Loopring:
Un protocole décentralisé d'échange de jetons

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 9, 2018

Abstract

Loopring est un protocole ouvert permettant de créer des échanges décentralisés. Loopring gère des contrats intelligents publiques pour les échanges et les transactions, avec un groupe hors-chaîne d'acteurs qui agrège et communique les ordres. Le protocole est gratuit, extensible, et sert de standard pour créer les blocs pour les applications décentralisées (dApps) qui incorporent des fonctionnalités d'échange. Son standard inter-opérable facilite les échanges anonymes sans confiance. Une amélioration importante par rapport aux protocoles d'échanges décentralisés actuels est la possibilité de mélanger et d'apparier les ordres avec d'autres ordres dissemblables, évitant ainsi les contraintes liés à l'utilisation de jetons différents et améliorant considérablement la liquidité. Loopring utilise également une solution unique et robuste pour éviter le front-running : la tentative déloyale de soumettre des transactions dans un bloc plus rapidement que le fournisseur d'origine. Loopring est agnostique et peut être déployé sur n'importe quelle chaîne de blocs permettant les fonctionnalités de contrat intelligents. Au moment d'écrire ces lignes, il est utilisable sur Ethereum [1] [2] et Qtum [3] et est en construction pour NEO [4]

.

1 Introduction

Avec la prolifération des actifs basés sur la chaîne de blocs, la nécessité d'échanger ces actifs entre parties prenantes s'est considérablement accrue. Au fur et à mesure que des milliers de nouveaux jetons sont introduits - y compris pour les actifs traditionnels qui créent leurs propre jetons - ce besoin est amplifié. Qu'il s'agisse d'échanger des jetons pour des motivations commerciales spéculatives ou pour accéder au réseau via leurs jetons utilitaires natifs, la capacité d'échanger un actif cryptographique contre un autre est fondamentale pour l'ensemble de l'écosystème. En effet, il y a une énergie potentielle dans les actifs [5], et libérer cette énergie - débloquer le capital - exige non seulement d'affirmer la propriété, ce que les chaînes de blocs ont permis grâce à leur immutabilité, mais aussi la capacité de transférer et de transformer librement ces actifs.

En tant que tel, l'échange sans confiance de jetons (valeur) est un cas d'utilisation convaincant de la technologie des chaînes de blocs. Jusqu'à présent, cependant, les passionnés de cryptographie se sont largement contentés d'échanger des jetons sur des places d'échanges centralisées traditionnelles. Le Protocole Loopring est nécessaire parce que, tout comme Bitcoin [6] a mis en avant qu'en ce qui concerne la monnaie électronique de pair à pair, "les principaux avantages sont perdus si un tiers de confiance est toujours nécessaire pour éviter les doubles dépenses", il en va de même pour les principaux avantages des actifs décentralisés s'ils doivent passer par des échanges centralisés, et contrôlés. Échanger des jetons décentralisés sur des places d'échanges centralisés n'a pas de sens d'un point de vue philosophique, car il est alors impossible de maintenir les vertus que ces projets décentralisés épousent. Il existe également de nombreux risques et limites pratiques liés à l'utilisation des places d'échanges centralisées, qui sont décrits ci-dessous. Les échanges décentralisés (DEX) [7] [8] [8] [9] ont cherché à résoudre ces problèmes et, dans de nombreux cas, ont réussi à atténuer les risques pour la sécurité en utilisant des chaînes de blocs pour la désintermédiation. Toutefois, comme la capacité des DEX devient cruciale pour la nouvelle économie, il existe une marge de manœuvre importante pour améliorer leurs performances. Loopring vise à fournir des outils modulaires pour ladite infrastructure avec son protocole ouvert dApp agnostique.

2 État des lieux des places d'échange actuelles

2.1 Insuffisances des places d'échange centralisés

Les trois principaux risques liés aux places d'échange centralisés sont : 1) le manque de sécurité, 2) le manque de transparence et 3) le manque de liquidité.

Le manque de sécurité résulte du fait que les utilisateurs cèdent généralement le contrôle de leurs clés privées (fonds) à une entité centralisée. Cela expose les utilisateurs à la possibilité que les places d'échange centralisés soient la proie de pirates malveillants. Les risques de sécurité et de piratage auxquels sont confrontés toutes les places d'échange centralisées sont bien connus [10] [11], mais sont souvent acceptés en tant que "table des risques" pour les échanges de jetons. Les places d'échange centralisés continuent d'être alléchantes pour les pirates informatiques, car leurs serveurs ont la garde de millions de dollars de fonds d'utilisateurs. Les développeurs des ces places d'échange peuvent également faire des erreurs accidentelles en toute bonne foie avec les fonds des utilisateurs. clairement, les utilisateurs ne sont pas en contrôle de leurs propres jetons lorsqu'ils sont déposés auprès d'une place d'échange centralisée.

Le manque de transparence expose les utilisateurs au risque que des places d'échange malhonnêtes agissent de manière déloyale. La différence provient ici des intentions malveillantes de l'opérateur de la place d'échange, car les utilisateurs n'y négocient pas vraiment leurs propres actifs, mais plutôt une reconnaissance de dette. Lorsque les jetons sont envoyés dans le portefeuille de la place d'échange, elle en prend la garde et offre une reconnaissance de dette à sa place. Toutes les transactions se font alors entre les reconnaissances de dette des utilisateurs. Pour se retirer, les utilisateurs échangent leur reconnaissance de dette avec la place d'échange et reçoivent leurs jetons sur leur portefeuille externe. Tout au long de ce processus, il y a un manque de transparence, et la place d'échange peut fermer, geler votre compte, faire faillite, etc. Il est également possible qu'ils utilisent les actifs de l'utilisateur à d'autres fins pendant qu'ils en ont la garde, par exemple en les prêtant à des tiers. Le manque de transparence peut coûter aux utilisateurs, outre la perte totale de fonds, des frais de négociation plus élevés, des retards pour passer des ordres en période de pointe, des risques réglementaires et des ordres en front run.

Le manque de liquidité du point de vue des opérateurs des places d'échange, la fragmentation de la liquidité empêche l'entrée de concurrents en raison de leur situation dominante. Tout d'abord, la place d'échange avec le plus grand nombre de paires d'échange gagne, parce que les utilisateurs trouvent souhaitable d'effectuer toutes leurs transactions sur une seule place. Deuxièmement, la place d'échange avec le plus gros carnet d'ordres gagne, en raison des écarts favorables entre l'offre et la demande pour chaque paire. Cela décourage la concurrence de nouveaux arrivants parce qu'il leur est difficile d'accumuler des liquidités initiales. Par conséquent, de nombreuses places d'échange détiennent une part de marché élevée malgré les plaintes des utilisateurs et les incidents de piratage majeur. Il convient de noter qu'à mesure que les place d'échange centralisées gagnent des parts de marché, elles deviennent une cible de piratage de plus en plus importante.

Du point de vue des utilisateurs, la fragmentation de la liquidité réduit considérablement le confort d'utilisation. Dans une place d'échange centralisée, les utilisateurs ne peuvent négocier qu'au sein des pools de liquidité qu'ils possède, contre leurs propres carnets d'ordres et pour les jetons qu'ils prennent en charge. Pour échanger des jetons `A` contre des jetons `B`, les utilisateurs doivent se rendre sur une place d'échange qui prend en charge les deux jetons ou s'inscrire sur différentes place d'échange, en divulguant des informations personnelles. Les utilisateurs ont souvent besoin d'exécuter des transactions préliminaires ou intermédiaires, généralement contre BTC ou ETH, en payant les écarts entre l'offre et la demande dans le processus. Enfin, les carnets d'ordres peuvent ne pas être suffisamment important pour terminer la transaction correctement. Même si la place d'échange prétend traiter des volumes importants, il n'y a aucune garantie que ce volume et cette liquidité soient vrais[12].

Il en résulte des réserves de liquidité déconnectés et un écosystème fragmenté qui ressemble à l'ancien système financier, avec un important volume de transactions centralisées sur quelques places d'échanges. Les promesses de liquidité globale des chaînes de blocs sont nulles au sein des places d'échange centralisées.

2.2 Insuffisances des places d'échange décentralisés

Les places d'échange décentralisées diffèrent des centralisées en partie parce que les utilisateurs conservent le contrôle de leurs clés privées (actifs) en effectuant des transactions directement sur la chaîne de blocs sous-jacente. En tirant parti de la technologie sans confiance des crypto-monnaies elles-mêmes, ils réussissent à atténuer bon nombre des risques susmentionnés entourant la sécurité. Cependant, des problèmes persistent en ce qui concerne la performance et les limites structurelles.

La liquidité demeure souvent un problème car les utilisateurs doivent rechercher des parties contractantes parmi des pools de liquidité et des normes disparates. Des effets de liquidité fragmentés sont présents si les DEX ou les dApps en général n'utilisent pas de normes cohérentes pour l'interopérabilité, et si les ordres ne sont pas partagés/propagés sur un large réseau. La liquidité des carnets d'ordres à cours limité et, plus précisément, leur résilience - la rapidité avec laquelle les ordres à cours limité sont régénérés - peut avoir une incidence importante sur les stratégies de négociation optimales [13]. L'absence de telles normes a entraîné non seulement une réduction de la

liquidité, mais aussi une exposition à un éventail de contrats intelligents propriétaires potentiellement peu sûrs.

De plus, étant donné que les opérations sont effectuées en chaîne, les DEX héritent des limites de la chaîne de blocs sous-jacente, à savoir : changement d'échelle, les retards dans l'exécution (processus de minage) et les modifications coûteuses des ordres. Ainsi, les carnets de commandes d'une chaîne de blocs ne changent pas particulièrement bien d'échelle, car l'exécution du code sur la chaîne de blocs entraîne un coût (gaz), ce qui rend les cadences d'annulation de commandes multiples excessivement chères.

Enfin, parce que les carnets d'ordres de la chaîne de blocs sont publics, la transaction pour passer un ordre est visible par les mineurs étant donné qu'il est extrait dans le bloc suivant et placé dans un carnet d'ordres. Ce retard expose l'utilisateur au risque de front-run et de voir le prix ou l'exécution se retourner contre lui.

2.3 Solutions hybrides

Pour les raisons susmentionnées, les places d'échange purement fondés sur la chaîne de blocs ont des limites qui les rendent non concurrentiels par rapport aux places d'échange centralisés. Il y a un compromis entre l'absence de confiance inhérente à la chaîne de blocs et à la vitesse et à la flexibilité des places d'échange centralisées. Des protocoles tels que Loopring et 0x [14] étendent une solution de règlement en chaîne avec gestion des ordres hors chaîne. Ces solutions tournent autour de contrats intelligents ouverts, mais naviguent dans les limites de changement d'échelle en exécutant plusieurs fonctions hors chaîne et en donnant aux nœuds la flexibilité nécessaire pour remplir des rôles critiques pour le réseau. Toutefois, des inconvénients subsistent pour les modèles hybrides également [15]. A travers ce document, le protocole Loopring propose une solution hybride avec des différences d'approches significatives.

3 Le protocol Loopring

Loopring n'est pas un DEX, mais un protocole modulaire pour construire des DEX sur plusieurs chaînes de blocs. Nous démontons les éléments constitutifs d'une place d'échange traditionnelle et proposons à sa place un ensemble de contrats publics intelligents et d'acteurs décentralisés. Les rôles dans le réseau comprennent les portefeuilles, les relais, les consortiums de chaînes de blocs de partage de liquidité, les navigateurs pour les carnets d'ordres, les mineurs du réseau et les services de création de jetons pour les actifs. Avant de définir chacun d'eux, nous devons d'abord comprendre les ordres de Loopring.

3.1 Anneau d'ordres

Les ordres Loopring sont exprimés dans ce que nous appelons un modèle d'ordre unidirectionnel (UDOM)[16]. L'UDOM exprime les ordres sous forme de demandes d'échange de jetons, `montantS/montantB`, (montant à vendre/acheter) au lieu d'offres et de demandes. Comme chaque ordre n'est qu'un taux de change entre deux jetons, une caractéristique puissante du protocole est le mélange et l'appariement de plusieurs ordres dans des transactions circulaires. En utilisant jusqu'à 16 ordres au lieu d'une seule paire de négociation, il y a une augmentation spectaculaire de la liquidité et potentiellement du prix.

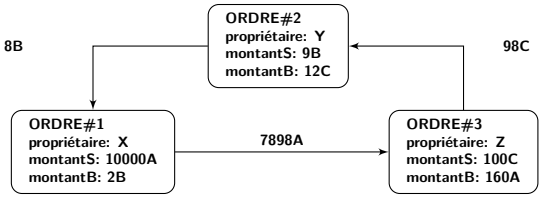

Figure 1: Un anneau d'ordres à 3 ordres

La figure ci-dessus montre un anneau d'ordres de 3 ordres. Chaque ordre de vente de jetons (`jetonS`) est le jeton d'un autre ordre d'achat (`jetonB`). Il crée une boucle qui permet à chaque ordre d'échanger les jetons souhaités sans avoir besoin d'un ordre opposé pour sa paire. Les paires d'ordres d'échanges traditionnelles peuvent, bien sûr, être exécutées dans ce qui est essentiellement un cas particulier d'un groupe d'ordres.

Definition 3.1 (Anneau d'ordres) *Soient C_0, C_1, $\cdots$, C_{n-1}, n jetons différents, et $O_{0\to1}$, $\cdots$, $O_{i\to i\oplus1}$, $\cdots$, $O_{n-1\to0}$ sont n ordres. Ces ordres peuvent former un anneau d'ordre pour les échanges :*

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

où n est la longueur de l'anneau d'ordres, et $i \oplus 1 \equiv i + 1 \mod n$.

Un anneau d'ordres est valide lorsque toutes les transactions qui le compose peuvent être exécutées à un taux de change égal ou supérieur au taux initial spécifié implicitement par l'utilisateur. Pour vérifier la validité de l'anneau d'ordres, les contrats intelligent du protocol Loopring doivent recevoir des anneaux d'ordres de la part des mineurs d'anneau lorsque le produit des taux de change originaux de toutes les commandes est égal ou supérieur à 1.

Supposons qu'Alice et Bob veulent échanger leurs jetons `A` et `B`. Alice a 15 jetons `A`et elle en veut 4 jetons `B` en échange ; Bob a 10 jetons `B` et il en veut 30 jetons `A` en échange.

Qui achète et qui vend ? Cela dépend uniquement de l'actif que nous choisissons pour fixer les cotations de prix. Si le jeton `A` est la référence, Alice achète le jeton `B` pour le prix de $\frac{15}{4} = 3.75$ `A`, tandis que Bob vend 10 jetons `B` pour le prix de $\frac{30}{10} = 3.00$ `A`. Dans le cas où l'on choisit le jeton `B` comme référence, nous disons qu'Alice vend 15 jeton `A` pour le prix de $\frac{4}{15} = 0.26666667$ `B` et Bob achète 10 jetons `A` pour le prix de $\frac{10}{30} = 0.3333333334$ `B`. Par conséquent, dire qui est l'acheteur ou le vendeur est un choix arbitraire.

Dans la première situation, Alice est prête à payer un prix plus élevé (3, 75A) que le prix auquel Bob vend ses jetons (3.00A), alors que dans la seconde situation Bob est prêt à payer un prix plus élevé (0, 33333334B) que le prix auquel Alice vend ses jetons (0, 2666666667B). Il est clair qu'une transaction est possible lorsque l'acheteur est prêt à payer un prix égal ou supérieur au prix du vendeur.

$$\frac{\frac{15}{4}}{\frac{30}{10}} = \frac{\frac{10}{30}}{\frac{4}{15}} = \frac{15}{4} \cdot \frac{10}{30} = 1.25 > 1 \quad (1)$$

Ainsi, pour qu'un ensemble d'ordres n puisse être rempli, totalement ou en partie, il faut savoir si le produit de chacun des taux de change comme les ordres d'achat se traduit par un nombre supérieur ou égal à 1. Si c'est le cas, tous les ordres n peuvent être partiellement ou totalement exécutés [17].

Si nous introduisons une troisième partie prenante, Charlie, de sorte qu'Alice veut donner x_1 jetons A et recevoir y_1 jetons B, Bob veut donner x_2 jetons B et recevoir y_2 jetons C, et Charlie veut donner x_3 jetons C et recevoir y_3 jetons A. Les jetons nécessaires sont présents, et le commerce est possible si :

$$\frac{x_1 \cdot x_2 \cdot x_3}{y_1 \cdot y_2 \cdot y_3} \geq 1 \quad (2)$$

Voir section 7.1 pour avoir plus de détails concernant les ordres de Loopring.

4 Participants à l'écosystème

Les participants à l'écosystème suivants fournissent conjointement toutes les fonctionnalités qu'un échange centralisé a à offrir.

- **Portefeuilles** : Un service ou une interface pour les portefeuilles courant qui permet aux utilisateurs d'accéder à leurs jetons et d'envoyer des commandes au réseau Loopring. Les portefeuilles seront incités à produire des ordres en partageant les récompenses avec les mineurs (voir la section 8). Ayant la conviction que l'avenir du trading n'aura lieu que grâce à la sécurité des portefeuilles des utilisateurs individuels, il est primordial de connecter les pools de liquidité par le biais de notre protocole.
- **Chaîne de blocs de partage de liquidité/ maille de relais** : Un réseau de mailles de relais pour les ordres & le partage de liquidité. Lorsque les nœuds exécutent le logiciel de relais Loopring, ils sont en mesure de rejoindre un réseau existant et de partager des liquidités avec d'autres relais par le biais d'un consortium de chaîne de blocs. Le consortium de chaîne de blocs que nous construisons en tant que première implémentation a un partage d'ordres en temps quasi réel (1-2 secondes par blocs), et réduit l'ancien historique pour permettre un téléchargement plus rapide pour les nouveaux nœuds. Notamment, les relais n'ont pas besoin de se joindre à ce consortium ; ils peuvent agir seuls et ne pas partager la liquidité avec d'autres, ou, ils peuvent démarrer et gérer leur propre réseau de partage de liquidité.
- **Relais/Mineurs d'anneaux** : Les relais sont des nœuds qui reçoivent des ordres des portefeuilles ou du maillage de relais, tiennent à jour les carnets d'ordres publics et l'historique des transactions, et éventuellement diffusent des ordres à d'autres relais (par l'intermédiaire de n'importe quel moyen hors chaîne) et/ou des nœuds de maillage de relais. Le minage en anneau est une caractéristique - et non une exigence - des relais. Il demande des capacités de calculs importantes et se fait complètement hors chaîne. Nous appelons relais, avec la fonction de minage d' anneau activée, "les mineurs d'anneaux", qui produisent des anneaux d'ordres en assemblant des ordres disparates. Les relais sont libres de choisir (1) comment ils communiquent entre eux, (2) comment ils construisent leurs carnets de commandes, et (3) comment ils exploitent les anneaux d'ordres (algorithmes de minage).
- **Les contrats intelligents du Protocole Loopring(LPSC)** : Un ensemble de contrats intelligents publics et gratuits qui vérifie les anneaux d'ordres reçus des mineurs, règle et transfère sans confiance les jetons au nom des utilisateurs, incite les mineurs et les portefeuilles avec des récompenses, et émet des événements. Les relais/navigateurs d'ordres surveillent ces événements pour tenir à jour leurs carnets de commandes et l'historique des transactions. Voir l'annexe **??** pour plus de détails.
- **Services de création de jetons d'actifs (ATS)**: Un pont entre les actifs qui ne peuvent pas être négociés directement sur Loopring. Il s'agit de services centralisés gérés par des entreprises ou des organisations dignes de confiance. Les utilisateurs déposent des actifs (réels, fiat ou jetons d'autres chaînes) et obtiennent des jetons émis, qui peuvent être rachetés pour le dépôt à l'avenir. Loopring n'est pas un protocole d'échange inter-chaînes (jusqu'à ce qu'une solution appropriée existe), mais ATS permet de négocier des jetons ERC20 [18] avec des actifs physiques ainsi que des actifs sur d'autres chaînes de blocs.

5 Processus d'échange

1. **Autorisation du protocole** : Dans la figure 2, l'utilisateur Y qui veut échanger des jetons autorise le LPSC à gérer le montantSde jetons B que l'utilisateur veut vendre. Cela ne bloque pas les jetons de l'utilisateur, qui reste libre de les déplacer pendant le traitement de la commande.

2. **Création d'ordres** : Le taux actuel et le carnet de commandes pour les jetons `B` vs les jetons `C`, sont fournis par des relais ou d'autres agents connectés au réseau, tels que les navigateurs de carnet de commandes. L'utilisateur `Y` passe un ordre (ordre à cours limité) en spécifiant le`montantS` et le `montantB` et d'autres paramètres par le biais de n'importe quelle interface de portefeuille intégrée. Un montant de LRx peut être ajouté à la commande en tant que récompense pour les mineurs d'anneau ; des récompenses de LRx plus élevés signifient une meilleure chance d'être traités rapidement par les mineurs d'anneau. Le hash de la commande est signé avec la clé privée de l'utilisateur `Y`.

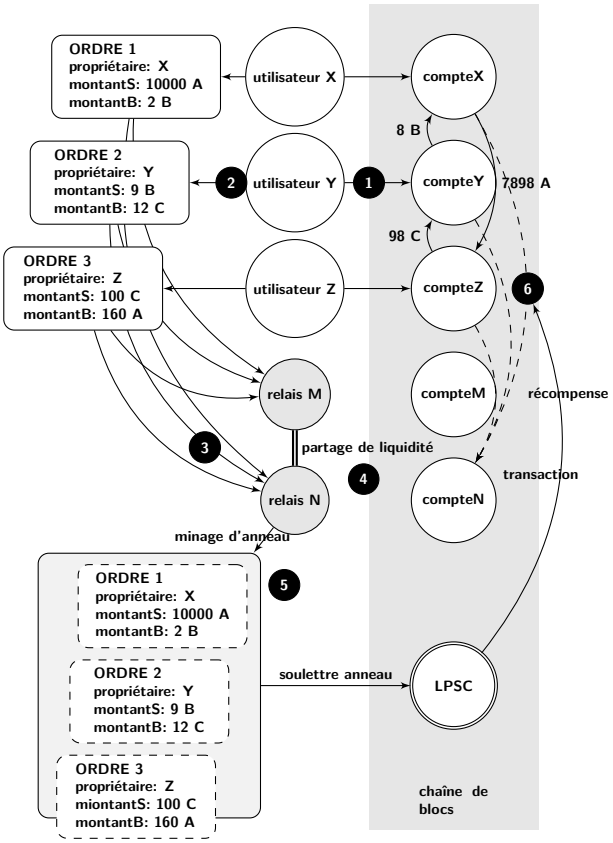

Figure 2: Processus d'échange Loopring

3. **Diffusion d'ordre** : Le portefeuille envoie la commande et sa signature à un ou plusieurs relais. Les relais mettent à jour leur carnet d'ordre public. Le protocole n'exige pas que les carnets de commandes soient construits d'une certaine façon, comme le principe du premier arrivé, premier servi. Au lieu de cela, les relais ont le pouvoir de prendre leurs propres décisions de conception en construisant leurs carnets de commandes.

4. **Partage de Liquidité** : Les relais diffusent l'ordre à d'autres relais par n'importe quel moyen de communication. Ici encore, il y a une certaine souplesse quant à la façon dont les nœuds interagissent. Pour faciliter un certain niveau de connectivité du réseau, il y a un relais de partage de liquidité intégré utilisant un consortium de chaîne de blocs. Comme mentionné dans la section précédente, ce maillage de relais est optimisé pour la vitesse et la globalité.

5. **Minage en anneaux (appariement des ordres)** : Les mineurs d'anneaux essaient de satisfaire l'ordre entièrement ou partiellement au taux de change donné ,ou mieux, en l'appariant avec de multiples autres ordres. Le minage en anneau est la principale raison pour laquelle le protocole est capable de fournir une grande liquidité sur n'importe quelle paire. Si le taux d'exécution est meilleur que ce que l'utilisateur Y a spécifié, la marge est partagée entre tous les ordres dans l'anneau d'ordres. En guise de récompense, le mineur choisit entre réclamer une partie de la marge (partage de marge, et rendre le LRx à l'utilisateur), ou simplement garder les récompense de LRx.

6. **Vérification & Transaction** : L'anneau d'ordre est reçu par le LPSC. Il effectue de multiples vérifications pour vérifier les données fournies par les mineurs d'anneau et détermine si l'anneau d'ordres peut être réglé entièrement ou partiellement (en fonction du taux d'accomplissement de l'anneau d'ordres et des jetons dans le portefeuille de l'utilisateur). Si tous les contrôles sont réussis, le contrat transfère "atomiquement" les jetons aux utilisateurs et paie en même temps le mineur d'anneau et les frais de portefeuille. Si le solde de l'utilisateur `Y` tel que déterminé par le LPSC est insuffisant, il sera considéré comme réduit : un ordre réduit sera automatiquement ramené à sa taille originale si suffisamment de fonds sont déposés à son adresse, contrairement à une annulation, qui est une opération manuelle à sens unique et ne peut pas être annulée.

6 Flexibilité opérationnelle

Il est important de noter que le standard ouvert de Loopring permet aux participants une grande flexibilité dans la façon dont ils opèrent. Les acteurs sont libres de mettre en œuvre de nouveaux modèles d'affaires et de fournir de la valeur aux utilisateurs, en percevant des honoraires LRx sur le volume ou d'autres mesures dans le processus (s'ils le souhaitent). L'écosystème est modulaire et vise à soutenir la participation à partir d'une multitude d'applications.

6.1 Carnet de commande

Les relais peuvent concevoir leurs carnets de commandes de différentes manières pour afficher et faire correspondre les ordres des utilisateurs. Une première mise en œuvre de notre propre carnet d'ordres suit un modèle OTC, où les ordres à

cours limité sont positionnés sur la seule base du prix. En d'autres termes, l'horodatage des commandes n'a aucune incidence sur le carnet de commandes. Cependant, un relais est libre de concevoir son carnet d'ordres de manière à imiter le moteur d'appariement d'une place d'échanges centralisée typique, où les ordres sont classés par prix, tout en respectant les horodatages. Si un relais était enclin à offrir ce type de carnet de commandes, ils peuvent posséder ou s'intégrer avec un portefeuille, et faire envoyer ces commandes de portefeuille exclusivement au relais unique, qui serait alors en mesure d'apparier les ordres en fonction du temps. Une telle configuration est possible.

Alors que d'autres protocoles DEX exigent parfois des relais d'avoir des ressources - des soldes de jetons initiaux pour passer des ordres de preneur - les relais Loopring n'ont besoin que de trouver des ordres compatibles pour effectuer une transaction, et peuvent le faire sans jetons initiaux.

6.2 Partage de liquidité

Les relais sont libres de concevoir comment ils partagent la liquidité (ordres) les uns avec les autres. Notre consortium de chaîne de blocs n'est qu'une solution pour y parvenir, et l'écosystème est libre de se mettre en réseau et de communiquer comme ils le souhaitent. En plus de se joindre à un consortium, ils peuvent construire et gérer leur propre chaîne de blocs, en créant des règles/incitations comme ils l'entendent. Les relais peuvent aussi fonctionner seuls, comme on peut le voir dans la mise en place d'un portefeuille sensible au temps. Bien sûr, il y a des avantages évidents à communiquer avec d'autres relais dans la recherche d'effets de réseau, cependant, différents modèles d'affaires pourraient mériter des conceptions particulières de partage et des frais partagés de plusieurs façons.

7 Spécification du protocole

7.1 Anatomie d'un ordre

Une commande est un ensemble de données qui décrit l'intention de donner un ordre de l'utilisateur. Un ordre Loopring est défini à l'aide du modèle d'ordre unidirectionnel, ou UDOM, comme suit :

```
message Order {
  address protocol;
  address owner;
  address tokenS;
  address tokenB;
  uint256 amountS;
  uint256 amountB;
  unit256 lrcFee
  unit256 validSince; // Seconds since epoch
  unit256 validUntil; // Seconds since epoch
  uint8   marginSplitPercentage;  // [1-100]
  bool    buyNoMoreThanAmountB;
  uint256 walletId;
  // Dual-Authoring address
  address authAddr;
  // v, r, s are parts of the signature
  uint8   v;
  bytes32 r;
  bytes32 s;
  // Dual-Authoring private-key,
  // not used for calculating order's hash,
  // thus it is NOT signed.
  string  authKey;
  uint256 nonce;
}
```

Pour assurer l'origine de l'ordre, il est signée avec le hachage de ses paramètres, à l'exclusion de `authAddr`, avec la clé privée de l'utilisateur. Le paramètre `authAddr` est utilisé pour signer les anneaux d'ordre dont cet ordre fait partie, ce qui empêche le front-running. Veuillez vous référer à la section 9.1 pour plus de détails. La signature est représentée par les champs `v`, `r`, et `s`, et est envoyée avec les paramètres de l'ordre sur le réseau. Ceci garantit que la commande reste immuable pendant toute sa durée de vie. Même si l'ordre ne change jamais, le protocole peut toujours calculer son état actuel en fonction du solde de son adresse et d'autres variables.

L'UDOM n'inclut pas un prix (qui doit être un nombre à virgule flottante par nature), mais utilise plutôt le terme `rate` ou r, qui est exprimé en `montantS`/`montantS`/`montantB`. Le taux n'est pas un nombre à virgule flottante mais une expression qui ne sera évaluée qu'avec d'autres entiers non signés sur demande, afin de conserver tous les résultats intermédiaires comme entiers non signés et d'augmenter la précision du calcul.

7.1.1 Montants d'achat

Quand un mineur d'anneau exécute un ordre, il est possible qu'un meilleur taux soit alors exécutable, permettant aux utilisateurs d'obtenir plus de `jetonB` que le `montantB` qu'ils ont spécifié. Cependant, si `buyNoMoreThanAmountB` est réglé sur `True`, le protocole garantit que les utilisateurs ne reçoivent pas plus que `montantB` de `jetonB` de `jetonB`. Ainsi, le paramètre UDOM `buyNoMoreThantokenB` détermine quand une commande est considérée comme complètement remplie. `buyNoMoreThantokenB` applique un plafond sur les montants `montantS` ou `montantB`, et permet aux utilisateurs d'exprimer des intentions plus fines que les ordres d'achat/de vente traditionnels.

Par exemple : avec `montantS` = 10 et `montantB` = 2, le taux $r = 10/2 = 5$. Ainsi, l'utilisateur est prêt à vendre 5 `jetonS` pour chaque `jetonB`. Le mineur d'anneaux travaille et trouve pour l'utilisateur un taux de 4, permettant à l'utilisateur de recevoir 2,5 `jetonB` au lieu de 2. Cependant, si l'utilisateur ne veut que 2 `jetonB` et met le drapeau `buyNoMoreThanAmountB` sur `True`, le LPSC effectue la transaction à un taux de 4 et l'utilisateur vend 4 `jetonS`

pour chaque `jetonB`, sauvegardant ainsi 2 `jetonS`. Gardez à l'esprit que ceci ne tient pas compte des frais de minage (voir section 8.1). En effet, si l'on utilise :

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

pour représenter un ordre sous une forme simplifiée, alors pour les marchés ETH/USD sur une place d'échanges traditionnelle, le modèle traditionnel d'achat-vente peut exprimer le 1er et le 3ème ordre ci-dessous, mais pas les deux autres :

1. Vendre 10 ETH au prix de 300 USD/ETH. Cet ordre peut être exprimé de la façon suivante : `Order(10, ETH, 3000, USD, False)`.

2. Vendre ETH au prix de 300 USD/ETH pour obtenir 3000 USD. Cet ordre peut être exprimé de la façon suivante: `Order(10, ETH, 3000, USD, True)`.

3. Acheter 10 ETH au prix de 300 USD/ETH, Cet ordre peut être exprimé de la façon suivante: `Order(3000, USD, 10, ETH, True)`.

4. Dépenser 3000 USD pour acheter autant d'ETH que possible au prix de 300 USD/ETH, Cet ordre peut être exprimé de la façon suivante: `Order(3000, USD, 10, ETH, False)`.

7.2 Vérification de l'anneau

Les contrats intelligents Loopring n'effectuent pas de calculs de taux de change ou de montant, mais doivent recevoir et vérifier ce que les mineurs d'anneaux fournissent pour ces valeurs. Ces calculs sont effectués par les mineurs d'anneau pour deux raisons principales : (1) le langage de programmation pour les contrats intelligents, comme solidity[19] sur Ethereum, ne supporte pas les calculs avec virgule flottante, en particulier $pow(x, 1/n)$ (calcul de la n-ème racine d'un nombre à virgule flottante), et (2) il est souhaitable que le calcul soit effectué hors chaîne pour réduire l'impact en termes de poids de calcul et de coût.

7.2.1 Vérification des sous-anneaux

Cette étape empêche les arbitragistes de capter injustement toute la marge dans un anneau 'ordres en mettant en œuvre de nouveaux ordres au sein d'un anneau. Effectivement, une fois qu'un anneau d'ordre valide est trouvé par un mineur d'anneaux, il pourrait être tentant d'ajouter d'autres ordres à cet anneau pour absorber complètement la marge des utilisateurs (rabais de taux). Comme illustré par la figure 3 ci-dessous, un calcul méticuleux de $x1$, $y1$, $y1$, $x2$ et $y2$ fera en sorte que le produit de toutes les commandes soit exactement 1, donc il n'y aura pas de réduction de taux.

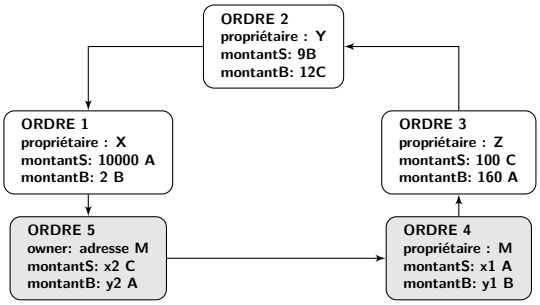

Figure 3: Un anneau d'ordres avec sous-anneau

Il s'agit d'un risque nul, d'une valeur ajoutée nulle pour le réseau et d'un comportement déloyal de la part du mineur d'anneau. Pour éviter cela, Loopring exige qu'une boucle valide ne puisse pas contenir de sous-anneaux. Pour vérifier cela, le LPSC s'assure qu'un jeton ne peut pas être à la fois en position d'achat ou de vente. Dans le diagramme ci-dessus, nous pouvons voir que le jeton `A` est à la fois un jeton de vente et un jeton d'achat, ce qui serait refusé.

7.2.2 Vérification du taux de réalisation

Les calculs du taux de change dans l'anneau d'ordrse sont effectués par les mineurs d'anneau pour les raisons indiquées ci-dessus. C'est le LPSC qui doit vérifier qu'ils sont corrects. Tout d'abord, il vérifie que le taux d'achat que le mineur d'anneaux peut exécuter pour chaque ordre est égal ou inférieur au taux d'achat initial fixé par l'utilisateur. Cela permet de s'assurer que l'utilisateur obtient au moins le taux de change qu'il a demandé ou mieux sur la transaction. Une fois que les taux de change sont confirmés, le LPSC s'assure que chaque ordre dans l'anneau d'ordres partage le même rabais de taux. Par exemple, si le taux d'escompte est γ, alors le prix de chaque commande sera :

$r_{0\to1} \cdot (1-\gamma)$, $r_{1\to2} \cdot (1-\gamma)$, $r_{2\to0} \cdot (1-\gamma)$, et satisfait :

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

d'où :

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

Si la transaction dépasse n ordres, le `rabais` est:

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

où r^i est le taux de rotation des ordres du i-ème ordre. Évidemment, c'est seulement quand le taux d'escompte est $\gamma \geq 0$, que ces ordres peuvent être exécutés ; et le taux de change réel du i-ème ordre (O^i) est $\hat{r^i} = r^i \cdot (1-\gamma)$, $\hat{r^i} \leq r^i$.

7.2.3 Suivi des transactions & Annulation

Un utilisateur peut annuler partiellement ou totalement une commande en envoyant une transaction spéciale au LPSC, contenant les détails de la commande et les montants à annuler. Le LPSC prend cela en compte, stocke les montants à annuler et émet un événement `OrderCancelled` au réseau. Le LPSC garde la trace des montants remplis et annulés en stockant leurs valeurs en utilisant le hachage de l'ordre comme identificateur. Ces données sont accessibles au public et les événements `OrderCancelled` / `OrderFilled` sont émis lorsqu'ils changent. Le suivi de ces valeurs est essentiel pour le LPSC pendant l'étape de règlement du cycle d'ordres.

Le LPSC prend également en charge l'annulation de tous les ordres pour toute paire d'échange avec l'événement `OrdersCancelled` et l'annulation de tous les ordres pour une adresse avec l'événement `AllOrdersCancelled`.

7.2.4 Échelonnement des ordres

Les ordres sont échelonnées en fonction de l'historique des montants exécutés et annulés et du solde courant des comptes des expéditeurs. Le processus recherche l'ordre avec le plus petit montant à remplir en fonction des caractéristiques ci-dessus et l'utilise comme référence pour l'échelonnement de toutes les transactions dans le cycle d'ordre.

Trouver l'ordre de la valeur la plus basse peut aider à déterminer le volume de remplissage pour chaque commande. Par exemple, si le i-ème ordre est l'ordre de valeur la plus basse, alors le nombre de jetons vendus de chaque ordre $\hat{s}$ et le nombre de jetons achetés $\hat{b}$ de chaque ordre peut être calculé comme suit :

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

où $\overline{s}_i$ est le solde restant après l'exécution partielle des ordres.

Au cours de la mise en œuvre, nous pouvons en toute sécurité assumer n'importe quel ordre dans l'anneau d'ordres pour avoir la valeur la plus basse, puis itérer à travers l'anneau d'ordres au plus deux fois pour calculer le volume de remplissage de chaque ordre.

Exemple : si la plus petite quantité à remplir par rapport à l'ordre d'origine est de 5%, toutes les transactions de l'anneau d'ordres sont réduites de 5%. Une fois les transactions terminées, l'ordre qui était considéré comme ayant le plus petit montant restant à remplir devrait être complètement rempli.

7.3 Transaction en anneau

Si l'anneau d'ordres remplit toutes les vérifications précédentes, l'anneau d'ordre peut être clôturé et des transactions peuvent être effectuées. Cela signifie que tous les ordres n forment un anneau d'ordres fermé, connecté comme dans la figure 4 :

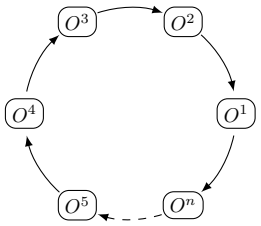

Figure 4: Transaction en anneau

Pour effectuer les transactions, le LPSC utilise le contrat intelligent `TokenTransferDelegate`. L'introduction d'un tel représentant facilite la mise à jour du protocole du contrat intelligent puisque toutes les commandes doivent seulement autoriser ce représentant au lieu de versions différentes du protocole.

Pour chaque commande de l'anneau d'ordres, un paiement de `JetonS` est effectué à la commande suivante ou précédente en fonction de l'exécution. Ensuite, les frais pour le mineur d'anneaux sont payés en fonction du modèle qu'il a choisit. Enfin, une fois que toutes les transactions sont effectuées, un événement `RingMined` est émis.

7.3.1 Événements émis

Le protocole émet des événements qui permettent aux relais, aux navigateurs d'ordres et aux autres acteurs de recevoir les mises à jour du carnet de commandes aussi efficacement que possible. Les événements émis sont :

- **OrderCancelled**: Un ordre particulier a été annulée.
- **OrdersCancelled**: Tous les ordres d'une paire d'échange à partir d'une adresse propre ont été annulés.
- **AllOrdersCancelled**: Tous les ordres de toutes les paires d'échange à partir d'une adresse propre ont été annulés.
- **RingMined**: Un anneau d'ordres a été établi avec succès. Cet événement contient les données relatives à chaque transfert de jeton de l'anneau intérieur.

8 Jetons LRx

LRx est notre façon générique de nommer les jetons.. LRC est le jeton Loopring sur Ethereum, LRQ sur Qtum, et LRN sur NEO, etc. D'autres types de LRx seront introduits à l'avenir à mesure que Loopring sera déployé sur d'autres chaînes de blocs publics.

8.1 Modèle de frais

Lorsqu'un utilisateur crée un ordre, il spécifie un montant de LRx à payer au mineur d'anneau en tant que frais, en conjonction avec un pourcentage de la marge (`marginSplitPercentage`) réalisée sur la commande que le mineur d'anneau peut réclamer. C'est ce qu'on appelle le fractionnement de la marge. Le choix fait (frais ou partage de la marge) appartient au mineur d'anneau.

Représentation du fractionnement de la marge :

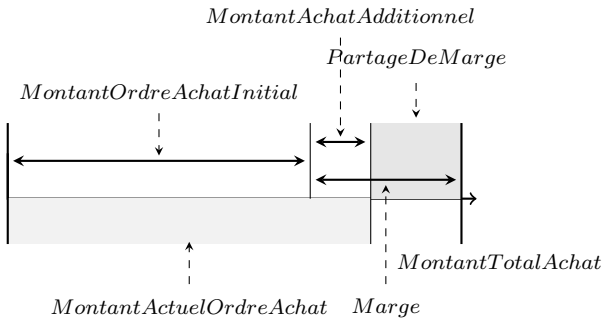

Figure 5: Un partage de marge de 60%

Si la marge sur l'anneau d'ordres est trop petite, un mineur d'anneau choisira les frais de LRx. Si, au contraire, la marge est suffisamment importante pour que le partage de la marge qui en résulte ait une valeur beaucoup plus élevée que les frais de LRx, le mineur choisira le partage de la marge. Il y a toutefois une autre condition : lorsque le mineur choisit le partage de la marge, il doit payer à l'utilisateur (créateur de l'ordre) une redevance, qui est égale au LRx que l'utilisateur aurait payé au mineur en tant que redevance. Cela augmente le seuil à partir duquel le mineur choisira le partage de la marge jusqu'à deux fois les frais LRx de l'ordre, ce qui augmente la propension au choix des frais LRx. Ceci permet aux mineurs d'anneau de recevoir un revenu constant sur les anneaux d'ordres à faible marge avec le compromis de recevoir moins de revenu sur les anneaux d'ordres à marge plus élevée. Notre modèle d'honoraires est basé sur le fait qu'au fur et à mesure que le marché va croître et arrive à maturité, il y aura moins d'anneaux d'ordres à marge élevée, ce qui nécessitera des frais fixes de LRx comme récompense.

Nous arrivons finalement au graphique suivant :

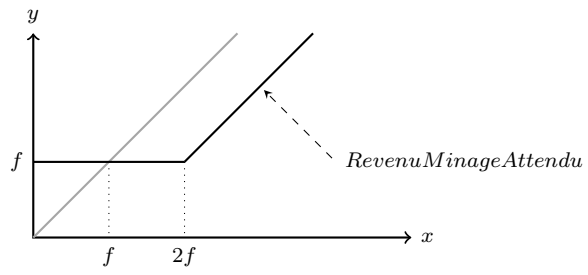

Figure 6: Modèle de frais Loopring

où f est la commission LRx, x est le partage de la marge, y est le revenu du minage. $y = max(f, x-f)$ comme indiqué par la ligne en trait plein ; si les frais de LRx pour l'ordre est 0, l'équation est $y = max(0, x-0)$ ce qui se simplifie à $y = x$ comme indiqué par la ligne grise.

Les conséquences sont :

1. Si le partage de la marge est de 0, les mineurs d'anneau choisiront le forfait LRx et seront toujours récompensés.

2. Si les frais de LRx sont de 0, le résultat de la ligne grise et le revenu est basé sur un modèle linéaire général.

3. Quand le revenu du partage de la marge est supérieur à 2x(LRx fee), les mineurs d'anneau choisissent le partage de la marge et paient LRx à l'utilisateur.

Il convient de noter que si les frais de LRx ne sont pas nuls, quelle que soit l'option choisie par le mineur d'anneau, il y aura toujours un transfert de LRx entre le mineur d'anneau et l'expéditeur de l'ordre. Soit le mineur d'anneau gagne les frais de LRx, soit il rembourse les frais de LRx à l'expéditeur pour prendre le partage de la marge.

Les mineurs d'anneaux partageront un certain pourcentage des frais avec les portefeuilles. Lorsqu'un utilisateur passe un ordre via un portefeuille et qu'il est satisfait, le portefeuille est récompensé par une partie des frais ou du partage de la marge. Bien que cela soit modulaire et que des modèles d'affaires ou des implémentations uniques soient possibles, notre tendance est que les portefeuilles reçoivent environ 20%-25% des frais gagnés. Les portefeuilles représentent une cible principale pour l'intégration du protocole Loopring car ils ont une base d'utilisateurs, mais peu ou pas de source de revenu.

8.2 Gouvernance décentralisée

À la base, le protocole Loopring est un protocole social en ce sens qu'il repose sur la coordination entre ses membres pour fonctionner efficacement vers un but. Cela n'est pas différent des protocoles crypto-économiques au sens large, et en effet, son utilité est largement protégée par les mêmes mécanismes de problèmes de coordination [20], un équilibre "grim trigger" et une rationalité limitée. A cette fin, les jetons LRx ne sont pas seulement utilisés pour payer les frais, mais aussi pour harmoniser les incitations financières des différents participants au réseau. Un tel ajustement est nécessaire pour l'adoption à grande échelle de tout protocole, mais il est particulièrement important pour les protocoles d'échange, étant donné que le succès repose en grande partie sur l'amélioration de la liquidité dans un écosystème décentralisé robuste.

Les jetons LRx seront utilisés pour effectuer des mises à jour de protocole par le biais d'une gouvernance décentralisée. Les mises à jour des contrats intelligents seront régies par les détenteurs de jetons afin d'assurer la continuité et la sécurité et d'atténuer les risques de

liquidité siphonnée par une incompatibilité. Étant donné que les contrats intelligents ne peuvent pas être modifiés une fois déployés, il existe un risque que les dApps ou les utilisateurs finaux continuent d'interagir avec des versions obsolètes et s'excluent eux-mêmes des contrats mis à jour. L'évolutivité est cruciale pour le succès du protocole car il doit s'adapter aux demandes du marché et aux chaînes de blocs sous-jacentes. La gouvernance décentralisée par les parties prenantes de LRx permettra de mettre à jour les contrats intelligents sans perturber les dApps ou les utilisateurs finaux, ou en s'appuyant trop sur l'abstraction des contrats intelligents. Dans un premier temps, cela se fera par le biais d'un simple contrat intelligent multisignature, en vue de progresser vers un mécanisme de type DAO.

9 Protections contre les fraudes et les attaques

9.1 Prévention du Front-running

Dans les places d'échanges décentralisés, le front-running se produit lorsque quelqu'un essaie de copier la solution d'échange d'un autre nœud et de la faire miner avant la transaction originale qui est dans le pool de transactions en attente (mempool). Ceci peut être réalisé en spécifiant des frais de transaction plus élevés (prix du gas). Le schéma principal du front-running dans Loopring (et tout protocole pour l'appariement des ordres) est le voleur-d'ordre : lorsqu'un front-runner vole un ou plusieurs ordres d'une transaction de règlement d'ordre en attente ; et, spécifique à Loopring : lorsqu'un front-runner vole l'ensemble de l'ordre d'une transaction en attente.

Lorsqu'une transaction submitRing n'est pas confirmée et se trouve toujours dans le pool de transactions en attente, n'importe qui peut facilement la repérer et remplacer `minerAddress` par sa propre adresse (l'adresse `filcherAddress`), alors ils peuvent re-signer la charge utile avec `filcherAddress` pour remplacer la signature de l'anneau d'ordre. Le fincher peut fixer un prix du gas plus élevé et soumettre une nouvelle transaction en espérant que les mineurs de bloc choisiront sa nouvelle transaction dans le bloc suivant au lieu de la transaction originale.

Les solutions précédentes à ce problème présentaient d'importants inconvénients : il fallait plus de transactions et donc plus de gaz pour les mineurs d'anneaux ; et il fallait au moins deux fois plus de blocs pour régler un anneau d'ordres. Notre nouvelle solution, le Dual Authoring[21], implique le mécanisme de mise en place de deux niveaux d'autorisation pour les ordres - l'un pour le règlement et l'autre pour le minage en anneaux.

Processus de double création :

1. Pour chaque commande, le logiciel de portefeuille génère une paire aléatoire clé publique/clé privée, et place la paire de clés dans l'extrait JSON de la commande. (Une alternative est d'utiliser l'adresse dérivée de la clé publique au lieu de la clé publique elle-même pour réduire le nombre d'octets nécessaires. Nous utilisons `authAddr` pour représenter une telle adresse, et `authKey` pour représenter la clé privée correspondante de `authAddr`).
2. Calculez le hachage de l'ordre avec tous les champs de l'ordre sauf `r`, `v`, `s`, et `authKey`), et signez le hachage en utilisant la clé privée du `propriétaire` (pas `authKey`).
3. Le portefeuille enverra la commande avec le `authKey` à des relais pour le minage d'anneau. Les mineurs d'anneau vérifieront que `authKey` et `authAddr` sont correctement appariés et que la signature de la commande est valide en ce qui concerne l'adresse du `propriétaire`.
4. Lorsqu'un anneau d'ordres est identifié, le mineur utilisera la clé de chaque ordre pour signer le hachage de l'anneau, l'adresse `minerAddress` et tous les paramètres de minage. Si un anneau d'ordres contient des ordres n, il y aura n signatures par les n `authKey`s. Nous appelons ces signatures `authSignature`s. Les mineurs d'anneau peuvent aussi avoir besoin de signer le hachage de l'anneau ainsi que tous les paramètres de minage en utilisant la clé privée de `minerAddress`.
5. Le mineur d'anneau appelle la fonction submitRing avec tous les paramètres, ainsi que tous les autres `authSignature`s. Notez que les `authKey`s ne font PAS partie de la transaction en chaîne et restent donc inconnus des parties autres que le mineur d'anneau lui-même.
6. Le protocole Loopring va maintenant vérifier chaque `authSignature` par rapport à la `authAddr` correspondante de chaque ordre, et rejeter l'anneau d'ordres si une `authSignature` est manquante ou invalide.

Il en résulte que :

- La signature de la commande (par la clé privée de l'adresse du `propriétaire`) garantit que la commande ne peut pas être modifiée, y compris par le `authAddr`.
- La signature du mineur (par la clé privée du `minerAddress`), si elle est fournie, garantit que personne ne peut utiliser son identité pour exploiter un anneau d'ordres.
- Le `authSignature` garantit que l'ensemble de l'anneau d'ordres ne peut pas être modifié, y compris l'adresse `minerAddress`, et qu'aucune commande ne peut être volée.

Le Dual Authoring empêche le vol-d'anneau et le vol-d'ordre tout en s'assurant que le règlement des anneaux d'ordres peut se faire en une seule transaction. De plus, le Dual Authoring permet aux relais de partager les ordres de deux manières : le partage non compatible et le partage compatible. Par défaut, Loopring utilise un modèle OTC

et ne prend en charge que les ordres à cours limité, ce qui signifie que les horodatages des ordres sont ignorés. Cela implique qu'une opération en avance n'a pas d'impact sur le prix réel de cette opération, mais a un impact sur le fait qu'elle soit exécutée ou non.

10 Autres types d'attaques

10.1 Attaque Sybil ou DOS

Les utilisateurs malveillants – agissant à découvert ou sous de fausses identités – pourraient envoyer un grand nombre de petites commandes pour attaquer les nœuds Loopring. Cependant, puisque nous permettons aux nœuds de rejeter des ordres en fonction de leurs propres critères – qu'ils peuvent cacher ou révéler – la plupart de ces ordres seront rejetés parce qu'ils ne produisent pas un profit satisfaisant lorsqu'ils sont appariés. En donnant aux relais le pouvoir de dicter la façon dont ils gèrent les ordres, nous ne considérons pas qu'une attaque massive d'ordres minuscules soit dangereuse.

10.2 Solde insuffisant

Les utilisateurs malveillants pourraient signer et diffuser des ordres dont la valeur d'ordre n'est pas nulle mais dont l'adresse a en réalité un solde nul. Les nœuds pourraient surveiller et remarquer que le solde réel de certains ordres est égal à zéro, les mettre à jour, puis les éliminer. Les nœuds doivent consacrer du temps à la mise à jour d'un ordre, mais peuvent aussi choisir de minimiser l'effort, par exemple, en inscrivant des adresses sur une liste noire et en supprimant les ordres connexes.

11 Résumé

Le protocole Loopring se veut une fondation pour les places d'échanges décentralisés. Ce faisant, elle a des répercussions profondes sur la façon dont les gens échangent leurs actifs et leurs valeurs. La monnaie, en tant que produit intermédiaire, facilite ou remplace le troc et résout la double coïncidence des besoins [22], où deux parties prenantes doivent désirer le bien ou le service distinct de l'autre. De même, le protocole Loopring a pour but de dispenser de nos dépendances sur la coïncidence des besoins dans les paires d'échange, en utilisant l'appariement en anneau pour des échanges plus facilement réalisés. Cela a un sens par rapport à la façon dont la société et les marchés échangent des jetons, des actifs traditionnels et bien plus encore. En effet, tout comme les crypto-monnaies décentralisées menacent le contrôle d'un pays sur l'argent, un protocole combinatoire qui permet de faire correspondre les commerçants (consommateurs/producteurs) à grande échelle est une menace théorique pour le concept même de l'argent.

Les avantages du protocole comprennent :

- La gestion des commandes hors chaîne et le règlement en chaîne signifie qu'il n'y a pas de sacrifices en termes de performance pour la sécurité.
- Une plus grande liquidité grâce au minage en anneau et au partage des commandes.
- Le Dual Authoring résout le problème pernicieux du front-running auquel sont confrontés tous les DEX et leurs utilisateurs aujourd'hui.
- Des contrats publics intelligents et gratuits permettent à n'importe quelle dApp de construire ou d'interagir avec le protocole.
- La normalisation entre les opérateurs permet d'obtenir des effets de réseau et d'améliorer l'expérience de l'utilisateur final.
- Réseau maintenu avec flexibilité dans la gestion des carnets d'ordres et de la communication.
- La réduction des barrières à l'entrée signifie des coûts moindres pour les nœuds qui se joignent au réseau et les utilisateurs finaux.
- Échange anonyme directement à partir du portefeuille de l'utilisateur.

12 Remerciements

Nous aimerions exprimer notre gratitude à nos mentors, conseillers et aux nombreuses personnes de la communauté qui nous ont partagé leurs connaissances avec nous. En particulier, nous aimerions remercier Shuo Bai (de ChinaLedger) ; le professeur Haibin Kan ; Alex Cheng, Hongfei Da ; Yin Cao ; Xiaochuan Wu ; Zhen Wang, Wei Yu, Nian Duan, Jun Xiao, Jiang Qian, Jiangxu Xiang, Yipeng Guo, Dahai Li, Kelvin Long, Huaxia Xia, Jun Ma et Encephalo Path pour la relecture et les remarques sur ce projet.

References

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

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

[3] 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.

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

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

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

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

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

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

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

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

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

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

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

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

[16] 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.

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

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

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

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

[21] 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.

[22] 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.