geq r$ である。その場合、 $(A_r - H_r) \leq \ 1 + (A_{s_k} - H_{s_k})$.[16] である。

結果

$$\Pr\left(\max_{z\in G_r^u\cap honest}\{|future_a(z,G_r^u)|-|future_h(z,G_r^u)|\}>l\right)=\Pr(A_r-H_r>l)\leq\Pr(A_{s_k}-H_{s_k}>l-1)=\Pr(X_k>l-1)=1-\Pi((l-1)^+).$$

パートIII:

証明を完成させるために、*k*の帰納法によって主張を証明する。$k=0$の場合、$s_0=0$である。 0の時点で、*genesis*ブロックの作成後に、(A_0-H_0)の値は0である。 $future(genesis)\cap G_0^{oracle}=\emptyset$であり、同様に$X_0=0$である。*k*でこれを証明したものと仮定し、次に*k*+1でもこれを証明する。まず、$(A_{s_k}-H_{s_k})>0$と仮定する。否定によって$s_k<time(z_{s_k})+d$と仮定する。その後、$(s_k),\ s_k=time(z_{s_k})$を構築する。これは、誠実なネットワークがの$s_k$時間内に$z_{s_k}$を作成したことを示唆する。そのため、$z_{s_k}$が$s_k$時点で作成されているために$(A_{s_k}-H_{s_k})=0$である。$z_{s_k}$は$\arg\max_{z\in G_r^{oracle}\cap honest}\{A_{s_k}-H_{s_k}\}$内にあると推定されるため、これは我々の$A_{s_k}-H_{s_k}>0$という仮定と矛盾する。そのため、$(A_{s_k}^z-H_{s_k}^z)>0$は$s_k\geq time(z_{s_k})+d$を示唆する。

結果として、$(A_{s_k}-H_{s_k})>0$の場合、誠実なネットワーク全体がブロックz_{s_k}について把握していることを保証される。そのため、誠実なネットワークは$(1-\alpha\cdot\lambda)$の速度で$future(z_{s_k})$にブロックを追加する一方で攻撃者はαの速度でブロックを追加する。その後、誠実なネットワークの全てのブロックは$|future_a(z,G_s^{oracle})|$を1増加させる。

そのため、攻撃者のブロックw.p.αを追加することで$(A_{s_k}-H_{s_k})$が1増加し、w.p.$(1-\alpha)$によって1減少する。 $X_k>0$であることを条件に、X_{k+1}の分布が同じ挙動を示す：

$$\Pr(X_{k+1}=X_k+1\mid X_k>0)=1-\Pr(X_{k+1}=X_k-1\mid X_k>0)=\alpha$$

16 $r=s_k$である場合、これは自明である。それ以外の場合、その推移マトリックスに従って(*Xi*)は全てのステップで多くとも1減少するため間隔(r,s_k)内で誠実なネットワークは$future(z_{s_{k-1}})$を多くとも1ブロック増加させる。そのため、間隔$(r,s_k)\subseteq(s_{k-1},s_k)$内で誠実なネットワークは多くとも1個のブロックを作成する。

$(A_{s_k}-H_{s_k})=0$と仮定する。攻撃者が誠実なネットワーク（s_k時点まで）によって作成された最後のブロックに対して少なくとも1以上の優位性を持つことになるため

s_k の時間内に作成されたブロックが攻撃者に帰属することはありえない。そのため、このブロックは誠実なネットワークに帰属する。z_{s_k}の定義によって、s_kの時間内に作成されたブロックとなる。結果として、$(s_k, s_k + d)$の間隔内で誠実なネットワークは $future(z_{s_k})$ にブロックを追加していないことになる（ここでは最悪のケースを想定している。例：誠実なブロック1個あたり伝播時間d秒）。この間隔中に攻撃者はパラメータ$\alpha \cdot \lambda$のポアソン過程に従ってブロックを作成している。そのため、$\left(A^{z_{s_k}}_{s_k+d} - H^{z_{s_k}}_{s_k+d}\right) = i$ w.p. $\mathcal{P}_{oiss}(\alpha \cdot \lambda \cdot d, i)$である。この場合、$s_k + d$ 後に作成されるシステム内の次のブロックは攻撃者の w.p.α である。この場合、合計ギャップが$j+1$だけ増加する。つまり $\left(A^{z_{s_{k+1}}}_{s_{k+1}} - H^{z_{s_{k+1}}}_{s_{k+1}}\right) = \left(A^{z_{s_k}}_{s_k+d} - H^{z_{s_k}}_{s_k+d}\right) + 1$ これに代わって、s_k+d後に作成される次のブロックが誠実なネットワークの w.p.$(1-\alpha)$である場合、$\left(A^{z_{s_{k+1}}}_{s_{k+1}} - H^{z_{s_{k+1}}}_{s_{k+1}}\right) = \max\left\{\left(A^{z_{s_k}}_{s_k+d} - H^{z_{s_k}}_{s_k+d}\right) - 1, 0\right\}$ となる。これを $\Pr\left(X_{k+1} \mid X_k = 0\right)$ と比較することでこの場合にも変数 X_{k+1} の挙動が $\left(A^{z_{s_{k+1}}}_{s_{k+1}} - H^{z_{s_{k+1}}}_{s_{k+1}}\right)$ と同じであることを確認できる。

補題25

正の定数B_l, C_lで$f_{pre_mine}(l) \leq C_l \cdot e^{-B_l \cdot future_a(x, G^v_t)}$である。

この結果を理解する場合、$d = 0$の時に反射ランダムウォークの定常分布は $\left(\frac{\alpha}{1-\alpha}\right)^l$ に比例していることに着目する必要がある。そして、$d > 0$の場合、この関係は $l \gg d \cdot \lambda$でもあてはまる。

証明

$n > 1$の場合、定常分布π は$\pi(n) = (1-\alpha) \cdot \pi(n+1) + \alpha \cdot \pi(n-1) + e^{-\delta} \cdot \frac{\delta^n}{n!} \cdot \pi(0)$の関係を満たしている。

$n \geq 0$で$\pi(n) = C_n \cdot \left(\frac{\alpha}{1-\alpha}\right)^n$ とする。以下が成立する。

$$C_n \cdot \left(\frac{\alpha}{1-\alpha}\right)^n = (1-\alpha) \cdot C_{n+1} \cdot \left(\frac{\alpha}{1-\alpha}\right)^{n+1} + \tag{37}$$

$$\alpha \cdot C_{n-1} \cdot \left(\frac{\alpha}{1-\alpha}\right)^{n-1} + e^{-\delta} \cdot \frac{\delta^n}{n!} \cdot \pi(0) \Longrightarrow \tag{38}$$

$$C_n = C_{n+1} \cdot \alpha + C_{n-1} \cdot (1-\alpha)^{-1} + e^{-\delta} \cdot \frac{\left(\delta \cdot \frac{(1-\alpha)}{\alpha}\right)^n}{n!} \cdot \pi(0). \tag{39}$$

*n*の値が十分大きい場合、上記の数式内の最後の被加数は無視できる。そのため、$\forall n : C_n \approx C$ とすると、値の大きな*n*で上記の関係が成り立つ（最後の被加数の許容可能な誤差まで）。そのため、定数 $C, \pi(n) \leq C \cdot \left(\frac{\alpha}{1-\alpha}\right)^n$ では $1 - \Pi((n-1)^+) = \sum_{k=n}^{\infty} \pi(k) \leq B_l \cdot e^{-C_l \cdot n}$ であり、値の大きい n と一部の定数 $B_l, C_l > 0.$ となる。

以下の結論は補題24から抜粋したものである。

結論26

補題21において、l が判明していない場合、以下が成り立つ。

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) \leq \sum_{l=0}^{\infty} \pi(l) \cdot \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d \cdot \lambda, h) \cdot \tag{40}$$

$$\sum_{m=0}^{\infty} \binom{n_x + m - 1}{m} \cdot (1-\alpha)^{n_x} \cdot \alpha^m. \tag{41}$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-h-l-m)^+}.$$

同様に、結論22では以下が成り立つ。

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) \leq \sum_{l=0}^{\infty} \pi(l) \cdot \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d \cdot \alpha\lambda, h) \cdot \tag{42}$$

$$\sum_{m=0}^{\infty} \binom{n_x + m - 1}{m} \cdot (1-\alpha)^{n_x} \cdot \alpha^m. \tag{43}$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-h-l-m)^+}. \tag{44}$$

最後に結論23では以下が成り立つ。

$$\Pr\left(\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) \leq \sum_{l=0}^{\infty} \pi(l) \cdot \sum_{h=0}^{\infty} \mathcal{P}_{oiss}((3-2\cdot\alpha)\cdot d \cdot \lambda, h) \cdot$$

$$\sum_{m=0}^{\infty} \binom{n_x + m - 1}{m} \cdot (1-\alpha)^{n_x} \cdot \alpha^m \cdot \left(\frac{\alpha}{1-\alpha}\right)^{(n_x-h-l-m)^+}.$$

この結論を用いて、アルゴリズム7で説明するオンラインポリシーが使用する限界を証明できる。

以下のように表記する。

$$risk\_hidden(T,g) := \sum_{l=0}^{\infty} \pi(l) \cdot \sum_{m=0}^{\infty} \mathcal{P}_{oiss}((T+2\cdot d)\cdot\alpha\cdot\lambda)\cdot \tag{45}$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{(g-l-m)^+} \tag{46}$$

結論27

アルゴリズム7がϵ以下の値を返す場合、$\Pr\left(\cup_{y\in G_\infty^{pub}\setminus G_t^{pub}} \widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) < \epsilon$

となる。

証明

最初に、結論22で使用する変数gは$\min_{x'\in\overline{anticone_h}(x,G_t^v)} |future(x',G_t^v)|$で置き換えられる。これは、$y$に対する仮定によって$future(x,G_t^v)$内の全てのブロックは$x$に投票するためである。アルゴリズム7の5行目で$g$に割り当てられる値は$\min_{x'\in\overline{anticone}(x,G_x)} |future(x',G_x)|$によって上限を設定される。これは$G_x$に$G_t^v$内の全ての誠実なブロックが含まれるためである。次に、3行目でTに$time_now - received^v(x) = t - received^v(x)$の値を割り当てる。$m$は$\mathcal{P}_{oiss}(m,(t-time(x))\cdot\alpha\cdot\lambda)$に従って表記される[17]。

$time(x) \geq received^v(x)+d$であるため、パラメータ$(T+d)\cdot\alpha\cdot\lambda$のポアソン変数によって上限を設定できる。その後、結論26の第2項の結果を調整する。hとmの分布（hは(42)から抽出）を組み合わせることによって以下の結論を導き出す。

$$\Pr\left(\cup_{y\in G_\infty^{pub}\setminus G_t^{pub}} \widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)^{\complement}\right) \leq \tag{47}$$

$$\sum_{l=0}^{\infty}\pi(l)\cdot\sum_{m=0}^{\infty}\mathcal{P}_{oiss}((T+2\cdot d)\cdot\alpha\cdot\lambda)\cdot\left(\frac{\alpha}{1-\alpha}\right)^{(g-l-m)^+} = \tag{48}$$

$$risk\_hidden(T,g). \tag{49}$$

我々の分析では最悪のケースでは全ての攻撃者ブロックがこのセットの全てのyで$y \prec x$に投票すると仮定しているために$G_\infty^{pub}\setminus G_t^{pub}$内の異なる$y$で合併上界を適用する必要はない。さらに、$G_t^v$内の全ての誠実なブロックはこのセットの全ての$y$で常に$x \prec y$に投票する（過去に$y$が存在しないため）。

そのため、最悪のケースを想定した分析では$G_\infty^{pub}\setminus G_t^{pub}$内の$y$で攻撃者が関係$x \prec y$を逆転させることに成功している事象は特定の$y$でこれに成功している事象に相当する。結論的に、アルゴリズム7でϵ以下の値を返す場合、

$\Pr\left(\cup_{y\in G_{\infty}^{pub}\setminus G_{t}^{pub}}\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)^{\complement}\right)<\epsilon$ であることが判明する。

$dist\_gap(b,G)$ は $gap\left(b,\langle G,b,K\rangle\right)=0$ である最小の k を意味する。

補題28

*b*を誠実なブロックとする。その場合、

$$\Pr\left(\cup_{u\in honest,s\in[time(b),\infty)}dist\_gap\left(b,G_{s}^{u}\right)>K\right)\leq \tag{50}$$

$$\sum_{l=0}^{\infty}\pi(l)\cdot\sum_{h=0}^{\infty}\mathcal{P}_{oiss}(d\cdot\lambda,h)\cdot$$

$$\sum_{m=0}^{\infty}\binom{n_x+m-1}{m}\cdot(1-\alpha)^{n_x}\cdot\alpha^{m}\cdot \tag{51}$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{(K-h-l-m)^{+}}.$$

$f_{distgap}(K)$ は(50)のRHSを意味する。

証明

その定義により、$dist\_gap\left(b,G_{s}^{u}\right)>K$ である事象は $anticone\left(b,G_{time(b)}^{node(b)}\right)\cup\left(G_{s}^{u}\setminus G_{time(b)}^{node(b)}\right)$ 内の一部のブロックが $vote\left(virtual\left(\langle G_{s}^{u},b,K\rangle\right)\right)$ に従って b に対して優先される（もしくはタイを有効にする）事象に相当する。$\langle G_{s}^{u},b,K\rangle$ では b はその他全ての $y\notin\overline{past}\left(b\right)$ に対抗して*b*に好意的に投票する追加の*K*ブロック $b_1,\ldots,b_K$ がある。実際、$y\notin\overline{past}\left(b\right)$ の場合は $y\notin\overline{past}\left(b_i\right)$ である。結果として、$G_{time(b)}^{node(b)}$ において $future\left(b\right)$ 内の*K*ブロックはそのanticone内のあらゆるブロックに対応して自身に有利な投票を行う。そのため、$n_x=K,\ X=\{b\}$ 及び $Y=G_{\infty}^{oracle}\setminus\overline{past}\left(b\right)$ である結論26の最初のパートを適用することで(50)は $anticone\left(b,G_{time(b)}^{node(b)}\right)$ 内のブロックが未来の $s\geq time(b)$ の $\langle G_{s}^{u},b,k\rangle$ のペア順序において b に対して優先される（もしくはタイとなる）確率の上限であると結論付けることができる。

[17]我々の以前の分析では n_x を用いて m を測定している。構造ベースであり Π にアクセスできないためである。

補題29

全ての $n_x\in\mathbb{N}$ に関して

$$\Pr\left(\left|future_h\left(x, G_t^v\right)\right| > n_x\right) \leq \tag{52}$$

$$\left|future\left(x, G_t^v\right)\right| \cdot f_{distgap}\left(\sqrt{\left|future\left(x, G_t^v\right)\right|}\right). \tag{53}$$

最後の不等号のRHSは $f_{post\_pub}\left(\left|future\left(x, G_t^v\right)\right|\right)$ と表記される。

証明

$y = NULL$ の場合、$n_x = future\left(x, G_t^v\right) \geq future_h\left(x, G_t^v\right)$ となるため証明の必要がない。$y \neq NULL$ と仮定する。

$K := \sqrt{\left|future\left(x, G_t^v\right)\right|}$ とする。アルゴリズム3では $dist\_gap > K$ のブロック数である*M*を $future\left(x, G_t^v\right)$ から引くことによって n_x を算出する。*b*は $future\left(x, G\right)$ 内の誠実なブロックである。補題28では $dist\_gap\left(b, G_t^v\right)$ が K より大きい確率は多くとも $f_{distgap}\left(K\right)$ である。合併上界により、$future_h\left(x, G_t^v\right)$ 内の*b*で $dist\_gap\left(b, G_t^v\right) > K$ は多くとも $\left|future_h\left(x, G_t^v\right)\right| \cdot f_{distgap}\left(K\right) \leq \left|future\left(x, G_t^v\right)\right| \cdot f_{distgap}\left(K\right) = f_{post\_pub}\left(\left|future\left(x, G_t^v\right)\right|\right)$ である。

不等号（50）のRHSは以下の内容を示唆する。

補題30

正の定数 B_c, C_c で $f_{post\_pub}\left(\left|future\left(x, G_t^v\right)\right|\right) \leq C_c \cdot e^{-B_c \cdot future_a\left(x, G_t^v\right)}$ である。

補題31

$n_j \in \mathbb{N}$ で $j := gap\left(x, G\right) + n_j$ とする。

$$\Pr\left(\left\{\left|anticone_h\left(x, G_t^{oracle}\right)\right| > j\right\}\right) \leq \tag{54}$$

$$f_{pre\_mine}\left(\sqrt{n_j}\right) + \sum_{h'=0}^{\infty} \mathcal{P}_{oiss}((1-\alpha) \cdot \lambda \cdot d, h') \cdot \tag{55}$$

$$f_{post\_mine}\left(n_j, n_j - h' + 1, \sqrt{n_j}\right) \tag{56}$$

この不等号のRHSを $f_{pre\_pub}(n_j)$ とする。この結果を理解するために、ブロックはそのブロックの公開時点付近もしくは公開後に公開されたブロックに対してのみ優先される、という事実を思い出す必要がある。

証明

パート*I:* $t_x := publication(x)$ とする。

$L_n := \{z \in anticone_h\left(x, G_t^v\right) : future_h\left(z, anticone_h\left(x, G_{t_x}^v\right)\right) \geq n\}$ と定義する。（この定義では t_x を

使用している。）A_n によって $\{\exists z \in L_n : z \in X_{win}\left(x, G_t^v\right)\}$ の事象を明記する。最後に、z_e を $L_{n'}^{\complement} \cap anticone_h\left(x, G_t^{oracle}\right)$ 内の最初のブロックとし、$n_j := \sqrt{|future\left(x, G_t^v\right)|}$ で $n' := n_j - \left|\overline{anticone_h}\left(z_e, G_t^{oracle}\right)\right| + 1$ とする。

$X_{win}\left(x, G\right)$ によって*G*の仮想投票のペア順序内で*x*が優先される（もしくはタイとなる）ブロックのセットを、$X_{lose}\left(x, G\right)$ によって残りのブロックを表記する。
その場合、以下が成り立つ。

$$\left\{\left|anticone_h\left(x, G_t^{oracle}\right)\right| > gap\left(x, G\right) + n_j\right\} =$$
$$\left\{\left|X_{win}\left(x, G_t^{pub}\right) \cap anticone_h\left(x, G_t^{oracle}\right)\right| +$$
$$\left|X_{lose}\left(x, G_t^{pub}\right) \cap anticone_h\left(x, G_t^{oracle}\right)\right| > gap\left(x, G\right) + n_j\right\} =$$
$$\left\{\left|X_{win}\left(x, G_t^{pub}\right) \cap anticone_h\left(x, G_t^{oracle}\right)\right| + gap\left(x, G_t^v\right) > gap\left(x, G\right) + n_j\right\} =$$
$$\left\{\left|X_{win}\left(x, G_t^{pub}\right) \cap anticone_h\left(x, G_t^{oracle}\right)\right| > n_j\right\} =$$
$$\left(\left\{\left|X_{win}\left(x, G_t^{pub}\right) \cap anticone_h\left(x, G_t^{oracle}\right)\right| > n_j\right\} \cap A_{n'}\right) \cup$$
$$\left(\left\{\left|X_{win}\left(x, G_t^{pub}\right) \cap anticone_h\left(x, G_t^{oracle}\right)\right| > n_j\right\} \cap A_{n'}^{\complement}\right) \subseteq$$
$$A_{n'} \cup \left\{\left|L_{n'}^{\complement} \cap anticone_h\left(x, G_t^{oracle}\right)\right| > n_j\right\} =$$
$$A_{n'} \cup \left\{\left|L_{n'}^{\complement} \cap anticone_h\left(x, G_t^{oracle}\right) \cap \overline{anticone_h}\left(z_e, G_t^{oracle}\right)\right| +\right.$$
$$\left.\left|L_{n'}^{\complement} \cap anticone_h\left(x, G_t^{oracle}\right) \cap future_h\left(z_e, G_t^{oracle}\right)\right| > n_j\right\} =$$
$$A_{n'} \cup \left\{\left|L_{n'}^{\complement} \cap anticone_h\left(x, G_t^{oracle}\right) \cap future_h\left(z_e, G_t^{oracle}\right)\right| >\right.$$
$$\left. n_j - \left|L_{n'}^{\complement} \cap anticone_h\left(x, G_t^{oracle}\right) \cap \overline{anticone_h}\left(z_e, G_t^{oracle}\right)\right|\right\} \subseteq$$
$$A_{n'} \cup \left\{\left|anticone_h\left(x, G_t^{oracle}\right) \cap future_h\left(z_e, G_t^{oracle}\right)\right| >\right.$$
$$\left. n_j - \left|\overline{anticone_h}\left(z_e, G_t^{oracle}\right)\right|\right\}.$$

$z_e \in L_{n'}$ であり、またn'の定義によって$anticone_h\left(x, G_t^{oracle}\right) \cap future_h\left(z_e, G_t^{oracle}\right)$ に $n_j - \left|\overline{anticone_h}\left(z_e, G_t^{oracle}\right)\right|$ 以上のブロックが含まれることはない。そのため、
$\left\{\left|anticone_h\left(x, G_t^{oracle}\right) \cap future_h\left(z_e, G_t^{oracle}\right)\right| > n_j - \left|\overline{anticone_h}\left(z_e, G_t^{oracle}\right)\right|\right\}$
の事象が発生し、
$\Pr\left(\left\{\left|anticone_h\left(x, G_t^{oracle}\right)\right| > gap\left(x, G\right) + n_j\right\}\right) \leq \Pr\left(A_{n'}\right)$ が算出され

る。
あらゆる$z \in L_{n'}$において$future\left(z, anticone_h\left(x, G_{t_x}^v\right)\right)$内の全てのブロックは$x$に対抗して$z$に有利に投票する。そして定義によって$t_x$の時点でそのような投票者が少なくとも$n'$人いる。結果として、次のパラメータに関して結論23の結果を適用できる: $v = pub$, $t = t_x$、$X = L_{n'}$のリーフブロック、$Y = \{x\}$, $g := n'$, $n_x := n_j$そして
$l' = \max_{z \in G_{t_x}^{oracle} \cap honest} \{|future_a\left(z, G_{t_x}^{oracle}\right)| - |future_h\left(z, G_{t_x}^{oracle}\right)|\}$となり、以下が算出される。

$$\Pr\left(A_{n'}\right) = \Pr\left(\exists z \in L_{n'} : z \in X_{win}\left(x, G_t^v\right)\right) \leq \Pr\left(\exists s > t_x, \exists z \in L_{n'} : z \in X_{win}\left(x, G_s^v\right)\right) \leq f_{post\_mine}\left(n_j, n', l'\right).$$

我々はl'の値を把握していないため、補題24を用いて確率$\geq 1 - f_{pre_mine}(l)$では、値が多くともlであることを証明する。$l = \sqrt{n_j}$に固定する。同様に、n'の値も判明していない。
ただし、$\overline{anticone_h}\left(z_e, G_t^{oracle}\right)$内のブロックは間隔$[time(z_e), time(z_e) + d]$内に作成されている（選択による）。そのため、$\left|\overline{anticone_h}\left(z_e, G_t^{oracle}\right)\right|$はパラメータ$(1-\alpha) \cdot \lambda \cdot d$のポアソン変数である。
そのため、以下のように結論付ける。

$$\Pr\left(\left\{\left|anticone_h\left(x, G_t^{oracle}\right)\right| > gap\left(x, G\right) + n_j\right\}\right) \leq f_{pre\_mine}\left(\sqrt{n_j}\right) + \sum_{h'=0}^{\infty} \mathcal{P}_{oiss}((1-\alpha) \cdot \lambda \cdot d, h') \cdot f_{post\_mine}\left(n_j, n_j - h' + 1, \sqrt{n_j}\right) = f_{pre\_pub}(n_j).$$

$f_{post_mine}\left(n_j, n_j - h' + 1, \sqrt{n_j}\right)$が急速に減少することは簡単に確認できる（後の補題でもこの確認を行う）。従って:
補題32
正の定数f B_j, C_j *において*$f_{pre\_pub}(n_j) \leq C_j \cdot e^{-B_j \cdot n_j}$ *である。*
以下の補題では$oracle^u$は（理論上の）ノードであり、$G_s^{oracle_j} := G_s^u \cup \left(G_s^{oracle} \cap malicious\right)$となる。

補題18

事象 $\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)$ *を条件に、* τ *後に固定されたままとなる* m^* *（そして時間* t *までの事象によって決定される* $\mathbb{E}[m^*]$ *）において*

$$\forall s \geq \tau\colon\ M(oracle^u, s) \geq \left|future_a\left(x, G^{oracle^u}_s\right) \cap G^{oracle}_{[t,s]} \setminus V_{x\prec y}(G^{oracle^u}_s)\right| - m^*$$

となる時間 $\tau \in [t,\infty)$ *が存在する。*

証明

パート *I:* $y \notin G^{oracle^u}_s$ の場合、$M(oracle^u, s) = 0$（6行目）、$V_{x\prec y}(G^{oracle^u}_s) = future\left(x, G^{oracle^u}_s\right)$であり、必要な不等号が得られる。

$y \in G^{oracle^u}_s$ と仮定する。 *G*はある（仮想ブロックの可能性もある）ブロックの過去セットに等しいブロックDAGとする。

$\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)$ を条件に、*t*の時点で決定される定数*Ct*において、

$\left|G_{[t,s]} \cap malicious\right| - \left|G^{oracle^u}_{[t,s]} \cap honest\right| < -C_t$ の場合、

$vote_{x,y}\left(virtual\left(G\right)\right) = -1$.[18] となる。

これは補題14の証明に従う: (17)のLHSを抽出して*g*を

$\sum_{z'\in future(x,G_t)} vote_{x,y}\left(z', G\right)$ で置き換え、残りの項の値

$2\cdot\left|anticone_h\left(z_{late}, G^{oracle^u}_{time(z_{late})}\right)\right| - \left|future_h\left(z, G^{oracle^u}_{time(z_{late})}\right)\right| + \left|future_a\left(z, G^{oracle^u}_{time(z_{late})}\right)\right| + \left|anticone_h\left(z_{late}, G^{oracle^u}_{[time(z_{late}),time(z_{late})+d]}\right)\right| + \left|G^{oracle}_{[t-d,t]} \cap honest\right| + \left|future_h\left(z_{late}, G^v_t\right) \setminus future_h\left(x, G^v_t\right)\right| + \left|future_a\left(z_{late}, G^{oracle^u}_t\right) \setminus future_a\left(x, G^v_t\right)\right|$

が時間 t までに決定されるため、C_t と表記する。

$z \in future\left(x, G^{oracle}_{t,s}\right)$ とする。$\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)$の条件付けによって $z \in malicious$ となる。 $\mathrm{DAG}\ G^z := past\left(z\right)$ を固定する。上記の引数は特に G^z であてはまる。 $\left|G^z_{[t,s]} \cap malicious\right| - \left|G^z_{[t,s]} \cap honest\right| < -C_t$ の場合、$vote_{x,y}\left(z\right) = vote_{x,y}\left(virtual\left(past\left(z\right)\right)\right) = -1$ （*z*は強い投票者であるため、その投票の文脈を指定する必要はない） となる。

[18]ここで $G^z \cap before(s) \setminus before(t)$ において $G^z_{[t,s]}$ を表記する。

結果として、$z \in G^{oracle}_{t,s} \setminus V_{x\prec y}\left(G^{oracle}_s\right)$ の場合、

$\left|G^z_{[t,time(z)]} \cap malicious\right| - \left|G^z_{[t,time(z)]} \cap honest\right| \geq -C_t$ となる[19] 。

以下の重要な論理包含にたどり着く:

$z \in future\left(x, G_s^{oracle^u}\right) \setminus V_{x \prec y}\left(G_s^{oracle^u}\right)$ の場合、

$$\left|anticone\left(z, G_s^{pub}\right)\right| \geq \left|anticone\left(z, G_{time(z)}^{pub}\right)\right| \geq \tag{57}$$

$$\left|anticone\left(z, G_{time(z)}^{pub}\right) \setminus G_t^{oracle}\right| = \tag{58}$$

$$\left|G_{time(z)}^{pub} \setminus G_t^{oracle}\right| - \left|\overline{past}\left(z\right) \setminus G_t^{oracle}\right| \geq \tag{59}$$

$$\left|G_{time(z)}^{pub} \cap honest \setminus G_t^{oracle}\right| - G_{[t,time(z)]}^{z} \cap honest \geq \tag{60}$$

$$\left|G_{[t,time(z)]}^{pub} \cap honest\right| - G_{[t,time(z)]}^{z} \cap malicious - C_t. \tag{61}$$

パート *II:* $z_1, z_2, \cdots$ を $future_a\left(x, G_s^{oracle^u} \setminus G_t^{oracle}\right) \setminus V_{x \prec y}\left(G_s^{oracle^u}\right)$ 内のブロック作成順序とする。

z_mを固定し、b_mを $anticone_h\left(z_m, future\left(x, G_s^{oracle^u}\right)\right)$ 内の最初のブロックとする。

確率 $\mathcal{P}_{oiss}(d \cdot (1-\alpha) \cdot \lambda, h')$ の場合、

$\left|anticone_h\left(b_m, G_\infty^{pub}\right)\right| = h'$ となる。

(57)とb_mの選択によって以下が導き出される。

$$\left|future_h\left(b_m, G_{time(z_m)}^{pub}\right)\right| =$$
$$\left|anticone_h\left(z_m, G_{time(z_m)}^{pub}\right) \setminus anticone_h\left(b_m, G_{time(z_m)}^{pub}\right)\right| \geq$$
$$\left|anticone_h\left(z_m, G_{time(z_m)}^{pub}\right)\right| - \left|anticone_h\left(b_m, G_{time(z_m)}^{pub}\right)\right| \geq$$
$$\left|G_{[t,time(z_m)]}^{pub} \cap honest\right| - \left|G_{[t,time(z_m)]}^{z_m} \cap malicious\right| - C_t - h' =$$
$$\left|G_{[t,time(z_m)]}^{pub} \cap honest\right| - m - C_t - h', \tag{62}$$

b_mの選択によって $past\left(b_m\right) \cap anticone_h\left(z_m\right) = \emptyset$ 及び $anticone_h\left(z_m, G_{time(z_m)}^{pub}\right) = antipast_h\left(z_m, G_{time(z_m)}^{pub}\right)$ という事実を活用する。

パート*III:* mとすると、$\left|G_{[t,time(z_m)]}^{pub} \cap honest\right|$ は負の二項分布 $\Pr\left(\left|G_{[t,time(z_m)]}^{pub} \cap honest\right| = n\right) = \binom{n+m-1}{n} \cdot (1-\alpha)^n \cdot \alpha^m$ に従って分布する。誠実なブロックb_mがの順序内でz_mによって優先される確率は多くとも

$$\sum_{l=0}^{\infty}\pi(l)\cdot\sum_{k=0}^{\infty}\mathcal{P}_{oiss}(5\cdot d\cdot(1-\alpha)\cdot\lambda,k)\cdot\sum_{h=0}^{\infty}\mathcal{P}_{oiss}(d\cdot(1-\alpha)\cdot\lambda,h)\cdot$$
$$\sum_{n=0}^{\infty}\binom{n+m-1}{m}\cdot(1-\alpha)^n\cdot\alpha^m\cdot\left(\frac{\alpha}{1-\alpha}\right)^{(n-m-K-2\cdot h-C_t-k-l)^+}. \quad (63)$$

であると主張する。

[19] G_zには$time(z)$までに作成されたブロックのみが含まれる。

これは補題14と結論23の証明において行った類似の分析に従うものである。実際、$time(z_m)$ の時点で$future\,(b_m)\setminus\overline{future}\,(z_m)$内に少なくとも$n-m-h'-C_t$個のブロックがあり、下限は上記の通り $\left|future_h\left(b_m,G^{pub}_{time(z_m)}\right)\right|$ である。$future_a\left(b_m,G^{pub}_{time(z_m)}\right)=\emptyset$, で $b_m\notin past\,(z_m)$ である一方で、$\langle G_s^{oracle^u},z_m,K\rangle$の構築によって$y\prec x$と投票する追加の$K$個の理論上のブロックが存在する。（結論23内で指定する限界同様に）指数内のh' を減じる代わりに、ポアソン変数の合計はポアソン変数であるために変数kに$2\cdot d\cdot(1-\alpha)\cdot\lambda$を追加する。最終的に補題24の結果を用いて$\pi(l)$ がl上の分布の上限となっていることを確認する[20]。

$dist\_gap(G_s^{oracle^u},z_m)\leq K$ では $virtual\left(\langle G_s^{oracle^u},z_m,K\rangle\right)$の順序内で b_m に対して優先されるために z_mを必要とするため、(63)は $dist\_gap(G_s^{oracle^u},z_m)\leq K$ という確率に対する上限の役割も果たす。

パート*IV:* 補題25を用いることで定数a,b 及びW の存在を確認し、$\Pr(k+l+2\cdot h>W)\leq e^{-a\cdot W+b}$ であることを簡単に確認できる。$K(oracle^u,s)=\sqrt{|future\,(x,G_s^{oracle^u})|}$である。ブロック$z_m$は$dist_gap(G_s^{oracle^u},z_m)>K(oracle^u,s)$の場合にはアルゴリズム3の9行目の$M(oracle^u,s)$ にカウントされる。(63) から z_mが$M(oracle^u,s)$ の値を1増加させない確率は以下を上限とすると結論付けることができる。

$$\Pr\left(dist\_gap(G_s^{oracle^u}, z_m) \leq K(oracle^u, s)\right) \leq \tag{64}$$

$$\sum_{n=0}^{\infty}\binom{n+m-1}{m}\cdot(1-\alpha)^n\cdot\alpha^m\cdot\left(\frac{\alpha}{1-\alpha}\right)^{(n-m-K(oracle^u,s)-W-C_t)^+} <$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{-W-C_t-K(oracle^u,s)}\cdot\sum_{n=0}^{\infty}\binom{n+m-1}{n}\cdot(1-\alpha)^n\cdot\alpha^m\cdot\left(\frac{\alpha}{1-\alpha}\right)^{(n-m)^+} = \tag{65}$$

$$\left(\frac{\alpha}{1-\alpha}\right)^{-W-C_t-K(oracle^u,s)}\cdot\left(\Pr_{n\sim Z(m,1-\alpha)}(n>m)+\Pr_{n\sim Z(m,\alpha)}(n\leq m)\right), \tag{66}$$

$Z(n,p)$ は負の二項分布のランダム変数である。

次に最後の項が $e^{-D\cdot m}$ を上限とすることの証明を目指す。証明は以下の補題19で示すものと非常に類似している。

パートV:

十分大きな*m*で$Z(1-\alpha,m)$ に従って分布する変数は中央値$m\cdot\frac{\alpha}{1-\alpha}$ で分散 $m\cdot\frac{\alpha}{(1-\alpha)^2}$ の標準変数に収束する[21]。 そのため、(66)内の二番目の被乗数は*m*の増加に従って以下に収束する。

$$\Pr_{z\sim\mathcal{N}(0,1)}\left(z\leq\frac{m-\frac{1-\alpha}{\alpha}\cdot m}{\sqrt{\frac{1-\alpha}{\alpha^2}\cdot m}}\right)+\Pr_{z\sim\mathcal{N}(0,1)}\left(z\geq\frac{m-\frac{\alpha}{1-\alpha}\cdot m}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot m}}\right)= \tag{67}$$

$$\Pr_{z\sim\mathcal{N}(0,1)}\left(z\geq\frac{\frac{1-\alpha}{\alpha}\cdot m-m}{\sqrt{\frac{1-\alpha}{\alpha^2}\cdot m}}\right)+\Pr_{z\sim\mathcal{N}(0,1)}\left(z\geq\frac{m-\frac{\alpha}{1-\alpha}\cdot m}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot m}}\right). \tag{68}$$

[20] / はここでは

$\max_{z\in G^{oracle}_{time(b_m)}\cap honest}\left\{\left|future_a\left(z,G^{oracle^u}_{time(b_m)}\right)\right|-\left|future_h\left(z,G^{oracle^u}_{time(b_m)}\right)\right|\right\}$ を表す。

[21] 我々はここでは補題14の証明で指定する仮定に依存している。この仮定に従い、最悪のケースでは*t*後に攻撃者が全てのブロックを作成直後に全てのノードに公開する。

以下の不等号はKomatu (1955)によるものである。$x\geq 0$ 及び$z\sim\mathcal{N}(0,1)$ とする。その場合、$\Pr(z>x)\leq\frac{1}{\sqrt{2\cdot\pi}}\cdot\frac{2\cdot e^{-x^2/2}}{x+\sqrt{2+x^2}}$ である。 $x_1:=\frac{\frac{1-2\cdot\alpha}{\alpha}\cdot m}{\sqrt{\frac{1-\alpha}{\alpha^2}\cdot m}}$ 及び $x_2:=\frac{\frac{1-2\cdot\alpha}{1-\alpha}\cdot m}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot m}}$. とする。

(68)の上限が得られる。

$$\frac{1}{\sqrt{2\cdot\pi}}\cdot\frac{2\cdot e^{-x_1^2/2}}{x_1+\sqrt{2+x_1^2}}+\frac{1}{\sqrt{2\cdot\pi}}\cdot\frac{2\cdot e^{-x_2^2/2}}{x_2+\sqrt{2+x_2^2}}\leq \tag{69}$$

$$C_1\cdot e^{-x_1^2/2}+C_2\cdot e^{-x_2^2/2}=C_1\cdot e^{-D_1\cdot m}+C_2\cdot e^{-D_2\cdot m}\leq C_3\cdot e^{-D_3\cdot m} \tag{70}$$

α に依存する正の定数 C_i, D_i (以下の定数にもあてはまる特性)

この項を $\left(\frac{\alpha}{1-\alpha}\right)^{-W-C_t-K(oracle^u,s)}$ で乗じる場合、以下が算出される。

$$\left(\frac{\alpha}{1-\alpha}\right)^{-W-C_t-K(oracle^u,s)} \cdot C_3 \cdot e^{-D_3 \cdot m} \leq \tag{71}$$

$$C_4 \cdot e^{-D_3 \cdot m + D_4 \cdot K(oracle^u,s)} = C_4 \cdot e^{-D_3 \cdot m + D_4 \cdot \sqrt{|future(x, G_s^{oracle^u})|}}. \tag{72}$$

そのため、M_1 が存在し、$m > \left|future\left(x, G_s^{oracle^u}\right)\right| > M_1$ となる場合、C_5, D_5 の最後の数式の上限は $C_5 \cdot e^{-D_5 \cdot m}$ である。

パート*VI:* ψ （期待値 M_1/λ ）後に条件 $\left|future\left(x, G_s^{oracle^u}\right)\right| \geq M_1$ が満たされる。$s_m := time(z_m)$ とし、$s_m \geq \psi$ と仮定する。$\sum_{m=\sqrt{|future(x, G_{z_m}^{oracle^u})|}+1}^{\infty} C_5 \cdot e^{-D_5 \cdot m} < \infty$ であるため、ファトゥの補題は $m^* > \sqrt{\left|future\left(x, G_{s_m}^{oracle^u}\right)\right|}$ が存在し、全ての $m \geq m^*$ で、$dist\_gap(z_m) > K(oracle^u, s_m)$ であることを示唆する。

z_{m^*} の予想される待ち時間は有限である[22]。$\tau = \max\{\psi, time(z_{m^*})\}$ と定義する。その場合、

$s \geq \tau$: $M(oracle^u, s) \geq \left|future_a\left(x, G_s^{oracle^u}\right) \cap G_{[t,s]}^{oracle} \setminus V_{x \prec y}(G_s^{oracle^u})\right| - m^*$.[23]

となる。

[22] $\Pr(m^* \geq r) \leq \sum_{m=\sqrt{|future(x, G_s^{oracle^u})|}+1}^{r-1} C_5 \cdot e^{-D_5 \cdot m}$ である。 そのため、

$$\mathbb{E}[m^*] \leq \sum_{r=\sqrt{|future(x, G_{z_m}^{oracle^u})|}+1}^{\infty} \sum_{m=\sqrt{|future(x, G_{z_m}^{oracle^u})|}+1}^{r-1} C_5 \cdot e^{-D_5 \cdot m} =$$

$$\sum_{m=\sqrt{|future(x, G_{z_m}^{oracle^u})|}+1}^{\infty} \sum_{r=m+1}^{\infty} C_5 \cdot e^{-D_5 \cdot m} = \sum_{m=\sqrt{|future(x, G_{z_m}^{oracle^u})|}+1}^{\infty} C_6 \cdot e^{-D_6 \cdot m} \leq$$

$$C_7 \cdot e^{-D_7 \cdot \sqrt{|future(x, G_{z_m}^{oracle^u})|}}.$$

となる。

補題19

$\psi \in [t, \infty)$ *が存在するため* $\Pr\left(\mathcal{E}_{t \to \infty}^{all}(x, y, \epsilon)^{\complement} \mid \mathcal{E}_t^v(x, y, \epsilon)\right) < \epsilon$ *である。さらに、* $\mathbb{E}[\psi - t] < \epsilon$ *である。*

証明

パート *I:* 全ての誠実なブロックが*x*に投票する場合、全ての誤差関数は0に収束する

ことを証明している。実際、事象$\mathcal{E}_t^v(x,y,\epsilon)$ は$f_{pre_mine}(l(G_t^v)) + f_{pre_pub}(n_j(G_t^v)) + f_{post_pub}(|future(x,G_t^v)|) + f_{post_mine}(n_x(G_t^v), g(G_t^v), l(G_t^v)) < \epsilon$を示唆する。合併上界と補題24, 29及び31により、確率$\geq 1-\epsilon$で以下の関係が成立する。

- $\max_{z \in G_t^{oracle} \cap honest} \left\{ \left| future_a\left(z, G_{time(x)}^{oracle}\right) \right| - \left| future_h\left(z, G_{time(x)}^{oracle}\right) \right| \right\} \leq l(G_t^v)$
- $|future_h(x, G_t^v)| \leq n_x$
- $\left| anticone_h\left(x, G_t^{oracle}\right) \right| \leq gap(x,G) + n_j =: j$

これらの関係を条件に、結論17により、事象$\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)$が$\geq 1 - f_{post_mine}(n_x(G_t^v), g(G_t^v), l(G_t^v))$の確率で発生する。全体的に見て、$\mathcal{E}_t^v(x,y,\epsilon)$を条件に事象$\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)$が$\geq 1-\epsilon$の確率で発生する。

パートII:

$\widehat{\mathcal{E}}_{t\to\infty}^{all}(x,y)$ 及び上記の関係を条件に、全ての$u \in honest$で時間の経過と共に$Risk(G_s^u, x, y)$の値が（ほぼ間違いなく）0になることを証明する[24]。

*s*の増加にともなって$f_{pre_mine}(l(G_s^u)) + f_{pre_pub}(n_j(G_s^u)) + f_{post_pub}(|future(x,G_s^u)|)$が0になるという結論は補題25, 32及び30に従うものである。$\epsilon_0 > 0$とする。有限期待値τ後に$f_{post_mine}(n_x(G_s^u), g(G_s^u), l(G_s^u)) < \epsilon_0$であることを証明する。

以下を主張する。

$$M(oracle^u, s) + g(oracle^u, s) - n_x(oracle^u, s) \geq -2 \cdot \left| G_{[time(x),t]}^{oracle} \right| - m^* \qquad (73)$$

m^*は補題18で明記する変数である。最初に$malicious \cap G_s^{oracle} \subseteq G_s^{oracle^u}$と仮定する。$future\left(x, G_s^{oracle^u}\right)$を以下のように分解する。

- $G_{[time(x),t]}^{oracle}$内のブロック。このセット内のブロック数が*s*と共に増加しないことは明白である。これらの寄与率の下限は$-2 \cdot \left| G_{[time(x),t]}^{oracle} \right|$である。
- $V_{x \prec y}(G_s^{oracle^u}) \setminus G_t^{oracle}$内のブロック。このセット内の全ての*z*は$g(oracle^u, s)$に$(+1)$を追加する。*z*は$M(oracle^u, s) - n_x(oracle^u, s)$の値を1以上減少させることはできないため、このセットの寄与率は少なくとも0である[25]。

[24] z_mの予想される待ち時間は最後の項を$\alpha \cdot \lambda$で割った数値である。

[25] $\mathbb{E}[m^*]$は*t*の時間までの事象によって決定される。前の証明の数式内の予想される値を採用する。$\left| future\left(x, G_{s_m}^{oracle^u}\right) \right|$の値（そして$s_m$自身）上の分布は$\left| future\left(x, G_t^{oracle^u}\right) \right|$を条件と

している（予想される値を最大化する $oracle^u$）。

[24]我々は $\max\{Risk(G_s^u, x, y)\}$ が0になることを証明する必要がある。しかし、我々の以下の分析では *u*に関して最悪の事態を想定している。つまり、*u*からのメッセージと*u*へのメッセージは*d*の遅延で到着することを想定している。これらの事象は最悪の事態に相当するものであるため、*u*を固定の誠実なノードとする。

[25]関係 $|future_h(x, G_t^v)| \leq n_x$ を条件とすることで、我々は全ての誠実なブロックがこのカテゴリーに属することを知っているため、より厳格な限界
$M(oracle^u, s) + g(oracle^u, s) - n_x(oracle^u, s) \geq$
$-\left|G^{oracle}_{[time(x),t]} \cap malicious\right| - m^*$ にたどり着く。

. $G_s^{oracle^u} \setminus \left(V_{x \prec y}(G_s^{oracle^u}) \cup G_t^{oracle}\right)$ 内のブロック： 補題18では事象 $\widehat{\mathcal{E}}^{all}_{t\to\infty}(x, y)$ を条件に τ 後に公開され、 $V_{x\prec y}(G_s^{oracle^u})$ に属しない少なくとも $\left|future_a\left(x, G_s^{oracle^u}\right) \cap G^{oracle}_{[t,s]}\right| - m^*$ 個のブロックが存在すること、つまり $g(oracle^u, s)$ に (-1) を追加し[26]、$M(oracle^u, s)$ の値に $(+1)$ を追加することを保証している。言い換えると、セット $future_a\left(x, G_s^{oracle^u}\right) \cap G^{oracle}_{[t,s]} \setminus V_{x\prec y}(G_s^{oracle^u})$ の多くとも m^* 個のブロックが (-1) を $g(oracle^u, s)$ に加え、$M(oracle^u, s)$ の値の $(+1)$ の増加によって打ち消されない。

そのため、このセットの寄与率の下限は $-m^*$ である。

パートIII:

以下を主張する。

$$M\left(G_s^u\right) + g\left(G_s^u\right) - n_x\left(G_s^u\right) \geq -2 \cdot \left|G^{oracle}_{[time(x),t]}\right| - m^* \tag{74}$$

$C(z)$ を(73)に対する*z* の寄与率とし、 $c(z)$ を(74)に対する寄与率とする。最初に、$C(z) \geq -2$ とする。そのため、以前と同じように全ての $z \in G^{oracle}_{[time(x),t]}$ の寄与率は少なくとも $-2 \cdot \left|G^{oracle}_{[time(x),t]}\right|$ である。

$z \in G_s^{oracle^u} \setminus G_t^{oracle}$ と仮定し、*x* ≺ *y*と投票すると仮定する。その場合、*z*は $M(oracle^u, s)$ にカウントされないため、
$M(oracle^u, s) + g(oracle^u, s) - n_x(oracle^u, s)$ に対する寄与率は $0 + 1 - 1 = 0$, i.e., $c(z) = 0$. である。そして引数は同じ $C(z) = 0$ である。

$z \in G_s^{oracle^u} \setminus G_t^{oracle}$ であり、*y* ≺ *x*と投票すると仮定する。その場合、

$z \in malicious$ となる（$\widehat{\mathcal{E}}^{all}_{t\to\infty}(x,y)$ の条件付けによる）。補題18の分析では以下の最悪のケースを想定した:任意の3つのブロック $v,z,w \in G_s^{oracle^u}$ で $v,z \in malicious$ で $w \in honest$ であり、*v*は $z \prec w$ に対して好意的に投票する[27]。この攻撃者ブロックの投票に関する最悪の想定下では、$G_s^{oracle^u} \setminus G_s^u$ には攻撃者ブロックしか含まれないため、 $dist\_gap(z, G_s^{oracle^u}) \le dist\_gap(z, G_s^u)$ である。そのため、*z*を $M(oracle^u, s)$ にカウントする場合、$M\left(G_s^u\right)$ 内でもカウントされる。特に $C(z) \ge c(z)$ である。
結果として、補題18の分析を用いて以下が導き出される。

$$-m^* \le \sum_{z \in G_s^{oracle^u} \setminus G_t^{oracle}} c(z) \le \sum_{z \in G_s^{oracle^u} \setminus (G_t^{oracle} \cup V_{x \prec y}(G_s^{oracle^u}))} c(z) = \sum_{z \in G_s^u \setminus (G_t^{oracle} \cup V_{x \prec y}(G_s^{oracle^u}))} c(z) + \sum_{z \in G_s^{oracle^u} \setminus (G_s^u \cup V_{x \prec y}(G_s^{oracle^u}))} c(z) \le \sum_{z \in G_s^u \setminus (G_t^{oracle} \cup V_{x \prec y}(G_s^{oracle^u}))} c(z) \le \sum_{z \in G_s^u \setminus (G_t^{oracle} \cup V_{x \prec y}(G_s^{oracle^u}))} C(z).$$

全体的に見て、$M\left(G_s^u\right) + g\left(G_s^u\right) - n_x\left(G_s^u\right) \ge -2 \cdot \left|G^{oracle}_{[time(x),t]}\right| - m^*$ となる。

パートIV:

証明の残りの部分では場合によって $n_x\left(G_s^u\right)$ を省略し、単純に*n*xと表記する場合がある。残りの変数でも便宜上の問題で類似の表記を行う場合がある。補題 25 と 32では定数*a, b*及び*W* が存在し、$\Pr\left(k+l+2\cdot h+j > W\right) \le e^{-a\cdot W+b}$ であることが示唆される（前の補題の証明と同じ、ただし同じ定数とは限らない）。

[26]これらの変数では強い投票者だけがカウントされるために0を追加することはできない。

[27]実際、誠実なブロックでは誠実な投票者だけをカウントしている。補題20のように疑似投票者を使用することでこれを正式化することができる。

$e^{-a\cdot W+b} < \epsilon_0/4$ となる *W*を取る。そのため、確率が $\ge 1-\epsilon_0/4$ の場合:

$$f_{post\_mine}\left(n_x\left(G_s^u\right), g\left(G_s^u\right), l\left(G_s^u\right)\right) =$$

$$\sum_{k=0}^{\infty} \mathcal{P}_{oiss}(3\cdot d\cdot(1-\alpha)\cdot\lambda, k) \cdot \sum_{h=0}^{\infty} \mathcal{P}_{oiss}(d\cdot(1-\alpha)\cdot\lambda, h) \cdot \tag{75}$$

$$\left(\sum_{m'=M}^{\infty} \binom{n_x+j+h+m'-1}{m'} \cdot (1-\alpha)^{n_x+j+h} \cdot \alpha^{m'}\right)^{-1} \cdot \tag{76}$$

$$\sum_{m=M}^{\infty} \binom{n_x+j+h+m-1}{m} \cdot (1-\alpha)^{n_x+j+h} \cdot \alpha^m \cdot \left(\frac{\alpha}{1-\alpha}\right)^{(g-2\cdot h-k-j-l-(m-M))^+}. \tag{77}$$

十分大きい n_x の場合、この項は多くとも以下から $\epsilon_0/4$ 離れている。

$$\left(\frac{\alpha}{1-\alpha}\right)^{g+M-n_x-W} \cdot \left(\sum_{m'=M}^{\infty} \binom{n_x+m'-1}{m'} \cdot (1-\alpha)^{n_x} \cdot \alpha^{m'}\right)^{-1} \cdot \tag{78}$$

$$\sum_{m=M}^{\infty} \binom{n_x+m-1}{m} \cdot (1-\alpha)^{n_x} \cdot \alpha^{m} \cdot \left(\frac{\alpha}{1-\alpha}\right)^{(n_x-m)^+} . \tag{79}$$

パートV:

(78)の最初の被乗数の場合、この証明のパートIIによって、有限期待 τ 後に $M\left(G_s^u\right)+g\left(G_s^u\right)-n_x\left(G_s^u\right) \geq -\left|G^{oracle}_{[time(x),t]}\right| - m^* =: D_2$ となる（時間 τ が決定する定数）。$s \geq \tau$ と仮定する。項 $\left(\frac{\alpha}{1-\alpha}\right)^{g+M-n_x-W}$ の上限は $e^{D_3 \cdot D_4}$ (with $D_3 = \ln\left(\frac{1-\alpha}{\alpha}\right)$) であると結論付ける。そのため、(78)が消滅することを証明するためには以下が消滅することを証明するだけで十分である。

$$\left(\sum_{m'=M}^{\infty} \binom{n_x+m'-1}{m'} \cdot (1-\alpha)^{n_x} \cdot \alpha^{m'}\right)^{-1} \cdot \tag{80}$$

$$\sum_{m=M}^{\infty} \binom{n_x+m-1}{m} \cdot (1-\alpha)^{n_x} \cdot \alpha^{m} \cdot \left(\frac{\alpha}{1-\alpha}\right)^{(n_x-m)^+} \tag{81}$$

最後の項は以下に等しい。

$$\left(\Pr_{m\sim Z(1-\alpha,n_x)}(m \geq M)\right)^{-1} \cdot \left(\Pr_{m\sim Z(\alpha,n_x)}(m \leq n_x) + \Pr_{m\sim Z(1-\alpha,n_x)}(m \geq n_x)\right). \tag{82}$$

十分に大きな n_x の場合、$Z(1-\alpha, n_x)$ に従って分布する変数は中央値 $n_x \cdot \frac{\alpha}{1-\alpha}$ で分散 $n \cdot \frac{\alpha}{(1-\alpha)^2}$ の通常変数に収束する。そのため、最後の項は多くとも以下から $\epsilon_0/4$ だけ離れている。

$$\left(\Pr_{z\sim\mathcal{N}(0,1)}\left(z \geq \frac{M - \frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}\right)\right)^{-1} \cdot \tag{83}$$

$$\left(\Pr_{z\sim\mathcal{N}(0,1)}\left(z \leq \frac{n_x - \frac{1-\alpha}{\alpha}\cdot n_x}{\sqrt{\frac{1-\alpha}{\alpha^2}\cdot n_x}}\right) + \Pr_{z\sim\mathcal{N}(0,1)}\left(z \geq \frac{n_x - \frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}\right)\right) = \tag{84}$$

$$\left(\Pr_{z\sim\mathcal{N}(0,1)}\left(z \geq \frac{M - \frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}\right)\right)^{-1} \cdot \tag{85}$$

$$\left(\Pr_{z\sim\mathcal{N}(0,1)}\left(z \geq \frac{\frac{1-\alpha}{\alpha}\cdot n_x - n_x}{\sqrt{\frac{1-\alpha}{\alpha^2}\cdot n_x}}\right) + \Pr_{z\sim\mathcal{N}(0,1)}\left(z \geq \frac{n_x - \frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}\right)\right). \tag{86}$$

$x \geq 0$では Komatu (1955)による以下の不等号を使用し、標準通常変数 $z \sim \mathcal{N}(0,1)$: $\frac{1}{\sqrt{2\cdot\pi}} \cdot \frac{2\cdot e^{-x^2/2}}{x+\sqrt{4+x^2}} \leq \Pr(z > x) \leq \frac{1}{\sqrt{2\cdot\pi}} \cdot \frac{2\cdot e^{-x^2/2}}{x+\sqrt{2+x^2}}$. とする。

$x_1 := \frac{M-\frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}$, $x_2 := \frac{\frac{1-\alpha}{\alpha}\cdot n_x - n_x}{\sqrt{\frac{1-\alpha}{\alpha^2}\cdot n_x}}$ 及び $x_3 := \frac{n_x - \frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}$ とする。

(86)の上限を導き出す。

$$\sqrt{\pi/2}\cdot\left(x_1+\sqrt{4+{x_1}^2}\right)\cdot e^{{x_1}^2/2}\cdot\left(\frac{1}{\sqrt{\pi/2}}\cdot\frac{e^{-x_2^2/2}}{x_2}+\frac{1}{\sqrt{\pi/2}}\cdot\frac{e^{-x_3^2/2}}{x_3}\right) = \tag{87}$$

$$\left(x_1+\sqrt{4+{x_1}^2}\right)\cdot e^{{x_1}^2/2}\cdot\left(\frac{e^{-x_2^2/2}}{x_2}+\frac{e^{-x_3^2/2}}{x_3}\right) \tag{88}$$

さらに、大きな n_xで $x_2 \geq C_2\cdot\sqrt{n_x}$、正の定数 C_iで $x_3 \geq C_3\cdot\sqrt{n_x}$ であることを観察する (これは以下の全ての定数にあてはまる)。そのため、 $\frac{\left(x_1+\sqrt{4+{x_1}^2}\right)}{\min\{x_2,x_3\}} \leq C_1/\max\{C_2,C_3\} =: D_1$ である。上の項は最大で乗法因子D_1まで以下の上限によって拘束される。

$$e^{x_1^2/2-x_2^2/2}+e^{x_1^2/2-x_3^2/2} = e^{0.5\cdot\left(\left(\frac{M-\frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}\right)^2-\left(\frac{\frac{1-\alpha}{\alpha}\cdot n_x-n_x}{\sqrt{\frac{1-\alpha}{\alpha^2}\cdot n_x}}\right)^2\right)}+e^{0.5\cdot\left(\left(\frac{M-\frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}\right)^2-\left(\frac{n_x-\frac{\alpha}{1-\alpha}\cdot n_x}{\sqrt{\frac{\alpha}{(1-\alpha)^2}\cdot n_x}}\right)^2\right)} \leq e^{0.5\cdot\left(\frac{(1-\alpha)^2}{\alpha\cdot n_x}\cdot\left(M-\frac{\alpha}{1-\alpha}\cdot n_x\right)^2-\frac{(1-2\cdot\alpha)^2}{1-\alpha}\cdot n_x\right)}+e^{0.5\cdot\left(\frac{(1-\alpha)^2}{\alpha\cdot n_x}\cdot\left(M-\frac{\alpha}{1-\alpha}\cdot n_x\right)^2-\frac{(1-2\cdot\alpha)^2}{\alpha}\cdot n\right)}. \tag{89}$$

関係 $\left|future_h\left(x,G_t^{oracle}\right)\right| \leq n_x,\ M \leq future_a\left(x,G_s^u\right)$を条件としているため、期待される値は多くとも $\frac{\alpha}{1-\alpha}\cdot n_x$ となる。あらゆる$\delta > 0$で、大数の法則によって、τ （有限期待）後に、 $\forall s \geq \tau : M \leq (1+\delta)\cdot\mathbb{E}[M] \leq (1+\delta)\cdot\frac{\alpha}{1-\alpha}\cdot n_x.$ となる。

結果的に、(89)の上限は以下となる。

$$e^{0.5\cdot\frac{(1-\alpha)^2}{\alpha\cdot n_x}\cdot\left(M-\frac{\alpha}{1-\alpha}\cdot n_x\right)^2-0.5\cdot\frac{(1-2\cdot\alpha)^2}{1-\alpha}\cdot n_x}+e^{0.5\cdot\frac{(1-\alpha)^2}{\alpha\cdot n_x}\cdot\left(M-\frac{\alpha}{1-\alpha}\cdot n_x\right)^2-0.5\cdot\frac{(1-2\cdot\alpha)^2}{\alpha}\cdot n_x} \leq \tag{90}$$

$$e^{0.5\cdot\frac{(1-\alpha)^2}{\alpha\cdot n_x}\cdot\left(\delta\cdot\frac{\alpha}{1-\alpha}\cdot n_x\right)^2-0.5\cdot\frac{(1-2\cdot\alpha)^2}{1-\alpha}\cdot n_x}+e^{0.5\cdot\frac{(1-\alpha)^2}{\alpha\cdot n_x}\cdot\left(\delta\cdot\frac{\alpha}{1-\alpha}\cdot n_x\right)^2-0.5\cdot\frac{(1-2\cdot\alpha)^2}{\alpha}\cdot n_x} \leq \tag{91}$$

$$e^{R_1/n_x-R_2\cdot n_x}+e^{R^3/n_x-R_4\cdot n_x} \leq e^{-R_5\cdot n_x}, \tag{92}$$

正の定数R_iでは十分大きな n_xで最後の不等号が成り立ち、前の不等号は十分小さな $\delta\text{'s}\ (\delta < 1/n_x)$で成り立つ。

n_xが $n_x > \ln\left(4\cdot D_1/\epsilon_0\right)/R_5$ より大きいとすると、十分大きな n_xで以下の結論が成り立つ。

$$f_{post\_mine}\left(n_x\left(G_s^u\right),g\left(G_s^u\right),l\left(G_s^u\right)\right) < 4\cdot\epsilon_0/4 = \epsilon_0. \tag{93}$$

($\forall j \in honest : n_x(u,\psi)$ の最初のτ の予想される待ち時間は全ての誠実なブロックの作成で多くとも$n_0 \cdot ((1-\alpha)\cdot\lambda)^{-1} + d$: it is $1/((1-\alpha)\cdot\lambda)$であり、最後のブロックが全てのノードにたどり着くまでの時間は*d*である。)
以下では補題19の証明で使用するものと同じテクニックを用いて進捗特性（提案9）を証明している。これは全ての誠実なノードで時間の経過と共に全ての誤差関数を集計する項が消滅するという証明である。特に、*v*では（最初にトランザクションを受け入れたノード）、ϵ' より小さくなる。以下では似たような形でこの引数を使用することで弱い生存性（提案10）の証明を行っている。実際、後者では$y = NULL$の場合を考慮するだけでよい。この場合、公開済みの全てのブロックは*x*に投票する強い投票者である。そのため、補題14やそれに続く分析を経ることなく誤差関数の収束を保証する。

G. 弱い生存性の証明 (ブロック)

誤差関数 $f_{pre_mine}(l(G_s^u)),\quad f_{pre_pub}(n_j(G_s^u))$と
$f_{post\_pub}(|future(x,G_s^u)|)$ は*s*の増加と共に0になることが判明している。あらゆる$s < \psi$において、$y \notin G_s^{pub}$ であるためアルゴリズム3の5行目によって、$g(G_s^v) = |future(x,G)| = n_x(G_s^u)$ 及び $M(G_s^v) = 0$となる。特に、関係(73)は明確に満たされており、補題19の証明における分析があてはまり、時間の経過と共に項 f_{post_mine} が消滅することが証明される。特に、これらの関数は指数関数的に減少するために、順序$\mathcal{O}(\ln(1/\epsilon))$ 内で誠実なブロックがいくつか作成された後にϵ以下となる。このために予想される待ち時間はこの数字を $(1-\alpha)\cdot\lambda$で割る（そして全ての誠実なブロックで*d*を足すことでこれらのブロックを受信する）ことによって算出される。

H. 進捗の証明 (ブロック)

これは事象$\widehat{\mathcal{E}}^{all}_{\to\infty}t(x,y)$を条件に*s*の無限の増加に対応して
$f_{pre\_mine}(l(G_s^v)) + f_{pre\_pub}(n_j(G_s^u)) + f_{post\_pub}(|future(x,G_t^u)|) +$
$f_{post\_mine}(n_x(G_s^u), g(G_s^u), l(G_s^u))$ が消滅することを証明した補題19の証明に従うものである。補題14では確率ϵまで、事象$\mathcal{E}_t^v(x,y,\epsilon)$は$\widehat{\mathcal{E}}^{all}_{\to\infty}t(x,y)$ 内に含まれることが証明されている (例：前者が確率$\geq 1-\epsilon$の事象と交わる場合)。

I. 安全性の証明

パート *I:* D インプットG_s^u （誠実な*u*）、*tx*及び*subG*（*subG*は（仮想の可能性もある）ブロックの一部である）が判明している場合に
$risk_{acc}(G_s^u, tx, subG)$ $(risk_{rej})$ によってアルゴリズム4のアウトプットを表記する。$z \in [tx] \cap subG$ の場合、$risk_{acc}^z(G_s^u, tx, subG)$ によっての2行目のループが

z上で終了する場合の*risk*変数の値を表記する。

同様に、*RiskTxReject*内の変数*minrisk*に関して$minrisk^{z}_{rej}(G^u_s, tx, subG)$を表記する。

確率$> 1 - risk_{acc}(G^v_t, tx, subG)$の場合に、$\tau_{acc}$の有限期待が存在し、全ての$s \geq \tau_{acc}$、全ての$u \in honest$、そして全ての$subG' \supseteq subG$で以下が成立する。

$$risk_{acc}(G^v_t, tx, subG) \geq risk_{acc}(G^u_s, tx, subG') \tag{94}$$

同様に、確率$> 1 - risk_{rej}(G^v_t, tx, subG)$の場合に、$\tau_{rej}$の有限期待が存在し、全ての$s \geq \tau_{rej}$及び全ての$u \in honest$で以下が成立する。

$$risk_{rej}(G^v_t, tx, subG) \geq risk_{rej}(G^u_s, tx, subG). \tag{95}$$

これをサイズ$< k$の全ての$subG$でこれを証明済みであると仮定する。これをサイズkの$subG_k$でも証明する。

$risk_{acc}$の定義によって、$z_{tx} \in subG_k \cap [tx]$が存在し、
$risk_{acc}(G^v_t, tx, subG') = risk^{z_{tx}}_{acc}(G^v_t, tx, subG')$となる。

パートII:

Z_2は$z_1 = z_{tx}$の第1ループの反復内の第3ループ変数z_2のインスタンス化のセットである。提案8と9により、あらゆるϵ'で確率
$\geq 1 - Risk\left(G^v_t, (vote(z'))_{z' \in C}, z_1, z_2\right)$の$\forall z_2 \in Z_2$で$\tau$（有限期待）後に、
$\forall z'_2 \in (G^u_s \setminus G^v_t) \cup \{z_2\} : Risk\left(G^u_s, (vote(z))_{z \in C}, z_{tx}, z'_2\right) \leq \epsilon'$
が成り立つ。

さらに、提案19の証明において*s*の時点でこの特性が成り立つ最小のϵ'は*n*(*s*と共に直線的に増加する*s*)と共に急速に減少する。そのため、τより大きな全て*s*のにおいて、
$\sum_{z'_2 \in (G^u_s \setminus G^v_t) \cup Z_2} : Risk\left(G^u_s, (vote(z))_{z \in C}, z_{tx}, z'_2\right) \leq$
$\sum_{z'_2 \in Z_2} Risk\left(G^v_t, (vote(z))_{z \in C}, z_{tx}, z'_2\right)$が成り立つ。

パートIII:

同様に、提案8によって、少なくとも$Risk\left(G^v_t, (vote(z))_{z \in C}, z_{tx}, \emptyset\right)$の確率で、$\tau$（有限期待）後に、
$Risk\left(G^u_s, (vote(z))_{z \in C}, z_{tx}, \emptyset\right) \leq Risk\left(G^v_t, (vote(z))_{z \in C}, z_{tx}, \emptyset\right)$が成立する。

パートIV:

$\epsilon_i(G^u_s, tx, subG)$は*RiskTxAccept*の7行目（インプット$(G^u_s, tx, subG)$の場合）の

*RiskTxAccept*への呼び出し、と*RiskTxAccept*（これらのインプット）の9行目の*RiskTxAccept*への呼び出しによって返される一連の値である。帰納法の仮定によって、確率$\geq 1-\epsilon_i$の場合にτ経過後に$\epsilon_i(G_s^u, tx, past(z_{tx})) \leq \epsilon_i(G_t^v, tx, past(z_{tx}))$が成立する28。

パートV:

上記では確率$\geq 1 - risk_{acc}^{z_{tx}}(G_t^v, tx, subG)$の場合に値$risk_{acc}^{z_{tx}}(G_s^u, tx, subG')$の増加値の合計の上限は$risk_{acc}^{z_{tx}}(G_t^v, tx, subG)$の値の増加値の合計となることを証明している。全ての$s \geq \tau$において、τは有限期待である。

28厳密にはこの不等号の両側のインデックスiは注意深く表記する必要がある。面倒な表記を避けるあために、我々は読者の理解に依存している。正式ではないが、*RiskTxAccept*内（インプット$(G_t^v, tx, subG)$の場合）のループ変数の全てのインスタンス化は*RiskTxAccept*の未来の呼び出しによっても実現される（インプット$(G_s^u, tx, subG')$の場合）。そのため、我々は前者の増加と後者の増加の結果を比較している。これは逆の場合もあてはまる($z_1 = z_{tx}$): $z_1 = z_{tx}$上の第1ループの反復内で*past* (z_{tx})が時間の経過によって変化しないために*RiskTxAccept*と*RiskTxReject*に対して全く同じ呼び出しを行っている。

$risk_{acc}(G_s^u, tx, subG') \leq risk_{acc}^{z_{tx}}(G_s^u, tx, subG')$及び$risk_{acc}(G_t^v, tx, subG) = risk_{acc}^{z_{tx}}(G_t^v, tx, subG)$であるため、確率$\geq 1 - risk_{acc}^{z_{tx}}(G_t^v, tx, subG)$の場合に不等号$risk_{acc}(G_s^u, tx, subG') \leq risk_{acc}(G_t^v, tx, subG)$が成立する。

パートVI:

同様の論理によって*RiskTxReject*に関する帰納法手順が証明される。この証明内の違いは$risk_{rej}^{z_1}$が合計ではなく最小値であるため、$anticone(z_1, G_s^u)$が時間の経過と共に増加し、$risk_{rej}^{z_1}$の値をさらに削減する可能性のあるループ反復が追加される可能性があるという事実を無視できる。 *RiskTxReject*に関する帰納的主張は$subG' = subG$の場合に限定される。そのため、最初のループ変数は$Z_G([tx]) \cap subG$から選択されるためセット$Z_G([tx])$が時間と共に成長する可能性があるという事実は重要ではない。そのため、確率$\geq 1 - risk_{rej}(G_s^u, tx, subG_k)$の場合に、有限期待の$\tau$が存在し、全ての$s \geq \tau$と全ての$u \in honest$: $risk_{rej}(G_s^u, tx, subG_k) \leq risk_{rej}(G_t^v, tx, subG_k)$において$risk_{rej}(G_t^v, tx, subG_k)$であると結論付けられる。これによって帰納的主張の証明が完了する。

パートVII:

アルゴリズム6では*RiskTxAccept*がϵより小さい値を返した場合にのみ*tx*を含むセットを返す。上記の主張では$risk_{acc}(G_t^v, tx, G_t^v) < \epsilon$で確率が$\geq 1-\epsilon$の場合、有限期

待のτの全ての$s > \tau$そして全ての$u \in honest$で$risk_{acc}(G_s^u, tx, G_s^u) < \epsilon$となることを示唆している。言い換えると、$\mathcal{A}_t^v(tx, \epsilon)$を条件に、事象$\cap_{u \in honest, s \in (\tau(t), \infty)} \mathcal{A}_s^u(tx, \epsilon)$が$\geq 1 - \epsilon$の確率で発生する。

J. 生存性の証明

$G = G_t^v$で$z_1 \in Z_G([tx])$を固定する。$\psi(t)$: $conflict(tx) \cap G_s^{pub} = \emptyset$という条件は*RiskTxAccept*の6行目と7行目が$risk_{acc}(G_s^u, tx, subG)$の値に寄与しないことを示唆している。$\sum_{[tx_i] \in inputs(tx)} RiskTxAccept(G_t^v, (vote(z))_{z \in C}, [tx_i], G_t^v) < \epsilon/2$という仮定は確率$\geq 1 - \epsilon/2$の場合、4番目のループの$risk_{acc}^{z_1}(G_s^u, [tx], G_s^u)$の値に対する全体的な寄与率は多くとも$\epsilon/2$（$\tau$後）であることを示唆している。最後に、提案10により、3行目の$risk_{acc}^{z_1}$に対する寄与率は有限期待のτ後に$\epsilon/2$以下となる。 有限期待のτ後に$risk_{acc}^{z_1}(G_s^u, [tx], G_s^u)$の値は全ての$s \geq \tau$及び$u \geq s$において$\epsilon/2 + \epsilon/2 = \epsilon$以下となるため、$risk_{acc}(G_s^u, [tx], G_s^u) < \epsilon$となり、事象$\cap_{u \in honest, s \in (\tau(t), \infty)} \mathcal{A}_s^u(tx, \epsilon)$が示唆される。

K. 進捗の証明

この提案の証明は提案2の証明の構造と類似している。我々は既に帰納法によって3行目の$risk_{acc}^{z_{tx}}$の値に対する寄与率（そして同様に$risk_{rej}^{z_{tx}}$の値）そして6行目が0になり、7行目と9行目の増加値が0になるという主張を行っている。

そのため、$\geq 1 - risk_{acc}(G_t^v, tx, G_t^v)$の確率で時間の経過と共に$risk_{acc}(G_s^v, tx, G_s^v)$が0になる。$\epsilon > risk_{acc}(G_t^v, tx, G_t^v)$であるため、$\geq 1 - \epsilon$の確率で全ての$G_s^u$と$s \geq \tau$及び$u \in honest$でアルゴリズム6が*tx*を含むセットを返すと結論付ける。