

METHODS AND DEVICES FOR PROVIDING PRIVACY-PRESERVING BLOCKCHAIN-BASED AUCTION

TECHNICAL FIELD

[0001] The specification relates generally to computer technologies, and more particularly, to methods and devices for providing a privacy-preserving blockchain-based auction.

BACKGROUND

[0002] A Dutch auction may refer to a public offering auction in which the price of the offering is set after taking in all bids to determine the highest price at which the offering can be sold. Dutch auctions may be used for other types of offerings, including, e.g., offerings of treasury bills, notes, bonds, properties, and the like.

[0003] When a Dutch auction is used to conduct an initial public offering (IPO) of shares of a company, for example, the company that is providing the offering does not set a fixed price for its shares. Instead, the company decides the number of shares to be offered and let the bidders (investors) determine the price. Investors may submit bids indicating the number of shares they would like to purchase and the specific bid prices. All bids received during the auction period may be collected and processed to determine the highest price at which all the shares offered by the company can be sold. Once the price is determined, the company may allow the investors to purchase the offered shares at the determined price. All investors may pay the same price for the offered shares.

[0004] Currently, IPOs conducted using Dutch auctions are handled by a trusted third party. The trusted third party must be trusted by the company offering

the IPO and the investors bidding on the IPO. The trusted third party is also responsible for ensuring that each auction is conducted in a secure and fair manner.

SUMMARY

[0005] In one aspect, a computer-implemented method for providing a privacy-preserving blockchain-based auction includes: receiving, from a first user, data representing one or more assets offered to be auctioned off and a set of parameters for the auction; receiving one or more encrypted bids from one or more second users; receiving a plurality of shared pieces of a private key from a plurality of third users; reconstructing the private key based on the plurality of shared pieces received from the plurality of third users, the reconstructed private key being configured to decrypt the received encrypted bids; and automatically settling the auction based on the parameters for the auction, the received encrypted bids, and submission times at which the encrypted bids are received.

[0006] In another aspect, a device for providing a privacy-preserving blockchain-based auction includes: 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, from a first user, data representing one or more assets offered to be auctioned off and a set of parameters for the auction; receive one or more encrypted bids from one or more second users; receive a plurality of shared pieces of a private key from a plurality of third users; reconstruct the private key based on the plurality of shared pieces received from the plurality of third users, the reconstructed private key being configured to decrypt the received encrypted bids; and automatically settle the auction based on the parameters for the auction, the received encrypted bids,

and submission times at which the encrypted bids are received.

[0007] In still another aspect, a non-transitory computer-readable medium has stored therein instructions that, when executed by a processor of a device, cause the device to perform a method for providing a privacy-preserving blockchain-based auction. The method includes: receiving, from a first user, data representing one or more assets offered to be auctioned off and a set of parameters for the auction; receiving one or more encrypted bids from one or more second users; receiving a plurality of shared pieces of a private key from a plurality of third users; reconstructing the private key based on the plurality of shared pieces received from the plurality of third users, the reconstructed private key being configured to decrypt the received encrypted bids; and automatically settling the auction based on the parameters for the auction, the received encrypted bids, and submission times at which the encrypted bids are received.

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 computing device for implementing a node in a blockchain system, according to an embodiment.

[0011] FIG. 3 is a flow chart of a method for providing a privacy-preserving blockchain-based auction, according an embodiment.

[0012] FIG. 4 is a flow chart of a method for providing a privacy-preserving blockchain-based auction, according an embodiment.

[0013] FIG. 5 is a block diagram of an apparatus for providing a privacy-preserving blockchain-based auction, according to an embodiment.

DETAILED DESCRIPTION

[0014] Embodiments of the specification provide methods and devices for providing a privacy-preserving blockchain-based auction. The methods and devices may allow a first user of a blockchain system, referred to as an Offeror, to generate a pair of public and private keys to facilitate a bidding process. The methods and devices may allow the Offeror to split the private key into N pieces and share the pieces of the private key with a plurality of third users of the blockchain system, referred to as Key Holders. The methods and devices may further allow the Offeror to setup parameters for an auction and record the auction on the blockchain system. The methods and devices may record the auction in a smart contract, which may be utilized to collect bids submitted by one or more second users of the blockchain system, referred to as Bidders, during the bidding process. At the end of the bidding process, the methods and devices may request at least a subset of Key Holders to provide their shared pieces of the private key to the smart contract so that the smart contract can reconstruct the private key to decrypt the bids submitted by the Bidders and settle the auction result.

[0015] Embodiments disclosed in the specification have one or more technical effects. In some embodiments, the methods and devices record and facilitate auctions on a blockchain system. This allows the methods and devices to ensure that data related to the auctions can be securely and immutably stored. In some

embodiments, the methods and devices allow the Offeror to split the private key into N pieces and share the pieces of the private key with a plurality of Key Holders, e.g., using Shamir's secret sharing scheme. This allows the methods and devices to provide safekeeping of the private key generated by the Offeror. In some embodiments, the methods and devices utilize a smart contract to manage the bidding process and collect encrypted bids submitted by one or more Bidders during the bidding process. This allows the methods and devices to automatically handle the bidding process without relying on any single party's trust. This also allows the methods and devices to record and track all bids in a secure and immutable manner and allows the auction to be conducted efficiently. In some embodiments, the methods and devices may request at least a subset of Key Holders to provide their shared pieces of the private key to the smart contract so that the smart contract can reconstruct the private key and use the reconstructed private key to settle the auction result. This allows the methods and devices to preserve privacy and secrecy of the bids received because no Key Holder is able to independently decrypt any of the received bids. In this manner, no bidder is allowed to see other bidders' bids, effectively protecting the bidders' privacy and ensuring the fairness of the auction.

[0016] Blockchain systems, also known as distributed ledger systems (DLSs) or consensus systems, may enable participating parties 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.

[0017] A blockchain system is implemented using a peer-to-peer (P2P) network, in which the nodes communicate directly with each other, e.g., without the need of a fixed, central server. Each node in the P2P network may initiate communication with another node in the P2P network. A blockchain system maintains one or more blockchains.

[0018] A blockchain is a data structure that stores data, e.g., transactions, in a way that may prevent tampering and manipulation of the data by malicious parties. The transactions stored in this manner 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 the 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.

[0019] 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. 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 for the block to be valid 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.

[0020] 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 perfect state 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).

[0021] 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).

[0022] 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 valid, 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 valid, and added to the blockchain.

[0023] 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.

[0024] 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.

[0025] 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.

[0026] FIG. 2 illustrates a schematic diagram of a computing device 200 for implementing a node, e.g., the node 102 (FIG. 1), in a blockchain system, according to an embodiment. Referring to FIG. 2, the computing device 200 may include a communication interface 202, a processor 204, and a memory 206.

[0027] The communication interface 202 may facilitate communications between the computing device 200 and devices implementing other nodes, e.g., nodes 104-110 (FIG. 1), in the network. In some embodiments, the communication interface 202 is configured to support one or more communication standards, such as an Internet standard or protocol, an Integrated Services Digital Network (ISDN) standard, etc. In some embodiments, the communication interface 202 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 some embodiments, the communication interface 202 may be based on public cloud infrastructure, private cloud infrastructure, hybrid public/private cloud infrastructure.

[0028] The processor 204 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 204 is coupled with the memory 206 and is configured to execute instructions stored in the memory 206.

[0029] The memory 206 may store processor-executable instructions and data, such as a copy of the blockchain 120 (FIG. 1). The memory 206 may include 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. When the instructions in the memory 206 are executed by the processor 204, the computing device 200 may perform an operation on the blockchain 120.

[0030] FIG. 3 illustrates a flow chart of a method 300 for providing a privacy-preserving blockchain-based auction according to an embodiment. Referring to FIG. 3, multiple users may have accounts on a Blockchain, e.g., the blockchain 120 (FIG. 1). The Blockchain may be implemented to support various types of users, or parties, including, e.g., individuals, businesses, banks, financial institutions, investors, as well as other types of companies, organizations, and the like.

[0031] For illustrative purposes, multiple users are depicted in FIG. 3, which includes an Offeror, a plurality of Key Holders, and a plurality of bidders including,

e.g., a First Bidder and a Second Bidder. The Offeror may be a party, e.g., a company or an organization, offering to sell its assets, such as shares, bonds, or other types of properties through a Dutch auction. The Offeror may also be an auction house or a broker who is offering to sell the assets on behalf of another entity.

[0032] At step 302, the Offeror may generate a pair of public and private keys to facilitate a bidding process. The Offeror may generate the public and private keys using various types of key generation schemes or cryptographic algorithms. In some embodiments, the public key (PK) may be disseminated publicly while the private key (SK) may be kept by the Offeror as a secret.

[0033] At step 304, the Offeror may split the private key into N pieces for sending to N Key Holders. In some embodiments, the Offeror may split the private key into N pieces with a threshold value of k so that the private key can be reconstructed by knowing k out of N pieces using Shamir's secret sharing scheme. Shamir's secret sharing is a cryptographic algorithm where a secret is divided into pieces and each participant is given its own unique shared piece of the secret. Shamir's secret sharing scheme may allow a subset of such participants (e.g., k out of N participants) to reconstruct the original secret. To that end, the Offeror may distribute the N pieces to N different Key Holders. As will be described below, at the end of the auction, as long as k out of the N Key Holders can provide their shared pieces of the private key to a smart contract executing on the Blockchain, the smart contract can reconstruct the original private key and settle the auction result.

[0034] In some embodiments, the Offeror may select any user of the Blockchain to serve as a Key Holder, and the Key Holder may or may not be a trusted user. In some embodiments, to ensure that the private key can be

reconstructed at the end of the auction, the Offeror may require the total number of Key Holders that are considered to be honest, L , to be greater than half of the total number of Key Holders, N , and that L be greater than the threshold value k ($L > 1/2 \times N$ and $L > k$). In some embodiments, the Offeror may set k to be one third of the total Key Holders ($k = 1/3 \times N$) and the total number of honest Key Holders to be greater than or equal to two thirds of total Key Holders ($L \geq 2/3 \times N$).

[0035] In some embodiments, the Offeror may maintain a list of users whom the Offeror considers to be dishonest (e.g., users who have maliciously or prematurely released their shared pieces of secret in the past). Alternatively, or additionally, the Offeror may maintain a list of users whom the Offeror considers to be honest (e.g., users who have kept their shared pieces of secret and participated in reconstructions of private keys in good faith in the past). The Offeror may utilize these lists to determine which users may serve as the Key Holders at step 304. It is to be understood, however, that Shamir's secret sharing scheme and the relationship between L , k , and N as described above are provided as examples and are not meant to be limiting. It is contemplated that other secret sharing schemes and other configurations may be utilized to implement step 304.

[0036] At step 306, the Offeror may submit a transaction to deposit the assets to be auctioned off into a smart contract executing on the Blockchain. Smart contracts are computer protocols implemented in the form of computer code that are incorporated into the Blockchain, to facilitate, verify, or enforce the negotiation or performance of contracts. For example, users of the Blockchain 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 on the Blockchain, 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. In this manner, when the auction settles, the smart contract may automatically transfer the assets deposited therein to one or more bidders.

[0037] The Offeror may also specify the parameters for the auction and record the parameters on the Blockchain at step 306. In some embodiments, the parameters may include S , T_n , P_0 , and P_t . S may represent the total quantity offered, which may be expressed as, e.g., the total number of shares. T_n may represent the auction period, which may be expressed as a duration (e.g., 8 hours after the auction starts) or a particular end time (e.g., a time instance when the auction ends). P_0 may represent a start price, which is typically a high asking price for Dutch auction. The smart contract may gradually decrease the price throughout the auction period, and in some embodiments, the smart contract may decrease the price so that the price at the end of the auction is decreased to a specified end price P_t . In some embodiments, the price may be decreased linearly from P_0 to P_t over the auction period once the auction starts.

[0038] At step 308, the Offeror may trigger the smart contract to start the auction. For example, the Offeror may issue a command to start the auction via a function call. The start time of the auction may be denoted as T_0 , after which the bidders may submit their bids to the smart contract at any time instance T_i , where $T_0 < T_i < T_n$. In some embodiments, the bidders may be required to use the public key (PK) generated by the Offeror at step 302 to encrypt their bids.

[0039] In some embodiments, the bidders may submit their bids to indicate the quantity (e.g., the number of shares) they want to purchase. In some embodiments,

the bidders do not need to indicate the price in their bids. Instead, a bidder may choose to submit its bid at particular time instance T_i and let the smart contract determine the price to be associated with the bid based on its corresponding time instance T_i . In this manner, as the smart contract gradually decreases the price throughout the auction period, bids submitted at different time instances may be associated with different prices. In some embodiments, the rate at which the price is decreased with respect to time may be known to the bidders because it is a function of T_n , P_0 , and P_t , which are all recorded on the Blockchain. In this manner, the bidders can submit their bids at appropriate times of their choosing.

[0040] For example, at step 310 and at time instance T_1 , the smart contract may receive a first encrypted bid from the First Bidder. The smart contract may also retrieve from the first encrypted bid a first encrypted value R_1 , which may represent an encrypted value of a first quantity Q_1 that the First Bidder wants to purchase. However, because R_1 is an encrypted value, no other bidders are able to determine the actual value of Q_1 , effectively preserving the privacy of the First Bidder.

[0041] Subsequently, at step 312 and at time instance T_2 , the smart contract may receive a second encrypted bid from the Second Bidder. The smart contract may also retrieve from the second encrypted bid a second encrypted value R_2 , which may represent an encrypted value of a second quantity Q_2 that the Second Bidder wants to purchase (Q_1 and Q_2 may be the same or may be different). Because R_2 is an encrypted value, no other bidders are able to determine the actual value of Q_2 , effectively preserving the privacy of the Second Bidder.

[0042] The smart contract may continue to receive additional encrypted bids from additional bidders. The smart contract may also continue to retrieve from the additional encrypted bids encrypted values R_i representing encrypted values of Q_i

that the additional bidders wants to purchase. In some embodiments, the smart contract may continue to receive additional encrypted bids until the end of the auction period, at time T_n . After time T_n , the auction may end, and the smart contract may stop accepting additional bids.

[0043] At step 314, the Key Holders who have received the shared pieces of the private key at step 304 may submit their share pieces of the private key to the smart contract. In some embodiments, step 314 may be triggered automatically at time T_n .

[0044] At step 316, the smart contract may collect the shared pieces of the private key from the Key Holders. When at least k share pieces have been collected, the smart contract may reconstruct the original private key, which can be used by the smart contract to settle the auction.

[0045] To settle the auction, the smart contract may arrange all encrypted values of R_i into an ordered list based on their corresponding submission time T_i . In some embodiments, the smart contract may arrange the encrypted values of R_i sequentially according to the order in which they are received. The smart contract may then set an accumulated bid quantity Q to 0, and for each encrypted value R_i , $i \in \{1, 2, 3, \dots\}$, the smart contract may decrypt R_i using the reconstructed private key to obtain the value of Q_i and add the value of Q_i to the accumulated bid quantity Q . The smart contract may repeat this process until the addition of the current value of Q_i to Q makes Q greater than or equal to the total quantity offered S . The smart contract may then determine the time instance T_i at which the current value of Q_i is received. The smart contract may also use the time instance T_i to determine the price P_i , and once the price P_i is determined, every bidder who submitted their bids at or prior to T_i may be contracted to purchase the number of shares they bid for at

the determined price P_i . In some embodiments, the smart contract may automatically execute the purchasing transactions and transfer the ownership of the assets deposited into the smart contract to the bidders to settle the auction.

[0046] FIG. 4 illustrates a flow chart of a method 400 for providing a privacy-preserving blockchain-based auction, according to an embodiment. The method 400 may be performed by one or more nodes in a blockchain system, e.g., the nodes 102-110 in the blockchain system 100 (FIG. 1). The nodes 102-110 in the blockchain system 100 may perform operations on a blockchain, e.g., the blockchain 120 (FIG. 1). The blockchain 120 may be implemented as the Blockchain in the examples described above.

[0047] At step 402, a node, e.g., the node 102, may receive data representing one or more assets that are offered to be auctioned off using a blockchain-based auction and a set of parameters for the auction. The node 102 may receive the assets and the parameters from a first user, e.g., the Offeror (FIG. 3). In some embodiments, the assets offered to be auctioned off may include, e.g., shares, bonds, or other types of properties. The parameters may include a total quantity of the assets offered (S), an auction period (T_n), a start price (P_0), and an end price (P_t). In some embodiments, the node 102 may receive a start command from the Offeror and start the auction.

[0048] At step 404, the node 102 may receive one or more encrypted bids from one or more users, e.g., Bidders (FIG. 3). In some embodiments, the encrypted bids are received after the start of the auction and before the end of the auction period.

[0049] At step 406, the node 102 may receive a plurality of shared pieces of a private key from a plurality of users, e.g., Key Holders (FIG. 3). In some

embodiments, the plurality of shared pieces of the private key may be received after the end of the auction period.

[0050] At step 408, the node 102 may reconstruct the private key based on the plurality of shared pieces received from the key holders. The node 102 may then utilize the reconstructed private key to decrypt the received encrypted bids.

[0051] At step 410, the node 102 may automatically settle the auction based on the parameters for the auction, the received encrypted bids, and the submission times at which the encrypted bids are received. In some embodiments, the node 102 may arrange the received encrypted bids based on their corresponding submission times. The node 102 may also decrypt the received encrypted bids using the reconstructed private key and identify the i -th bid at which the accumulated bid quantity (Q) equals to or exceeds the total quantity of the assets offered (S). The node 102 may then determine the price P_i , which may correspond to the time T_i at which the i -th bid was submitted. The node 102 may settle the auction based on the determined price P_i .

[0052] In some embodiments, the node 102 may automatically execute transactions and transfer the ownership of the assets deposited into the smart contract to the bidders to settle the auction. For example, the node 102 may identify the bids received at or prior to the time T_i , and for each bid received at or prior to the time T_i , the node 102 may determine a quantity specified in the bid, automatically transfer the specified quantity of the assets from the offeror to the bidder who submitted the bid, and automatically transfer an amount equal to the determined price P_i multiplied by the specified quantity from the bidder to the offeror. In some embodiments, the node 102 may determine the price P_i as a function of the auction period, the start price, and the end price. In some embodiments, the determined

price P_i may decrease linearly from the start price to the end price over the auction period after the auction starts.

[0053] FIG. 5 is a block diagram of a privacy-preserving blockchain-based auction apparatus 500, according to an embodiment. The apparatus 500 may be an implementation of a software process, and may correspond to the method 400 (FIG. 4). Referring to FIG. 5, the apparatus 500 may include a receiving module 502, a reconstruction module 504, and a settling module 506.

[0054] The receiving module 502 may receive data representing one or more assets that are offered to be auctioned off using a blockchain-based auction and a set of parameters for the auction. The receiving module 502 may also receive one or more encrypted bids from one or more bidders and receive a plurality of shared pieces of a private key from a plurality of key holders at the end of the auction. The receiving module 502 may provide the received shared pieces of the private key to the reconstruction module 504, which may reconstruct the private key. The reconstructed private key may be utilized by the settlement module 506 to decrypt the received encrypted bids and settle the auction.

[0055] In some embodiments, the settlement module 506 may automatically settle the auction based on the parameters for the auction, the received encrypted bids, and the submission times at which the encrypted bids are received. In some embodiments, the settlement module 506 may arrange the received encrypted bids based on their corresponding submission times. The settlement module 506 may also decrypt the received encrypted bids using the reconstructed private key and identify the i -th bid at which the accumulated bid quantity (Q) equals to or exceeds the total quantity of the assets offered (S). The settlement module 506 may then determine the price P_i , which may correspond to the time T_i at which the i -th bid was

submitted. The settlement module 506 may settle the auction based on the determined price P_i as described above.

[0056] 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 500 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.

[0057] For an implementation process of functions and roles of each module in the apparatus 500, references can be made to corresponding steps in the above-described methods. Details are omitted here for simplicity.

[0058] In some embodiments, a 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.

[0059] 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.

[0060] 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).

[0061] 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

[0062] 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.

[0063] 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.

[0064] Although the specification has been described in conjunction with specific embodiments, 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.

What Is Claimed Is:

1. A computer-implemented method for providing a privacy-preserving blockchain-based auction, the method comprising:

receiving, from a first user, data representing one or more assets offered to be auctioned off and a set of parameters for the auction;

receiving one or more encrypted bids from one or more second users;

receiving a plurality of shared pieces of a private key from a plurality of third users, the private key being generated by the first user and being split into the share pieces for safekeeping;

reconstructing the private key based on the plurality of shared pieces received from the plurality of third users, the reconstructed private key being configured to decrypt the received encrypted bids; and

automatically settling the auction based on the parameters for the auction, the received encrypted bids, and submission times at which the encrypted bids are received.

2. The method of claim 1, wherein the set of parameters for the auction comprises a total quantity of the assets offered, an auction period, a start price, and an end price.

3. The method of claim 2, further comprising:

receiving a command to start the auction; and

starting the auction in response to receiving the command.

4. The method of claim 3, further comprising:
receiving the one or more encrypted bids after the start of the auction and before an end of the auction period.
5. The method of claim 4, further comprising:
receiving the plurality of shared pieces of the private key after the end of the auction period.
6. The method of any of claims 2 to 5, wherein the automatically settling the auction further comprises:
arranging the received encrypted bids based on their corresponding submission times;
decrypting the received encrypted bids using the reconstructed private key;
identifying an i -th bid at which an accumulated bid quantity equals to or exceeds the total quantity of the assets offered;
determining a price P_i corresponding to a time T_i at which the i -th bid was submitted; and
settling the auction based on the determined price P_i .
7. The method of claim 6, wherein the settling the auction based on the determined price P_i further comprises:
for each encrypted bid received at or prior to the time T_i :
determining a quantity specified in the decrypted bid corresponding to the encrypted bid;

automatically transferring the specified quantity of the assets from the first user to a second user who submitted the bid; and

automatically transferring an amount equal to the determined price P_i multiplied by the specified quantity from the second user to the first user.

8. The method of claim 6 or 7, wherein the determined price P_i is determined as a function of the auction period, the start price, and the end price.

9. The method of claim 8, wherein the determined price P_i decreases linearly from the start price to the end price over the auction period after the auction starts.

10. A device for providing a privacy-preserving blockchain-based auction, 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 9.

11. An apparatus for providing a privacy-preserving blockchain-based auction, the apparatus comprising a plurality of modules for performing the method of any of claims 1 to 9.

12. 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 9.