Loopring:
Un protocolo de intercambio de tokens descentralizado

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 15, 2018

Resumen

Loopring es un protocolo abierto para la creación de casas de cambio descentralizadas. Loopring opera como un conjunto público de contratos inteligentes (*smart contracts*) responsables del intercambio y la liquidación, con un grupo de actores externos (*off-chain*) añadiendo y comunicando órdenes. El protocolo es libre, extensible y sirve como un bloque de código estandarizado para aplicaciones descentralizadas (dApps) que quieran incorporar la funcionalidad de intercambio. Sus estándares interoperables facilitan el intercambio anónimo y libre de confianza en terceras partes. Una importante mejora frente a los protocolos de intercambio descentralizados actuales es su habilidad para mezclar y combinar diferentes órdenes entre sí, obviando las restricciones asociadas al uso de pares de cambio entre dos tokens, y mejorando considerablemente la liquidez. Loopring también emplea una solución única y robusta para prevenir la inversión ventajista, o *front-running*: el intento de enviar transacciones a un bloque de manera más rapida que el proveedor de la solución original. Loopring es una tecnología agnóstica, por lo que puede lanzarse en cualquier cadena de bloques que tenga la funcionalidad de los contratos inteligentes. En el momento de redactarse este artículo, Loopring ya es operable en Ethereum [1] [2], Qtum [3], y próximamente en NEO [4].

1 Introducción

Con la proliferación de activos basados en la cadena de bloques, la necesidad de intercambiar estos activos entre contrapartes ha aumentado considerablemente. A medida que miles de tokens nuevos son introducidos - incluyendo la tokenización de activos tradicionales - esta necesidad seguirá aumentando. Ya sea intercambiando tokens por motivos especulativos, o bien digitalizando los activos para acceder a las redes mediante el uso de tokens de utilidad nativos, la habilidad de intercambiar un criptoactivo por otro es fundamental para el ecosistema. Efectivamente, hay una energia potencial en los activos [5], y hacer efectiva esta energía - desbloqueando capital - requiere no sólo reivindicar la propiedad, que es lo que las cadenas de bloque han permitido indudablemente, sino también poder transferir y transformar estos activos libremente.

Por ello, el intercambio de tokens (o valor) libre de confianza es un ejemplo convincente de la tecnología de la cadena de bloques. Hasta ahora, sin embargo, los entusiastas de las criptomonedas se han conformado mayormente con intercambiar tokens en las casas de cambio centralizadas. El protocolo Loopring es necesario porque, tal como Bitcoin enfatizó responsablemente [6], en relación al dinero electrónico entre pares (del inglés, *peer-to-peer*), "las principales ventajas se pierden si la confianza en una tercera parte es aún necesaria para evitar el doble gasto". De la misma manera, las principales ventajas de los activos descentralizados se pierden si deben ser confiados y transferidos a través de casas de cambio centralizadas.

Intercambiar tokens descentralizados en casas de cambio centralizadas no tiene sentido desde un punto de vista lógico, puesto que falla en apoyar las virtudes que estos proyectos de descentralización defienden. El uso de casas de cambio centralizadas también presenta numerosos riesgos y limitaciones prácticas que se describen a continuación. Las casas de cambio descentralizadas (comúnmente referidas como DEX, del inglés, *decentralized exchange*) [7] [8] [9] han intentado hacer frente a estos problemas, en muchos casos disminuyendo los riesgos de seguridad mediante el uso de cadenas de bloques para la desintermediación. Sin embargo, todavía existe un margen de mejora considerable a medida que la capacidad de los DEX los conviertan en una infraestructura crucial para la nueva economía. Loopring pretende proveer las herramientas modulares para dicha infraestructura con su protocolo agnóstico y abierto para dApps.

2 Panorama actual de las casas de cambio

2.1 Deficiencias de las casas de cambio centralizadas

Los tres riesgos principales de las casas de cambio centralizadas son; 1) falta de seguridad, 2) falta de transparencia y 3) falta de liquidez.

La **falta de seguridad** surge cuando los usuarios ceden el control de sus claves privadas (y por consiguiente, sus fondos) a una entidad central. Esto expone a los usuarios al riesgo de que las casas de cambio centralizadas caigan en manos de *hackers* maliciosos. Los riesgos de seguridad y *hackeo* a los que toda casa de cambio centralizada se expone son bien conocidos [10] [11], y aún así comúnmente aceptados como un "riesgo implícito" en el intercambio de criptomonedas. Las casas de cambio centralizadas continuan siendo un centro de atención para los *hackers*, ya que sus servidores controlan millones de dolares en fondos de sus usuarios. Los desarrolladores de estas casas de cambio también pueden cometer errores accidentales y sin maldad con dichos fondos. Sencillamente, los usuarios no tienen el control de sus propios tokens cuando éstos son depositados en una casa de cambio centralizada.

La **falta de transparencia** expone a los usuarios al riesgo de que las casas de cambio descentralizadas deshonestas actúen injustamente. La diferencia aquí reside en las intenciones maliciosas de los operadores de estas casas de cambio. En las casas de cambio centralizadas, los usuarios no están intercambiando sus propios activos, sino un IOU (del inglés, *I owe you*), un pagaré o instrumento de deuda informal. Cuando los tokens se envian a la cartera virtual de la casa de cambio, ésta adquiere la custodia de los tokens y ofrece unos IOU en su lugar. Todos los intercambios son efectivamente entre los IOU de los usuarios. A la hora de realizar retiros, los usuarios canjean sus IOU con la casa de cambio, y reciben sus tokens en su dirección externa de monedero. Es a través de este proceso donde hay una falta de transparencia; una casa de cambio puede cerrar, congelar su cuenta, declararse en quiebra, etc. También es posible que usen los activos de los usuarios para otros propósitos mientras están en su custodia, como por ejemplo, para prestarlos a terceras partes. La falta de transparencia puede costar a los usuarios una pérdida total de sus fondos, impuestos de transacción más altos, retrasos durante máxima demanda, riesgo regulatorio u órdenes expuestas a inversión ventajista.

Falta de liquidez. Desde el punto de vista de los operadores de casas de cambio, la liquidez fragmentada impide la entrada de nuevas casas de cambio debido a dos escenarios de "ganador absoluto". En primer lugar, la casa de cambio con el mayor número de pares de criptomonedas gana, debido a que los usuarios prefieren realizar todas sus operaciones en una sola casa de cambio. En segundo lugar, la casa de cambio con el libro de órdenes más grande gana, debido a que posee una horquilla de precios (comúnmente denominado en inglés *spread*) favorable para cada par de criptomonedas. Esto disuade la competencia por parte de nuevas casas de cambio, puesto que es difícil para éstas crear liquidez inicial. Como resultado, muchas casas de cambio siguen poseyendo una cuota de mercado alta a pesar de las quejas de sus usuarios y *hackeos* importantes ocurridos. Merece la pena apuntar que, a medida que una casa de cambio centralizada gana cuota de mercado, ésta se convierte en un objectivo para *hackers* mayor.

Desde el punto de vista de los usuarios, la liquidación fragmentada reduce considerablemente la experiencia del usuario. En una casa de cambio centralizada, los usuarios únicamente son capaces de comerciar dentro de los fondos de liquidez de la propia casa de cambio, con su propio libro de órdenes y entre los pares de criptomonedas a los que se da soporte. Para intercambiar el token `A` por el token `B`, los usuarios deben, o bien acudir a una casa de cambio que de soporte a ambos tokens, o bien registrarse en diferentes casas de cambio, con la consiguiente divulgación adicional de información personal. A menudo, los usuarios necesitan realizar intercambios intermedios o iniciales, normalmente BTC o ETH, pagando la correspondiente horquilla en el proceso. Además, el libro de órdenes puede no ser lo suficientemente grande para completar un intercambio sin realizar deslizamiento (en inglés *slippage*). Incluso si la casa de cambio parece procesar un volumen grande, no hay garantía de que ese volumen y liquidez no sean falsos [12].

El resultado es un conjunto de silos de liquidez desconectados y un ecosistema fragmentado que se asemeja al antiguo sistema financiero, con los únicos volúmenes importantes centralizados en unas pocas casas de cambio. Las promesas de liquidez global de la cadena de bloques no tienen ninguna virtud en las casas de cambio centralizadas.

2.2 Deficiencias de las casas de cambio descentralizadas

Las casas de cambio descentralizadas difieren parcialmente de las casas de cambio centralizadas en que los usuarios mantienen el control de sus claves privadas, y por tanto sus activos, al realizarse los intercambios directamente en las cadenas de bloques subyacentes. Mediante el uso de la tecnología libre de confianza de las criptomonedas mismas, estas casas de cambio mitigan muchos de los riesgos de seguridad mencionados anteriormente. Sin embargo, aún existen problemas relacionados con el rendimiento y las limitaciones estructurales.

La liquidez sigue siendo un problema frecuente, ya que los usuarios deben buscar contrapartes a través de fondos de liquidez y estándares dispersos. Los efectos de la liquidación fragmentada estarán presentes si la mayoría de DEX o dApps no se valen de estándares comunes para interoperar, o si las órdenes no se comparten o envían a traves de una red lo suficientemente amplia. La liquidez de los libros de órdenes limitadas, y especialmente su resiliencia – cuán rapido se regeneran estas órdenes limitadas – puede afectar

considerablemente a las estrategias de comercio [13]. La ausencia de estos estándares ha resultado, no sólo en una liquidez reducida, sino también en una exposición a una variedad de contratos inteligentes potencialmente inseguros.

Además, como los intercambios se realizan en la cadena, los DEX heredan las limitaciones de la cadena de bloques subyacente, a saber: escalabilidad, retrasos en la ejecución (minado), y modificaciones de órdenes costosas. Debido a ello, los libros de órdenes en las cadenas de bloques no escalan especialmente bien, puesto que ejecutar código en la cadena de bloques incurre en un coste (gas), y esto hace que el coste de una cancelación de órdenes múltiples sea prohibitivo.

Finalmente, debido a que los libros de órdenes en las cadenas de bloques son públicos, la transacción para realizar una orden es visible por los mineros mientras espera a ser minada en el siguiente bloque y colocada en el libro de órdenes. Este retraso expone al usuario al riesgo de sufrir una inversión ventajista que mueva el precio en contra suya.

2.3 Soluciones híbridas

Por las razones antes expuestas, las casas de cambio basadas estrictamente en la cadena de bloques presentan limitaciones que les impiden competir con las casas de cambio centralizadas. Hay una solución intermedia entre la libertad de confianza inherente a la cadena de bloques y la velocidad y flexibilidad de órdenes de una casa de cambio centralizada. Protocolos como Loopring y 0x [14] ofrecen una solución intermedia de liquidación de intercambios dentro de la cadena con una gestión de órdenes fuera de la cadena. Estas soluciones giran en torno a contratos inteligentes abiertos, pero sortean ciertas limitaciones de escalabilidad ejecutando varias funciones fuera de la cadena y dando a los nodos flexibilidad en el cumplimiento de roles críticos para la red. Sin embargo, los inconvenientes de un modelo híbrido se mantienen [15]. El protocolo Loopring propone, a través de este artículo, diferencias significativas en nuestro enfoque de una solución híbrida.

3 Protocolo Loopring

Loopring no es un DEX, sino un protocolo modular para la creación de DEX en multiples cadenas de bloques. Desensamblamos las partes que componen una casa de cambio tradicional y las ofrecemos como un conjunto público de contratos inteligentes y actores descentralizados en su lugar. Los roles de éstos en la red incluyen carteras virtuales, relés, consorcio de compartición de líquidez en la cadena de bloques, exploración del libro de órdenes, mineros de anillo, y servicios de tokenización de activos. Previamente a definir cada uno de ellos, debemos entender primero las órdenes de Loopring.

3.1 Anillo de órdenes

Las órdenes de Loopring se basan en lo que definimos como un modelo de órden unidireccional (MOU)[16].El MOU expresa las órdenes como peticiones de intercambio de tokens, `cantidadV`/`cantidadC`, (cantidad a vender/comprar) en lugar de precios de compra y venta. Como cada orden es simplemente un tipo de cambio entre dos tokens, una característica potente del protocolo es la mezcla y combinación de múltiples órdenes en un intercambio circular. Al usar hasta 16 órdenes en lugar de un sólo par de criptomonedas, esto permite que haya un aumento drástico de la liquidez y la posibilidad de una mejora del precio.

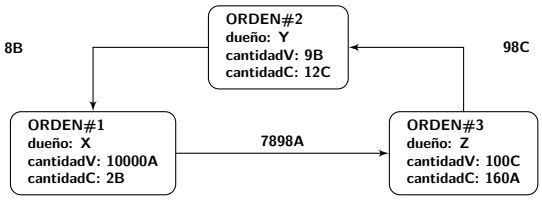

Figura 1: Un anillo compuesto de 3 órdenes

La figura anterior muestra un anillo compuesto de 3 órdenes. Cada orden de venta de un token (`tokenV`) es otra orden de compra del token (`tokenC`). Esto crea un bucle que permite a cada orden intercambiar los tókenes deseados sin requerir de una órden opuesta para su par de criptomonedas. Los intercambios mediante par de criptomonedas tradicionales aún pueden ser realizados, en lo que básicamente es un caso especial de anillo de órdenes.

Definición 3.1 (anillo de órdenes) *Sean C_0, C_1, $\cdots$, C_{n-1} n tokens distintos, y sean $O_{0\to1}$, $\cdots$, $O_{i\to i\oplus1}$, $\cdots$, $O_{n-1\to0}$ n órdenes distintas. Estas órdenes pueden formar un anillo de órdenes para el siguiente intercambio:*

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

donde n es la longitud del anillo, y $i \oplus 1 \equiv i + 1 \mod n$.

Un anillo de órdenes es válido cuando todas las transacciones que la componen pueden realizarse a un tipo de cambio igual o mejor que el valor original especificado implícitamente por el usuario. Para verificar la validez del anillo de órdenes, los contratos inteligentes del protocolo Loopring deben recibir anillos de órdenes de mineros de anillos cuyo producto del tipo de cambio original de todas las órdenes es igual o mayor que 1.

Suponiendo que Alicia y Bob quieren cambiar sus tokens `A` y `B`. Alicia tiene 15 tokens `A` y quiere 4 tokens `B` por ellos; Bob tiene 10 tokens `B` y quiere 30 tokens `A` por ellos.

¿Quién de los dos está comprando y quién está vendiendo? Esto depende únicamente del activo en que nos basemos para determinar el precio. Si el token `A` es la referencia, entonces Alicia está comprando el token `B` a un precio de $\frac{15}{4} = 3.75$`A`, mientras que Bob está vendiendo 10 tokens `B` a un precio de $\frac{30}{10} = 3.00$`A`. En el caso de

que escojamos el token B como referencia, diremos entonces que Alicia está vendiendo 15 tokens A a un precio de $\frac{4}{15} = 0.26666667$B y Bob está comprando 10 tokens A a un precio de $\frac{10}{30} = 0.33333334$B. Por tanto, quién es el comprador o vendedor es algo puramente arbitrario.

En el primer escenario, Alicia está dispuesta a pagar un precio más alto (3.75A) que el precio de venta de Bob por sus tokens (3.00A), mientras que en el segundo escenario Bob está dispuesto a pagar un precio más alto (0.33333334B) que el precio de venta de Alicia por sus tokens (0.26666667B). Queda claro que un intercambio es posible cuando el comprador está dispuesto a pagar un precio igual o más alto que el precio del vendedor.

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

Por consiguiente, para que un conjunto de n órdenes pueda ejecutarse, bien parcialmente, parcialmente o completamente, se necesita saber si el producto de cada uno de los tipos de cambios de las órdenes de compra es mayor que 1. De ser así, todas las órdenes n pueden ser parcialmente o completamente ejecutadas [17].

Si introducimos una tercera parte, Charlie, tal que ahora Alicia quiere dar x_1 tokens A y recibir y_1 tokens B, Bob quiere dar x_2 tokens B y recibir y_2 tokens C, y Charlie quiere dar x_3 tokens C y recibir y_3 tokens A. Existen los tokens necesarios, y el intercambio es posible si:

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

La sección 7.1 incluye más detalles sobre las órdenes de Loopring.

4 Participantes del ecosistema

Los siguientes participantes del ecosistema ofrecen de manera conjunta todas las funcionalidades que una casa de cambio centralizada puede ofrecer:

- **Carteras**: Un servicio común de cartera o interfaz que da a los usuarios acceso a sus tokens y una manera de enviar órdenes a la red de Loopring. Las carteras serán incentivadas a producir órdenes mediante la compartición de tasas con los mineros de anillo (lea la sección 8). Con la creencia de que el futuro del intercambio tendrá lugar dentro de la seguridad de las carteras inviduales de los usuarios, es crucial conectar estos fondos de liquidez a través de nuestro protocolo.

- **Cadena de bloques de consorcio para la compartición de liquidez/malla de relés**: Una malla de relés para la compartición de órdenes y liquidez. Cuando los nodos ejecutan el código de relé de Loopring, éstos pueden unirse a redes existentes y compartir la liquidez con otros relés a traves de una cadena de bloques de consorcio. Esta primera versión de la cadena de bloques de consorcio que estamos construyendo permite la compartición de órdenes casi en tiempo real (bloques de 1-2 segundos) y reduce el historial antiguo para permitir descargas más rápidas por parte de los nodos nuevos. Cabe destacar que los relés no necesitan unirse a este consorcio; pueden actuar de forma individual sin compartir liquidez con otros o crear y gestionar su propia red de compartición de liquidez.

- **Relés/Mineros de anillos**: Los relés son nodos que reciben órdenes de las carteras y de la malla de relés, mantienen los libros de órdenes e historial de intercambios públicos, y de manera opcional, retransmiten las órdenes a otros relés (mediante un medio externo arbitrario) y/o mallas de relés. El minado de anillos de órdenes es una característica – no un requisito – de los relés. Esta es una aplicación computacionalmente intensiva y se realiza completamente fuera de la cadena. Los relés con la capacidad de minar estos anillos, llamados "Mineros de anillos", son los que producen anillos uniendo órdenes dispersas. Los relés tienen libertad para (1) escoger cómo comunicarse entre ellos, (2) cómo crear los libros de órdenes, y (3) qué algoritmo de minado usar para minar los anillos de órdenes.

- **Contratos inteligentes del protocolo Loopring (CIPR)**: Un conjunto de contratos inteligentes público y abierto que comprueba los anillos de órdenes recibidos de los mineros de anillos, confirma y transfiere los tokens en nombre de los usuarios, recompensa a los mineros de anillos y las carteras con las tasas de transacción y emite eventos. Los relés y exploradores de órdenes escuchan estos eventos para mantener sus libros de órdenes e historial de intercambios actualizados.

- **Servicios de tokenización de activos (STA)**: Una pasarela para los activos que no pueden ser intercambiados directamente en Loopring. Como por ejemplo servicios centralizados gestionados por compañías u organizaciones de confianza. Los usuarios depositan activos (reales, dinero fíat o tokens de otras cadenas) y a cambio reciben tokens que pueden ser canjeados en el futuro. Loopring no es un protocolo de intercambio entre-cadenas (mientras no exista una solución adecuada para ello), pero el STA permite el intercambio de tokens ERC20 [18] por activos físicos o activos digitales de otras cadenas de bloques.

5 Proceso de intercambio

1. **Autorización del protocolo**: En la figura 2, el usuario `Y` que quiere cambiar sus tokens autoriza al CIPR a gestionar una `amountS` de tokens `B` para su venta. Esto no congela los tokens del usuario, que siguen libres de ser movidos mientras la orden no se ejecute.

2. **Creación de la orden**: El tipo de cambio y el libro de órdenes para el par token`B`/token`C` viene dado por los relés u otros agentes conectados a la red, como por ejemplo los exploradores de libros de órdenes. El usuario `Y` crea una orden (orden límite) especificando la `cantidadV`, la `cantidadC` y otros parámetros a través de la interfaz de cartera virtual integrada. Una cantidad de LRx puede ser añadida a la órden como tasa para los mineros de anillos; una cantidad mayor de LRx supone una mayor probabilidad de que los mineros procesen la orden más rapidamente. El *hash* de la orden se firma con la clave privada del usuario `Y`.

3. **Envío de la orden**: La cartera virtual envía la orden y su firma a uno o más relés. Estos actualizan su libro de órdenes público. El protocolo no requiere que estos libros estén estructurados de una manera específica; por ejemplo, en orden de llegada. En su lugar, los relés tienen el poder de tomar sus propias decisiones al construir sus libros de órdenes.

4. **Compartición de la liquidez**: Los relés envían la orden a otros relés a traves de un medio de comunicación arbitrario. De nuevo, se ofrece una flexibilidad en la manera de definir cómo interactuan los nodos entre sí. Para facilitar un cierto nivel de conectividad de la red, existe una malla de relés fija para la compartición de liquidez dentro de la cadena de bloques de consorcio. Como se ha mencionado anteriormente, esta malla de relés ha sido optimizada para ser inclusiva y rápida.

5. **Minado del anillo (Combinación de órdenes)**: Los mineros de anillos intentan ejecutar las órdenes parcial o completamente a un tipo de cambio igual o mejor emparejándola con una u otras órdenes. El minado de anillos es la razón principal de por qué el protocolo es capaz de ofrecer alta liquidez en cualquier par de criptomonedas. Si el tipo de cambio al que la órden se ejecutó es mejor que la que especificó el usuarioIf the executed rate is better than what user `Y`, la diferencia es compartida entre todas las órdenes del anillo. Como recompensa, el minero del anillo escoge entre adjudicarse parte de esa diferencia (`marginSplitPercentage`) y devolver la tasa de transacción en LRx al usuario, o sencillamente quedarse con la tasa de transacción en LRx.

6. **Verificación y acuerdo**: El anillo de órdenes es recibido por el CIPR. Este realiza múltiples comprobaciones para verificar los datos proporcionados por el minero del anillo y determina si el anillo puede ser if the order-ring can be settled fully or partially (depending on the fill rate of orders in-ring and tokens in users' wallets). If all checks are successful, the contract atomically transfers the tokens to users and pays the ring-miner and wallet fees at the same time. If user `Y`'s balance as determined by the LPSC is insufficient, it will be considered scaled-down: a scaled-down order will automatically scale up to its original size if sufficient funds are deposited to its address, unlike a cancellation, which is a one way manual operation and cannot be reversed.

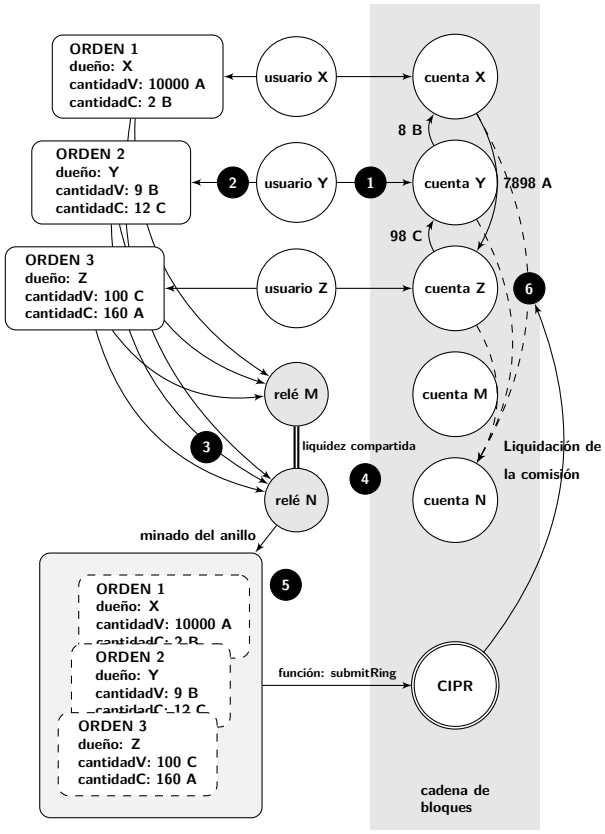

Figura 2: Proceso de intercambio con Loopring

6 Operational Flexibility

It's important to note that Loopring's open standard allows participants significant flexibility in how they operate. Actors are free to implement novel business models and provide value for users, earning LRx fees on volume or other metrics in the process (if they so choose). The ecosystem is modular and meant to support participation from a multitude of applications.

6.1 Order Book

Relays can design their order books in any number of ways to display and match users' orders. A first implementation

of our own order book follows an OTC model, where limit orders are positioned based on price alone. Timestamps of orders, in other words, have no bearing on the order book. However, a relay is free to design their order book in such a way as to emulate a typical centralized exchange's matching engine, where orders are ranked by price, while respecting timestamps as well. If a relay was inclined to offer this type of order book, they can own/integrate with a wallet, and have those wallet orders sent solely to the single relay, who would then be able to match orders based on time. Any such configuration is possible.

Whereas other DEX protocols at times require Relays to have resources - initial token balances to place taker orders - Loopring Relays need only find matchable orders to consummate a trade, and can do so without initial tokens.

6.2 Liquidity Sharing

Relays are free to design how they share liquidity (orders) with each other. Our consortium blockchain is but one solution to accomplish this, and the ecosystem is free to network and communicate as they wish. Besides joining a consortium blockchain, they can build and manage their own, creating rules/incentives as they see fit. Relays can also work alone, as seen in the time-sensitive wallet implementation. Of course, there are clear advantages in communicating with other Relays in pursuit of network effects, however, different business models could merit peculiar sharing designs and split fees in any number of ways.

7 Protocol Specification

7.1 Anatomy of an Order

An order is a pack of data that describes the intent of the user's trade. A Loopring order is defined using the Uni-Directional Order Model, or UDOM, as follows:

```
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;
}
```

To ensure the origin of the order, it is signed against the hash of its parameters, excluding `authAddr`, with the user's private-key. The `authAddr` parameter is used for signing order-rings that this order is part of, which prevents front-running. Please reference section 9.1 for more details. The signature is represented by the `v`, `r`, and `s` fields, and is sent alongside the order parameters over the network. This guarantees the order stays immutable during its whole lifetime. Even though the order never changes, the protocol can still compute its current state based on the balance of its address along with other variables.

UDOM doesn't include a price (which must be a floating-point number by nature), but, instead uses the term `rate` or r, which is expressed as `amountS`/`amountB`. The rate is not a floating-point number but an expression that will only be evaluated with other unsigned integers on demand, to keep all intermediate results as unsigned integers and increase calculation accuracy.

7.1.1 Buy Amounts

When a ring-miner ring-matches orders, it's possible that a better rate will be executable, allowing users to get more `tokenB` than the `amountB` they specified. However, if `buyNoMoreThanAmountB` is set to `True`, the protocol ensures users receive no more than `amountB` of `tokenB`. Thus, UDOM's `buyNoMoreThantokenB` parameter determines when an order is considered completely filled. `buyNoMoreThantokenB` applies a cap on either `amountS` or `amountB`, and allows users to express more granular trade intentions than traditional buy/sell orders.

For example: with `amountS` = 10 and `amountB` = 2, the rate $r = 10/2 = 5$. Thus the user is willing to sell 5 `tokenS` for each `tokenB`. The ring-miner matches and finds the user a rate of 4, allowing the user to receive 2.5 `tokenB` instead of 2. However, if the user only wants 2 `tokenB` and set the `buyNoMoreThanAmountB` flag to `True`, the LPSC performs the transaction at a rate of 4 and the user sells 4 `tokenS` for each `tokenB`, effectively saving 2 `tokenS`. Keep in mind this does not take into account mining fees (See section 8.1).

Indeed, if we use

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

to represent an order in a simplified form, then for ETH/USD markets on a traditional exchange, traditional buy-sell modeling can express the 1st and the 3rd order below, but not the other two:

1. Sell 10 ETH at price of 300 USD/ETH. This order can expressed as: `Order(10, ETH, 3000, USD, False)`.

2. Sell ETH at price of 300 USD/ETH to get 3000 USD. This order can expressed as: `Order(10, ETH, 3000, USD, True)`.

3. Buy 10 ETH at price of 300 USD/ETH, This order can expressed as: `Order(3000, USD, 10, ETH, True)`.

4. Spend 3000 USD to buy as many ETH as possible at price of 300 USD/ETH, This order can expressed as: `Order(3000, USD, 10, ETH, False)`.

7.2 Ring Verification

The Loopring Smart Contracts do not perform exchange rate or amount calculations, but must receive and verify what the ring-miners supply for these values. These calculations are done by ring-miners for two main reasons: (1) the programming language for smart contracts, such as solidity[19] on Ethereum, does not have support for floating point math, especially $pow(x, 1/n)$ (calculating the n-th root of a floating point number), and (2) it is desirable for the computation to be made off-chain to reduce blockchain computation and cost.

7.2.1 Sub-Ring Checking

This step prevents arbitrageurs from unfairly realizing all the margin in an order-ring by implementing new orders within it. Essentially, once a valid order-ring is found by a ring-miner, it could be tempting to add other orders to the order-ring to fully absorb the users' margin (rate discounts). As illustrated by figure 3 below, carefully calculated $x1$, $y1$, $x2$ and $y2$ will make the product of all orders' rate be exactly 1 so there will be no rate discount.

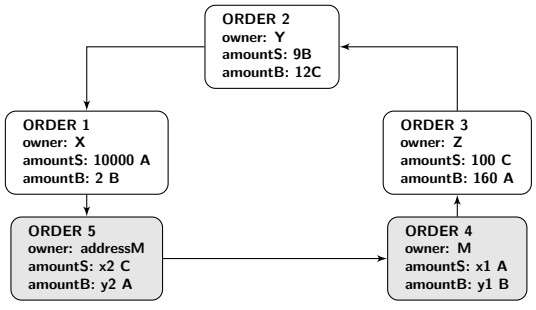

Figura 3: An order-ring with sub-ring

This is zero-risk, zero-value add to the network, and is considered unfair conduct by the ring-miner. To prevent this, Loopring requires that a valid loop cannot contain any sub-rings. To check this, the LPSC ensures a token cannot be in a buy or sell position twice. In the above diagram, we can see that token `A` is a sell token twice and a buy token twice, which would be disallowed.

7.2.2 Fill Rate Checking

The exchange rate calculations in the order-ring are made by ring-miners for reasons stated above. It is the LPSC that must verify they're correct. First, it verifies that the buy rate the ring-miner can execute for each order is equal to or less than the original buy rate set by the user. This ensures the user gets at least the exchange rate they asked for or better on the transaction. Once the exchange rates are confirmed, the LPSC ensures that each order in the order-ring shares the same rate discount. For instance, if the discounted rate is γ, then the price for each order will be:

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

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

hence:

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

If the transaction crosses n orders, the `discount` is:

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

where r^i is the order turnover rate of i-th order. Obviously, only when the discount rate is $\gamma \geq 0$, can these orders be filled; and the i-th order (O^i)'s actual exchange rate is $\hat{r^i} = r^i\cdot(1-\gamma)$, $\hat{r^i} \leq r^i$.

Recall our prior example where Alice has 15 token `A` and wants 4 token `B` for them, Bob has 10 token `B` and wants 30 token `A` for them. If token `A` is the reference, then Alice is buying token `B` for $\frac{15}{4} = 3.75$`A`, while Bob is selling token `B` for $\frac{30}{10} = 3.00$`A`. To calculate the discount: $\frac{150}{120} = 1.25$ thus $\frac{1}{1.25} = 0.8 = (1-\gamma)^2$. Thus the exchange rate that renders the trade equitable for both parties is $\sqrt{0.8}\cdot 3.75 \approx 3.3541$ token `A` per token `B`.

Bob gives 4 token `B` and receives 13.4164 token `A`, more than the 12 he was expecting for those 4 tokens. Alice receives 4 token `B` as intended but gives only 13.4164 token `A` in exchange, less than the 15 she was willing to give for those 4 tokens. Note, a portion of this margin will go towards paying fees to incentivize miners (and wallets). (See section 8.1).

7.2.3 Fill Tracking & Cancellation

A user can partially or fully cancel an order by sending a special transaction to the LPSC, containing the details about the order and the amounts to cancel. The LPSC takes that into account, stores the amounts to cancel, and emits an `OrderCancelled` event to the network. The LPSC keeps track of filled and cancelled amounts by storing their values using the order's hash as an identifier. This data is publicly accessible and `OrderCancelled` / `OrderFilled` events are emitted when it changes. Tracking these values is critical for the LPSC during the order-ring settlement step.

LPSC also supports cancelling all orders for any trading pair with the `OrdersCancelled` event and cancelling all orders for an address with the `AllOrdersCancelled` event.

7.2.4 Order Scaling

Orders are scaled according to the history of filled and cancelled amounts and the current balance of the senders' accounts. The process finds the order with the smallest amount to be filled according to the above characteristics and uses it as a reference for scaling all transactions in the order-ring.

Finding the lowest value order can help to figure out the fill volume for each order. For instance, if the i-th order is the lowest value order, then the number of tokens sold from each order $\hat{s}$ and number of tokens purchased $\hat{b}$ from each order can be calculated as:

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

where $\overline{s}_i$ is the balance left after orders are partially filled.

During implementation we can safely assume any order in the order-ring to have the lowest value, then iterate through the order-ring at most twice to calculate each order's fill volume.

Example: If the smallest amount to be filled compared to the original order is 5%, all the transactions in the order-ring are scaled down to 5%. Once the transactions are completed, the order that was considered to have the smallest amount remaining to be filled should be completely filled.

7.3 Ring Settlement

If the order-ring fulfills all the previous checks, the order-ring can be closed, and transactions can be made. This means that all n orders form a closed order-ring, connected as in figure 4:

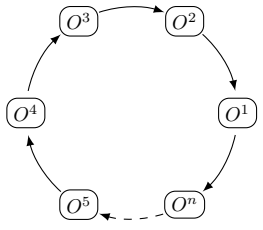

Figura 4: Ring Settlement

To make the transactions, the LPSC uses the `TokenTransferDelegate` smart contract. The introduction of such a delegate makes upgrading the protocol smart contract easier as all orders only need to authorize this delegate instead of different versions of the protocol.

For each order in the order-ring, a payment of `tokenS` is made to the next or the previous order depending on the implementation. Then the ring-miner's fee is paid depending on the fee model chosen by the ring-miner. Finally, once all the transactions are made, a `RingMined` event is emitted.

7.3.1 Emitted Events

The protocol emits events that allow relays, order browsers, and other actors to receive order book updates as efficiently as possible. The emitted events are:

- **OrderCancelled**: A specific order has been cancelled.
- **OrdersCancelled**: All orders of a trading pair from an owning address have been cancelled.
- **AllOrdersCancelled**: All orders of all trading pairs from an owning address have been cancelled.
- **RingMined**: An order-ring has been settled successfully. This event contains data related to each inner-ring token transfer.

8 LRx Token

LRx is our generalized token notation. LRC is the Loopring token on Ethereum, LRQ on Qtum, and LRN on NEO, etc. Other LRx types will be introduced in the future as Loopring is deployed on other public blockchains.

8.1 Fee Model

When a user creates an order, they specify an amount of LRx to be paid to the ring-miner as a fee, in conjunction with a percentage of the margin (`marginSplitPercentage`) made on the order that the ring-miner can claim. This is called the margin split. The decision of which one to choose (fee or margin split) is left to the ring-miner.

A representation of the margin split:

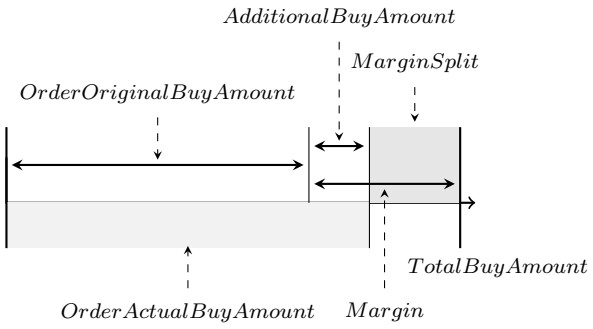

Figura 5: A 60% Margin Split

If the margin on the order-ring is too small, a ring-miner will choose the LRx fee. If, on the contrary, the margin is substantial enough for the resulting margin split to be worth much more than the LRx fee, a ring-miner will choose the margin split. There is another proviso, however: when the ring-miner chooses the margin split, they must pay the user (order creator) a fee, which is equal to the LRx the user would have paid to the ring-miner as a fee. This increases the threshold of where the ring-miner will choose the margin split to twice the LRx fee of the order, increasing the propensity of the LRx fee choice. This allows ring-miners to receive a constant income on low margin order-rings for the tradeoff of receiving less income on higher margin order-rings. Our fee model is based on the expectation that as the market grows and matures, there will be fewer high margin order-rings, thus necessitating fixed LRx fees as incentive.

We end up with the following graph:

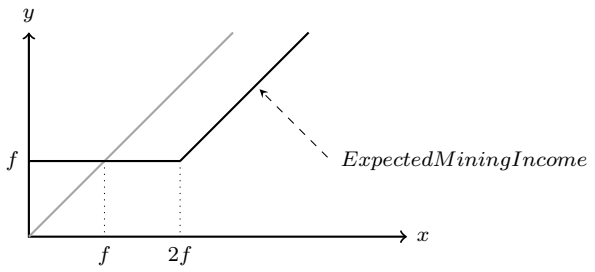

Figura 6: Loopring's Fee Model

where f is the LRx fee, x is the margin split, y is the mining income. $y = max(f, x - f)$ as indicated by the solid line; if the LRx fee for the order is 0, the equation is $y = max(0, x - 0)$ that simplifies to $y = x$ as indicated by the gray line.

The consequences are:

1. If the margin split is 0, ring-miners will choose the flat LRx fee and are still incentivized.

2. If the LRx fee is 0, the gray line results and the income is based on a general linear model.

3. When the margin split income is greater than 2x(LRx fee), ring-miners choose the margin split and pay LRx to the user.

It should be noted that if the LRx fee is non-zero, no matter which option the ring-miner chooses, there will always be a transfer of LRx between the ring-miner and the order's sender. Either the ring-miner earns the LRx fee, or pays the LRx fee back to the sender to take the margin split.

Ring-miners will share a certain percentage of fees with wallets. When a user places an order through a wallet and gets filled, the wallet is rewarded with a portion of the fees or margin split. Although this is modular, and unique business models or implementations are possible, our inclination is for wallets to receive approximately 20%-25% of earned fees. Wallets represent a primary target for Loopring protocol integration as they have the user base, but little or no source of income.

8.2 Decentralized Governance

At its root, the Loopring protocol is a social protocol in the sense that it relies on coordination amongst members to operate effectively towards a goal. This is not dissimilar to cryptoeconomic protocols at large, and indeed, its usefulness is largely protected by the same mechanisms of coordination problems [20], grim trigger equilibrium, and bounded rationality. To this end, LRx tokens are not only used to pay fees, but also to align the financial incentives of the various network participants. Such alignment is necessary for broad adoption of any protocol, but is particularly acute for exchange protocols, given that success rests largely on improving liquidity in a robust decentralized ecosystem.

LRx tokens will be used to effectuate protocol updates through decentralized governance. Smart contract updates will, in part, be governed by token holders to ensure continuity and safety, and to attenuate the risks of siphoned liquidity through incompatibility. Given that smart contracts cannot be altered once deployed, there is a risk that dApps or end users continue to interact with deprecated versions and preclude themselves from updated contracts. Upgradeability is crucial to the protocol's success as it must adapt to market demands and the underlying blockchains. Decentralized governance by LRx stakeholders will allow for protocol smart contract updates without disrupting dApps or end users, or relying too heavily on smart contract abstraction. LRx tokens have a fixed supply, and in the case of LRC, certain percentages are frozen from the Loopring Foundation, and allocated to community-purposed funds [21].

However, LRx token owners are not the only stakeholders to consider in steering the protocol's direction: relays/ring-miners, wallets, developers, and others are an integral part of the ecosystem and their voice must be heard. In fact, given that these agents need not hold any LRx to perform their respective roles (since traditional makers/takers and market-makers are nonexistent, initial token reserves are not mandatory) we must allow alternative methods for their interests to be respected. Furthermore, "simple" token-based voting, both on-chain and off, is an imperfect salve for disagreement, as low voter turnout and token ownership concentration pose risks. Thus, the goal is to implement a governance model that is built in layers, and rests on a shared knowledge that some set of decision-making processes is the norm. This can be facilitated by coordination institutions that offer signals from a diverse set of participants, and, perhaps, from pre-established protocol focal points. As this comes to fruition, the Loopring Foundation will inevitably evolve from protocol developers into protocol stewards.

9 Fraud and Attack Protections

9.1 Front-running Prevention

In decentralized exchanges, front-running is when someone tries to copy another node's trade solution, and have it mined before the original transaction that is in the pending transaction pool (mempool). This can be achieved by specifying a higher transaction fee (gas price). The major scheme of front-running in Loopring (and any protocol for order-matching) are order-filch: when a front-runner steals one or more orders from a pending order-ring settlement transaction; and, specific to Loopring: when a front-runner steals the entire order-ring from a pending transaction.

When a submitRing transaction is not confirmed and is still in the pending transaction pool, anyone can easily spot such a transaction and replace `minerAddress` with their own address (the `filcherAddress`), then they can re-sign the payload with `filcherAddress` to replace the order-ring's signature. The filcher can set a higher gas price and submit a new transaction hoping block-miners will pick his new transaction into the next block instead of the original submitRing transaction.

Previous solutions to this problem had important downsides: requiring more transactions and thus costing ring-miners more gas; and taking at least twice the blocks to settle an order-ring. Our new solution, Dual Authoring[22], involves the mechanism of setting up two levels of authorization for orders - one for settlement, and one for ring-mining.

Dual Authoring process:

1. For each order, the wallet software will generate a random public-key/private-key pair, and put the key pair into the order's JSON snippet. (An alternative is to use the address derived from the public-key instead of the public-key itself to reduce byte size. We use `authAddr` to represent such an address, and `authKey` to represent `authAddr`'s matching private-key).

2. Compute the order's hash with all fields in the order except `r`, `v`, `s`, and `authKey`), and sign the hash using the `owner`'s private-key (not `authKey`).

3. The wallet will send the order together with the `authKey` to relays for ring-mining. Ring-miners will verify that `authKey` and `authAddr` are correctly paired and the order's signature is valid with respect to `owner` address.

4. When an order-ring is identified, the ring-miner will use each order's `authKey` to sign the ring's hash, `minerAddress`, and all the mining parameters. If an order-ring contains n orders, there will be n signatures by the n `authKeys`. We call these signatures the `authSignatures`. The ring-miner may also need to sign the ring's hash together with all mining parameters using `minerAddress`'s private-key.

5. The ring-miner calls the submitRing function with all the parameters, as well as all the extra `authSignatures`. Notice that `authKeys` are NOT part of the on-chain transaction and thus remain unknown to parties other than the ring-miner itself.

6. The Loopring Protocol will now verify each `authSignature` against the corresponding `authAddr` of each order, and reject the order-ring if any `authSignature` is missing or invalid.

The result is that now:

- The order's signature (by the private-key of the `owner` address) guarantees the order cannot be modified, including the `authAddr`.
- The ring-miner's signature (by the private-key of the `minerAddress`), if supplied, guarantees nobody can use his identity to mine an order-ring.
- The `authSignatures` guarantees the entire order-ring cannot be modified, including `minerAddress`, and no orders can be stolen.

Dual Authoring prevents ring-filch and order-filch while still ensuring the settlement of order-rings can be done in one single transaction. In addition, Dual Authoring opens doors for relays to share orders in two ways: non-matchable sharing and matchable sharing. By default, Loopring operates an OTC model and only supports limit-price orders, meaning that orders' timestamps are ignored. This implies that front-running a trade has no impact on the actual price of that trade, but does impact whether it gets executed or not.

10 Other Attacks

10.1 Sybil or DOS Attack

Malicious users – acting as themselves or forged identities – could send a large amount of small orders to attack Loopring nodes. However, since we allow nodes to reject orders based on their own criteria – which they may hide or reveal – most of these orders will be rejected for not yielding satisfying profit when matched. By empowering relays to dictate how they manage orders, we do not see a massive tiny order attack as a threat.

10.2 Insufficient Balance

Malicious users could sign and spread orders whose order value is non-zero but whose address actually has zero balance. Nodes could monitor and notice that some orders actual balance is zero, update these order states accordingly and then discard them. Nodes must spend time to update the status of an order, but can also choose to minimize the effort by, for example, blacklisting addresses and dropping related orders.

11 Summary

The Loopring protocol sets out to be a foundational layer for decentralized exchange. In so doing, it has profound repurcussions in how people exchange assets and value. Money, as an intermediate commodity, facilitates or replaces barter exchange and solves the double coincidence of wants problem [23], whereby two counterparties must desire each other's distinct good or service. Similarly, Loopring protocol aims to dispense of our dependencies on coincidence of wants in trading pairs, by using ring matching to more easily consummate trades. This is meaningful for how society and markets exchange tokens, traditional assets, and beyond. Indeed, just as decentralized cryptocurrencies pose threat to a nation's control over money, a combinatorial protocol that can match traders (consumers/producers) at scale, is a theoretical threat to the concept of money itself.

Protocol benefits include:

- Off-chain order management and on-chain settlement means no sacrifice in performance for security.
- Greater liquidity due to ring-mining and order sharing.
- Dual Authoring solves the pernicious problem of front running faced by all DEXs and their users today.
- Free, public smart contracts enable any dApp to build or interact with the protocol.
- Standardization among operators allows for network effects and an improved end user experience.
- Network maintained with flexibility in running order books and communicating.
- Reduced barriers to entry means lower costs for nodes joining the network and end users.
- Anonymous trading directly from user wallets.

12 Acknowledgements

We would like to express our gratitude to our mentors, advisers and to the many people in the community that have been so welcoming and generous with their knowledge. In particular, we would like to thank Shuo Bai (from ChinaLedger); Professor 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, and Encephalo Path for reviewing and providing feedback on this project.

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] Loopring Foundation. Lrc token documents. `https://docs.loopring.org/English/token/`, Accessed: 2018-03-05.

[22] 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.

[23] 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.