Plasma

スケーラブルな自律型スマートコントラクト

ジョセフ・プーン joseph@lightning.network
ヴィタリック・ブテリン vitalik@ethereum.org

2017年8月11日

ワーキングドラフト https://plasma.io

概要

Plasmaは、世界中の分散型金融アプリケーションのほとんどを扱えるたったひとつのブロックチェーンを可能にする、秒あたりの状態の更新を多くの量（潜在的に十億）までスケールができるスマートコントラクトの強制執行と、インセンティブのために提案されたフレームワークである。これらのスマートコントラクトは、トランザクションの状態遷移を強制する基礎となるブロックチェーン（例：Ethereum）に最終的に依存するネットワークの取引手数料を介して、自律的に動作を継続する動機づけがされている。

私たちは、金融活動だけでなく、中央集権型サーバファームへの代替を提案するであろう、世界的で永続的なデータ・サービスの経済的インセンティブを構築も行う、自律分散的なアプリケーションのための処理の拡張方法を提案する。

Plasmaは、設計の二つの主要部分から構成されています：すべてのブロックチェーンの計算を、MapReduce関数のセットとして、およびナカモトコンセンサスのインセンティブがblock withholding攻撃を阻止することを理解した上で、既存のブロックチェーンの上にPoS(Proof of Stake)トークンを結合するためのオプションの方法として、再構築することである。

この構造は、詐欺の証拠を使用することによって状態遷移を親チェーンで強制することができるスマートコントラクトをメインブロックチェーンに構成することにより、達成される。私たちは、ツリー階層にブロックチェーンを構成し、ブロックチェーンの履歴とマークル証明を約束したMapReduce計算を強制した、個々の独立した分岐ブロックチェーンとして、それぞれを扱う。親チェーンによって執行された子チェーンに、誰かの台帳の取引を作成することで、その人は最小限の信頼（Rootchainの可用性と正確性を仮定する）で、信じられないほどのスケールを実現することができる。

非グローバルデータのグローバルな執行の周りの最大の複雑さは、データの可用

性とBlock withholding攻撃を中心に展開され、データの継続的な正しい実行を動機づけ、強制するメカニズムを作成しながら、障害のあるチェーンからの脱出を可能にすることによって、Plasmaは、この問題の緩和策を持っている。

障害のない状態の間、Rootchain（すなわちEthereum）に、定期的にマークル値だけがブロードキャストされているため、これは非常にスケーラブルで、低コストな取引および計算を可能にすることができる。Plasmaは、高いスケールで持続的に動作する、分散型アプリケーションを可能にする。

1　スケーラブルなマルチパーティ計算

一般的に、ブロックチェーンにおいて、正確さを強制するための解決策は、すべての参加者がチェーン自体を検証していることである。

新しいブロックを受け入れるには、正確さを保証するために、完全にブロックを検証する必要がある。

ブロックチェーントランザクションの容量を拡張するための、多くの努力（例えば、ライトニングネットワーク[1]）は、忠実なボンド（アサート/チャレンジ契約）を構築するために、タイムコミッティーを使用する必要があり、そのため、状態を強制するために、アサートされたデータは、ブロックチェーンの参加者のための紛争期間の対象でなければならない。

このアサート/チャレンジの設計は、1つの特定の状態が正しいことを主張することができ、もしその値が正しくない場合、一定の合意時間の前に、別の観察者がその主張に挑戦証拠を提供できる、紛争期間が存在する。

詐欺や障害のある行動が発生した場合には、ブロックチェーンは、障害のある行動者を罰することができる。

これは、もし誤った状態がアサートされた場合にのみ、強制することが奨励される、参加者のためのメカニズムを作っている。

このアサート/チャレンジ証明構造を有することにより、当事者は、ルートチェーン上（例えばイーサリアム[2] [3]）の非当事者にグラウンドトゥルースを主張することができる。

アサート/チャレンジ証明構造は、支払いのために使用されるだけでなく、ブロックチェーンがコントラクトの裁定層になるための、計算自体の拡張に使用することができる。

しかし、仮定は、すべての当事者が、計算の検証の参加者であるということである。

例えば、ライトニングネットワークでは、すべての当事者が計算の検証の参加者であり、そのためコントラクトの状態を計算する約束を確立できる。（例、条件付き状態の、

マルチシグネチャ取引の、事前に署名されたツリーと共に）

これらの構造は、スケールで非常に強力な計算を可能にするが、しかしながら、外部の状態の多くの合計値/マークル値（全体のシステム/市場、シャードされた/不完全な大量のデータの計算、多くの貢献者の合計値）を必要とするという、いくつかの問題がある。

マルチパーティーオフチェーン状態への約束のこの形式（ステートチャンネル[4]）は、完全に計算を検証するための参加者、もしくは計算自体に確立されたかなりの量の信頼、またはシングルラウンドのゲーム、を必要とする。

また、通常は”ラウンド”の仮定があることにより、コントラクトの開始の前に、実行パスが完全にアンロールされなければならず、参加者にExitとオンチェーンでの高価な計算を強制する機会を与える。（どのパーティーが停止されているかを証明することはできない）

その代わりに、我々は、オフチェーンで、しかし最終的には、最小限のオンチェーンの更新とともに、毎秒数十億の計算までスケーラブルである強制的なオンチェーンで、計算が可能になるシステムを設計しようとする。

これらの状態の更新は、詐欺の証明によって強制される、正しい行動に向けてインセンティブされた、Proof of Stakeの承認者の自律型セットを横断的に発生させる、これは簡単に計算サービスを停止できる単一の行動者なしに、計算が起こるの許可する。

これは、データ可用性の問題（すなわちblock withholding）の周りの課題を最小限に抑えること、ルートチェーン上のリスクディスカウンティッドな取引手数料を防ぐ、ビザンチンな行動の際に必要な、ルートチェーン上の状態の更新を最小限に抑えること、また状態の変更を強制するためのメカニズムを必要とする。

ライトニングネットワークと同様に、プラズマは、後日のネット決済/引出とともに、コントラクトの状態の中の、資金2を保持することができることを保証しながら執行を確保するための、既存のブロックチェーン上で動作する、一つの契約のシリーズである。

2 Plasma

プラズマは、コントラクト作成者がアクティブな状態遷移管理をせずに、自律的かつ持続的にチェーンを運営する経済的インセンティブを作成するための構造をもった、ブロックチェーンでスケーラブルな計算を行う方法である。

ノード自体はチェーンを運営するようにインセンティブ化されている。

さらに、ビットマップ内の単一のビットとのコントラクトからの支出として表される資金を最小限に抑えることによって、重要なスケーラビリティが達成されます、そのため、

1つのトランザクションと署名が、多くの参加者と合体した支払いを表す。

私たちは、結合したスマートコントラクトによって強制された、スケーラブルな計算を構築することができるようにするために、MapReduce [5]のフレームワークとこれを組み合わせる。

この構造により、外部化された当事者は、マイナーに似て、資金を保持して契約を計算することができるが、統合された状態の更新のための最小限のデータをオンチェーンに持ちながら、Plasmaは既存のブロックチェーンの上で実行されるため、すべての状態更新のために基になるチェーンにトランザクションを作成する必要はない。（新規ユーザーの元帳エントリの追加を含む）

図1　誰もがさまざまなユースケースのためのスマート契約のスケーラビリティのためのカスタムPlasmaチェーンを作成することができる

プラズマは、ルートチェーン内に多くのブロックチェーンをおくことを可能にしたスマートコントラクトのシリーズである。

ルートチェーンはプラズマチェーンの状態を強制する。

ルートチェーンは、世界的にすべての計算の執行者であるが、詐欺の証拠がある場合にのみ、計算され、ペナルティが課される。

多くのPlasmaブロックチェーンは、独自のビジネスロジックとスマートコントラクトの条項と共存することができる。

イーサリアムでは、プラズマはイーサリアムで直接実行されるEVMスマートコントラクトで構成されるが、nonByzantineの場合には、非常に大量の計算と財務会計記録を表すことができる小さなコミットメントしか処理しない。

プラズマは、5つの主要コンポーネントで構成される。

経済的に効率的な方法で契約を永続的に計算するためのインセンティブ層、低コストの効率とトランザクションのネット決済を最大限にするための、ツリー形式で子チェーンを配置するための構造、これらのネストされたチェーン内での状態遷移の詐欺証明を構築してツリー構造と互換性があり、状態遷移を高度にスケーラブルに再構成するためのMapReduceコンピューティングフレームワーク Nakamoto [6]コンセンサスインセンティブの結果を再現しようとするルートブロックチェインに依存するコンセンサスメカニズムルートブロックチェーンからの正確な状態遷移を確実にし、massexitコストを最小限に抑えるためのビットマップ　UTXOコミットメント構造。

データの利用不可能性やその他のビザンチンな行動の終了を可能にすることは、Plasmaの運用における重要な設計ポイントの1つである。

2.1　Plasmaブロックチェーン、または外部化された複数のチャンネル

我々は、マルチパーティーオフチェーンチャネルが他者のために状態を保持できる方法を提案する。

私たちはこのフレームワークをPlasmaブロックチェーンと名付ける。

Plasmaチェーンで保有されている資金については、これにより、Plasmaチェーンへの資金の預金および払戻しが可能になり、詐欺の証拠によって状態の移行が強制される。

これは、ルートチェーンに保持されている資金と一致するPlasmaブロックを考慮して、預金と出金が可能であるため、強制的な状態と代替性を可能にする（Plasmaはフラクショナルリザーブバンク設計に適合するようには設計されていない)。

図2　Plasmaブロックチェーンは、ブロックチェーン内のチェーンである

このシステムは、詐欺の拘束証明によって強制される。

Plasmaブロックチェーンは、ルートチェーン上（例、イーサリアム）のブロックチェーンの内容物を明らかにしていない。

その代わりに、ブロックヘッダハッシュがルートチェーン上に提出され、ルートチェインに詐欺が提出されたことが証明された場合、ブロックはロールバックされ、ブロック作成者にペナルティが課せられる。

これは非常に効率的で、多くの状態の更新は、単一のハッシュ（いくつかの小さな関連データ）で表される。

この更新は、ルートブロックチェーンには表示されない残高を表すことができる（アリスはルートチェーン上の元帳残高を持たず、元帳はPlasmaチェーン上にあり、ルートチェーン内の残高はPlasmaチェーン自体を強制するスマートコントラクトを表している。）

灰色のアイテムは古いブロックであり、黒はルートチェーン上で伝播されコミットされた最新のブロックである。

信じられないほど大量のトランザクションがこのPlasmaチェーンでコミットされ、ルートブロックチェーンに入るデータは最小限に抑えられる。

参加者は、既存の参加者ではない参加者への転送を含む、誰にでも資金を移転すること

ができる。

これらの転送は、ルートブロックチェーンの本来のコイン/トークンに資金を払い込んで（多少の時間遅延と証拠で）資金を引き出すことができる。

Plasmaでは、ルートブロックチェーン上の元帳の完全な永続的な記録なしに、第三者に保管上の信頼を与えることなく、誰か（または、Proof of Stakeの参加者のネットワーク）がブロックチェーンを管理できるようになる。

最悪の場合、資金はロックされ、timevalueはブロックチェーン上のmassexitsによって失われる。

私たちは、このチャンネルで状態を強制するルートブロックチェーン上のスマートコントラクト[7]として一連の不正防止プルーフを構築し、不正行為または非ビザンチン行為の試みを削ることができる。

これらの詐欺証明は、資金出金のインタラクティブプロトコルを強制する。

ライトニングネットワークと同様に、資金を引き出す際には、Exitには時間が必要である。

私たちは、出席者が出金を要求するUTXOモデルに配置された、参加者の元帳OUTPUTのビットマップを証明する、インタラクティブゲームを構築する。

ネットワーク上の誰でも、既に資金が使われているかどうかを証明する代替*bonded*証明書を提出することができる。

これが間違っている場合、ネットワーク上の誰も不正行為を証明し、認証をロールバックするためにbondsを破棄することができます。

十分な時間が経過すると、2番目の*bonded*ラウンドでは、コミットされたタイムスタンプの*前*の状態のbondである引き出しを行うことができる。

これは、欠陥のあるPlasmaチェーンが急速に終了するように、一括して取り出すことを可能にする。

調整された大量出金イベントでは、参加者は、親ブロックチェーンで消費されるブロックスペースが2ビット未満で終了することがある（すなわち、ワーストケースのシナリオではオンチェーンのイーサリアム）

Block withholding攻撃が発生した場合、参加者は迅速かつ安価に大量退出を行うことができ、これまでのオフチェーンの提案と比較して大幅なコスト削減が可能である。

さらに、これは承認ノードの連合（Sidechain Functionaries、Fishermen）に信頼を置いていない。

赤いブロック（ブロック＃4）は、ルートチェーン上で保留されコミットされたブロックであるが、アリスはプラズマブロック＃4を取得できない。

図3　Block wtihholdingの場合の資金の出金

彼女はルートチェーンに資金の証拠を伝播することによってExitし、出金は紛争を許容するために遅れて処理される。

ライトニングネットワークを閉じる方法と同様に、2人の参加者間で無制限の支払いを可能にするためのインタラクティブなメカニズムであり、これにより、n人の参加者間のインタラクティブなメカニズムが可能になる。

主な違いは、すべての参加者が状態を更新するためにオンラインになる必要はなく、参加者は参加を可能にするためにルートチェーンにエントリの記録を必要としないことである。

これらのPlasmaチェーンを、ツリー形式で構築する際に取引を確認するための最小限のデータで、オン・チェーンで直接的な相互作用なしにPlasmaに資金を置くことができる。

ブロックチェーンの強制されたブロックチェーン

図4　Plasmaはツリー状にブロックチェーンを構成する

ブロックコミットメントはフローダウンし、終了は任意の親チェーンに提出でき、最終的にルートブロックチェーンにコミットされる。

裁判制度と同様の仕組みを構築する。

ライトニングネットワークが最終的にルートチェーン上で執行可能な支払いのために裁定層を使用する場合、nonByzantineの状態では可用性を最大限に高め、コストを最小限に抑えるために、裁判所の上層階と下層階のシステムを作成します。

チェーンがByzanitneの場合は、その親（ルートブロックチェーンを含む）のいずれかに行き、現在のコミットされた状態で実行を続行するか終了するかを選択できる。

Inctementingnonce状態（取り消しを介して）を実行する代わりに、これらの連鎖階層の残高と状態遷移を強制する詐欺証明のシステムを構築する。

実際には、親チェーンに定期的にコミットされるだけの状態遷移を作成することができる。（ルートブロックチェーンに流れる）。

これにより、ビザンチン条件で親（またはルート）チェーンにraw dataを送信すること

しかできないため、計算量とアカウント状態の驚異的なスケールが可能になる。

ビザンチンの部分的な状態からの復旧は、親プラズマ・チェーンに行って状態を強制することができるため、コストを最小限に抑える。

この子ブロックチェーンはルートブロックチェーン（Ethereumなど）の上で実行され、ルートブロックチェーンの観点からは、ブロックチェーンのProof of Stakeコンセンサスルールとビジネスロジックの実施のために契約でトークンが結合された定期的なコミットメントのみが表示される。

これは、ブロックの可用性を最大化し、コインの承認のためのStakeを最小限に抑える上で重要な利点を有する。

ただし、すべてのデータがすべての関係者に伝播されるわけではないため（特定の状態を検証したい人のみ)、当事者は、彼らが定期的に興味を持っている特定のチェーンを監視し、詐欺行為を罰する責任があり、同様にBlock withholding攻撃の場合には、個人的にチェーンを急速に脱出する。

図5　障害のあるブロックチェーン（赤で塗りつぶされている）は、その親のPlasmaおよびルートチェーン（右の点線の青い線）にコミットメントをブロードキャストすることでルーティングができる

第3階層目のPlasmaチェーンの参加者は、一定期間後に別のチェーンに一緒に移動します（青色の点線)。

この構造は、非ビザンチン環境では、ブロックチェーンの状態のツリーを結合し、すべてのPlasmaチェーンを更新する。

すべてのチェーンにわたる更新セット全体は、署名付きの32バイトハッシュで証明されます。

2.2　Plasma ProofofStake

3　デザインスタックとスマートコントラクト

歴史的に、多くの人々はブロックチェーンが決済システムとしてトランザクションの支払いに最も適したものであると信じている。

しかし、決済システム全体のスケールが難しいことも判明している。

Lightning Network、payment channel networkのような現時点での解決策はブロック

チェーンの構造を変更し参加者間でほぼ無限の取引を実現する。

ブロックチェーン上でチャネルがスケール問題を解決することでトランザクションの容量は劇的に増加する。

決済はそれらのチェーンを通して送られるのである。

この構造は、さらに効果的に瞬時の支払いが可能にする。

これは、瞬時に送金したい決済に有効なだけではなく、コントラクトにも有効である。

Plasmaは、トランザクションが子チェーンでファイナリティが迅速に決定するけれど、それ自体が、ファイナリティの迅速な確保の為に作られているわけではない。Plasmaはルートブロックチェーンでのファイナリティを確保する。

チャネルはローカルで迅速に決済とコントラクト(enforcible onchain)のファイナリティを確保することができる必要がある。

スマートコントラクトでは、”free option problem” の問題がある。スマートコントラクトの受信者（2番目もしくは最後）がコントラクトを実行するためにそれに署名し拡散する必要がある。ーその間に、もしそのコントラクトが受信者によって関心のあるものでなければ、受信者はコントラクトをfree optionとして扱い、署名を拒否することができる可能性がある。

信頼されていない相手と取引をするときにスマートコントラクトが最も効果的であるとして、これは悪化する。

Plasmaはこの問題を解決しない。ブロックチェーンのインタラクティブなプロトコルの最初と2番目の署名ステップに保証人がいないからだ。

Lightning(Plasma上のLightningも含む）は許容範囲内のローカライズされたファイナリティによって劇的に迅速なアップデートを可能にする。

最後の参加者に任意で与えるsingle paymentの代わりに、決済情報は多くの小さな決済情報に分割され渡される。

これは分割された支払いにおけるfree optionを最小化する。

スマートコントラクトの第二者のみが、分割された支払いのfree optionを持つので、free optionの価値自体が最小化される。

Plasmaは、最小のルートチェーンのState commitmentsによるアップデートができるので、上記のユースケースのように、LightningはPlasma上の迅速なフィナンシャル決済/取引の主要なインターフェイスレイヤーになるかもしれない。

コントラクトそれ自体はルートブロックチェーンに存在する。

Plasmaチェーンはルートブロックチェーンの変化を反映した現在のstateを含む。

Fraud proofsは資金が補填される為に存在する。

図6　コントラクトと決済を判定するレイヤー、ルートブロックチェーンについて

Plasmaはブロックチェーン上で最小トランザクション数のスケールする方法で資金を引きざす為に作られた下層Plasmaチェーンを代表する

Lightning Networkの上層レイヤーで瞬時の決済がPlasmaとブロックチェーン間で行われる。

3.1　最も重要なShardingの問題は情報である

分割されたデータセットには、各々のShardで情報を公開することを拒否する重大なリスクがある。

それによって、fraud proofsを生成することが不可能である。

私達は3つの戦略でこの問題を解こうとしています。

1. ブロックの伝播を促進する新しいProofofStakeのメカニズム。根底にあるメカニズムは現在のインセンティブの機能性に完全に依拠していない。これは誤った行動を劇的に減少させる。
2. Significant withdrawal delays which allow for accurate withdrawal proofs. Individuals don’t need to watch the blockchain that frequently and fraud on higher plasma chains can be prevented on the root blockchain by any honest actor on the same plasma chain as the user. ブロックを隠し持っている場合には、Plasmaチェーンは即座に資金を凍結し、攻撃者が不正な証拠を提出するのを妨げる。固定されている資金よりも多くの額を攻撃者が引き出そうとした際には、攻撃しているplasmaチェーンはデポジットを失う。
3. どの親チェーンにもトランザクションを伝播できる子チェーンを作る。これによって、ネットワークの参加者はトランザクションを深い子チェーンへ伝播したいと思うようになる。これは、高い手数料を払うことのできないルートブロックチェーン上の少額の取引に経済効率性を創り出す。従って、流動資金は少額となる。したがって、参加者は多くの価値を代表する子チェーンを作ることを促進される。注意しなければ行けないのは、とても少量の額をもっている個人にとっていくつかのチェーン選択に関する前提が存在しているということで、

このセキュリティモデルがPlasmaチェーンの鍵である。

4　関連プロジェクト

幾つかの関連プロジェクトはproof of computation用に、簡略化されたメルカトルツリーを提案している。しかし、この提案は主に、経済的にインセンティブ設計された持続的なシャードチェーングループを経由したプロトコルによりデータの可用性と不正検知のコストを最小にすることである。

他の関連プロジェクトは子ブロックチェーンのシステムを提案した。しかし、アプローチ方法に大きな乖離があった。Plasmaは子チェーンを履行する為にmerkleized proof を使っているのだ。

4.1　TrueBit

PlasmaはTrueBitの不正検知システムと似た類似性を持っている。

不正検知の作成方法はTrueBitと似ている。そして、TrueBitによるほぼ全ての活動、特にstate transitionsのmerkleized proofsなどはPlasmaに直接適応される。

TrueBitのデザインはイーサリアムブロックチェーンにcompact proofsが送られるように設計されている。これはPlasmaにとって必要なことである。TrueBitの提案書とそのチームによる手間がかかる作業のほぼ全てはこの設計に直接当てはめることができる。

マークル木を用いた証明が生成されるこの認証ゲームの使用は、計算量の増加を減らすという点でメリットがある。

TrueBitの適応と似た想定、特に、computational stateは計算可能かつオンラインで拡散可能でなければならない。（大きなデータが複数の小さなデータに分割された状態）データの可用性の問題は緩和される必要があり、通信の失敗は公開されなければならない。

私達はこれらの問題、特に後の2つの問題を緩和しようとしている。

PlasmaがTrueBit上に構築している主な要素は、共有されたstateを計算する必要のある複数の参加者の概念である。例えば、幾つかの参加者が特定のデータや計算だけを注目し、関連したデータのみを計算するという場合がある。（例：BBS or exchange)

私達はまた、オフチェーンを経由して計算を実行することでオンチェーンでの計算量の問題を緩和するこころみを行っている。

4.2　ブロックチェーンのSharding

ブロックチェーン上でシャーディングをする現在の試みの多くは、似た技術を用い、似た目的を持っている。例：Ethereum Sharding

これは上位層レイヤーで互換性があるかもしれない。

もし、ルートブロックチェーンがシャーディングされていれば、Plasmaチェーンはルートチェーン上でスケーラビリティーを持ち、他の恩恵を受ける。

これはまた、イーサリアムや他の安定しているブロックチェーンで基本的な作業を行う為に、コンセンサスの変更が必要なしに異なるシャーディング技術がテスト可能である。

4.3　Federatedなサイドチェーン

結合しているプラズマサイドチェーンは正直な一連の活動や、チェーン上でstateを履行する完全に信頼された参加者に依存しないので、単体のサイドチェーンではない

Plasmaはまた他のブロクチェーンで同じコインやトークンを使用することを可能にする。しかし、それはもし、fraud proof が利用可能なら履行可能な立証である。

Plasmaは参加者の強力な強力に依存していない。それらの参加者の正しさには非常に大きな潜在的なリスクがある。そのことによってPlasmaは結合された接続口のあるサイドチェーンではない。

Drivechainsはバリデーターが匿名かつランダムでありより分散化されている点を除き結合されたサイドチェーンとにている。

4.4　マージマイニングされるブロックチェーン

例としては親ブロックチェーン[14]と共存するブロックを作成するNamecoinがある

これは従って、ブロックチェーンのフルバリデーションはスケーラビリティー問題に解決策を提示しないと想定されている。

拡張されたブロックは主要なブロックチェーンとmergemined chain(with an enforcement mechanism of the full set of miners as a consensus rule on the root chain)の間で資金のやり取りをするmergemined chainの一例である。

Mergemined chainsは、新しいコンセンサスのルールと所属しているチェーン（だけ）の正当性を確認するユーザーの選出するが、マイナーと検証者は全てを検証しないといけない。

Plasmaのゴールは、ユーザーとマイナーだけが彼ら自身が関与しているチェーンを検証することができることを確立することである。

4.5 Treechains

Treechainsは子ブロックをProof of Workを使って検証できるようなツリー構造のブロックチェーンを提示している。

ルートチェーンは全ての子ブロックチェーンのproof of workを要約したものである。

その積み重ねの下層ほどセキュリティーが強くなる。しかし、上層はバリーションとかけた計算コストによってセキュリティーのレベルが決まるのでセキュリティーが強いこともあれば弱いこともある。

ツリーチェーンのトポロジーは複数の階層に分かれているけれど、その構造はマイニングセキュリティーの総和に依存している。

セキュリティモデルはProof of Workによって機能しているが、「葉」の部分では弱くなる。Plasmaはルートチェーンでのみ完全なセキュリティのマイニングが行われる。なので、セキュリティと検証証明はルートチェーンに依拠している。

同様の活動が、tree formationのブロックの正当性を証明する作業にも見受けられる。

4.6 zkSNARKsとzkSTARKs

証明者と承認者がリアルタイムで互いにやり取りをすることのない証明は処理性能の向上に多大な恩恵がある。

zkSNARKs/STARKs と証明者と承認者がリアルタイムで互いにやり取りをすることのないコンパクトな証明はPlasmaにメリットがある。

証明が、マークル木の計算結果に伴って提供される。それに加えて、子Plasmaチェーンで少額の量を保有している際に、systemic attacksの可能性を減らす恩恵もある。MapReduce functionality用にSNARKsは既に研究が行われている。[17] そして私達はこれが研究を加速させ、Plasmaがいくつかのブロックチェーン上で順序付け可能で履行可能な証明を作ることによってこの機能を拡張させることを期待している。

さらに、より早く同期とチェーン自体の検証作業を行うことで、計算量の証明にも恩恵がある。

zkSNARKsはデータの可用性の問題をとかないことは注意しないといけない。計算と必要なデータ量を減らすだけである。

これは特に、実時間性に問題を抱える仕組みを置き換えたり補完するのに有効である。

zkSNARKsは高度な安全性を確立するのに有益である。セキュリティの最終レイヤーが好みの暗号を使用していないブロックチェーンでは、セカンドレイヤーにはzk-SNARKsが使える。そしてファーストレイヤーには信頼されたハードウェアがあればよい。

Plasmaチェーンからの引き出しは少額であっても、bitmapを必要としないメリットがあるzkSNARKsによって守られる。

4.7 Cosmos/Tendermint

Cosmos[18] はCosmos “Hub”でブロックチェーンが繋がり、Pfoor of Stateで正当性が決まる子ブロックチェーンである”Zones”を持っている。子ブロックチェーンの構造と類似点が見受けられる。しかし、Plasmaは子チェーンでstateを履行する為のfraud proofsに依拠しており、多くのチェーンに適応することができる。CosmosのPfoof of Stake構造はCosmos Zoneの検証者を含め2/3が正直な検証者であるという想定でなりたっている。

4.8 Polkadot

Polkadot[19] もまたブロックチェーンの階層構造を形成している。これは、Polkadotのデザインと幾つかの類似点がある。”fishermen”（Polkadotのwhitepaper参照）のバリデーターがブロックの正当性を保証する構造の代わりに、マークル証明を経由してstateを履行する子ブロックチェーンを複数構成する。Polkadotの構成は子ブロックチェーン（”parachains”）のstateとfishermenによって履行される情報の可用性に依存している。

4.9 Lumino

Lumino[20]はブロックチェーンにアップデートするEVMコントラクト用に設計されたものである。これは参加者が最小限のコミットstateを更新するだけでよくなっている。Plasmaのアウトプットマネジメント設計はtakes things further with only a single bit denoting a particular output. これは迅速でコスト安な子Plasmaチェーン上での引き出しを可能にする。

5 Multiparty OffChain State

overview

目標は、参加者がそのサイドチェーンを下支えするネイティブのブロックチェーンの通貨やトークンの資金を保持したまま、多岐にわたるオンチェーンステートを用いることのない手法を開発することだ。Plasmaはオンチェーンとオフチェーンの間の境界線を曖昧にすることからはじまる（例：シャーディングにおけるshardたちはオンチェーン？オフチェーン?)。オフチェーンで複数の管理者が存在するチャネルを作成しようとする際、主に2つの問題がある。ひとつは、その系（オフチェーン）をアップデートする際に、全ての参加者の間で状態を同期しなければいけないこと（でなければ、グローバルな状態更新をできる能力をトレードオフにすることになる）で、したがって参加者は全てオンラインであることが求められる。二つ目は、参加者をチャネルに追加・削除することが、大規模なオンチェーンでの状態更新を要求することや、追加・削除された参加者を列挙することだ。それよりむしろ、大規模なルートチェーンでの状態更新なく多くの参加者の追加・削除ができ、全ての参加者が参加することない(参加者の残高更新かビザンチン障害が検出されたときだけ参加する必要がある）内部的な状態更新が可能であるようなメカニズムを作ることが好まれるだろう。一般的な解法としては、ルートチェイン（例：Ethereum）のスマートコントラクトに存在する保有残高を許容する子チェーンになる。その

スマートコントラクトの残高は、子Plasmaチェーンでまとめられたブロックたちからのアロケーションで表される。これはユーザーにルートチェーンの残高と完全に等価なトークンを子チェーンで保有することを可能にし、一定の紛争調停期間ののちに引き出すことを可能にする。これを達成するために、我々はUTXO(Unspent Transaction Output)モデルを台帳として導入する。これは明確な要件ではないが、高性能な残高引き出しを実現するのをたやすくするだけの理由付けがある。UTXOモデルを用いることの理論的裏付けは、コンパクトにその状態（例：残高）がすでに使用されているかどうかを表せるからだ。これはしばしばマークル化された証明としてのトライ木（マークル木）の中で表現され、第三者によってparseできるコンパクトな表現としてのビットマップとしても表される。言い換えると、スマートコントラクトはルートチェーン中のアカウントに保存されているが、Plasmaチェーンはルートチェーンアカウントに保持されている残高の割当に対してUTXOの残高セットを保持する。状態遷移に関して大した要件を持たない子チェーンは、より複雑で頻繁な状態遷移を用いる「アカウントモデル」を使用することも可能だが、親チェーンたちのblock spaceの可用性により依存することになる。

差し当たって、ユーザーは子Plasmaチェーンのブロックを選ぶ単一のブロックのリーダーを想定することができる。これをProof of Stake集合や、M人のうちのN人のバリデーターとして解釈することはできる。しかしこれらの例では我々は

単一の命名をされたバリデーターを簡単のために用いている。バリデーターの役割はトランザクションを搬送する役割に従事するブロックを提案することである。そのバリデーター（提案者）はルートチェーンのコントラクトに作成された不正防止機構によって制限されている。もしそのバリデーターが不正な状態遷移を含むブロックを伝播した場合、他のブロックを受け取った参加者は親チェーン上のマークル化された不正証明を提出でき、不正なブロックはペナルティとともにロールバックされる。そのブロックはブロックを監視したい参加者（残高を持つ参加者や、個別のPlasmaチェーン上での計算処理を観測/強制したい参加者を含む）へ伝播される。オフチェーンの状態に関するデポジットを維持するのには最小限の複雑さで済む一方で、オフチェーン状態遷移と引き出しはさらなる複雑さを生む。

5.1 Fraud Proofs

子チェーンに含まれる全ての状態は、どのような参加者にも不正ブロックを規制することを可能にする不正証明を通して強制され、ブロックデータの可用性を期待できる。しかし、これらを作成する上での最大の難関は、データとブロックの可用性に明確な保証がないことである。ルートチェーン（例：Ethereum）において、ブロックデータが利用可能なときに全ての状態遷移が正しいことを確かめる不正証明のリストがある。複雑な計算処理のために、状態遷移は効率的な証明のためのマークル化をされていなければならない。加えて、

状態遷移は不適切な子チェーンからの離脱が不可能であることを確かめるzkSNARKs/STARKsによって強制されたものにもなる。zkSNARKsの作成は最大限の効率のために再帰的なSNARKsを必要とし、したがってさらなる可能性の研究を必要とするだろう。しかし、システムはSNARKsなしでも稼働するように設計されている。

図7　block1 4の全員がブロックデータを持っている。Block4がコミットした状態のトランザクションはおそらくマークル化されたBlock4とそれ以前のブロックのデータのコミットメントによると不正である

不正証明は全ての状態遷移は検証されていることを確かめる。不正証明として挙げられるのは、トランザクションの利用可能性の証明（資金は現在のUTXOの中で利用可能）と、状態遷移の証明（アウトプットが利用されうる能力の署名のチェック含む）と、ブロックを横断した包含/排他の証明と、デポジット/引き出しの証明だ。その他のより複雑な証明はインタラクティブな駆け引きを必要とする。一般的な解釈はブロックの証明に機能的な（関数型の）アプローチをとることだ。

もしあるユーザーがこのコンセンサスメカニズムをSolidityでプログラムしたとして、関数ごとにブロックのマークル証明が引数としてひとつ追加されるだろうし、アウトプットは検証が正当であるかを返すだろう。(TODO)ユーザーはそれから、コンパクトなマークル化された証明（ユーザーは不正証明を生成するためにブロック全体を処理する必要がない）の中で処理をするために、単純にコンセンサス

の検証のソースコードを複製する。

図8　Aliceは全てのブロックデータのコピーを持っており、彼女は不正証明をルートチェーンに提出する。Block4は無効化され、ロールバックされる。Block4の提出者はスマートコントラクト内に保持された供託金(bond,deposit)を失うことでペナルティを受ける。現在のブロックはBlock3(図中青)になる。ある程度時間が経ったあと、ブロックは確定(finalize)され、不正証明はもう提出されない。ユーザーは完全に検証されたブロックたちから不正だと証明されえないブロックの上にしかブロックを生成するべきではない

この解決法に最小限の証明を持たせるために、けれども、全てのブロックはマークル化された現在の状態の木と、使用されたアウトプットの木と、トランザクションの木と、前の状態への参照のコミットメントを提供しなければならない。不正証明は参加者の連合がペナルティなしに不正なブロックを作ることができないことを確かめなければならない。不正なブロックが検出されルートチェーン（もしくは親のPlasmaチェーン）で証明されたイベントにおいて、不正なブロくはロールバックされる。これは個々の参加者がビザンチン障害的な振る舞いに対抗するインセンティブを持たせ、bitcoinサイドチェーンのfederated pegのような状態遷移の脆弱性を解決できる。その結果、スケーラブルな状態遷移がPlasmaチェーン内で可能になる一方で、ブロックデータにアクセス権を持つ監視者たちは不正な状態遷移を証明する（そしてそれゆえ抑制する）ことができるようになる。言い換えれば、支払いはルートチェーンでの周期的なコミットとともにこのチェーンでおきる。

5.2　デポジット

ルートチェーンからのデポジットは直接マスターコントラクトに送られる。そのコントラクトは現在の状態のコミットの追跡と、不正なコミットに対しての不正証明を用いたペナルティと、資金の引き出しの処理に対して責任がある。ルートチェーンの完全なバリデーターである子Plasmaチェーンとして、流入するトランザクションは2フェーズロックインを用いて処理されねばならない。

デポジットは目的地となるチェーンのblockhashをチェーンを特定するために含んでいなければならず、そのデポジットは多段処理を用いてその資本が回復不可能ではないことを確認する。

図9　AliceはETHアカウントに1ETHを持っている。彼女はそれをPlasmaチェーンに送りたい。彼女はPlasmaコントラクトにそれを送る

1. そのコイン/トークン(例：ETHやERC20トークン)はルートチェーンのPlasmaコントラクトに送信される。そのコインは一定期間はchallengeやレスポンスを用いて回復可能である。
2. Plasmaチェーンは流入トランザクションの証明を含んでいなければならない。この時点でPlasmaチェーンはトランザクションが流入していて、トランザクションのロックインかデポジットした人の使用のイベントによって利

用可能になる事実をコミットしている。これが取り込まれた時、そのチェーンは引き出しリクエストを受け取るという事実に対してコミットしている。しかし、デポジットした人が不正証明を生成するために十分な情報を持っている確認がまだないために、デポジットした人からのコミットはまだ存在しないことになる。このブロックは状態ツリーと、ビットマップと、トランザクションツリーの追加を取り込み、コンパクトな正しい取り込みの証明が得られる。

3. デポジットした人は子Plasmaチェーンにトランザクションに署名をし、トランザクションをアクティベーションし、それはフェーズ2におけるチェーンのコミットメントと共に彼らがブロックを観測したというコミットメントを取り込む。このプロセスののち、チェーンは彼らがこれらのコインを処理し、アロケーションを与えるという事実に対してコミットしたことになり、したがって引き出しはコンパクトに証明されうる。フェーズ3によってユーザーは彼らは引き出し可能であるということを署名している。

図10　AliceはいまETHをPlasmaブロックに持っている。彼女は彼女が資金を観測し、それが今ロックインされたことをコミットした。その資金はルートチェーンのスマートコントラクトの中に保持されており、しかしこのPlasmaチェーンにある台帳の記録（したがって状態遷移、すなわち資金を他者かスマートコントラクトに送ること）は、重大なルートチェーンでの消費なく発生する

デポジットした人がPhase3を通過していないという状態では、この人はルートチェーンでの引き出しを試みることができることができる。デポジットをした人は未確認の引き出しリクエストを提出し、さらに長い期間ネットワーク上の誰かがデポジットした人が署名して資金をPlasmaチェーンにロックインしたという不正証明を出してくれるのを待たねばならない。もし証明がなければ、デポジットした人はその未確認の資金を引き出すかもしれない。この引き出しは大きなルートチェインでの供託金が必要で、ビザンチン障害がないことを確認しなければならない。

5.3 "bitmap化された状態"とmass withdrawal(大量引き出し)

この系についての初めの関心は状態を検証することへの不適格性だ。状態トランザクションの最大の圧縮を提供できるようにするために、アウトプットは選択的にビットマップで表現されうる。これは引き出しの照明に必要でルートチェーンに伝えるにはとても高価になりうる。この構造の目標は少額の資本をPlasmaチェーンに持たせることだ。これらの残高は完全な予約をルートチェーンのコントラクトの中に保持され、しかし完全な台帳はルートチェーンの中にはない。初歩的の予期されるべき攻撃は、不正なブロックの保持だ（ルートチェーンへのコミットメントも含める)。このイベントの中でシステムは不正な状態遷移を観測し、参加者はたくさんのイグジットをトランザクションとして生成する。ビットマップの構造とともに、引き出しは署名されたexitを希望するトラ

ンザクションのビットマップを含む。スマートコントラクトにより情報の正しさを確認することを強制される、ゲームあるいはプロトコルが後構築される。ビットマップはみんながアウトプットが使われたかどうかの理由づけができるかを確認する。ビットマップとしては、UTXOとして表現された状態を、少額残高の最大限の効能のために必要とする。利用痕跡はコンパクトに証明され得る、そして状態遷移の大きな集合はきれいに強制される。事前に定義された送金時間の後でそのビットは再利用される。高い支払いには高い精度、安い支払いには低い精度と言う勾配がある。

1. ルートチェーンの上での台帳の状態
2. オンチェーンでの単一のトランザクションの経済的に強制可能な、プラズマの上での台帳の状態
3. ビットマップを用いた経済的に強制可能なプラズマ上での台帳の状態
4. ビットマップをルートチェーン上で用いた、経済的に強制可能ではにプラズマ上での台帳の状態。1から2ビットによる大規模引き出しはコストがとても高い。それらの保持されているルートチェーンの上で強制できる残高のために、ビットマップ化されたフォーマットのUTXOを作る要件は必要ない。しかし、もしその12ビットのトランザクションのガスがルートチェーンでとても低かったときだけそれらの保持されている残高と強制は可能である。

4番目のタイプ（12ビットのケースのオンチェーンコストがとても高くて、大規模引き出しが起こる）については、そのシステムはまだレジリエントであるように設計されている。しかし、いくらかの名前付きエンティティーが信頼できるだろうという推測は含んでいる)。ヒエラルキー的なブロックチェーンの構造をたくさん作るケースでユーザーは経済的に大規模引き出しが可能であることはこの論文の後のセクションで詳しく説明する。加えて、もしトランザクションの総額が4番目のカテゴリでトークン総額よりもだいぶ低い場合、それはゲームを原理的にそれらの残高をトークンホルダーが評判へのダメージで苦しむように攻撃するにはとても高額なものにする。

5.4　状態遷移

もともとPlasmaチェーンでの状態遷移は同じようなマルチフェーズのプロセスでデポジットとして実行される。これはユーザが状態遷移を提供すことができる情報を確認するためだ。しかしデポジット構築とは異なるように、一度トランザクションが署名されとブロックに組み込まれると、参加するコミットメントがある。この理由のために状態遷移は署名や、状態アップデート（例：目的地、残高の量、トークンの種類、そして他の種類の関連する状態データ）や、そして同様に何らかの種類の失効を意味するTTL、そして特定のブロックのコミットメントを含むべきだ。このTTLは必要とされずとも、副次的なexit条件が知られていることのexit証明の構

築時間より短くあるべきだ。事前に署名されたトランザクションはもちろんTTLを含むべきではない。(TODO)弱い活動性の推測ではそこにすでに活動性の推測が引き出しとと共にあるような構造、大きいreorgとともに認識される。ブロックへのコミットメントは、エンティティーがトランザクションをPlasmaチェーンにブロードキャストする残高利用者によるコミットメントで、それはその時点までチェーンを観測し、証明を矯正でき、使用されたアウトプットが生じたブロックまで存在しなければならない。マルチフェーズコミットは以下の高速なfinalizationにしたがって生じる。

1. AliceはPlasmaチェーンの中にいてUTXOを同じPlasmaチェーンの中にいるBobのために使いたい（ブロックチェーンに実際の完全なトランザクションレコードを発行することはしたくない)。Aliceはひとつの UTXOを消費するトランザクションをPlasmaチェーンに作成し、署名し、トランザクションをブロードキャストする。
2. トランザクションはプラズマチェーンのバリデーターによってブロック内に含まれている。ヘッダーは親のプラズマチェーンのルートチェーンのブロックの中に含まれている。究極的にはルートブロックチェーンにしまわれることになる。
3. アリスとボブはトランザクションを観察する。そしてトランザクションとブロックを承認したということを署名する。この承認と署名は別のプラズマチェーンに保存さ

れる。ファイナリティを得るのに時間がかかる場合は、ステップ1だけが起こる必要がある。

承認のあと、トランザクションはファイナリティを得たと考えられる。ステップ3が必要な理由は、AliceとBobがブロックが利用可能かどうか確認できるようにするためだ。ステップ3はなくともシステムは機能するが、ファイナリティを得るのに時間がかかるだろう。根拠としては、そのブロックのすべての参加者から正当さと情報の利用可能性がトランザクションに対して十分に証明されるまで、トランザクションはファイナライズされたと見られるべきではないからだ。

ステップ1ではアリスはまだ彼女がUTXOを使ったのかどうか定かではない。もしトランザクションが既にブロックに取り込まれていて、ステップ3が終わっていなかったら、そのトランザクションは未確認として扱われる。したがってアリスは、もし彼女がルートと親のチェーンにファイナライズされる前に引き出しメッセージに署名をしていなかったら、それらの資金をまだ引き出すことができる。ファイナライズされた後にはAliceは引き出すことはできず、Bobに送金されたものと見なされる。もしブロックがファイナライズの前（ステップ1と2の間）に保留されるとしたら、アリスもしくはボブがそれを観測したときその場合アリスはまだファイナライズドされていない資金を引きだすだろう。もしステップ2とステップ3の間にブロックが保留されるとしたら、Bobは引き出すのに資金を引き出す十分な情報持っていると推測さ

れる。しかしAliceもBobも完全にペイメント支払いをコミットしないために、それは完了したとみなされない。それはその参加者が理論的に資金を主張することができるかどうかという情報の利用可能性に依存する。もし双方がステップ3に署名した場合、完全にファイナライズされたと見なされる。Paytocontracthash執行はこのステップが完了したあとに生じる、特に署名がオンチェーンで確かに確認されたあとに生じる。一方が署名やブロックの保留を拒んだ際には、リデンプションプルーフ（償還証明）に依存する。全ての状態が結果的にマークル証明を介してコミットされると、支払いとしてのPaytocontracthashが確かにファイナライズのあとに施行可能になるのは信頼性が低い。

ステップ3はスマートコントラクトに対して選択的なのを注意されたい、両者の署名ではない、他には、状態はHTLCプレイメージのリリースに対しても選択的である。これは複数チェーンや複数トランザクションのアトミック性を実現する。コントラクト作成の複雑さは増加し、これらの機能が必要ならばより高レベルな言語やツールが必要になるだろう。

5.5　ルートチェーンへの周期的なコミット

プラズマチェーンはブロックチェーンの順番を作成できなくてはならない。プラズマチェーンではブロックの順序があり、それら自身では証明されも整列されもしていない。結果的に、ルートブロックチェーンにコミットメントを作る必要がある。プラズマチェーンはそれ自身のブロックヘッダを他

者のための情報の利用可能性についてのデータと共にルートチェーンに作成する。他のどんな参加者でも不正証明を発行でき、コミットメントとブロックは発行者を罰しつつロールバックする。これらのコミットはのちの不明確さのない正しい整列を可能にする。不確かな情報が試みられた場合、十分な不正証明があるだろうし、作成者はペナルティされうる。しばらくした後にブロックはファイナライズされ、結果的にルートチェーンもファイナリティにいたる。

5.6　引き出し

Plasmaはルートチェーンを離れて、ネイティブなコインやトークン（例：ETHやERC20トークン）で資金を預けることを可能にする。さらに、情報可用性（提出されたUTXOの正当性）の供給されたルートチェーンによって制限されたプラズマチェーン上での状態遷移も可能にする。ただし情報可用性の失敗のイベントにおいては、mass exitをPlasmaチェーン試みねばならない。最終的には、Plasmaチェーンに保持された資金の単純な引き出しもまた可能である。しかし、通常の操作おいては、ユーザーは単純な引き出しができる。

5.6.1　単純引き出し

単純な引き出しについて、ユーザーはルートチェーンにコミットされ、Plasmaチェーンにおいて最終的にはファイナライズされた資金を引き出すことだけを許されている。デポジット、コンパクトに表現された台帳の状態、そして状態遷移

の設計についてはすでに解説した。このフェーズまで、不正証明をのぞいたどんな現在のPlasmaチェーンの台帳の状態のルートチェーンへの発行はなかった。しかし引き出しによって、資金が現在もPlasmaチェーンに保持されているという特別な証明が必要になる。出金はPlasmaチェーンとルートチェーンの間の硬貨の交換可能性を保証するので、最もクリティカルな構成要素である。もしユーザーが資金をPlasmaチェーンにデポジットすることが可能なら、状態遷移をして、それらの当事者は資金を引き出すことができ、そして値はルートチェーン上のコインの値と密接に対応する必要がある。場合によってはPlasmaチェーン上の資金はより大きなトランザクションキャパシティを持つので便利ですが、セキュリティは最終的にルートチェーンに依存します。単純引き出しについて、全ての資金は大きな供託金を必要とし、全ての引き出しは大きな供託金を不正証明として含まねばなりません。もし現在のブロックデータが利用可能なら、サードパーティサービスがPlasmaチェーンがアクティブであることを検証でき、引き出し証明が正当であると保証できるので、サードパーティがこの証明を例外的に安いコストで提供することが可能である。全てのPlasmaチェーンの参加者は状態を更新する際に、全ての親Plasmaチェーンとルートチェーンを　特定のアカウント/アウトプットに対して進行中の引き出しがないと保証するために検証しなければならない。

　もし引き出しが進行中であれば、後続のブロックはコインやトークンを使用できず、いかなるビザンチン障害を引き起

こしうる振る舞いはコンセンサスを違反するし、不正証明とペナルティとルートチェーンのPlasmaコントラクトによるブロックのロールバックを受けなければならない。

引き出しは以下のステップで行われる。

1. 署名された引き出しのトランザクションがルートチェーンか親Plasmaチェーンに提出される。引き出し額は全アウトプット額と等しくなければならない（部分引き出しは不可)。複数のアウトプットも同時に引き出しされうるが、同じPlasmaチェーンに存在するアウトプットでなければならない。アウトプットのビットマップの位置は引き出しの一部として公開される。追加の供託金は偽の引き出しリクエストをペナルティするために引き出しの一部として設置される。
2. 申し立て(dispute)を許容するための既定のタイムアウト期間が存在する。これはLighning Networkの”dispute period”と同じものである。このケースでは、もし誰かがアウトプットが既に引き出し先のチェーン（多くの場合ルートチェーン）の中で使用されていると証明できるなら、引き出しはキャンセルされ供託された引き出しリクエストはロストする。チェーンを監視している誰もがこの申し立てを行える。もし使用されたアウトプットの不正証明が用いられた場合、供託金は失われ、引き出しはキャンセルされる。
3. 他のいかなるより少ない(小さい) block confirmation

height を持つ引き出しリクエストのタイムアウトを待つための二度目の遅延が存在する。これは順番に整列された引き出しを特定のPlasmaチェーンやルートチェーンで強制するためのものである。

4. もしPlasmaのスマートコントラクトで定義された同意された申し立て時間が過ぎ、不正証明がルートチェーンや親チェーンで供給されなかった場合、その引き出しは正当であると推定され、引き出し主は資金をルート/親チェーンに償還できる。引き出しはUTXO（あるいは実装によってはアカウント）の生成時刻が古いものから順に処理される。

注釈として、Plasmaチェーン上での ”block withholding attack” のイベント中に引き出しをすることは、経済的に可能な条件において、ありえるのを留意してほしい。不正証明は、コンパクトに証明可能な、ネットワーク上の誰かが同じアウトプットからの重複した署名の消費を証明することだけを必要とする。Lightning Networkと他のstate channelにおいては、追加要求はより高いnonceを同様に証明しないといけない。チャネルについては、もし低いnonceの引き出しが試みられた場合、資金はPlasmaチェーンの中に残る、正当な署名での引き出しで依然利用可能ということになる。他の実装もまた可能であるが、Plasmaチェーンでの不正証明スマートコントラクトの生成の一部として事前読み出しされた設計が必要であろう。

通常の引き出しが遅く、高額な処理であるので、引き出しが一つの引き出しに合体するか、他者が喜んでコインを他のチェーンのためにLightning NetworkやAtomic Swapを遣ってスワップするというのはありそうなことである。

5.6.2 高速引き出し

高速引き出しは単純引き出しと同じ構造で、しかし資金はAtomic Swapを行うコントラクトに送られる。スワップされる資金は、low timelockなルート/親チェーンの資金と、high timelockなPlasmaチェーンをexitする資金だ。高速引き出しはインスタントではない。しかしPlasmaチェーンがビザンチン（分散合意がない/block withholding attackも含む）ではない条件において、劇的にトランザクションがファイナリティを得るまでの引き出し時間を減らす。この理由から、高速引き出しのスワップはblock withholding attackの最中には行えず、遅いmass withdrawalリクエストが代わりに必要になるだろう。高速引き出しは以下のステップで実現される。

1. Aliceは資金をルートチェーンに引き出したいが、待ちたくない。彼女はその利便性のために付加価値を払う準備がある。Larry（流動性供給者）はこれをサービスとして供給する。AliceとLarryはルートチェーンへの引き出しのために協力する。Plasmaチェーンはビザンチンではない(分散合意がある)とみなす。
2. 資金は、Plasmaチェーンの特定のアウトプットのコントラクトにロックされる。これは通常の送金と同じ方法

であり、双方の当事者はトランザクションをブロードキャストし、あとでPlasmaブロックで観測したトランザクションをコミットする。コントラクトのtermsは、もしコントラクトがルートチェーンにブロードキャストされてファイナライズされるなら、支払いはPlasmaチェーンまで回される。もしトランザクション証明がないなら、Aliceは資金を償還できる。Aliceがpreimageを生成することでHTLC(LNに用いられるエスクロー的プロトコル)として実装することも可能であり、彼女がそれを受け入れ可能で資金が送信されたとみなしたあとでのみ解放される。

3. 上記のPlasmaブロックがfinalizeされ、Larryがコントラクトの条件に合致したイベントで資金を償還できる確信があるとき、LarryはAliceに特定の金額（この金額は彼が受け取るであろう金額は彼がサービスによって徴収する金額より小さい必要がある）の支払いを可能にするオンチェーンのコントラクトを作成する。

この例で流動性提供者であるLarryはアクティブで、このスワップを受け入れる前にPlasmaチェーンで完全にvalidateされていなければならない。もしLarryがPlasmaチェーンで完全にvalidateされえない（それかルートチェーンで定義された不正証明のスマートコントラクトに沿わない）とき、彼は引き出しを実行すべきではない。もしLarryがこのチェーンの資金を欲さず、ルートチェーンの資金を欲する場

合、Larryは引き出しの一部としてのこのAtomic Swapを実行したあとか完遂したあとに引き出しを開始できる。多くの場合、Plasmaチェーン間の流動性提供者と共に行う差金決済(netsettlement)は費用対効果がよい。送金はLingtning NetworkかAtomic Swapのような迅速なfinalityを提供する手段でPlasmaチェーン間で行われる。これはAtomic crosschain swapなので、AliceとLarryは保管上の(custodial)信頼を互いに必要としない。

Aliceは彼女の資金をルート/親チェーンで保有し、Larryはのちにルート/親チェーンに完全なアクセスを得る。低コストのブロック利用とfinalityのあるルートチェーンのnon-Byzantineな振る舞いにおいて、Larryはたとえ彼がPlasmaチェーン自体を信頼していなくとも、資金を受け取れることを十分確信できる。

5.7　敵対的Mass Withdrawal

が、これはプロトコルに必要というわけではなく、これは状態の経済的ロバストさ（低いgas代）をblock withholding攻撃中に得るための設計だ。もし誰かがアカウントの状態をPlasmaチェーン内で使いたいと考えた時、その人は支払いの階層構造のような他の設計に頼る。加えて、UTXOモデルがここでは用いられていることに留意してほしく、しかしこのシステムはルートチェーンがアカウントモデルを用いているときにワークする。さらに、もし大量引き出しが必要でないか望まれてない昨日だった場合、アカウントモデルをPlasmaチ

ェーン内での資金の保管に使うことが可能だし、単純な引き出しが可能になる（インクリメンタルなIDを割り振る形で)。はじめに紹介したPlasmaの設計の考察はblock withholding攻撃を不正証明（あるいはその他のデータ欠損の痕跡）で防ぐことに関係していて、データの使用不可能性の特定作業を緩和する必要がある。もしブロックの利用不可能性がユーザーによってPlasmaチェーンで検出された場合、特定の日時までに参加者がチェーンを離脱することは急務である。もし期日までに離脱しきれなかった場合の結果はLightning Networkで不正引き出しに申し立てをしなかった場合と同じである。このメカニズムはPlasmaを正しく動作させるためのキーである。Plasmaはもしユーザーがblock withholdingを検知したら、すぐさまチェーンを離脱する責任があるという事実に依拠している。その根拠としてはルートチェーンは子チェーンのブロックがwithholdされていること（ユーザーがブロックをまだ受け取っていないと主張することか、Plasmaチェーンがユーザーがブロックが利用可能であると認識することを拒否したり嘘をついている際に申し立てること）を知ることができないからだ。結果として、申し立てられたブロックの利用不可能性に関するコストは伝統的に現状のオンチェーンを公開しようとする（Lightning Networkと同様)。しかし、大きいブロックと状態遷移について、これはかなり高額になり、それらのコストを誰が支払うのか不透明なのでPlasmaはこの実装を用いない。代わりに、Plasmaチェーンが敵対的にブロックをwithholdしているとユーザーが信じていて、将来、状態遷

移を強制する能力に影響を与えかねなくて、別のチェーンに可能な限り速やかに移動しなければならないことをPlasmaは想定している。したがって、ブロックが利用不可である限りにおいてこれを敵対的大量引き出しと定義し、Plasmaチェーンが敵対的かビザンチンであると想定する。大量exitはPlasmaチェーンのビザンチンな振る舞いが一定時間の遅延とチェーンの停止よりほかはユーザーの資本等に影響を与えないことを保証する。これは追加のSNARKsを用いたセキュリティ緩和が将来的に利用されることを想定されるが、しかし特定の設計はオープンクエスチョンとして残される。この実装はルートチェーンでの観察者の周期的な活動性による引き出しに関してSNARKsに依存するわけではなく、しかし状態遷移をPlasmaチェーン内で強制することによって、攻撃する能力や、ビザンチンなPlasmaチェーンを、周期的に観察していないユーザーから資金を盗むための敵対的なblock withholdに導くことができる能力は、SNARKsのセキュリティ特性により最小化され強制される。

このケースでは、状態遷移をするためのSNARKsの証明とより大きな状態遷移への保証を得るためのSNARKsの証明を必要とするだろう。しかしPlasmaの目的はユーザーがチェーンを監視していて、スマートコントラクトがこのメカニズムを適切に処理することと、ルートチェーン上で引き出しを可能にする条件において正しい状態遷移の振る舞いをSNARKsに依存することではない。スマートコントラクトをサポートしたサードパーティーチェーンによる再帰的なSNARKs証明の

コミットによってのみオフチェーンの状態が可能であることを保証することによって、現在の状態の正当性を保証することは、Lightning Networkにも同様の利点として存在する。Plasmaチェーンは何層もの防衛手段によってセキュアになる。まず第一にセキュアな要素とハードウェアによって、次にSNARKs/STARKsによって、最後にオンチェーンのゲーム理論的な相互作用によってだ。第一ラインが突破されても、第二ラインが優れた暗号化方式で防衛する。最終ラインは公開された透明なゲーム理論的駆け引きだ。まずは我々は最終ラインを使って提案する。大量引き出しは以下の方法で行われるexitの相互ゲームによって達成される。

1. Aliceは他者と強調してPlasmaチェーンからの大量exitを試みる。複数の大量exitが同時に起こるが、彼らは引き出しを重複することはない。このシナリオでかられは残高を更新し、順番に処理され、重複を生み出したものはペナルティを受ける。全ての当事者は資金を他のPlasmaチェーンに送金せねばならない。 exitを処理する役目のPatはexitを喜んでさばく。
2. Patは目的地となるPlasmaチェーンに資金を送金する手はずを整え、自動的に大量exitがfinalizeされた際に新しいチェーンで資金が利用可能になるようにする。
3. PatはPlasmaチェーンを情報の利用可能性に関して検証する。この指摘は申し立て受け入れ可能期間になされねばならないしPlasmaのfinalityが得られるまでの期間（ル

ートチェーンとは別）までに、そしてスマートコントラクトで決められた期間までになされなければならない。Patはpending中の新しいPlasmaチェーン上の目的地の台帳を参加者に公開する。Patはexitを希望する全ての署名を参加者から集める（Aliceも含む)。Patは最新のブロックにおいて全ての参加者がexitを行う権利があるかを検証する。Patは大量の供託金とともにexitトランザクションを作成する（ルートチェーンのスマートコントラクトを用いる)。Patは参加者から利用料を徴収する。

4. ユーザーは署名を手に入れたあと再度大量引き出しに署名する。これはPatがもはやペナルティされないという状態をユーザーに可能にし、ロックインされる。二度目の署名を提出していないユーザーは含まれない。
5. Patは他にexitがないか監視する。重複があれば取り除く。exitトランザクションに署名してルートチェーンか親チェーンにブロードキャストする。重複のシナリオでチェーンの親は主導権を取り（ルートチェーンがもっとも優先)。先のトランザクションの方が優先である。MEIT(大量exit開始トランザクション/Mass Exit Initiation Transaction)のブロードキャストに際して、Patは以下の情婦を正しく持っていることの証明を供託する。ブロックの正当性、そのblock heightのUTXOたち、finalizeしていないこと、マークル化されたbitmapからUTXOへのMap、コミットされた額（高速証明のためのマークル木)、Aliceとその他の署名などは、申し立てが起きた時に利用可能

である。MEITの一部としてPatは完全なexit中の状態のbitmapを発行する。参加者はもし何かが不正に感じられたら、ルート/親チェーンを監視していて何がexitしようとしていたり申し立てられているのかを検証できる。MEITのFinalityは長く、数週間かかるなのでMEITは最後の手段的なトランザクションになる（将来の高速化はSNARKsでおそらく可能）

6. もし重複引き出しがあればPatは短い猶予期間の間に引き出されようとしている残高やbitmapを更新する選択肢がある。
7. いかなるネットワークの参加者もMEITで証明されたデータにDMET(申し立てられた大量exitトランザクション/Disputed MassExit Transaction)で申し立てがてきる。しかしPatは、将来にブロックがアウトプットを置き換えられるかどうか知ることができず、Patはトランザクションが表ライのブロックで使用されてもペナルティされないはずである（ユーザーはされる)。もし申し立てが行われると、資金が申し立ての完了までロックされる。これらの申し立ては猶予期間の初期に行われるはずで、申し立てが政党ならばPatは引き出される残高を更新しなければならない。
8. もし問題がなければ、既定のMEITのfinalize期間を経て、ユーザーは資金を得る。

Plasmaチェーンのfinalizeの期間はチェーンを周期的に監

視しなければいけない期間である。その期間ののち、全てのPlasmaチェーンの参加者はブロックデータの利用可能性を保持していることになる。事実上、MEITがPatに作られた際、Patはそれぞれの引き出しのアウトプットへの署名を持っており、特定のPlasmaチェーンの高さにおいて情報の正しさを証明する。Patは証明期間のあとにアウトプットが二重支払いされていても罰されない（block withholding攻撃もPatをペナルティしない)。

5.7.1　Mass Withdrawalへの異議申し立て:不正な引き出しへの訴え

Aliceのようなユーザーが、Patが彼女の同意なしに大量引き出しを試みているのを見たら、彼女はその引き出しを申し立てを用いて無効化できる。

1. AliceはPatが彼女のアウトプットを含んだPlasmaチェーンの大量引き出しを試みているのを見た。このひきひとつのbitmapフィールドは利用可能だ。Aliceは大量の供託金とともに申し立てる。この供託は申し立ては処理されないということを証明している。
2. もししばらく経っても異議申し立てがない場合、Aliceは供託金をリファンドできる。そして全MEITはキャンセルされる。もしPatかいずれかの当事者が彼女の不正確な引き出し申請に対して不正証明を出し、出金申請への異議申し立てが証明されれば、彼女の供託金は切り捨てられる。

参加者は署名がMEITの第二フェーズで必要ということについて自覚があり、したがって彼らはもし出金申請が詐欺的だったら、その申請に異議を申し立てるための十分な情報を持つ。インセンティブは不正な出金申請をすることに対して逆らっていて、ルートチェーンでの非検閲性やブロックの可用性の条件で、彼らはペナルティを受ける。

5.7.2　異議を申し立てられたMassExitトランザクション

アウトプットがのちのブロックでのMEITによって使用されたとき、Patはこれを知らないし、彼はblock withholdingを証明できないので、彼はペナルティを受けるべきではない。おそらくここでは似たようなbitmapされた集合に向けた複数の異議申し立てがあり、しかしそれらは大きな供託が備えてなければならない。いかなる参加者も大きな供託付きの支払いのbitmap/rangeを表現できる。高額の供託はコインがブロックヘッダーへのコミットとともにのちのブロックで使われたことの証明である。しかしこの異議申し立てはコンパクトには証明さえれず、したがって他の繰り返される出金申請は、CDMET(Challenge on the Disputed MassExit Transaction / 異議を申し立てられたMassExitトランザクションにおける出金申請)を発行することが可能だこの異議申し立てにおける出金申請は以下の通りだ

1. Aliceは誰か（例：そのチェーンでblock withholdingをしている操作主）が彼女が参加している大量引き出しに異議申し立てをしていることに気づく。彼女は異議申し立

て中に,その申し立ての提出者の申し立ては正当になりえないことを証明する高額な供託とともに出金申請を提出する。

2. 出金への異議申し立ての提出者は、一定時間内にその出金申請に応えなければならない。もし提出者が支払い証明(本質的にはのちにそのUTXOを使用したトランザクションの署名)を提供できない場合、Aliceは放免され、全体の異議申し立てがキャンセルされる（これが重複した異議申し立てが受け入れられる理由だ)。もし提出者がコイン(UTXO)が使用されていると証明できるなら、Aliceは高額な供託を失い、異議申し立ては継続する。

5.8 UTXOのリサイクル

使用したアウトプットがfinalizeされたあと、スペース節約のためにUTXO bitmapを再利用するのは可能である。

5.9 要約

「大量引き出しゲーム」の結果、最も楽観的なケースで1引き出しごとに1 2bitだけを使う引き出しが可能になる。

大量exitはblock withholding攻撃のシナリオで必要になるものである。しかしこれはとても高くつく処理になるだろう。多くの理由で、ルートチェーンに依存しない別の戦略もまた必要になるだろう。この実装は多くの参加者に彼らの資金を子チェーンで持つことを可能にし、もしブロック情報が利用

可能なら状態の無効化は不正証明を経由して起こり、状態遷移（例：支払い）は可能で、引き出しも可能で、大量exit（けれども遅延はある）もblock withholding攻撃のシナリオにおいて可能になる。

6　ブロックチェーンの中にあるブロックチェーン

すでに説明したように、Plasmaのコアの設計は計算処理をスケールさせるためのものである。しかし我々は不正証明を発行したり、ブロックスペースの可用性を確保するためにblock withholding問題と争わねばならない。Plasmaにおけるblock withholdingへの解決策は、チェーンの停止やwithholdingに対して大量exitを実行できるシステムである。しかし大量exitトランザクションは特にもしUTXOたちの数が多くそのbitmapが発行されねばならないとき非常に高価だ。加えて、単一のexitだけを実行する方が望ましい。大量引き出しトランザクションは多数の参加者を巻き込んだ複雑な相互ゲームを要求する。最後の手段として用いるべきだ。代わりに、状態を証明するための存在できる高次と低次の法廷を用意した。当事者はルートチェーンを上級法廷としてみることができ、それは下級法廷すべての代表である。彼らの法廷での力を可能にするのはルートチェーンの法である。これは法廷にスケーラビリティを与え、下級法廷の状態が異議申し立てか停止に陥ったとき、これらはより法廷を代表できる上級法廷に移動する必要がある。状態の検証の高級法廷へのブロードキャストは常に可能だが、それは高くつくだろう。

全ての状態はマークル化されていて、ルートチェーンにコミットされる。もっとも楽観的なケースではブロックヘッダが直接親チェーンに発行されて、親チェーンはそれをさらに親に伝え、ルートチェーンに届くまで続ける。ブロックヘッダの中身はマークル化された親チェーンのブロックに対するコミットだ。トランザクションはPlasmaチェーンやなんらかの親のPlasmaチェーン、あるいはルートチェーンにに提出される。その目的はfungibilityと検閲耐性を保証するためだ。特に、チェーンの停止やブロックの非開示のイベントでは、ユーザーはまだ資金を引き出しうる。ブロックのコミットが提出された時、それが承認される前にルートチェーンに一定の量のconfirmationが承認されるのを待たねばならない。その間、不正証明がルートチェーンと中間Plasmaチェーンに提出されるかもしれない（それはブロックルートを通してルートチェーンにコミットされる)。どの個別のPlasmaチェーンもPlasmaブロックにバンドルされたコミットのステートマシンを実行している。個々のPlasmaチェーンは子Plasmaチェーンの詳細を詮索するかもしれないし、しないかもしれない。

代わりに、彼らは確認されたPlasmaチェーンの残高を持っている。子Plasmaチェーンが自身の状態を更新した時、彼らはPlasmaブロックヘッダのハッシュをどれかの親に提出する。これは特定のブロックステートは複数の親チェーンに提出されうることを示唆している。複製イベント中、それは不正にはならないかもしれない（しかしアプリケーションに依存した特定のコンセンサスの状態においてペナルティを受けうる)。

一方で、もし状態のごまかし、例えば親1にコミットされた状態が親2と異なるなどがあった場合、Plasmaチェーンへの供託者は彼らのデポジットを失う。

新たな子の状態更新は以下の彼らの状態更新メッセージを使うだろう：支払われた手数料（そして通貨単位)、コミットされたルートブロックハッシュ、前のブロックは種、コミットする親ブロックハッシュ、デポジット証明、引き出し証明など。親チェーンになにがコミットされていようが、子チェーンは再帰的な全ての親のそれらのデータを観測している。kれはそれがコミットするはずのそれが持つ証明は、ごまかされも二重支払いトランザクションもされていない（し、それゆえにトランザクションをごまかしのイベント中でslashにさらしている)。ごまかしのイベント中で、親チェーンの状態は常に優先される。インセンティブは彼らをしるあらゆる当事者によってごまかしの開示を作る。デポジットと引き出しはPlasmaと親チェーン両方で可能である。

引き出しはPlasmaチェーンの間で十分な流動性と他の当事者が喜んで着資金をどこか他の場所に置く条件で可能である。これはクロスチェーンのアトミックスワップを通してなされえる。もし誰かがメインチェーンを使ってクリアしたいと願ったとき、HTLCのチェーン間のオンチェーンライトニングペイメントのように実装するのは可能である。すべての不正証明はマークル化されたチェーンコミットの証明を提供しなければならない。偽の証明は特定の詐欺ブロックに責任のあるPlasmaチェーンをペナルティする。初めのデザインの複雑

性は検閲耐性に関する複数の親チェーン間のトランザクション状態のブロードキャストである。早期のイテレーションは状態遷移やトランザクションが個々のPlasmaチェーンの中だけで実行されていると推測でき、他のチェーンとのインタラクションはコミットされた親/子チェーンへのmessage passingとデポジット・引き出しだけである（訳注：Shard間通信のないごくシンプルなShardingということ)。その方法では、初期においては、デポジットと引き出しの証明絡みの部分だけが複雑な程度である。データのコミットは取り込み証明の一部分として推測される。

6.1 チェーン内で資金を受け取る

ブロックチェーン群の中のこの階層的なフレームワークでは、AliceがPlasmaチェーンの中でBobに資金を送ろうとする場合、次のような処理が行われる。

3つの深いレベル：

1. アリスはボブに資金を送ろうとしていることをボブと調整します。AliceはBobに、Bobが資金を受け取るPlasmaチェーンを開示します。支払いを受け取るため、ボブは特別に、彼が資金を受け取るつもりであるルートチェーン上のスマートコントラクトを確認する必要があります。（スマートコントラクトの契約コード/メカニズム、許容可能なconsensus exit delaysなど）
2. 確認した結果、支払いに問題なければ、彼らは支払い条

件を定義するステートメントに事前署名しますが、多くの場合、十分な成熟度を持つブロックチェーンにブロックを含めることにより支払いの証拠となりますが、状況次第では、paytocontracthashにすることもできます。これはオンチェーンではなく、単に他人に証明するための決済の条件を付け加えるだけです。

3. AliceはPlasmaチェーン内で支払いを行います。ブロックはバリデータによって承認され、そしてブロックヘッダーへのコミットメントは親チェーンのブロック内に公開されます。マークライズ化された子Plasmaチェーンへのコミットメントはすべての親チェーンのブロックに含まれ、最終的にルートブロックチェーンに含まれます。
4. Bobはルートブロックチェーンと完全に同期し、資金が入っているチェーンとその親のいくつかを検証しますBobは、彼の資金が含まれていない他のPlasmaチェーンを検証する必要はありません。最悪の場合は、ボブはアリスが十分な成熟度をもつPlasmaチェーンで支払いを行ったことを完全に検証することができます。

しかし、迅速な最終決定が求められる場合、アリスは新しいブロックで支払われた支払いを承認することができます(Plasmaチェーン内での支払い受け取りについては、以前の定義を参照してください)。

アリスが支払いを承認して、ボブがそれを受け入れるなら(彼は出金を証明することができるので)、最終的に支払いは

達成されたと見なされます。 BobはこのPlasmaチェーンから資金を引き出すことができます。

この設計の重要な側面は、子ブロックチェーンの検証に全責任を負うということです。 BobがPlasmaチェーンとすべての親を検証しない場合（最終的にはルートチェーンへの定期的なコミットメントが公開されている)、それが達成されたものとして扱われるべきではありません。Lightning Networkの構築と同様に、Bobは他のPlasmaブロックチェーンで何が起こるかを気にする必要はありません。彼は彼にとって重要なチェーンの正確さだけを観察します。彼はコインを使用する能力を持っているとき、彼はそれらを使うことができると確信しています。

6.2 親チェーンから資金を受け取る

親チェーンから資金を受け取ることは、ルートブロックチェーンからのデポジットに似ています。唯一の違いは、受信者がすべての親プラズマ・チェーン（単にプラズマ・チェーン自体ではなく）を検証する必要があるということです。子プラズマチェーンへの入金は速いです。

6.3 ツリーからウェブへ

上記の説明は単一の親チェーンに関するものですが、プラズマチェーンは複数のルートブロックチェーンを監視することが可能です。これにより、子チェーンとのバランスを更新することができます。ある親の障害がすぐにすべての参加者

によって認識されない可能性があるため、注意を払う必要があり、cascading systemic failuresはtimedelaysとクロスチェーン流動性の推定を最小化することにより、緩和される必要がありますこのための正しい構成は未解決の問題です。

6.4　ブロック源泉徴収問題の緩和

出金トランザクションをブロードキャストすることができる多くのvenuesを作ることにより、停止しているか、ブロックを保留しているチェーンから出る可能性のある多くの場を存在させることができます。

子チェーンが失敗した場合、ルートチェインでトランザクションが高価になっても、親チェーンで個々の単純な終了を処理できます。これにより、親プラズマ・チェーンの1つが正しく動作していることが保証されていれば、Plasmaチェーン上のマイクロ・ペイメント・アウトプットを保持する際にある程度の信頼感を持たせることができます。このゴールは、第一の動機であり、cascading failuresの影響を緩和します。

十分に大きな出力バランスが保持されており、有効な時間がない場合には、有効な引受を行う必要はありませんが、一つのlowvalue output（取引手数料の支払いが効果になってしまう場合）を保有している場合は、親のPlasmaチェーンの1つが有効であるという、ある程度の裏付けが必要です。より大きな裏付けが欲しい場合は、各レベルで各Plasmaチェーンを実行する多くの独立した当事者が深くネストされたチェーンを実行できます。このようにすると、いくつかのトレードオ

フが存在しますが、特定のプラズマチェーンがビザンチンになるように、誰もが新しいチェーンにmasswithdrawlする必要があります。ビザンチンではない親がいる場合ですが、親がビザンチンチェーンのコミットメントを処理することを拒否した場合は、別のチェーンへの迅速な移行と操作を継続することが可能です。

これには、子チェーンが失敗した場合のプロセストランザクション以外の処理を行わないサービスが発生する可能性があります。

このサービスのオペレータは、子チェーンが失敗しない限り、何もする必要はありません（障害が発生するまで仮想的にサーバをオフにできるだけの十分な受動的なポイントまで、ブロックヘッダは自動的にそれらをスキップしてパッシブオペレータの上のレベルにチェーンします)。親チェーンでは非常に高いトランザクション量（ブロックサイズ/ガス制限）を持つことができるため、親チェーンの引き出しの多くは、一括引き出しの代わりに単純な引き出しになると予想しています。

6.5 Exit処理

親チェーンまたはルートチェーンにMass exitsが可能です。子チェーンがビザンチンの動作を開始した場合、ネストされた親チェーンがないPlasmaチェーンと同じく、どのステートも無効であると推測されます。同様に、Mass exitsは、ビザンチン親チェーンから迅速に抜け出す方法です。親チェーン

（または子チェーン自体）をその親またはルートチェーンにスキップすることは可能です。

デザインは複雑なように見えるかもしれませんが、いくつかのチェーンがビザンチンであれば、そのすべての子チェーンが行動しなければならないということです。ハートビートを介して、整合なしに抜け出す最適化があります。（出口はユーザ側での署名なしに取り消すことができるのがデフォルトで、プラズマチェーン自体がそれを受け取ったというコミットメントをしていますが、時期尚早の最適化である可能性があります）

この設計は基本的にsimple exitまたはmass exitと同じですが、ネストされたチェーンをサポートするためにデザインに若干の変更があります。出口は重複することがありますが、親チェーンの出口は常に優先されます。親チェーンがビザンチンの動作を開始すると、出口もルートチェーンでコミットできます。（ビザンチンプラズマチェーンであると認識される）親/ルートチェーンの複製された出口の状態を反映および更新し、複製された出口をそれ自身の鎖で取り消すのは、その責任です。しかし、そうしなければ、ユーザーの資金はルートチェーン上で利用できるようになります。

親がビザンチンであるが、資金を保有している子チェーンが正しく動作している場合、複雑なmass exit transactionを避けることができます。

参加者は資金を送付するための新しいチェーンを見つけ、流動性プロバイダーがこの子チェーンで資金を受け取れるよ

うに簡単なexitを行い、他のユーザーは新しいチェーン（ビザンチンの親なし）で資金を受け取ります。子チェーンブロックのコミットメントは、ルートチェーンまたは上位レベルの親に公開されます（ビザンチンノードを避ける)。ユーザーはすぐに新しいチェーンに資金をもち、流動性提供者は親または最も高い親に後で資金を出す。これは、新しい資金が新しいチェーンに迅速に配分され、exitが急速に行われることを目的としています。

6.6　スケーラビリティ

これにより、UTXOのビットマップスケーラビリティが可能になります。ビットマップが大きくなりすぎると、ビットマップを複数の子チェーンに分割するだけで済みます。子チェーンへは、出力の代わりにブロック高さnonces（および候補チェーンヒント）を持つ口座残高として表されると推定されます。同様に、UTXOの代わりにアカウントを使用することを選り好むステートでは、簡単な引き出しをサポートするだけのトレードオフも可能です。最終的な結果は、ユーザーにとって大きなスケーラビリティです。彼らは、彼らの資金が保持されているプラズマチェーン（その親チェーン）を観察するだけでよい。これは、自分自身に影響を与えるデータセットの検証を効果的に分担することができます。

7　PlasmaのProof of Stake

この章ではシンプルなPoSの構築を目的とする。おそらく

最適なPoSの設計にはなりそうにないが、Plasmaチェーンで何が可能かということの概説になればと思う。これまで我々はPlasmaチェーンのブロックに署名する責任のあるオペレーターは単一のエンティティだと仮定してきた。もし彼らが不正なブロックを生成した場合、ブロックデータを持つ誰もが不正証明を生成し、その部録をロールバックし、オペレーターにペナルティを与えることができる。この証明はオペレーターがブロックに署名をしているから可能なことである。ルートチェーンのPlasmaブロックマークル化されたコミットの発行（そしてPlasmaチェーンのもっともblock heightの高いブロックがその子チェーンの状態更新のコミットを含んでいること）は、したがって状態更新が整列され、適切な振る舞いになるように制約されている。

しかし、多くのケースでシングルパーティーののPoA(Proof of Authority)チェーンよりもPoSチェーンを作ることが好ましい。block withholdingのリスクを最小化できるからだ(PoSとPoAのチェーンをネストさせて両者のメリットを得ることも可能である)。トークンのあるPoSチェーンは、トークン価値をビザンチンな振る舞いが毀損するので、トークンホルダーに正しくオペレーションするようインセンティブを与えることもできる。Plasmaチェーンのトークン利用についての詳細は後に詳解する。ルートチェーンのロバストさに依存しているので、PoSの構築はPlasmaの中で構築するほうが容易い。withholding攻撃、ファイナリティ、その他の要素に関する問題はルートチェーンの信頼性に依存する。Plasmaチェ

ーンはルートチェーンと同等にセキュアになれる。もしルートチェーンがPoWベースで動いているなら、これはPoW上のPoSだ（PoWの上のPlasmaとも言える）。もしルートチェーンがPoSなら、この構造はPoS上のPoSで、PoSのメカニズムはルートチェーンのPoSよりも幾分シンプルになるだろう。

ナカモトコンセンサス（PoW）のインセンティブを模倣してみる。もっとも重要なインセンティブは、他のマイナーへのブロック伝播だ。多くの既存のPoSメカニズムの提案はリーダー選挙に依存していて、時刻t0においてリーターが立候補し、時刻t1においてリーダーはブロックを生成する権利を得る。これはブロック伝播におけるナカモトコンセンサスのインセンティブを模倣できていない。ナカモトコンセンサスはリーダー選出をするが、それは確率的なリーダー選出である。もし誰かがブロックを見つけたら、その人は自分がリーダーだろうと信じるが、それは完全に確かなわけではない。他の誰かも同時にブロックを掘る可能性は残されているのだ。オッズを最大化する最善の方法は発見したブロックを可能な限り遠く広くに素早く伝播することなのだ。これは情報可用性に対してインセンティブをもたらす。 PlasmaのPoSの設計はこれと似たようなことをする必要がある。ブロックを可能な限り遠く広く伝播することを支援したいのでトレードオフを考えなければならない。しかし他の設計もありえるかもしれない（特に、ブランチにランダムなスコアを付与し、もっとも高いスコアのブランチに決定した事実に基づいたランダムで確率的なリーダー選出を用いて）。

これはPoSのシンプルな設計の提案だが、おそらく最適なものにかなり近いと思われる。ゴールはPlasmaの使用に耐えうるシンプルな設計を得ることだ。

強制力をともなうメカニズムを作る代わりに、正しい振る舞い（即時のブロック伝播）へ単純にインセンティブを作るアプローチだ。フィーはルートチェーンのコントラクトによって配布され、もし必要なら定期的に払い出しできる。しかし監査はPlasmaチェーン内で行われる。 Stakingコントラクトの一部として、資金は移譲したstakerに配布される。stakerはユーザーの代わりに振る舞う責任があり、stakerが不正を行うとユーザーがペナルティを受ける。Stakingは一定時間でコミットされる（例：三ヶ月）。stakerごとの最少額は全トークンの1%で、最大額は最大トークン量の5%だ。もし誰かが5%以上割り当てたいのなら、複数のアイデンティティを使うとよい（51%カルテルの効能を最小化してデータの分散化を最大化することが目的だ）。資金は100Plasmaブロックが全参加者の代表であるように割り当てされる。例えば誰かが3%の資金をstakeしたとして彼らは過去の100ブロックの3%でないといけない。もしその量よりもstakeが多ければ、そのstakerはブロックの過大なコミットによる余剰分の報酬は得られない。もし過去の100ブロックのうち3つより少ないブロックしかなかった場合、現在のブロック生成者はより少ないリワードを得る。ルートチェーンのブロック1つに対して1ブロックしか割り当てられない。

これは全参加者がみなのブロックを平等に保持することを

促進する。想定では強制メカニズムは必要なく、参加者は報酬の最大化のためになんらかのスキーム（例：ラウンドロビン）を用意および保証するだろう。もし彼らが不正確なブロックの数のせいでトランザクション手数料を最大化できなかったとき、資金はプールに割り当てられ、将来のブロックで割り当てられる。結果として全員の参加を経済的に促進する。しかし、これはstakerたちの正確な参加を促進しているだけなので、完全ではない。(TODO)毎ブロックは、過去100ブロックに関するブロックのランダムな部分での、データへのマークル化されたコミットである。これはstakerに全ブロックデータ持つことを強制し、ブロック作成者に継続的にブロックを全stakerに伝播することを強制する。このチェーンのチップは最大の報酬によって決定され、もし並行するブランチがあるなら、勝者は最大の手数料報酬を最大の参加者から得た者である。この設計は51%攻撃を防止するために設計したものではなく、代わりにブロック伝播を促進するために設計された(もし誰かがblock withholdを行った場合に対して)。加えて、この設計は情報可用性とルートチェーンに偏りなくブロックが含まれていることに依存していて、この手のPoSをルートチェーンで作るのはデータ可用性と検閲インセンティブのせいで不可能である。

8　経済的なインセンティブ

PoSバリデーションモデルの中で、コントラクトを用いて正しいオペレーションに沿ったインセンティブを設定すること

は可能であった。忠誠を示す供託はチェーンへの正確さを保証する一方で、我々はデータ可用性と停止行為への抑止へのインセンティブを作らなければならない。

Plasmaチェーンのみで用いることのできるトークンでstakingを可能にすることで、トークン価値はstakingによる将来利益をディス館とした者になるので、オペレーションを続けるインセンティブを保証できる。続けてネットワーク障害はトークンの価値を減じるため、個々の参加者はネットワークの継続に強いインセンティブを持つ。Plasmaチェーンのオペレーターはトランザクションをオンチェーンでブロードキャストすることで手数料を得る。計算処理はオペレーションごとに異なり、特に複雑なオペレーションに関しては手数料は滝のように渾々と流れる。深い子チェーンでより多くトランザクションが実行されることにインセンティブがある一方で、送信された資金のコミットメントや子チェーンでの計算処理を作ることは可能であり、伝播される。これは親チェーンに子チェーンの計算について課金することを可能にし、不正確なコミットがあればブロックデータは無効化され、予期されない。これは必須ではないが、多くの場合、親チェーンでの手数料を多く得るために子チェーンでの多くの計算処理を好む。システムのアップグレードに関して、同じトークンを受け入れ、移行期間をアナウンスする別のコントラクトを作る（それかコミュニティが網羅的に非中央集権システムとして決定する）ことでシステムをアップグレードすることは可能である。これは自走するシステムを作るだろう。誰か

がクラウド上でデータを保存したり計算処理をするサイトを運営してお金を払わないといけない一方で、今や不正証明付きでトークンや十分な数の参加者の手数料や継続的に正しく計算処理を続けるオペレーションをするstakerたちや、真に不定形のクラウド上で計算処理をすることなど、ひとまとまりのスマートコントラクトが設計できた。

これらの不正証明と最終的にルートチェーンで保持される供託はネイティブトークン（例：ETH）か、それとは別のPlasmaチェーンのコンセンサスルールを維持するトークンになりうる。ネイティブチェーンを使うのが明らかにシンプルだが、しかし興味深い経済的セキュリティへの示唆もある。もし目的がチェーンの停止と不正な振る舞いを防ぐことなら、ブロックチェーンアプリケーションによって、ETHだけを使うと不十分なインセンティブになっているかもしれない。トークンの価値はPlasmaチェーンが止まったりビザンチンな振る舞いを受けると下がる。加えてトークン価値はおおよそ将来にわたってのトランザクション手数料の総現在価値(NPV)と等しくなる。もし誰かがETHでstakeした場合、そのstakingの価値は手数料と供託の時間価値の相対的なものになる。この供託されている額がトークンの総現在割引価値よりも低くなることは想定できる。加えて、チェーンが停止したりblock withholdingの証明が難しいか消極的なとき、もし供託されたETHがstaking期間のあとに戻ったら、非ビザンチンに振る舞うインセンティブが不足するが、トークンを用いればその状況でビザンチンな振る舞いは起きない。

9　ブロックチェーンのためのMapReduce

MapReduce上で計算できるほぼ全てのものがこのチェーン上でも同様に計算可能でなければならない。これは我々のブロックチェーンにおける計算とプログラミングの考え方に重大なパラダイムシフトを要する。これはMapReduceではあるが、不正証明がともにある。それぞれのノードはブロックチェーンを表す。これは前のセクションで述べられたように、Plasmaチェーンの構造と高度に互換性がある。例えば、もしユーザーが単純なワードカウントをやりたいとして、あなたはreduce関数を実行するチェーンのマークルツリーを作れる。もし不正の証拠がそこにあれば、そのノード（チェーン）はペナルティを受ける。もし加算のreduce関数をあなたが生成できるなら、あなたは平均もまた生成できる。例えば平均価格などだ。Map関数はただ計算処理をここのチェーンに送って、結果を返すだけだ。明らかにデータスループットの制約があり、それがreduceの不正証明が必要な理由だ。任意の計算処理、というのは不可能で、しかし多くの種類の問題を解くことができる。たいてい、ソートアルゴリズムのようなメモリ使用量に制約をかけられる問題は解決できてる。それはPlasmaチェーン間のトラフィックとトレードオフだ。

もしノードが計算を証明する実際のブロックを生成できなかったとき、彼らの結果は廃棄され、ロールバックする。覚えておいてほしいのが、これはMapReduceが行うような計算のスケーラビリティを保証するわけではなく（なぜならコンセ

ンサスを維持するためにチェーンを監視しなければいけないから)、しかし、活動の強制とアクターたちのための拡張性を得る。結果として、はじめにわかる限界は、その特定の計算に関わる当事者たちは、その一連の計算を監視しなければならないという事実だ。もしその当事者が小規模な領域だけを監視していればいいのであれば、それはそれでよいのだが、しかしその当事者が計算に関わる全体を監視しなければならないとき、スケーラビリティのメリットは得られない（scalable assuranceしか得られない)。それはつまり、この方法で多くの問題が解決される。例えばDEX(Decentralized EXchange/もし他の皆の処理が強制可能であなたが詳細を気にしないなら、あなたのmapされた計算の集合はあなたのトレードに関するもの)などだ。

ブロックのフォーマットはTrueBitの設計にある計算可能なものと互換可能なものでないといけない。そこには状態へのコミット（UTXOと状態遷移の証明を可能にする状態木を構築できるように)、アカウントの木（子チェーンと複雑な状態遷移のために)、手数料のコミット（状態遷移をコミットするための手数料)、マークル化されたトランザクション、親/子チェーンから渡されたデータへのコミット、観測された親/子のブロック（再整列の防止のため)、そして様々なビジネスロジック（例：ワードカウントの例はマークル化されてソートされた単語がどこで見つかったかのコミット）がある。マークルコミットを構築することで、不正な状態遷移を証明できるルート/親チェーンで証明可能なスマートコントラクトを作

成できる。このフォーマットと互換可能でないいくつかの問題セットがあるが、過剰なメモリ要件なしの一般的な計算は可能である。不正証明に許される最大のデータと同値な形で、計算のための最大メモリサイズが扱われるメンタルモデルを持つとよい。

一連のmapとreduceの関数は、それによって処理する義務のあるような形でブロックチェーンを操作することを可能にする。これは親と子のチェーンに処理の義務を作るように要求する。子は親の渡したものをデータに含まなければならず、さもなくばチェーンは停止する。親チェーンは子に計算を強制でき、もし子が停止すればその矯正は親のチェーンで発生し、その場で証明を検証する。最初のTrueBitの設計での脅威はチェーンの停止に関わるもので、もし子チェーンが止まってもオペレーションが継続できるように設計されねばらなず、けれどもそれは多大な、特に時間に関する複雑さを要求することとなる（データセットが変更可能で、かつ時系列的生合成を持つと、特定の問題を片付けるのに大変骨が折れる）。子チェーンとのmapとreduceのフレームワークでブロックチェーンの計算を設計することによって、既存の計算機科学の研究を利用し、直接ブロックチェーンの分散システムの問題に適用することが可能になる。多様なビジネスアプリケーションを生み出すSolidityコントラクトをスケーラブルなやり方で実装することも可能である。そして個々のチェーンはただ計算し、活動を検証するだけでそれが作られるのだ。

10　アプリケーションの例

分散アプリケーションはトークンによって正当な活動が経済的にインセンティブ付けされたMapReduceの問題に置き換えることができる。

10.1　ブロックチェーン上でのRedditのクローン

これは主にデータのストレージ（CRUD）に関してである。

主に計算と証明作業はコントロール、認証（投票や投稿)、緩和などに用いられる。

多くのwebアプリケーションは実はバックエンドでCRUDを動かしているだけである。

ルートブロックチェーンはスマートコントラクトのルールとfraud proofsを含む。

親チェーンの上層には、サブredditのアカウントが含まれる。

各サブredditはPlasmaブロックチェーンであり、親チェーンの子である。

そのサブredditの中に、投稿のPlasmaチェーンがある。

投稿の子チェーンはコメントも同様に含む。

コンセンサスメカニズムはアクセス権限を強化する。

1つ前のブロックのデータのランダムなコミット（親チェーンによって提供されるランダムなノンスも）は次のブロックのヘッダーにコミットされる。

定期的な削減機能は、１番上の投稿と他の統計によって計

算される。各々のユーザーのコンピューターはデータとソフトウェアをローカル環境にダウンロードする。

データの提出にはデータの包含をインセンティブ付けするためにトランザクション手数料がかかる。データの可用性によっては古いデータをダウンロードすることにも手数料がかかるかもしれない。

特定の投稿を見るために、ユーザーはルートチェーンのコミットメントを検証する。そして、

Connects to a DHT network to discover nodes on the subreddit, downloads the chaintip (plus back n blocks to verify) of the subreddit and view list of posts, download the state trie as a light client and raw data for the relevant post with comments.

Users only need to observe parts of the Plasma chains which are relevant only to themselves (download only the posts and 40 subreddits relevant to themselves).

これはブロックチェーン上で計算処理を行うデータストレージの簡単な例である。

検証者が全てのノードを検証することは可能である。しかし、それを分割することも可能である。

シャーディングが上手く行われない場合、情報の可用性が心配される。これを緩和する方法の1つは、子チェーンの全権限をサブreddditのオーナーに加えることだ。

10.2　DEX 分散取引所

ブロックチェーン上でのredditのクローンは、CRUDウェブアプリケーションへの影響が明確だが、site statisticsを除きMapReduceオペレーションの恩恵を多く受けるわけではない。

分散取引所は、容量の多い取引を高速にすることを可能にする。

沢山のstatesがあるので、アウトプット上限はUTXOの代わりにアカウントで決まる。もしくは、state machineの各々のステップでより大きなbitmap が単体のbitmapの中のbooleanの代わりに各々のstateを代表する為に使われる。

サブredditに似ていて、取引相手を示しめしている子チェーンの木構造が有る。

それぞれのなかに、スケーラビリティーを最大化する為に、木構造のチェーンがある。（低活動のペアにとってはただ1つのPlasmaチェーンかもしれないが、高活動のチェーンにとってはもっと沢山の子チェーンをもっているかもしれない。）

これらのチェーンの各々は、アクティビティーと結びついており、ラウンド毎の取引できる量は事前に決定された量によって制限をかけることができる。

最初のステップは子チェーンに残高をもつことである。それで、決済に用いるPlasmaチェーンのようなものになる。

次に子チェーンの中にオーダーが発行される。

親チェーンへのコミットの一部として、全ての取引の順序

は順序付けされたメークル木構造化され、そのチェーン自体が1つの順序を表しているように整理される。

このステップは、最も高次の親Plasmaチェーンに到達するまで行われ、順序が整理されることで扱うチェーンの数を削減する。

順番が受け取られたら、受付を終了しbatch fashionで取引が実行される。

この削減ステップが完了し、ルートチェーンにコミットされたら後、各々のチェーンはmap ステップを通じて自分の配置場所を知ることになる。

親チェーンは子チェーンの順序の配置を既定された情報として通告する。

与えられた子チェーンは他のチェーンの順序も見ることができるなら（このステップで親チェーンをみることができるということを示唆している。）子チェーンは、このmapステップで正しく配置が実行されたことを証明することができる。

配置を受け取った後、そのmapステップはそのチェーンの子チェーンに同じ作業を行う。

これが完了すると、全てのfundのアップデートをブロックにコミットし、ブロックヘッダーを親チェーンに提出することにより最後の削減ステップが行われる。

高額が動く際に、複数のMapReduceを許容することでさらなる最適化という手法もある。しかし、この構造は一般的に極めて高量の時に可能となる。

このフレームワーク内の世界の全ての取引活動を実行する

ことも理論的には可能である。わからないです。

このタイプの構造は数々の金融活動や計算に役立つ。

10.3　分散メールシステム

DMailを創るために、Plasmaチェーンのアカウントを紐付け、メールを受け取る為に決済を行うようにすることもできる。

送信はユーザーの公開鍵で暗号化される。

知らない存在からのメールは支払いを行いわないと見ることができなかったり、zkSNARKsを使うことでさらなる最適化を履行することができる。

親チェーンはチェーンのディレクトリーを含み、決済を強制する。

シンプルなデザインである。

10.4　分散化CDN

分散化したCDNも可能である。

イーサリアムのShardingプロトコルと似ている。

各々の子チェーンをシャードとして扱う。

ランダムな指標を持つ。（ルートブロックチェーンなど）

nブロックごとに、シャード間でデータのシャッフルを行う。

親チェーンはシャッフルを実行する責任を持つ。

他のチェーンはアーカイブを保存する。

データの紛失は公開され、そのアーカイブを持っていた人

に報酬が入る。

1つのシャードだけが情報を公開し、他のシャードが公開しなければ1つのシャードだけが報酬を受けるのことになるので、情報を拡散するインセンティブがある。

この仕組みの頑丈さはその”流れ”の長さ依存している。頑強になればなるだけ、シャードはより多くのコピーをどの時点においても持つ必要がある。

ここでの鍵となる知見はストレージがa function of bandwidthということである。

データはディスクの中で固定されたものとして扱われない。

全てのデータは実際に動いていて次の目的地まで移動している。とてもTaoist的なアプローチである。

データをダウンロードするためには、親チェーンのシャードとランダムな指標を検証して、どのシャードが必要なデータをもっているのかを知る。そして、DHTを経由してそのシャードにアクセスし認証を経てデータをダウンロードする。

10.5　プライベートチェーン

参加者はチェーンのデータを他のチェーンに公開する責任を負わない。（公開することにデメリットはないが）そして、その結果、もし参加者がチェン上でルートチェーンによって履行されたプライベートブロックチェーンネットワークを持ちたければ、そうすることができる。

これは、イントラネットとインターネットの構造と類似している。取引はローカルのプライベートブロックチェーン上

で行われるが、取引のコミュニケーションとフィナンシャルアクティビティーはパブリックブロックチェーン上で行われる。

11　攻撃、リスク、そして緩和

よいスマートコントラクトのコーディングは難しい。ネットワークのセキュリティというのは、不正証明を行う処理に完全に依存してしまう。例えばある不正証明がブロックに含まれず、無効な状態遷移がルートチェーンに取り込まれてしまうことだって可能である。

トランザクションがメインチェーン閉じられるリスクがあるが、経済的に不可能である。少額ずつを足し合わせて大きな額にする、ある種のexit scamによって、発生しうる。これはexit条項を設けることで緩和される。全てのexitトランザクションをexit条項に基づいてソートし、チェーンからの離脱の前に一定の調停期間を経てからルートチェーンに移れるようにする。これはサードパーティーのウォッチャーに代わりに観察させることを可能にする。しかし、この設計は設計の複雑さを圧倒的に増させる。これらの事前署名済みの合体したトランザクションはチェーンのどの部分が失敗したかに依存して、他の親ネットワーク、あるいは最終的にルートチェーンまで伝播しうる。しかし、これは正しく振る舞えう「名前付きの当事者」に依存し、それが個々人が全てのUTXOをひとつあるいは少数のアウトプットとして合体させねばならない理由で、それによってexitトランザクションは経済的に可能と

なる。加えて、ユーザーはチェーンに価値の高いトークンに連動されたマイクロペイメントを残せ、その価格に乗っかることができる（Plasmaチェーンを完全に殺してしまうような圧倒的な逆インセンティブがあるので)。もし一般化された再帰的なSNARKs/STARKsが可能になれば、理論的にはwithhold blockであっても、認証済みexitを行う権限がないことを保証することが可能になる。

exitの異議申し立て期間は本質的にfinalityの推測を要する。もし依存するチェーンがfinalityを強制するするためのreorgをするための際立った供託コストを持つ場合、チェーン間の同期性を失わせるdeep chain reorgのリスクを劇的に減らすことができる。緩和施策の例としては、Ethereum本体で計画されているCASPER finality gadgetなどが挙げられる。

所定の緩和なしにGas代が過剰に高額になってしまった際、規定の時間以内にexitするのは不可能になるかもしれない。例えばもしGas代が50倍になるか、ルートチェーンにexitトランザクションのための十分なスペースがないのにマイナーがgas limitをあげないときなどだ。 exitをつつがなく処理するためにexit countingを停止することによるexitの遅延の延長などの緩和策がある。過去数ブロックにexitトランザクションがある限りexitを停止することで実現される。exitトランザクションが最近1つある条件で、制限時間内に全てのユーザーをexitさせることができる。もしexitが他者のexitよりも先に起きたとき、カウンターはリセットされる。その結果は、どのくらい資金が戻ってくるのは待たねばならないのかわからないので、

流動性供給者レートの支出を増加させる。もし平均gas costが一定以上だった場合、引き出し処理の「時計」をシンプルなメカニズムが止めることになる。（一定の妥当な上界値によってどのくらいの時間停止していられるかの閾を用意するほうがよい）

ユーザーが保持している資金は、少なくともひとつのPlasmaチェーンが高い確度で情報可用性を持っていることを保障しなければならない。（理想的には複数の独立した親チェーンによって）

12　今後の研究余地

以下ではセキュリティに関する補助的な将来の研究テーマに関して言及する。”Generalized Recursive SNARKs/STARKs” が現在の研究領域であり、これはexitトランザクションのセキュリティを圧倒的に向上させることができる。これはさらなる防衛力を期待することができ、したがって最終防衛ラインは異議申し立ての証明を可能にする非中央集権exitメカニズムになるだろう。第一防衛ラインは新規なcryptographyとセキュアなハードウェア装置になる。さらなるcryptographyのペア化や、準同型暗号（HE : Homomorphic Encryption）に関する開発も助けになるだろう。より大きな特異性が複数のルートチェーンを同期されるものが残っている間に一度に監視する能力に対して要求されます。（単純に難しい同期を強制することを超えて）さらなる研究ではfinalityや複数のチェーンに対するインタラクション

が必要とされる。同様にさらなるexitリスクの最小化も必要とされる（SNARKs/STARKsが助けになるはず）

13　結論と要約

Plasmaはデータを圧縮しながら情報の可用性を保証することに第一のフォーカスを置いて設計された（とくにblock withholding攻撃に対して)。我々が提案したメカに済みによって、ユーザーはルートチェーンによって状態を強制されたチェーンの中で資金を保持することを可能にする強制力のあるコミットを提出できる。これは圧倒的な計算とストレージを不定形の広大なコンピューターネットワークで可能にする。コミットを強制している経済的アクターによって活動は供託されていて、最終的に全ての親チェーンを横断して構成可能で、そして強制の流れは、強制している真実のスマートコントラクトを持つルートチェーンに降りていく。この設計はユーザーにルートチェーンではコスト感が合わないような状態遷移を可能にする。この設計によってひとつのブロックチェーンにもとづいて、世界中の全ての金融的な計算処理を行うことが可能である（多くのワーキングメモリを必要としない処理だけで組み上げる必要はある)。もし無効な計算を証明するものが提出された時だけ、それらのコミットはロールバックされる。ユーザーはcustodial trust(保管上の信用)を第三者であるチェーンのオペレーターに任せる必要がなくなる。チェーンを停止させたりビザンチンな行いをするインセンティブを減らすために、手数料がインセンティブの役目を担う。

チェーンの停止はそのPlasmaチェーンのためだけのトークンが存在する場合、抑止される。オペレーターがオペレーションを停止させようと言うモチベーションは、自身が稼ぎ出し供託しているトークンの価値を下げるために、経済的に圧倒的なインセンティブとなる。このインセンティブと構造は、トランザクション手数料により継続的に金銭的な対価のある自律的な非中央集権プログラムを作成することを可能にする。この自律的な非中央集権プログラムはそれ自身によってデータが処理され、バリデートされ、その構成員は絶えず更新され不定形な、真のクラウドコンピューティングを可能にする。Plasmaは過剰な負担や制限なしに多くのユーザーに一般的なアプリケーションを提供するようなスケーリングをブロックチェーンに可能にする。アプリケーションのクリエイターは単にスマートコントラクトのコードを書き、それをブロックチェーンに提出して、インセンティブシステムがそのPlasmaチェーンにユーザーが手数料を払い続ける限り、オペレーションを永続させる。

14　謝辞

TrueBitの作者たちにマークル化証明の設計と実装に関して、Christian Reitwießnerにレビューに関しての感謝の意を述べたい。そしてVlad Zanfirは多くのインスピレーションとアイデアの枠組みのおかげで、これらのアイデアを形式化するのに大変助けになった、ありがとう。そしてThomas Greco, Piotr DobaczewskiそしてPaweł Peregudはフィードバックと

コントリビューションに対する感謝を。

TODO: 他の人への感謝を述べる TODO: 参考文献を改善し、引用を増やす TODO: ダイアクラムを描き終える

参考文献

[1] Joseph Poon and Tadge Dryja. Lightning Network. https://lightning.network/ lightningnetworkpaper.pdf, Mar 2015. [2] Ethereum. Ethereum. https://ethereum.org. [3] Gavin Wood. ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER. http://gavwood.com/paper.pdf, Feb 2015. [4] Raiden. Raiden Network. https://raiden.network/. [5] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In OSDI, pages 137–150. USENIX Association, 2004. [6] Satoshi Nakamoto. Bitcoin: A Peertopeer Electronic Cash System. https://bitcoin.org/bitcoin.pdf, Oct 2008. [7] Nick Szabo. Formalizing and Securing Relationships on Public Networks. http://szabo.best.vwh.net/formalize.html, Sep 1997. [8] Fred Erhsam. Blockchain Tokens and the dawn of the Decentralized Business Model. https://blog.coinbase.com/ appcoinsandthedawnofthedecentralizedbusinessmodel8b8c951e734f.
[9] Naval Ravikant. The Bitcoin Model for Crowdfunding. https://startupboy.com/ 2014/03/09/thebitcoinmodelforcrowdfunding/. [10] Jason Teutsch and Christian Reitwiessner. A scalable verification solution for blockchains.

https://people.cs.uchicago.edu/ teutsch/papers/truebit.pdf, Mar 2017. 46 [11] Vitalik Buterin. Ethereum Sharding FAQ. https://github.com/ethereum/wiki/ wiki/Sharding-FAQ. [12] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timn, and Pieter Wuille. Enabling Blockchain Innovations with Pegged Sidechains. https://blockstream.com/sidechains.pdf, Oct 2014. [13] Paul Sztorc. Drivechain The Simple Two Way Peg. http://www.truthcoin.info/ blog/drivechain/. [14] Bitcoin Wiki. Merged mining specification. https://en.bitcoin.it/wiki/Merged_mining_specification. [15] Peter Todd. Tree Chains. https://github.com/petertodd/treechainspaper. [16] Eli BenSasson, Alessandro Chiesa, Eran Tromer, and Mardas Virza. Succinct NonInteractive Zero Knowledge for a von Neumann Architecture. https://eprint.iacr.org/2013/879.pdf, May 2015. [17] Alessandro Chiesa, Eran Tromer, and Madars Virza. Cluster Computing in Zero Knowledge. https://eprint.iacr.org/2015/377.pdf, Apr 2015. [18] Jae Kwon. Cosmos: A Network of Distributed Ledgers. https://github.com/cosmos/ cosmos/blob/master/WHITEPAPER.md, Sep 2016. [19] Gavin Wood. POLKADOT: VISION FOR A HETEROGENEOUS MULTICHAIN FRAMEWORK. https://github.com/w3f/polkadot-whitepaper/raw/master/ PolkaDotPaper.pdf, Nov 2016. [20]

Sergio Demian Lerner. lumino transaction compression protocol (ltcp). https: //uploads.strikinglycdn.com/files/9dcb08c5-f5a9430eb7ba6c35550a4e67/ LuminoTransactionCompressionProtocolLTCP.pdf, Feb 2017. [21] Ilja Gerhardt and Timo Hanke. Homomorphic Payment Addresses and the PaytoContract Protocol. http://arxiv.org/abs/1212.3257, Dec 2012. [22] Tier Nolan. Re: Alt chains and atomic transfers. https://bitcointalk.org/index. php?topic=193281.msg2224949#m