

TRANSACTION CONSENSUS PROCESSING METHOD AND APPARATUS FOR BLOCKCHAIN AND ELECTRONIC DEVICE

TECHNICAL FIELD

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

BACKGROUND

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

SUMMARY

[0003] The present specification provides a transaction consensus processing method for a blockchain. The blockchain includes a plurality of node devices, a target node device of the plurality of node devices that initiates a proposition performs
20 compression processing on proposed transaction data based on a predetermined compression algorithm, and fragments the compressed transaction data into a number of data fragments based on an erasure code algorithm, and the method includes: receiving a data fragment of the transaction data that is sent by the target node device in a unicast mode, data fragments sent by the target node device to node devices in the
25 unicast mode being different; 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; determining whether a 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 fragment based on an erasure code reconstruction algorithm, performing
5 decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm to obtain original content of the transaction data, and completing consensus processing on the obtained original content of the transaction data.

[0004] The present specification further provides a transaction consensus processing
10 apparatus for a blockchain. The blockchain includes a plurality of node devices, a target node device of the plurality of node devices that initiates a proposition performs compression processing on proposed transaction data based on a predetermined compression algorithm, and fragments the compressed transaction data into a number of data fragments based on an erasure code algorithm, and the apparatus includes: a
15 receiving module, configured to receive a data fragment of the transaction data that is sent by the target node device in a unicast mode, data fragments sent by the target node device to node devices in the unicast mode being different; 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
20 other node devices; a determining module, configured to determine whether a 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 fragment based on an
25 erasure code reconstruction algorithm, perform decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm to obtain original content of the transaction data, and complete consensus processing on the obtained original content of the transaction data.

[0005] The present specification further provides an electronic device, including: a
30 processor; and a memory, configured to store a machine-executable instruction, where by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: receive a data fragment of transaction data

that is sent by a target node device in a unicast mode, where the target node device that initiates a proposition performs compression processing on the proposed transaction data based on a predetermined compression algorithm, and fragments the compressed transaction data into a number of data fragments based on an erasure code algorithm, and data fragments sent by the target node device to node devices in the unicast mode are different; 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; determine whether a 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 on the received data fragment based on an erasure code reconstruction algorithm, perform decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm to obtain original content of the transaction data, and complete consensus processing on the obtained original content of the transaction data.

[0006] According to the above technical solutions, when proposing the to-be-added transaction data to the secondary node devices, the primary node device performs compression processing on the proposed transaction data based on the compression algorithm, and fragments the to-be-added transaction data based on the erasure code algorithm before transmission.

[0007] When proposing 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 a data fragment obtained after fragmenting the compressed transaction data to each secondary node device in the unicast mode. Therefore, data transmission bandwidth consumed for transmitting the to-be-added transaction data to the node devices participating in the consensus can be significantly reduced, and the to-be-added transaction data can be transmitted to the secondary node devices in a short time to complete the consensus, thereby improving consensus processing efficiency.

[0008] In addition, because the primary node device sends only a data fragment of the transaction data to each secondary node device in the unicast mode, the primary node device cannot learn of complete content of a complete transaction that the data

fragment belongs. Therefore, the following scenarios can be effectively avoided: the primary node device selectively sends some specific transactions to other node devices participating in the consensus in the unicast mode, and conduct a selective consensus on some specific transactions. As such, fairness of the consensus can be ensured, and
5 the primary node device can be prevented from doing "evil" through the selective consensus.

BRIEF DESCRIPTION OF DRAWINGS

[0009] FIG. 1 is a flowchart illustrating a transaction consensus processing method for a blockchain, according to an example implementation;

10 [0010] FIG. 2 is a schematic structural diagram illustrating an electronic device, according to an example implementation; and

[0011] FIG. 3 is a block diagram illustrating a transaction consensus processing apparatus for a blockchain, according to an example implementation.

DESCRIPTION OF IMPLEMENTATIONS

15 [0012] The present specification is intended to provide technical solutions that, in a consensus processing process of a blockchain, a compression algorithm and an erasure code algorithm are introduced to perform compression processing on to-be-added transaction data and fragment the compressed transaction data before transmission, to reduce data transmission bandwidth consumed for transmitting the to-be-added
20 transaction data to node devices participating in the consensus and improve consensus efficiency.

[0013] During implementation, first, in each round of consensus in the blockchain, a target node device (or primary node device) that initiates a proposition can perform compression processing on proposed transaction data in advance based on a
25 predetermined compression algorithm, fragment the compressed transaction data into a number of data fragments based on an erasure code algorithm, and then send the data fragments obtained through fragmentation to other node devices separately in a unicast mode. The data fragments sent by the primary node device to the other node device in the unicast mode are different.

[0014] Second, upon receiving a data fragment of the transaction data that is sent by the target node device in the unicast mode, each 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.

5 **[0015]** Finally, any node device 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 yes, it indicates that the currently collected data fragments are sufficient to recover the transaction data. In this case, data recovery can be performed on the
10 collected data fragments based on an erasure code reconstruction algorithm, decompression processing can be further performed on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm to obtain original content of the transaction data, and consensus processing on the obtained original content of the transaction data can be completed.

15 **[0016]** According to the above technical solutions, when proposing the to-be-added transaction data to the other node devices, the target node device performs compression processing on the proposed transaction data based on the compression algorithm, and fragments the to-be-added transaction data based on the erasure code algorithm before transmission.

20 **[0017]** When proposing 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 a data fragment obtained after fragmenting the compressed transaction data to each node device in the unicast mode. Therefore, data transmission bandwidth consumed for transmitting the to-be-added transaction data to
25 the node devices participating in the consensus can be significantly reduced, and the to-be-added transaction data can be transmitted to the secondary node devices in a short time to complete the consensus, thereby improving consensus processing efficiency.

[0018] In addition, because the target node device sends only a data fragment of the transaction data to each node device in the unicast mode, the target node device cannot
30 learn of complete content of a complete transaction that the data fragment belongs. Therefore, the following scenarios can be effectively avoided: the target node device selectively sends some specific transactions to other node devices participating in the consensus in the unicast mode, and conduct a selective consensus on some specific

transactions. As such, fairness of the consensus can be ensured, and the target node device can be prevented from doing "evil" through the selective consensus.

[0019] FIG. 1 illustrates a transaction consensus processing method for a blockchain, according to an implementation of the present specification. The method is applied to any node device in a blockchain network ("blockchain"). The blockchain includes a plurality of node devices, a target node device of the plurality of node devices that initiates a proposition performs compression processing on proposed transaction data based on a predetermined compression algorithm, and fragments the compressed transaction data into a number of data fragments based on an erasure code algorithm, and the method includes the following steps.

[0020] Step 102: Receive a data fragment of the transaction data that is sent by the target node device in a unicast mode, where data fragments sent by the target node device to node devices in the unicast mode are different.

[0021] 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.

[0022] Step 106: Determine whether a number of received data fragments of the transaction data reaches an erasure code recovery threshold.

[0023] Step 108: If yes, perform data recovery on the received data fragment based on an erasure code reconstruction algorithm, perform decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm to obtain original content of the transaction data, and complete consensus processing on the obtained original content of the transaction data.

[0024] The blockchain described in the present specification can specifically include a private blockchain network, a public blockchain network, a consortium blockchain network, and the like. This is not specifically limited in the present specification.

[0025] For example, in a scenario, the blockchain can be specifically a consortium blockchain network 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 network can deploy consortium blockchain network-based online services such as cross-border transfer and asset transfer online based on the consortium blockchain network.

[0026] 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 finally submitted to a distributed database of the blockchain.

[0027] Transactions in the blockchain include a transaction in a narrow sense and a transaction in a broad sense. The transaction in the narrow sense refers to a value transfer submitted 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. The transaction in the broad sense refers to service data with a service/business intention that is submitted by a user to the blockchain. For example, an operator can create a consortium blockchain network 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 a value transfer in the consortium blockchain network. In such a consortium blockchain network, the transaction can be a service message or a service request with a service/business intention that is submitted by a user to the consortium blockchain network.

[0028] It should be noted that a consensus algorithm of 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.

[0029] The Byzantine fault tolerance algorithm refers to a distributed fault tolerance algorithm, for example, a PBFT algorithm or a HoneyBadgerBFT algorithm, in which a Byzantine node (*e.g.*, 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 an evil node and a faulty node 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 a faulty node exists in the blockchain.

[0030] The target node device described in the present specification refers to a node device in the blockchain that has latest to-be-added block data to be proposed to all node devices participating in a transaction consensus.

[0031] For example, the consensus algorithm supported by the blockchain is the PBFT algorithm. In the PBFT protocol, only an elected primary node device has permission to propose transaction data to each secondary node device. Therefore, in this scenario, the target node device can be specifically an elected primary node device.

5 **[0032]** For another example, the consensus algorithm supported by the blockchain is the HoneyBadgerBFT algorithm. In the HoneyBadgerBFT protocol, all node devices in the blockchain have permission to propose transaction data to each other node device. Therefore, in this scenario, the target node device can be specifically any node device in the blockchain.

10 **[0033]** The transaction data proposed by the target node device and described in the present specification specifically refers to a to-be-added transaction that is collected by the target node device and submitted by a user client by using a connected node device.

[0034] In practice, the target node device can initiate a consensus process for a single transaction, or can initiate a consensus process for a created transaction list.

15 **[0035]** For example, in an implementation, a node device can broadcast, in the blockchain network, a transaction submitted by a user by using a user client. The primary node device can collect the to-be-added transactions broadcast by various node devices in the blockchain network in a consensus time period of a current round, then create a transaction list based on the collected to-be-added transactions, and transmit
20 the transaction list to other node devices as proposed transaction data.

[0036] The following provides descriptions by using an example that the blockchain is a consortium blockchain network, and the consortium blockchain network uses one or more of the PBFT algorithm and the HoneyBadgerBFT algorithm as the consensus algorithm to perform consensus processing on the transaction list proposed by the target
25 node device.

[0037] 1) The blockchain uses the HoneyBadgerBFT algorithm as the consensus algorithm.

[0038] The HoneyBadgerBFT consensus algorithm includes a reliable broadcast (RBC) subprotocol and a byzantine agreement (BA) subprotocol. The RBC subprotocol
30 is used to transmit transaction data (for example, a transaction list) proposed by a node device to other node devices, and receive transaction data proposed by the other node devices. The RBC subprotocol can ensure that at least $2f+1$ node devices can receive the same proposition when a total number of node devices in the blockchain is $N=3f+1$.

The BA subprotocol is mainly used to reach a consensus on a proposition initiated by a single node device. An input and an output of the BA subprotocol can only be 0 or 1. When the output is 1, it indicates that the transaction data proposed by the node device can enter a final consensus result.

5 **[0039]** In the present specification, based on an existing transaction data transmission mechanism of the RBC subprotocol of the HoneyBadgerBFT consensus algorithm, a compression and decompression mechanism is introduced to optimize and improve the existing transaction data transmission mechanism of the RBC subprotocol, so as to reduce data transmission bandwidth consumption for performing consensus
10 interaction between node devices in the blockchain using the HoneyBadgerBFT consensus algorithm and improve consensus processing efficiency.

[0040] In the consortium blockchain network using the HoneyBadgerBFT consensus algorithm, each node device can collect transactions submitted by clients and create a transaction list based on the collected transactions. In this case, the created transaction
15 list is transaction data that is proposed by the target node device/primary node device and requires consensus processing.

[0041] In the consortium blockchain network using the HoneyBadgerBFT consensus algorithm, each node device can initiate a consensus process to a transaction list proposed in the consortium blockchain network.

20 **[0042]** In the present specification, when the target node device in the consortium blockchain network proposes a created transaction list to the consortium blockchain network, the target node device can perform, based on a compression algorithm used by the consortium blockchain network, compression processing on the transaction list that needs to be proposed.

25 **[0043]** It should be noted that in the present specification, a specific algorithm type of the compression algorithm used by the consortium blockchain network is not specifically limited, and a person skilled in the art can flexibly select an algorithm based on an actual requirement in practice.

[0044] Further, the target node device can fragment the compressed transaction list
30 based on an erasure code algorithm, to fragment the transaction list into a number of data fragments.

[0045] 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 network. For example, assuming that there are N node devices in the consortium blockchain network, the transaction list can be fragmented into N data fragments.

5 [0046] 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.

10 [0047] 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 (*e.g.*, 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 (*e.g.*, an inverse algorithm of the erasure code algorithm).

15 [0048] In the present specification, after the target node device fragments the compressed transaction list based on the erasure code algorithm, based on the existing transaction data transmission mechanism of the RBC subprotocol, the target node device can send a VAL message to each other node device in the unicast mode to send the data fragments obtained through fragmentation to the other node devices separately. The VAL message is used by the node device that initiates the proposition to broadcast the transaction list to the other consensus nodes. The data fragments sent by the target node device to the other node devices need to be different.

20 [0049] It should be noted that, in addition to the data fragment that needs to be synchronized to each other node device, the VAL message can include check information used to perform validity and integrity check on the data fragment included in the VAL message.

25 [0050] In an illustrated implementation, after fragmenting the compressed transaction list based on the erasure code algorithm, the target node device can construct a Merkle tree based on hash values of the N data fragments obtained through fragmentation, and then add a hash value corresponding to a root node of the constructed Merkle tree, and hash values of neighboring nodes, on other branches, of a node of the Merkle tree that corresponds to a hash value of a data fragment to the VAL

message as the above check data.

[0051] For example, based on the RBC subprotocol, a format of the VAL message includes a View field, a RootHash field, a Branches field, and a BlockData field.

[0052] The View field is used to include a view number. The view number is usually
5 an integer varying consecutively, representing the number of rounds of successfully
completed consensus in the consortium blockchain network. The BlockData field is
used to include a data fragment, the RootHash field is used to include the hash value
corresponding to the root node of the Merkle tree, and the Branches field is used to
include the hash values of the neighboring nodes, on the other branches, of the node of
10 the Merkle tree that corresponds to the hash value of the data fragment.

[0053] In the present specification, a node device receiving the VAL message can
perform verification on the content included in the received VAL message according to
the RBC subprotocol.

[0054] For example, during implementation, the node device receiving the VAL
15 message can reconstruct a Merkle tree based on the hash value of the data fragment
included in the VAL message and the hash values of the neighboring nodes, on the other
branches, of the node of the Merkle tree that corresponds to the hash value of the data
fragment. Then, the node device can compare a hash value of a root node of the
reconstructed Merkle tree with the hash value of the root node of the Merkle tree
20 included in the VAL message to determine whether they are the same, to complete
verification on the data fragment. If the hash value corresponding to the root of the
reconstructed Merkle tree does not change, it is considered that the validity verification
on the data fragment succeeds. Otherwise, it can be considered that the data fragment
included in the VAL message is invalid, and in this case, the data fragment can be
25 discarded.

[0055] In the present specification, after the data fragment included in the VAL
message is verified, the node device receiving the VAL message can obtain and store
the data fragment included in the VAL message, and send the received data fragment
to other node devices, *e.g.*, by broadcasting an ECHO message to each other node
30 device based on the existing transaction data transmission mechanism of the RBC
subprotocol. The ECHO message is used to synchronize, by using a network
transmission method similar to Gossip, the data received by the node to each other node
device. A data format of the ECHO message is the same as that of the VAL message,

and the details are omitted for simplicity.

[0056] Correspondingly, a node device receiving the ECHO message can still perform verification on content included in the ECHO message in a similar way according to the RBC subprotocol. Detailed description of a specific verification process is omitted for simplicity.

[0057] After the data fragment included in the ECHO message is verified, the node device receiving the ECHO message can obtain and store the data fragment included in the ECHO message, and further determine whether the number of received ECHO messages broadcast by other node devices (including an ECHO message broadcast by the node device itself) reaches $2f+1$.

[0058] Herein, f represents the number of fault-tolerable nodes in the HoneyBadgerBFT algorithm, a specific value of f can be obtained through conversion according to an equation $N=3f+1$, and N represents the total number of node devices in the consortium blockchain network.

[0059] If the number of received ECHO messages reaches $2f+1$, the node device can further determine whether the number of collected and/or stored data fragments reaches an erasure code recovery threshold supported by the erasure code algorithm. If yes, 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.

[0060] 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.

[0061] In an illustrated implementation, if a specific value of the erasure code recovery threshold is larger than $2f+1$, the node device may not have sufficient data fragments to recover the original content of the to-be-added 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 $2f+1$.

[0062] 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 HoneyBadgerBFT algorithm, so that data fragments obtained by fragmenting the transaction list can be as small as possible while it is ensured that the node device has sufficient data fragments to recover the original content of the to-be-added transaction list.

[0063] In the present specification, after the node device performs data recovery calculation on the collected data fragments based on the erasure code algorithm to recover the original content of the transaction list, because a data fragment transmitted by each node device is a data fragment obtained after 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.

[0064] In this case, after completing the data recovery calculation, the node device can obtain the original content of the transaction list by performing decompression processing on a recovered compressed transaction list based on a decompression algorithm corresponding to the compression algorithm, and further perform validity and integrity check on the recovered transaction list.

[0065] For example, in practice, the ECHO message can also include, as check data, the hash value of the root node of the Merkle tree and the hash values of the neighboring nodes, on the other branches, of the node of the Merkle tree that corresponds to the hash value of the data fragment. After performing verification on the original content of the recovered transaction list, the node device still fragments the recovered transaction list based on the erasure code algorithm to obtain N data fragments, reconstructs a Merkle tree based on the N data fragments, and compares a hash value of a root node of the reconstructed Merkle tree with the hash value of the root node of the Merkle tree included in the ECHO message. If they are the same, it indicates that the validity and integrity check on the recovered transaction list succeeds, and the node device receives a valid and complete transaction list proposed by the target node device.

[0066] Further, after the validity and integrity check on the original content of the recovered transaction list succeeds, the node device can locally store the recovered transaction list, and broadcast a ready message to each other node device based on the existing transaction data transmission mechanism of the RBC subprotocol, to notify the other node devices that the node device receives a valid transaction list proposed by the

target node device.

[0067] In addition, after receiving a ready message, the node device can perform verification on the ready message, to determine whether a transaction list that is proposed by the target node device and received by another node device is exactly the same as the transaction list that is proposed by the target node device and received by the device.

[0068] For example, based on the RBC subprotocol, a format of the ready message includes a View field, a ProposeId field, and a RootHash field.

[0069] The View field is used to include a view number, the ProposeId field is used to include a sequence number of the node device that initiates the current proposition (e.g., a sequence number of the target node device), and the RootHash field is used to include a hash value corresponding to a root node of a Merkle tree of the current proposition.

[0070] After receiving the ready message, the node device can fragment the verified transaction list that is proposed by the target node device and locally stored, reconstruct a Merkle tree based on N data fragments obtained through fragmentation, and compare a hash value of a root node of the reconstructed Merkle tree with the hash value included in the ready message, to verify whether a transaction list that is proposed by the target node device and received by a node device sending the ready message is exactly the same as the transaction list that is proposed by the target node device and received by the device. If they are the same, the ready message is verified.

[0071] Further, after the received ready message is verified, the node device can determine whether the number of received ready messages broadcast by other node devices (including a ready message broadcast by the node device itself) reaches $2f+1$.

If the number of received ready messages reaches $2f+1$, it means that $2f+1$ nodes in the current consortium blockchain network all receive the same transaction list proposed by the target node device. In this case, a next phase of the HoneyBadgerBFT consensus algorithm can be entered, to perform consensus processing on the transaction list proposed by the target node device according to the BA subprotocol of the HoneyBadgerBFT consensus algorithm, and determine, based on an output result of the BA subprotocol, whether the transaction list proposed by the target node device can enter a final consensus result.

[0072] A specific process of performing consensus processing on the transaction list

proposed by the target node device according to the BA subprotocol is omitted for simplicity in the present specification.

[0073] (2) The blockchain uses the PBFT algorithm as the consensus algorithm.

[0074] In the present specification, based on three existing phases of pre-prepare, prepare, and commit in the PBFT algorithm, an erasure code algorithm is 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 consumption for performing consensus interaction between the primary node device and the secondary node devices and improve consensus processing efficiency.

[0075] In the present specification, before each round of consensus in the consortium blockchain network starts, one primary node device (*e.g.*, the target node device) can be elected from node devices in the consortium blockchain network.

[0076] For example, based on the PBFT algorithm, before each round of consensus in the consortium blockchain network starts, one primary node device can be calculated for the current round according to the following equation:

$P = v \bmod R$, where

P represents a calculated sequence number of the primary node device of the current round of consensus and v represents a current view number of the consortium blockchain network. The view number is usually an integer varying consecutively, representing the number of rounds of successfully completed consensus in the consortium blockchain network. 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 network, the view number can automatically increase by 1.

R represents a total number of node devices in the consortium blockchain network.

[0077] Each node device can perform the above calculation, and perform matching on the calculated sequence number of the primary node device and a sequence number of the device to determine whether it is elected as the primary node device.

[0078] If a node device determines that it is not elected as the primary node device, the node device can broadcast a received transaction submitted by a client to node devices in the consortium blockchain network.

[0079] If a node device determines that it 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.

[0080] In the present specification, when the primary node device proposes the created transaction list to other node devices, the primary node device can perform, based on a compression algorithm used by the consortium blockchain network, compression processing on the transaction list that needs to be proposed.

[0081] Further, the primary node device can fragment the proposed transaction list based on an erasure code algorithm, to fragment the transaction list into a number of data fragments.

[0082] When the primary node device completes fragmenting the transaction list based on the erasure code algorithm, the primary node device can send the data fragments obtained through fragmentation to other node devices separately, by sending a pre-prepare message to each secondary node device. In the present specification, the data fragments sent by the primary node device to the secondary node devices need to be different.

[0083] 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 network, to transmit the complete transaction list to secondary node devices that need to participate in a transaction consensus in the consortium blockchain network.

[0084] 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 network, 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 network is occupied to transmit the transaction data, and a relatively high requirement is imposed on bandwidth performance of the consortium blockchain network. Once the bandwidth performance of the consortium blockchain network is insufficient, the to-be-added transaction list cannot be transmitted to other secondary node devices participating in the consensus in a short time.

[0085] In view of this, in the present specification, the existing transaction data transmission mechanism in the pre-prepare and prepare phases of the PBFT algorithm

are optimized and improved.

[0086] The pre-prepare message sent by the primary node device to each secondary node device does not include the complete transaction list, but includes only one data fragment of the transaction list.

5 [0087] In addition, a method used by the primary node device to transmit the pre-prepare message to each other secondary node device is changed from broadcasting the pre-prepare message to all other nodes to sending a pre-prepare message to each secondary node device separately in the unicast mode. For example, pre-prepare messages can be constructed separately for ones of the secondary node devices, where
10 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 the secondary node devices receive different data fragments.

[0088] The secondary node device receiving the pre-prepare message can first perform verification on the received pre-prepare message according to the PBFT
15 algorithm, to determine whether to accept the received pre-prepare message.

[0089] In the present specification, performing verification on the received pre-prepare message is a process of performing verification on content included in the received pre-prepare message. Detailed description of a specific verification process is omitted in the present specification.

20 [0090] For example, according to the PBFT algorithm, the pre-prepare message can include information such as to-be-added transaction data (a data fragment in the present specification), a view number v , a digest of the to-be-added transaction data (a digest of the data fragment in the present specification), and a digital signature. Correspondingly, in the present specification, the following verification process can be
25 specifically used to perform verification on the pre-prepare message:

[0091] (1) Verify whether the view number v is consistent with a locally recorded view number.

[0092] (2) Perform verification on the digital signature in the pre-prepare message.

[0093] (3) Verify whether the data fragment included in the pre-prepare message is
30 valid.

[0094] 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 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
5 comparison on a hash value corresponding to a 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 considered that the validity verification on the data fragment succeeds. Otherwise, it can be considered that the data fragment included in the pre-prepare message is invalid, and in this case, the data
10 fragment can be discarded.

[0095] Certainly, in implementations, content included in the pre-prepare message can also be extended based on an actual requirement. For example, the pre-prepare message can further include a consensus height h . The consensus height h is functionally similar to the view number v , and is usually an integer varying
15 consecutively, representing the number of consensus rounds (instead of representing the number of rounds of successful consensus) in the consortium blockchain network. 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, a consensus height h can immediately increase by 1. Because this round of consensus has
20 not been successful, a view number v does not increase by 1, and remains unchanged at the original view number v .

[0096] In this case, for the verification on the pre-prepare message, the following verification can be performed besides the above verification process.

[0097] (4) Verify whether the consensus height h is consistent with a locally recorded
25 consensus height h .

[0098] 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
30 PBFT protocol, to broadcast a prepare message confirming the pre-prepare message to each other node device in the consortium blockchain network. A format of content included in the prepare message can be consistent with that of the pre-prepare message.

[0099] Based on the existing PBFT protocol, the prepare message broadcast by the

node device to each other node device in the consortium blockchain network usually includes only data such as a view number v and a digest of the to-be-added transaction data. In the present specification, the format of the prepare message can be further extended, and each node device can also add, to the prepare message, a data fragment sent by the primary node device by using the pre-prepare message in the unicast mode.

[0100] Each node device can broadcast a prepare message including a data fragment to each other node device in the consortium blockchain network, to further transmit the received data fragment sent by the primary node device in the unicast mode to each other node device in the consortium blockchain network.

[0101] In addition, 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.

[0102] If the verification succeeds, it indicates that the node device accepts the received prepare message.

[0103] The node device can obtain and store a data fragment included in the prepare message.

[0104] In addition, the node device can further determine whether the number of received prepare messages broadcast by other node devices (including a prepare message broadcast by the node device itself) reaches $2f+1$.

[0105] 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 an equation $N=3f+1$, and N represents the total number of node devices in the consortium blockchain network.

[0106] 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 other node device in the consortium blockchain network, and continue to complete the consensus processing process of the original content of the transaction list.

[0107] In the present specification, after broadcasting the commit message to each other node device in the consortium blockchain network, the node device in the

consortium blockchain network can also receive a commit message broadcast by each other node device, and perform verification on the received commit message to determine whether to accept the received commit message.

[0108] 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-added transaction data included in the pre-prepare message and the prepare message is a data digest 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.

[0109] 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.

[0110] 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 (including a commit message broadcast by the node device itself) reaches $2f+1$.

[0111] If the number of received commit messages reaches $2f+1$, the node device can further determine whether the number of collected and/or stored data fragments reaches an erasure code recovery threshold supported by the erasure code algorithm. If yes, 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.

[0112] 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.

[0113] In an illustrated implementation, if a specific value of the erasure code recovery threshold is larger than $2f+1$, the secondary node device may not have sufficient data fragments in the commit phase to recover the original content of the to-be-added 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 $2f+1$.

[0114] 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 as possible while it is ensured that the data fragments are sufficient to recover the original content of the to-be-added transaction list in the commit phase.

[0115] In the present specification, after the node device performs data recovery calculation on the collected data fragments based on the erasure code algorithm to recover the original content of the transaction list, because a data fragment transmitted by each node device is a data fragment obtained after 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.

[0116] In this case, after completing the data recovery calculation, the node device can obtain the original content of the transaction list by performing decompression processing on a recovered compressed transaction list based on a decompression algorithm corresponding to the compression algorithm.

[0117] In the above implementation, descriptions are provided by using an example that the consortium blockchain network uses each of the PBFT algorithm and the HoneyBadgerBFT algorithm as the consensus algorithm to perform consensus processing on the transaction list proposed by the target node device.

[0118] It should be further noted that in the present specification, both the target node device in the HoneyBadgerBFT consensus algorithm and the primary node device in the PBFT algorithm can further preprocess the transaction list based on a supported related algorithm before fragmenting the transaction list.

[0119] In another illustrated implementation, before the transaction list is fragmented, encryption preprocessing can be further performed on the transaction list, and encryption processing can be performed on the transaction list based on a predetermined encryption algorithm and an encryption key.

[0120] The encryption algorithm used 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.

[0121] After the encryption processing on the transaction list is completed, the encrypted transaction list is fragmented based on the erasure code algorithm, to fragment the encrypted transaction list into the number of data fragments.

[0122] Correspondingly, when the original transaction list is recovered based on the collected data fragments, 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.

[0123] In this case, after completing the data recovery calculation, the node device can obtain the original content of the transaction list by performing decryption processing on a recovered encrypted transaction list based on a decompression algorithm corresponding to the compression algorithm and a decryption key corresponding to the encryption key, to complete the consensus processing on the original content of the transaction list. Specific implementation details are omitted for simplicity.

[0124] 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.

[0125] In this scenario, the decryption key can be specifically fragmented into a number of sub-keys, and the sub-keys are held by node devices respectively.

[0126] 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 network. For example, assuming that there are N node devices in the consortium blockchain network, the decryption key can be fragmented into N sub-keys, which are held by the N node devices in the consortium blockchain network respectively.

[0127] 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 held sub-keys (e.g., a part of the decryption key).

[0128] 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.

[0129] For example, the consensus algorithm used in the consortium blockchain network 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.

5 [0130] For another example, the consensus algorithm used in the consortium blockchain network is still the HoneyBadgerBFT algorithm. The node device can add a sub-key held by the node device to an ECHO message exchanged with another node device.

10 [0131] Certainly, in practice, each node device can define an interactive message used to transmit a held sub-key, and synchronize the sub-key held by the node device to another node device by using the interactive message.

[0132] If the number of sub-keys collected by the node device reaches the predetermined decryption threshold, it indicates that the node device has sufficient 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
15 recover the original decryption key, and then obtain the original content of the transaction list by performing decryption processing on a recovered encrypted transaction list based on the threshold decryption algorithm and the recovered original decryption key.

20 [0133] 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.

[0134] 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
25 technology. A specific process is omitted for simplicity.

[0135] 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 to each node device can be effectively improved, and some unauthorized node devices cannot view the original content of the transaction list even
30 if they collect sufficient data fragments to recover the transaction list, thereby improving data security of consensus interaction between the node devices.

[0136] It should be further noted that, in the above implementation, detailed descriptions are provided by using the example that the existing transaction data

transmission mechanisms of the PBFT algorithm and the HoneyBadgerBFT algorithm are used to transmit the to-be-added transaction list to each node device participating in the consensus.

5 [0137] It should be emphasized that, in practice, when the to-be-added transaction list needs to be transmitted to each node device participating in the consensus, in addition to using the existing transmission mechanism in the consensus algorithm of the consortium blockchain network to transmit the transaction list, a separate transmission protocol can be defined to perform the same function.

10 [0138] In addition, in the above implementation, only the PBFT algorithm and the HoneyBadgerBFT algorithm are used as examples of the consensus algorithm in the consortium blockchain network for description. Clearly, in practice, the technical solutions of the present specification can also be applied to another form of consensus algorithm used in the consortium blockchain network.

15 [0139] 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 and the HoneyBadgerBFT algorithm. Specific implementation details are omitted 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.

20 [0140] Corresponding to the above method implementation, the present specification further provides an implementation of a transaction consensus processing apparatus for a blockchain. The implementation of the transaction consensus processing apparatus for the blockchain in the present specification can be applied to an electronic device. The apparatus implementation can be implemented by software, hardware, or a
25 combination of hardware and software. Software implementation is used as an example. As a logical apparatus, the apparatus is formed by reading a corresponding computer program instruction in a non-volatile memory to a memory by a processor of an electronic device where the apparatus is located. In terms of hardware, FIG. 2 is a hardware structural diagram illustrating an electronic device where the transaction
30 consensus processing apparatus for the blockchain is located in the present specification. In addition to a processor, a memory, a network interface, and a non-volatile memory shown in FIG. 2, the electronic device where the apparatus is located in the implementations can usually include other hardware based on an actual function of the

electronic device. Details are omitted for simplicity.

[0141] FIG. 3 is a block diagram illustrating a transaction consensus processing apparatus for a blockchain, according to an example implementation of the present specification.

5 [0142] Referring to FIG. 3, the transaction consensus processing apparatus 30 for the blockchain can be applied to the above electronic device shown in FIG. 2, and includes a receiving module 301, a sending module 302, a determining module 303, and a recovery module 304.

10 [0143] The receiving module 301 is configured to receive a data fragment of transaction data that is sent by a target node device in a unicast mode, where the target node device that initiates a proposition performs compression processing on the proposed transaction data based on a predetermined compression algorithm, and fragments the compressed transaction data into a number of data fragments based on an erasure code algorithm, and data fragments sent by the target node device to node
15 devices in the unicast mode are different; the sending module 302 is configured to broadcast the received data fragment to other node devices in a blockchain, and receive data fragments of the transaction data that are broadcast by the other node devices; the determining module 303 is configured to determine whether a number of received data fragments of the transaction data reaches an erasure code recovery threshold; and the
20 recovery module 304 is configured to: if yes, perform data recovery on the received data fragments based on an erasure code reconstruction algorithm to obtain original content of the transaction data, and complete consensus processing on the obtained original content of the transaction data.

[0144] In this implementation, the apparatus 30 further includes: a fragmentation
25 module 305 (not shown in FIG. 3), configured to generate transaction data proposed by the node device, and perform compression processing on the transaction data proposed by the node device based on the predetermined compression algorithm, and fragment the compressed transaction data into a number of data fragments based on the erasure code algorithm; and the sending module 302 is further configured to: send the number
30 of data fragments to other node devices separately in the unicast mode.

[0145] In this implementation, a consensus algorithm of the blockchain is a HoneyBadgerBFT algorithm; and the receiving module 301 is configured to: receive a VAL message sent by the target node device in the unicast mode, where the VAL

message includes the obtained data fragment of the transaction data; and obtain and store the data fragment in the VAL message.

5 [0146] In this implementation, a consensus algorithm of the blockchain is a HoneyBadgerBFT algorithm; and the sending module 302 is configured to: broadcast an ECHO message to the other node devices in the blockchain, where the ECHO message includes the received data fragment, so that upon receiving the ECHO message, the other node devices obtain and store the data fragment in the ECHO message.

10 [0147] In this implementation, a consensus algorithm of the blockchain is a PBFT algorithm; and the receiving module 301 is configured to: receive a pre-prepare message sent by the target node device in the unicast mode, where the pre-prepare message includes the obtained data fragment of the transaction data; and obtain and store the data fragment in the pre-prepare message.

15 [0148] In this implementation, a consensus algorithm of the blockchain is a PBFT algorithm; and the sending module 302 is configured to: broadcast a prepare message to the other node devices in the blockchain, where the prepare message includes the received data fragment, so that upon receiving the prepare message, the other node devices obtain and store the data fragment in the prepare message.

20 [0149] In this implementation, the fragmentation module 305 is configured to: perform encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key; and fragment the encrypted transaction data into the number of data fragments based on the erasure code algorithm; and the recovery module 304 is configured to: perform data recovery on the received data fragment based on the erasure code reconstruction algorithm; and obtain the original content of the transaction data by 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.

25 [0150] 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 number of sub-keys, and the sub-keys are held by node devices respectively; and the recovery module 304 is further configured to: collect sub-keys held by the other node devices; determine whether the number of collected sub-keys reaches a predetermined decryption threshold; and if yes, 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.

5 [0151] In this implementation, the transaction data proposed by the target node device is a transaction list, and the number is a total number of node devices in the blockchain.

[0152] In this implementation, the blockchain is a consortium blockchain network.

10 [0153] 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.

[0154] 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 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 ordinary skill in the art can understand and implement the implementations of the present specification without making innovative efforts.

20 [0155] 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 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.

25 [0156] Corresponding to the above method implementation, the present specification further provides an implementation of an electronic device. The electronic device includes a processor and a memory configured to store a machine-executable instruction. 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 device or component.

[0157] In this implementation, by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: receive a data fragment of transaction data that is sent by a target node device in a unicast mode, where
5 the target node device that initiates a proposition performs compression processing on the proposed transaction data based on a predetermined compression algorithm, and fragments the compressed transaction data into a number of data fragments based on an erasure code algorithm, and data fragments sent by the target node device to node devices in the unicast mode are different; broadcast the received data fragment to other
10 node devices in a blockchain, and receive data fragments of the transaction data that are broadcast by the other node devices; determine whether a number of received data fragments of the transaction data reaches an erasure code recovery threshold; and if yes, perform data recovery on the received data fragment based on an erasure code reconstruction algorithm, perform decompression processing on the recovered
15 transaction data based on a decompression algorithm corresponding to the compression algorithm to obtain original content of the transaction data, and complete consensus processing on the obtained original content of the transaction data.

[0158] In this implementation, by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction
20 consensus processing for a blockchain, the processor is configured to: generate transaction data proposed by a node device; perform compression processing on the transaction data proposed by the node device based on the predetermined compression algorithm, and fragment the compressed transaction data into a number of data fragments based on the erasure code algorithm; and send the number of data fragments
25 to other node devices separately in a unicast mode.

[0159] In this implementation, a consensus algorithm of the blockchain is a HoneyBadgerBFT algorithm; and by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: receive a VAL
30 message sent by the target node device in a unicast mode, where the VAL message includes the obtained data fragment of the transaction data; and obtain and store the data fragment in the VAL message.

[0160] In this implementation, a consensus algorithm of the blockchain is a

HoneyBadgerBFT algorithm; and by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: broadcast an ECHO message to the other node devices in the blockchain, where the ECHO message includes the received data fragment, so that upon receiving the ECHO message, the other node devices obtain and store the data fragment in the ECHO message.

[0161] In this implementation, a consensus algorithm of the blockchain is a PBFT algorithm; and by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: receive a pre-prepare message sent by the target node device in a unicast mode, where the pre-prepare message includes the obtained data fragment of the transaction data; and obtain and store the data fragment in the pre-prepare message.

[0162] In this implementation, by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: broadcast a prepare message to the other node devices in the blockchain, where the prepare message includes the received data fragment, so that upon receiving the prepare message, the other node devices obtain and store the data fragment in the prepare message.

[0163] In this implementation, by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: perform encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key; and fragment the encrypted transaction data into the number of data fragments based on the erasure code algorithm.

[0164] In this implementation, by reading and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: perform data recovery on the received data fragment based on the erasure code reconstruction algorithm; and obtain the original content of the transaction data by performing decryption processing on a recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption key corresponding to the encryption key.

[0165] 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 number of sub-keys, and the sub-keys are held by node devices respectively; and by reading
5 and executing the machine-executable instruction that is stored in the memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to: collect sub-keys held by the other node devices; determine whether the number of collected sub-keys reaches a predetermined decryption threshold; and if yes, reconstruct the decryption key based on the collected sub-keys, and perform
10 decryption processing on the recovered transaction data based on the threshold decryption algorithm corresponding to the threshold encryption algorithm and the decryption key.

[0166] A person skilled in the art can easily figure out other implementations of the present specification after considering the present specification and practicing the
15 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 common knowledge or conventional techniques that are not disclosed in the technical field of the present specification. The present specification and the implementations are
20 merely considered as examples, and the actual scope and the spirit of the present specification are pointed out by the following claims.

[0167] It should be understood that the present specification is not limited to the 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
25 the present specification. The scope of the present specification is limited by the appended claims only.

[0168] The above descriptions are merely preferred implementations of the present specification, but are not intended to limit the present specification. Any modification, equivalent replacement, or improvement made without departing from the spirit and
30 principle of the present specification shall fall within the protection scope of the present specification.

CLAIMS

1. A transaction consensus processing method for a blockchain, the blockchain including a plurality of nodes, wherein a target node of the plurality of nodes that initiates a proposition performs compression processing on proposed transaction data based on a predetermined compression algorithm, and fragments the compressed transaction data into a number of data fragments based on an erasure code algorithm, and wherein a consensus algorithm of the blockchain is a practical Byzantine fault tolerance, PBFT, algorithm, the method comprising:

receiving (102), by a node of the plurality of nodes, a data fragment of the number of data fragments of the transaction data that is sent by the target node in a unicast mode, including:

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

obtaining and storing, by the node, the data fragment included in the pre-prepare message, wherein the pre-prepare message sent by the target node to the node and each pre-prepare message sent by the target node to other nodes of the plurality of nodes include different data fragments of the transaction data;

broadcasting (104), by the node, the received data fragment to the other nodes of the plurality of nodes in the blockchain, and receiving, by the node, further data fragments of the transaction data that are broadcast by the other nodes; determining (106), by the node, whether a number of received data fragments of the transaction data, including the data fragment received (102) from the target node and the further data fragments received from the other nodes, reaches an erasure code recovery threshold of an erasure code reconstruction algorithm; and

in response to determining (106) that the number of received data fragments of the transaction data reaches the erasure code recovery threshold, performing (108), by the node, data recovery on the received data fragments based on the erasure code reconstruction algorithm to recover the transaction data, performing, by the node, decompression processing on the recovered transaction data based on a decompression algorithm corresponding to the compression algorithm to obtain original content of the transaction data, and completing, by the node, consensus processing on the obtained original content of the transaction data.

2. The method according to claim 1, further comprising:
generating, by the node, transaction data proposed by a further node of the plurality of nodes;
performing, by the node, compression processing on the transaction data proposed by the further node based on the predetermined compression algorithm, and fragmenting the compressed transaction data into a number of data fragments based on the erasure code algorithm; and
sending, by the node, the number of data fragments to the other nodes separately in the unicast mode.

3. The method according to claim 1, wherein the broadcasting (104) the received data fragment to the other nodes in the blockchain includes:
broadcasting a prepare message to the other nodes in the blockchain, wherein the prepare message includes the obtained data fragment, so that upon receiving the prepare message, the other nodes obtain and store the data fragment in the prepare message.

4. The method according to claim 1 or 2, wherein the fragmenting the proposed transaction data into the number of data fragments based on the erasure code algorithm includes:
performing encryption processing on the proposed transaction data based on a predetermined encryption algorithm and an encryption key; and
fragmenting the encrypted transaction data into the number of data fragments based on the erasure code algorithm; and
the performing data recovery on the received data fragment based on the erasure code reconstruction algorithm includes:
performing data recovery on the received data fragment based on the erasure code reconstruction algorithm; and
obtaining the original content of the transaction data by performing decryption processing on recovered transaction data based on a decryption algorithm corresponding to the encryption algorithm and a decryption key corresponding to the encryption key.

5. The method according to claim 4, 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 number of sub-keys, and the sub-keys are held by nodes respectively; and

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 other nodes;

determining whether a 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.

6. The method according to claim 1, wherein the transaction data proposed by the target node is a transaction list, and the number is a total number of nodes in the blockchain.

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

8. 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 memory and corresponds to control logic of transaction consensus processing for a blockchain, the processor is configured to perform a method according to any one of the preceding claims.