



(51) International Patent Classification:
Not classified

(21) International Application Number:
PCT/US2024/035970

(22) International Filing Date:
28 June 2024 (28.06.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/511,141 29 June 2023 (29.06.2023) US

(71) Applicant: **PRESIDENT AND FELLOWS OF HARVARD COLLEGE** [US/US]; 17 Quincy Street, Cambridge, MA 02138 (US).

(72) Inventors: **FERREIRA, Matheus Venturynne Xavier**; c/o Harvard University, Smith Center - Suite 727, 1350 Massachusetts Ave., Cambridge, MA 02138 (US). **PARKES,**

David C.; c/o Harvard University, Smith Center - Suite 727, 1350 Massachusetts Ave., Cambridge, MA 02138 (US).

(74) Agent: **RUDOY, Daniel G.** et al.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210-2206 (US).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV,

(54) Title: CREDIBLE EXCHANGE DESIGN VIA A VERIFIABLE SEQUENCING RULE

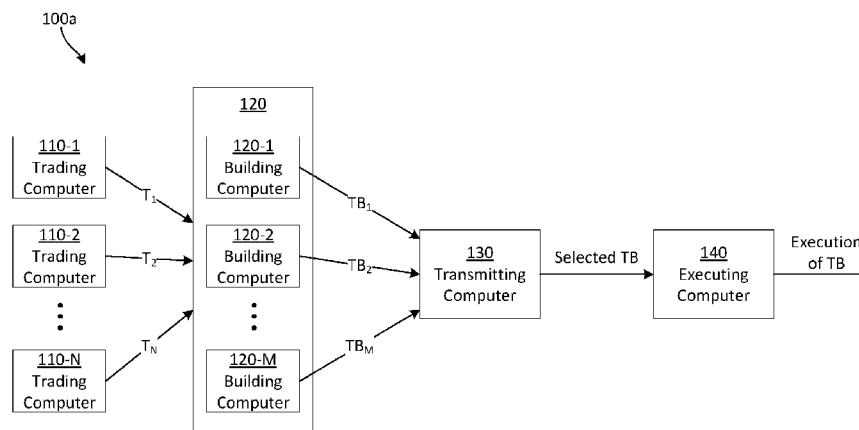


FIG. 1A

(57) Abstract: Systems and methods are provided for implementing a verifiable transaction block on an exchange, the verifiable transaction block including transactions (e.g., purchases and/or sales) that are to be executed on the exchange in accordance with an ordering of the transactions. The techniques provided include ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions, and causing the execution of the ordered transactions. The transactions may indicate a purchase and/or a sale of at least one token of two or more tokens on a liquidity pool.

GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

CREDIBLE EXCHANGE DESIGN VIA A VERIFIABLE SEQUENCING RULE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit under 35 U.S.C. § 119(e) of U.S. Provisional Patent Application No. 63/511,141, filed June 29, 2023, and titled “Credible Exchange Design Via a Verifiable Sequencing Rule,” which is incorporated herein by reference in its entirety.

BACKGROUND

[0002] Decentralized finance offers the ability to perform financial transactions without relying on conventional intermediaries such as brokerages or banks. Decentralized financial transactions are typically performed on distributed ledgers (e.g., blockchains), with transactions being made directly between participants or mediated by smart contract programs.

SUMMARY

[0003] In some aspects, the techniques described herein relate to a computer-implemented method for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the computer-implemented method including: receiving transaction information specifying a set of transactions, wherein: transactions of the set of transactions are to be ordered prior to execution, and each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens; ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions; and causing the execution of the ordered transactions.

[0004] In some aspects, the techniques described herein relate to a computer-implemented method, wherein causing the execution of the ordered transactions includes transmitting, from a first computer to a second computer, the transaction block for execution.

[0005] In some aspects, the techniques described herein relate to a computer-implemented method, wherein the verifiable sequencing rule is arranged to ensure that: (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction, (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0006] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes, if the set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

[0007] In some aspects, the techniques described herein relate to a computer-implemented method, further including, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block: removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

[0008] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes: if the updated set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0009] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes, if the updated set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

[0010] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes: if the set of transactions includes only one of purchases or

sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0011] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions further includes executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

[0012] In some aspects, the techniques described herein relate to at least one non-transitory computer readable medium storing processor-executable instructions that, when executed by at least one processor, cause the at least one processor to perform a method of implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the method including: receiving transaction information specifying a set of transactions, wherein: transactions of the set of transactions are to be ordered prior to execution, and each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens; ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions; and causing the execution of the ordered transactions.

[0013] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, wherein causing the execution of the ordered transactions includes transmitting, from the at least one processor to another at least one processor, the transaction block for execution.

[0014] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, wherein the verifiable sequencing rule is arranged to ensure that: (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction, (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0015] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes, if the set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of

the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

[0016] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, further including, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block: removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

[0017] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes: if the updated set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0018] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes, if the updated set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

[0019] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes: if the set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0020] In some aspects, the techniques described herein relate to an at least one non-transitory computer readable medium, wherein ordering the transactions of the set of transactions further

includes executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

[0021] In some aspects, the techniques described herein relate to a system for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the system including: at least one processor; and at least one non-transitory computer readable medium storing processor-executable instructions that, when executed by the at least one processor, cause the at least one processor to perform a method including: receiving transaction information specifying a set of transactions, wherein: transactions of the set of transactions are to be ordered prior to execution, and each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens; ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions; and causing the execution of the ordered transactions.

[0022] In some aspects, the techniques described herein relate to a system, wherein causing the execution of the ordered transactions includes transmitting, from the at least one processor to another at least one processor, the transaction block for execution.

[0023] In some aspects, the techniques described herein relate to a system, wherein the verifiable sequencing rule is arranged to ensure that: (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction, (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0024] In some aspects, the techniques described herein relate to a system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes, if the set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

[0025] In some aspects, the techniques described herein relate to a system, wherein ordering the transactions of the set of transactions further includes, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block: removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

[0026] In some aspects, the techniques described herein relate to a system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes: if the updated set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0027] In some aspects, the techniques described herein relate to a system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes, if the updated set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

[0028] In some aspects, the techniques described herein relate to a system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes: if the set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0029] In some aspects, the techniques described herein relate to a system, wherein ordering the transactions of the set of transactions further includes executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

[0030] In some aspects, the techniques described herein relate to a computer-implemented method for implementing a verifiable transaction service on a decentralized exchange of a liquidity pool of two or more tokens, the computer-implemented method including: transmitting, from trading computers to one or more building computers, transaction information specifying a

set of transactions, wherein: transactions of the set of transactions are to be ordered prior to execution, and each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens; ordering, at each of the one or more building computers and using a verifiable sequencing rule, transactions of the set of transactions to obtain a transaction block of ordered transactions; transmitting, from the one or more building computers to a transmitting computer, ordered transaction blocks; selecting, using the transmitting computer, a transaction block of the ordered transaction blocks; transmitting, from the transmitting computer to an executing computer, the selected transaction block; and executing, using the executing computer, the transaction block.

[0031] In some aspects, the techniques described herein relate to a computer-implemented method, wherein the verifiable sequencing rule is arranged to ensure that: (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction, (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain from the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0032] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes, if the set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

[0033] In some aspects, the techniques described herein relate to a computer-implemented method, further including, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block: removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

[0034] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes: if the updated set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0035] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes, if the updated set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

[0036] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes: if the set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0037] In some aspects, the techniques described herein relate to a computer-implemented method, wherein ordering the transactions of the set of transactions further includes executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

[0038] In some aspects, the techniques described herein relate to a plurality of non-transitory computer readable media storing processor-executable instructions that, when executed by at least one processor, cause the at least one processor to perform a method of implementing a verifiable transaction on a decentralized exchange of a liquidity pool of two or more tokens, the method including: transmitting, from trading computers to one or more building computers, transaction information specifying a set of transactions, wherein: transactions of the set of transactions are to be ordered prior to execution, and each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens; ordering, at each of the one or

more building computers and using a verifiable sequencing rule, transactions of the set of transactions to obtain a transaction block of ordered transactions; transmitting, from the one or more building computers to a transmitting computer, ordered transaction blocks; selecting, using the transmitting computer, a transaction block of the ordered transaction blocks; transmitting, from the transmitting computer to an executing computer, the selected transaction block; and executing, using the executing computer, the transaction block.

[0039] In some aspects, the verifiable sequencing rule is arranged to ensure that: (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction, (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain from the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0040] In some aspects, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes, if the set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

[0041] In some aspects, the techniques further include, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block: removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

[0042] In some aspects, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes: if the updated set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0043] In some aspects, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes, if the updated set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the

token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

[0044] In some aspects, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes: if the set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0045] In some aspects, ordering the transactions of the set of transactions further includes executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

[0046] In some aspects, the techniques described herein relate to a distributed system including: a plurality of processors; and a plurality of non-transitory computer readable medium storing processor-executable instructions that, when executed sequentially by each processor of the plurality of processors, cause the plurality of processors to perform a method of implementing a verifiable transaction on a decentralized exchange of a liquidity pool of two or more tokens, the method including: transmitting, from trading computers to one or more building computers, transaction information specifying a set of transactions, wherein: transactions of the set of transactions are to be ordered prior to execution, and each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens; ordering, at each of the one or more building computers and using a verifiable sequencing rule, transactions of the set of transactions to obtain a transaction block of ordered transactions; transmitting, from the one or more building computers to a transmitting computer, ordered transaction blocks; selecting, using the transmitting computer, a transaction block of the ordered transaction blocks; transmitting, from the transmitting computer to an executing computer, the selected transaction block; and executing, using the executing computer, the transaction block.

[0047] In some aspects, the techniques described herein relate to a distributed system, wherein the verifiable sequencing rule is arranged to ensure that: (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the

first transaction, (ii) an execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0048] In some aspects, the techniques described herein relate to a distributed system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes, if the set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

[0049] In some aspects, the techniques described herein relate to a distributed system, wherein ordering the transactions of the set of transactions further includes, after appending the purchase or sale transaction to an end of the ordered transactions in the transaction block: removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

[0050] In some aspects, the techniques described herein relate to a distributed system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes: if the updated set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0051] In some aspects, the techniques described herein relate to a distributed system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further includes, if the updated set of transactions includes both purchases and sales: if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or if a current price of a token is greater than an

initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

[0052] In some aspects, the techniques described herein relate to a distributed system, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule includes: if the set of transactions includes only one of purchases or sales: appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0053] In some aspects, the techniques described herein relate to a distributed system, wherein ordering the transactions of the set of transactions further includes executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

[0054] In some aspects, the techniques described herein relate to a computer-implemented method for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the computer-implemented method including: receiving transaction blocks, each including ordered transactions, wherein: the ordered transactions are to be executed according to an ordering of the transactions, and each of the transactions in the ordered transactions indicates a purchase or a sale of at least one token of the two or more tokens; verifying that the ordered transactions of the received transaction blocks were ordered in compliance with a verifiable sequencing rule; selecting a verified transaction block of the received transaction blocks; and causing the execution of the ordered transactions of the verified transaction block.

[0055] In some aspects, the techniques described herein relate to a computer-implemented method, wherein causing the execution of the ordered transactions includes transmitting, from a first computer to a second computer, the transaction block for execution.

[0056] In some aspects, the techniques described herein relate to a computer-implemented method, wherein the verifiable sequencing rule is arranged to ensure that: (i) an execution price of a first transaction of the ordered transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction, (ii) an execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0057] In some aspects, the techniques described herein relate to a computer-implemented method, wherein verifying that the ordered transactions were ordered in compliance with the verifiable sequencing rule includes executing a smart contract, the smart contract being configured to verify that the ordered transactions were ordered in compliance with the verifiable sequencing rule.

[0058] In some aspects, the techniques described herein relate to at least one non-transitory computer readable medium storing processor-executable instructions that, when executed by at least one processor, cause the at least one processor to perform a method of implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the method including: receiving transaction blocks, each including ordered transactions, wherein: the ordered transactions are to be executed according to an ordering of the transactions, and each of the transactions in the ordered transactions indicates a purchase or a sale of at least one token of the two or more tokens; verifying that the ordered transactions of the received transaction blocks were ordered in compliance with a verifiable sequencing rule; selecting a verified transaction block of the received transaction blocks; and causing the execution of the ordered transactions of the verified transaction block.

[0059] In some aspects, the techniques described herein relate to a system for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the system including: at least one processor; and at least one non-transitory computer readable medium storing processor-executable instructions that, when executed by the at least one processor, cause the at least one processor to perform a method including: receiving transaction blocks, each including ordered transactions, wherein: the ordered transactions are to be executed according to an ordering of the transactions, and each of the transactions in the ordered transactions indicates a purchase or a sale of at least one token of the two or more tokens; verifying that the ordered transactions of the received transaction blocks were ordered in compliance with a verifiable sequencing rule; selecting a verified transaction block of the received transaction blocks; and causing the execution of the ordered transactions of the verified transaction block.

BRIEF DESCRIPTION OF DRAWINGS

[0060] Various aspects and embodiments will be described with reference to the following figures. It should be appreciated that the figures are not drawn to scale. In the drawings, each

identical or nearly identical component that is illustrated in various figures is represented by a like numeral. For purposes of clarity, not every component may be labeled in every drawing.

[0061] FIG. 1A is a schematic diagram illustrating a process for ordering and executing a set of transactions on an exchange, in accordance with some embodiments of the technology described herein.

[0062] FIG. 1B is a schematic diagram illustrating a process for ordering and executing a set of transactions on an exchange, in accordance with some embodiments of the technology described herein.

[0063] FIG. 1C is a schematic diagram illustrating another process for ordering and executing a set of transactions on an exchange, in accordance with some embodiments of the technology described herein.

[0064] FIG. 2A is a schematic diagram of an illustrative distributed system for executing transactions on an exchange using an auction system, in accordance with some embodiments of the technology described herein.

[0065] FIG. 2B is a schematic diagram of an illustrative distributed system for executing transactions on an exchange, in accordance with some embodiments of the technology described herein.

[0066] FIG. 2C is a schematic diagram of another illustrative distributed system for executing transactions on an exchange, in accordance with some embodiments of the technology described herein.

[0067] FIG. 3 is a plot illustrating a liquidity pool of two tokens, in accordance with some embodiments of the technology described herein.

[0068] FIG. 4 is a schematic diagram illustrating how the ordering of transactions can cause a miner to profit at a trader's loss.

[0069] FIG. 5 is a schematic diagram illustrating the same set of transactions of FIG. 4 when ordered using a verifiable sequencing rule, in accordance with some embodiments of the technology described herein.

[0070] FIG. 6 is a schematic diagram illustrating another example of a set of transactions ordered using a verifiable sequencing rule, in accordance with some embodiments of the technology described herein.

[0071] FIG. 7 is a flowchart illustrating a process 700 of ordering a set of transactions using a verifiable sequencing rule, in accordance with some embodiments of the technology described herein.

[0072] FIG. 8 is a flowchart illustrating a process 800 of verifying the ordering of a set of transactions in compliance with a verifiable sequencing rule, in accordance with some embodiments of the technology described herein.

[0073] FIG. 9 is a flowchart illustrating a process 900 of building and executing a transaction block of ordered transactions, in accordance with some embodiments of the technology described herein.

[0074] FIG. 10 is a diagram of an illustrative computer system, in accordance with some embodiments.

[0075] FIG. 11 is a plot illustrating demand and supply curves for tokens available for sale or purchase on an order book, in accordance with some embodiments of the technology described herein.

DETAILED DESCRIPTION

[0076] Provided herein are techniques for executing verifiable transactions on an exchange where builders compose blocks of transactions to be executed on the exchange by selecting the ordering of transactions within the transaction blocks. Conventionally, builders may exploit their position within the exchange to build transaction blocks that include transactions of their own (e.g., by front-running trader transactions), thus gaining risk-free profits at the expense of other traders on the exchange. This problem is known as miner extractable value, or MEV. The techniques described herein implement a verifiable sequencing rule that prevents or mitigates builders from generating transaction blocks that provide MEV at a small cost to execution quality. Additionally, the verifiable sequencing rule enables third party verification of builders' compliance with the verifiable sequencing rule.

[0077] Trading on exchanges (e.g., decentralized exchanges) has been a primary use case for permissionless blockchains, with daily trading volume exceeding billions of U.S. dollars and circulating supplies of tokens on the order of hundreds of millions (e.g., Uniswap has a maximum supply of one billion tokens). Conventionally, traders broadcast transactions (e.g., purchases or sales) they wish to execute on the exchange, and builders compose blocks of transactions and select an ordering of execution of those transactions in the block. Builders may act on behalf of

miners or validators who operate the consensus protocol that determines what is written to the blockchain. Alternatively, miners, validators, or other entities could be responsible for the composition of blocks. Due to the lack of regulatory framework, it is common for builders or similar parties to exploit their position to front-run trader transactions to obtain MEV. Traders may implement countermeasures to mitigate losses caused by MEV, such as setting limit prices or splitting large transactions into multiple smaller transactions.

[0078] The inventors have recognized and appreciated that trading on exchanges may be improved by modifying the interaction between builders and traders. In particular, the inventors have recognized and appreciated that price manipulation by builders may be limited if builders commit to respecting a verifiable sequencing rule which constrains an execution ordering of transactions within a transaction block and if price manipulation can be verified from publicly available information associated with the transactions selected to form a transaction block. Compliance with this rule is verifiable by outside observers such that non-compliant builders may be identified, either manually or automatically (e.g., through the execution of a smart contract).

[0079] Accordingly, the inventors have developed techniques for implementing a verifiable sequencing rule on a financial exchange. The verifiable sequencing rule ensures, for any user transaction *A*, that either (i) the execution price of *A* is at least as good as if *A* was the only transaction in the block (the “standalone” price), (ii) the execution price of *A* is worse than the standalone price but the builder does not profit when including *A* in the transaction block, or (iii) builder misconduct to subvert the verifiable sequencing rule is observable by third parties.

[0080] The verifiable sequencing rule described herein provides execution price guarantees for traders and improves the user experience of trading on exchanges. Unlike conventional technology for mitigating builder misconduct, the techniques described herein improve upon such technology by providing a guarantee that such misconduct is not present and processing multiple transactions in a batch, reducing the overall computational burden of implementing a mitigation strategy. For example, rather than implementing countermeasures to reduce losses, traders can securely request single large transactions instead of splitting a single transaction into many smaller transactions.

[0081] As a result, throughput on the exchange will increase because traders will request fewer transactions overall. Additionally, each computational step to write a transaction block to the exchange will include a larger trading volume. For example, transaction blocks may include at least 10, 100, 1000, 10,000, 100,000, 500,000, 1,000,000, between 10 and 1000, between 100 and

100,000, between 2 and 100,000, between 2 and 500,000, and/or between 2 and 1,000,000 transactions, or any other suitable range of transactions within these ranges. Such an increase in throughput will also reduce an energy cost per transaction, as a greater volume of trades will be executed in a same computational step.

[0082] In some embodiments, a method for ordering transactions to obtain a verifiable transaction block for executing on an exchange is provided. The exchange may be a liquidity pool having two or more tokens, in some embodiments. The method may include receiving transaction information specifying a set of transactions, and the transactions are to be ordered prior to their execution on the exchange. The transactions may also indicate a purchase or a sale on the exchange (e.g., of at least one token, in the example of a liquidity pool). The method may also include ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions and causing the execution of the ordered transactions.

[0083] In some embodiments, causing the execution of the ordered transactions may include transmitting the transaction block for execution from a first computer (e.g., a building computer or a transmitting computer) to a second computer (e.g., an executing computer). In some embodiments, causing the execution of the ordered transactions may include executing the ordered transactions (e.g., at a same building computer).

[0084] In some embodiments, the verifiable sequencing rule may be a rule or set of rules arranged to ensure that builders do not gain profit at the cost of trader execution price from assembling the transaction block if they comply with the verifiable sequencing rule. For example, the verifiable sequencing rule may be arranged to ensure that: (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction, (ii) an execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0085] In some embodiments, ordering the transactions using the verifiable sequencing rule to generate a transaction block includes, if the set of transactions includes both purchases and sales and if a current price of a token is less than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block. In some embodiments, ordering the transactions using the verifiable sequencing rule to generate

a transaction block includes, if the set of transactions includes both purchases and sales and if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block. The appended sale or purchase transaction may then be removed from the set of transactions to obtain an updated set of transactions.

[0086] In some embodiments, ordering the transactions using the verifiable sequencing rule to generate a transaction block includes, if the set of transactions or the updated set of transactions includes only one of purchases or sales, appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

[0087] In some embodiments, ordering the transactions to generate a transaction block may comprise executing a smart contract. The smart contract may be configured to order the transactions according to the verifiable sequencing rule.

[0088] In some embodiments, a method for verifying the ordering of transactions in a verifiable transaction block is provided. The method may include receiving transaction blocks comprising ordered transactions, where the ordered transactions are to be executed according to the ordering of the transactions. Each of the transactions in the ordered transactions may indicate a purchase or a sale on the exchange (e.g., of at least one token, in the example of a liquidity pool). The method may further include verifying, using the verifiable sequencing rule, that the ordered transactions were ordered in compliance with the verifiable sequencing rule. In some embodiments, verifying that the ordered transactions were ordered in compliance with the verifiable sequencing rule comprises executing a smart contract. Thereafter, the method may include selecting a verified transaction block of the received transaction blocks and causing the execution of the ordered transactions of the verified transaction block.

[0089] In some embodiments, a method for implementing a verifiable relay service on an exchange is provided. The method includes transmitting, from a trading computer to one or more building computers, transaction information specifying a set of transactions to be ordered prior to execution. Each of the transactions in the ordered transactions may indicate a purchase or a sale on the exchange (e.g., of at least one token, in the example of a liquidity pool). The method may include ordering, at each of the one or more building computers and using a verifiable sequencing rule, transactions of the set of transactions to obtain a transaction block of ordered transactions. Thereafter, the method may include transmitting, from the one or more building computers to a transmitting computer, the ordered transaction blocks. The method may include selecting, using

the transmitting computer, a transaction block of the ordered transaction blocks and transmitting, from the transmitting computer to an executing computer, the selected transaction block. Finally, the method may include executing, using the executing computer, the transaction block.

[0090] Following below are more detailed descriptions of various concepts related to, and embodiments of, techniques for implementing a credible decentralized exchange using a verifiable sequencing rule. It should be appreciated that various aspects described herein may be implemented in any of numerous ways. Examples of specific implementations are provided herein for illustrative purposes only. In addition, the various aspects described in the embodiments below may be used alone or in any combinations and are not limited to the combinations explicitly described herein.

[0091] FIG. 1A is a schematic diagram illustrating a process 100a for ordering and executing a set of transactions on an exchange (e.g., a decentralized exchange, an exchange implemented using an auction system, etc.), in accordance with some embodiments of the technology described herein. Trading computers 110-1 through 110-N may be configured to generate transactions $T_1, T_2, \dots, T_N$, where N is an integer number. Each of the transactions $T_1, T_2, \dots, T_N$ may include information indicative of a purchase or sale on the exchange. For example, on an exchange of a liquidated pool including two or more tokens, each of the transactions $T_1, T_2, \dots, T_N$ may include information indicative of a purchase or sale of at least one token of the liquidated pool. Alternatively, each of the transactions $T_1, T_2, \dots, T_N$ may indicate the purchase or sale of a stock, security, and/or currency, and the exchange may be a centralized exchange.

[0092] In some embodiments, the trading computers 110-1 through 110-N may be computers, computing devices, or distributed computing systems associated with traders acting on the exchange. The trading computers may be configured to transmit transactions $T_1, T_2, \dots, T_N$ from the trading computers 110-1 through 110-N to building computers 120-1 through 120-M, where M is an integer that may be equal to or different than the integer N .

[0093] In some embodiments, the building computers 120-1 through 120-M may assemble transaction blocks $TB_1, TB_2, \dots, TB_M$ using some or all of the transactions $T_1, T_2, \dots, T_N$. The building computers 120-1 through 120-M may also include additional transactions (e.g., transactions that the builders want to execute on the exchange) in the transaction blocks $TB_1, TB_2, \dots, TB_M$.

[0094] When assembling the transaction blocks $TB_1, TB_2, \dots, TB_M$, the building computers 120-1 through 120-M may arrange the transactions in any order consistent with a verifiable

sequencing rule implemented on the exchange. The transactions may be ordered to indicate an order of execution on the exchange (e.g., a first transaction in the transaction block may be executed first, a second transaction in the transaction block may be executed second after the first transaction is executed, etc.).

[0095] In some embodiments, the building computers 120-1 through 120-M may be single computers and/or computing devices or may be distributed computing systems performing multiparty computation. For example, in some embodiments, a building computer of the building computers 120-1 through 120-M may build a transaction block using multiple, communicatively coupled processors housed separately from one another.

[0096] In some embodiments, the building computers 120-1 through 120-M may transmit the transaction blocks $TB_1, TB_2, \dots, TB_M$ to a transmitting computer 130. The transmitting computer 130 may be configured to verify that the ordering of transactions in each of the transaction blocks $TB_1, TB_2, \dots, TB_M$ complies with the verifiable sequencing rule. In some embodiments, the transmitting computer 130 may be a computer, a computing device, or a distributed computing systems associated with transmitters (e.g., relays, auctions, or other services) acting on the exchange.

[0097] In some embodiments, the transmitting computer 130 may be additionally configured to select a transaction block for execution (e.g., acting as a relay or an auctioneer in an auction service). In some embodiments, the transmitting computer 130 could combine more than one transaction block into a single transaction block. The transmitting computer 130 may then transmit the selected or combined transaction block to executing computer 140 for execution. The executing computer 140 may be a computer, a computing device, or a distributed computing systems associated with executors (e.g., validators, miners, or others executing transactions) acting on the exchange.

[0098] In some embodiments, the executing computer 140 may be configured to execute the selected transaction block. For example, the executing computer 140 may be a proof of work miner, a proof of stake miner, or a validator. The executing computer 140 may be configured to execute the selected transaction block by completing a computational step configured to add a new block or blocks to the blockchain representing the exchange. In some embodiments, the executing computer 140 may be configured to execute the selected transaction block by executing a smart contract on the blockchain representing the exchange. Alternatively, in some embodiments

the executing computer 140 may execute the transactions of the transaction block on a centralized exchange.

[0099] In some embodiments, the process 100a of FIG. 1A may be performed using a system 200a as illustrated by FIG. 2A. The system 200a includes trading computers 110-1 through 110-N, building computers 120-1 through 120-M, transmitting computer 130, and executing computer 140. Each of these computers is communicatively coupled by communications network 250. For example, the communications network 250 may be a local network (e.g., a local area network (LAN), wide area network (WAN), or intranet) or an open network (e.g., the internet). Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0100] FIG. 1B is a schematic diagram illustrating an alternative process 100b for ordering and executing a set of transactions on an exchange, in accordance with some embodiments of the technology described herein. As in the example of FIG. 1A, trading computers 110-1 through 110-N may be configured to generate transactions $T_1, T_2, \dots, T_N$. In process 100b, however, trading computers 110-1 through 110-N may be configured to transmit transactions $T_1, T_2, \dots, T_N$, to a single building computer 120-1.

[0101] In some embodiments, the single building computer 120-1 may be configured to generate the transaction block TB by arranging the transactions in any order consistent with a verifiable sequencing rule implemented on the exchange. The transactions may be ordered to indicate an order of execution on the exchange (e.g., a first transaction in the transaction block may be executed first, a second transaction in the transaction block may be executed second after the first transaction is executed, etc.).

[0102] In some embodiments, the single building computer 120-1 may be a single computer and/or computing device or may be a distributed computing system performing multiparty computation. For example, in some embodiments, the building computer 120-1 may build the transaction block TB using multiple, communicatively coupled processors housed separately from one another.

[0103] In some embodiments, the single building computer 120-1 may be configured to transmit the transaction block TB to the executing computer 140 for execution of the ordered transactions in the transaction block TB . The executing computer 140 may be configured to verify that the ordering of the transactions in the transaction block TB complies with the verifiable sequencing rule and/or to execute the transaction block TB on the exchange.

[0104] For example, in some embodiments, the executing computer 140 may be a proof of work miner, a proof of stake miner, or a validator. The executing computer 140 may be configured to execute the selected transaction block by completing a computational step configured to add a new block or blocks to the blockchain representing the exchange. In some embodiments, the executing computer 140 may be configured to execute the selected transaction block by executing a smart contract on the blockchain representing the exchange. Alternatively, in some embodiments the executing computer 140 may execute the transactions of the transaction block on a centralized exchange. In some embodiments, the executing computer 140, prior to executing the selected transaction block, may verify that the ordering of the transactions in the selected transaction block complies with the verifiable sequencing rule. For example, the executing computer 140 may use a smart contract to verify that the ordering of the transactions complies with the verifiable sequencing rule.

[0105] In some embodiments, the process 100b of FIG. 1B may be performed using a system 200b as illustrated by FIG. 2B. The system 200b includes trading computers 110-1 through 110-N, a building computer 120-1, and executing computer 140. Each of these computers is communicatively coupled by communications network 250. For example, the communications network 250 may be a local network (e.g., a local area network (LAN), wide area network (WAN), or intranet) or an open network (e.g., the internet). Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0106] FIG. 1C is a schematic diagram illustrating an alternative process 100c for ordering and executing a set of transactions on an exchange, in accordance with some embodiments of the technology described herein. As in the example of FIG. 1A, trading computers 110-1 through 110-N may be configured to generate transactions $T_1, T_2, \dots, T_N$. In process 100c, however, trading computers 110-1 through 110-N may be configured to transmit transactions $T_1, T_2, \dots, T_N$, to a single building and executing computer 150.

[0107] The building and executing computer 150 may be configured to generate the transaction block TB by ordering the transactions $T_1, T_2, \dots, T_N$ in any order consistent with a verifiable sequencing rule implemented on the exchange. The building and executing computer 150 may also be configured to, after generating the transaction block TB , execute the transaction block TB on the exchange.

[0108] In some embodiments, the building and executing computer 150 may be a computer, a computing device, or a distributed computing system performing multiparty computation. For example, in some embodiments, the building and executing computer 150 may build and/or execute the transaction block TB using multiple, communicatively coupled processors housed separately from one another.

[0109] In some embodiments, the building and executing computer 150 may be a proof of work miner, a proof of stake miner, or a validator. The building and executing computer 150 may be configured to execute the selected transaction block by completing a computational step configured to add a new block or blocks to the blockchain representing the exchange. In some embodiments, the building and executing computer 150 may be configured to execute the transaction block TB by executing a smart contract on the blockchain representing the exchange. Alternatively, in some embodiments the building and executing computer 150 may execute the transactions of the transaction block on a centralized exchange.

[0110] In some embodiments, the process 100c of FIG. 1C may be performed using a system 200c as illustrated by FIG. 2C. The system 200c includes trading computers 110-1 through 110-N and a building and executing computer 150. Each of these computers is communicatively coupled by communications network 250. For example, the communications network 250 may be a local network (e.g., a local area network (LAN), wide area network (WAN), or intranet) or an open network (e.g., the internet). Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0111] In some embodiments, the process 100c of FIG. 1C may be implemented on a centralized exchange. The centralized exchange may be, for example, running an order book including bids and asks. The process 100c may be implemented on such a centralized exchange by matching bids and asks on the order book according to the verifiable sequencing rule described herein.

[0112] In some embodiments, on an order book, a buy order $Buy(q, p)$ signals the intent to purchase up to q units of token one at a maximum price of p per unit, where the price is denominated in units of token two. A sell order $Sell(q, p)$ signals the intent to sell up to q units of token one at a minimum price of p per unit, where the price is denominated in units of token two. An order book exchange has a state, $\mathcal{D}$, representing the set of buys and sells in the order book. A match in $\mathcal{D}$ is a tuple, $M = (Buy(q, p), Sell(q', p'), p^*) \in \mathcal{D} \times \mathcal{D} \times \mathbb{R}_{\geq 0}$, where

$Buy(q, p)$ is a buy order in the state, $Sell(q', p')$ is a sell order in the state, and p^* is the execution price. A match is valid if $p' \leq p^* \leq p$ (e.g., the price is at least the limit price for the seller and at most the limit price of the buyer).

[0113] A valid match results in the following transfer of assets, as well as updates to the state $\mathcal{D}$:

1. Transfer $\min\{q, q'\}$ token one units from the owner of $Sell(q', p')$ to the owner of $Buy(q, p)$.
2. Transfer $\min\{q, q'\} \cdot p^*$ token two units from the owner of $Buy(q, p)$ to the owner of $Sell(q', p')$.
3. Replace $Buy(q, p)$ with $Buy(q - \min\{q, q'\}, p)$ in $\mathcal{D}$. The new buy order has the same owner as the old buy order.
4. Replace $Sell(q', p')$ with $Sell(q' - \min\{q, q'\}, p')$ in $\mathcal{D}$. The new sell order has the same owner as the old sell order.

[0114] A matching sequence $(M_1, M_2, \dots, M_m)$ for state $\mathcal{D}$ is a list of matches. Let $\mathcal{D}_0 = \mathcal{D}$ and for all $i \in [m]$, let $\mathcal{D}_i$ be the state after M_i executes on state $\mathcal{D}_{i-1}$. A matching sequence is valid if, for each $i \in [m]$, M_i is a valid match in $\mathcal{D}_i$. A sequencing rule is a function that takes any state $\mathcal{D}$ and outputs a valid matching sequence.

[0115] In some embodiments, a sequencing rule, using a batch auction and censorship-resistant order book, may be implemented to generate a verifiable order book. An order book is “censorship-resistant” if whenever $\mathcal{D}$ is the result of tampering (e.g., the adding or removing of transactions to a sequence $\mathcal{D}^*$ consisting of only transactions owned by traders), then $\mathcal{D} \supseteq \mathcal{D}^*$. That is, a sequencer can add but not remove transactions from a censorship-resistant order book.

[0116] FIG. 11 illustrates an example of a demand curve, $D(p)$, and supply curve, $S(p)$, where the demand curve illustrates the amount of token one that buyers in $\mathcal{D}$ would purchase as a function of price, p , and the supply curve illustrates the amount of token one that sellers in $\mathcal{D}$ would sell as a function of price, p . The sequencing rule is then:

1. Initialize M as empty.
2. Determine the curve C , where $D(p)$ intersects with $S(p)$. Let the line segment $L = [x, y]$ be the projection of C to the price axis. Then determine the midpoint, p^* , of the line segment, L , as $p^* = \frac{x+y}{2}$.
3. While there is a buy $Buy(q, p) \in \mathcal{D}$ and a sell $Sell(q', p') \in \mathcal{D}$, where $p' \leq p^* \leq p$,

append $(Buy(q, p), Sell(q', p'), p^*)$ to M and update $\mathcal{D}$ as the state after the match executes.

This sequencing rule is publicly verifiable because any party can check if all transactions in a block pay the same price.

[0117] Assuming $\mathcal{D}^*$ is the untampered order book state, and M is a valid matching sequence obtaining the above sequencing rule for a potentially tampered order book state $\mathcal{D} \supseteq \mathcal{D}^*$, then for any trader transaction $T \in \mathcal{D}^*$, the execution price of T will be:

- at least as good as the execution price the trader could obtain from any valid matching sequence for $\mathcal{D}^*$, the untampered price; or
- worse than the untampered price.

The converse of the above is that if the sequencer derives positive utility from including a trader transaction, then the trader must receive an execution price at least as good as if the sequencer did not tamper with the order book.

[0118] In some embodiments, the sequencing protocol may be arranged so that a network of agents agree about which transactions enter the order book state rather than being arranged to determine the global clock time for when a transaction enters the order book state. Such a protocol is preferable due to inherent latency in any communication network (e.g., different agents will receive transactions at different times and orders). Assumptions include that traders communicate with the exchange via a communication network where any message from a sender eventually reaches the designated receiver and that for each transaction T a trader wishes to include into the order book $\mathcal{D}$, T will eventually enter the order book state $\mathcal{D}$.

[0119] In some embodiments, to make the exchange censorship-resistant, there may be multiple verifiers $(V_1, V_2, \dots, V_n)$ in this network who vote on the state of the order book. Alternatively or additionally, there may be external observers who do not actively participate in the protocol but only contribute by replaying messages in the network (e.g., a regulatory agency). Each agent (e.g., a verifier, sequencer, or external observer) has a local view of the order book. Initially, each local view of the order book is empty.

[0120] In some embodiments, if a trader wants to include a transaction T in the local view of the order book of every agent, the trader may broadcast T to every agent by some known deadline. Once a verifier V_i receives T , then V_i broadcasts an acknowledgement of T to every agent. Once an agent receives an acknowledgment of T from the strict majority of verifiers, the agent then adds

T to their local view of the order book. The sequencer may then pick a matching sequence M for their local view, $\mathcal{D}$, of the order book. M may be verified as valid by:

1. V_i votes that M is valid if and only if $\mathcal{D}$ is equal to V_i 's local view of the order book.
2. M is considered valid if, and only if, the strict majority of the verifiers vote that M is valid.

The procedure to check that M is valid requires the strict majority of participants to be honest by following the above procedure. Further, if the strict majority of verifiers are honest, then the sequencer can only generate valid matching sequences for order books states that were not censored.

[0121] In some embodiments, a mechanism that allows any external observer to check whether the strict majority are honest may be provided. Whenever the external observer receives a transaction T , the observer broadcasts the transaction T to all verifiers. If T does not enter the order book in the observer's view, then most verifiers are dishonest. Therefore, if the strict majority of nodes are not honest, any external observer can detect that most verifiers are dishonest.

[0122] Returning to the processes of FIGs. 1A, 1B, or 1C, these processes may be implemented on a decentralized exchange that is a liquidity pool of two or more tokens. In a liquidity pool, liquidity providers lock capital in reserves of tokens to be traded. An example of a decentralized exchange using a liquidity pool is Uniswap.

[0123] Let X_1 be the number of tokens of a first kind and X_2 be the number of tokens of a second kind in a liquidity pool of two tokens. In a constant function market maker (e.g., Uniswap or other suitable decentralized exchanges), the product $X_1 \cdot X_2$ defines the potential corresponding to a current state of the exchange. As illustrated in FIG. 3, a trader can submit a transaction on the exchange against the liquidity pool and, for example, withdraw $q > 0$ units of token 1 as long as they deposit a quantity $p > 0$ of token two that preserves the potential:

$$(X_1 - q)(X_2 + p) = X_1 \cdot X_2.$$

Alternatively, some decentralized exchanges using a liquidity pool charge a fee to implement transactions. On such exchanges, a transaction must preserve the potential according to:

$$(X_1 - q)(X_2 + p\partial) = X_1 \cdot X_2.$$

where ∂ is a constant and $\partial < 1$.

[0124] On an exchange based on a liquidity pool, those assembling transaction blocks can easily manipulate the transaction ordering to gain profit, or miner-extractable value (MEV), as

illustrated in FIG. 4. In the example of FIG. 4, the horizontal axis represents the amount of token 1 in reserve, with the value increasing from left to right. In this example, the builder has added two transactions, T_1 and T_3 , to “sandwich” the trader’s transaction, T_2 . In a first transaction, T_1 , the builder purchases a number q of token 1, raising its price relative to its initial price. In a second transaction, T_2 , the builder implements the trader’s transaction to also purchase a number q of token 1, further raising the price of token 1. In a third transaction, the builder can implement their own transaction T_3 to sell a number q of token 1 at a higher price than the builder purchased token 1 in transaction T_1 . Thus, the builder makes a profit in what is known as a “sandwich attack.”

[0125] The inventors have recognized and appreciated that such sandwich attacks, along with other types of MEV attacks, may be mitigated if builders are required to comply with the following verifiable sequencing rule, in accordance with some embodiments described herein. Given an initial set of transactions, $Q_1, \dots, Q_N$, that can be described by the sets Q^{buy} and Q^{sell} , where Q^{buy} is a set of the purchasing transactions in $Q_1, \dots, Q_N$ and Q^{sell} is a set of the sale transactions in $Q_1, \dots, Q_N$, and an initial state of the liquidity pool $X_0 = (X_{0,1}, X_{0,2})$, where $X_{0,1}$ and $X_{0,2}$ are the initial reserves of the first and second tokens, respectfully, an execution ordering T , where T is initially empty and $t = |T|$, can be determined by:

1. While Q^{buy} and Q^{sell} are both non-empty sets:
 - (a) If $X_{t,1} = X_{0,1}$, append any Q in Q^{buy} or any Q in Q^{sell} to T and remove the selected Q from Q^{buy} or Q^{sell} ;
 - (b) If $X_{t,1} > X_{0,1}$, append any Q in Q^{buy} to T and remove the selected Q from Q^{buy} ;
 - (c) Else, append any Q in Q^{sell} to T and remove the selected Q from Q^{sell} ,
 where appending a transaction to T defines the state $X_{t+1} = (X_{t+1,1}, X_{t+1,2})$ as the state after the appended transaction executes on state X_t .
2. If Q^{buy} or Q^{sell} are empty sets: add the remaining transactions of the non-empty set of Q^{buy} or Q^{sell} to T in any order.

[0126] In some embodiments, in the case that $X_{t,1} = X_{0,1}$ and the current price is the same as the initial price of the token, the verifiable sequencing rule may be arranged to require appending only any Q in Q^{buy} or only any Q in Q^{sell} to T .

[0127] In some embodiments, part (1) of the above verifiable sequencing rule states that, if there are both purchase and sale transactions in the set of available transactions and a current price of a token is equal to an initial price of the token (or, alternatively, if the current reserves of a

token are equal to the initial reserves of the token), the next transaction block may be any of the purchase or sale transactions in the set of available transactions. The selected purchase or sale transaction that is appended to the transaction block may then be removed from the set of available transactions. Alternatively, if a current price of a token is less than an initial price of the token (or, alternatively, if the current reserves of a token are greater than the initial reserves of the token), the next transaction added to the transaction block should be any of the purchase transactions in the set of available transactions. The purchase transaction that is appended to the transaction block may then be removed from the set of available transactions. However, if a current price of a token is greater than an initial price of the token (or, alternatively, if the current reserves of a token are less than the initial reserves of the token), then the next transaction to be added to the transaction block should be any of the sale transactions in the set of available transactions. The sale transaction that is appended to the transaction block may then be removed from the set of available transactions. Part (2) of the above verifiable sequencing rule states that, if the set of available transactions includes only purchases or only sales, then the remaining transactions of the available transactions may be appended to the transaction block in any order.

[0128] In some embodiments, the verifiable sequencing rule may be arranged to ensure the following possibility theorem:

- (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,
- (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or
- (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

[0129] As an example of the implementation of the verifiable sequencing rule described above, FIG. 5 is an illustration of the same set of transactions of FIG. 4 when executed in accordance with a transaction order determined using a verifiable sequencing rule, in accordance with some embodiments of the technology described herein. In the example of FIG. 5, initially the set of purchase transactions, Q^{buy} , includes transactions T_1 and T_3 , each of size one, and the set of sale transactions, Q^{sell} , includes only transaction T_2 , also of size one. To select a first transaction to be executed, one looks to whether $X_{t,1} \geq X_{0,1}$; in this case, $X_{t,1} = X_{0,1}$ initially, and

any transaction in Q^{buy} or Q^{sell} may be initially appended to the transaction block T . Therefore, any of transactions T_1 , T_2 , or T_3 may be ordered as the first transaction in the transaction block T .

[0130] Assume that the builder's transaction T_1 is selected as the first transaction in the block T , after appending T_1 to the transaction block T , there are still transactions in both Q^{buy} and Q^{sell} , but $X_{t,1} < X_{0,1}$. Thus, the sale T_2 must be appended to the transaction block T as the second transaction.

[0131] After appending transaction T_2 to the transaction block T and removing it from Q^{sell} , Q^{sell} is now empty, and the rules require that the transactions remaining in Q^{buy} must be appended to the transaction block T . Thus, the transaction T_3 is appended as the third transaction in the transaction block T . The full set of transaction orderings that are allowable under the verifiable sequencing rule for the example of FIG. 5 is:

$$T_1, T_2, T_3$$

$$T_3, T_2, T_1$$

$$T_2, T_1, T_3$$

$$T_2, T_3, T_1$$

As may be appreciated from the example of FIG. 5 and the above allowable transaction orderings, using the verifiable sequencing rule results in the builder being unable to implement the sandwich attack of FIG. 4.

[0132] FIG. 6 is a schematic diagram illustrating another example of a set of transactions ordered using a verifiable sequencing rule, in accordance with some embodiments of the technology described herein. B_1 (a purchase of two units of a first token), S_2 (a sale of one unit of the first token), and S_5 (a sale of one unit of the first token) are transactions added to the transaction block by a builder while S_3 (a sale of one unit of the first token), B_4 (a purchase of two units of the first token), and B_6 (a purchase of two units of the first token) are transactions requested by traders. In this example, the builder is able to extract some MEV from the transaction block because the transactions B_1 , S_2 , and S_5 are inserted into the transaction block by the builder, but the transaction block is not in violation of the possibility theorem described above.

[0133] In the example of FIG. 6, Q^{buy} and Q^{sell} both initially include three transactions such that the verifiable sequencing rule allow for a selection of any Q in Q^{buy} or Q^{sell} to be appended to the transaction block T and to thereafter remove the selected Q from Q^{buy} or Q^{sell} . The builder may therefore include their purchase B_1 as the first transaction in the block. After this first

transaction, the token 1 reserves will decrease to $X_1 - 2 < X_1$, where X_1 denotes the initial reserves of the first token, and the verifiable sequencing rule requires that the second transaction in the transaction block T be selected from Q^{sell} . Thus, the builder may place any of S_2 , S_3 , or S_5 as the second transaction in the transaction block T . In the example of FIG. 6, the builder selected S_2 to be the second transaction.

[0134] After executing transaction S_2 , the token 1 reserves will be $X_1 - 1 < X_1$, such that the verifiable sequencing rule requires that the third transaction in the transaction block T be selected from Q^{sell} . The builder may therefore place either of S_3 or S_5 as the third transaction in the transaction block T . In the example of FIG. 6, the builder selected a trader's transaction S_3 to be the third transaction in the transaction block T .

[0135] After executing transaction S_3 , the token 1 reserves will be back at their initial value of X_1 . The verifiable sequencing rule therefore allows the fourth transaction of the transaction block T be selected from the remaining transactions in Q^{buy} or Q^{sell} . Thus, B_4 , B_6 , or S_5 could be placed as the fourth transaction in the transaction block T . Here, the builder selected transaction B_4 .

[0136] After executing transaction B_4 , the token 1 reserves are returned to $X_1 - 2 < X_1$, and the verifiable sequencing rule requires that the fifth transaction in the transaction block T be selected from Q^{sell} . The only remaining transaction in Q^{sell} is S_5 , which is added as the fifth transaction in the transaction block T .

[0137] At this point, Q^{sell} is now empty, and the verifiable sequencing rule requires that any remaining transactions of the non-empty set of Q^{buy} be appended to the transaction block T . In this example, only transaction B_6 remains in Q^{buy} , and therefore B_6 is appended as the sixth and last transaction in the transaction block T .

[0138] As should be appreciated from the example of FIG. 6, the builder is still able to extract a small amount of MEV from the transaction block T , as the builder can split the sale of the two purchased tokens in transaction B_1 into two smaller sale transactions S_2 and S_5 . However, the verifiable sequencing rule does prevent the builder from extracting a large amount of MEV, as the builder cannot create a transaction ordering that is a pure sandwich attack (e.g., $B_1, B_4, B_6, S_2, S_5, S_3$) which would cause the traders additional loss in comparison to the ordering shown in the example of FIG. 6. Additionally, the ordering of the transaction block T of FIG. 6 adheres to the possibility theorem described herein for each of the trader's transactions S_3 , B_4 , and

B_6 : (i) an execution price of B_4 is as good as a standalone price for the first transaction, and (ii) the execution price of S_3 and B_6 is worse than the standalone price, but the builder does not gain profit from including S_3 and B_6 in the transaction block T .

[0139] FIG. 7 is a flowchart illustrating a process 700 of ordering a set of transactions using a verifiable sequencing rule, in accordance with some embodiments of the technology described herein. The process 700 may be executed using any suitable computing device. For example, in some embodiments, the process 700 may be performed by a single computing device. As another example, in some embodiments, the process 700 may be performed by a distributed computing system comprising one or more processors co-located with each other (e.g., in a same room, a same facility) or remote from one another and communicatively coupled via the internet. In some embodiments, the process 700 may be executed using any one of the building computers 110-1 through 110-N or building and executing computer 150, described in connection with FIGs. 1A-2C herein.

[0140] In some embodiments, the process 700 may begin with an act 702 of receiving transaction information specifying a set of transactions. The transactions may be generated, for example, by one or more remote computers and transmitted by a communications network (e.g., communications network 250) to the computer implementing the process 700. Each of the transactions may indicate a purchase or a sale on the exchange. For example, each of the transactions in the set of received transactions may indicate a purchase or a sale of at least one token on an exchange having a liquidity pool of two or more tokens.

[0141] In some embodiments, after act 702, the process 700 may proceed to an act 704 of ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions. The verifiable sequencing rule may be arranged to ensure adherence to the possibility theorem discussed herein.

[0142] In some embodiments, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule may include appending any purchase or sale transaction in the set of transactions to an end of the ordered transactions in the transaction block if the available set of transactions includes both purchases and sales and if a current price of a token is equal to an initial price of the token (or alternatively, if the current reserves of the token are equal to the initial reserves of the token).

[0143] Alternatively or additionally, in some embodiments, ordering the transactions in the transaction block using the verifiable sequencing rule may include appending any purchase

transaction in the set of transactions to an end of the ordered transactions in the transaction block if the set of transactions includes both purchases and sales and if a current price of a token is less than an initial price of the token (or alternatively, if the current reserves of the token are greater than the initial reserves of the token). Alternatively or additionally, in some embodiments, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule may include appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block if the set of transactions includes both purchases and sales and if a current price of a token is greater than an initial price of the token (or alternatively, if the current reserves of the token are less than the initial reserves of the token).

[0144] In some embodiments, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule may include appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block if the initial or updated set of transactions includes only one of purchases or sales.

[0145] In some embodiments, the appended purchase or sale transaction may be removed from the set of transactions to obtain an updated set of transactions after the purchase or sale is appended to the transaction block.

[0146] In some embodiments, ordering the transactions of the set of transactions may include executing a smart contract. The smart contract may be arranged to order the transactions according to the verifiable sequencing rule.

[0147] In some embodiments, after act 704, the process 700 may proceed to an act 706 of causing the execution of the ordered transactions. Causing the execution of the ordered transactions may include transmitting the transaction block from the computer implementing the process 700 to another computer (e.g., to a transmitting computer 130 or an executing computer 140 as described in connection with FIGs. 1A-1B and 2A-2B herein) configured to further transmit or execute the transaction block. In some embodiments, causing the execution of the ordered transactions may alternatively include executing the transactions of the transaction block at the computer implementing the process 700.

[0148] FIG. 8 is a flowchart illustrating a process 800 of verifying the ordering of a set of transactions using a verifiable sequencing rule, in accordance with some embodiments of the technology described herein. The process 800 may be executed using any suitable computing device. For example, in some embodiments, the process 800 may be performed by a single

computing device. As another example, in some embodiments, the process 800 may be performed by a distributed computing system comprising one or more processors co-located with each other (e.g., in a same room, a same facility) or remote from one another and communicatively coupled via the internet. In some embodiments, the process 800 may be executed using transmitting computer 130 described in connection with FIGs. 1A-1B and 2A-2B herein.

[0149] In some embodiments, process 800 may begin with an act 802 of receiving transaction blocks. The transaction blocks may include ordered transactions, where the ordered transactions are to be executed according to the ordering of the transactions in the transaction blocks. The transaction blocks may be generated, for example, by one or more remote computers (e.g., building computers 110-1 through 110-N) and transmitted by a communications network (e.g., communications network 250) to the computer implementing the process 700. Each of the transactions of the ordered transactions may indicate a purchase or a sale on the exchange. For example, each of the transactions in the ordered transactions may indicate a purchase or a sale of at least one token on an exchange having a liquidity pool of two or more tokens.

[0150] In some embodiments, after act 802, process 800 may proceed to an act 804 of verifying that the ordered transactions of the received transaction blocks were ordered in compliance with the verifiable sequencing rule described herein. The received transaction blocks may be verified according to the following test, where $t = 1, 2, \dots, |T|$ is an index of transactions in the transaction block:

- (1) If the transactions of the transaction block are all of the same type (e.g., all purchases or all sales), then the transaction block obeys the verifiable sequencing rule with the transactions being ordered in any order.
- (2) If $X_{t-1,1} \leq X_{0,1}$ and T_t is a purchase transaction, then the transaction block obeys the verifiable sequencing rule.
- (3) If $X_{t-1,1} \geq X_{0,1}$ and T_t is a sale transaction, then the transaction block obeys the verifiable sequencing rule.

[0151] In some embodiments, after act 804, process 800 may proceed to an act 806 of selecting a verified transaction block of the received transaction blocks. For example, the transmitting computer (e.g., transmitting computer 130) may select a verified transaction block using an auction system, although it should be appreciated that the verified transaction block may be selected in any suitable way.

[0152] In some embodiments, after act 806, process 800 may proceed to an act 808 of causing the execution of the ordered transactions of the verified transaction block. Causing the execution of the ordered transactions may include transmitting the transaction block from the computer implementing the process 800 to another computer (e.g., to an executing computer 140 as described in connection with FIGs. 1A-1B and 2A-2B herein) configured to execute the transaction block. In some embodiments, causing the execution of the ordered transactions may alternatively include executing the transactions of the transaction block at the computer implementing the process 800.

[0153] FIG. 9 is a flowchart illustrating a process 900 of building and executing a transaction block of ordered transactions, in accordance with some embodiments of the technology described herein. The process 900 may be executed using any suitable computing devices. For example, in some embodiments, the process 900 may be performed by a distributed computing system comprising one or more processors co-located with each other (e.g., in a same room, a same facility) or remote from one another and communicatively coupled via the internet. In some embodiments, the process 900 may be executed using the systems described in connection with FIGs. 1A and 2A herein.

[0154] In some embodiments, process 900 may begin with an act 902 of transmitting, from a trading computer to one or more building computers, transaction information specifying a set of transactions. The transactions may be generated, for example, by one or more remote trading computers (e.g., trading computers 110-1 through 110-N) and transmitted by a communications network (e.g., communications network 250) to building computers (e.g., building computers 120-1 through 120-M or building and executing computer 150). Each of the transactions may indicate a purchase or a sale on the exchange. For example, each of the transactions in the set of received transactions may indicate a purchase or a sale of at least one token on an exchange having a liquidity pool of two or more tokens.

[0155] In some embodiments, after act 902, process 900 may proceed to an act 904 of ordering the received transactions at each of the one or more building computers and using a verifiable sequencing rule. Ordering the transactions may generate transaction blocks of ordered transactions. The verifiable sequencing rule may be arranged to ensure adherence of the ordered transactions to the possibility theorem discussed herein.

[0156] In some embodiments, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule may include appending any purchase or sale

transaction in the set of transactions to an end of the ordered transactions in the transaction block if the available set of transactions includes both purchases and sales and if a current price of a token is equal to an initial price of the token (or alternatively, if the current reserves of the token are equal to the initial reserves of the token).

[0157] Alternatively or additionally, in some embodiments, ordering the transactions in the transaction block using the verifiable sequencing rule may include appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block if the set of transactions includes both purchases and sales and if a current price of a token is less than an initial price of the token (or alternatively, if the current reserves of the token are greater than the initial reserves of the token). Alternatively or additionally, in some embodiments, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule may include appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block if the set of transactions includes both purchases and sales and if a current price of a token is greater than an initial price of the token (or alternatively, if the current reserves of the token are less than the initial reserves of the token).

[0158] In some embodiments, ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule may include appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block if the initial or updated set of transactions includes only one of purchases or sales.

[0159] In some embodiments, the appended purchase or sale transaction may be removed from the set of transactions to obtain an updated set of transactions after the purchase or sale is appended to the transaction block.

[0160] In some embodiments, ordering the transactions of the set of transactions may include executing a smart contract. The smart contract may be arranged to order the transactions according to the verifiable sequencing rule.

[0161] In some embodiments, after act 904, process 900 may proceed to an act 906 of transmitting, from the one or more building computers to a transmitting computer, ordered transaction blocks. For example, the one or more building computers (e.g., building computers 120-1 through 120-M) may transmit the transaction blocks to the transmitting computer (e.g., transmitting computer 130) using a communications network (e.g., communications network 250).

[0162] In some embodiments, after act 906, process 900 may proceed to an act 908 of selecting, using the transmitting computer, a transaction block of the ordered transaction blocks. In some embodiments, selecting the transaction block may include verifying that the received transaction blocks comply with the verifiable sequencing rule. The received transaction blocks may be verified according to the following test, where $t = 1, 2, \dots, |T|$ is an index of transactions in the transaction block:

- (1) If the transactions of the transaction block are all of the same type (e.g., all purchases or all sales), then the transaction block obeys the verifiable sequencing rule with the transactions being ordered in any order.
- (2) If $X_{t-1,1} \leq X_{0,1}$ and T_t is a purchase transaction, then the transaction block obeys the verifiable sequencing rule.
- (3) If $X_{t-1,1} \geq X_{0,1}$ and T_t is a sale transaction, then the transaction block obeys the verifiable sequencing rule.

Where a received transaction block does not comply with the verifiable sequencing rule, the transmitting computer may be configured to discard, or not select, the noncompliant transaction block.

[0163] In some embodiments, after act 908, process 900 may proceed to an act 910 of transmitting, from the transmitting computer to an executing computer, the selected transaction block. For example, the transmitting computer (e.g., transmitting computer 130) may transmit the selected transaction block to an executing computer (e.g., executing computer 140) using a communications network (e.g., communications network 250). In some embodiments, act 910 may be optional and act 912 may be performed by a same computer as the computer performing act 908 (e.g., building and executing computer 150).

[0164] In some embodiments, after act 910, process 900 may proceed to an act 912 of executing the transaction block using the executing computer. For example, the executing computer may be a proof of work miner, a proof of stake miner, or a validator. The executing computer may be configured to execute the selected transaction block by completing a computational step configured to add a new block or blocks to the blockchain representing the exchange. In some embodiments, the executing computer may be configured to execute the selected transaction block by executing a smart contract on the blockchain representing the exchange. Alternatively, in some embodiments the executing computer may execute the transactions of the transaction block on a centralized exchange.

[0165] FIG. 10 is a diagram of an illustrative computer system on which embodiments described herein may be implemented. An illustrative implementation of a computer system 1000 that may be used in connection with any of the embodiments of the disclosure provided herein is shown in FIG. 10. In some embodiments, any one of the processes described herein may be implemented on and/or using the computer system 1000. The computer system 1000 may include one or more processors 1010 and one or more articles of manufacture that comprise tangible (e.g., non-transitory) computer-readable storage media (e.g., memory 1020 and one or more non-volatile storage media 1030). The processor 1010 may control writing data to and reading data from the memory 1020 and the non-volatile storage device 1030 in any suitable manner. To perform any of the functionality described herein, the processor 1010 may execute one or more processor-executable instructions stored in one or more non-transitory computer-readable storage media (e.g., the memory 1020), which may serve as non-transitory computer-readable storage media storing processor-executable instructions for execution by the processor 1010.

A. Mathematical Background

[0166] **Lemma A.1** (AM-GM Inequality). Let $x_1, x_2, \dots, x_n \geq 0$. Then:

$$\frac{1}{n} \sum_{i=1}^n x_i \geq \sqrt[n]{\prod_{i=1}^n x_i}$$

[0167] **Theorem A.1** (Intermediate Value Theorem). Let f be a real-valued continuous function with domain, $\text{dom}(f)$, equal to the interval $[a, b]$. If $\min\{f(a), f(b)\} \leq u \leq \max\{f(a), f(b)\}$, then there is a $c \in [a, b]$ such that $f(c) = u$.

[0168] **Lemma A.2.** A real-valued function f is quasiconcave if and only if all its superlevel sets are convex sets.

[0169] *Proof.* First, consider the case where f is quasiconcave. Let S_c be a superlevel set of f . Then for any $x, y \in S_c(f)$ and $\alpha \in [0, 1]$, quasiconcavity implies $f(\alpha x + (1 - \alpha)y) \geq \min\{f(x), f(y)\} \geq c$, where the last inequality follows from the fact x and y are in the superlevel set $S_c(f)$. The inequality witnesses that the convex combination of x with y also belongs to the superlevel set $S_c(f)$. This proves $S_c(f)$ is a convex set.

[0170] Next, consider the case where all superlevel sets $S_c(f)$ of f are convex sets. Then for any $x, y \in \text{dom}(f)$ and $\alpha \in [0, 1]$, a $c = \min\{f(x), f(y)\}$ can be selected. Then, convexity implies the convex combination $\alpha x + (1 - \alpha)y \in S_c(f) \subseteq \text{dom}(f)$. Thus, $f(\alpha x + (1 - \alpha)y) \geq c = \min\{f(x), f(y)\}$. This proves that f is quasiconcave.

[0171] **Definition A.1** (Graph and Epigraph). The graph of a real-valued function f is defined as:

$$\text{graph}(f) = \{(x, f(x)): x \in \text{dom}(f)\}.$$

A function f generates a set S , if $S = \text{graph}(f)$. The *epigraph* of a real-valued function f is defined as

$$\text{epi}(f) = \{(x, y): x \in \text{dom}(f); y \geq f(x)\}.$$

[0172] **Lemma A.3.** A real-valued function f is convex if and only if $\text{epi}(f)$ is a convex set.

[0173] *Proof.* Consider the case where f is convex. Let $(x_1, y_1), \dots, (x_m, y_m) \in \text{epi}(f)$ and $\sum_{i=1}^m \alpha_i = 1$. Then:

$$\begin{aligned} \sum_{i=1}^m \alpha_i \cdot (x_i, y_i) &\geq \sum_{i=1}^m \alpha_i \cdot (x_i, f(x_i)) \\ &= \left(\sum_{i=1}^m \alpha_i x_i, \sum_{i=1}^m \alpha_i f(x_i) \right) \\ &\geq \left(\sum_{i=1}^m \alpha_i x_i, f\left(\sum_{i=1}^m \alpha_i x_i\right) \right) \end{aligned}$$

{By convexity of f }. The above chain of inequalities proves $\sum_{i=1}^m \alpha_i x_i, \sum_{i=1}^m \alpha_i y_i \in \text{epi}(f)$. Thus $\text{epi}(f)$ is convex.

[0174] Next, consider the case where $\text{epi}(f)$ is a convex set. Let $x, \dots, x_m \in \text{dom}(f)$, and $\sum_{i=1}^m \alpha_i = 1$. By definition, $(x_i, f(x_i)) \in \text{epi}(f)$. Then:

$$\left(\sum_{i=1}^m \alpha_i \cdot x_i, \sum_{i=1}^m \alpha_i \cdot f(x_i) \right) = \sum_{i=1}^m \alpha_i \cdot (x_i, f(x_i)) \in \text{epi}(f)$$

{By convexity of $\text{epi}(f)$ }. This proves $\sum_{i=1}^m \alpha_i \cdot x_i \in \text{dom}(f)$ and $\sum_{i=1}^m \alpha_i \cdot f(x_i) \geq f(\sum_{i=1}^m \alpha_i \cdot x_i)$. Thus, f is a convex function.

[0175] **Restatement (Lemma B.3).** If $\mathbb{R} \rightarrow \mathbb{R}$ is a convex function, then the slope function R (Definition B.2) of f is increasing on all dimensions. That is, for all $x, y, z \in \text{dom}(f)$,

- if $x \leq z$, then $R(x, y) \leq R(z, y)$, and
- if $y \leq z$, then $R(x, y) \leq R(x, z)$.

[0176] *Proof.* R is a symmetric function because for all $x, y \in \text{dom}(f)$, it is obtained that $R(x, y) = R(y, x)$. Thus it suffices to show that for fixed $x \in \text{dom}(f)$, the function $h(y) = R(x, y)$ defined over $\text{dom}(f)$ is an increasing function. Pick any point $z \in \text{dom}(f)$ such that $z > y$. Then it suffices to show that $h(y) \leq h(z)$. Consider separately the cases where (1) $x \leq y \leq z$,

(2) $y \leq x \leq z$, and (3) $y \leq z \leq x$. For each case, it is observed that the midpoint can be defined as a convex combination of the extreme points. That is, the first case implies there is an $\alpha \in [0, 1]$ such that $y = \alpha z + (1 - \alpha)x$. Then convexity of f implies

$$f(y) \leq \alpha f(z) + (1 - \alpha)f(x).$$

Rearranging the inequality and observing that $\alpha = \frac{y-x}{z-x}$, one obtains that

$$h(y) = \frac{f(y) - f(x)}{y - x} \leq \frac{f(z) - f(x)}{z - x} = h(z).$$

The analysis of the second and third cases is similar. This proves the slope function is an increasing function as desired.

B. Proof of Pricing Lemma

[0177] Restatement (Lemma 2.2). Consider states X and X' , where $\phi(X) = \phi(X')$ and $X'_1 < X_1$, and assume the potential function ϕ is quasiconcave and strictly increasing. Then the following hold:

- If $\text{BUY}(q)$ can successfully execute at both X and X' , then $Y(X, \text{BUY}(q)) \leq Y(X', \text{BUY}(q))$
- If $\text{SELL}(q)$ can successfully execute at both X and X' , then $Y(X, \text{SELL}(q)) \leq Y(X', \text{SELL}(q))$

[0178] Consider FIG. 3, where the curve represents the set of reachable states. It is shown hereing that the assumption the potential function is strictly increasing and quasiconcave implies the curve is the graph of a convex function f . Then the payment inequalities will follow from observing the slope of f is decreasing (since f is a convex function).

[0179] One might wonder if there are two distinct reachable states $(X_1, X_2), (X_1, X'_2) \in L_c(\phi)$ or $(X_1, X_2), (X'_1, X_2) \in L_c(\phi)$. Interestingly, the fact ϕ is strictly increasing precludes this because there is a bijective real-valued function f that generates the set of reachable states $L_c(\phi)$.

[0180] Lemma B.1. Let ϕ be a strictly increasing potential function. Then there is a bijective real-valued function f that generates level set $L_c(\phi)$ in the following sense: for all $(X_1, X_2) \in L_c(\phi)$, $X_2 = f(X_1)$ and $X_1 = f^{-1}(X_2)$.

[0181] Proof. Fix $(X_1, X_2) \in L_c(\phi)$, and $(X_1, X'_2) \in \text{dom}(\phi)$. Without loss of generality, let $X_2 > X'_2$. Then the fact ϕ is strictly increasing implies $\phi(X_1, X_2) > \phi(X_1, X'_2)$. Thus (X_1, X_2) and (X_1, X'_2) are not in the same level set. This proves for X_1 , there is a unique X_2 , such that $\phi(X_1, X_2) = c$. Define $f(X_1) = X_2$ for all such X_1 .

[0182] With a similar argument, it can be shown that for each X_2 , there is a unique X_1 such that $\phi(X_1, X_2) = c$. Define $g(X_2) = X_1$ for all such X_2 . Finally, claim $g = f^{-1}$. To see this, observe $g(f(X_1)) = g(X_2) = X_1$. Thus, g is the inverse function of f , as desired.

[0183] Lemma B.1 ensures the existence of a generator f for the set of reachable states $L_c(\phi)$, as long as the potential function ϕ is strictly increasing. An interpretation for this result is that the reserves for token i uniquely determine the reserves for token $3 - i$ (for $i \in \{1, 2\}$). That is, if X_1 is known, then $X_2 = f(X_1)$. Equivalently, if X_2 is known, then $X_1 = f^{-1}(X_2)$.

[0184] Lemma 2.2 may then be reinterpreted as stating that the price for token i increases as X_i decreases (since X_{3-i} is completely determined from X_i). To prove Lemma 2.2, it can be argued that f is a convex function, which follows from the quasiconcavity of ϕ . If ϕ is quasiconcave, a well-known fact is that the superlevel set $S_c(\phi)$ is a convex set. Interestingly, $S_c(\phi)$ can be related with f by observing $S_c(\phi) = \text{epi}(f)$, which follows from the fact ϕ is strictly increasing. All that is left is to use a well-known property for convex functions: a real-valued function f is convex if and only if $\text{epi}(f)$ is a convex set.

[0185] **Definition B.1** (Convex function). A real-valued function f is convex if $\text{dom}(f)$ is a convex set, and for all $x, y \in \text{dom}(f)$, and all $\alpha \in [0, 1]$:

$$f(\alpha \cdot x + (1 - \alpha) \cdot y) \leq \alpha \cdot f(x) + (1 - \alpha) \cdot f(y).$$

[0186] **Lemma B.2.** Let ϕ be strictly increasing and quasiconcave. Then there is a bijective convex function f that generates level set $L_c(\phi)$.

[0187] *Proof.* Applying Lemma B.1 with strictly increasing function ϕ implies the existence of a bijective function f that generates $L_c(\phi)$. Claim $\text{epi}(f) = S_c(\phi)$ for superlevel set $S_c(\phi)$. First, consider the case where $(X_1, X_2) \in S_c(\phi)$. By definition, $\phi(X_1, f(X_1)) = c$ and $\phi(X_1, X_2) \geq c$. Because ϕ is strictly increasing, it can be concluded that $X_2 \geq f(X_1)$. This proves $(X_1, X_2) \in \text{epi}(f)$. Next, consider the case where $(X_1, X_2) \in \text{epi}(f)$. By definition, $X_2 \geq f(X_1)$. Because ϕ is strictly increasing, $\phi(X_1, X_2) \geq \phi(X_1, f(X_1)) = c$. This proves $(X_1, X_2) \in S_c(\phi)$. Both cases prove $\text{epi}(f) = S_c(\phi)$. Next, consider two well-known facts from convexity theory:

- A real-valued function is quasiconcave if and only if all its superlevel sets are convex sets.
- A real-valued function is convex if and only if its epigraph is a convex set.

The first bullet implies $S_c(\phi) = \text{epi}(f)$ is a convex set since ϕ is quasiconcave. The second bullet implies f is a convex function, as desired.

[0188] All that is left is the following well-known fact from convexity theory.

[0189] **Definition B.2** (Slope function). The *slope function* of $f: \mathbb{R} \rightarrow \mathbb{R}$ maps $x, y \in \text{dom}(f)$ to $R(x, y) = \frac{f(y)-f(x)}{y-x}$, the slope of the line connecting $(x, f(x))$ to $(y, f(y))$.

[0190] **Lemma B.3.** If f is a convex function, then the slope function R of f is increasing on all dimensions. That is, for all $x, y, z \in \text{dom}(f)$,

- if $x \leq z$, then $R(x, y) \leq R(z, y)$, and
- if $y \leq z$, then $R(x, y) \leq R(x, z)$.

A proof of the above is included in Section A, and conclude with the proof of Lemma 2.2.

[0191] *Proof of Lemma 2.2.* Let f be the generator of level set $L_c(\phi)$ and recall f is a convex function as per Lemma B.2. Using the fact that $X'_1 \leq X_1$, the following is obtained:

$$\begin{aligned} R(X'_1 - q, X'_1) &\leq R(X_1 - q, X'_1) \quad \{\text{by Lemma B.3}\} \\ &\leq R(X_1 - q, X_1) \quad \{\text{by Lemma B.3}\}. \end{aligned}$$

Expanding on the definition of R , the following is derived:

$$f(X'_1 - q) - f(X'_1) \geq f(X_1 - q) - f(X_1).$$

[0192] **Claim B.1.** Let A be an order that can successfully execute at state $X = (X_1, f(X_1))$.

The following are true:

- if $A = \text{BUY}(q)$, then $Y(X, A) = f(X_1 - q) - f(X_1)$, and
- if $A = \text{SELL}(q)$, then $Y(X, A) = f(X_1 + q) - f(X_1)$.

[0193] *Proof.* Let $Z = (Z_1, f(Z_1))$ be the state after A executes at X . By definition, $Y(X, A) = f(Z_1) - f(X_1)$. If $A = \text{BUY}(q)$, then $Z_1 = X_1 - q$. If $A = \text{SELL}(q)$, then $Z_1 = X_1 + q$. This proves the claim.

[0194] The claim implies that $Y(X', \text{BUY}(q)) = f(X'_1 - q) - f(X'_1) \geq f(X_1 - q) - f(X_1) = Y(X, \text{BUY}(q))$, as desired. The proof that $Y(X, \text{SELL}(q)) \leq Y(X', \text{SELL}(q))$ follows a similar format: because R is increasing and $X'_1 \leq X_1$, the following is obtained:

$$f(X'_1 + q) - f(X'_1) = q \cdot R(X'_1 + q, X'_1) \geq q \cdot R(X_1 - q, X_1) = f(X_1 + q) - f(X_1).$$

Finally, observe that $Y(X', \text{SELL}(q)) = f(X'_1 + q) - f(X'_1)$ and $Y(X, \text{SELL}(q)) = f(X_1 + q) - f(X_1)$. This proves Lemma 2.2.

C. Proof of Duality Theorem

[0195] *Restatement* (Theorem 5.1). Consider any liquidity pool exchange with potential ϕ . For any pair of states $X, X' \in L_c(\phi)$ either:

- any buy order receives a better execution at X than X' , or

- any sell order receives a better execution at X than X' .

[0196] First, a *monotonicity property* is proved for the quality of execution of an order. Consider states $X, X' \in L_c(\phi)$ where the token 1 reserves at X are smaller than at X' . It is argued that any buy order receives a better execution at X' than at X . Similarly, it is argued that any sell order receives a better execution at X than at X' .

[0197] **Observation C.1.** Let f be the generator of $L_c(\phi)$ and $X \in L_c(\phi)$, (Lemma B.1). $\text{BUY}(q)$ can successfully execute at X if and only if $X_1, X_1 - q \in \text{dom}(f)$. $\text{SELL}(q)$ can successfully execute at X if and only if $X_1, X_1 + q \in \text{dom}(f)$.

[0198] *Proof.* The fact $X_1 \in \text{dom}(f)$ follows from the fact f generates $L_c(\phi)$ and $X \in L_c(\phi)$. First, we will consider a buy order $\text{BUY}(q)$. If $\text{BUY}(q)$ can successfully execute at X , then $(X_1 - q, f(X_1 - q)) \in L_c(\phi)$ which implies $X_1 - q \in \text{dom}(f)$. For the converse, if $f(X_1 - q) \in \text{dom}(f)$, then $(X_1 - q, f(X_1 - q)) \in L_c(\phi)$ which implies $\text{BUY}(q)$ can successfully execute at X .

[0199] Next, consider a sell order $\text{SELL}(q)$. If $\text{SELL}(q)$ can successfully execute at X , then $(X_1 + q, f(X_1 + q)) \in L_c(\phi)$ which implies $X_1 + q \in \text{dom}(f)$. For the converse, if $f(X_1 + q) \in \text{dom}(f)$, then $(X_1 + q, f(X_1 + q)) \in L_c(\phi)$ which implies $\text{SELL}(q)$ can successfully execute at X .

[0200] **Lemma C.1.** Consider states $X, X' \in L_c(\phi)$ where $X_1 \leq X'_1$. If $\text{BUY}(q, p)$ can successfully execute at X , then $\text{BUY}(q, p)$ can successfully execute at X' . If $\text{SELL}(q, p)$ can successfully execute at X' , then $\text{SELL}(q, p)$ can successfully execute at X .

[0201] *Proof.* Let f be the convex function that generates the set of reachable states $L_c(\phi)$ (Lemma B.2). First, it will be proven that $\text{BUY}(q, p)$ can successfully execute at X' . The fact that $X, X' \in L_c(\phi)$ and $\text{BUY}(q, p)$ can successfully execute at X implies $X'_1, X_1, X_1 - q \in \text{dom}(f)$ (Observation C.1). Note $X'_1 - q$ is a point between $X_1 - q$ and X'_1 and thus the convexity of $\text{dom}(f)$ implies $X'_1 - q \in \text{dom}(f)$. This proves $\text{BUY}(q)$ can successfully execute at X' (Observation C.1), and remaining to show is $Y(X', \text{BUY}(q)) \leq q \cdot p$. Because the token 1 reserves at X are lower than at X' , $\text{BUY}(q)$, pays less by executing at X' than X (Lemma 2.2). Thus the payment by executing at X' is at most $q \cdot p$ since the payment by executing at X is at most $q \cdot p$. This proves $\text{BUY}(q, p)$ can successfully execute at X' .

[0202] Next, it will be proven that $\text{SELL}(q, p)$ can successfully execute at X . The fact that $X, X' \in L_c(\phi)$ and $\text{SELL}(q, p)$ can successfully execute at X' implies $X_1, X'_1, X'_1 + q \in \text{dom}(f)$

(Observation C.1). Note that $X_1 + q$ is a point between X_1 and $X'_1 + q$ and thus the convexity of $\text{dom}(f)$ implies $X_1 + q \in \text{dom}(f)$. This proves that $\text{SELL}(q)$ can successfully execute at X (Observation C.1), and it remains to show $Y(X, \text{SELL}(q)) \geq q \cdot p$. Because token 1 reserves at X are lower than X' , $\text{SELL}(q)$ receives more tokens by executing at X than X' (Lemma 2.2). Thus $\text{SELL}(q)$ receives at least $q \cdot p$ tokens by executing at X , since $\text{SELL}(q)$ receives at least $q \cdot p$ by executing at X' . This proves $\text{SELL}(q, p)$ can successfully execute at X .

[0203] Corollary C.1. Consider states $X, X' \in L_c(\phi)$, where $X_1 \leq X'_1$. $\text{BUY}(q, p)$ receives a better execution at X' than X . Moreover, $\text{SELL}(q, p)$ receives a better execution at X than X' .

[0204] Proof. If $\text{BUY}(q, p)$ fails to execute at X , then $\text{BUY}(q, p)$ receives a better execution at X' (by definition). If $\text{BUY}(q, p)$ can successfully execute at X , then $\text{BUY}(q, p)$ also successfully executes at X' (Lemma C.1). From the Pricing Lemma (Lemma 2.2): $Y(X', \text{BUY}(q))$. This proves that $\text{BUY}(q, p)$ receives a better execution at X' than X .

[0205] If $\text{SELL}(q, p)$ fails to execute X' , then $\text{SELL}(q, p)$ receives a better execution at X (by definition). If $\text{SELL}(q, p)$ can successfully execute at X' then $\text{SELL}(q, p)$ can also successfully execute at X (Lemma C.1). From the Pricing Lemma (Lemma 2.2): $Y(X, \text{SELL}(q)) \geq Y(X', \text{SELL}(q))$. This proves that $\text{SELL}(q, p)$ receives a better execution at X than X' .

[0206] Proof of Theorem 5.1. First, consider the case where the token 1 reserves at X are smaller than those at X' . From Corollary C.1, any sell order receives a better execution at X than X' . Second, consider the case where the token 1 reserves at X are larger than at X' . From Corollary C.1, any buy order receives a better execution at X than X' .

[0207] Having thus described several aspects and embodiments of the technology set forth in the disclosure, it is to be appreciated that various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications, and improvements are intended to be within the spirit and scope of the technology described herein. For example, those of ordinary skill in the art will readily envision a variety of other means and/or structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein, and each of such variations and/or modifications is deemed to be within the scope of the embodiments described herein. Those skilled in the art will recognize or be able to ascertain using no more than routine experimentation many equivalents to the specific embodiments described herein. It is, therefore, to be understood that the foregoing embodiments

are presented by way of example only and that, within the scope of the appended claims and equivalents thereto, inventive embodiments may be practiced otherwise than as specifically described. In addition, any combination of two or more features, systems, articles, materials, kits, and/or methods described herein, if such features, systems, articles, materials, kits, and/or methods are not mutually inconsistent, is included within the scope of the present disclosure.

[0208] The above-described embodiments can be implemented in any of numerous ways. One or more aspects and embodiments of the present disclosure involving the performance of processes or methods may utilize program instructions executable by a device (e.g., a computer, a processor, or other device) to perform, or control performance of, the processes or methods. In this respect, various inventive concepts may be embodied as a computer readable storage medium (or multiple computer readable storage media) (e.g., a computer memory, one or more floppy discs, compact discs, optical discs, magnetic tapes, flash memories, circuit configurations in Field Programmable Gate Arrays or other semiconductor devices, or other tangible computer storage medium) encoded with one or more programs that, when executed on one or more computers or other processors, perform methods that implement one or more of the various embodiments described above. The computer readable medium or media can be transportable, such that the program or programs stored thereon can be loaded onto one or more different computers or other processors to implement various ones of the aspects described above. In some embodiments, computer readable media may be tangible (e.g., non-transitory) computer readable media. In some embodiments, the computer readable media may comprise a persistent memory.

[0209] The terms “program” or “software” are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects as described above. Additionally, it should be appreciated that according to one aspect, one or more computer programs that when executed perform methods of the present disclosure need not reside on a single computer or processor but may be distributed in a modular fashion among a number of different computers or processors to implement various aspects of the present disclosure.

[0210] Computer-executable instructions may be in many forms, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0211] Also, data structures may be stored in computer-readable media in any suitable form. For simplicity of illustration, data structures may be shown to have fields that are related through location in the data structure. Such relationships may likewise be achieved by assigning storage for the fields with locations in a computer-readable medium that convey relationship between the fields. However, any suitable mechanism may be used to establish a relationship between information in fields of a data structure, including through the use of pointers, tags or other mechanisms that establish relationship between data elements.

[0212] When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers.

[0213] Further, it should be appreciated that a computer may be embodied in any of a number of forms, such as a rack-mounted computer, a desktop computer, a laptop computer, a quantum computer, or a tablet computer, as non-limiting examples. Additionally, a computer may be embedded in a device not generally regarded as a computer but with suitable processing capabilities, including a Personal Digital Assistant (PDA), a smartphone, or any other suitable portable or fixed electronic device.

[0214] Also, a computer may have one or more input and output devices. These devices can be used, among other things, to present a user interface. Examples of output devices that can be used to provide a user interface include printers or display screens for visual presentation of output and speakers or other sound generating devices for audible presentation of output. Examples of input devices that can be used for a user interface include keyboards, and pointing devices, such as mice, touch pads, and digitizing tablets. As another example, a computer may receive input information through speech recognition or in other audible formats.

[0215] Such computers may be interconnected by one or more networks in any suitable form, including a local area network or a wide area network, such as an enterprise network, and intelligent network (IN) or the Internet. Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0216] Also, as described, some aspects may be embodied as one or more methods. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may

include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[0217] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[0218] The indefinite articles “a” and “an,” as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean “at least one.”

[0219] The use of “coupled” or “connected” is meant to refer to elements, or signals, that are either directly linked to one another or are linked through intermediate components. Elements that are not “coupled” or “connected” are “decoupled” or “disconnected.”

[0220] The use of “between” in a coupled signal chain is not meant to require a particular direction of signal flow in the signal chain unless stated otherwise. For instance, where element B is described as coupled between elements A and C in a signal chain, signals may flow from element A to element C through element B and/or from element C to element A through element B unless stated otherwise.

[0221] The phrase “and/or,” as used herein in the specification and in the claims, should be understood to mean “either or both” of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with “and/or” should be construed in the same fashion, i.e., “one or more” of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the “and/or” clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to “A and/or B,” when used in conjunction with open-ended language such as “comprising” can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment, to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

[0222] As used herein in the specification and in the claims, the phrase “at least one,” in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list

of elements to which the phrase “at least one” refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, “at least one of A and B” (or, equivalently, “at least one of A or B,” or, equivalently “at least one of A and/or B”) can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally including more than one, B, with no A present (and optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

[0223] In the claims, as well as in the specification above, all transitional phrases such as “comprising,” “including,” “carrying,” “having,” “containing,” “involving,” “holding,” “composed of,” and the like are to be understood to be open-ended, i.e., to mean including but not limited to. Only the transitional phrases “consisting of” and “consisting essentially of” shall be closed or semi-closed transitional phrases, respectively.

[0224] The terms “approximately” and “about” may be used to mean within $\pm 20\%$ of a target value in some embodiments, within $\pm 10\%$ of a target value in some embodiments, within $\pm 5\%$ of a target value in some embodiments, within $\pm 2\%$ of a target value in some embodiments. The terms “approximately” and “about” may include the target value.

[0225] Use of ordinal terms such as “first,” “second,” “third,” etc., in the claims to modify a claim element does not by itself connote any priority, precedence, or order of one claim element over another or the temporal order in which acts of a method are performed, but are used merely as labels to distinguish one claim element having a certain name from another element having a same name (but for use of the ordinal term) to distinguish the claim elements.

CLAIMS

What is claimed is:

1. A computer-implemented method for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the computer-implemented method comprising:
 - receiving transaction information specifying a set of transactions, wherein:
 - transactions of the set of transactions are to be ordered prior to execution, and
 - each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens;
 - ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions; and
 - causing the execution of the ordered transactions.
2. The computer-implemented method of claim 1, wherein causing the execution of the ordered transactions comprises transmitting, from a first computer to a second computer, the transaction block for execution.
3. The computer-implemented method of claim 1 or 2, wherein the verifiable sequencing rule is arranged to ensure that:
 - (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,
 - (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or
 - (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.
4. The computer-implemented method of any one of claims 1-3, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises, if the set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

5. The computer-implemented method of claim 4, further comprising, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block:

removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

6. The computer-implemented method of claim 5, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises:

if the updated set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

7. The computer-implemented method of claim 5, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises, if the updated set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

8. The computer-implemented method of claim 1, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises:

if the set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

9. The computer-implemented method of any one of claims 1-8, wherein ordering the transactions of the set of transactions further comprises executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

10. At least one non-transitory computer readable medium storing processor-executable instructions that, when executed by at least one processor, cause the at least one processor to perform a method of implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the method comprising:

receiving transaction information specifying a set of transactions, wherein:

transactions of the set of transactions are to be ordered prior to execution, and

each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens;

ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions; and

causing the execution of the ordered transactions.

11. The at least one non-transitory computer readable medium of claim 10, wherein causing the execution of the ordered transactions comprises transmitting, from the at least one processor to another at least one processor, the transaction block for execution.

12. The at least one non-transitory computer readable medium of claim 10 or 11, wherein the verifiable sequencing rule is arranged to ensure that:

(i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,

(ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or

(iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

13. The at least one non-transitory computer readable medium of any one of claims 10-12, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises, if the set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

14. The at least one non-transitory computer readable medium of claim 13, further comprising, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block:

removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

15. The at least one non-transitory computer readable medium of claim 14, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises:

if the updated set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

16. The at least one non-transitory computer readable medium of claim 14, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises, if the updated set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

17. The at least one non-transitory computer readable medium of claim 10, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises:

if the set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

18. The at least one non-transitory computer readable medium of any one of claims 10-17, wherein ordering the transactions of the set of transactions further comprises executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

19. A system for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the system comprising:
- at least one processor; and
 - at least one non-transitory computer readable medium storing processor-executable instructions that, when executed by the at least one processor, cause the at least one processor to perform a method comprising:
 - receiving transaction information specifying a set of transactions, wherein:
 - transactions of the set of transactions are to be ordered prior to execution,
 - and
 - each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens;
 - ordering, using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions; and
 - causing the execution of the ordered transactions.
20. The system of claim 19, wherein causing the execution of the ordered transactions comprises transmitting, from the at least one processor to another at least one processor, the transaction block for execution.
21. The system of claim 19 or 20, wherein the verifiable sequencing rule is arranged to ensure that:
- (i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,
 - (ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or
 - (iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

22. The system of any one of claims 19-21, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises, if the set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

23. The system of claim 22, wherein ordering the transactions of the set of transactions further comprises, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block:

removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

24. The system of claim 23, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises:

if the updated set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

25. The system of claim 23, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises, if the updated set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

26. The system of claim 19, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises:

if the set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

27. The system of any one of claims 19-26, wherein ordering the transactions of the set of transactions further comprises executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

28. A computer-implemented method for implementing a verifiable transaction service on a decentralized exchange of a liquidity pool of two or more tokens, the computer-implemented method comprising:

transmitting, from trading computers to one or more building computers, transaction information specifying a set of transactions, wherein:

transactions of the set of transactions are to be ordered prior to execution, and

each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens;

ordering, at each of the one or more building computers and using a verifiable sequencing rule, transactions of the set of transactions to obtain a transaction block of ordered transactions;

transmitting, from the one or more building computers to a transmitting computer, ordered transaction blocks;

selecting, using the transmitting computer, a transaction block of the ordered transaction blocks;

transmitting, from the transmitting computer to an executing computer, the selected transaction block; and

executing, using the executing computer, the transaction block.

29. The computer-implemented method of claim 28, wherein the verifiable sequencing rule is arranged to ensure that:

(i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,

(ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain from the transaction block, or

(iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

30. The computer-implemented method of claim 28 or 29, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises, if the set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

31. The computer-implemented method of claim 30, further comprising, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block:

removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

32. The computer-implemented method of claim 31, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises:

if the updated set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

33. The computer-implemented method of claim 31, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises, if the updated set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

34. The computer-implemented method of claim 28, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises:

if the set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

35. The computer-implemented method of any one of claims 28-34, wherein ordering the transactions of the set of transactions further comprises executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

36. A plurality of non-transitory computer readable media storing processor-executable instructions that, when executed by at least one processor, cause the at least one processor to perform a method of implementing a verifiable transaction on a decentralized exchange of a liquidity pool of two or more tokens, the method comprising:

transmitting, from trading computers to one or more building computers, transaction information specifying a set of transactions, wherein:

transactions of the set of transactions are to be ordered prior to execution, and

each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens;

ordering, at each of the one or more building computers and using a verifiable sequencing rule, transactions of the set of transactions to obtain a transaction block of ordered transactions;

transmitting, from the one or more building computers to a transmitting computer, ordered transaction blocks;

selecting, using the transmitting computer, a transaction block of the ordered transaction blocks;

transmitting, from the transmitting computer to an executing computer, the selected transaction block; and

executing, using the executing computer, the transaction block.

37. The plurality of non-transitory computer readable media of claim 36, wherein the verifiable sequencing rule is arranged to ensure that:

(i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,

(ii) the execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain from the transaction block, or

(iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

38. The plurality of non-transitory computer readable media of claim 36 or 37, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises, if the set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

39. The plurality of non-transitory computer readable media of claim 38, further comprising, after appending the purchase or sale transaction to the end of the ordered transactions in the transaction block:

removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

40. The plurality of non-transitory computer readable media of claim 39, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises:

if the updated set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

41. The plurality of non-transitory computer readable media of claim 39, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises, if the updated set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

42. The plurality of non-transitory computer readable media of claim 36, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises:

if the set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

43. The plurality of non-transitory computer readable media of any one of claims 36-42, wherein ordering the transactions of the set of transactions further comprises executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

44. A distributed system comprising:

a plurality of processors; and

a plurality of non-transitory computer readable medium storing processor-executable instructions that, when executed sequentially by each processor of the plurality of processors, cause the plurality of processors to perform a method of implementing a verifiable transaction on a decentralized exchange of a liquidity pool of two or more tokens, the method comprising:

transmitting, from trading computers to one or more building computers,

transaction information specifying a set of transactions, wherein:

transactions of the set of transactions are to be ordered prior to execution,

and

each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens;

ordering, at each of the one or more building computers and using a verifiable sequencing rule, transactions of the set of transactions to obtain a transaction block of ordered transactions;

transmitting, from the one or more building computers to a transmitting computer, ordered transaction blocks;

selecting, using the transmitting computer, a transaction block of the ordered transaction blocks;

transmitting, from the transmitting computer to an executing computer, the selected transaction block; and

executing, using the executing computer, the transaction block.

45. The distributed system of claim 44, wherein the verifiable sequencing rule is arranged to ensure that:

(i) an execution price of a first transaction of the set of transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,

(ii) an execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or

(iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

46. The distributed system of claim 44 or 45, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises, if the set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the set of transactions to an end of the ordered transactions in the transaction block.

47. The distributed system of claim 46, wherein ordering the transactions of the set of transactions further comprises, after appending the purchase or sale transaction to an end of the ordered transactions in the transaction block:

removing the appended purchase or sale transaction from the set of transactions to obtain an updated set of transactions.

48. The distributed system of claim 47, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises:

if the updated set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

49. The distributed system of claim 47, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule further comprises, if the updated set of transactions includes both purchases and sales:

if a current price of a token is equal to an initial price of the token, appending any sale transaction or any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block;

if a current price of a token is less than an initial price of the token, appending any purchase transaction in the updated set of transactions to an end of the ordered transactions in the transaction block; or

if a current price of a token is greater than an initial price of the token, appending any sale transaction in the updated set of transactions to an end of the ordered transactions in the transaction block.

50. The distributed system of claim 44, wherein ordering the transactions of the set of transactions in the transaction block using the verifiable sequencing rule comprises:

if the set of transactions includes only one of purchases or sales:

appending, in any order, any remaining transactions of the set of transactions to an end of the ordered transactions of the transaction block.

51. The distributed system of any one of claims 44-50, wherein ordering the transactions of the set of transactions further comprises executing a smart contract, the smart contract being configured to order the transactions of the set of transactions according to the verifiable sequencing rule.

52. A computer-implemented method for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the computer-implemented method comprising:

receiving transaction blocks, each comprising ordered transactions, wherein:

the ordered transactions are to be executed according to an ordering of the transactions, and

each of the transactions in the ordered transactions indicates a purchase or a sale of at least one token of the two or more tokens;

verifying that the ordered transactions of the received transaction blocks were ordered in compliance with a verifiable sequencing rule;

selecting a verified transaction block of the received transaction blocks; and

causing the execution of the ordered transactions of the verified transaction block.

53. The computer-implemented method of claim 52, wherein causing the execution of the ordered transactions comprises transmitting, from a first computer to a second computer, the transaction block for execution.

54. The computer-implemented method of claim 52 or 53, wherein the verifiable sequencing rule is arranged to ensure that:

(i) an execution price of a first transaction of the ordered transactions is as good as a standalone price for the first transaction, where the standalone price is a price of the first transaction if the transaction block were to include only the first transaction,

(ii) an execution price of the first transaction is worse than the standalone price, but a builder assembling the transaction block does not gain profit from including the first transaction in the transaction block, or

(iii) actions by the builder to subvert the verifiable sequencing rule and to gain profit are observable by third parties.

55. The computer-implemented method of any one of claims 52-54, wherein verifying that the ordered transactions were ordered in compliance with the verifiable sequencing rule comprises executing a smart contract, the smart contract being configured to verify that the ordered transactions were ordered in compliance with the verifiable sequencing rule.

56. At least one non-transitory computer readable medium storing processor-executable instructions that, when executed by at least one processor, cause the at least one processor to perform a method of implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the method comprising:

receiving transaction blocks, each comprising ordered transactions, wherein:

the ordered transactions are to be executed according to an ordering of the transactions, and

each of the transactions in the ordered transactions indicates a purchase or a sale of at least one token of the two or more tokens;

verifying that the ordered transactions of the received transaction blocks were ordered in compliance with a verifiable sequencing rule;

selecting a verified transaction block of the received transaction blocks; and

causing the execution of the ordered transactions of the verified transaction block.

57. A system for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the system comprising:

at least one processor; and

at least one non-transitory computer readable medium storing processor-executable instructions that, when executed by the at least one processor, cause the at least one processor to perform a method comprising:

receiving transaction blocks, each comprising ordered transactions, wherein:

the ordered transactions are to be executed according to an ordering of the transactions, and

each of the transactions in the ordered transactions indicates a purchase or a sale of at least one token of the two or more tokens;

verifying that the ordered transactions of the received transaction blocks were ordered in compliance with a verifiable sequencing rule;

selecting a verified transaction block of the received transaction blocks; and

causing the execution of the ordered transactions of the verified transaction block.

58. A computer-implemented method for implementing a verifiable transaction block on an exchange of a liquidity pool of two or more tokens, the computer-implemented method comprising:

receiving, at a building computer and from one or more trading computers, transaction information specifying a set of transactions, wherein:

transactions of the set of transactions are to be ordered prior to execution, and

each of the transactions in the set of transactions indicates a purchase or a sale of at least one token of the two or more tokens;

ordering, at the building computer and using a verifiable sequencing rule, the transactions of the set of transactions to obtain a transaction block of ordered transactions; and

causing, at the building computer or at an executing computer, the execution of the ordered transactions.

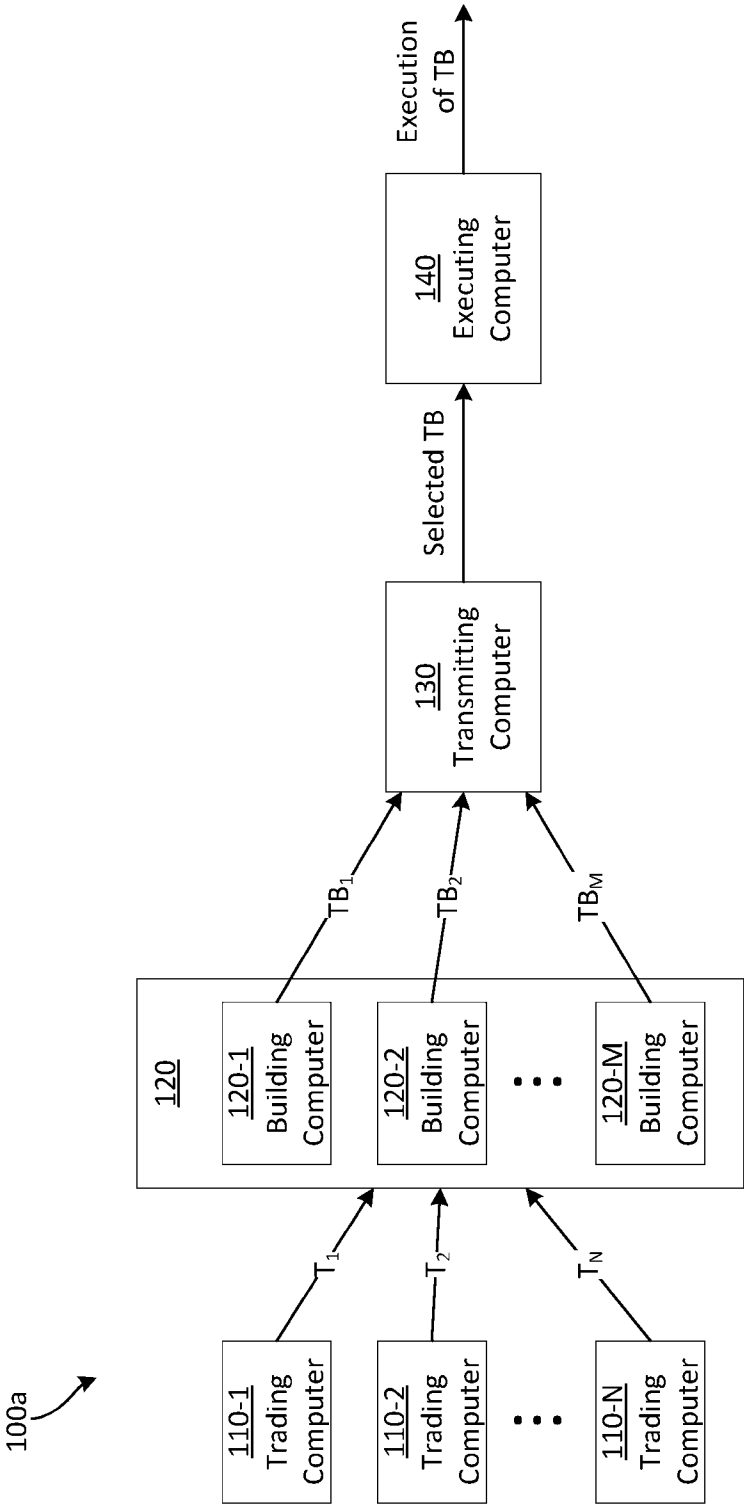


FIG. 1A

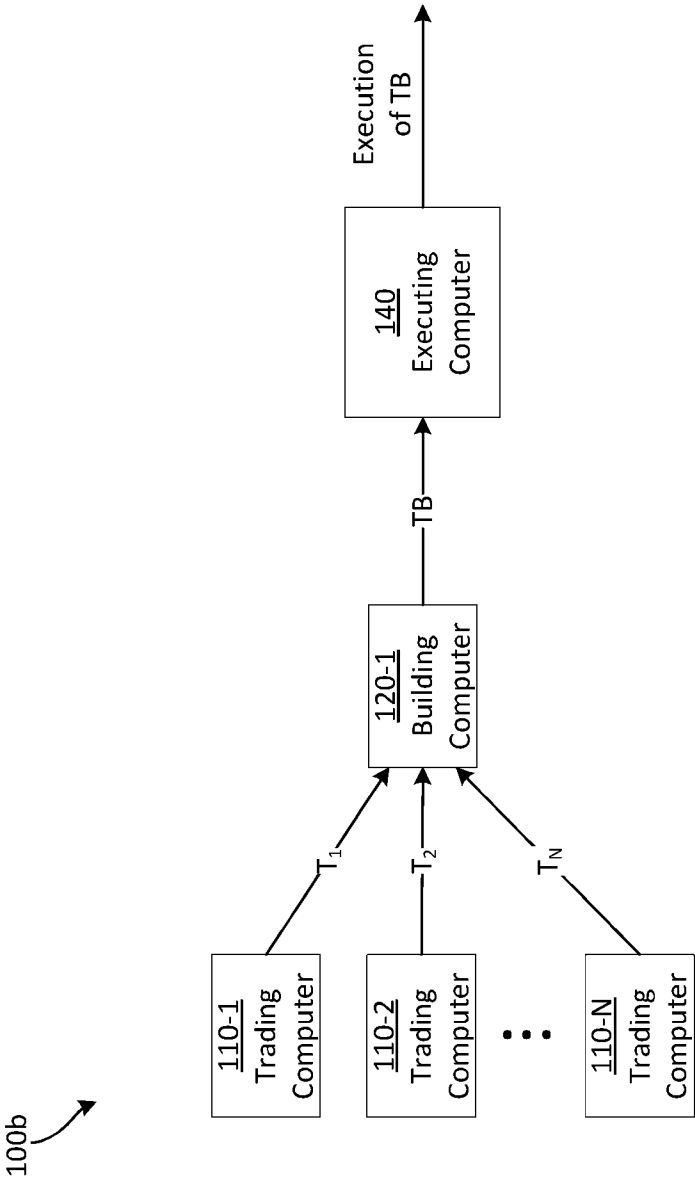


FIG. 1B

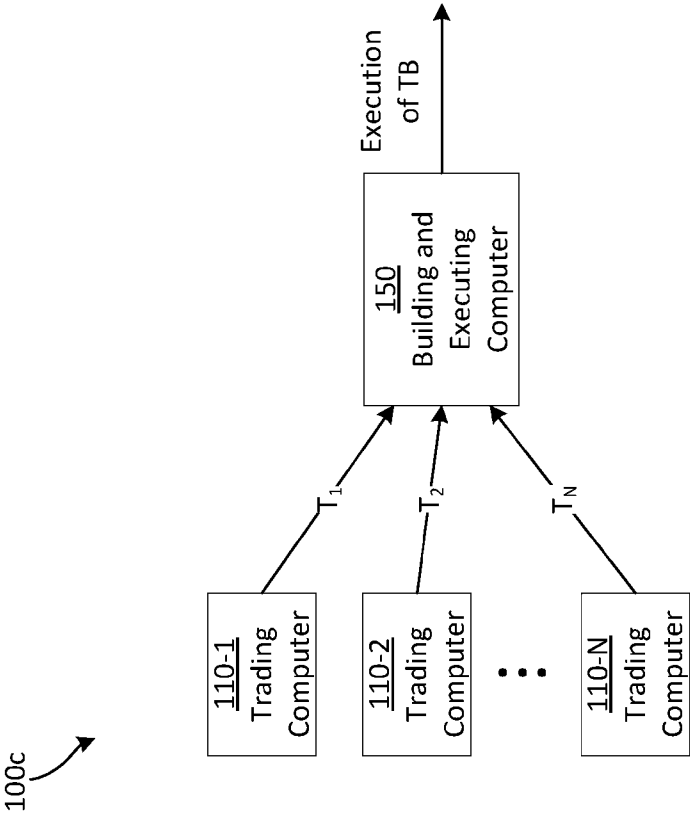


FIG. 1C

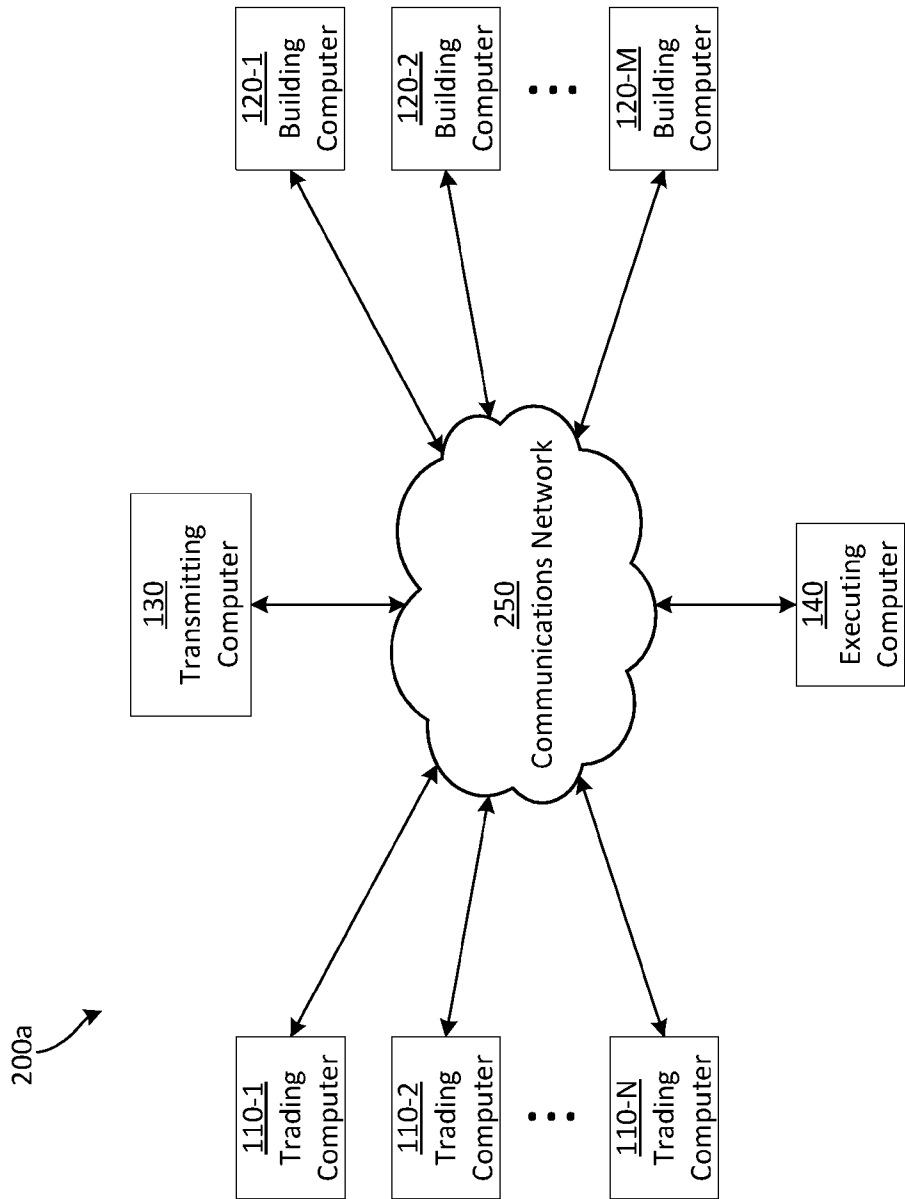


FIG. 2A

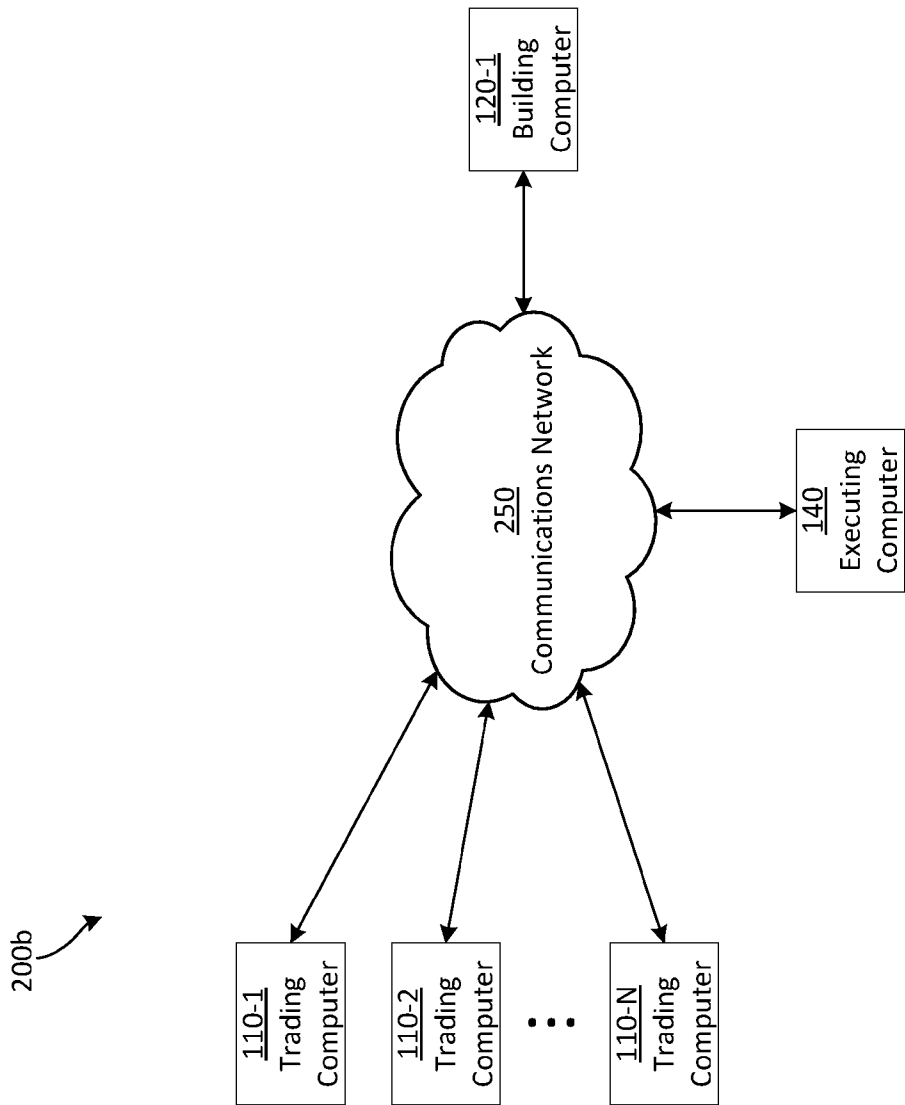


FIG. 2B

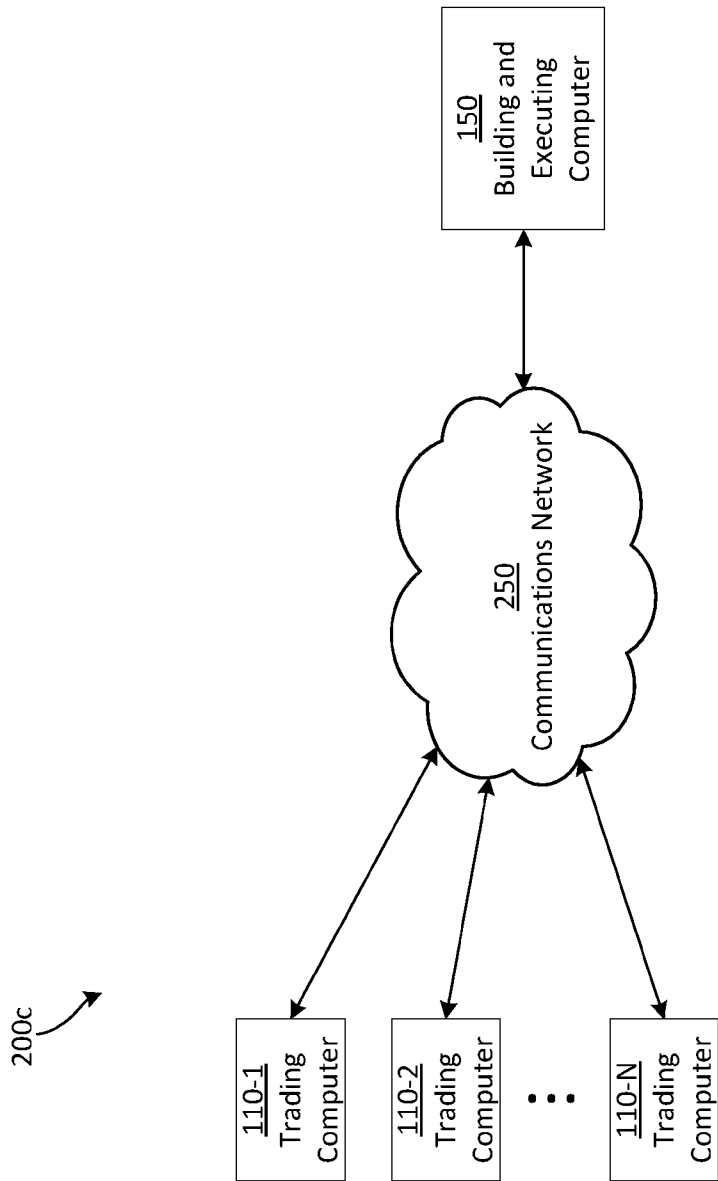
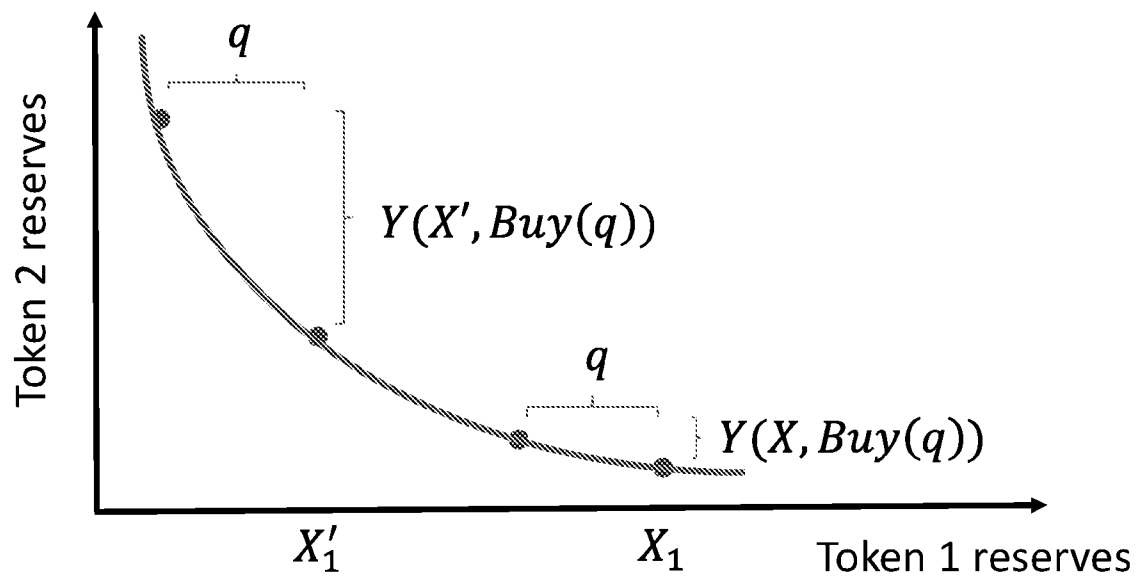
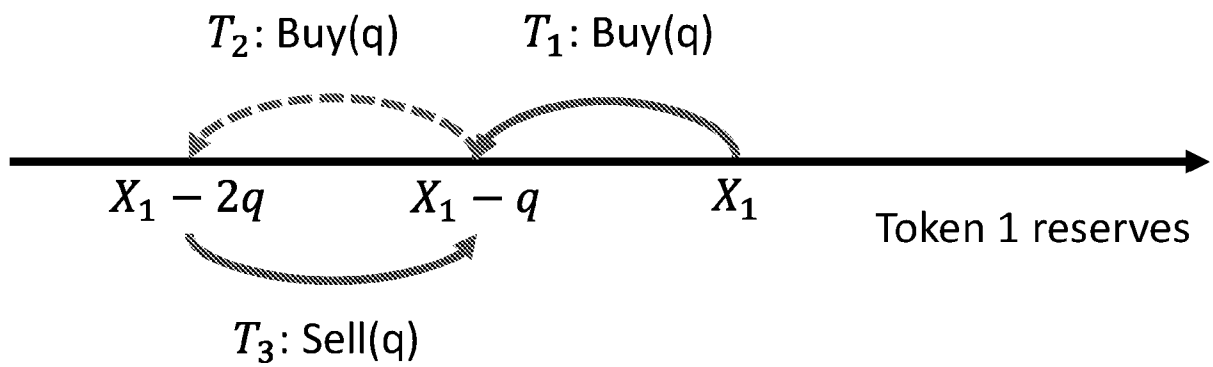
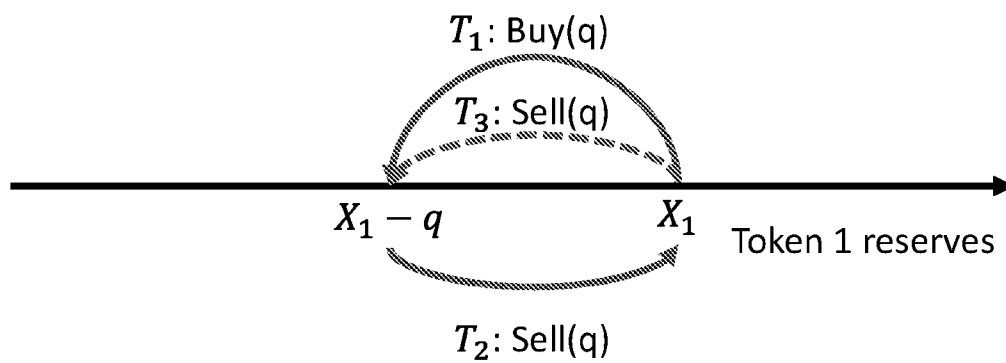


FIG. 2C

**FIG. 3**

**FIG. 4****FIG. 5**

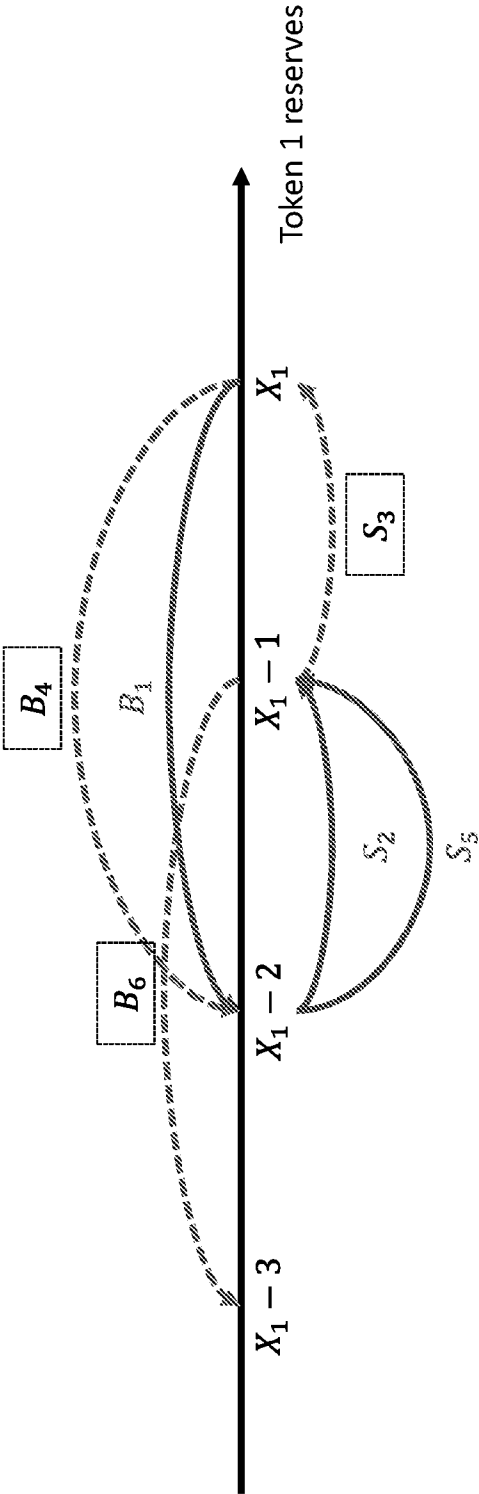


FIG. 6

700
↘

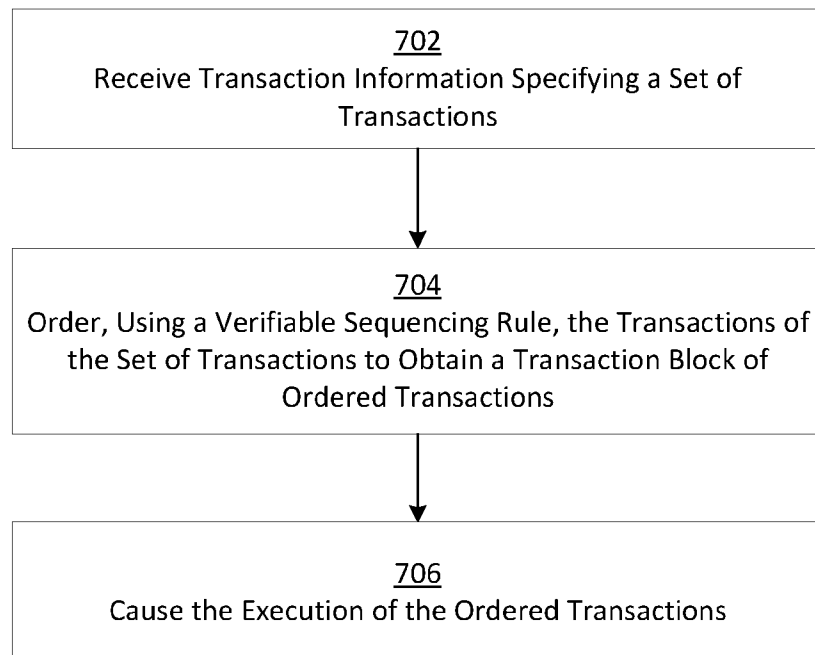


FIG. 7

800
↓

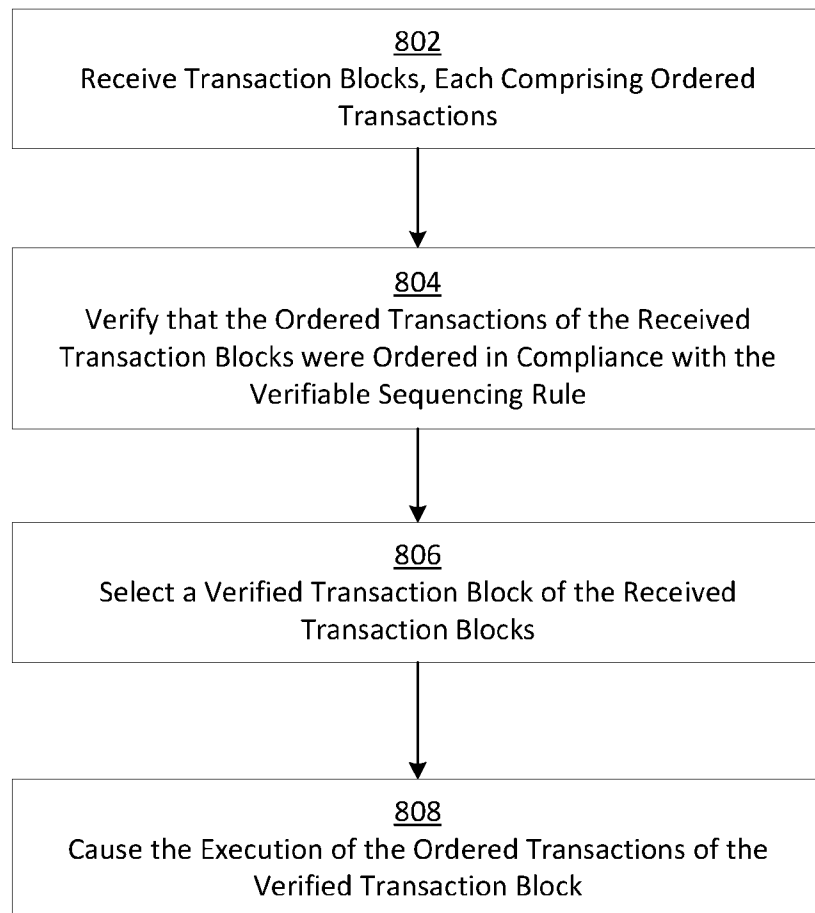
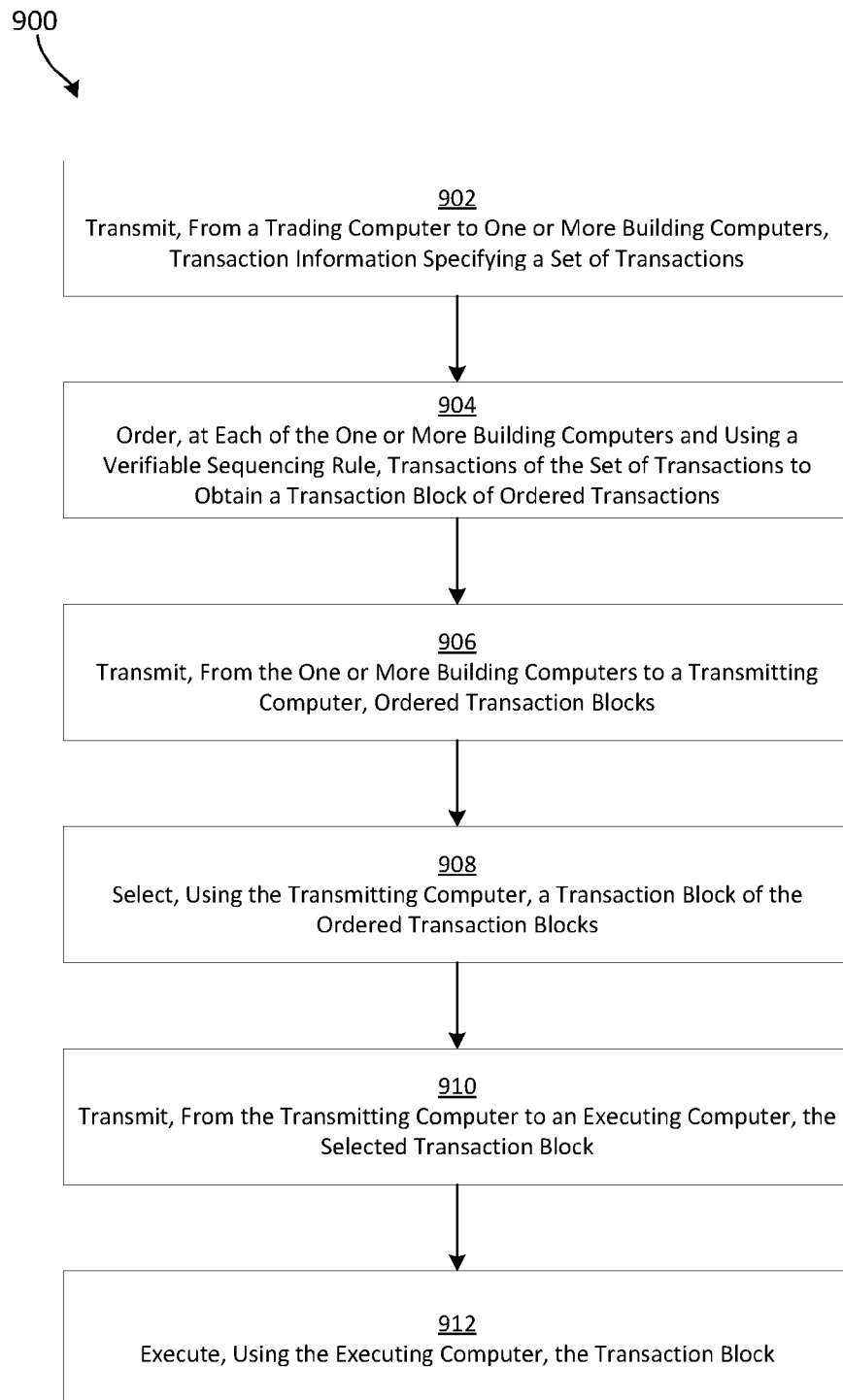


FIG. 8

**FIG. 9**

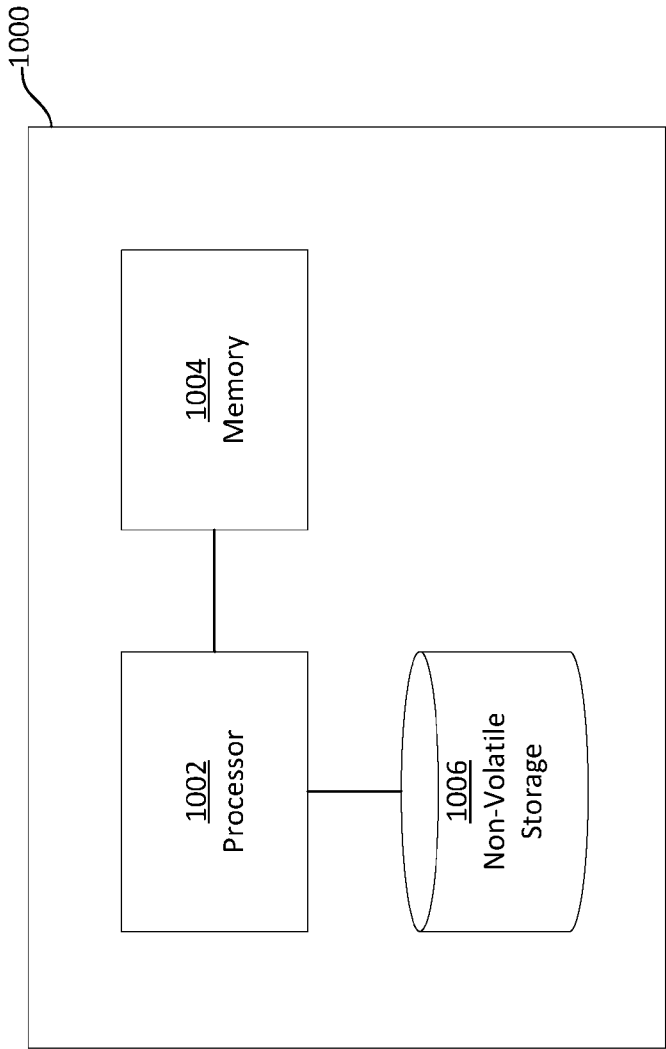


FIG. 10

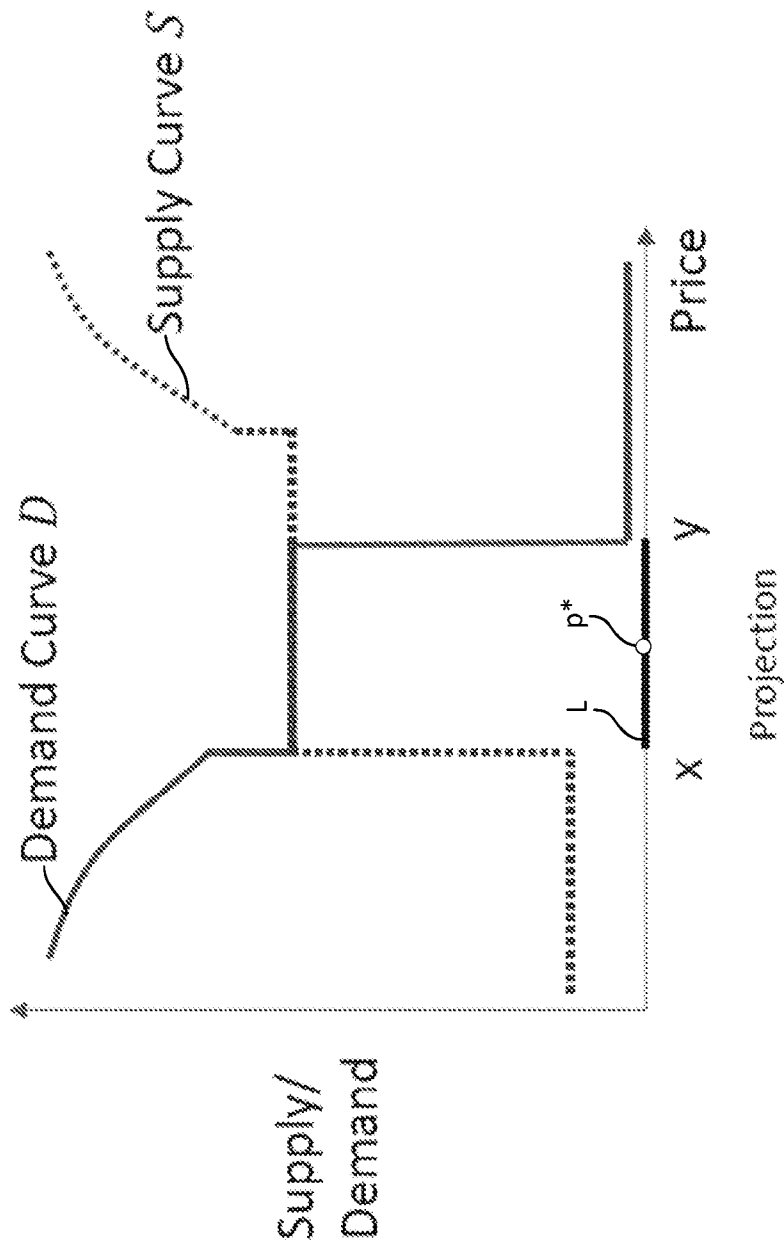


FIG. 11