ループリング (LOOPRING): 分散型トークン取引プロトコル

Daniel Wang
daniel@loopring.org

Jay Zhou
jay@loopring.org

Alex Wang
alex@loopring.org

Matthew Finestone
matt.finestone@gmail.com

https://loopring.org

May 21, 2018

要 約

ループリングは、分散型取引所を構築するためのオープン型プロトコルである。一連の取引・決済機能を持つパブリックスマートコントラクトとして運用し、オフチェーン参加者グループにもオーダー集約・通信できる機能を持つ。ループリングプロトコルは無料且つ拡張可能であり、両替機能を組み込んだ分散型アプリケーション (*dApps*) 用の標準化構成要素としても機能する。この相互運用標準は、トラストレス (trustless= 信用不要) 匿名取引の実現を促進する。現在の分散型取引プロトコルより最も重要な改良は、オーダー間の整合性 (ミックスアンドマッチ) であり、異なる注文オーダーで、2 トークンの取引ペアの制約を取り除くと流動性を大幅に改善する。ほかにも、ループリングは、独自のロバストソリューションを用いて、フロントランニング行為を防ぐ: オリジナルソリューションの提出者より速くブロックにトランザクションを提出しようとする不正行為である。ループリングはブロックチェーンに依存せず、スマートコントラクト機能を備えたブロックチェーンで展開できる。執筆時点では、NEO[1] や、Ethereum [2] [3] や、Qtum [4] において動作検証済みである。

1 はじめに

ブロックチェーンに基づく資産の増加に伴い、取引相手との資産を交換するニーズも大きく増えた。従来型資産のトークン化を含む数千もの新しいトークンが導入され、ニーズはさらに拡大している。投機的なトークン交換か、交換したユーティリティトークンを介してネットワークへアクセスかにかかわらず、より大きなエコシステムにとって基礎となるのは暗号化資産間での交換能力である。実は、これらの資産に潜在的なエネルギー [5] が蓄えられている、エネルギーを解放するには、いわゆる資本解錠は、所有権を主張するのみならず、資産の自由譲渡や、変換が必要となる。

そのため、トラストレストークン (対価) の交換は、ブロックチェーン技術において魅力的な実例になる。しかし、現在までも暗号化通貨の愛好家が、殆どのトークン取引を伝統的な集中型取引所で行っていた。従って、ループリングプロトコルが必要とされるのは、ビットコイン [5] がずっと強調したように、P2P 電子キャシューにとって、「二重支払いを防ぐ為に、信用できるサードパーティー (第三者) が必要とされることは主な利点を失くしている」。同じように、信頼的な、ゲートのあるような中央集権型取引所で取引しなければいけないのは、分散型資産の主な利点を失った。

哲学観点からも、中央集権型取引所における分散型トークンの取引はつじつまが合わない、分散型プロジェクトの理念に反している。中央集権型取引所を利用する際に、多数の実用上のリスクと限界ついては、後ほど詳しく述べる。分散型取引所は (*DEXs*) [6] [7] [8] これらの問題を対処しようとしている。多くの場合、既存金融機関離れためのブロックチェーン技術を用いて、セキュリティリスクの軽減に成功した。しかし、DEX は新しい経済に重要なインフラストラクチャーになるにつれ、パフォーマンスの改善余地が多大にある。ループリングは、dApp の第三者に非依存的のオープンプロトコルを利用して、インフラストラクチャーためのモジュラー・ツールの提供を目指している。

2 取引所の現状

2.1 中央集権型取引所の不備

中央集権型取引所の主なリスクは 3 つ:1) 安全性の欠如、2) 透明性の欠如、3) 流動性の欠如。

安全性の欠如は、通常、ユーザーがプライベートキー (資金) の管理を 1 つ集中管理型の実体に委ねることによって生じる。中央集権型取引所にハッカーの不正侵入により、ユーザーが犠牲になる可能性は明らかである。我

々は、すべての中央集権型取引所がセキュリティとハッキングのリスクに直面していることについてよく知っているにもかかわらず [9] [10] 、トークン取引のための掛け金 (テーブルステークス) として受け入れられている。中央集権型取引所は、引き続きハッカーの「ハニーポット」として攻撃を受けているのは、サーバ上に数百万ドルのユーザー資金を保有しているためである。それに、取引所の開発者も、ユーザーの資金管理に、誠実かつ偶然に誤りを犯すかもしれない。簡単に言えば、ユーザーは一旦中央集権型取引所に入金したら、自分のトークン管理ができなくなる。

透明性の欠如は、ユーザーに取引所の不正行為による取引リスクをもたらす。ここでは、取引事業者が行った意図的な不正行為を指す。ユーザーは中央集権型取引所で取引しているのが個人の資産より *IOU* である。トークンは一旦取引所ウォレットに送られると、取引所が保管することになり、代わりに一個の *IOU* が提供される。これ以後、ユーザーのすべての取引は、実質的に *IOU* の間である。換金する際、取引所と自分の IOU を清算して、外部ウォレットアドレスにトークンを受け取る。プロセス全体について透明性が欠けており、取引所はシャットダウンしたり、アカウントを凍結したり、倒産などの可能がある。また、ユーザー資産を第三者に貸与するなど、保管しているユーザー資産をその他目的で使用することもあり得る。透明性の欠如は、ユーザーに資金全額がなくなる可能性はないが、取引手数料の増加、ピーク時の需要遅延、規制上のリスク、注文が先行されるなどの犠牲があるかもしれない。

流動性の欠如は、取引事業者から見れば、断片化された流動性は、新しい取引業者の参入を阻害しようとする、2 人の勝者が総取りのシナリオになる。まず、取引ペアが最も多い取引所は勝つ、それはユーザーがすべての取引を 1 つの取引所でしたがる。その次、オーダーブックが最も大きい取引所は勝つ、それは各トレードペアの好都合な売買スプレッドのためである。このような状況では、初期流動性の構築が難しくなるため、新規参入が困難になる。結果的には、ユーザーから苦情や重大ハッキングがあったにもかかわらず、多くの取引所が高いマーケットシェアを占めている。注意しなければいけないのは、中央集権型取引所は市場シェアが高ければ高いほど、より大きなハッキングの標的になる。

ユーザーから見れば、断片化した流動性はユーザー体験を大幅に低下させる。中央集権型取引所における取引では、ユーザーは取引所自身の流動性プール内で、オーダーブックと、サポートされているトークンペア間でしか取引できないトークン`A` をトークン`B` と交換するには、ユーザーは両方のトークンをサポートする取引所を選ぶか、異なる取引所で登録する、個人情報を開示して。また、売買スプレッドを支払って、BTC または ETH に対して一旦予備または仲介取引も行う必要がある。結局、ユーザーは、売買スプレッドを支払って、BTC または ETH に対して予備または仲介取引も行わなければならない。最後に、オーダーブックの深さが足りなかったら、品物を減らせなければ取引が完成できない可能性もある。たとえ取引所は自分のプロセス出来高が大きいと称しても、この出来高と流動性が偽りではないとの保証がない [11]。

その結果、流動性がばらばらになり、エコシステムが断片化され、従来の金融システムのように、わずかの取引所に大量の取引が集中する。グローバルな流動性を持つブロックチェーンには、中央集権型取引所は役に立たない。

2.2 分散型取引所の欠点

分散型取引所は中央集権型取引所と一つの違いとは、ユーザーが常にプライベートキー (資産) のコントロールによって、ブロックチェーン基盤で直接に取引する。暗号化通貨のトラストレステクノロジーを活用することで、上記のセキュリティリスクが和らげられる。しかしながら、性能及び構造上の限界に関しては問題が依然として存在する。

流動性がしばしば問題にされるのは、ユーザーは異なる流動性プールと基準を渡って取引相手を探さなければならない。DEXs または dApps が相互運用ための基準を統一しなければ、オーダーはネットワークの広い範囲に渡って共有/伝播されない、断片化した流動性は起こる。板 (リミットオーダーブック) の流動性、いわゆる弾力性:確定されたリミットオーダー (指値) どれほど速く再生されるのは、最適取引ストラテジーに著しく影響を及ぼす可能性がある [12]。このような基準欠如は、流動性の低下につながるのみならず、ユーザーを潜在的に不確かな専有スマートコントラクトに暴露される。

さらに、取引がオンチェーンで行われるため、DEX はブロックチェーン基盤の制約を受け継いでしまう、つまり:スケーラビリティ、実行遅延 (マイニング)、オーダー修正に高い費用がつく。結果的には、ブロックチェーンにおけるオーダーブックはうまく調整されず、ブロックチェーンでコード実行はコスト (ガス) がかかる、頻繁に複数の注文取消しは非常に高価になる。

最後に、ブロックチェーンのオーダーブックは一般公開されているため、オーダーを発するトランザクションは、次のブロックでマイニングされ、新しいオーダーブックに置かれるためマイナーに見えるようになっている。このようなリレーはユーザーにフロントランニングリスクと執行価格リスクをもたらす。

2.3 ハイブリッドソリューション

上記の理由から、純粋なブロックチェーンベースの取引所には制約があるため、中央集権型取引所と競合できない。オンチェーン固有のトラストレスと中央集権型取引所の取引スピード、オーダー柔軟性の間には妥協点が探せるはずである。ループリングと 0x [13] のようなプロトコルでは、オンチェーン決済のソリューションを拡張して、オフチェーンのオーダー管理と結合する。これらのソリューションは、オープン型スマートコントラクトを中心に展開され、複数の関数をオフチェーンで実行することによって、ノードに柔軟性を与え、ネットワークに重要な役割を果たせて、スケーラビリティの限界を突き止めたいが、ハイブリッドモデルにも同じく欠点が残っている [14]。

この論文を通じて、ループリングプロトコルは、ハイブリッドソリューションに関する建設的な意見を提案する。

3 ループリングプロトコル

ループリング自体は DEX ではない、複数のブロックチェーンに渡って、DEX を構築するプロトコルモジュールである。我々は従来の取引所の構成要素を分解し、パブリックスマートコントラクトと分散型参加者に代替させる。ネットワークでの役割は、ウォレット、リレー、流動性シェアリング・ブロックチェーン・コンソーシアム、オーダーブックブラウザ、リングマイナー、資産トークン化サービスを含む。それぞれの参加者を定義する前に、まずループリングオーダーを知る必要がある。

3.1 オーダーリング

我々はループリングオーダーを単一指向性オーダーモデル「UDOM」[15] で表す。ビットとアスクの代わりに、UDOM はオーダーをトークン取引リクエスト「$amountS/amountB$」(売/買数量) と表す。オーダーは単なる 2 つトークンの間での交換レードになる、ループリングプロトコルの強い特徴は環状トレードにおけるマルチオーダーのミックスとマッチングである。1 つ取引ペアの代わりに、16 個までのオーダーを利用し、流動性の大幅な向上および潜在的な価格改善が期待されている。

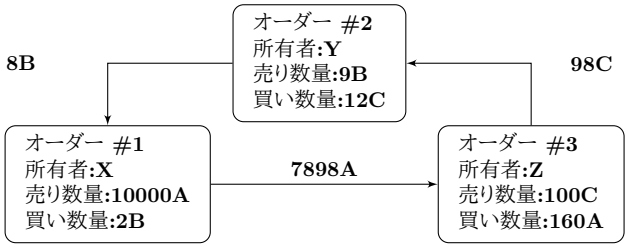

Figure 1: 3 つのオーダーでのオーダー・リング

上の図は 3 つオーダーで組み合わせたオーダーリングを示している。各オーダーの売りたいトークン (tokenS) は、ほかのオーダーの買いたいトークン (tokenB) になっている。各オーダーに反対オーダーを建つ必要とせず、欲しいトークンに交換できるループが生まれる。このようなオーダーリングにおいて、一般的な取引ペアも当然取引できるが、本質的にはオーダーリングのスペシャルケースである。

Definition 3.1 (オーダーリング) $C_0, C_1, \cdots, C_{n-1}$ は n 個の異なる種類のトークンとし、$O_{0\to1}, \cdots, O_{i\to i\oplus1}, \cdots, O_{n-1\to0}$ は n 個の注文とする。これらの注文は一つのオーダーリングを形成する:

$$O_{0\to1} \to \cdots \to O_{i\to i\oplus1} \to \cdots \to O_{n-1\to0},$$

n はオーダーリングの長さである場合、$i \oplus 1 \equiv i + 1 \mod n$ となる。

全構成要素のトランザクションが、ユーザーが暗黙に指定したオリジナルの交換レートと等しいか、より良いレートで執行されるとオーダーリングは有効になる。その有効性を検証するには、ループリングプロトコルのスマートコントラクトがリングマイナーからオーダーリングを受け取らなければならない、それに、オーダーのオリジナル交換レートの積は 1 以上になる必要だ。

アリスとボブがトークン A と B を取引したいと仮定してみよう。アリスはトークン A を 15 個所有し、それを使ってトークン B を 4 個に取替えしたい。ボブはトークン B を 10 個所有しトークン A を 30 個にしたい。

買い手と売り手はそれぞれ誰でしょうか?我々がどっちの資産を参照価格に想定しようによる。例えば、トークン A が参照価格にされた場合、アリスはトークン B を値段 $\frac{15}{4} = 3.75A$ で購入した、ボブは $\frac{30}{10} = 3.00A$ の値段でトークン B を 10 個販売した。トークン B を参考価格に想定する場合、アリスは $\frac{4}{15} = 0.26666667B$ の値段で 15 個のトークン A を販売し、ボブは $\frac{10}{30} = 0.33333334B$ の値段で 10 個のトークン A を購入したい。よって、だれが買い手や売り手は勝手だ。

1 番目のケースには、アリスはボブが自分のトークンに付けた売り価格 (3.00A) より高い価格の (3.75A) を支払いたい、2 番目のケースでは、ボブはアリスが自分トークンに付けた売り価格 (0.26666667B) より高い価格 (0.33333334B) を支払いたい。買い手は売り手が付けた値段以上を支払う意思があるならば、取引はいつでも可能のが明白だ。

$$\frac{\frac{15}{4}}{\frac{30}{10}} = \frac{\frac{10}{30}}{\frac{4}{15}} = \frac{15}{4} \cdot \frac{10}{30} = 1.25 > 1 \tag{1}$$

したがって、個注文集合の全部あるいは一部を満たすには、それぞれ買い注文の交換レートの積は 1 以上かどうかを知る必要がある。そうであれば、注文 n の一部或いは全部は満たされる [16]。

さらに第三相手のチャーリーが登場すると、アリスがトークンA の x_1 をトークンB の y_1 に、ボブはトークンB の x_2 をトークン C の y_2 に、そして、チャーリーはトークン C の x_3 を、トークンA の y_3 に、それぞれを取替えたい。

必要なトークンはそろっていれば、次の式を満たせば取引は可能になる:

$$\frac{x_1 \cdot x_2 \cdot x_3}{y_1 \cdot y_2 \cdot y_3} \geq 1 \tag{2}$$

ループリングオーダーの詳細について、7.1 で後述する。

4 エコシステムの参加者

次のエコシステムの参加者は、中央集権型取引所が提供すべき全機能を共同で提供する。

- ウォレット：普通のウォレットサービス或いはインターフェイスは、ユーザーにトークンへのアクセスとループリングネットワークへのオーダー送信を提供する。ウォレットはリングマイナーと手数料をシェアリングすることによって、オーダー生成に意欲を発揮する (詳細は 8を参照)。将来の取引がユーザー個人のウォレット内で安全に行われると信じ、プロトコルを通じて、これらの流動性 (資金) プールを結びつけるのが最も重要である。

- 流動性シェアリング・ブロックチェーン・コンソーシアム/リレーメッシュ：オーダーと流動性シェアリングためのリレーメッシュネットワークである。ノードがループリングリレーソフトウェアを実行すると、既存のネットワークに接続できる、コンソーシアムブロックチェーンを介して他のリレーと流動性をシェアリングできる。我々は構築している最初のコンソーシアムブロックチェーンでは、リアルタイム（1-2 秒ブロック）でオーダーシェアリングできるし、新たなノードでより速いダウンロードのために旧い履歴を削除する。とりわけ、リレーはコンソーシアムに参加する必要がない、彼らは単独で動けるし、他者と流動性をシェアリングしないし、或いは自身の流動性シェアリングネットワークをはじめ、管理する。

- リングマイナー：：リレーは、ウォレットやリレーメッシュからオーダーを受け取り、パブリックオーダーブックと取引履歴を維持し、任意でその他リレー (任意のオフチェーン媒体を介して) および/或いはリレーメッシュにオーダーを伝送するノードである。リングマイニングはリレーの特徴であって、要件ではない、計算が重く、完全にオフチェーンで行われる。我々はリングマイニング機能を持つリレーを「リングマイナー」(“Ring-Miners”) と呼ぶ、異なるオーダーをつなげ、オーダーリングを生成する。リレーは以下の三つにおいて自由である (1) その他リレーとコミュニケーションの取る方法、(2) オーダーブックの構築する方法、(3) オーダーリング (マイニングアルゴリズム) のマイニング方法。

- ループリングプロトコル・スマートコントラクト **(LPSC)**：公開且つ無料のスマートコントラクトのセットである、リングマイナーから受信したオーダーリングを検査し、ユーザーのためにトークンをトラストレス的に処理・転送し、リングマイナーとウォレットに手数料を割り当て、イベントを起す。リレー/オーダーブラウザは、これらのイベントに監視し、オーダーブックと取引履歴を最新の状態に保つ。詳細については、付録 **??**を参照してください。

- 資産トークン化サービス **(ATS)**：ループリングで直接に取引できない資産間のブリッジである、信頼できる企業や組織に運営されている中央集権型サービスである。ユーザーは、資産（実物、法定通貨、その他チェーンのトークン）を預託し、トークンを発行し、将来これらを預金に引き換えられる。ループリングは（適切な解決策が存在するまで）クロスチェンで取引プロトコルではないが、ATS は ERC20 トークンを実物資産やその他のロックチェーンの資産と取引出来るように [17] する。

5 交換プロセス

1. プロトコル認証：図 2 にように、トークンを交換したいユーザーY は、売りたいトークンB(amountS) の取り扱いを LPSC に認可し、処理できるように。これより、ユーザーのトークンはロックされず、オーダーがプロセス中にも自由に移動できる。

2. オーダー生成：トークンBvs トークンC の成り行きレートとオーダーブック (板) は、リレーやオーダーブックブラウザのようなその他ネットワーク接続仲介経由で提供された。ユーザーY は、任意の統合ウォレット・インターフェースを介して、amountS、amountB とその他パラメータを明記した上オーダー (指値オーダー) を発注する。一定数量の LRx が、リングマイナーの手数料として、オーダーに加えられる。LRx 料金は、高ければ、高いほどリングマイナーにより早く処理される。オーダーハッシュにはユーザーY のプライベートキーで署名される。

3. オーダーブロードキャスト：ウォレットはオーダーとその署名を 1 つまたは複数のリレーに送信してから、リレーはパブリックオーダーブックを更新する。このプロトコルでは、先着順などで特定方法で、オーダーブックを作成するように要求しない。それより、リレーはオーダーブックを作成する際に自由裁量の余地を持つ。

4. 流動性シェアリング：リレーは任意の通信媒体を経由して他のリレーにオーダーをブロードキャストする。また、ノード間はどのように相互作用や、相互作用するかどうかは柔軟性がある。安定しているネットワーク接続を促進するために、ブロックチェーンコンソーシアムに流動性シェアリングリレーメッシュを作り上げた。前のセクションで述べたように、このリレーメッシュは速度と包括性のため最適化されている。

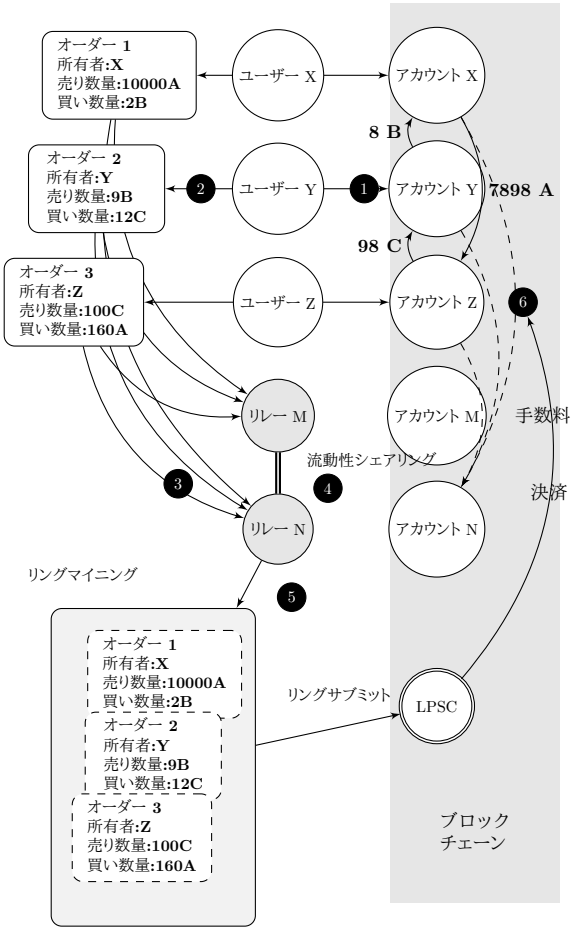

Figure 2: ループリング取引プロセス

5. リングマイニング(オーダーマッチング):リングマイナーは、オーダーをその他多数のオーダーをマッチングすることによって、指定されたまたはそれ以上の交換レートで、オーダーの全部または一部を満たせようとする。リングマイニングこそ、プロトコルがあらゆるペアに高い流動性を提供できる訳である。もし実行レートがユーザーY の指定より良いであれば、マージンはオーダーリング内に全オーダーでシェアリングされる。報酬として、リングマイナーは、マージンの一部 (マージン・スピリット、LRx をユーザーに返す) を請求するか、または単に LRx 料金を保留する。

6. 検証と決済:LPSC がオーダーリングを受信し、複数チェックを行う、リングマイナーから受け取ったデータを検証するためである。それに、オーダーリングが完全にまたは部分的に決済されるかどうかを決める (リング内オーダーの充足率とユーザーウォレット内のトークン残高による)。全てのチェックが合格であれば、コントラクト (LPSC) は自動的にトークンをユーザーに譲渡すると同時に、リングマイナーとウォレット料金の支払いを行う。もし LPSC はユーザー Y が残高不足と判明した場合は、縮尺が実施される;仮に十分な資金がアドレスに入金されたら、縮小されたオーダーは自動的に元サイズまで拡大される、キャンセルと違うのは、取り消しはリバース不可の片方向マニュアル操作である。

6 運用上の柔軟性

注意しなければならないのは、ループリングのオーペン基準は参加者に非常に大きな操作柔軟性を与える。参加者は、新たなビジネスモデルを無料で実装できるし、ユーザーに価値を提供すると同時にプロセスの出来高やその他基準によ x 料金を稼げる (選択した場合)。エコシステムはモジュール化されており、より多くのアプリケーションの参加をサポートする。

6.1 オーダーブック

リレーがオーダーブックを設計する際、好きな方法でユーザーオーダーを表示し、マッチングできる。最初に実装した独自のオーダーブックは OTC モデルで、価格のみに基づいて配置される。つまり、オーダーのタイムスタンプは、オーダーブックとは無関係だ。しかし、リレーは自分のオーダーブックを自由に設計できる、典型的な中央集権型取引のマッチングエンジンのように、タイムスタンプを重視しながら注文値段でランク付ける。リレーがこの種のオーダーブックを提供する傾向になったら、ウォレットを所有/統合して、オーダーをただ 1 つのリレーのみに送信する。そうなると、このリレーが時間順にオーダーをマッチングできる。このような構成は可能となる。

一方、ほかの DEX プロトコルでは、リレーに一定のリソースを要求する:買いオーダー発注用の初期トークン残高が求められる、その一方、ループリングリレーでは、取引成立にマッチング可能なオーダーを見つければ、初期トークンがなくても可能である。

6.2 流動性のシェアリング

リレーは、互いにどのように流動性 (オーダー) をシェアリングするのが自由に設計できる。我々のブロックチェーンコンソーシアムは、流動性シェアリングをやり解ける為の一つの解決策に過ぎず、エコシステムは好きなように、自由にコミュニケーションする。ブロックチェーンコンソーシアムに参加するほかに、自身の流動性シェアリングを構築して、管理する、自分に相応しいルールやインセンティブを決める。前述のように時間順にオーダー実行のウォレットのように、リレーは単独で動作することもできる。もちろん、ネットワーク効率を求めるために、他のリレーと通信するのは、明らかに有利であるが、異なるビジネスモデルは独自のシェアリングデザインもを評価し、多様な方法で手数料分割する。

7 プロトコルの仕様

7.1 オーダーの解析

オーダーというのは、ユーザーの取引の意図を表すデータの群れである。ループリングにおけるオーダーは、片方向オーダーモデル (the Uni-Directional Order Model) あるいは UDOM で次のように定義されている。

```
message Order {
 address protocol;
 address owner;
 address tokenS;
 address tokenB;
 uint256 amountS;
 uint256 amountB;
 unit256 lrcFee
 unit256 validSince; // システム時間
 unit256 validUntil; // システム時間
 uint8 marginSplitPercentage; // [1-100]
 bool buyNoMoreThanAmountB;
 uint256 walletId;  // デュアル     アドレス
 address authAddr;  // v, r, s は  名のパーツ
 uint8 v;
 bytes32 r;
 bytes32 s;
 // 中ある     プライベートキー
 // オーダーハッシュ  計算に使わない,
 // よって  名されてない.
 string authKey;
}
```

オーダーの発信元を確認するため、ユーザーのプライベートキーでパラメータのハッシュ値に対して署名する `authAddr` を除く。`authAddr` パラメータは、オーダーの一部であるオーダーリングに署名するために使われており、フロントランニングを防ぐために。詳細は 9.1を参照してください。フィールド `v`、`r`、と `s` は署名を表している、オーダーパラメータと共にネットワーク上に送信される。存続期間にオーダーは変更されないことが保証される。オーダーは決して変更されないが、プロトコルは他の変数やアドレス残高に基づいて現在の状態を算出する。

UDOM には価格 (本来であれば浮動小数点数でなければならない) が含まれていないが、代わりにレート (`rate`) や r を使って、`amountS`/`amountB` をあらわす。レートは浮動小数点数ではなく、式であり、必要に応じて他の符号なし整数で計算し、すべての中間結果を符号なし整数として維持し、計算精度を向上させる。

7.1.1 買い数量

リングマイナーがオーダーをリングマッチする時、より良いレートで実行される可能性があり、ユーザーが指定した買い数量 (`amountB`) より多いトークン `tokenB` を得られる。しかし、もし `buyNoMoreThanAmountB` が `True` に設定すると、ループリングプロトコルがユーザーにトークン `tokenB` を `amountB` 以上の受け取らないようにする。したがって、UDOM の `buyNoMoreThantokenB` パラメータは、全オーダーが満たされる時期を決める。`buyNoMoreThantokenB` は、売り数量 (`amountS`) または買い数量 (`amountB`) のいずれかに上限に適用し、ユーザーが従来の売買オーダーよりも細かく取引意向を設定できる。

たとえば、`amountS` $= 10$ で、`amountB` $= 2$ の場合、取引レート $r = 10/2 = 5$ となる。したがって、ユーザーは 1 つのトークン `B` に、5 つのトークン `S` を売りたい。リングマイナーは、ユーザーにレート 4 を見つけた、つまりユーザーはトークン `B` を、2 ではなく 2.5 を受け取れる。しかし、ユーザーがトークン `B` を 2 つしかほしくない場合は、`buyNoMoreThanAmountB` のフラグを `True` に設定すれば、LPSC は取引を倍率レート 4 で実行し、ユーザーにはトークン `B` ごとに、4 つのトークン `tokenS` を売る、つまりトークン `tokenS` を 2 つ節約できた。注意してほしいのは、これらにはマイニング手数料は考慮されていない (詳細は 8.1 を参照)。

実際、我々はオーダーを簡略化された形式でも表現できる。

```
Order(amountS,tokenS,
      amountB,tokenB,
      buyNoMoreThantokenB)
```

従来の取引所の ETH / USD マーケットには、一般な売買モデルでは、下記 1 番と 3 番でオーダーを表現できるが、その他 2 つオーダーはできない。

1. 価格 300 USD/ETH で 10 ETH を売る。このオーダーは、`Order(10, ETH, 3000, USD, False)` で表す。
2. 価格 300 USD/ETH で 10 ETH を売って、3000 USD を入手する。このオーダーは次のように表現できる:`Order(10, ETH, 3000, USD, True)`。
3. 価格 300 USD/ETH で 10 ETH を買い。このオーダーは、`Order(3000, USD, 10, ETH, True)` で表す。
4. 3000 米ドルを費やし、価格 300 USD/ETH でできるだけ多くの ETH を買う。この注文は、`Order(3000, USD, 10, ETH, False)` で表す。

7.2 リングの検証

ループリングスマートコントラクトは交換レートや取引数量に関するの計算をしないが、リングマイナーから提供されたこれらの値を受け取って、検証しなければならない。計算は、リングマイナーに実行される理由は主に以下の 2 つ:(1) スマートコントラクトのプログラミング言語は、例えば、Ethereum 上の solidity [18]、浮動小数点数演算をサポートしていない、特に $pow(x\ 1/n)$ (浮動小数点数の n 乗根を計算する)。(2) ブロックチェーン上の計算とコストを減らすため、計算プロセスはオフチェーンにするのが望ましい。

7.2.1 サブリングの検証

このステップは、アービトレイジャーがオーダーリング内で新規注文の実行による利益の不正取得。本来では、一旦リングマイナーが有効なオーダーリングを見つけると、オーダーリングに他のオーダーを追加してユーザーのマージン (取引倍率の割引) を全部奪おうとなる。以下図 3 に示しているように、$x1$, $y1$, $x2$, $y2$ を慎重に計算し、全オーダーレートの積をピッタリ 1 にすれば、レート割引がない。

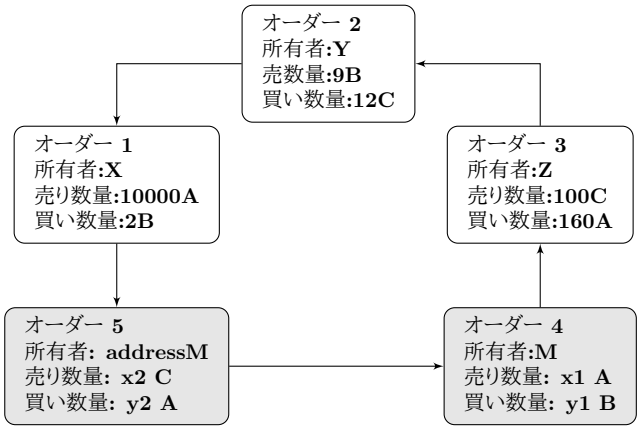

Figure 3: サブリングを含むオーダーリングの例

このようなやり方ではネットワークにとってゼロリスクだが、ゼロバリューでもあるので、リングマイナーによる不公平な行為とみなされる。防ぐために、ループリングは、有効なループにはサブリングが含まれない必要がある。このために、LPSC は 1 つのトークンが買いまたは売りポジションに 2 回とできないを確保する。上の図では、トークン `A` が売りトークン、買いトークンとして 2 回取引されたことがわかる、これは禁じられている。

7.2.2 約定レートチェック

オーダーリングにおける交換レートの計算は、リングマイナーに実行されなければならない理由について上記に説明した。すなわち、LPSC が正確さを検証しなければならない。まず、LPSC はリングマイナーが各オーダーに対して実行できる購入レートが、ユーザーが設定した元の購入レート以下であることを検証する。これより、ユーザーは少なくとも、取引中に設定したまたはより良い交換レートを得られる。次に、一旦交換レートが確定されると、LPSC はオーダーリング内の全てのオーダーを同じレート割引をシェアリングする。たとえば、レート割引が γ の場合、各注文の価格は次のようになる:

$r_{0\to1}\cdot(1-\gamma), r_{1\to2}\cdot(1-\gamma), r_{2\to0}\cdot(1-\gamma)$ 且つ、以下を満たす:

$$r_{0\to1}\cdot(1-\gamma)\cdot r_{1\to2}\cdot(1-\gamma)\cdot r_{2\to0}\cdot(1-\gamma)=1 \quad (3)$$

従って:

$$\gamma = 1-\frac{1}{\sqrt[3]{r_{0\to1}\cdot r_{1\to2}\cdot r_{2\to0}}} \quad (4)$$

全てのトランザクションに n 注文が含まれる場合、割引は以下のようになる:

$$\gamma = 1-\frac{1}{\sqrt[n]{\prod_{i=0}^{n-1} r^i}} \quad (5)$$

ここでは、r^i が第 i 個オーダーの回転率である。割引率 $\gamma \geq 0$ の時のみ、これらのオーダーが成立する、第 i 番目のオーダー $\left(O^i\right)$ の実際交換レートは $\hat{r^i} = r^i\cdot(1-\gamma), \hat{r^i} \leq r^i$

前回の例を思い出してみてください。アリスが 15 個のトークン `A` を持ち、それを 4 個のトークン `B` に交換したい、ボブはトークン `B` が 10 個持ち、それをトークン `A` に交換したい。もしトークン `A` が基準の場合、アリスは $\frac{15}{4}=3.75A$ でトークン `B` を購入し、ボブは $\frac{30}{10}=3.00A$ でトークン `B` を販売している。従って、割引の算出は: $\frac{150}{120}=1.25$ となる、$\frac{1}{1.25}=0.8=(1-\gamma)^2$、当事者双方にとって、平等な取引レートは $\sqrt{0.8}\cdot 3.75\approx 3.3541$ トークン `A`/ トークン `B`。

ボブは 4 個のトークン `B` を引き渡して、トークン A を 13.4164 個受け取る。期待していた「12」(4 つのトークン `B` ので) より多めにもらった。アリスは期待通り 4 つのトークン `B` を受け取った代わりに、13.4164 個のトークン `A` しか引き渡していない、予想していた「15」より少ない。注:この一部のマージンは、インセンティブとしてマイナー (とウォレット) に支払う。(詳細は 8.1 を参照)。

7.2.3 約定トラッキングと取消し

ユーザーは、LPSC に既に提出したオーダーの詳細や、キャンセルする数量などの情報を記載したスペシャルトランザクションを送信すれば、オーダーの一部或いは全部を取消しできる。LPSC はアカウントに入れ、取消し数量を記憶し、イベント `OrderCancelled` を発してネットワークに送信する。LPSC は、オーダーハッシュ値を識別子として利用し、値を格納して、執行された量と取り消された量を追跡する。このデータは公開アクセスできる、変更されるとイベント `OrderCancelled` / `OrderFilled` が発行される。オーダーリング決済ステップの間、これらの値の追跡は LPSC にとって重要だ。

LPSC は、イベント `OrdersCancelled` を持って、任意の取引ペアの全オーダー取消しや、イベント `AllOrdersCancelled` で特定のアドレスに全てのオーダーを取消すこともサポートする。

7.2.4 オーダースケーリング

オーダーは、執行された取引履歴や取消された量や、アカウント現在の残高に応じてスケーリング可能である。プロセスは、この特性に従って最小値のオーダーを探し出せ、オーダーリング中のすべての取引のスケーリング (拡大縮小) 標準として使用する。

最小注文値を見つけることは、各オーダーの取引数量の算出に役立つ。たとえば、第 i 番目のオーダーが最小値の場合、各オーダーにトークン売りの数 $\hat{s}$ とトークン買いの数 $\hat{b}$ は、次のように算出する。

$$\hat{s}^i = \overline{s}_i,\ \hat{b}^i = \hat{s}^i/\hat{r}^i,\ ;$$
$$\hat{s}^{i\oplus 1} = \hat{b}^i,\ \hat{b}^{i\oplus 1} = \hat{s}^{i\oplus 1}/\hat{r}^{i\oplus 1};$$
$$\hat{s}^{i\oplus 2} = \hat{b}^{i\oplus 1},\ \hat{b}^{i\oplus 2} = \hat{s}^{i\oplus 2}/\hat{r}^{i\oplus 2};$$
$$\ldots$$

ここでは $\overline{s}_i$ は一部取引が約定した後のアカウントの残高である。

実行中には、オーダーリング内のすべてのオーダーに最小値があると想定し、オーダーリング内で最大で 2 回計算を繰り返せば、各オーダーの執行ボリュームが得られる。

例えば、オリジナルオーダーと比較して、執行可能の最小量は 5% とすると、オーダーリング内全ての取引は 5% に縮小される。一旦このトランザクションが完了したら、この最小量と思われたオーダーも完全に終了していたはずだ。

7.3 リング決済

オーダーリングが前のチェックを全部終了した場合は、オーダーリングはクローズし、トランザクションが行われる。つまり、この n 個のオーダーが閉じられた環を形成した、下記の図で示したように。

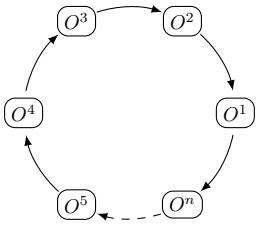

Figure 4: リング決済

トランザクションを完成させるために、LPSC はスマートコントラクト `TokenTransferDelegate` を利用する。このようなデリゲートの導入は、スマートコントラクトプロトコルのアップグレードが簡単になる、異なるバージョンのプロトコルを考慮する代わりに、オーダーがデリゲートを承認する。

オーダーリング内の各注文に対して、履行状況に応じて前後のオーダーに一定のトークン S が支払われる。リングマイナーへの手数料は、リングマイナーが選んだ料金モデルに応じて支払われる。最後に、一旦すべてのトランザクションが行われると、`RingMined` イベントが送り出される。

7.3.1 イベントの発生

ループリングプロトコルは、リレーや、オーダーブラウザや、ほかの参加者が可能な限りに効率的にオーダーブックのアップデートを受け取れるようにイベントを発す。発するイベントは下記通り:

- **OrderCancelled**:特定の注文がキャンセルされた際のイベント。
- **OrdersCancelled**:所有しているアドレスにある 1 つ取引ペアに関する全てのオーダーがキャンセルされた際のイベント。
- **AllOrdersCancelled**:所有しているアドレスにある全ての取引ペアの全てのオーダーがキャンセルされた際のイベント。
- **RingMined**:オーダーリングが正常に完了した際のイベント。このイベントには、リング内全トークンの譲渡データが含まれている。

8 トークン LRX

「LRx」はループリング系トークンの一般表記である。「LRx」は「Ethereum」におけるループリングトークンで、「Qtum」に「LRQ」、「NEO」に「LRN」などがある。今後、ループリングはほかのパブリックブロックチェーンに配備される際、その他タイプの「LRx」が導入される予定である。

8.1 手数料モデル

ユーザーがオーダーを引き起こす際、リングマイナーに支払う手数料として、一定金額の LRx を指定し、それにリングマイナーはオーダーによるマージン`marginSplitPercentage` 割合も主張できる、マージンスプリットと呼ばれる。どちらを選ぶのか (料金またはマージンスプリット)、リングマイナーに任せる。

マージンスプリットの説明は以下の通り:

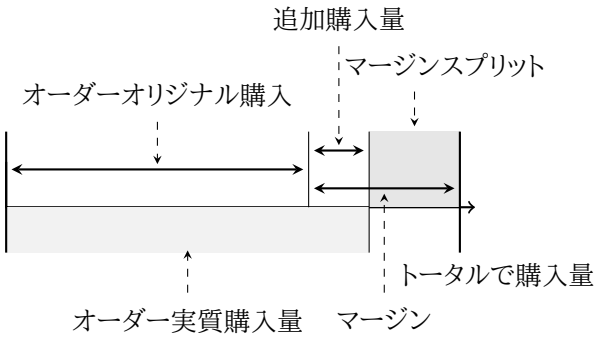

Figure 5: マージンスプリットが 60%の場合

オーダーリングのマージンが少なすぎる場合は、リングマイナーが LRx 料金を選ぶ。逆に、マージンが LRx 料金よりもはるかに上回る場合、リングマイナーはマージンスプリットを選ぶ。ただし、リングマイナーがマージンスプリットを選んだ場合、ユーザが支払うべき LRx と等しい金額をユーザ (オーダー作成者) に支払う必要がある。従って、リングマイナーがマージンスプリットを選ぶ入り口はオーダーの 2 倍 LRx 料金から始まる、LRx 料金を選択するの傾向が高まる。これにより、リングマイナー

は、高マージンのオーダーリングでより低収入に妥協して、低マージンのオーダーリングで安定の収入を得る。このような手数料モデルは、今後マーケットの成長と成熟につれ、高マージンのオーダーリングが減っていき、従って、安定な LRx 手数料がインセンティブとして必要されるとの予想に基づいてある。

最終モデルグラフは以下のように:

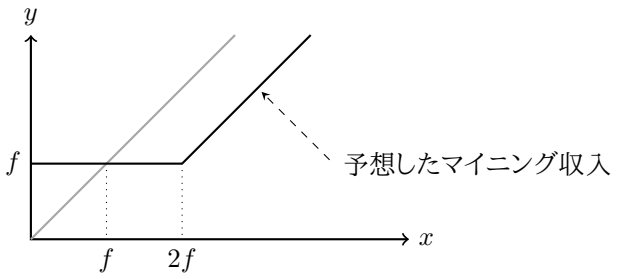

Figure 6: ループリングの手数料モデル

ここで、f は LRx 料金、x はマージンスプリット、y はマイニング収入である。実線で示すように、$y = max(f, x - f)$ となる。もしオーダーの LRx 料金が 0 の場合、$y = max(0, x - 0)$ ば灰色の実線で示されるように y = x。

結果は以下の通りとなる:

1. マージンスプリットが 0 の場合は、リングマイナーはフラットな LRx 料金を選び、インセンティブを受ける。

2. LRx 料金が 0 の場合、灰色の線が結果となり、収入は一般的な線形モデルになる。

3. マージンスプリットが 2 倍 (LRx 手数料) を超える場合、リングマイナーはマージンスプリットを選び、ユーザーに LRx を支払う。

注意すべきなのは、LRx 手数料はゼロじゃない場合、リングマイナーはどんな選択肢しても、リングマイナーとオーダー引き起こす者の間は LRx 転送が発生する。リングマイナーは LRx 手数料を徴収するか、オーダー発する者に LRx を返還して、マージンスプリットをもらう。

リングマイナーはウォレットと一定の割合の手数料をシェアリングする。ユーザーがウォレットからオーダーし、約定した場合は、ウォレットにも手数料の一部またマージンスプリットをもらう。これはモジュール化されているが、必要に応じて独自のビジネスモデルや実装が可能である、私たちではウォレットに手数料の約 20% ～ 25% を支払う傾向と考えている。ウォレットは、ループリングプロトコルのインテグレイションの主要ターゲットであるが、ユーザー基盤が持っているものの収入源はほとんどゼロである。

8.2 分散型ガバナンス

そもそもループリングプロトコルが、メンバー間のコーディネーションに基づいて、より効率的に目標実現ためのソーシャルプロトコルである。暗号化エコノミックのプロトコルとは大した差異がないが、実際には、ループリングプロトコルの有用性は、コーディネーション問題と同じメカニズムや、厳格な誘引均衡や、限定合理性に保護されている。このため、LRx トークンは費用の支払いのみならず、さまざまなネットワーク参加者への奨励金の調整にも使用されている。このような協調は、あらゆるプロトコルの幅広い採択に必要だが、交換プロトコルにとって特に重要なのは、健全な分散型エコシステムにおける流動性の改善にかかっている。

メンバー間の協調による効率的な目標達成という意味では、ループリングプロトコルは根本的にソーシャルプロトコルである。この点は、一般の暗号通貨界隈のプロトコルと異なるわけではなく、実際にその有用性は調整問題 [19]、グリムトリガー均衡、限定合理性等と同様なメカニズムによって保護されている。つまり、LRx トークンは手数料として支払うためのものだけでなく、多数のネットワーク参加者の報酬金を調整するためにも使用される。このような調整機能はどのプロトコルが幅広く採用されるために必要であるが、取引所は分散型エコシステムにおける流動性の向上によって成否が決定されるため、こうした調整機能は取引所のプロトコルにとって特に重要である。

LRx トークンが、分散型ガバナンスを通じてプロトコルの更新に使われる。将来、継続性と安全性を確保しながら、非互換性によって流動性のリスクを軽減するために、スマートコントラクトの更新はトークン所有者に管理される。スマートコントラクトは、一度導入されると変更できないのを考慮したら、dApps またはエンドユーザーは非推奨バージョンでやりとり続くと、更新されたコントラクトに排除されるリスクが起こり得る。プロトコルの成功にはアップグレードビィティが欠かせない、マーケットの需要とブロックチェーン基盤にも適応しなければならない。LRx 所有人に管理されている分散型ガバナンスは、過度にスマートコントラクトの抽出に頼らず、dApp やエンドユーザーにも影響せず、スマートコントラクトのアップデートが可能になる。LRx トークンの発行枚数は固定されており、LRC の場合では、一定割合はループリング財団に所有しており、コミュニティの発展に一部の資金を割り当てられる [21]。

しかし、LRx のトークン所有者は、プロトコルの方向性を調整する際に、考慮すべき唯一の利害関係者ではない:リレー/リングマイナーや、ウォレットや、開発者などはエコシステムにとって不可欠な存在あり、彼らの意見を聞かなければいけない。実際、彼らは、それぞれの役割を果たすに LRx を所有しなければならない立場ではないので (従来のメーカー/テイカーとマーケットメーカーは存在しないため、初期のトークン準備は必須ではない) 。彼らの利益を大事にする代替方法を認めなければならない。さらに、「単純な」トークンベースの (オンチェーン、オフ・チェーンともに) 投票は、低い投票率とトークン所有権の集中がリスクが発生する可能性があって、相違が生じるため不完全である。したがって、ゴールのレイヤを埋め込んだガバナンスモデルの実装は、シェアリング知識に基づいた一般的な意思決定プロセスになる。色々

な参加者から寄せたシグナルを提供する協力機構や、もしかして事前に確立したプロトコルの焦点などで促進される。これを実現するにつれて、ループリング財団はプロトコル開発者からプロトコル管理者にに進化するでしょう。

9 詐欺と攻撃からの保護

9.1 フロントランニングの防止

分散型取引所では、フロントランニングとは、ほかのノードの取引ソリューションをコピーし、ペンディングのトランザクションプール (mempool) にあるオリジナルのトランザクションよりも前にマイニングされる。これは、より高い取引手数料 (ガス価格) を指定するによって達成可能になる。ループリング (オーダーマッチングためのプロトコル) でのフロントランニングの主な不正はオーダー盗みになる。フロントランナーがペンディングのオーダーリングの決済トランザクションから 1 つまたは複数の注文を盗むとき。ループリングに特有なのは、フロントランナーがペンディングのトランザクションからオーダーリング全体を盗む。

サブミットリングがまだ承認されずにペンディングトランザクションプールに残っている時、誰でも簡単にこの取引を見つけ、マイナーアドレス (`minerAddress`) に自分のアドレス (`filcherAddress`) を置き換えられる、次にペイロードに再署名すれば、オーダーリングの署名は `filcherAddress` に置き換えられる。ぱっくり者は、次のブロックでブロックマイナーに、オリジナルのサブミットトランザクションの代わりに、自分の新しいトランザクションを拾えるために、より高い値段のガスを設定する。

この問題に対して、これまでの解決策には、重要な欠点があった:より多くのトランザクションとリングマイナーにより多いガスがかかる;1 個のオーダーリングを解決するのには少なくとも 2 倍ブロック時間がかかる。我々の新しいソリューションのデュアルオーサリング [20] では、オーダーに 2 重認証の設定メカニズムが含まれている一決済用の 1 つと、リングマイニング用のもう 1 つである。

1. 各オーダーのために、ウォレット・ソフトウェアはランダムなパブリックキー/プライベートキーのペアを生成し、そのキーペアをオーダーの JSON スニペットに入れる。(代案は、パブリックキーの代わりにパブリックキーから派生したアドレスを使用する、バイト数を減らすためである。我々は「`authAddr`」を使ってこのアドレスを表す、「`authKey`」は「`authAddr`」のマッチングプライベートキーを示す) 。

2. `r`, `v`, `s`, 及び `authKey` を除いて、以外すべてのフィールドで、オーダーのハッシュ値を算出し、所有者のプライベートキー ($authKey$) を使ってこのハッシュ値に署名する。

3. ウォレットは `authKey` をオーダーと一緒にリングマイニング用のリレーに送る。リングマイナーは、`authKey` と `authAddr` が正しくペアされているか、それにオーダーの署名が所有者アドレスに対して有効であるのを検証する。

4. オーダーリングが判明されると、リングマイナーは各オーダーの `authKey` を使って、リングのハッシュ値、`minerAddress`、全てのマイニングパラメータに署名する。オーダーリングに n 個のオーダーが含まれている場合、n 個の `authKey` に n 個署名が生成される。これらの署名を `authSignatures` と呼ぶ。リングマイナーは、`minerAddress` のプライベートキーを使って、すべてのマイニングパラメータとリングのハッシュ値にも署名する。

5. リングマイナーは、全てのパラメータとエキストラ `authSignatures` を使って関数 submitRing を呼び出す。注意してほしいのは、`authKey` はオンチェーントランザクションの一部ではないため、リングマイナー以外の当事者は知らない。

6. ループリングプロトコルは各オーダーの `authAddr` に対して \verbauthSignature| を検証する、`authSignature` が紛失や、無効の場合はオーダーリングを拒否する。

従って、結果は次のようになる:

- オーダーの署名 (所有者アドレスのプライベートキーによる) は、`authAddr` とオーダーが変更されないと保証する。
- リングマイナーの署名 (`minerAddress` のプライベートキーによる) が提供すれば、誰にも自分のアイデンティティを使ってオーダーリングをマイニングできない。
- `authSignaturess` は、`minerAddress` を含むオーダーリング全体を変更不能とオーダーの横取りされないことを保証する。

デュアル認証は、リングパックリとオーダーパックリを防ぎながら、一回のトランザクションでオーダーリングの決済を確実に行われるのを保証する。さらに、デュアル認証は、2 つの方法でリレーのオーダーシェアリングする:非適合 (nonmatchable) シェアリングと可適合 (matchable) シェアリング。デフォルトでは、ループリング操作は OTC モデルになっている、指値 (limit price) のみをサポートする。つまり、オーダーのタイムスタンプは無視される。要するに、フロントランニングは取引が実行されるかどうかに影響を及ぼすが、実際の取引価格に影響を与えない。

10 その他の攻撃

10.1 Sybil 攻撃または DOS 攻撃

悪意のあるユーザー(自分または偽造者としての役割を果たす)は、大量の小規模なオーダーを送信するによっ

てループリングノードを攻撃できる。しかし、我々はノードが自分の基準に基づいてオーダーを拒否できるようにしているので、基準は隠す或いは開示も可能。これらのオーダーはマッチングされても、満足できる利益がないため拒否されるのがほとんどである。我々はリレーにオーダーの管理に権限を与えて、このような大規模な小さなオーダー攻撃は脅威ではないと認識している。

10.2 残高不足

悪意のユーザーは、オーダーバリューはゼロではないが、実際のアドレスの残高がゼロのオーダーに署名し、広げることが出来る。ノードは、ゼロ残高のオーダー注意を払って、モニタリングする、変化に応じてオーダー状況を更新し、破棄することができる。ノードはオーダー状況更新に時間を費やさなければいけないが、アドレスをブラックリストに載せたり、関連オーダーを落としたりなど、影響を最小限に抑えることも可能である。

11 サマリー

ループリングプロトコルは、分散型取引の基盤層となるのを目指している。その際、人々が資産や価値をどのように交換するのかに深く反響を及ぼす。中間財の金銭は、物々交換を促進や代替する、「欲望の二重の一致」問題 [21] を解決し、つまり、2 つの取引相手がお互いの明確な財産やサービスを求めなければならない。同様に、ループリングプロトコルは、取引をより簡単に完結させるためにリングマッチングを利用するによって、取引ペアへの依存欲求をなくすことを目指している。社会や市場がトークンや伝統的な資産などをどのように交換することに意義がある。実際に、分散型暗号化通貨が、国家の支配権に脅威を与えるように、取引者 (消費者/生産者) を大規模に組み合わせるプロトコルは、理論上では通貨のコンセプト自体に対するの脅威である。

ループリングプロトコル利益は以下のようにまとめる:

- オフチェーンのオーダー管理とオンチェーンの決済とは、セキュリティのためにパフォーマンスに犠牲することなく。
- リングマイニングとオーダーシェアリングによる流動性の向上。
- デュアル認証は、現在のすべての DEX とそのユーザーが直面している悪質なフロントランニング問題を解決する。
- 無料のパブリックスマートコントラクトによって、dApp はループリングプロトコルで構築または相互操作が出来る。
- 事業者間の標準化により、ネットワーク効率とエンドユーザーエクスペリエンスの向上を実現する。
- オーダーブックの管理とコミュニケーションにはネットワークがより柔軟的である。
- 参入障壁の低減は、ノードにとってネットワークとエンドユーザーと結合するコストの低減。
- ユーザーウォレット間の匿名取引が実現できる。

12 謝辞

今回の論文に際して、各指導者、アドバイザ、及びループリングコミュニティに参加しているの方々に感謝の気持ちと御礼を申し上げたい。特に、中国分佈式總⊠基礎協議聯盟 (ChinaLedger) 白碩氏、闞海濱博士; Alex Cheng 氏、達鴻飛氏、曹寅氏、呉小川氏、Zhen Wang 氏、於偉氏、段念氏、肖軍氏、錢江氏、向江旭氏、郭一鵬氏、李大海氏、Kelvin Long 氏、夏華夏氏、馬俊氏、Encephalo Path 氏よりプロジェクトレビューやフィードバックをいただきまして、誠にありがとうございました。

References

[1] Viktor Atterlönn. A distributed ledger for gamification of pro-bono time, 2018.

[2] Vitalik Buterin. Ethereum: a next generation smart contract and decentralized application platform (2013). *URL {http://ethereum. org/ethereum. html}*, 2017.

[3] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.

[4] Patrick Dai, Neil Mahi, Jordan Earls, and Alex Norta. Smart-contract value-transfer protocols on a distributed mobile application platform. *URL: https:// qtum. org/ uploads/ files/ cf6d69348ca50dd985b60425ccf282f3. pdf*, 2017.

[5] Hernando de Soto. *The Mystery Of Capital*. Basic Books, 2000.

[6] Fabian Schuh and Daniel Larimer. Bitshares 2.0: Financial smart contract platform, 2015.

[7] Bancor protocol. *URL https://bancor.network/*, 2017.

[8] Yaron Velner Loi Luu. Kybernetwork: A trustless decentralized exchange and payment service. `https://kr.kyber.network/assets/KyberNetworkWhitepaper.pdf`, Accessed: 2018-03-05.

[9] Fortune. How to steal $500 million in cryptocurrency. `http://fortune.com/2018/01/31/coincheck-hack-how`, Accessed: 2018-03-30.

[10] Robert McMillan. The inside story of mt. gox, bitcoin's 460 dollar million disaster. 2014.

[11] Sylvain Ribes. Chasing fake volume: a crypto-plague. https://medium.com/@sylvainartplayribes/chasing-fake-volume-a-crypto-plague-ea1a3c1e0b5e, Accessed: 2018-03-10.

[12] Rossella Agliardi and Ramazan Gençay. Hedging through a limit order book with varying liquidity. 2014.

[13] Will Warren and Amir Bandeali. 0x: An open protocol for decentralized exchange on the ethereum blockchain, 2017.

[14] Iddo Bentov and Lorenz Breidenbach. The cost of decentralization. http://hackingdistributed.com/2017/08/13/cost-of-decent/, Accessed: 2018-03-05.

[15] Daniel Wang. Coinport's implemenation of udom. https://github.com/dong77/backcore/blob/master/coinex/coinex-backend/src/main/scala/com/coinport/coinex/markets/MarketManager.scala, Accessed: 2018-03-05.

[16] Supersymmetry. Remarks on loopring. https://docs.loopring.org/pdf/supersimmetry-loopring-remark.pdf, Accessed: 2018-03-05.

[17] Fabian Vogelsteller. Erc: Token standard. *URL https://github.com /ethereum /EIPs /issues /20*, 2015.

[18] Chris Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.

[19] Vitalik Buterin. Notes on blockchain governance. https://vitalik.ca/general/2017/12/17/voting.html, Accessed: 2018-03-05.

[20] Daniel Wang. Dual authoring —loopring' s solution to front-running. *URL https://medium.com/ loopring-protocol/ dual-authoring-looprings-solution-to-front-running-d0fc9c348ef1*, 2018.

[21] Nick Szabo. Menger on money: right and wrong. http://unenumerated.blogspot.ca/2006/06/menger-on-money-right-and-wrong.html, Accessed: 2018-03-05.