

METHOD, APPARATUS AND ELECTRONIC DEVICE FOR BLOCKCHAIN-BASED TRANSACTION CONSENSUS PROCESSING

TECHNICAL FIELD

5 [0001] One or more implementations of the present specification relate to the field of blockchain technologies, and in particular, to a method, an apparatus and an electronic device for blockchain-based transaction consensus processing.

BACKGROUND

10 [0002] Blockchain technology, also referred to as distributed ledger technology, is an emerging technology that several computing devices participate in "accounting" to jointly maintain a complete distributed database. Blockchain technology is characterized by decentralization, openness, and transparency, each computing device can participate in database recording, and data can be synchronized rapidly between the computing devices. Therefore, blockchain technology has been widely applied in
15 many fields.

SUMMARY

[0003] The present specification provides a method for blockchain-based transaction consensus processing. Node devices in a blockchain include at least one primary node device and several secondary node devices, the primary node device
20 fragments proposed transaction data into a specified number of data fragments based on an erasure code algorithm, and the method includes:

[0004] receiving a data fragment of the transaction data that is sent by the primary node device in a unicast mode, where respective data fragments sent by the primary node device to individual node devices in a unicast mode are different from one
25 another;

[0005] broadcasting the received data fragment to other node devices in the blockchain, and receiving data fragments of the transaction data that are broadcast by

the other node devices;

[0006] determining whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold; and

5 [0007] in response to determining that the number of received data fragments of the transaction data reaches the erasure code recovery threshold, performing data recovery on the received data fragments based on an erasure code reconstruction algorithm to obtain original content of the transaction data, to complete consensus processing with respect to the original content of the transaction data.

[0008] Optionally, the method further includes:

10 [0009] determining whether a current node device is elected as the primary node device;

[0010] in response to determining that the current node device is elected as the primary node device, fragmenting the proposed transaction data into the specified number of data fragments based on the erasure code algorithm; and

15 [0011] separately sending the specified number of data fragments to each of the other node devices in a unicast mode.

[0012] Optionally, a consensus algorithm equipped on the blockchain is a practical Byzantine fault tolerance (PBFT) algorithm; and

20 [0013] where receiving the data fragment of the transaction data that is sent by the primary node device in a unicast mode includes:

[0014] receiving a pre-prepare message sent by the primary node device in a unicast mode, where the pre-prepare message includes a data fragment of the transaction data; and

25 [0015] obtaining and storing the data fragment included in the pre-prepare message.

[0016] Optionally, broadcasting the received data fragment to the other node devices in the blockchain includes:

[0017] broadcasting a prepare message to the other node devices in the blockchain, where the prepare message includes the received data fragment, so that when

receiving the prepare message, the other node devices obtain and store the data fragment included in the prepare message.

[0018] Optionally, fragmenting the proposed transaction data into the specified number of data fragments based on the erasure code algorithm includes:

5 [0019] performing compression processing on the proposed transaction data based on a predetermined compression algorithm; and

[0020] fragmenting the compressed transaction data into the specified number of data fragments based on the erasure code algorithm; and

10 [0021] where performing data recovery on the received data fragments based on the erasure code reconstruction algorithm to obtain the original content of the transaction data includes:

[0022] performing data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the compressed transaction data; and

15 [0023] performing decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm, to obtain the original content of the transaction data.

[0024] Optionally, fragmenting the proposed transaction data into the specified number of data fragments based on the erasure code algorithm includes:

20 [0025] performing encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key;

[0026] fragmenting the encrypted transaction data into the specified number of data fragments based on the erasure code algorithm; and

25 [0027] where performing data recovery on the received data fragments based on the erasure code reconstruction algorithm to obtain the original content of the transaction data includes:

[0028] performing data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the encrypted transaction data; and

[0029] performing decryption processing on the recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption

key corresponding to the encryption key, to obtain the original content of the transaction data.

[0030] Optionally, the encryption algorithm is a threshold encryption algorithm, the decryption algorithm is a threshold decryption algorithm corresponding to the threshold encryption algorithm, the decryption key is fragmented into a specified number of sub-keys, and each of the sub-keys is held by a respective node device; and

[0031] performing decryption processing on the recovered transaction data based on the decryption algorithm corresponding to the encryption algorithm and the decryption key corresponding to the encryption key includes:

10 [0032] collecting sub-keys held by the other node devices;

[0033] determining whether the number of collected sub-keys reaches a predetermined decryption threshold; and

[0034] in response to determining that the number of collected sub-keys reaches the predetermined decryption threshold, reconstructing the decryption key based on the collected sub-keys, and performing decryption processing on the recovered transaction data based on the threshold decryption algorithm corresponding to the threshold encryption algorithm and the decryption key.

[0035] Optionally, the transaction data proposed by the primary node device is a transaction list constructed using to-be-negotiated transactions broadcast by user clients in a current consensus period of the primary node device, and the specified number is a total number of node devices in the blockchain.

[0036] Optionally, the blockchain is a consortium blockchain.

[0037] The present specification further provides an apparatus for blockchain-based transaction consensus processing. Node devices in a blockchain include at least one primary node device and several secondary node devices, the primary node device fragments proposed transaction data into a specified number of data fragments based on an erasure code algorithm, and the apparatus includes:

[0038] a receiving module, configured to receive a data fragment of the transaction data that is sent by the primary node device in a unicast mode, where respective data fragments sent by the primary node device to individual node devices in a unicast

mode are different from one another;

[0039] a sending module, configured to broadcast the received data fragment to other node devices in the blockchain, and receive data fragments of the transaction data that are broadcast by the other node devices;

- 5 **[0040]** a determining module, configured to determine whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold; and

[0041] a recovery module, configured to, in response to determining that the number of received data fragments of the transaction data reaches the erasure code recovery threshold, perform data recovery on the received data fragments based on an
10 erasure code reconstruction algorithm to obtain original content of the transaction data, to complete consensus processing with respect to the original content of the transaction data.

[0042] Optionally, the apparatus further includes:

- 15 **[0043]** a fragmentation module, configured to determine whether the current node device is elected as the primary node device; and in response to determining that the current node device is elected as the primary node device, fragment the proposed transaction data into the specified number of data fragments based on the erasure code algorithm; and

- 20 **[0044]** the sending module is further configured to:

[0045] separately send the specified number of data fragments to each of the other node devices in a unicast mode.

[0046] Optionally, a consensus algorithm equipped on the blockchain is a PBFT algorithm; and

- 25 **[0047]** the receiving module is configured to:

[0048] receive a pre-prepare message sent by the primary node device in a unicast mode, where the pre-prepare message includes a data fragment of the transaction data; and

[0049] obtain and store the data fragment included in the pre-prepare message.

[0050] Optionally, the sending module is configured to:

[0051] broadcast a prepare message to the other node devices in the blockchain, where the prepare message includes the received data fragment, so that when receiving the prepare message, the other node devices obtain and store the data
5 fragment included in the prepare message.

[0052] Optionally, the fragmentation module is configured to:

[0053] perform compression processing on the proposed transaction data based on a predetermined compression algorithm; and

[0054] fragment the compressed transaction data into the specified number of data
10 fragments based on the erasure code algorithm; and

[0055] the recovery module is configured to:

[0056] perform data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the compressed transaction data; and

[0057] perform decompression processing on the recovered transaction data based
15 on a decompression algorithm corresponding to the compression algorithm, to obtain the original content of the transaction data.

[0058] Optionally, the fragmentation module is configured to:

[0059] perform encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key;

20 **[0060]** fragment the encrypted transaction data into the specified number of data fragments based on the erasure code algorithm; and

[0061] the recovery module is configured to:

[0062] perform data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the encrypted transaction data; and

25 **[0063]** perform decryption processing on the recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption key corresponding to the encryption key, to obtain the original content of the transaction data.

- [0064] Optionally, the encryption algorithm is a threshold encryption algorithm, the decryption algorithm is a threshold decryption algorithm corresponding to the threshold encryption algorithm, the decryption key is fragmented into a specified number of sub-keys, and each of the sub-keys is held by a respective node device, and
- 5 [0065] the recovery module is further configured to:
- [0066] collect sub-keys held by the other node devices;
- [0067] determine whether the number of collected sub-keys reaches a predetermined decryption threshold; and
- [0068] in response to determining that the number of collected sub-keys reaches the
10 predetermined decryption threshold, reconstruct the decryption key based on the collected sub-keys, and perform decryption processing on the recovered transaction data based on the threshold decryption algorithm corresponding to the threshold encryption algorithm and the decryption key.
- [0069] Optionally, the transaction data proposed by the primary node device is a
15 transaction list constructed using to-be-negotiated transactions broadcast by user clients in a current consensus period of the primary node device, and the specified number is a total number of node devices in the blockchain.
- [0070] Optionally, the blockchain is a consortium blockchain.
- [0071] The present specification further provides an electronic device, including:
- 20 [0072] a processor; and
- [0073] a memory, configured to store machine-executable instructions,
- [0074] where by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:
- 25 [0075] receive a data fragment of transaction data that is sent by a primary node device in a unicast mode, where node devices in a blockchain include at least one primary node device and several secondary node devices, the primary node device fragments the proposed transaction data into a specified number of data fragments, and respective data fragments sent by the primary node device to individual node

devices in a unicast mode are different from one another;

[0076] broadcast the received data fragment to other node devices in the blockchain, and receive data fragments of the transaction data that are broadcast by the other node devices;

5 [0077] determine whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold; and

[0078] in response to determining that the number of received data fragments of the transaction data reaches the erasure code recovery threshold, perform data recovery on the received data fragments based on an erasure code reconstruction algorithm to
10 obtain original content of the transaction data, to complete consensus processing with respect to the original content of the transaction data.

[0079] According to the above technical solutions, when proposing the to-be-negotiated transaction data to the secondary node devices, the primary node device fragments the to-be-negotiated transaction data based on the erasure code
15 algorithm before transmission.

[0080] As such, in one aspect, at the time when the primary node device proposes the transaction data to the secondary node devices, the primary node device no longer needs to broadcast the complete transaction data to the secondary node devices, but can send data fragment(s) to each secondary node device in a unicast mode. Thus,
20 data transmission bandwidth consumed for transmitting the to-be-negotiated transaction data to the node devices participating in the consensus can be significantly reduced, and the to-be-negotiated transaction data can be transmitted to the secondary node devices in a short time to complete the consensus, thereby improving consensus processing efficiency.

25 [0081] In another aspect, because the primary node device sends only data fragment(s) of the transaction data to each secondary node device in a unicast mode, the primary node device cannot learn of the complete content of a complete transaction that the data fragment belongs. Therefore, the following case can be effectively avoided: The primary node device selectively sends some specific
30 transactions to other node device(s) participating in the consensus in a unicast mode, and conducts a selective consensus on some specific transactions. As such, fairness of

the consensus can be ensured, and the primary node device can be prevented from doing "evil" through the selective consensus.

BRIEF DESCRIPTION OF DRAWINGS

5 **[0082]** FIG. 1 is a flowchart illustrating a method for blockchain-based transaction consensus processing, according to an example implementation;

[0083] FIG. 2 is a schematic diagram illustrating interactions in three phases of an optimized PBFT algorithm, according to an example implementation;

[0084] FIG. 3 is a schematic diagram illustrating a structure of an electronic device, according to an example implementation; and

10 **[0085]** FIG. 4 is a block diagram illustrating an apparatus for blockchain-based transaction consensus processing, according to an example implementation.

DESCRIPTION OF IMPLEMENTATIONS

15 **[0086]** The present specification is intended to provide technical solutions that, in a course of blockchain consensus processing, an erasure code algorithm is introduced to fragment to-be-negotiated transaction data before transmission, to reduce data transmission bandwidth consumed for transmitting the to-be-negotiated transaction data to node devices participating in the consensus and to improve consensus efficiency.

20 **[0087]** During implementation, before each round of consensus in the blockchain starts, one primary node device can be elected from node devices in the blockchain (for example, one primary node device is re-elected for each round of consensus and other node devices serve as secondary node devices), and the primary node device initiates a transaction consensus, and is responsible for creating the latest block for the blockchain based on consensus-agreed transaction data.

25 **[0088]** When initiating a round of transaction consensus, the primary node device can first fragment proposed to-be-negotiated transaction data into a specified number of data fragments based on an erasure code algorithm, and then send the data fragments obtained through fragmentation to the other node devices separately in a

unicast mode. The respective data fragments sent by the primary node device to individual node devices in a unicast mode are different from one another.

5 **[0089]** Second, upon receiving a data fragment of the transaction data that is sent by the primary node device in a unicast mode, each secondary node device can continue to broadcast the received data fragment to other node devices in the blockchain, and can further collect data fragments broadcast by the other node devices.

10 **[0090]** Finally, any node device (including the primary node device and the secondary node devices) in the blockchain can determine whether the number of collected data fragments of the transaction data reaches an erasure code recovery threshold defined when the erasure code algorithm is used to fragment the transaction data. If so, it indicates that the currently collected data fragments are sufficient to recover original content of the transaction data. In this case, data recovery can be performed on the collected data fragments based on an erasure code reconstruction
15 algorithm to obtain the original content of the transaction data, to complete consensus processing with respect to the original content of the transaction data.

20 **[0091]** According to the above technical solutions, when proposing the to-be-negotiated transaction data to the secondary node devices, the primary node device fragments the to-be-negotiated transaction data based on the erasure code algorithm before transmission.

25 **[0092]** As such, in one aspect, at the time when the primary node device proposes the transaction data to the secondary node devices, the primary node device no longer needs to broadcast the complete transaction data to the secondary node devices, but can send data fragment(s) to each secondary node device in a unicast mode. Thus, data transmission bandwidth consumed for transmitting the to-be-negotiated transaction data to the node devices participating in the consensus can be significantly reduced, and the to-be-negotiated transaction data can be transmitted to the secondary node devices in a short time to complete the consensus, thereby improving consensus processing efficiency.

30 **[0093]** In another aspect, because the primary node device sends only data fragment(s) of the transaction data to each secondary node device in a unicast mode, the primary node device cannot learn of the complete content of a complete

transaction that the data fragment belongs. Therefore, the following case can be effectively avoided: The primary node device selectively sends some specific transactions to other node device(s) participating in the consensus in a unicast mode, and conducts a selective consensus on some specific transactions. As such, fairness of the consensus can be ensured, and the primary node device can be prevented from doing "evil" through the selective consensus.

[0094] The following describes the present specification by using specific implementations with reference to specific application scenarios.

[0095] FIG. 1 shows a method for blockchain-based transaction consensus processing, according to an implementation of the present specification. The method is applied to any node device in a blockchain. Node devices in the blockchain include at least one primary node device and several secondary node devices. The primary node device fragments proposed transaction data into a specified number of data fragments based on an erasure code algorithm. The following steps are performed.

[0096] Step 102: Receive a data fragment of the transaction data that is sent by the primary node device in a unicast mode, where respective data fragments sent by the primary node device to individual node devices in a unicast mode are different from one another.

[0097] Step 104: Broadcast the received data fragment to other node devices in the blockchain, and receive data fragments of the transaction data that are broadcast by the other node devices.

[0098] Step 106: Determine whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold.

[0099] Step 108: If so, perform data recovery on the received data fragments based on an erasure code reconstruction algorithm to obtain original content of the transaction data, to complete consensus processing with respect to the original content of the transaction data.

[00100] The blockchain described in the present specification can specifically include a private blockchain, a public blockchain, a consortium blockchain, or the like. No specific limitation is provided in the present specification.

[00101] For example, in a scenario, a specific blockchain can be a consortium blockchain that includes the following member devices: a server of a third-party payment platform, a server of a domestic bank, a server of an overseas bank, and several user node devices. An operator of the consortium blockchain can deploy consortium blockchain-based online services such as cross-border transfer and asset transfer online based on the consortium blockchain.

[00102] A transaction described in the present specification refers to data that is created by a user by using a client of a blockchain and needs to be eventually published to a distributed database of the blockchain.

[00103] Transactions in the blockchain include a transaction in a narrow sense and a transaction in a broad sense. A transaction in the narrow sense refers to value transfer published by a user to the blockchain. For example, in a conventional bitcoin blockchain network, the transaction can be a transfer initiated by the user in the blockchain. A transaction in the broad sense refers to service data with a service intention that is published by a user to the blockchain. For example, an operator can create a consortium blockchain based on an actual service requirement, and deploy some other types of online services (for example, a rental service, a vehicle scheduling service, an insurance claim service, a credit service, and a medical service) that are not related to value transfer in the consortium blockchain. In such a consortium blockchain, the transaction can be a service message or service request with a service intention that is published by a user to the consortium blockchain.

[00104] It should be noted that a consensus algorithm equipped on the blockchain is not specifically limited in the present specification. In practice, specifically, a Byzantine fault tolerance algorithm can be used as the consensus algorithm, or a non-Byzantine fault tolerance algorithm can be used as the consensus algorithm.

[00105] The Byzantine fault tolerance algorithm refers to a distributed fault tolerance algorithm, for example, a PBFT algorithm, in which a Byzantine node (namely, an evil node) needs to be considered in a distributed network including several node devices. If the Byzantine fault tolerance algorithm is used to perform consensus processing in the blockchain network, it is considered that both evil node(s) and faulty node(s) exist in the blockchain. Correspondingly, the non-Byzantine fault tolerance algorithm refers to a distributed fault tolerance algorithm, for example, a raft

algorithm, in which no Byzantine node is considered in a distributed network including several node devices. If the non-Byzantine fault tolerance algorithm is used to perform consensus processing in the blockchain network, it is considered that no evil node exists and only faulty node(s) exists in the blockchain.

5 **[00106]** The transaction data proposed by the primary node device described in the present specification specifically refers to to-be-negotiated transaction(s) that is collected by the primary node device and submitted by user client(s) by using connected node device(s). For example, in an implementation, a user client can broadcast a transaction initiated by the user in the blockchain by using a connected
10 node device.

[00107] In practice, the primary node device can initiate a consensus for a single transaction, or can initiate a consensus for a created transaction list.

[00108] For example, in an implementation, a user client can broadcast, in the blockchain network, a transaction submitted by a user. The primary node device can
15 collect to-be-negotiated transactions broadcast by various user clients in the blockchain network in a consensus time period of a current round, then create a transaction list based on the collected to-be-negotiated transactions, and use the transaction list as proposed transaction data for transmitting to other node devices.

[00109] The following provides descriptions by using an example where the
20 blockchain is a consortium blockchain, and the consortium blockchain uses the PBFT algorithm as the consensus algorithm to perform consensus processing on the transaction list proposed by the primary node device.

[00110] In the present specification, based on three existing phases, *i.e.*, pre-prepare, prepare, and commit in the PBFT algorithm, an erasure code algorithm can be
25 introduced to optimize and improve an existing transaction data transmission mechanism in the pre-prepare and prepare phases of the PBFT algorithm, so as to reduce data transmission bandwidth for performing consensus interaction between the primary node device and the secondary node devices and improve consensus processing efficiency.

30 **[00111]** FIG. 2 is a schematic diagram illustrating interactions in the three phases of an optimized PBFT algorithm, according to the present specification.

[00112] As shown in FIG. 2, before each round of consensus in a consortium blockchain starts, one primary node device can be elected from node devices in the consortium blockchain.

[00113] For example, based on the PBFT algorithm, before each round of consensus in the consortium blockchain starts, one primary node device can be calculated for the current round according to the following equation:

$$P=v \bmod R,$$

[00114] where P represents a calculated index number of the primary node device of the current round of consensus and v represents a current view index number of the consortium blockchain. The view index number is usually an integer that grows consecutively, representing the number of rounds of successfully completed consensus in the consortium blockchain. For example, after a round of consensus is successfully completed, and the elected primary node device successfully writes a new block into the consortium blockchain, the view index number can automatically increase by 1. R represents a total number of node devices in the consortium blockchain.

[00115] Each node device can perform the above calculation, and perform matching between the calculated index number of the primary node device and an index number of the current device to determine whether the current device is elected as the primary node device.

[00116] In one aspect, if a node device determines that the current device is not elected as the primary node device, the node device can broadcast received transaction(s) that is submitted by client(s) to the node devices in the consortium blockchain.

[00117] In another aspect, if a node device determines that the current device is elected as the primary node device, the node device can collect transactions broadcast by user clients in a consensus time period of the current round, and create a transaction list based on the collected transactions. In this case, the created transaction list is the transaction data that is proposed by the primary node device for consensus processing.

[00118] Further, the primary node device can fragment the proposed transaction list

based on an erasure code algorithm, to fragment the transaction list into a specified number of data fragments.

[00119] It should be noted that the number of data fragments obtained by fragmenting the transaction list can be consistent with the total number of node devices in the consortium blockchain. For example, assuming that there are N node devices in the consortium blockchain, the transaction list can be fragmented into N data fragments.

[00120] A specific process of fragmenting the transaction list based on the erasure code algorithm is omitted for simplicity in the present specification. A person skilled in the art can refer to descriptions in related technologies when implementing the technical solutions of the present specification.

[00121] For example, assume that the transaction list is fragmented into N data fragments based on the erasure code algorithm, and the N data fragments include K data blocks and M check blocks. M represents the number of fault-tolerable data fragments in the N data fragments. K represents the minimum number (namely, an erasure code recovery threshold) of data fragments required to recover the original transaction list. In other words, by using any K data fragments in the N data fragments, original content of the transaction list can be recovered by using an erasure code reconstruction algorithm (namely, an inverse algorithm of the erasure code algorithm).

[00122] When the primary node device completes fragmenting the transaction list based on the erasure code algorithm, the primary node device can send a pre-prepare message to each secondary node device, to separately send the data fragments obtained through fragmentation to the other node devices. In the present specification, the respective data fragments sent by the primary node device to individual secondary node devices need to be different.

[00123] Based on an existing PBFT protocol, the primary node device can add the complete transaction list to a pre-prepare message, and then broadcast the pre-prepare message to node devices in the consortium blockchain, to transmit the complete transaction list to secondary node devices that need to participate in the transaction consensus in the consortium blockchain.

[00124] However, based on an existing transaction data transmission mechanism of the PBFT protocol, because the pre-prepare message needs to include the complete transaction list, a large number of data copies are generated when the pre-prepare message is being broadcast. For example, based on the total number N of node devices in the consortium blockchain, N copies of the pre-prepare message need to be made, and then are broadcast. Therefore, in the PBFT protocol, a large amount of network bandwidth of the consortium blockchain is occupied to transmit the transaction data, and a relatively high requirement is imposed on bandwidth performance of the consortium blockchain. Once the bandwidth performance of the consortium blockchain is insufficient, the to-be-negotiated transaction list cannot be transmitted to other secondary node devices participating in the consensus in a short time.

[00125] In view of this, in the present specification, the existing transaction data transmission mechanism in pre-prepare and prepare phases of the PBFT algorithm can be optimized and improved.

[00126] In one aspect, the pre-prepare message sent by the primary node device to each secondary node device no longer needs to include the complete transaction list, but includes only one data fragment of the transaction list.

[00127] In another aspect, a method used by the primary node device to transmit the pre-prepare messages to each of other secondary node devices can be changed from broadcasting the pre-prepare message(s) to sending a pre-prepare message to each secondary node device in a unicast mode. For example, respective pre-prepare messages can be constructed separately for individual secondary node devices, where the pre-prepare messages include different data fragments, and then the constructed pre-prepare messages are sent to the secondary node devices one by one, so that each secondary node device can receive a different data fragment.

[00128] Referring back to FIG. 2, the secondary node device receiving the pre-prepare message can first perform verification on the received pre-prepare message according to the PBFT algorithm, to determine whether to accept the received pre-prepare message.

[00129] In the present specification, performing verification on the received pre-prepare message is a process of performing verification on the content included in

the received pre-prepare message. A specific verification process is omitted for simplicity in the present specification.

[00130] For example, according to the PBFT algorithm, the pre-prepare message can include information such as to-be-negotiated transaction data (a data fragment in accordance with the present specification), a view index number v , a digest of the to-be-negotiated transaction data (a digest of the data fragment in accordance with the present specification), a digital signature, etc. Correspondingly, in accordance with the present specification, the following verification process can be specifically used to perform verification on the pre-prepare message:

10 **[00131]** (1) Verify whether the view index number v is consistent with a locally recorded view index number.

[00132] (2) Verify the digital signature included in the pre-prepare message.

[00133] (3) Verify the validity of the data fragment included in the pre-prepare message.

15 **[00134]** For example, during implementation, the primary node device can construct a Merkle tree based on all data fragments, and add a hash value of each branch node of the Merkle tree to the pre-prepare message. When performing verification on the data fragment included in the received pre-prepare message, the node device can recalculate a hash value of the data fragment included in the pre-prepare message, and
20 reconstruct a Merkle tree based on the hash value and the hash value of each branch node of the Merkle tree included in the pre-prepare message. Then, the node device makes a comparison using a hash value corresponding to the root of the reconstructed Merkle tree, to perform verification on the data fragment. If the hash value corresponding to the root of the reconstructed Merkle tree does not change, it is
25 considered that the validity verification on the data fragment succeeds. Otherwise, it can be considered that the data fragment(s) included in the pre-prepare message is invalid, and in this case, the data fragment(s) can be discarded.

[00135] Certainly, in practice, content included in the pre-prepare message can also be expanded based on an actual requirement. For example, the pre-prepare message
30 can further include a consensus height h . The consensus height h is functionally similar to the view index number v , and is usually an integer that grows consecutively,

representing the number of consensus rounds (instead of representing the number of rounds of successful consensus) in the consortium blockchain. For example, in a round of consensus, if a primary node device is faulty, an attempt to switch is usually triggered, and a primary node device is re-elected. In this case, the consensus height h can immediately increase by 1. Because this round of consensus has not been successful, the view index number v does not increase by 1, and remains unchanged at the original value of view index number v .

[00136] In this case, for the verification on the pre-prepare message, the following verification can be performed in addition to the above verification process.

10 **[00137]** (4) Verify whether the consensus height h is consistent with a locally recorded consensus height h .

[00138] With continued reference to FIG. 2, after the node device receiving the pre-prepare message performs the above verification process, if the verification succeeds, it indicates that the node device accepts the pre-prepare message. In this case, the node device can obtain and store the data fragment included in the pre-prepare message, and enter the prepare phase of the PBFT protocol, to broadcast a prepare message used to confirm the pre-prepare message to each of the other node devices in the consortium blockchain. A format of content included in the prepare message can be consistent with that of the pre-prepare message.

20 **[00139]** Based on the existing PBFT protocol, the prepare message broadcast by the node device to each of the other node devices in the consortium blockchain usually includes only such data as a view index number v and a digest of to-be-negotiated transaction data. In the present specification, the format of the prepare message can be further expanded, and each node device can also add, to the prepare message, a data fragment sent by the primary node device via the pre-prepare message in a unicast mode.

[00140] In one aspect, each node device can broadcast a prepare message including a data fragment to each of the other node devices in the consortium blockchain, to further spread and transmit, to each of the other node devices in the consortium blockchain, the received data fragment sent by the primary node device in a unicast mode.

[00141] In another aspect, each node device can also receive a prepare message broadcast by each other secondary node device, and perform verification on the received prepare message, to determine whether to accept the received prepare message. For a specific process of performing verification on the received prepare message, references can be made to the specific process of performing verification on the pre-prepare message. Details are omitted for simplicity in the present specification.

[00142] If the verification succeeds, it indicates that the node device accepts the received prepare message.

10 [00143] In one aspect, the node device can obtain and store a data fragment included in the prepare message.

[00144] In another aspect, the node device can further determine whether the number of received prepare messages broadcast by other node devices reaches $2f+1$ (including a prepare message broadcast by the node device itself).

15 [00145] Herein, f represents the number of fault-tolerable nodes in the PBFT algorithm, a specific value of f can be obtained through conversion according to equation $N=3f+1$, where N represents the total number of node devices in the consortium blockchain.

20 [00146] If the number of received prepare messages reaches $2f+1$, the node device can enter a commit phase of the PBFT protocol. The node device can broadcast a commit message to each of the other node devices in the consortium blockchain, and continue to complete the consensus process of the original content of the transaction list.

25 [00147] In the present specification, after broadcasting the commit message to each of the other node devices in the consortium blockchain, the node device in the consortium blockchain can also receive commit message(s) broadcast by other node device(s), and perform verification on the received commit message(s) to determine whether to accept the received commit message(s).

30 [00148] In the present specification, a format of content included in the commit message can be consistent with the formats of the pre-prepare message and the prepare message. However, it should be emphasized that the digest of the

to-be-negotiated transaction data included in the pre-prepare message and the prepare message is a digest of some data of the transaction list, while in the commit phase, because the node device has recovered the complete transaction list, the commit message can include a digest of the recovered complete transaction list.

- 5 **[00149]** For a specific process of performing verification on the received commit message, references can be made to the specific process of performing verification on the pre-prepare message. Details are omitted for simplicity in the present specification.

- 10 **[00150]** If the verification succeeds, it indicates that the node device accepts the received commit message, and the node device can further determine whether the number of received commit messages broadcast by other node devices reaches $2f+1$ (including a commit message broadcast by the node device itself).

- 15 **[00151]** If the number of received commit messages reaches $2f+1$, the node device can further determine whether the number of collected data fragments reaches an erasure code recovery threshold supported by the erasure code algorithm. If so, the node device can immediately perform data recovery calculation on the collected data fragments based on the erasure code reconstruction algorithm to recover the original content of the transaction list.

- 20 **[00152]** A specific process of performing data recovery calculation on the collected data fragments based on the erasure code reconstruction algorithm is omitted for simplicity in the present specification. A person skilled in the art can refer to descriptions in related technologies when implementing the technical solutions of the present specification.

- 25 **[00153]** In an illustrated implementation, once a specific value of the erasure code recovery threshold is larger than $2f+1$, the secondary node device does not have sufficient data fragments in the commit phase to recover the original content of the to-be-negotiated transaction list. Therefore, in the present specification, the specific value of the erasure code recovery threshold supported by the erasure code algorithm needs to be less than or equal to $2f+1$. For example, the value can be exactly equal to
30 $2f+1$.

- [00154]** As such, when fragmenting the transaction list based on the erasure code

algorithm, the primary node device can control the final number of data fragments obtained through fragmentation based on the number f of faulty-tolerable nodes supported by the PBFT algorithm, so that data fragments obtained by fragmenting the transaction list can be as small in size as possible while it is ensured that there are
5 sufficient data fragments to recover the original content of the to-be-negotiated transaction list in the commit phase.

[00155] In the present specification, before fragmenting the transaction list, the primary node device can further preprocess the transaction list based on a supported related algorithm.

10 **[00156]** In an illustrated implementation, before fragmenting the transaction list, the primary node device can further perform compression preprocessing on the transaction list, to perform compression on the transaction list based on a supported compression algorithm.

[00157] The compression algorithm used by the primary node device is not
15 specifically limited in the present specification.

[00158] After completing the compression processing on the transaction list, the primary node device fragments the compressed transaction list based on the erasure code algorithm, to fragment the compressed transaction list into the specified number of data fragments.

20 **[00159]** Correspondingly, in the prepare phase of the PBFT protocol, because a data fragment transmitted by each node device is a data fragment obtained by fragmenting the compressed transaction list, the node device obtains only the compressed transaction list after performing data recovery on the collected data fragments based on the erasure code reconstruction algorithm.

25 **[00160]** In this case, after completing the data recovery calculation, the node device can perform decompression processing on the recovered compressed transaction list based on a decompression algorithm corresponding to the compression algorithm, to obtain the original content of the transaction list through decompression, and then continue to perform the consensus process of the commit phase. Specific
30 implementation details are omitted for simplicity.

[00161] As such, before fragmenting the transaction list, the primary node device

performs compression processing on the transaction list in advance, so that a size of a compressed data fragment can be effectively reduced, and data transmission bandwidth for performing consensus interaction between the primary node device and the secondary node devices can be reduced to the greatest extent.

- 5 **[00162]** In another illustrated implementation, before fragmenting the transaction list, the primary node device can further perform encryption preprocessing on the transaction list, to perform encryption on the transaction list based on a predetermined encryption algorithm and an encryption key.

10 **[00163]** The encryption algorithm used by the primary node device is not specifically limited in the present specification. For example, in practice, the encryption algorithm can be a symmetric encryption algorithm, or can be an asymmetric encryption algorithm.

15 **[00164]** After completing the encryption processing on the transaction list, the primary node device fragments the encrypted transaction list based on the erasure code algorithm, to fragment the encrypted transaction list into the specified number of data fragments.

20 **[00165]** Correspondingly, in the prepare phase of the PBFT protocol, because a data fragment transmitted by each node device is a data fragment obtained by fragmenting the encrypted transaction list, the node device obtains only the encrypted transaction list after performing data recovery on the collected data fragments based on the erasure code reconstruction algorithm.

25 **[00166]** In this case, after completing the data recovery calculation, the node device can perform decryption processing on the recovered encrypted transaction list based on a decompression algorithm corresponding to the compression algorithm and a decryption key corresponding to the encryption key, to obtain the original content of the transaction list through decryption, and then continue to perform the consensus process of the commit phase. Specific implementation details are omitted for simplicity.

30 **[00167]** In an illustrated implementation, the encryption algorithm can be specifically a threshold encryption algorithm. The decryption algorithm can be specifically a threshold decryption algorithm corresponding to the threshold

encryption algorithm.

[00168] In this scenario, the decryption key can be specifically fragmented into a specified number of sub-keys, and each of the sub-keys is held by a respective node device.

5 **[00169]** It should be noted that the number of sub-keys obtained by fragmenting the decryption key can be consistent with the total number of node devices in the consortium blockchain. For example, assuming that there are N node devices in the consortium blockchain, the decryption key can be fragmented into N sub-keys, each held by one of the N node devices in the consortium blockchain, respectively.

10 **[00170]** In this case, the node device can further collect sub-keys held by other node devices, and determine whether the number of collected sub-keys reaches a predetermined decryption threshold. For example, when the decryption threshold is N , it indicates that N node devices need to jointly decrypt encrypted data based on the sub-keys (namely, a part of the decryption key) they hold.

15 **[00171]** A specific method used by the node device to collect the sub-keys held by the other node devices is not specifically limited in the present specification.

[00172] For example, the consensus algorithm used in the consortium blockchain is still the PBFT algorithm. The node device can add a sub-key held by the node device to a commit message exchanged with another node device. Certainly, in practice, node
20 devices can separately define an interactive message used to transmit sub-keys they hold, and synchronize the sub-key held by one node device to another node device by using the interactive message.

[00173] If the number of sub-keys collected by the node device reaches the predetermined decryption threshold, it indicates that the node device has sufficient
25 fragments to reconstruct the decryption key. In this case, the node device can perform reconstruction processing on the decryption key based on the collected sub-keys to recover the original decryption key, and then perform decryption processing on the recovered encrypted transaction list based on the threshold decryption algorithm and the recovered original decryption key, to obtain the original content of the transaction
30 list through decryption.

[00174] A specific process of reconstructing the decryption key based on the

collected sub-keys is omitted for simplicity in the present specification. A person skilled in the art can refer to descriptions in related technologies when implementing the technical solutions of the present specification.

5 [00175] For example, fragmenting the decryption key and recovering the decryption key based on the collected sub-keys can also be implemented by using an erasure code technology. A specific process is omitted for simplicity.

10 [00176] As such, before fragmenting the transaction list, the primary node device performs encryption processing on the transaction list in advance, so that data security of sending a data fragment by the primary node device to each node device can be effectively improved, and some unauthorized node devices cannot view the original content of the transaction list even if they collect sufficient data fragments to recover the transaction list, thereby improving

[00177] data security of consensus interaction between the primary node device and the secondary node devices.

15 [00178] It should be further noted that, in the above implementation, detailed descriptions are provided by using the example where the existing transaction data transmission and spreading mechanism in the pre-prepare and prepare phases of the PBFT algorithm is used to transmit and spread the to-be-negotiated transaction list to each node device participating in the consensus.

20 [00179] It should be emphasized that, in practice, when the to-be-negotiated transaction list needs to be transmitted and spread to each node device participating in the consensus, in addition to or instead of using the existing transmission and spreading mechanism in the consensus algorithm equipped on the consortium blockchain to transmit and spread the transaction list, a separate transmission and spread protocol can be defined to perform the same function.

25 [00180] For example, the consensus algorithm used in the consortium blockchain is still the PBFT algorithm. Before the primary node device and the secondary node devices start the consensus interaction of three phases specified in the PBFT algorithm, the primary node device can perform consensus interaction with the secondary node devices by using a defined transmission and spread protocol, to transmit and spread the to-be-negotiated transaction list to the secondary node devices

in advance. In other words, the node device can transmit and spread the to-be-negotiated transaction list to each node device participating in the consensus by using the defined transmission and spread protocol, and then start the consensus interaction of three phases specified in the PBFT algorithm, to complete the consensus processing on the transaction list.

[00181] In this case, the existing transaction data transmission mechanism in the pre-prepare and prepare phases of the PBFT algorithm does not need to be optimized as described in the above implementation(s). For example, the pre-prepare message and the prepare message no longer need to include the complete transaction list or a data fragment of the transaction list, but can directly include the digest value of the transaction list.

[00182] In addition, in the above implementation, only the PBFT algorithm is used as an example of the consensus algorithm in the consortium blockchain for description. Clearly, in practice, the technical solutions of the present specification can also be applied to other forms of consensus algorithm used in the consortium blockchain.

[00183] In other words, the transmission logic shown in steps 102 to 106 can be applied to other similar consensus algorithms such as a raft algorithm besides the PBFT algorithm. Specific implementation details are omitted for simplicity in the present specification. A person skilled in the art can refer to the descriptions in the above implementation when implementing the technical solutions of the present specification.

[00184] It can be learned from the above implementations that when proposing the to-be-negotiated transaction data to the secondary node devices, the primary node device can fragment the to-be-negotiated transaction data based on the erasure code algorithm before transmission.

[00185] As such, in one aspect, at the time when the primary node device proposes the transaction data to the secondary node devices, the primary node device no longer needs to broadcast the complete transaction data to the secondary node devices, but can send data fragment(s) to each secondary node device in a unicast mode. Thus, data transmission bandwidth consumed for transmitting the to-be-negotiated transaction data to the node devices participating in the consensus can be significantly

reduced, and the to-be-negotiated transaction data can be transmitted to the secondary node devices in a short time to complete the consensus, thereby improving consensus processing efficiency.

[00186] In another aspect, because the primary node device sends only data fragment(s) of the transaction data to each secondary node device in a unicast mode, the primary node device cannot learn of the complete content of a complete transaction that the data fragment belongs. Therefore, the following case can be effectively avoided: The primary node device selectively sends some specific transactions to other node device(s) participating in the consensus in a unicast mode, and conducts a selective consensus on some specific transactions. As such, fairness of the consensus can be ensured, and the primary node device can be prevented from doing "evil" through the selective consensus.

[00187] Corresponding to the above method implementation, the present specification further provides an implementation of an apparatus for blockchain-based transaction consensus processing. The implementation of the apparatus for blockchain-based transaction consensus processing in the present specification can be applied to an electronic device. The apparatus implementation can be implemented by software, hardware, or a combination of hardware and software. Software implementation is used as an example. As a logical apparatus, the apparatus is formed by reading corresponding computer program instructions in a non-volatile storage device to a memory by a processor of an electronic device where the apparatus is located. In terms of hardware, FIG. 3 is a hardware diagram illustrating a structure of an electronic device where the apparatus for blockchain-based transaction consensus processing is located in accordance with the present specification. In addition to a processor, a memory, a network interface, and a non-volatile storage device shown in FIG. 3, the electronic device where the apparatus is located in accordance with the implementations can usually include other hardware based on an actual function of the electronic device. Details are omitted for simplicity.

[00188] FIG. 4 is a block diagram illustrating an apparatus for blockchain-based transaction consensus processing, according to an example implementation of the present specification.

[00189] Referring to FIG. 4, the apparatus 40 for blockchain-based transaction

consensus processing can be applied to the above electronic device shown in FIG. 3, and includes a receiving module 401, a sending module 402, a determining module 403, and a recovery module 404.

5 [00190] The receiving module 401 is configured to receive a data fragment of transaction data that is sent by a primary node device in a unicast mode, where node devices in a blockchain include at least one primary node device and several secondary node devices, the primary node device fragments the proposed transaction data into a specified number of data fragments, and respective data fragments sent by the primary node device to individual node devices in a unicast mode are different
10 from one another;

[00191] the sending module 402 is configured to broadcast the received data fragment to other node devices in the blockchain, and receive data fragments of the transaction data that are broadcast by the other node devices;

15 [00192] the determining module 403 is configured to determine whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold; and

[00193] the recovery module 404 is configured to: if so, perform data recovery on the received data fragments based on an erasure code reconstruction algorithm to obtain original content of the transaction data, to complete consensus processing with
20 respect to the original content of the transaction data.

[00194] In this implementation, the apparatus 40 further includes:

[00195] a fragmentation module 405, configured to determine whether the current node device is elected as the primary node device; and if so, fragment the proposed transaction data into the specified number of data fragments based on the erasure code
25 algorithm; and

[00196] the sending module 402 is further configured to:

[00197] separately send the specified number of data fragments to each of the other node devices in a unicast mode.

[00198] In this implementation, a consensus algorithm equipped on the blockchain is
30 a PBFT algorithm; and

[00199] the receiving module 401 is configured to:

[00200] receive a pre-prepare message sent by the primary node device in a unicast mode, where the pre-prepare message includes a data fragment of the transaction data; and

5 [00201] obtain and store the data fragment included in the pre-prepare message.

[00202] In this implementation, the sending module 402 is configured to:

[00203] broadcast a prepare message to the other node devices in the blockchain, where the prepare message includes the received data fragment, so that when receiving the prepare message, the other node devices obtain and store the data
10 fragment included in the prepare message.

[00204] In this implementation, the fragmentation module 405 is configured to:

[00205] perform compression processing on the proposed transaction data based on a predetermined compression algorithm;

[00206] fragment the compressed transaction data into the specified number of data
15 fragments based on the erasure code algorithm; and

[00207] the recovery module 404 is configured to:

[00208] perform data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the compressed transaction data; and

[00209] perform decompression processing on the recovered transaction data based
20 on a decompression algorithm corresponding to the compression algorithm, to obtain the original content of the transaction data.

[00210] In this implementation, the fragmentation module 405 is configured to:

[00211] perform encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key;

25 [00212] fragment the encrypted transaction data into the specified number of data fragments based on the erasure code algorithm; and

[00213] the recovery module 404 is configured to:

[00214] perform data recovery on the received data fragments based on the erasure

code reconstruction algorithm, to obtain the encrypted transaction data; and

5 [00215] perform decryption processing on the recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption key corresponding to the encryption key, to obtain the original content of the transaction data.

[00216] In this implementation, the encryption algorithm is a threshold encryption algorithm, the decryption algorithm is a threshold decryption algorithm corresponding to the threshold encryption algorithm, the decryption key is fragmented into a specified number of sub-keys, and each of the sub-keys is held by a respective node
10 device.

[00217] The recovery module 404 is further configured to: collect sub-keys held by the other node devices;

[00218] determine whether the number of collected sub-keys reaches a predetermined decryption threshold; and

15 [00219] if so, reconstruct the decryption key based on the collected sub-keys, and perform decryption processing on the recovered transaction data based on the threshold decryption algorithm corresponding to the threshold encryption algorithm and the decryption key.

[00220] In this implementation, the transaction data proposed by the primary node
20 device is a transaction list constructed using to-be-negotiated transactions broadcast by user clients in a current consensus period of the primary node device, and the specified number is a total number of node devices in the blockchain.

[00221] In this implementation, the blockchain is a consortium blockchain.

25 [00222] For an implementation process of functions and roles of each module in the apparatus, references can be made to an implementation process of a corresponding step in the above method. Details are omitted herein for simplicity.

[00223] Because apparatus implementations basically correspond to method implementations, for related parts, references can be made to related descriptions in the method implementations. The above apparatus implementations are merely
30 examples. The modules described as separate parts may or may not be physically

separate, and parts displayed as modules may or may not be physical modules, and can be located in one position, or can be distributed on a plurality of network modules. Some or all of the modules can be selected based on actual requirements to achieve the objectives of the solutions of the present specification. A person of
5 ordinary skill in the art can understand and implement the implementations of the present specification without making innovative efforts.

[00224] The system, apparatus, or module illustrated in the above implementations can be specifically implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical implementation
10 device is a computer, and the computer can be specifically 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, a wearable device, or any combination of these devices.

[00225] Corresponding to the above method implementation, the present
15 specification further provides an implementation of an electronic device. The electronic device includes a processor and a memory configured to store machine-executable instructions. The processor and the memory are usually connected to each other by using an internal bus. In other possible implementations, the device can further include an external interface for communication with another
20 device or component.

[00226] In this implementation, by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00227] receive a data fragment of transaction data that is sent by a primary node
25 device in a unicast mode, where node devices in a blockchain include at least one primary node device and several secondary node devices, the primary node device fragments the proposed transaction data into a specified number of data fragments, and respective data fragments sent by the primary node device to individual node devices in a unicast mode are different from one another;

[00228] broadcast the received data fragment to other node devices in the blockchain,
30 and receive data fragments of the transaction data that are broadcast by the other node devices;

[00229] determine whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold; and

[00230] if so, perform data recovery on the received data fragments based on an erasure code reconstruction algorithm to obtain original content of the transaction data,
5 to complete consensus processing with respect to the original content of the transaction data.

[00231] In this implementation, by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

10 **[00232]** determine whether a current node device is elected as the primary node device;

[00233] if so, fragment the proposed transaction data into the specified number of data fragments based on the erasure code algorithm; and

15 **[00234]** separately send the specified number of data fragments to each of the other node devices in a unicast mode.

[00235] In this implementation, a consensus algorithm equipped on the blockchain is a PBFT algorithm.

20 **[00236]** By reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00237] receive a pre-prepare message sent by the primary node device in a unicast mode, where the pre-prepare message includes a data fragment of the transaction data; and

[00238] obtain and store the data fragment included in the pre-prepare message.

25 **[00239]** In this implementation, by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00240] broadcast a prepare message to the other node devices in the blockchain, where the prepare message includes the received data fragment, so that when

receiving the prepare message, the other node devices obtain and store the data fragment included in the prepare message.

5 **[00241]** In this implementation, by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00242] perform compression processing on the proposed transaction data based on a predetermined compression algorithm; and

[00243] fragment the compressed transaction data into the specified number of data fragments based on the erasure code algorithm.

10 **[00244]** In this implementation, by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00245] perform data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the compressed transaction data; and

15 **[00246]** perform decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm, to obtain the original content of the transaction data.

20 **[00247]** In this implementation, by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00248] perform encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key; and

[00249] fragment the encrypted transaction data into the specified number of data fragments based on the erasure code algorithm.

25 **[00250]** In this implementation, by reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00251] perform data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the encrypted transaction data; and

[00252] perform decryption processing on the recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption key corresponding to the encryption key, to obtain the original content of the transaction data.

5 [00253] In this implementation, the encryption algorithm is a threshold encryption algorithm, the decryption algorithm is a threshold decryption algorithm corresponding to the threshold encryption algorithm, the decryption key is fragmented into a specified number of sub-keys, and each of the sub-keys is held by a respective node device.

10 [00254] By reading and executing the machine-executable instructions stored in the memory and corresponding to control logic of blockchain-based transaction consensus processing, the processor is configured to:

[00255] collect sub-keys held by the other node devices;

15 [00256] determine whether the number of collected sub-keys reaches a predetermined decryption threshold; and

[00257] if so, reconstruct the decryption key based on the collected sub-keys, and perform decryption processing on the recovered transaction data based on the threshold decryption algorithm corresponding to the threshold encryption algorithm and the decryption key.

20 [00258] A person skilled in the art can easily figure out other implementations of the present specification after thinking over the present specification and practicing the present disclosure herein. The present specification is intended to cover any variations, uses, or adaptations of the present specification, and these variations, uses, or adaptations follow the general principles of the present specification and include
25 common knowledge or conventional techniques that are not disclosed in the technical field of the present specification. The present specification and the implementations are merely considered as examples, and the actual scope and the spirit of the present specification are pointed out by the following claims.

[00259] It should be understood that the present specification is not limited to the
30 precise structures that have been described above and shown in the drawings, and various modifications and changes can be made without departing from the scope of

the present specification. The scope of the present specification is limited by the appended claims only.

[00260] The above descriptions are merely preferred implementations of the present specification, but are not intended to limit the present specification. Any modification,
5 equivalent replacement, or improvement made without departing from the spirit and principle of the present specification shall fall within the protection scope of the present specification.

CLAIMS

What is claimed is:

1. A method for processing transaction consensus in a blockchain network, wherein the blockchain network include at least one primary node and a plurality of secondary nodes and wherein a primary node fragments proposed transaction data into a specified number of data fragments based on an erasure code algorithm, the method comprising:

receiving a data fragment of the transaction data that is sent by the primary node in a unicast mode, wherein respective data fragments sent by the primary node in a unicast mode to individual nodes are different from one another;

broadcasting the received data fragment to other nodes in the blockchain network, and receiving data fragments of the transaction data that are broadcast by the other nodes;

determining whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold; and

in response to determining that the number of received data fragments of the transaction data reaches the erasure code recovery threshold, performing data recovery on the received data fragments based on an erasure code reconstruction algorithm to obtain original content of the transaction data, to complete consensus processing with respect to the original content of the transaction data.

2. The method according to claim 1, wherein the method further comprises:

determining whether a current node is elected as the primary node;

in response to determining that the current node device is elected as the primary node device, fragmenting the proposed transaction data into the specified number of data fragments based on the erasure code algorithm; and

separately sending the specified number of data fragments to each of the other node devices in a unicast mode.

3. The method according to claim 1, wherein a consensus algorithm equipped on the blockchain is a practical Byzantine fault tolerance (PBFT) algorithm; and wherein

the receiving the data fragment of the transaction data that is sent by the primary node device in a unicast mode includes:

receiving a pre-prepare message sent by the primary node device in a unicast mode, wherein the pre-prepare message includes a data fragment of the transaction data; and

obtaining and storing the data fragment included in the pre-prepare message.

4. The method according to claim 3, wherein the broadcasting the received data fragment to the other node devices in the blockchain network comprises:

broadcasting a prepare message to the other node devices in the blockchain network, wherein the prepare message includes the received data fragment, so that when receiving the prepare message, the other node devices obtain and store the data fragment included in the prepare message.

5. The method according to claim 1 or 2, wherein the fragmenting the proposed transaction data into the specified number of data fragments based on the erasure code algorithm comprises:

performing compression processing on the proposed transaction data based on a predetermined compression algorithm; and

fragmenting the compressed transaction data into the specified number of data fragments based on the erasure code algorithm; and

wherein the performing data recovery on the received data fragments based on the erasure code reconstruction algorithm to obtain the original content of the transaction data includes:

performing data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the compressed transaction data; and

performing decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm, to obtain the original content of the transaction data.

6. The method according to claim 1 or 2, wherein the fragmenting the proposed transaction data into the specified number of data fragments based on the erasure code

algorithm comprises:

performing encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key; and

5 fragmenting the encrypted transaction data into the specified number of data fragments based on the erasure code algorithm; and

wherein the performing data recovery on the received data fragments based on the erasure code reconstruction algorithm to obtain the original content of the transaction data includes:

10 performing data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the encrypted transaction data; and

performing decryption processing on the recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption key corresponding to the encryption key, to obtain the original content of the transaction data.

15 7. The method according to claim 6, wherein the encryption algorithm is a threshold encryption algorithm, the decryption algorithm is a threshold decryption algorithm corresponding to the threshold encryption algorithm, the decryption key is fragmented into a specified number of sub-keys, and each of the sub-keys is held by a respective node device, and

20 wherein the performing decryption processing on the recovered transaction data based on the decryption algorithm corresponding to the encryption algorithm and the decryption key corresponding to the encryption key includes:

collecting sub-keys held by the other node devices;

25 determining whether the number of collected sub-keys reaches a predetermined decryption threshold; and

in response to determining that the number of collected sub-keys reaches the predetermined decryption threshold, reconstructing the decryption key based on the collected sub-keys, and performing decryption processing on the recovered transaction data based on the threshold decryption algorithm corresponding to the threshold encryption algorithm and the decryption key.

30

8. The method according to claim 1, wherein the transaction data proposed by the primary node device is a transaction list constructed using to-be-negotiated transactions broadcast by user clients in a current consensus period of the primary node device, and the specified number is a total number of nodes in the blockchain network.

9. The method according to claim 1, wherein the blockchain network is a consortium blockchain network.

10. An apparatus for processing transaction consensus in a blockchain network, wherein nodes in the blockchain network include at least one primary node and a plurality of secondary nodes, and wherein the primary node fragments proposed transaction data into a specified number of data fragments based on an erasure code algorithm, and the apparatus comprising:

a receiving module, configured to receive a data fragment of the transaction data that is sent by the primary node in a unicast mode, wherein respective data fragments sent by the primary node to individual nodes in a unicast mode are different from one another;

a sending module, configured to broadcast the received data fragment to other nodes in the blockchain network, and receive data fragments of the transaction data that are broadcast by the other nodes;

a determining module, configured to determine whether the number of received data fragments of the transaction data reaches an erasure code recovery threshold; and

a recovery module, configured to: in response to determining that the number of received data fragments of the transaction data reaches the erasure code recovery threshold, perform data recovery on the received data fragments based on an erasure code reconstruction algorithm to obtain original content of the transaction data, to complete consensus processing with respect to the original content of the transaction data.

11. The apparatus according to claim 10, wherein the apparatus further comprises:

a fragmentation module, configured to determine whether a current node is

elected as the primary node; and in response to determining that the current node is elected as the primary node, fragment the proposed transaction data into the specified number of data fragments based on the erasure code algorithm; and

wherein the sending module is further configured to separately send the specified
5 number of data fragments to each of the other nodes in a unicast mode.

12. The apparatus according to claim 10, wherein a consensus algorithm equipped on the blockchain is a PBFT algorithm; and

wherein the receiving module is configured to:

receive a pre-prepare message sent by the primary node in a unicast mode,
10 wherein the pre-prepare message includes a data fragment of the transaction data; and
obtain and store the data fragment included in the pre-prepare message.

13. The apparatus according to claim 12, wherein the sending module is configured to:

broadcast a prepare message to the other nodes in the blockchain network,
15 wherein the prepare message includes the received data fragment, so that when receiving the prepare message, the other nodes obtain and store the data fragment included in the prepare message.

14. The apparatus according to claim 11, wherein the fragmentation module is configured to:

20 perform compression processing on the proposed transaction data based on a predetermined compression algorithm; and

fragment the compressed transaction data into the specified number of data fragments based on the erasure code algorithm; and

wherein the recovery module is configured to:

25 perform data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the compressed transaction data; and

perform decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm, to obtain the original content of the transaction data.

15. The apparatus according to claim 11, wherein the fragmentation module is configured to:

perform encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key; and

5 fragment the encrypted transaction data into the specified number of data fragments based on the erasure code algorithm; and

wherein the recovery module is configured to:

perform data recovery on the received data fragments based on the erasure code reconstruction algorithm, to obtain the encrypted transaction data; and

10 perform decryption processing on the recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption key corresponding to the encryption key, to obtain the original content of the transaction data.

16. The apparatus according to claim 15, wherein the encryption algorithm is a threshold encryption algorithm, the decryption algorithm is a threshold decryption algorithm corresponding to the threshold encryption algorithm, the decryption key is fragmented into a specified number of sub-keys, and each of the sub-keys is held by a respective node; and

wherein the recovery module is further configured to:

20 collect sub-keys held by the other node devices;

determine whether the number of collected sub-keys reaches a predetermined decryption threshold; and

in response to determining that the number of collected sub-keys reaches the predetermined decryption threshold, reconstruct the decryption key based on the collected sub-keys, and perform decryption processing on the recovered transaction data based on the threshold decryption algorithm corresponding to the threshold encryption algorithm and the decryption key.

25

17. The apparatus according to claim 10, wherein the transaction data proposed by the primary node is a transaction list constructed using to-be-negotiated

transactions broadcast by user clients in a current consensus period of the primary node, and the specified number is a total number of node devices in the blockchain network.

18. The apparatus according to claim 10, wherein the blockchain network is a
5 consortium blockchain network.

19. An electronic device, comprising:

a processor; and

a memory, configured to store machine-executable instructions, wherein

by reading and executing the machine-executable instructions stored in the
10 memory and corresponding to control logic of processing transaction consensus in a blockchain network, the processor is configured to:

receive a data fragment of transaction data that is sent by a primary node in
a unicast mode, wherein node devices in the blockchain network include at least one
primary node device and a plurality of secondary node devices, the primary node
15 device fragments the proposed transaction data into a specified number of data
fragments, and respective data fragments sent by the primary node to individual nodes
in a unicast mode are different from one another;

broadcast the received data fragment to other nodes in the blockchain
network, and receive data fragments of the transaction data that are broadcast by the
20 other nodes;

determine whether the number of received data fragments of the transaction
data reaches an erasure code recovery threshold; and

in response to determining that the number of received data fragments of the
transaction data reaches the erasure code recovery threshold, perform data recovery
25 on the received data fragments based on an erasure code reconstruction algorithm to
obtain original content of the transaction data, to complete consensus processing with
respect to the original content of the transaction data.