METHODS AND DEVICES FOR TRANSACTION MATCHING BASED ON BLOCKCHAIN SYSTEM

TECHNICAL FIELD

[0001] The specification relates generally to computer technologies, and more particularly, to methods and devices for transaction matching based on a blockchain system.

BACKGROUND

[0002] Blockchain systems, also known as distributed ledger systems (DLSs) or consensus systems, may enable participating entities to store data securely and immutably. Blockchain systems may include any DLSs, without referencing any particular use case, and may be used for public, private, and consortium blockchain networks. A public blockchain network is open for all entities to use the system and participate in the consensus process. A private blockchain network is provided for a particular entity, which centrally controls read and write permissions. A consortium blockchain network is provided for a select group of entities, which control the consensus process, and includes an access control layer.

[0003] A blockchain system maintains one or more blockchains. A blockchain is a data structure for storing data, such as transactions, that may prevent tampering and manipulation of the data by malicious parties.

[0004] An order book is an aggregation of buy orders and sell orders, showing a number of assets being sought or offered at each price level. In a conventional digital

asset exchange infrastructure, an order book is not visible to all traders, which makes the order book dependent on for-profit matchmakers and therefore susceptible to manipulation. It is desirable to ensure that an order matching process is free of manipulation, while keeping a trader's order information (e.g., a trader's identity, a price, and a quantity) private.

SUMMARY

[0005] In one embodiment, there is provided a computer-implemented method for transaction matching, the method comprising: receiving one or more transactions; determining at least one of a match between the one or more transactions and a stored data structure or a match between one and another of the one or more transactions; sending, to a blockchain system, a request for validation of a determination result; and updating the stored data structure in response to the validation of the determination result by the blockchain system.

[0006] In another embodiment, there is provided a device for transaction matching, the device comprising: one or more processors; and one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to: receive one or more transactions; determine at least one of a match between the one or more transactions and a stored data structure or a match between one and another of the one or more transactions; send, to a blockchain system, a request for validation of a determination result; and update the stored data structure in response to the validation of the determination result by the blockchain system.

[0007] In another embodiment, there is provided a non-transitory computer-readable medium having stored therein instructions that, when executed by a processor of a device, cause the device to perform a method for transaction matching, the method comprising: receiving one or more transactions; determining at least one of a match between the one or more transactions and a stored data structure or a match between one and another of the one or more transactions; sending, to a blockchain system, a request for validation of a determination result; and updating the stored data structure in response to the validation of the determination result by the blockchain system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiments. In the following description, which refers to the drawings, the same numbers in different drawings represent the same or similar elements unless otherwise represented.

[0009] FIG. 1 is a schematic diagram of a blockchain system, according to an embodiment.

[0010] FIG. 2 is a schematic diagram of a system for order matching, according to an embodiment.

[0011] FIG. 3 is a schematic diagram of a device for order matching, according to an embodiment.

[0012] FIG. 4 is a schematic diagram illustrating an order book, according to an embodiment.

[0013] FIGS. 5A-5D are schematic diagrams illustrating limit price order matchings, according to an embodiment.

[0014] FIG. 6A-6C are schematic diagrams illustrating market price order matchings, according to an embodiment.

[0015] FIG. 7 is a schematic diagram illustrating a method for order matching, according to an embodiment.

[0016] FIG. 8 is a flow chart of a method for order matching, according to an embodiment.

[0017] FIG. 9 is a schematic diagram of an apparatus for order matching, according to an embodiment.

DETAILED DESCRIPTION

[0018] Embodiments of the specification provide methods and devices for order matching and validating an order matching result in a blockchain system. In the methods, a computer system may receive an order, determine a match between the order and orders listed in an order book of the computer system, send a request for validation of an order matching result to a blockchain system, and update the order book in response to the validation of the order matching result by the blockchain system.

[0019] Embodiments disclosed in the specification have one or more technical effects. In some embodiments, the methods and devices may verify validity of an order matching result in a blockchain system, rather than by a central entity such as the exchange that performed the order matching. This allows decentralization of the

transaction, leading to an enhanced security and transparency of the transaction. In some embodiments, in submitting a transaction such as an order, the methods and devices may use a private identification of a user, a commitment (e.g., a Pedersen commitment) of a price and a commitment of a quantity, rather than actual name or actual values of the price and the quantity. This allows preservation of privacy of transaction data, while maintaining flexibility in handling the transaction data. In some embodiments, in requesting a blockchain system for validation of an order matching result, the methods and devices may provide some proofs (e.g., a zero-knowledge range proof, a zero-knowledge proof of knowledge of the discrete logarithm) to a blockchain system, so that a node of the blockchain system can verity validity of the order matching result without revealing actual values of the price and quantity. This allows enhanced security and efficiency of the transaction, while preserving privacy of the transaction data.

[0020] A blockchain is a data structure that stores data, e.g., transactions, in a way that the transactions may be immutable and subsequently verified. A blockchain includes one or more blocks. Each block is linked to a previous block immediately before it in the blockchain by including a cryptographic hash of the previous block. Each block also may include a timestamp, its own cryptographic hash, and one or more transactions. The transactions, which generally have already been verified by nodes of the blockchain system, may be hashed and encoded into a data structure, such as a Merkle tree. In a Merkle tree, data at leaf nodes of the tree is hashed, and all hashes in each branch of the tree may be concatenated at a root of the branch. This process continues up the tree to the root of the entire tree, which stores a hash that is

representative of all data in the tree. A hash purporting to be of a transaction stored in the tree can be quickly verified by determining whether it is consistent with the structure of the tree.

[0021] A blockchain system includes a network of computing nodes that manage, update, and maintain one or more blockchains. The network may be a public blockchain network, a private blockchain network, or a consortium blockchain network. In a public blockchain network, the consensus process is controlled by nodes of the consensus network. For example, numerous entities, such as hundreds, thousands, or even millions of entities, can operate in a public blockchain network, and each of the entities operates at least one node in the public blockchain network. Accordingly, the public blockchain network can be considered a public network with respect to the participating entities. Sometimes, a majority of entities (nodes) must sign every block in order for the block to be validated and added to the blockchain of the blockchain network. Examples of public blockchain networks include particular peer-to-peer payment networks that leverage a distributed ledger, referred to as blockchain.

[0022] In general, a public blockchain network may support public transactions. A public transaction is shared with all of the nodes in the public blockchain network, and is stored in a global blockchain. A global blockchain is a blockchain replicated across all nodes, and all nodes are in consensus with respect to the global blockchain. To achieve consensus (e.g., agreement to the addition of a block to a blockchain), a consensus protocol is implemented in the public blockchain network. Examples of consensus protocols include proof-of-work (POW) (e.g., implemented in the some crypto-currency networks), proof-of-stake (POS), and proof-of-authority (POA).

[0023] In general, a private blockchain network may be provided for a particular entity, which centrally controls read and write permissions. The entity controls which nodes are able to participate in the blockchain network. Consequently, private blockchain networks are generally referred to as permissioned networks that place restrictions on who is allowed to participate in the network, and on their level of participation (e.g., only in certain transactions). Various types of access control mechanisms can be used (e.g., existing participants vote on adding new entities, a regulatory authority can control admission).

[0024] In general, a consortium blockchain network may be private among the participating entities. In a consortium blockchain network, the consensus process is controlled by an authorized set of nodes, one or more nodes being operated by a respective entity (e.g., a financial institution, insurance company). For example, a consortium of ten (10) entities (e.g., financial institutions, insurance companies) can operate a consortium blockchain network, each of which operates at least one node in the consortium blockchain network. Accordingly, the consortium blockchain network can be considered a private network with respect to the participating entities. In some examples, each entity (node) must sign every block in order for the block to be validated and added to the blockchain. In some examples, at least a sub-set of entities (nodes) (e.g., at least 7 entities) must sign every block in order for the block to be validated and added to the blockchain.

[0025] FIG. 1 illustrates a schematic diagram of a blockchain system 100, according to an embodiment. Referring to FIG. 1, the blockchain system 100 may include a plurality of nodes, e.g., nodes 102-110, configured to operate on a blockchain

120. The nodes 102-110 may form a network 112, such as a peer-to-peer (P2P) network. Each of the nodes 102-110 may be a computing device, such as a computer or a computer system, configured to store a copy of the blockchain 120, or may be software running on the computing device, such as a process or an application. Each of the nodes 102-110 may have a unique identifier. The nodes 102-110 may communicate with one another by a wired or wireless communication. Such communication may adopt a reliable protocol such as a Transmission Control Protocol/Internet Protocol (TCP/IP).

[0026] The blockchain 120 may include a growing list of records in the form of data blocks, such as blocks B1-B5 in FIG. 1. Each of the blocks B1-B5 may include a timestamp, a cryptographic hash of a previous block, and data of the present block, which may be transactions such as monetary transactions. For example, as illustrated in FIG. 1, block B5 may include a timestamp, a cryptographic hash of block B4, and transaction data of block B5. Also, for example, a hashing operation may be performed on the previous block to generate the cryptographic hash of the previous block. The hashing operation may convert inputs of various lengths into cryptographic outputs of a fixed length through a hash algorithm, such as SHA-256.

[0027] The nodes 102-110 may be configured to perform an operation on the blockchain 120. For example, when a node, e.g., the node 102, wants to store new data onto the blockchain 120, that node may generate a new block to be added to the blockchain 120 and broadcast the new block to other nodes, e.g., the nodes 104-110, in the network 112. Based on legitimacy of the new block, e.g., validity of its signature and transactions, the other nodes may determine to accept the new block, such that the

node 102 and the other nodes may add the new block to their respective copies of the blockchain 120. As this process repeats, more and more blocks of data may be added to the blockchain 120.

[0028] In an embodiment, the blockchain system 100 may operate according to one or more smart contracts. Each smart contract may be a computer protocol in the form of computer code that is incorporated into the blockchain 120, to facilitate, verify, or enforce the negotiation or performance of a contract. For example, a user of the blockchain system 100 may program agreed terms into a smart contract using a programming language, such as C++, Java, Solidity, Python, etc., and when the terms are met, the smart contract may be automatically executed by the blockchain system 100, e.g., to perform a transaction. Also, for example, the smart contract may include a plurality of subroutines or functions, each of which may be a sequence of program instructions that performs a specific task. The smart contract may be operational code that is fully or partially executed without human interaction.

[0029] FIG. 2 is a schematic diagram illustrating a system 200 for order matching, according to an embodiment. Referring to FIG. 2, the system 200 may include an exchange device 250, a user interface system 220 operated by a user 210, and a user interface system 240 operated by a user 230. The exchange device 250 may be a computer system of an exchange, which is a place where things or services are exchanged (e.g., a market or center for trading in securities or commodities), and may interact with the blockchain system 100 (Fig. 1) and the user interface systems 220 and 240. It will be understood that this is for illustrative purpose only, and the number of the user interface systems and the number of users in each system are not so limited. For

example, a plurality of user interface systems may interact with the exchange device 250, or a plurality of users may share one user interface system. In an embodiment, each user interface system may be a computer system that includes a transaction data system of one or more users. In some embodiments, each user interface system may operate as a node of the blockchain system 100. In other embodiments, each user interface system may not operate as a node of the blockchain system 100.

[0030] In some embodiments, the user 210 is a buyer and the user interface system 220 may submit a transaction such as by placing a buy order, and the user 230 is a seller and the user interface system 240 may submit another transaction such as by placing a sell order. In these embodiments, each of the user interface systems 220 and 240 may also include an accounting system that obtains and records the buy order and the sell order of the users 210 and 230, respectively.

[0031] In an embodiment, the user interface system 220 may be a third-party transaction data system that manages transaction data for multiple users, including the user 210. Similarly, the user interface system 240 may be a third-party transaction data system that manages transaction data for multiple users, including the user 230. The user 210 may create an account in the user interface system 220 using a user-chosen password. Similarly, the user 220 may create an account in the user interface system 220 using a user-chosen password.

[0032] Each of the users 210 and 230 can be any entity, for example, an individual person, a corporation, a representative of a corporation, a financial institution, a research institution or a government entity that is involved in a transaction. The user interface systems 220 and 240 may obtain, assemble, transmit and maintain buy orders

and/or sell orders for the users 210 and 230, respectively. In an embodiment, each of the users 210 and 230 may place multiple orders. The multiple orders may be multiple buy orders, multiple sell orders, or a mixture of buy orders and sell orders.

[0033] In an embodiment, the user interface system 220 may upload a buy order placed by the user 210 to the blockchain system 100. The buy order may include an identification of the user 210, a type of transaction (“buy” in this case), a quantity of a commodity that the user 210 wishes to buy, and a unit price of the commodity acceptable to the user 210. The identification of the user 210 may be a nickname or a private ID number or any other secret numbers only known to the user 210. In an embodiment, a cryptographic commitment scheme, e.g., Pedersen commitment, may be used in the order data. For example, the user interface system 220 may transmit the buy order with a data structure of {"type": buy, "price": PC (p, r), "quantity": PC (q, r')}, where PC denotes a Pedersen commitment, p denotes a price, q denotes a quantity, r is a random number used to generate the Pedersen commitment of the price, and r’ is a random number used to generate the Pedersen commitment of the quantity.

[0034] Similarly, the user interface system 240 may upload the sell order to the blockchain system 100. The sell order may include an identification of the user 230, a type of transaction (“sell” in this case), a quantity of a commodity that the user 230 wishes to sell, and a unit price of the commodity. The identification of the user 230 may be a nickname or a private ID number or any other secret numbers only known to the user 230. In an embodiment, the user interface system 240 may transmit the sell order with a data structure of {"type": sell, "price": PC (p, r), "quantity": PC (q, r')}, where PC denotes a Pedersen commitment, p denotes a price, q denotes a quantity, r is a random

number used to generate the Pedersen commitment of the price, and r' is a random number used to generate the Pedersen commitment of the quantity.

[0035] A Pedersen commitment maintains data secrecy while preserving additive property. For example, a party can commit to a price value p by sending a commitment value PC (p, r) = $g^p h^r$ (where p denotes a price, r is a random number, and g and h are generator values) that is generated based on the random number r. The actual price value p is hidden in the Pedersen commitment and one party can only open the commitment by knowing both the price value p and the random number r. Similarly, a party can commit to a quantity value q by sending a commitment value PC (q, r') = $g^q h^{r'}$ that is generated based on a random number r'. Another party can only open the commitment by knowing both the quantity value q and the random number r'.

[0036] In an embodiment, in addition to the commitment values (e.g., a Pedersen commitment of a price and a Pedersen commitment of a quantity) of the buy order, the user interface system 220 may encrypt the parameters p, q, r, and r' of the buy order with a public key of the exchange device 250 or the exchange, and upload the encrypted parameters p, q, r, and r' to the blockchain system 100. Similarly, the user interface system 240 may encrypt the parameters p, q, r, and r' of the sell order with the public key of the exchange device 250 or the exchange, and upload the encrypted parameters to the blockchain system 100, along with the commitment values of a price and a quantity.

[0037] Upon receipt of the uploaded buy order and the sell order, the blockchain system 100 may broadcast the buy and sell orders. The exchange device 250 may receive the broadcasted buy and sell orders and decrypt the p, q, r, and r' of each of the

buy and sell orders using a private key of the exchange device 250 or the exchange. The exchange device 250 may further verify whether $p > 0$, $q > 0$, and correctness of the PC (p, r) and PC (q, r') values using the decrypted p, q, r, and r' values. If it is verified that $p > 0$, $q > 0$, the received PC (p, r) equals to the calculated $g^p h^r$ value, and the received PC (q, r') equals to the calculated $g^q h^{r'}$ value, the correctness of the received order data is confirmed and the exchange device 250 may proceed with an order matching.

[0038] By using the data structure of {"type": (buy or sell), "price": PC (p, r), "quantity": PC (q, r')}, the order submission is published to the blockchain system without revealing actual values of a price and a quantity. Thus, privacy of the transaction data is ensured, while maintaining flexibility of the transaction. Also, by using a private ID of a user, privacy of the user is preserved. Further, by encrypting the parameters p, q, r, and r' using a public key of the exchange device or the exchange, only the exchange device can decrypt the parameters using the corresponding private key, ensuring security of the transaction.

[0039] The exchange can be any entity, for example, but is not limited to, a financial institution, a corporation, a government entity, a research institution, or individual personal that is authorized to match orders placed by users. The exchange device 250 may maintain an order book to keep record of the orders placed by users and an order matching history. In an embodiment, the order book may be a data structure stored in a storage. In another embodiment, the order book may be a logical structure corresponding to data stored in a storage. Each time upon receipt of a new order, the exchange device 250 may identify a match between the new order and one or

more orders listed in the order book. The exchange device 250 may also upload an order matching result along with the corresponding data to the blockchain system 100. In an embodiment, an order matching may refer to identifying a match between a new order (buy or sell) and one or more orders (sell or buy) at a price and a quantity that are acceptable to all parties in the transaction.

[0040] The blockchain system 100 may perform validation of the order matching based on the data uploaded by the exchange device 250. The validation may be performed by one or more parties that are authorized to validate the order matching performed by the exchange device 250. The one or more parties may use one or more nodes of the blockchain system 100 to verify validity of the order matching result by the comparing the order matching result with the corresponding data provided by the exchange device 250. If the blockchain system 100 determines the order matching performed by the exchange device 250 is valid, the blockchain system 100 may broadcast a validation result.

[0041] Upon receipt of the broadcasted validation result, the exchange device 250 may update the order book to reflect changes caused by the order matching. In some embodiments, the exchange device 250 may operate as a node of the blockchain system 100. In other embodiments, the exchange device 200 may not operate as a node of the blockchain system 100.

[0042] By verifying validity of an order matching result in a blockchain system, rather than by a central entity such as the exchange that performed the order matching, decentralization of the transaction is achieved, leading to an enhanced security, transparency and fairness of the transaction.

[0043] In an embodiment, each of the users 210 and 230 may create an account in the user interface system 220 and 240, respectively, and the user interface systems may interface with the blockchain system 100 or the exchange device 250 for placement of the orders. This allows the users 210 and 230 to conveniently manage their accounts with only passwords, without maintaining cryptographic keys for the transaction data.

[0044] In an embodiment, the exchange device 250 may include instructions stored in a computer system. The computer system may execute the instructions to perform the functions of the exchange device 250. The user interface systems 220 and 240 may be or may not be implemented in that computer system.

[0045] In an embodiment, the exchange device 250 may be independent hardware that includes integrated circuits and storage devices that may be compatible with any user interface systems. For example, the exchange device 250 can interface with any other user interface systems.

[0046] In an embodiment, the user interface systems 220 and 240 may directly send the encrypted orders to the exchange device 250 without uploading the orders to a blockchain system. To ensure security in transmission, the orders may be encrypted with a public key of the exchange device 250 or the exchange before sending, so that the exchange device 250 can decrypt the orders using the corresponding private key.

[0047] FIG. 3 is a schematic diagram of a device 300 for order matching, according to an embodiment. For example, the device 300 may be the exchange device 250 (FIG. 2). The device 300 may take any forms, including but not limited to, a desktop computer, a laptop computer, a server computer, a tablet, a smartphone, or a

smart watch, or any other forms. The device 300 may include a processor 310, a memory 320, a user interface 330, and a communication interface 340 that communicate with one another through a bus 350.

[0048] The processor 310 may include one or more dedicated processing units, application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), or various other types of processors or processing units. The processor 310 is coupled with the memory 320 and may execute instructions stored in the memory 320.

[0049] The communication interface 340 may facilitate communications between the device 300 and a user interface system and a blockchain system, such as the user interface system 220 and the blockchain system 100 (FIG. 2). The communication interface 340 may receive a sell order and/or a buy order from the blockchain system or from the user interface system. The communication interface 340 may also upload an order matching result and relevant data to the blockchain system or another blockchain system, and receive a validation result from the blockchain system or another blockchain system.

[0050] In an embodiment, the communication interface 340 may support one or more communication standards, such as an Internet standard or protocol including the TCP/IP and TLS/SSL protocols, an Integrated Services Digital Network (ISDN) standard, etc. In an embodiment, the communication interface 340 may include one or more of a Local Area Network (LAN) card, a cable modem, a satellite modem, a data bus, a cable, a wireless communication channel, a radio-based communication channel, a cellular communication channel, an Internet Protocol (IP) based communication

device, or other communication devices for wired and/or wireless communications. In an embodiment, the communication interface 340 may be based on public cloud infrastructure, private cloud infrastructure, and hybrid public/private cloud infrastructure.

[0051] The memory 320 may include a storage that stores an order book 322, member account data 324 and key manager 326. The order book 322 includes records of buy orders and sell orders and results of order matchings. After each order matching, the order book may be updated upon validation by a blockchain system. The member account 324 may include information of the users placed the buy orders and the sell order, for example, the identification of the users, their order placement history and results of order matchings. The key manager 326 may maintain private and public key of the device 300.

[0052] In an embodiment, the device 300 may be shared by multiple exchanges and the member account data 324 may include information of the multiple exchanges. Each of the multiple exchanges may have their own member account data and order book(s) in the device 300.

[0053] The memory 320 may also store processor-executable instructions and data. The processor-executable instructions and data may include an order matching manager 328. The order matching manager 328, when executed by the processor 310, allows the device 300 to extract and analyze parameters included in orders received from user interface systems or a blockchain system, and perform an order matching based on the analysis.

[0054] The memory 320 may be any type of volatile or non-volatile memory devices, or a combination thereof, such as a static random-access memory (SRAM), an

electrically erasable programmable read-only memory (EEPROM), an erasable programmable read-only memory (EPROM), a programmable read-only memory (PROM), a read-only memory (ROM), a magnetic memory, a flash memory, or a magnetic or optical disk.

[0055] The user interface 330 may include a display and an input device to transmit user command to the processor 310, etc. The display may display transaction data, an order matching process, and a validation result obtained from a blockchain system, etc. The display may include, but is not limited to, cathode ray tube (CRT), liquid crystal display (LCD), light-emitting diode (LED), gas plasma, a touch screen, or other image projection devices for displaying information to a user. The input device may be any type of computer hardware equipment used to provide data and control signals from a user to the processor 310. The input device may include, but is not limited to, a keyboard, a mouse, a scanner, a digital camera, a joystick, a trackball, cursor direction keys, a touchscreen monitor, or audio/video commanders, etc.

[0056] In an embodiment, the device 300 may include a user interface system that places a buy order or a sell order for a user, such as the user interface systems 220 and 240 of FIG. 2, so that the processor 310 and the user interface 330 are part of the user interface system.

[0057] FIG. 4 is a schematic diagram illustrating an order book 400, according to an embodiment. Referring to FIG. 4, the order book 400 may be a collection of buy orders and sell orders that have been submitted to an exchange. Each order in the order book 400 is to be matched with another order so that a transfer of assets can occur. In an embodiment, the order book 400 may store two separately ordered lists

corresponding to buy orders and sell orders, respectively. The left most column of the order book 400 lists identification numbers (IDs) of the orders. An ID may be expressed as a natural number and it may be assigned by the exchange device 250 to each order according to the receiving time of the order. For example, order #1 is received earlier than order #4 or at the same time as order #4, but not later than order #4. The IDs may be ascending or descending according to the time of the order placement. The buy orders may be listed in a sequence of side, time, quantity, and price, starting from the second left most column to the right of the order book 400. A side indicates whether an order is a buy order or a sell order. A time indicates a time at which the order is received. A quantity of a buy order may be listed in two ways: (1) a Pedersen commitment PC (q, r') of the quantity; and (2) an actual value of the quantity. Similarly, a price of a buy order may be listed in two ways: (1) a Pedersen commitment PC (p, r) of the price; and (2) an actual value of the price. For example, the order book 400 shows two buy orders, i.e., an order #4 and an order #3. The order #4 is received at 09:08 with a quantity of a commodity of 200 at a unit price of the commodity of $20.20. In addition to the actual numbers of the quantity and the price, the order book 400 also includes a Pedersen commitment PC (q_4, r'_4) of the quantity and a Pedersen commitment PC (p_4, r_4) of the price. The order #3 is received at 09:06 with a quantity of a commodity of 100 at a unit price of the commodity of $20.15. In addition to the actual numbers of the quantity and the price, the order book 400 also includes a Pedersen commitment PC (q_3, r'_3) of the quantity and a Pedersen commitment PC (p_3, r_3) of the price.

[0058] Similarly, the sell orders may be listed in a sequence of side, time, quantity, and price, starting from right most column to the left of the order book 400. A quantity of a sell order may be listed in two ways: (1) a Pedersen commitment PC (q, r') of the quantity; and (2) an actual value of the quantity. Similarly, a price of a sell order may be listed in two ways: (1) a Pedersen commitment PC (p, r) of the price; and (2) an actual value of the price. For example, the order book 400 shows two sell orders, i.e., an order #1 and an order #2. The order #1 is received at 09:01 with a quantity of a commodity of 100 at a unit price of the commodity of $20.30. In addition to the actual numbers of the quantity and the price, the order book 400 also includes a Pedersen commitment PC (q_1, r'_1) of the quantity and a Pedersen commitment PC (p_1, r_1) of the price. The order #2 is received at 09:03 with a quantity of a commodity of 100 at a unit price of the commodity of $20.25. In addition to the actual numbers of the quantity and the price, the order book 400 also includes a Pedersen commitment PC (q_2, r'_2) of the quantity and a Pedersen commitment PC (p_2, r_2) of the price. The actual numbers of the quantities and the prices of the orders in the order book 400 may be confidential to a public blockchain system. As a mere example, the order book 400 only shows two buy orders and two sell orders. However, the number of the buy orders and the number of the sell orders are not so limited, and they can be any numbers.

[0059] Referring back to FIG. 2, after receiving a new order, the exchange device 250 may perform an order matching by comparing the new order against one or more orders currently listed in the order book 400. The orders currently listed in the order book 400 may be the orders having not been matched in previous order matchings. For example, for a new buy order, the exchange device 250 may compare a price and a

quantity of the new buy order against a price and a quantity of a sell order currently listed in the order book 400 to see whether the price and quantity match. Similarly, for a new sell order, the exchange device 250 may compare a price and a quantity of the new sell order against a price and a quantity of a buy order currently listed in the order book 400. After performing an order matching, the exchange device 250 may upload an order matching result and the corresponding data to a blockchain system such that the blockchain system may verify validity of the order matching result.

[0060] In an embodiment, the exchange device 250 may provide the blockchain system 100 with a range of a price using a zero-knowledge range proof (e.g., as illustrated in FIGS. 5A-5D below), without revealing the actual value of the price. The range proof may be generated through various schemes, including, but not limited to, Bulletproof scheme and Borromean Ring signatures scheme. For example, to prove a price of one order is larger than a price of another order, the following needs to be proved: $\{w= (p_1, r_1, p_2, r_2), x= (P_1, P_2), R: P_1 = g^{p1}h^{r1} \ \& \ P_2 = g^{p2}h^{r2} \ \& \ p_1 > p_2\}$. This can be translated into an equivalent formulation: $\{w = (\Delta p = p_1 - p_2, \Delta r = r_1 - r_2), x= (\Delta P = P_1 / P_2), R: \Delta P = g^{\Delta p}h^{\Delta r} \ \& \ \Delta p>0\}$, which corresponds to zero-knowledge range proof. Similarly, the exchange device 250 may also provide a range of a quantity using a zero-knowledge range proof (e.g., as illustrated in FIGS. 5D and 6C below), without revealing the actual value of the quantity.

[0061] In an embodiment, the exchange device 250 may use a zero-knowledge proof of knowledge of the discrete logarithm to prove that a quantity of one order is the same as a quantity of another order (e.g., as illustrated in FIG. 6A

below). This is because, to prove that a quantity of one order is the same as a quantity of another order, the following needs to be proved: $\{w = (q, r_1, r_2), x = (P_1, P_2), R: P_1 = g^q h^{r1} \ \& \ P_2 = g^q h^{r2}\}$. This can be translated into an equivalent formulation: $\{w = \Delta r = r_1 - r_2, x = \Delta P = P_1 / P_2, R: \Delta P = h^{\Delta r}\}$, which corresponds to a zero-knowledge proof of knowledge of the discrete logarithm.

[0062] In an embodiment, the exchange device 250 may use a zero-knowledge proof of knowledge of the discrete logarithm to prove a linear relationship between multiple quantities (e.g., as illustrated in FIGS. 5C and 6B below). This is because, to prove a linear relationship of multiple quantities, the following needs to be proved: $\{w = (q_1, r_1, q_2, r_2, q_3, r_3), x = (P_1, P_2, P_3), R: P_1 = g^{q1} h^{r1} \ \& \ P_2 = g^{q2} h^{r2} \ \& \ P_3 = g^{q3} h^{r3}, q_3 = q_1 + q_2\}$. This can be translated into: $\{w = \Delta r = r_1 + r_2 - r_3, x = \Delta P = P_1 * P_2 / P_3, R: h^{\Delta r} = \Delta P\}$, which corresponds to a zero-knowledge proof of knowledge of the discrete logarithm.

[0063] In an embodiment, a user may indicate a desired price (or a range of a price) in an order and an order matching may be performed at the price (or the range of the price) indicated by the user, and this may be referred to as a limit price order matching. A user may indicate a desired price in a buy order or a sell order. Alternatively, a user may not indicate a desired price in a buy order or a sell order and an order matching may be performed at a market price, and this may be referred to as a market price order matching. In an embodiment, for a buy order, the market price may be a lowest price listed in sell order(s) in the order book at the time of the order matching. In an embodiment, for a sell order, the market price may be a highest price listed in buy order(s) in the order book at the time of the order matching.

[0064] FIGS. 5A-5D are schematic diagrams illustrating limit price order matchings, according to an embodiment. Referring to FIG. 5A, in an embodiment, the exchange device 250 having the order book 400 receives a new buy order (ID #5) at 09:10. The new order includes a unit price of $20.19 and a quantity of 100. In addition to the actual numbers of the quantity and the price, the new buy order also includes a Pedersen commitment PC (q_5, r'_5) of the quantity and a Pedersen commitment PC (p_5, r_5) of the price. For the purpose of the order matching, this new order may be temporarily inserted between the buy order #4 and the buy order #3, according to descending order of price and indicated as order ID #5 in an updated order book 510. In the embodiment, the price of the new buy order cannot be matched with any sell orders in the order book, as prices in both sell order #1 and the sell order #2 are higher than $20.19. The result of the order matching is “no matching”. The exchange device 250 may request the blockchain system 100 to validate the order matching result (“no matching”) by uploading zero-knowledge range proofs: (1) $p_5 - p_3 > 0$; and (2) $p_4 - p_5 \geq 0$. Once the blockchain system 100 validates the order matching result, the exchange device 250 may update the order book 400 by inserting the buy order #5 into the order book 400, as shown in FIG. 5A.

[0065] Referring to FIG. 5B, in another embodiment, the exchange device 250 having the order book 400 receives a new buy order in which the buyer wishes to buy a commodity at a unit price of $20.25 with a quantity of 100. Compared with the sell orders in the order book 400 (FIG. 4), this new order can be fully matched with the sell order #2 in the order book 400. The result of the order matching is “full matching with a single sell order”. The exchange device 250 may request the blockchain system 100 to

validate this order matching result and upload a zero-knowledge range proof: $p_5 - p_2 \geq 0$; and a zero-knowledge proof of knowledge of the discrete logarithm: $q_5 = q_2$. Once the blockchain system 100 validates this order matching result, the exchange device 250 may update the order book 400 by removing the sell order #2 from the order book 400, as shown in an updated order book 520 in FIG. 5B. In an embodiment, after validation of the order matching result, the blockchain system may further perform settlement so the assets can be transferred.

[0066] Referring to FIG. 5C, in another embodiment, the exchange device 250 having the order book 400 receives a new buy order in which the buyer wishes to buy a commodity at a unit price of $20.30 with a quantity of 200. Compared with the sell orders of the order book 400 (FIG. 4), this new buy order can be fully matched with the combination of the sell order #1 and the sell order #2 in the order book 400. The result of the order matching is “full matching with multiple sell orders”. The exchange device 250 may request the blockchain system 100 to validate this order matching result and upload a zero-knowledge range proof: $p_5 - p_1 \geq 0$ (this also indicates $p_5 - p_2 \geq 0$); and a zero-knowledge proof of knowledge of the discrete logarithm: $q_5 = q_2 + q_1$. Once the blockchain system 100 validates this order matching result, the exchange device 250 may update the order book 400 by removing both the sell order #1 and the sell order #2 from the order book 400, as shown in an updated order book 530 in FIG. 5C. In an embodiment, after validation of the order matching, the blockchain system may further perform settlement so the assets can be transferred.

[0067] Referring to FIG. 5D, in another embodiment, the exchange device 250 having the order book 400 receives a new buy order in which the buyer wishes to buy a

commodity at a unit price of $20.26 with a quantity of 50. Compared with the sell orders in the order book 400 (FIG. 4), this new buy order can be partially matched with the sell order #2 in the order book 400. The result of the order matching is "partial matching with a single sell order". The exchange device 250 may request the blockchain system 100 to validate this order matching result and upload zero-knowledge range proofs: $p_5 - p_2 \geq 0$; and $q_2 - q_5 > 0$. Once the blockchain system 100 validates this order matching result, the exchange device 250 may update the order book 400 by changing the quantity of sell order #2 from 100 to 50 in the order book 400, as shown in an updated order book 540 in FIG. 5D.

[0068] For the sake of simplicity, FIGS. 5A-5D only show examples of limit price order matchings in which only a buy order is newly received. However, a similar matching method can be applied to a case in which a new sell order, or multiple buy orders or multiple sell orders are newly received by the exchange device 250. In an embodiment, the exchange device 250 may receive a mixture of one or more buy orders and one or more sell orders. The exchange device 250 may process the one or more new buy orders and the one or more new sell orders based on price-time priority of the new orders. In an embodiment, the exchange device 250 may process the new orders one by one based on their arrival times. In another embodiment, the new orders may arrive at the same time. In this case, the exchange device 250 may compare a new order against any other new orders and also compare the new order against the orders listed in the order book 400. For example, if the exchange device 250 receives a new buy order and a new sell order at the same time, the exchange device 250 may

compare the new buy order against the new sell order and also compare the new buy order against the sell order(s) listed in the order book 400.

[0069] In the above descriptions of FIGS. 5A-5D, in requesting for validation of an order matching result from the blockchain system 100, the exchange device 250 may upload to the blockchain system 100 at least one of: (1) a range proof of a price; (2) a range proof of a quantity; (3) an equal relationship between two quantities; or (4) a linear relationship between a plurality of quantities. In this way, one or more nodes of the blockchain system 100 may verify validity of the order matching result without knowing or revealing actual values of the price and the quantity, leading to an enhanced security and fairness of the transaction.

[0070] FIGS. 6A-6C are schematic diagrams illustrating market price order matchings, according to an embodiment. A market price order matching may be performed based on a best market price at the time of the order matching. For a new buy order, the best market price may be a lowest price among the sell orders listed in the order book 400 at the time of the order matching. For a new sell order, the best market price may be a highest price among the buy orders listed in the order book 400 at the time of the order matching. In an embodiment, for a new buy order, if a quantity of a sell order having a lowest price is less than a quantity of the new buy order, a sell order with a second lowest price may be considered. Since market price order matchings are performed based on a price listed in the order book 400, there is no need to upload zero knowledge range proofs on price information to the blockchain system 100 for validation of the order matching. The exchange device 250 may only upload zero knowledge range proofs for quantity information to the blockchain system 100.

[0071] Referring to FIG. 6A, in an embodiment, the exchange device 250 having the order book 400 receives a new buy order in which the buyer wishes to buy a commodity at a market price with a quantity of 100. Compared with the sell orders in the order book 400 (FIG. 4), this new buy order can be fully matched with the sell order #2 in the order book 400. The result of the order matching is “full matching with a single sell order”. The exchange device 250 may request the blockchain system 100 to validate this order matching result and upload a zero-knowledge proof of knowledge of the discrete logarithm: $q_5 = q_2$. Once the blockchain system 100 validates this order matching, the exchange device 250 may update the order book 400 by removing the sell order #2 from the order book 400, as shown in an updated order book 610 in FIG. 6A. In an embodiment, after validation of the order matching, the blockchain system may further perform settlement so the assets can be transferred.

[0072] Referring to FIG. 6B, in another embodiment, the exchange device 250 having the order book 400 receives a new buy order in which the buyer wishes to buy a commodity at a market price with a quantity of 200. Compared with the sell orders in the order book 400 (FIG. 4), this new buy order can be fully matched with the combination of the sell order #2 and the sell order #1 in the order book 400. The result of the order matching is “full matching with multiple sell orders”. The exchange device 250 may request the blockchain system 100 to validate this order matching result and upload a zero-knowledge proof of knowledge of the discrete logarithm: $q_5 = q_2 + q_1$. Once the blockchain system 100 validates this order matching, the exchange device 250 may update the order book 400 by removing both the sell order #2 and the sell order #1 from the order book 400, as shown in an updated order book 620 in FIG. 6B.

In an embodiment, after validation of the order matching, the blockchain system may further perform settlement so the assets can be transferred.

[0073] Referring to FIG. 6C, in another embodiment, the exchange device 250 having the order book 400 receives a new buy order in which the buyer wishes to buy a commodity at a market price with a quantity of 50. Compared with the sell orders in the order book 400 (FIG. 4), this new buy order can be partially matched with the sell order #2 in the order book 400. The result of the order matching is "partial matching with a single sell order". The exchange device 250 may request the blockchain system 100 to validate this order matching result and upload a zero-knowledge range proof: $q_2 - q_5 > 0$. Once the blockchain system 100 validates this order matching, the exchange device 250 may update the order book 400 by changing the quantity of the sell order #2 from 100 to 50 in the order book 400, as shown in an updated order book 630 in FIG. 6C. In an embodiment, after validation of the order matching, the blockchain system may further perform settlement so the assets can be transferred.

[0074] For the sake of simplicity, FIGS. 6A-6C only show examples of market price order matchings in which only a buy order is newly received. However, a similar matching method can be applied to a case in which a new sell order, or multiple buy orders or multiple sell orders are received by the exchange device 250. In an embodiment, the exchange device 250 may receive a mixture of one or more buy orders and one or more sell orders. The exchange device 250 may process the one or more new buy orders and the one or more new sell orders based on price-time priority of the new orders. In an embodiment, the exchange device 250 may process the new orders one by one based on their arrival times. In another embodiment, the new orders

may arrive at the same time. In this case, the exchange device 250 may compare a new order against any other new orders and the orders in the order book 400. For example, if the exchange device 250 receives a new buy order and a new sell order at the same time, the exchange device 250 may compare the new buy order against the new sell order and the sell order(s) listed in the order book 400.

[0075] In the above descriptions of FIGS. 6A-6C, in requesting validation of an order matching result from the blockchain system 100, the exchange device 250 may upload to the blockchain system 100 at least one of: (1) a range proof of a quantity; (2) an equal relationship between two quantities; or (3) a linear relationship between a plurality of quantities. In this way, one or more nodes of the blockchain system 100 may verify validity of an order matching result without knowing or revealing actual values of the price and the quantity, leading to an enhanced security and fairness of the transaction.

[0076] In an embodiment, along with a request for validation of an order matching result, the exchange device may upload commitments of a price and a quantity of an order (e.g., Pedersen commitment) to the blockchain system. In an embodiment, a user that placed the order may identify the order from the commitments and verify validity of the order matching result on a client of the blockchain system.

[0077] In an embodiment, an order matching result and transaction data used in the order matching may be investigated by a supervision division. The supervision division may request the exchange device to provide parameters (p, q, r, r') used in the order matching. The supervision division may determine whether a $g^p h^r$ value calculated using the provided parameters equals to the PC (p, r) value uploaded to the

blockchain system and determine whether a $g^q h^{r'}$ value calculated using the provided parameters equals to the PC (q, r') value uploaded to the blockchain system to ensure correctness of the original transaction data. The supervision division may further verify validity of the order matching using the methods described in FIGS. 5A-5D and FIGS. 6A-6C.

[0078] FIG. 7 is a schematic diagram illustrating a method 700 for order matching, according to an embodiment. A user 210 (e.g., a buyer) may create an account in the blockchain system 100 and upload a buy order to the blockchain system 100 (step 702). For example, the user 210 may upload the buy order with a data structure of {"type": buy, "price": PC (p, r), "quantity": PC (q, r')}, where PC is a Pedersen commitment. In an embodiment, the user 210 may encrypt the p, q, r, r' of the buy order using a public key of the exchange device 250 or the exchange, and attach it as a payload to an order submission transaction and upload it to the blockchain system (step 702). Alternatively, in another embodiment (not shown), the user 210 may directly transmit the buy order to the exchange device 250 without uploading it to the blockchain system 100.

[0079] Similarly, a user 230 (e.g., a seller) may create an account in the blockchain system 100 and upload a sell order to the blockchain system (step 704). For example, the user 230 may upload the sell order with a data structure of {"type": sell, "price": PC (p, r), "quantity": PC (q, r')}, where PC is a Pedersen commitment. In an embodiment, the user 230 may encrypt the p, q, r, an r' of the sell order using a public key of the exchange device 250 or the exchange, and attach it as a payload to an order submission transaction and upload it to the blockchain system (step 704). Alternatively,

in another embodiment (not shown), the user 230 may directly transmit the sell order to the exchange device 250.

[0080] The blockchain system 100 may broadcast the uploaded orders (step 706). The exchange device 250 may receive the broadcasted orders, decrypt the encrypted p, q, r, and r' using a private key, and confirm correctness of the order data (step 708). The exchange device 250 may further perform an order matching (step 710). For example, the exchange device 250 may perform a limit price order matching (as shown in FIGS. 5A-5D) or a market price order matching (as shown in FIGS. 6A-6C), depending on whether a user indicated a desired price or not. To request validation of a limit price order matching, the exchange device 250 may upload at least one of: a range proof of the price, a range proof of the quantity, an equal relationship between two quantities, or a linear relationship between a plurality of quantities, to the blockchain system 100. For example, the exchange device 250 uploads a zero-knowledge range proof of the price p and a zero-knowledge range proof of the quantity q (step 712). To request validation of a market price matching, the exchange device 250 may upload at least one of: a range proof of the quantity, an equal relationship between two quantities, or a linear relationship between a plurality of quantities, to the blockchain system 100. For example, the exchange device 250 uploads a zero-knowledge range proof of the quantity q (step 714). Upon receipt of the uploaded information from the exchange device 250, the blockchain system 100 may perform validation of the order matching by comparing the result of the order matching and the uploaded data (step 716). Once the order matching is validated, the blockchain system 100 may broadcast the validation result (step 718) and the exchange device 250 may

receive the broadcasted validation result (step 720). The exchange device 250 may further update an order book used in the order matching (step 722) to reflect changes occurred due to the order matching. The user 210 may also receive the broadcasted validation results so that the user has knowledge of the status of the buy order placement (step 724). Similarly, the user 230 may also receive the broadcasted validation results so that the user has knowledge of the status of the sell order placement (step 726).

[0081] In an embodiment, the blockchain system to which the users upload the orders may be or may not be same as the blockchain system that validates an order matching performed by the exchange device 250.

[0082] FIG. 8 is a flow chart of a method 800 for order matching, according to an embodiment. The method 800 may be performed by an exchange device in an exchange, such as the exchange device 250 (FIG. 2). Referring to FIG. 8, the exchange device may receive buy orders and sell orders (step 802). In an embodiment, the exchange device may receive the buy orders and the sell orders directly from the buyers and sellers. In another embodiment, the exchange device may receive the buy orders and the sell orders from a broadcast from a blockchain system, such as the blockchain system 100 (FIG. 1).

[0083] Upon receipt of the buy orders and the sell orders, the exchange device may perform order matching (step 804). In an embodiment, the exchange device may perform the order matching based on a price and a quantity indicated in a buy order or a sell order, and this may be referred to as a limit price order matching (FIGS. 5A-5D). In another embodiment, a buy order or a sell order may not indicate a desired price, and

the exchange device may perform an order matching based on a best market price at the time of the order matching, and this may be referred to as a market price order matching (FIGS. 6A-6C).

[0084] After the order matching, the exchange device may upload results and proofs to a blockchain system, such as the blockchain system 100 (FIG. 1), to request the blockchain system to validate the order matching result. The exchange device may upload some proofs (e.g., zero-knowledge range proof, zero-knowledge proof of knowledge of the discrete logarithm) so that a node of the blockchain system can verify validity of the order matching without knowing actual values of the price and quantity (step 806). For example, as to the limit price order matching, the exchange device may provide at least one of: a range proof of a price, a range proof of a quantity, an equal relationship between two quantities, or a linear relationship between a plurality of quantities. As to the market price order matching, the exchange device may only provide at least one of: a range proof of a quantity, an equal relationship between two quantities or a linear relationship between a plurality of quantities.

[0085] The order matching result may be validated by one or more nodes in the blockchain system, and the exchange device may receive the validation result directly from the blockchain system or through a broadcast by the blockchain system (step 808). If the order matching result is validated by the blockchain system, the exchange device may further update an order book used in the order matching to reflect the changes that occurred due to the order matching (step 810).

[0086] FIG. 9 is a schematic diagram of an apparatus 900 for order matching, according to an embodiment. The apparatus 900 may be an implementation of a

software process, and may correspond to the method 800 (FIG. 8). As shown in FIG. 9, the apparatus 900 may include a receiving module 810, a matching module 920, a requesting module 930, and an updating module 940.

[0087] The receiving module 910 receives buy orders and sell orders from buyers and sellers. The matching module 920 analyzes the buy orders and the sell orders and performs order matching. The requesting module 930 requests a blockchain system for validation of the order matching results and upload relevant data to the blockchain system. The updating module 940 updates an order book used in the order matching to reflect the changes due to the order matching once the apparatus 900 receives validation results from the blockchain system.

[0088] Each of the above described modules may be implemented as software, or hardware, or a combination of software and hardware. For example, each of the above described modules may be implemented using a processor executing instructions stored in a memory. Also, for example, each the above described modules may be implemented with one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), controllers, micro-controllers, microprocessors, or other electronic components, for performing the described methods. Further for example, each of the above described modules may be implemented by using a computer chip or an entity, or implemented by using a product having a certain function. In one embodiment, the apparatus 900 may be a computer, and the computer may be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media

player, a navigation device, an email receiving and sending device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[0089] For an implementation process of functions and roles of each module in the apparatus 900, references can be made to corresponding steps in the above-described methods. Details are omitted here for simplicity.

[0090] In an embodiment, there is also provided a computer program product. The computer program product may include a non-transitory computer-readable storage medium having computer-readable program instructions thereon for causing a processor to carry out the above-described methods.

[0091] The computer-readable storage medium may be a tangible device that can store instructions for use by an instruction execution device. The computer-readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer-readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing.

[0092] The computer-readable program instructions for carrying out the above-

described methods may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine-dependent instructions, microcode, firmware instructions, state-setting data, or source code or object code written in any combination of one or more programming languages, including an object-oriented programming language, and conventional procedural programming languages. The computer-readable program instructions may execute entirely on a computing device as a stand-alone software package, or partly on a first computing device and partly on a second computing device remote from the first computing device. In the latter scenario, the second, remote computing device may be connected to the first computing device through any type of network, including a local area network (LAN) or a wide area network (WAN).

[0093] The computer-readable program instructions may be provided to a processor of a general-purpose or special-purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the above-described methods.

[0094] The flow charts and diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of devices, methods, and computer program products according to various embodiments of the specification. In this regard, a block in the flow charts or diagrams may represent a software program, segment, or portion of code, which comprises one or more executable instructions for implementing specific functions. It should also be noted that, in some alternative implementations, the functions noted in the blocks may occur out of the order noted in

the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the diagrams and/or flow charts, and combinations of blocks in the diagrams and flow charts, may be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0095] It is appreciated that certain features of the specification, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the specification, which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable subcombination or as suitable in any other described embodiment of the specification. Certain features described in the context of various embodiments are not essential features of those embodiments, unless noted as such.

[0096] Although the specification has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the following claims embrace all such alternatives, modifications and variations that fall within the terms of the claims.

Claims

1. A computer-implemented method for transaction matching, comprising:

receiving one or more transactions;

determining at least one of a match between the one or more transactions and a stored data structure or a match between one and another of the one or more transactions;

sending, to a blockchain system, a request for validation of a determination result; and

updating the stored data structure in response to the validation of the determination result by the blockchain system,

wherein sending, to the blockchain system, the request for validation of the determination result further comprises:

uploading, to the blockchain system, a range proof of a price of matched transactions and at least one of:

a range proof of a quantity of the matched transactions;

an equal relationship between two quantities of the matched transactions; or

a linear relationship between a plurality of quantities of the matched transactions.

2. The method of claim 1, wherein the one or more transactions are received from a broadcast by the blockchain system or a broadcast by another blockchain system different from the blockchain system.

3. The method of claim 1, wherein the one or more transactions are received from one or more user interface devices.

4. The method of any preceding claim, wherein the one or more transactions comprise at least one of:

one or more buy orders;

one or more sell orders; or

one or more buy orders and one or more sell orders.

5. The method of any preceding claim, wherein each of the received one or more transactions includes a transaction type and at least one of: a commitment of a price, a commitment of a quantity, a price encrypted by a cryptographic key, a quantity encrypted by the cryptographic key, a random number that is used to generate the commitment of the price and encrypted by the cryptographic key, or a random number that is used to generate the commitment of the quantity and encrypted by the cryptographic key.

6. The method of claim 5, wherein the commitment of the price is a Pedersen commitment of the price and the commitment of the quantity is a Pedersen commitment of the quantity.

7. The method of any preceding claim, wherein determining at least one of the match between the one or more transactions and the stored data structure or the match between one and another of the one or more transactions is performed based on at least one price included in the one or more transactions.

8. The method of any of claims 1 to 6, wherein determining at least one of the match between the one or more transactions and the stored data structure or the match between one and another of the one or more transactions is performed based on:

one or more lowest prices of one or more sell orders included in the stored data structure; or

one or more highest prices of one or more buy orders included in the stored data structure.

9. The method of any preceding claim, wherein:

each of the range proof of the price of the matched transactions and the range proof of the quantity of the matched transactions is based on a zero-knowledge range proof, and

each of the equal relationship between two quantities of the matched transactions and the linear relationship between a plurality of quantities of the matched transactions is based on a zero-knowledge proof of knowledge of the discrete logarithm.

10. The method of any preceding claim, further comprising:

verifying, on one or more nodes of the blockchain system, validity of the determination result based on the request for validation of the determination result.

11. A device for matching one or more transactions, comprising:

one or more processors; and

one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to perform the method of any of claims 1 to 10.

12. An apparatus for matching one or more transactions, the apparatus comprising a plurality of modules for performing the method of any of claims 1 to 10.

13. A non-transitory computer-readable medium having stored therein instructions that, when executed by a processor of a device, cause the device to perform the method of any of claims 1 to 10.