

TRANSACTION UPLOADING METHOD, ASSOCIATED APPARATUS, AND MEDIUM

RELATED APPLICATION

[0001] This application claims priority to Chinese Patent Application No. 2023103616550, filed to the China National Intellectual Property Administration on March 31, 2023 and entitled "TRANSACTION UPLOADING METHOD, ASSOCIATED APPARATUS, AND MEDIUM", which is incorporated herein by reference in its entirety.

FIELD OF THE TECHNOLOGY

[0002] This application relates to the field of blockchains, and more particularly, to a transaction uploading method, an associated apparatus, and a medium.

BACKGROUND OF THE DISCLOSURE

[0003] A conventional blockchain hierarchical consensus system includes a local consensus network and a global consensus network. Consensus nodes in the local consensus network perform a blockchain uploading operation for locally generated transactions, while the global consensus network performs a blockchain uploading operation for global transactions generalized from local transactions. The local consensus network and the global consensus network upload different blockchains, thus realizing hierarchical governance, and the global consensus network may focus on global governance.

[0004] In the hierarchical consensus system, although data are isolated between different local consensus networks, transactions are stored in plain text on the blockchain. Therefore, for target transactions not expected to be accessed by third parties on the blockchain, the conventional technology cannot meet uploading requirements of the target transactions, and the uploading security of the target transactions is poor.

SUMMARY

[0005] This embodiment of this application provides a transaction uploading method, an associated apparatus, and a medium.

[0006] According to an aspect of this application, a transaction uploading method is provided. The method is performed by a first relay node and includes:

- [0007]** determining, when a first consensus node receives a local transaction, a parent type of the local transaction;
- [0008]** transmitting the local transaction to a second consensus node when the parent type is a first type, the local transaction being used for instructing the second consensus node to execute the local transaction for recording the local transaction to a second blockchain, generating first uploading verification information of the local transaction, and transmitting the first uploading verification information of the local transaction to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result of the local transaction, and the second blockchain being not accessible by nodes other than the second consensus node; and
- [0009]** obtaining the uploading verification result of the local transaction from the third consensus node.
- [0010]** According to an aspect of this application, a transaction uploading method is provided. The method is performed by a second consensus node and includes:
- [0011]** receiving a local transaction from a first relay node, a parent type of the local transaction being a first type;
- [0012]** executing the local transaction, and recording the local transaction to a second blockchain, the second blockchain being not accessible by nodes other than the second consensus node;
- [0013]** obtaining first uploading verification information of the local transaction; and
- [0014]** transmitting the first uploading verification information to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result.
- [0015]** According to an aspect of this application, a transaction uploading apparatus is provided. The apparatus includes:
- [0016]** a first determining unit, configured to determine, when a first consensus node receives a local transaction, a parent type of the local transaction;
- [0017]** a first transmitting unit, configured to transmit the local transaction to a second consensus node when the parent type is a first type, the local transaction being used for instructing

the second consensus node to execute the local transaction for recording the local transaction to a second blockchain, generating first uploading verification information of the local transaction, and transmitting the first uploading verification information of the local transaction to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result of the local transaction, and the second blockchain being not accessible by nodes other than the second consensus node; and

[0018] a first obtaining unit, configured to obtain the uploading verification result of the local transaction from the third consensus node.

[0019] According to an aspect of this application, a transaction uploading apparatus is provided. The apparatus includes:

[0020] a first receiving unit, configured to receive a local transaction forwarded by a first relay node, a parent type of the local transaction being a first type;

[0021] a second uploading unit, configured to execute the local transaction, and record the local transaction to a second blockchain, the second blockchain being not accessible by nodes other than the second consensus node;

[0022] a second obtaining unit, configured to obtain first uploading verification information of the local transaction; and

[0023] a second transmitting unit, configured to transmit the first uploading verification information to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result.

[0024] According to an aspect of this application, an electronic device is provided. The electronic device includes a memory and one or more processors. The memory stores computer-readable instructions. The processor, when executing the computer-readable instructions, implements the transaction uploading method as described above.

[0025] According to an aspect of this application, one or more readable storage media are provided. The storage medium stores computer-readable instructions. The computer-readable instructions, when executed by one or more processors, implement the transaction uploading method as described above.

[0026] According to an aspect of this application, a computer program product is provided. The computer program product includes computer-readable instructions. The computer-readable

instructions are read and executed by one or more processors of a computer device, whereby the computer device performs the transaction uploading method as described above.

[0027] Details of one or more embodiments of this application are provided in the accompanying drawings and descriptions below. Other features, objectives, and advantages of this application become apparent from the specification, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] To describe the technical solutions of the embodiments of this application more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description show only some embodiments of this application, and those of ordinary skill in the art may still derive other drawings from these accompanying drawings without creative efforts.

[0029] FIG. 1A is an architectural schematic diagram of a hierarchical blockchain network in one embodiment.

[0030] FIG. 1B is an architectural schematic diagram of a hierarchical blockchain network in another embodiment.

[0031] FIG. 1C is an architectural schematic diagram of a hierarchical blockchain network in another embodiment.

[0032] FIG. 2 is a schematic diagram of a transaction uploading method applied in an electronic invoice scenario in one embodiment.

[0033] FIG. 3 is a flowchart of a transaction uploading method in one embodiment.

[0034] FIG. 4 shows a flowchart of packaging and transmitting using a queue in step 330a of FIG. 3.

[0035] FIG. 5 shows a schematic diagram of packaging and transmitting using queues to which different child types belong according to an embodiment of this application.

[0036] FIG. 6 shows a flowchart of packaging and transmitting using a priority and a queue in step 330a of FIG. 3.

[0037] FIG. 7 shows a schematic diagram of packaging and transmitting using a priority and a queue in FIG. 6.

[0038] FIG. 8 shows a schematic diagram of signature verification using a digest algorithm in steps 330a and 330b of FIG. 3.

- [0039]** FIG. 9 shows a schematic diagram of selecting a first global consensus network using a child type in step 330a of FIG. 3.
- [0040]** FIG. 10 shows an architectural diagram of a case where multiple first global consensus networks exist in a blockchain network according to an embodiment of this application.
- [0041]** FIG. 11 shows a schematic diagram of generating a first tree and verifying using the first tree in steps 330c and 330d of FIG. 3.
- [0042]** FIG. 12A shows a schematic diagram of a data structure of first uploading verification information in step 330c of FIG. 11.
- [0043]** FIG. 12B shows a schematic diagram of generating a first tree according to an embodiment of this application.
- [0044]** FIG. 12C shows a schematic diagram of a first bypass node in a first path of a first tree according to an embodiment of this application.
- [0045]** FIG. 13 shows a flowchart of generating a second tree and verifying using the second tree in steps 330c and 330d based on FIG. 11.
- [0046]** FIG. 14A shows a schematic diagram of a data structure of first uploading verification information in step 330c of FIG. 13.
- [0047]** FIG. 14B shows a schematic diagram of generating a second tree according to an embodiment of this application.
- [0048]** FIG. 14C shows a schematic diagram of a second bypass node in a second path of a second tree according to an embodiment of this application.
- [0049]** FIG. 15 shows a schematic diagram of a data structure of first uploading verification information with a first signature.
- [0050]** FIG. 16 shows a schematic diagram of a data structure of first uploading verification information with transaction desensitization information.
- [0051]** FIG. 17 shows a schematic diagram of a data structure of first uploading verification information with instantaneous-ness information.
- [0052]** FIG. 18 shows a schematic diagram of a data structure of an uploading verification result request.
- [0053]** FIG. 19 is a flowchart of a transaction uploading method in another embodiment.
- [0054]** FIG. 20 is a flowchart of a transaction uploading method in another embodiment.

[0055] FIG. 21 is a multi-terminal interaction diagram of a detailed embodiment of this application.

[0056] FIG. 22 is a flowchart of a transaction uploading method performed by a second consensus node in one embodiment.

[0057] FIG. 23 is a module diagram of a transaction uploading apparatus located on a first relay node in one embodiment.

[0058] FIG. 24 is a module diagram of a transaction uploading apparatus located on a second consensus node in one embodiment.

[0059] FIG. 25 is a structural diagram of a terminal implementing a transaction uploading method in one embodiment.

[0060] FIG. 26 is a structural diagram of a server implementing a transaction uploading method in one embodiment.

DESCRIPTION OF EMBODIMENTS

[0061] To make the objects, technical solutions, and advantages of this application clearer, the following further describes this application in detail with reference to the accompanying drawings and the embodiments. It is to be understood that specific embodiments described herein are merely illustrative of this application and are not intended to be limiting thereof.

[0062] Before the embodiments of this application are further described in detail, nouns and terms in the embodiments of this application are described, and the nouns and terms in the embodiments of this application are applicable to the following explanations.

[0063] Hierarchical blockchain: A consensus network and a witness network are separated based on point-to-point transmission in a blockchain network (as shown in FIG. 1A). A consensus node of the consensus network mainly performs a consensus algorithm. The witness network is also referred to as a service network. A witness node of the witness network performs data clearing synchronization, and complete relevant specific services based on the witness node.

[0064] Sorting synchronization: After the consensus node receives a data synchronization request of the witness node, data permitted to be read by the witness node is synchronized to the witness node.

[0065] Hierarchical consensus system of hierarchical blockchain: The hierarchical blockchain divides the consensus network into a local consensus network and a global consensus network based on separating the witness network from the consensus network. The consensus node

in the local consensus network performs a blockchain uploading operation for locally generated transactions (for example, transactions generated by subsidiaries of a certain group), while the global consensus network performs a blockchain uploading operation for global transactions (for example, group transactions summarized from subsidiary transactions) generalized from local transactions. The local consensus network and the global consensus network upload different blockchains, thus realizing hierarchical governance. Accordingly, the global consensus network is not required to process a large number of local transactions, and may focus on global governance.

[0066] In the hierarchical consensus system, although data isolation is realized between different local consensus networks, transactions are stored in plain text on the blockchain. Some transactions are not expected to be exposed to third parties for a certain need, and therefore, the conventional technology cannot meet uploading requirements of target transactions not expected to be exposed to third parties. The uploading security of the target transactions is poor. Moreover, the hierarchical consensus system is relatively large, and each local consensus network produces many target transactions. Each local consensus network handles the uploading security of target transactions separately, which has the problems of high overhead and difficulty. Therefore, there is an urgent need for a technology that can improve the uploading security of target transactions and has more universality.

[0067] Descriptions of Applied System Architecture and Scenario of this Embodiment of this Application

[0068] The blockchain network shown in FIG. 1A includes a witness network and a consensus network. The consensus network refers to a network that uploads a to-be-uploaded transaction on a blockchain after reaching consensus, and includes multiple consensus nodes. Multiple means at least two. The witness network refers to a network that generates a to-be-uploaded transaction, does not upload the transaction on a blockchain, but witnesses uploading of the to-be-uploaded transaction, and includes multiple witness nodes. The consensus nodes are blockchain nodes.

[0069] The witness nodes do not need to participate in uploading consensus, and are mainly configured to generate the to-be-uploaded transaction and witness the uploading of the to-be-uploaded transaction. The consensus nodes are generally full nodes. The full nodes are nodes for maintaining all data of each block on the blockchain. Since the consensus nodes are to upload the to-be-uploaded transaction after generating blocks, the data of all blocks on the whole blockchain are definitely required to be maintained. The witness nodes are generally lightweight nodes. The lightweight nodes refer to nodes for maintaining part of the data of each block on the

blockchain. For example, after the full nodes complete the uploading, only block headers of uploaded blocks are transmitted to the lightweight nodes, and when the lightweight nodes need to verify whether the uploading is correct, digest values in the block headers are used for verifying Merkle trees.

[0070] The consensus node or the blockchain node may be a server in the blockchain network or a user terminal in the blockchain network. The specific form of the consensus node or the blockchain node is not limited herein. The witness network and the consensus network shown in FIG. 1A may be in different network environments. For example, in many implementations, the witness nodes are deployed in a public witness network, while the consensus nodes running a blockchain consensus protocol are deployed in a private consensus network. The witness nodes and the consensus nodes may interact through routing nodes, and the witness nodes and the consensus nodes may also directly transmit data without the routing nodes. Since the consensus network is in a relatively secure private cloud, a consensus mechanism is provided for mutual access to ensure security, and there is no need to add identity management and network control. The witness nodes are in the public network and may be accessed by other uncertain network terminals. Therefore, the behavior of the witness nodes and other possible nodes accessing the consensus network needs to be strictly controlled.

[0071] FIG. 1A shows an architecture diagram including a witness network and a consensus network. Next, taking FIG. 1B as an example, the architecture of hierarchical consensus using the consensus network based on FIG. 1A continues to be described.

[0072] Reference is made to a blockchain hierarchical consensus network shown in FIG. 1B. The blockchain hierarchical consensus network refers to a blockchain network for hierarchical consensus on transactions. The blockchain hierarchical consensus network includes a witness network, a local consensus network, a local network-level witness node, and a global consensus network. The witness network is the witness network in FIG. 1A. The local consensus network refers to a network for consensus and uploading of local transactions. The local consensus network includes multiple local consensus nodes. The local transactions are transactions that has impact within the local scope. For example, electronic invoice reimbursement transactions of a subsidiary of a certain group are the local transactions. The global consensus network is a network for consensus and uploading of global transactions derived from the local transactions. The global consensus network includes multiple global consensus nodes. The global transactions are transactions that impact transaction of a global scope. For example, electronic invoice reimbursement summarizing transactions of a subsidiary of a group are the global transactions. The

local network-level witness node is an interface node between the local consensus node and the global consensus node, and is configured to summarize the local transactions generated by the local consensus network into the global transactions, transmit the global transactions to the global consensus network for uploading through a global consensus network entry point, and witness the uploading of the global consensus network. There are two local network-level witness nodes in FIG. 1B, which play a double backup role. The two local network-level witness nodes work at the same time and perform the same actions. If one of the local network-level witness nodes fails, the loss of information and the interruption of workflow will not be caused.

[0073] The witness nodes in the witness network do not verify the uploading is correct in the global consensus network, but only verify whether the uploading is correct in the local consensus network. When it is required to verify whether the local transactions generated thereby are correctly uploaded by the local consensus node, data needed for verification is obtained from the local consensus node by the clearing synchronization as described above, and the local consensus node will not transmit all the uploading data thereto. However, the local network-level witness node only verifies the uploading correctness in the global consensus network. When it is required to verify whether the global transactions deduced from the local transactions are correctly uploaded by the global consensus node, data needed for verification is obtained from the global consensus node by the clearing synchronization as described above, and the global consensus node will not transmit all the uploading data thereto.

[0074] The architecture of FIG. 1C differs from that of FIG. 1B in that the two local network-level witness nodes of FIG. 1B are replaced with a first relay node, the global consensus network is further divided into a second global consensus network and a first global consensus network, and a second relay node is adopted between the second global consensus network and the first global consensus network. The first global consensus network refers to a blockchain network for uploading a target transaction, and records the target transaction to a second blockchain. The target transaction is a transaction not expected to be accessed and known by a third party. The second blockchain is used for specifically recording blocks of transactions that are not expected to be accessed and known by the third party, and the second blockchain is not expected to be accessed by the outside world. Both the third party and the outside world refer to blockchain nodes other than the second consensus node. For example, the third party and the outside world include witness nodes, first consensus nodes, third consensus nodes, first relay nodes, and second relay nodes. If it is expected to obtain verification information recorded to a block on the blockchain, the verification information can only be obtained by requesting a blockchain node in the first global consensus network recording the blockchain.

[0075] Since the target transaction is a transaction not expected to be accessed and known by a third party, the consent of an object that generates the target transaction shall be sought before the target transaction is uploaded. Specifically, when the object generates the target transaction and transmits the target transaction to the local consensus network for uploading, an inquiry box may be displayed to the object to inquire whether the object agrees to upload the target transaction. If the object does not agree, this implementation cannot be implemented, or only the part of the transaction agreed to be uploaded is uploaded.

[0076] The first consensus node of FIG. 1C differs from the local consensus node of FIG. 1B in that the first consensus node only uploads a local transaction of a second type. The first relay node delegates the second consensus node in the first global consensus network to upload a local transaction of a first type. The first type is a transaction type not expected to be accessed and known by a third party. The second type is a transaction type allowed to be accessed and known by a third party. The local transaction of the second type is uploaded to a blockchain that may be viewed by any external node, while the local transaction of the first type is uploaded to a blockchain that is only visible inside the first global consensus network, thereby improving the uploading security of the target transaction.

[0077] The second global consensus network of the global consensus network of FIG. 1C is similar to the global consensus network of FIG. 1B, and is mainly configured to upload the global transactions deduced from the local transactions. The second global consensus network includes multiple third consensus nodes. The first global consensus network in the global consensus network of FIG. 1C is an added network relative to FIG. 1B. The first global consensus network is used for completing uploading of the local transaction of the first type. The first global consensus network includes multiple second consensus nodes. When the first global consensus network uploads the local transaction of the first type, the blockchain for uploading cannot be viewed by the outside world. To enable external objects to verify whether the uploading is correct, the first global consensus network may generate uploading verification information to be transmitted to the second global consensus network. The second global consensus network verifies the uploading verification information. An uploading verification result is returned to the local consensus network through the first relay node. The first relay node is a node for transferring the local transaction of the first type to the first global consensus network for uploading. The second relay node is a node for transmitting uploading verification information generated by the first global consensus network to the second global consensus network for verification.

[0078] This embodiment of this application may be applied to various scenarios, including but not limited to electronic invoices, equity registration, securities trading, and other scenarios, such as the scenario of electronic tax invoice blockchain system shown in FIG. 2.

[0079] Referring to FIG. 2, the electronic tax invoice blockchain system includes a service layer, a routing agent layer, and a consensus network layer.

[0080] The service layer is a witness network, including at least one witness node. The service layer may receive local transaction requests from the electronic tax bureau, enterprises, or consumers, such as an electronic invoice issuing transaction uploading request of the electronic tax bureau, electronic invoice receiving transaction uploading requests of the enterprises, and electronic invoice uploading requests of the consumers. The service layer includes a server of a local tax bureau in a private tax network, an invoicing server in a public cloud (server for invoicing electronic invoices of non-critical enterprises), a reimbursement server (server for reimbursement of electronic invoices of non-critical enterprises), a key enterprise server (server for various transactions of electronic invoices of critical enterprises), a payment server in a private cloud (server for invoicing related transactions of consumers in the payment process of non-critical enterprises), a circulation server (server for invoicing related transactions of items in the circulation process of non-critical enterprises), and a critical enterprise server (server for invoicing related transactions of consumers in critical enterprises). The critical enterprises are enterprises which have a certain size and therefore needs to be provided with invoice-related transactions by a dedicated server. The electronic tax bureau is configured to provide online tax services. The electronic tax invoices refer to electronic invoices issued by the electronic tax bureau. The electronic invoices refer to invoices generated, stored, transmitted, and managed electronically. The private tax network refers to a network serving taxpayers, and the local tax bureau is a government agency in charge of local taxation.

[0081] The server of the local tax bureau in the private tax network generates an electronic invoice issuing transaction according to an electronic invoice issuing demand of the electronic tax bureau, and requests the consensus network layer to upload the transaction. The invoicing server in the public cloud generates an electronic invoice collection transaction according to the demands of non-critical enterprises to apply for electronic invoices, and requests the consensus network layer to upload the transaction. The reimbursement server in the public cloud generates an electronic invoice reimbursement transaction according to the demand of non-critical enterprises for reimbursement of electronic invoices, and requests the consensus network layer to upload the transaction. The critical enterprise server in the public cloud generates an electronic invoice collection or reimbursement

transaction according to the demand of critical enterprises to collect or reimburse electronic invoices, and requests the consensus network layer to upload the transaction. The payment server in the private cloud generates a transaction requiring electronic invoice according to the payment actions of consumers in non-critical enterprises, and requests the consensus network layer to upload the transaction. The circulation server in the private cloud generates an electronic invoice circulation transaction according to the demand of exchanging items between consumers of non-critical enterprises, and requests the consensus network layer to upload the transaction. The critical enterprise server in the private cloud generates a transaction requiring invoicing or circulation according to the payment or item circulation requirements of consumers of critical enterprises, and requests the consensus network layer to upload the transaction.

[0082] The various servers need the consent of the object when generating transactions and requesting the consensus network layer for uploading according to the various requirements of the object. Specifically, when a transaction is generated and transmitted to the consensus network layer for uploading, an inquiry box may be displayed to the object to inquire whether the object agrees to upload the generated transaction. If the object does not agree, this implementation cannot be implemented, or only the part of the transaction agreed to be uploaded is uploaded.

[0083] The routing agent layer refers to a layer that plays the role of routing between nodes. The routing agent layer is configured to route data related to the service layer and the consensus network layer. The routing agent layer includes an authentication service, a certificate cache, a routing service, and a point-to-point service. The routing service is configured to route a to-be-uploaded transaction generated by the service layer to the consensus network layer. The authentication service is configured to authenticate the to-be-uploaded transaction before routing the to-be-uploaded transaction to the consensus network layer. For example, a signature of the to-be-uploaded transaction is verified according to a key certificate. The certificate cache is dedicated to caching public and private key certificates required in the authentication service. The point-to-point service provides peer-to-peer communication between nodes. After the to-be-uploaded transaction is uploaded through the consensus network layer, the uploading verification information returned by the consensus network layer is received through the point-to-point service.

[0084] The consensus network layer includes a local consensus network as in FIG. 1C, a first global consensus network, and a first relay node located between. The local consensus network includes multiple first consensus nodes. Each first consensus node includes a permission contract for determining whether the first consensus node has an uploading permission, a memory used in

the uploading processing of the local transaction on the blockchain, and a block for locally maintaining the uploading thereof. The first global consensus network includes multiple second consensus nodes. Each second consensus node includes a permission contract for determining whether the first consensus node has an uploading permission, a memory used in the uploading processing of the target transaction, and a block for locally maintaining the uploading thereof.

[0085] In FIG. 2, the consensus network layer has three branches. If an electronic invoice issuing transaction uploading request forwarded by an agent node and generated by the local tax bureau server is received, the request is forwarded to the first consensus node of Branch 1. The first consensus node queries an own permission contract and finds that the first consensus node is not permitted to upload in the local transaction blockchain (because the issuance of electronic invoices is a transaction not expected to be known by a third party and the transaction must be transferred to the first global consensus network for uploading to the second blockchain). Therefore, the request is forwarded to the second consensus node in Branch 1 through the first relay node. If the electronic invoice and reimbursement transaction uploading request forwarded by the agent node and generated by various servers of the public cloud is received, the request is forwarded to the first consensus node of Branch 2. The first consensus node queries an own permission contract and finds that the first consensus node is not permitted to keep accounts in the local transaction blockchain (because the issuance and reimbursement of electronic invoices is a transaction not expected to be known by a third party and the transaction must be transferred to the first global consensus network for uploading to the second blockchain). Therefore, the request is forwarded to the second consensus node in Branch 2 through the first relay node. If an electronic invoice issuing and circulation transaction uploading request forwarded by an agent node and generated by various servers of the private cloud is received, the request is forwarded to the first consensus node of Branch 3. The first consensus node queries an own permission contract and finds that the first consensus node is not permitted to keep accounts in the local transaction blockchain (because the issuance of electronic invoices and the circulation of electronic invoices are transactions not expected to be known by a third party and the transaction must be transferred to the first global consensus network for uploading to the second blockchain). Therefore, the request is forwarded to the second consensus node in Branch 3 through the first relay node. After the second consensus node of Branches 1-3 records the target transaction to the second blockchain, the second blockchain is unavailable to the outside world, thus improving the uploading security of the target transaction.

[0086] General Description of this Embodiment of this Application

[0087] According to one embodiment of this application, a transaction uploading method is provided.

[0088] The transaction uploading method is generally applied to service scenarios where transaction data is not expected to be known to the outside world, such as a scenario of electronic tax invoice blockchain system shown in FIG. 2. Transactions not expected to be known to the outside world are referred to as target transactions. This embodiment of this application provides a solution for uniformly executing and recording a local transaction of a first type, which can improve the uploading security of a target transaction. Also, to ensure the true uploading of the target transaction, a verification mechanism is further provided. The transaction uploading method is performed by a first relay node in a blockchain hierarchical consensus network.

[0089] As shown in FIG. 3, the transaction uploading method according to one embodiment of this application may include:

[0090] Step 310: Determine, when a first consensus node receives a local transaction, a parent type of the local transaction.

[0091] Step 320: Instruct, when the parent type of the local transaction is a second type, the first consensus node to execute the local transaction and to record the local transaction to a first blockchain, the first blockchain being accessible by nodes other than the first consensus node.

[0092] The nodes other than the first consensus node refer to nodes in a blockchain system other than nodes in a local consensus network, including witness nodes, first relay nodes, second relay nodes, nodes in the first global consensus network, and nodes in the second global consensus network.

[0093] In step 320, since the first blockchain is allowed to be accessed by nodes other than the first consensus node, the first consensus node is instructed to execute the local transaction and record the local transaction to the first blockchain when the parent type of the local transaction is a second type, whereby the local transaction may be viewed by any external node.

[0094] Step 330: Transmit the local transaction to a second consensus node when the parent type of the local transaction is a first type, the local transaction being used for instructing the second consensus node to execute the local transaction for recording the local transaction to a second blockchain, generating first uploading verification information of the local transaction, and transmitting the first uploading verification information of the local transaction to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification

information to obtain an uploading verification result of the local transaction, and the second blockchain being not accessible by nodes other than the second consensus node.

[0095] Step 340: Obtain the uploading verification result of the local transaction from the third consensus node.

[0096] Steps 310 to 340 are described in detail below.

[0097] In step 310, the first relay node determines a parent type of a local transaction received by the first consensus node in the local consensus network. The parent type is the first type or the second type. The first type refers to a type of a transaction not allowed to be known to the outside world. The second type refers to a type of a transaction allowed to be known to the outside world.

[0098] The first relay node is a module for determining the parent type of the local transaction and forwarding the local transaction of the first type. The first relay node may be a separate device such as a separate computer or server, or a part of the separate device such as a virtual machine partitioned on the server. The local transaction of which the parent type is the first type is referred to in this application as a target transaction.

[0099] The first consensus node is located in the local consensus network and is a module for receiving the local transaction, executing the local transaction of the second type, and recording the local transaction to the first blockchain. The first blockchain records the local transaction and can be accessed by nodes other than the first consensus node. The first consensus node may be a separate device such as a separate computer or server, or a part of the separate device such as a virtual machine partitioned on the server.

[00100] Step 310 may be divided into an interactive process in which the first consensus node receives the local transaction, the first relay node discovers the local transaction, and the first relay node determines the parent type of the local transaction. The local transaction refers to a local transaction required to be executed by the consensus node and recorded in the blockchain, such as an invoice reimbursement transaction of an enterprise. The parent type of the local transaction is used for indicating whether the local transaction is allowed to be known by a third party. The parent type of the local transaction may be one of the first type (for example, the type of the invoice reimbursement transaction is the first type) or the second type (for example, a weather information transaction). The local transaction of the first type is required to be executed by a consensus node (for example, a second consensus node below) and recorded to a blockchain inaccessible to the outside world. The local transaction of the second type is required to be executed by a consensus

node (for example, the first consensus node) and recorded to a blockchain accessible by the outside world.

[00101] In some embodiments, the setting of a determination period of the parent type of the local transaction includes the following cases:

[00102] (1) After it is found that the first consensus node receives a local transaction, a parent type of the local transaction is immediately determined.

[00103] (2) According to a determination period preset by the first relay node, after it is found that a local transaction exists in the first consensus node and when the determination period has passed, a parent type of the local transaction is immediately determined. For example, the determination period preset by the first relay node is 1 hour. If it is found that the time when the first consensus node receives the local transaction is 7:00 am on a day, the parent type of the local transaction is determined at 8:00 am on the same day.

[00104] In some embodiments, the manner in which the parent type of the local transaction is determined includes, but is not limited to, the following manners:

[00105] (1) After it is found that the first consensus node receives a local transaction, a first key field is extracted from the content of the local transaction, and a field-to-parent type comparison table is searched according to the first key field to determine a parent type of the local transaction. The following is an example of the field-to-parent type comparison table:

Table 1

First key field	Parent type
Invoice	First type
Resource	First type
Weather	Second type

[00106] As shown in Table 1, when the first key field is invoice, the local transaction is not allowed to be exposed to the outside world. Therefore, it is determined that the parent type of the local transaction is the first type. When the first key field is resource, the local transaction is also not allowed to be exposed to the outside world. Therefore, it is determined that the parent type of the local transaction is the first type. When the first key field is weather, the local transaction is allowed to be exposed to the outside world. Therefore, it is determined that the parent type of the local transaction is the second type.

[00107] For example, the content of a local transaction includes "Reimbursement of electronic invoice 0001: amount 500 for accommodation", which includes the first key field of "Invoice". Therefore, it is determined that the parent type of the local transaction is the first type. For example, the local transaction is "100 first resources are converted into 10 second resources", which includes the first key field of "Resource". Therefore, it is determined that the parent type of the local transaction is the first type. For another example, the local transaction is "Weather 10001: sunny on a certain day", which includes the first key field of "Weather". Therefore, it is determined that the parent type of the local transaction is the second type.

[00108] (2) After it is found that the first consensus node receives a local transaction, a type identifier is obtained from the content of the local transaction, and a parent type of the local transaction is determined according to the type identifier.

[00109] The type identifier is divided into two cases. In one case, the type identifier represents the first type by characters. For example, "1", as the type identifier, represents the first type, and "0", as the type identifier, represents the second type. In the other case, the first type or the second type is directly used as the type identifier.

[00110] For example, the content of a local transaction includes "Reimbursement of electronic invoice 0001: amount 500 for accommodation, [first type]". It is determined, according to "[first type]" included in the content, that the parent type of the local transaction is the first type. For another example, the content of a local transaction includes "Reimbursement of electronic invoice 0001: amount 500 for accommodation, [1]". It is determined, according to "[1]" included in the content, that the parent type of the local transaction is the first type.

[00111] After step 310 determines the parent type of the local transaction, there are two cases. In the first case, if it is determined that the parent type of the local transaction is the second type, step 320 is performed. In the second case, if it is determined that the parent type of the local transaction is the first type, step 330 is performed.

[00112] The first case generated after step 310 is described first.

[00113] In step 320, in response to determining that the parent type of the local transaction is the second type, the local transaction is executed by the first consensus node and recorded to the first blockchain.

[00114] A common service smart contract (such as an order smart contract below) runs on the first consensus node, and the service logic is publicly reflected in contract code. A local transaction interacting with the common service smart contract is executed on the first consensus

node, and execution results are also recorded on the first blockchain maintained by the first consensus node. Furthermore, these execution results may also be transmitted back to the second global consensus network in the global consensus network through the first relay node.

[00115] For example, it is assumed that the local transaction of the second type is "Order 10001: purchase 50 office supplies". Then the first consensus node invokes the order smart contract to execute the local transaction, and the execution result is "Completed: order 10001: purchase 50 office supplies". The execution result is recorded on the first blockchain by the first consensus node, and all nodes other than the first consensus node may access the first blockchain.

[00116] The specific process of recording the local transaction to the first blockchain mentioned above is as follows:

[00117] Generally, the witness node generates a local transaction based on an operation behavior of an object, and transmits the local transaction and a digital signature of the witness node to the neighboring first consensus node on the local consensus network, which is convenient for the first consensus node to verify the authenticity of the local transaction. During verification, the first consensus node verifies the digital signature by using a public key of the witness node to obtain a first digest value of the local transaction, performs digest calculation on the local transaction according to a digest algorithm to obtain a second digest value of the local transaction, and compares the first digest value with the second digest value. If the first digest value and the second digest value are the same, the verification is successful, and the local transaction is a real transaction generated by the witness node.

[00118] After the verification is successful, the local transaction and other local transactions transmitted by the witness node are accumulated. When the accumulation reaches a certain degree (for example, the number of accumulated transactions reaches a certain threshold, or the accumulation time reaches a certain threshold), these local transactions are packaged into a block and transmitted to other first consensus nodes in the local consensus network for consensus. After the consensus is successful, each first consensus node records the block to the first blockchain maintained by the first consensus node.

[00119] The second case generated after step 310 is described next.

[00120] With continued reference to FIG. 3, step 330 may be divided into the following steps:

[00121] Step 330a (performed by the first relay node): Transmit the local transaction to the second consensus node in the first global consensus network in response to the parent type of the local transaction being the first type.

[00122] Step 330b (performed by the second consensus node): Execute the local transaction, and record the local transaction to the second blockchain.

[00123] Step 330c (performed by the second consensus node): Transmit first uploading verification information of the local transaction to the third consensus node in the second global consensus network.

[00124] Step 330d (performed by the third consensus node): Verify uploading of the local transaction according to the first uploading verification information.

[00125] Steps 330a to 330d are described in detail below.

[00126] In step 330a, after the first relay node determines that the parent type of the local transaction is the first type, since the target transaction is a transaction not expected to be accessed and known by a third party, the consent of an object that generates the target transaction shall be sought before the target transaction is uploaded. Specifically, an inquiry box may be displayed to the object to inquire whether the object agrees to upload the target transaction. If the object does not agree, this implementation cannot be implemented, or only the part of the transaction agreed to be uploaded is uploaded.

[00127] Then, the local transaction is transmitted to the second consensus node in the first global consensus network.

[00128] After the first relay node determines that the parent type of the local transaction is the first type, the local transaction is required to be executed and recorded to the second blockchain inaccessible to the outside world. Therefore, the local transaction is required to be executed by the second consensus node.

[00129] The second consensus node is located in the first global consensus network and is a module for receiving and executing the local transaction of the first type, and recording the local transaction to the second blockchain. The second blockchain is not accessible by nodes other than the second consensus node, thus ensuring the uploading security of the target transaction. The second consensus node may be a separate device such as a separate computer or server, or a part of the separate device such as a virtual machine partitioned on the server.

[00130] In step 330b, the specific process of executing the local transaction by the second consensus node and recording the local transaction to the second blockchain may be referred to the

specific process of executing the local transaction by the first consensus node and recording the local transaction to the first blockchain.

[00131] After the second consensus node records the local transaction to the second blockchain, to verify that the second consensus node actually uploads the local transaction, the second consensus node also generates first uploading verification information.

[00132] In step 330c, the first uploading verification information is transmitted to the third consensus node in the second global consensus network by the second consensus node.

[00133] In step 330d, the third consensus node is located in the second global consensus network and is a module for receiving and verifying according to the first uploading verification information. The third blockchain maintained by the third consensus node is accessible by nodes other than the third consensus node. When the third consensus node verifies the first uploading verification information, the third consensus node runs and deploys a target state verification smart contract. In the contract, a verification code for the first uploading verification information transmitted by the first global consensus network is maintained. The third consensus node may be a separate device such as a separate computer or server, or a part of the separate device such as a virtual machine partitioned on the server.

[00134] Next, after step 330, in step 340, the first relay node obtains an uploading verification result of the local transaction from the third consensus node.

[00135] In an embodiment, step 340 includes an interactive process in which the first relay node obtains the uploading verification result from the third consensus node and the first relay node transmits the uploading verification result to the local consensus network. Through the uploading verification result, it is known that the local transaction of the first type has been uploaded successfully.

[00136] Through steps 310 to 340, according to this embodiment of this application, the global consensus network is divided into a second global consensus network and a first global consensus network, the local consensus network can synchronize information of the second global consensus network, and the local transaction in the local consensus network can be transmitted back through the first relay node, thus realizing the secure uploading of the target transaction. That is, the local transaction is executed and recorded in the first global consensus network, and the uploading verification result is confirmed in the second global consensus network. The specific calculation process, calculation logic, and intermediate results are all completed in the first global consensus network. In the meantime, the consensus of the first global consensus network and the verification

in a main blockchain may effectively use the characteristics of blockchain non-tampering to ensure the correctness and credibility of transactions.

[00137] The above is the general description of steps 310 to 340. Since steps 320 and 330b have been detailed in the above general description, specific implementations of steps 330a, 330c to 330d, and 340 will be described in detail below.

[00138] Detailed Description of Step 330a

[00139] In step 330a, the first relay node transmits the local transaction to the second consensus node in the first global consensus network after determining that the parent type of the local transaction is the first type.

[00140] In some embodiments, the setting of a transmitting period of the local transaction includes the following cases:

[00141] (1) The local transaction is transmitted immediately after it is determined that the parent type of the local transaction is the first type.

[00142] (2) According to a first transmitting period preset by the first relay node, after it is determined that the parent type of the local transaction is the first type and when the first transmitting period has passed, the local transaction is transmitted. For example, the first transmitting period preset by the first relay node is 2 hours. If it is determined that the time when the parent type of the local transaction is the first type is 7:00 pm on a day, the time when the local transaction is transmitted is 9:00 pm on the same day.

[00143] For the decomposition of step 330a, this embodiment of this application provides three decomposition modes. Each decomposition mode expands the detailed description of step 330a from different perspectives. The first decomposition mode is described first.

[00144] Referring to the above description of the transmitting period of the local transaction, the manner of whether the local transaction is immediately transmitted or transmitted after the first transmitting period has passed is equivalent to establishing a connection with the first global consensus network every time a local transaction of a first type is encountered. Therefore, the overhead is too large. To reduce the overhead and reduce the network resource loss, in an embodiment, queuing may be performed first, and packaging and transmitting may be performed once when predetermined conditions are satisfied.

[00145] In this embodiment of this application, for the local transaction of the first type, the local transaction is transmitted to the second consensus node to instruct the second consensus node to record the local transaction to the second blockchain. Since the second blockchain is not allowed

to be accessed by nodes other than the second consensus node, the security of the local transaction of the first type is guaranteed. In the meantime, after completing the uploading of the local transaction, the second consensus node transmits the first uploading verification information after the uploading to the third consensus node. The third consensus node performs uploading verification to obtain an uploading verification result, and the uploading verification result is obtained from the third consensus node, thus ensuring that the uploading of the local transaction is authentic although the local transaction cannot be viewed by the outside world, thereby improving the uploading security of the local transaction.

[00146] In some embodiments, the first consensus node belongs to a local consensus network. The second consensus node belongs to the first global consensus network in global consensus networks. The third consensus node belongs to a second global consensus network in the global consensus networks. The first relay node is configured to transmit the local transaction of the first type to the second consensus node in the first global consensus network.

[00147] According to this embodiment of this application, the global consensus network is divided into a second global consensus network and a first global consensus network, the local consensus network can synchronize information of the second global consensus network, and the local transaction in the local consensus network can be transmitted back through the first relay node, thus realizing the secure uploading of the local transaction of the first type. That is, the local transaction is executed and recorded in the first global consensus network, and the uploading verification result is confirmed in the second global consensus network. The specific calculation process, calculation logic, and intermediate results are all completed in the first global consensus network. In the meantime, the consensus of the first global consensus network and the verification in a main blockchain may effectively use the characteristics of blockchain non-tampering to ensure the correctness and credibility of transactions. This embodiment of this application improves the uploading security of the target transaction in the blockchain hierarchical consensus system. This embodiment of this application may be applied to various scenarios such as blockchain, structured information processing, and security technology.

[00148] When this embodiment is specifically implemented, referring to FIG. 4, step 330a (performed by the first relay node) specifically includes the following steps:

[00149] Step 410: Place the local transaction in a queue for queuing.

[00150] Step 420: Package local transactions in the queue in response to the queue satisfying a predetermined condition, and transmit the local transaction to the second consensus node.

[00151] Steps 410 to 420 are described in detail below.

[00152] In step 410, a queue is provided in the first relay node. The queue is a transaction storage container for storing local transactions. Each time the first relay node determines that the parent type of a local transaction is the first type, the local transaction is placed in a queue for queuing.

[00153] There are generally two modes of placement in the queue.

[00154] (1) The local transaction is directly placed at the tail of the queue.

[00155] (2) A position of the local transaction to be inserted in the queue is determined first. The position of the local transaction to be inserted in the queue is referred to as a target position. The local transaction at the target position and the local transaction behind the target position are moved backward, and then the local transaction is placed at the target position of the queue.

[00156] In step 420, the first relay node packages local transactions in the queue in response to the queue satisfying a predetermined condition, and transmits the local transaction to the second consensus node.

[00157] The predetermined condition includes at least one of the following: the number of local transactions reaches a first threshold, and a time difference between a current time and a previous packaging time reaches a second threshold. For example, the first relay node packages local transactions in the queue in response to the number of local transactions in the queue reaching the first threshold, and transmits the local transaction to the second consensus node. For example, the first relay node packages local transactions in the queue in response to the time difference between the current time and the previous packaging time reaching the second threshold, and transmits the local transaction to the second consensus node. For another example, the first relay node packages local transactions in the queue in response to the number of local transactions in the queue reaching the first threshold and the time difference between the current time and the previous packaging time reaching the second threshold, and transmits the local transaction to the second consensus node.

[00158] In this embodiment, all the local transactions of the first type are placed in the same queue for packaging and transmitting, thereby improving the packaging efficiency.

[00159] Under the predetermined condition, if only the number reaches the first threshold, the timeliness of packaging and transmitting will be affected, resulting in some local transactions not being uploaded for a long time. If only the time difference reaches the second threshold, although the timeliness is guaranteed, the number of packages per time may be too small, thus

wasting network resources. Similarly, if the number reaches the first threshold and the time difference reaches the second threshold at the same time, there is a possibility that the time difference has met the second threshold, but the number cannot reach the first threshold all the time, which still affects the timeliness of packaging and transmitting. Therefore, in some embodiments, step 420 specifically includes:

[00160] packaging local transactions in the queue in response to the number of local transactions in the queue reaching a first threshold.

[00161] In this embodiment, the number of local transactions is determined as the condition for packaging, thereby avoiding too few packages each packaging time and wasting network resources.

[00162] In some embodiments, step 420 further specifically includes: determining a time difference between a current time and a previous packaging time in response to the number of local transactions in the queue not reaching a first threshold; and

[00163] packaging local transactions in the queue in response to the time difference reaching a second threshold.

[00164] In this embodiment, the number and the time difference are combined, so that the number of packages per packaging time is not too small and the network resources are wasted. Timeliness is considered, thus avoiding the situation that some local transactions cannot be uploaded for a long time.

[00165] Since the child types of the local transaction of the first type are not the same, the local transactions of the same child type are packaged into a packet and uploaded together, which is helpful to the uploading efficiency of the first global consensus network. Therefore, in other embodiments, each child type corresponds to a respective queue so that local transactions of the same child type are placed in the same queue. For example, two queues: M and N are shown in FIG. 5. Queue M is configured to store local transactions of child type a. Queue N is configured to store local transactions of child type b.

[00166] In a specific implementation of this embodiment, step 410 includes: determining a child type of the local transaction of the first type; determining a queue corresponding to the child type of the local transaction; and placing the local transaction in the queue corresponding to the child type for queuing.

[00167] With continued reference to FIG. 5, local transactions 1 to 8 have been placed in corresponding queues according to child types. That is, local transactions 1, 3, 5, and 7 of child type

a are placed in queue M, and local transactions 2, 4, 6, and 8 of child type b are placed in queue N. When local transaction 9 is required to be placed in the queue, the child type of local transaction 9 is determined first. If the child type of local transaction 9 is a, the local transaction is placed at the tail of queue M (at the dashed box of queue M shown in FIG. 5). If the child type of local transaction 9 is b, the local transaction is placed at the tail of queue N (at the dashed box of queue N shown in FIG. 5).

[00168] In this embodiment, local transactions of the same child type are packaged and transmitted. The second consensus node may use a target smart contract of the same child type (the target smart contract or target contract refers to a smart contract for uploading a target transaction of a certain child type, which contains code for uploading the target transaction of the child type) to uniformly execute the local transactions, thus reducing the re-invocation of the target smart contracts of different child types for execution, and improving the execution efficiency. For example, the second consensus node receives a packaged local transaction. All the local transactions are executed by using a target smart contract T. Then the second consensus node may process all the local transactions by using the target smart contract T, thus alleviating the problem that the processing efficiency of the second consensus node is reduced because the target smart contract needs to be replaced many times.

[00169] Step 420 includes: packaging, in response to the number of local transactions in a queue reaching a third threshold, local transactions in the queue. Taking local transaction 9 placed in queue M as an example, after local transaction 9 is placed, the number of local transactions in queue M reaches a third threshold (for example, the third threshold is 5), and the local transactions in queue M are packaged and transmitted to the second consensus node.

[00170] In this embodiment, the number of local transactions is determined as the condition for packaging, thereby avoiding too few packages each packaging time and wasting network resources.

[00171] The above is the detailed description of the first decomposition mode of step 330a, and the second decomposition mode will be described next.

[00172] When step 330a is decomposed into steps 410 to 420, local transactions are directly placed in a queue for queuing. However, for some local transactions to be urgently uploaded, if the local transactions are directly placed in the queue, the urgent uploading requirements of different local transactions cannot be met. Therefore, in an embodiment, in step 330a, the local transaction with high priority is first transmitted to the first global consensus network for uploading in

combination with the priority of the local transaction, so as to meet the urgent uploading requirement.

[00173] When this embodiment is specifically implemented, referring to FIG. 6, step 330a (performed by the first relay node) includes the following steps:

[00174] Step 610: Determine a child type of the local transaction of the first type.

[00175] Step 620: Determine a priority of the local transaction according to the child type.

[00176] Step 630: Place the local transaction in a queue for queuing in a descending order of priority.

[00177] Step 640: Sequentially take out the local transactions from the queue, and transmit the local transactions to the second consensus node.

[00178] Steps 610 to 640 are described in detail below.

[00179] In step 610, the first relay node determines a child type of the local transaction of the first type.

[00180] Before explaining the child types, it is emphasized again that the parent type of the local transaction is used for indicating whether the local transaction is required to be executed and recorded on the blockchain inaccessible by the outside world. Therefore, the local transaction of the first type indicates that the local transaction is required to be executed and recorded on the blockchain inaccessible to the outside world (that is, executed by the second consensus node and recorded on the second blockchain). Unlike the first type, the child type refers to a transaction type of the local transaction. For example, if the local transaction is an invoice reimbursement transaction, the transaction type of the invoice reimbursement transaction is an invoicing type, and therefore, the child type is an invoicing type. For example, if the local transaction is an enterprise purchase transaction, the transaction type of the enterprise purchase transaction is an order type, and therefore, the child type is an order type. If the child type is the invoicing type, the first relay node determines a priority corresponding to the invoicing type, so as to reflect an urgent uploading demand of an enterprise invoicing transaction with the child type as the invoicing type according to the priority.

[00181] In some embodiments, the manner in which the child type of the local transaction of the first type is determined includes, but is not limited to, the following manners:

[00182] (1) A second key field is extracted from the content of the local transaction of the first type, and a second key field-to-child type comparison table is searched according to the second

key field to determine a child type of the local transaction. The following is an example of the second key field-to-child type comparison table:

Table 2

Second key field	Child type
Reimbursement	Invoice reimbursement type
Invoice query	Invoice query type
Invoicing	Invoicing type
Resource exchange	Resource exchange type
Resource transfer	Resource transfer type
Resource statistics	Resource statistics type

[00183] As shown in Table 2, for the second key fields included in the local transaction, which are reimbursement, invoice query, and invoicing, it is determined that the child types of the local transaction are an invoice reimbursement type, an invoice query type, and an invoicing type. Similarly, when the second key fields included in the local transaction are resource exchange, resource transfer, and resource statistics, it is determined that the child types of the local transaction are a resource exchange type, a resource transfer type, and a resource statistics type.

[00184] In Table 1, the first key field is used for determining whether the parent type of the local transaction is the first type or the second type, while in Table 2, the child type is determined according to the second key field based on determining that the local transaction is the first type.

[00185] For example, the content of a local transaction includes "Electronic invoice reimbursement 0001: amount 500 for accommodation", which includes the first key field of "Invoice". Therefore, it is determined that the parent type of the local transaction is the first type, and it is determined that the child type is the invoice reimbursement type according to the included second key field "Reimbursement". For example, the local transaction is "Resource exchange, 100 first resources are converted into 10 second resources", which includes the first key field of "Resource". Therefore, it is determined that the parent type of the local transaction is the first type, and it is determined that the child type is the resource exchange type according to the included second key field "Resource exchange".

[00186] (2) After it is found that the first consensus node receives a local transaction, a child type identifier is obtained from the content of the local transaction, and a child type of the local transaction is determined according to the child type identifier.

[00187] For example, the content of a local transaction includes "Electronic invoice reimbursement 0001, amount 500 for accommodation, [invoice reimbursement type]". It is determined, according to the type identifier "[invoice reimbursement type]", that the child type is the invoice reimbursement type.

[00188] After determining the child type, in step 620, the manner of determining the priority according to the child type includes: determining the priority by looking up a child type-to-priority comparison table.

[00189] The child type-to-priority comparison table lists a corresponding relationship between child types and priorities. The following is an example of the child type-to-priority comparison table:

Table 3

Child type	Priority
Invoice reimbursement type	10
Invoice query type	9
Invoicing type	8
Resource exchange type	7
Resource transfer type	6
Resource statistics type	5

[00190] As shown in Table 3, for transactions with the child type being the invoice reimbursement type, the demand for urgent uploading of invoice reimbursement transactions is high. Therefore, the priority is set to 10. Similarly, according to different degrees of demand of each child type for emergency uploading, the priority of the invoice query type is 9, the priority of the invoicing type is 8, the priority of the resource exchange type is 7, the priority of the resource transfer type is 6, and the priority of the resource statistics type is 5.

[00191] For example, the content of a local transaction includes "Electronic invoice reimbursement 0001, amount 500 for accommodation". If it is determined that the child type is the invoice reimbursement type, the priority is 10. For example, the local transaction is "100 first

resources are exchanged with 10 second resources". If it is determined that the child type is the resource exchange type, the priority is 7.

[00192] After the priority is determined, in step 630, the first relay node places the local transaction in a queue for queuing in a descending order of priority.

[00193] Reference is made to FIG. 7. Queue L before local transaction 5 is placed and queue L after local transaction 5 is placed are shown in FIG. 7. Before local transaction 5 is placed in queue L, local transactions 1 to 4 are already stored in queue L, and the priorities of local transactions 1 to 4 are 9, 8, 7, and 6 in sequence. When a local transaction is required to be placed in queue L, the priority of the local transaction is first determined as 10, and a target position at which local transaction 5 is placed in queue L is determined in a descending order of priority. That is, the target position is the dotted box of queue L shown on the left side of FIG. 7. Local transaction 1 at the target position and local transactions 2 to 4 behind the target position are moved backward, and then local transaction 5 is placed at the target position of the queue. The placement result is queue L shown on the right side of FIG. 7.

[00194] In step 640, the local transactions are sequentially taken out from the queue, and transmitted to the second consensus node.

[00195] The manner of taking out the local transactions from the queue in step 640 includes:

[00196] (1) Referring to FIG. 7, if it is determined that the number of local transactions in the queue reaches a fourth threshold, the local transactions are sequentially taken out from the queue, packaged, and transmitted to the second consensus node in the first global consensus network.

[00197] (2) If it is determined that the time difference between the current time and the previous packaging time reaches a fifth threshold, the local transactions are sequentially taken out from the queue, packaged, and transmitted to the second consensus node in the first global consensus network.

[00198] By steps 610 to 640, urgent uploading demands of the target transaction are met according to the priority, and the uploading flexibility is improved.

[00199] The above is the detailed description of the second decomposition mode of step 330a, and the third decomposition mode will be described next.

[00200] When step 330a is expanded into steps 410 to 420 by the first decomposition mode, or step 330a is expanded into steps 610 to 640 by the second decomposition mode, the first relay node only transmits the local transaction to the second consensus node. That is, the second consensus node cannot determine whether the received local transaction is a real transaction

generated by the first relay node, and the security is low. Therefore, in some embodiments, in step 330a, the second consensus node in step 330b can ensure that the local transaction is a real transaction transmitted by the first relay node in combination with signature verification of the first relay node, thus improving the security.

[00201] When this embodiment is specifically implemented, referring to FIG. 8, step 330a (performed by the first relay node) includes the following steps:

[00202] Step 810: Generate a third digest for the local transaction according to a predetermined digest algorithm.

[00203] Step 811: Encrypt the third digest with a private key of the first relay node to obtain a second signature.

[00204] Step 812: Transmit the local transaction and the second signature to the second consensus node.

[00205] Correspondingly, step 330b (performed by the second consensus node) includes the following steps:

[00206] Step 820: Decrypt the second signature with a public key of the first relay node to obtain a fourth digest.

[00207] Step 821: Generate a third digest for the local transaction according to a predetermined digest algorithm.

[00208] Step 822: Execute the local transaction and record the local transaction to the second blockchain when the third digest is consistent with the fourth digest.

[00209] Steps 810 to 812 and steps 820 to 822 are described in detail below.

[00210] In steps 810 to 812, the signing process includes: digesting a to-be-signed subject with a digest algorithm (for example, a hash algorithm) and then encrypting the digest with a private key (or public key) of a signer. Different signature algorithms may be reflected in different digest algorithms. Therefore, when the first relay node generates the second signature, the third digest of the local transaction is obtained using the predetermined digest algorithm, and the third digest is encrypted with the private key of the first relay node.

[00211] In steps 820 to 822, the process of verifying the signature includes that a signature verifier decrypts the signature with the public key (or private key) of the signer, and then generates a digest for the to-be-signed subject with the same digest algorithm as the signature process. If the decrypted digest is the same as the generated digest, the signature verification is successful.

Therefore, when the second consensus node performs signature verification, the public key of the first relay node is used for decrypting the second signature to obtain the third digest, the same predetermined digest algorithm is used for generating the fourth digest for the local transaction, and the third digest is compared with the fourth digest. If the third digest is consistent with the fourth digest, the verification of the second signature is successful.

[00212] The above is a general description of three decomposition modes of step 330a. In the above decomposition modes, the first relay node transmits the local transactions of the first type to the same first global consensus network based on the global consensus network including a second blockchain (as shown in FIG. 1C). However, in another embodiment, the first global consensus network is selected from multiple candidate global consensus networks (multiple candidate global consensus networks as shown in FIG. 10), and the candidate global consensus networks correspond to child types of different local transactions. For example, there is a first global consensus network for processing transactions with child types being the invoicing type and a first global consensus network for processing transactions with child types being the order type, which is helpful to improve the uploading efficiency of transactions.

[00213] When this embodiment is specifically implemented, referring to FIG. 9, step 330a (performed by the first relay node) specifically includes the following steps:

[00214] Step 910: Determine a child type of the local transaction of the first type.

[00215] Step 920: Determine a candidate global consensus network corresponding to the child type as the first global consensus network.

[00216] Step 930: Transmit the local transaction to the second consensus node in the first global consensus network.

[00217] Steps 910 to 930 are described in detail below.

[00218] In step 910, the definition of the child type and the manner in which the child type is determined have been described in step 610.

[00219] In step 920, as shown in FIG. 10, multiple candidate global consensus networks are included in the global consensus network (FIG. 10 shows only four candidate global consensus networks, but there may be less than or greater than four candidate global consensus networks). Each candidate global consensus network corresponds to a child type.

[00220] According to the target smart contract demands of the child type, a child type-to-first global consensus network comparison table is made in advance, as shown in the following table:

Table 4

Child type	First global consensus network
Invoice reimbursement type	Candidate global consensus network 1
Invoicing type	Candidate global consensus network 2
Resource exchange type	Candidate global consensus network 3
Resource statistics type	Candidate global consensus network 4

[00221] As shown in Table 4, the child types of the local transaction may be an invoice reimbursement type, an invoicing type, a resource exchange type, and a resource statistics type. For the invoice reimbursement type, a target smart contract for reimbursement is needed. Therefore, candidate global consensus network 1 is selected as the first global consensus network, and candidate global consensus network 1 uniformly invokes the target smart contract for reimbursement to execute the local transaction of the invoice reimbursement type. For transactions of the invoicing type, a target smart contract for invoicing is needed. Therefore, candidate global consensus network 2 is selected as the first global consensus network, and candidate global consensus network 2 uniformly invokes the target smart contract for invoicing to execute the local transaction of the invoicing type. Similarly, for the resource exchange type, candidate global consensus network 3 uniformly invokes a target smart contract for exchange to execute the local transaction with the child type being the resource exchange type. Similarly, for the resource statistics type, candidate global consensus network 4 uniformly invokes a target smart contract for statistics to execute the local transaction with the child type being the resource statistics type.

[00222] After the comparison table is made, the comparison table may be looked up according to the child type to determine the first global consensus network.

[00223] After determining the first global consensus network, the first relay node transmits the local transaction to the second consensus node in the first global consensus network in step 930.

[00224] If the child type is the invoice reimbursement type, the first global consensus network is candidate global consensus network 1. The first relay node transmits the local transaction to the second consensus node in candidate global consensus network 1. The second consensus node invokes the target smart contract of reimbursement to execute the local transaction, and records the local transaction to the second blockchain in candidate second blockchain 1.

[00225] The specific process of executing the local transaction by the second consensus node and recording the local transaction to the second blockchain may be referred to the description of

executing the local transaction of the second type by the first consensus node and recording the local transaction to the first blockchain in the above embodiment. However, the difference is that the first blockchain can be accessed by nodes other than the first consensus node, while the second blockchain is not accessible by nodes other than the second consensus node.

[00226] According to the embodiments of steps 910 to 930, the local transactions of the same child type are effectively transmitted to the same candidate global consensus network through the comparison relationship between the child type and the first global consensus network, whereby the candidate second blockchain can uniformly invoke the target smart contract of the same type to execute the local transaction, and the processing efficiency of the first global consensus network in processing the local transaction is improved.

[00227] The above is the detailed description of step 330a.

[00228] Detailed Descriptions of Steps 330c to 330d

[00229] In step 330c, the second consensus node transmits the first uploading verification information of the local transaction to the third consensus node in the second global consensus network. In step 330d, the third consensus node verifies uploading of the local transaction according to the first uploading verification information.

[00230] The execution link of the second consensus node is divided into two cases. In one case, the second consensus node has a trusted execution environment (TEE). In the other case, the second consensus node does not have a TEE.

[00231] Before describing that the second consensus node has the TEE, the relevant contents of the TEE are explained first.

[00232] The TEE is a hardware-based secure computing solution. Through software and hardware, a secure region is constructed in a central processing unit (CPU) to ensure the confidentiality and integrity of programs and data loaded therein. The TEE divides hardware resources and software resources of the system into two execution environments: trusted part and untrusted part (ordinary part). The two environments are isolated, and the untrusted part cannot access the storage and memory of the trusted part.

[00233] The TEE solution includes: an Intel SGX solution, an arm trustzone solution, and an amd sev solution. The Intel SGX solution refers to a TEE solution proposed by Intel, which may support the TEE on a specific model of CPU produced by Intel. The TEE mentioned herein, unless otherwise specified, refers to the TEE based on the Intel SGX solution.

[00234] TEE hardware key: a hardware key that is baked in the CPU when producing the CPU by Intel. A private key thereof cannot be derived under any circumstances, and a public key thereof may be derived under certain usage scenarios, but Intel does not recommend this practice.

[00235] Based on the content of the TEE, in this embodiment, when the second consensus node has a TEE, the second blockchain is encrypted with a key of the TEE. The first uploading verification information is transmitted by the second consensus node to the second relay node in response to a first uploading verification information request transmitted by the second relay node. The first uploading verification information request is transmitted by the second relay node in response to adding an encrypted block on the second blockchain. The encrypted block is encrypted with the key of the TEE. The first uploading verification information is used for instructing the second relay node to forward the first uploading verification information to the third consensus node.

[00236] In this embodiment, the block is encrypted with the key of the TEE, and the first uploading verification information is forwarded by the second relay node, thereby further improving the uploading security of the target transaction of the first global consensus network.

[00237] The above is the description of the second consensus node having a TEE, followed by the detailed description of the second consensus node not having a TEE.

[00238] In some embodiments, as shown in FIG. 12A, the first uploading verification information includes a digest value of the local transaction and a digest value that is determined according to a first path in a first tree and recorded in a first bypass node. A leaf node in the first tree records a digest value of a transaction recorded on the second blockchain. The first path is a path from a leaf node in the first tree to a root node in the first tree, and the leaf node in the first tree records the digest value of the local transaction. The first bypass node is a sibling node in the first tree of nodes other than the root node in the first path. The first uploading verification information is further used for instructing the third consensus node to determine a first comparative digest value according to the digest value of the local transaction and the digest value recorded in the first bypass node. The verification is successful when the digest value recorded in the root node in the first tree and the first comparative digest value are consistent.

[00239] The first tree is generated by the second consensus node. Referring to FIG. 11, step 330c (performed by the second consensus node) includes the following steps:

[00240] Step 1110: Generate a first tree.

[00241] Step 1111: Determine a first bypass node of a first path.

[00242] Step 1112: Transmit a digest value of the local transaction and a digest value recorded in the first bypass node to the third consensus node.

[00243] Correspondingly, step 330d (performed by the third consensus node) includes the following steps:

[00244] Step 1120: Determine a first comparative digest value according to the digest value of the local transaction and the digest value recorded in the first bypass node.

[00245] Step 1121: Determine that the verification is successful when a digest value recorded in a root node in the first tree and the first comparative digest value are consistent.

[00246] Steps 1110 to 1112 and steps 1120 to 1121 are described in detail below.

[00247] In step 1110, the leaf node in the first tree is a digest value of a transaction recorded on the second blockchain. The recorded transaction includes the local transaction. The order of leaf nodes in the first tree may be the order in which the local transactions are received by the second consensus node. For example, when the second consensus node first receives local transaction 1 "Employee A reimburses 1,000 resources in Enterprise B" and then receives local transaction 2 "Employee C reimburses 500 resources in Enterprise D", local transaction 1 is taken as the first leaf node, and local transaction 2 is taken as the second leaf node. The digest algorithm used for obtaining the digest value may be one of the following:

[00248] (1) The second consensus node pre-specifies the same digest algorithm for an entity to which each local transaction belongs. For example, the same digest algorithm is used for an electronic invoice reimbursement transaction of Enterprise B and an electronic invoice reimbursement transaction of Enterprise D.

[00249] (2) The second consensus node pre-specifies different digest algorithms for the entity to which each local transaction belongs. For example, different digest algorithms are pre-specified for Enterprise B and Enterprise D. When it is determined that the local transaction belongs to Enterprise B, the digest algorithm pre-specified for Enterprise B is adopted. When it is determined that the local transaction belongs to Enterprise D, the digest algorithm pre-specified for Enterprise D is adopted. Since this embodiment adopts different digest algorithms for different entities, the difficulty of cracking local transactions from digest values is increased, and the uploading security of the local transactions is improved.

[00250] When the third consensus node verifies according to the digest value of the local transaction and the digest value recorded in the first bypass node, the digest algorithm adopted by the third consensus node is consistent with the above digest algorithm.

[00251] In this embodiment, there is at least one intermediate layer between a layer where the leaf node is located and a layer where the root node is located in the first tree. The intermediate layer includes at least one intermediate node. In two adjacent layers in the first tree, a node of the previous layer is generated by at least two nodes of the next layer, and the same node in the next layer is only configured to generate one node in the previous layer. The nodes include root nodes, intermediate nodes, and leaf nodes.

[00252] The first tree in this embodiment of this application is usually a binary tree or a multi-way tree. The first tree has all the characteristics of a tree structure. The tree structure often contains multiple nodes (nodes shown in FIG. 12B). The multiple nodes include at least one leaf node and one root node, and usually include multiple intermediate nodes between the leaf node and the root node.

[00253] Referring to the first tree shown in FIG. 12B, the leaf node of the first tree records digest values of transactions. In two adjacent layers in the first tree, a node of the previous layer is generated by at least two nodes of the next layer, and the same node in the next layer is only configured to generate one node in the previous layer. As shown in FIG. 12B, a digest value in an intermediate node 7 of the second layer is generated by the digest algorithm after a leaf node 1 and a leaf node 2 of the first layer are connected in series. A digest value in an intermediate node 8 of the second layer is generated by the digest algorithm after the leaf node 3 and the leaf node 4 are connected in series. A digest value in an intermediate node 9 is generated by the digest algorithm after a leaf node 5 and a leaf node 6 are connected in series. A digest value in an intermediate node 11 of the third layer is generated by the digest algorithm after the intermediate node 7 and the intermediate node 8 are connected in series. A digest value in an intermediate node 12 of the third layer is generated by the digest algorithm after the intermediate node 9 and the intermediate node 10 are connected in series. A digest value in the root node in the first tree is generated by the digest algorithm after the intermediate node 11 and the intermediate node 12 are connected in series.

[00254] In step 1111, the first path is a path between the leaf node corresponding to the digest value of the local transaction in the first tree and the root node in the first tree. The first bypass node is a sibling node in the first tree of nodes other than the root node in the first path.

[00255] As shown in FIG. 12C, assuming that the digest value of the local transaction is recorded in the leaf node 4, the first path is a path between the leaf node 4 in the first tree and the root node in the first tree. The first path includes the leaf node 4, the intermediate node 8, the intermediate node 11, and the root node in the first tree. Since the intermediate node 8 needs to be generated based on the leaf node 3 and the leaf node 4, the leaf node 3 needs to be obtained when

the leaf node 4 is known. Therefore, the leaf node 3 is determined as the first bypass node. Similarly, since the intermediate node 11 needs to be generated based on the intermediate node 7 and the intermediate node 8, the intermediate node 7 also needs to be obtained after the intermediate node 8 is generated according to the leaf node 3 and the leaf node 4. Therefore, the intermediate node 7 is determined as the first bypass node. Similarly, since the intermediate node 13 needs to be generated based on the intermediate node 11 and the intermediate node 12, the intermediate node 12 also needs to be obtained after the intermediate node 11 is generated according to the intermediate node 8 and the intermediate node 7. Therefore, the intermediate node 12 is determined as the first bypass node. Therefore, the first bypass node includes the leaf node 3, the intermediate node 7, and the intermediate node 12, and the first bypass node is represented in a thick black solid line box in FIG. 12C. Nodes in the first tree other than the first bypass node and the leaf node corresponding to the local transaction are not provided to the third consensus node. The advantage is that when the third consensus node verifies, related information of the root node generated layer by layer from the leaf node corresponding to the digest value of the local transaction has been provided to the third consensus node, so as to ensure that the third consensus node can verify correctly. In the meantime, information unrelated to verification shall be avoided from being provided to the third consensus node as far as possible, so as to reduce the information leakage of other transactions. In the meantime, even the first bypass node is generally only provided in the form of digest value, which reduces the real information leakage of related transactions.

[00256] After determining the first bypass node in step 1111, the second consensus node transmits a digest value of the local transaction and a digest value recorded in the first bypass node to the third consensus node in step 1112.

[00257] Next, after the third consensus node receives the digest value of the local transaction and the digest value recorded in the first bypass node, the third consensus node determines a first comparative digest value using the digest value recorded in the first bypass node and the digest value of the local transaction in step 1120.

[00258] The process of determining the first comparative digest value includes: determining the digest value of the local transaction, inferring the digest values recorded in the nodes in the first path layer by layer according to the digest value recorded in the first bypass node and the digest value of the local transaction (such as the part circled by the curve in FIG. 12C), and inferring the digest value of the root node in the first path as the first comparative digest value.

[00259] As shown in FIG. 12C, when determining the first comparative digest value using the digest value recorded in the first bypass node and the digest value of the local transaction, it is

assumed that the digest value of the local transaction is recorded in the leaf node 4, and the first bypass node includes the leaf node 3, the intermediate node 7, and the intermediate node 12. In this case, the digest value recorded in the intermediate node 8 is inferred from the digest value recorded in the leaf node 4 and the digest value recorded in the leaf node 3. The digest value in the intermediate node 11 is inferred from the inferred digest value in the intermediate node 8 and the digest value in the intermediate node 7. The digest value in the root node is inferred from the inferred digest value in the intermediate node 11 and the digest value in the intermediate node 12.

[00260] After obtaining the first comparative digest value in step 1120, the third consensus node compares the digest value recorded in the root node of the first tree with the first comparative digest value. In step 1121, the verification is successful if the digest value recorded in the root node in the first tree and the first comparative digest value are consistent.

[00261] Through steps 1110 to 1112 and steps 1120 to 1121, this embodiment of this application ensures that the third consensus node can accurately verify the local transaction and verify whether the transaction has been truly uploaded. In addition, since only the digest value of the local transaction and the digest value recorded in the first bypass node are shared to the third consensus node, the leakage of the real information of the related transaction is reduced, and the security is improved.

[00262] In other embodiments, in addition to verifying whether the transaction is actually recorded, associated data generated after the local transaction is executed may also be verified. The associated data refers to parameters that may be affected after the transaction is executed, such as the total amount of resources or the number of items. For example, the local transaction is "Electronic invoice reimbursement 0001, amount 500 for accommodation". After the transaction is executed and uploaded, the total amount of enterprise resources will be affected. Assuming that the original total amount of enterprise resources is 500,000, after reimbursement of the electronic invoice, the total amount of resources is changed into 499,500. The total amount of resources is the associated data generated by invoice reimbursement. For another example, the local transaction is "Give 600 movie tickets to Enterprise A". The remaining movie tickets in a cinema will be affected after the transaction is executed. Assuming that there are 5,000 original movie tickets in the cinema, the number of movie tickets in the cinema will be changed to 4,400 after the transaction is executed and uploaded. The number of movie tickets is the associated data generated by the ticket giving transaction. To verify the associated data obtained after execution of the local transaction, in another embodiment, the first uploading verification information includes, in addition to the digest value of the local transaction and the digest value recorded in the first bypass node (as shown in

FIG. 12A). The first uploading verification information further includes the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node (as shown in FIG. 14A).

[00263] In some embodiments, the first uploading verification information further includes a digest value of associated data obtained after execution of the local transaction and a digest value recorded in a second bypass node in a second tree. A leaf node in the second tree records a digest value of associated data obtained after execution of a transaction recorded on the second blockchain. A second path is a path from a leaf node in the second tree to a root node in the second tree, and the leaf node in the second tree records the digest value of the associated data obtained after execution of the local transaction. The second bypass node is a sibling node in the second tree of nodes other than the root node on the second path. The first uploading verification information is further used for instructing the third consensus node to determine, when the digest value recorded in the root node in the first tree and the first comparative digest value are consistent, a second comparative digest value according to the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node. The verification is successful when the digest value recorded in the root node in the second tree and the second comparative digest value are consistent.

[00264] The second tree is generated by the second consensus node. Under this embodiment, referring to FIG. 13, step 330c (performed by the second consensus node) further includes the following steps:

[00265] Step 1310: Generate a second tree.

[00266] Step 1311: Determine a second bypass node of a second path.

[00267] Step 1312: Transmit a digest value of associated data obtained after execution of the local transaction and a digest value recorded in the second bypass node to the third consensus node.

[00268] Correspondingly, step 1121 in step 330d (performed by the third consensus node) further includes the following steps:

[00269] Step 1320: Determine a second comparative digest value according to the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node when the digest value recorded in the root node in the first tree is consistent with the first comparative digest value.

[00270] Step 1321: Determine that the verification is successful when a digest value recorded in a root node in the second tree and the second comparative digest value are consistent.

[00271] Steps 1310 to 1312 and steps 1320 to 1321 are described in detail below.

[00272] In step 1310, the leaf node in the second tree records a digest value of associated data obtained after execution of a transaction recorded on the second blockchain. The recorded transaction includes the local transaction. The order of leaf nodes in the second tree may be a pre-specified order of associated data. For example, the order of enterprises A, B, C, D ... is taken as the order of the leaf nodes in the second tree. When the second consensus node receives the local transaction 1 "Employee X reimburses 1,000 resources in Enterprise B", the transaction causes a change in the total number of resources of Enterprise B, whereby the total resources of Enterprise B are changed into 49,000. Therefore, the digest value in the leaf node 2 in the second tree is changed into a digest value of 49,000. When the second consensus node receives the local transaction 2 "Employee Y reimburses 500 resources in Enterprise D", the transaction causes a change in the total number of resources of Enterprise D, whereby the total resources of Enterprise D are changed into 24,500. Therefore, the digest value in the leaf node 4 in the second tree is changed into a digest value of 24,500. Entities to which local transactions belong may correspond to respective digest algorithms, and digest values of the local transactions may be generated by the digest algorithms corresponding to the entities to which the local transactions belong. The digest algorithms corresponding to different entities may be the same or different. The digest algorithms may be pre-specified by the second consensus node.

[00273] In step 1311, the second path is a path between the leaf node corresponding to the digest value of the associated data obtained after execution of the local transaction in the second tree and the root node in the second tree. The second bypass node is a sibling node of nodes in the second tree other than the root node in the second path.

[00274] As shown in FIG. 14C, assuming that the digest value of the associated data obtained after execution of the local transaction is recorded in the leaf node 4, the second path is a path between the leaf node 4 in the second tree and the root node in the second tree. The second path further includes the leaf node 4, the intermediate node 8, the intermediate node 11, and the root node in the second tree. The second bypass node in the second path includes the leaf node 3, the intermediate node 7, and the intermediate node 12, and the second bypass node is represented in a thick black solid line box in FIG. 14C. The digest values in the second tree other than the digest value recorded in the second bypass node and the digest value of the associated data obtained after execution of the local transaction are not provided to the third consensus node.

[00275] After determining the second bypass node in step 1311, the second consensus node transmits the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node to the third consensus node in step 1312.

[00276] Then, after receiving the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node, the third consensus node determines a second comparative digest value according to the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node when the digest value recorded in the root node in the first tree is consistent with the first comparative digest value in step 1320.

[00277] The process of determining the second comparative digest value includes: inferring the digest values recorded in the nodes in the second path according to the digest value recorded in the second bypass node and the digest value of the associated data obtained after execution of the local transaction (such as the part circled by the curve in FIG. 14C), and inferring the digest value of the root node in the second path as the second comparative digest value.

[00278] After obtaining the second comparative digest value in step 1320, the third consensus node compares the digest value recorded in the root node of the second tree with the second comparative digest value. In step 1321, the verification is successful when the digest value recorded in the root node in the second tree and the second comparative digest value are consistent.

[00279] Through steps 1310 to 1312 and steps 1320 to 1321, this embodiment of this application can not only verify whether the local transaction has been truly uploaded through the local transaction, but also verify whether the local transaction has been truly uploaded through the generated associated data. The double verification further improves the verification accuracy.

[00280] In some embodiments, the first uploading verification information further includes a first signature of the second bypass node. The first signature is obtained by encrypting a first digest by the second consensus node with a first private key of the second consensus node. The first digest is generated by the second consensus node according to the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node. The first signature is used for instructing the third consensus node to decrypt, when the root node in the second tree is consistent with a second to-be-compared root node, the first signature with a first public key of the second consensus node to obtain a second digest, and to generate a third digest according to the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and

the digest value recorded in the second bypass node. The verification is successful when the third digest and the second digest are consistent.

[00281] Referring to FIG. 15, the first uploading verification information shown in FIG. 15 includes the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, the digest value recorded in the second bypass node, and the first signature.

[00282] The first signature is obtained by signing the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node by the second consensus node. The manner of generating the first signature by the second consensus node is as follows.

[00283] The digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node are connected in series to obtain series data.

[00284] A first digest is generated by processing the series data according to a predetermined digest algorithm.

[00285] The first digest is encrypted with a first private key of the second consensus node to obtain the first signature.

[00286] Then when the first uploading verification information further includes the first signature, step 1321 includes:

[00287] decrypting the first signature with a first public key of the second consensus node to obtain a second digest when the digest value recorded in the root node in the second tree and the second comparative digest value are consistent;

[00288] generating a third digest by processing the series data according to the predetermined digest algorithm; and

[00289] determining that the verification is successful if the third digest is consistent with the second digest.

[00290] The order of connecting the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node in series may be one of the following:

[00291] (1) a pre-specified order, such as a pre-specified order of the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node; and

[00292] (2) an initial character lexicographic order of the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node. For example, if the initial character of the digest value of the local transaction is t, the initial character of the digest value recorded in the first bypass node is m, the initial character of the digest value of the associated data obtained after execution of the local transaction is a, and the initial character of the digest value recorded in the second bypass node is c, then the digest values are connected in series according to the attribute of the digest value of the associated data obtained after execution of the local transaction, the digest value recorded in the second bypass node, the digest value recorded in the first bypass node, and the digest value of the local transaction. This embodiment is helpful to improve the verification complexity and the uploading security of the local transaction of the first type.

[00293] In steps 1310 to 1312 and steps 1320-1321, through the first signature of the second consensus node and the verification of the first signature by the third consensus node, it is ensured that the digest value of the to-be-uploaded local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the to-be-uploaded local transaction, and the digest value recorded in the second bypass node are truly transmitted by the second consensus node, thus further improving the uploading security of the local transaction of the first type.

[00294] The above is the detailed description of steps 330c to 330d.

[00295] Detailed Description of Step 340

[00296] In step 340, the first relay node obtains an uploading verification result of the local transaction from the third consensus node.

[00297] Before the first relay node obtains the uploading verification result, the uploading verification result is a result obtained after the third consensus node verifies the uploading of the local transaction according to the first uploading verification information. Referring to FIG. 16, the first uploading verification information includes the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after

execution of the local transaction, the digest value recorded in the second bypass node, and transaction desensitization information. The transaction desensitization information is obtained by desensitizing the local transaction by the second consensus node. The desensitization is performed according to a desensitization field in a target contract. The target contract corresponds to a child type of the local transaction.

[00298] When the first uploading verification information includes the transaction desensitization information, step 340 includes: obtaining the transaction desensitization information from the third consensus node as the uploading verification result. The transaction desensitization information refers to information that can be disclosed to a third party in the target transaction, such as "Transaction of XX type has been completed and uploaded".

[00299] The transaction desensitization information is generated by the second consensus node, and the method for generating the transaction desensitization information by the second consensus node is as follows:

[00300] obtaining a child type of the local transaction of the first type;

[00301] obtaining a target contract corresponding to the child type; and

[00302] desensitizing the local transaction according to a desensitization field in the target contract to obtain the transaction desensitization information.

[00303] In this embodiment, the child type is obtained in a manner similar to the manner in which the child type is determined in the above embodiment.

[00304] The target contract is a contract for uploading a first type transaction of a certain child type. For example, the first type transaction of an invoice child type corresponds to the target contract of the invoice child type for uploading the first type transaction of the invoice child type. The first type transaction of an order child type corresponds to the target contract of the order child type for uploading the first type transaction of the order child type. In many implementations, the target contract needs to be executed by all the second consensus nodes.

[00305] The target contract may be determined by looking up a child type-to-candidate target contract comparison table according to the child type.

[00306] The child type may be the foregoing invoicing type, order type, resource exchange type, and resource statistics type. According to the contract requirements of each child type, a child type-to-target contract comparison table is made, for example:

Table 5

Child type	Target contract
Invoice reimbursement type	Reimbursement target contract
Invoicing type	Invoicing target contract
Resource exchange type	Exchange target contract
Resource statistics type	Statistic target contract

[00307] As shown in Table 5, a unified reimbursement target contract may be set for the invoice reimbursement type as the child type. A unified invoicing target contract may be set for the invoicing type as the child type. Similarly, a unified exchange target contract is set for the resource exchange type, and a unified statistic target contract is set for the resource statistics type.

[00308] After the comparison table is made, the comparison table may be looked up according to the child type to determine the target contract corresponding to the child type.

[00309] After the target contract is determined, the local transaction is desensitized according to a desensitization field in the target contract to obtain the transaction desensitization information. The desensitization field is a field carrying information that may be disclosed to a third party. For example, in the transaction of electronic invoice reimbursement, the transaction child type and number do not contain contents that cannot be disclosed to the third party, but may be disclosed to the third party.

[00310] For example, the local transaction of the first type is "Reimbursement of electronic invoice 0001, amount 500 for accommodation". Then the child type of the local transaction is first obtained as the invoice reimbursement type (for example, if the invoice reimbursement is included according to the second key field, the child type is determined as the invoicing type). Secondly, the above comparison table is looked up in response to the child type being the invoice reimbursement type, and the target contract is a reimbursement target contract (a desensitization field of the reimbursement target contract includes amount and purpose). Then, according to the desensitization field in the reimbursement target contract, that is, the transaction child type and number, the amount and purpose in "Electronic invoice reimbursement 0001, amount 500 for accommodation" are removed, only the transaction child type and number are kept, and the obtained transaction desensitization information is "Electronic invoice reimbursement 0001 is successful".

[00311] The advantage of this embodiment is that the desensitization accuracy is improved by processing the desensitization field in the target contract, thereby improving the uploading security of the target transaction.

[00312] The first uploading verification information in the above embodiment includes transaction desensitization information. However, in another embodiment, the first uploading verification information does not include transaction desensitization information, but includes instantaneous-ness information.

[00313] In some embodiments, the first uploading verification information further includes instantaneous-ness information of the local transaction. The instantaneous-ness information is determined by the second consensus node according to a child type of the local transaction. The process of obtaining the uploading verification result of the local transaction from the third consensus node includes: receiving the uploading verification result returned by the third consensus node after the verification of the third consensus node is successful when the instantaneous-ness information indicates that the local transaction is an instantaneous transaction.

[00314] Referring to FIG. 17, the first uploading verification information shown in FIG. 17 includes the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, the digest value recorded in the second bypass node, and instantaneous-ness information. Then when the first uploading verification information includes the instantaneous-ness information, step 340 includes:

[00315] receiving the uploading verification result returned by the third consensus node after the verification of the third consensus node is successful when the instantaneous-ness information indicates that the local transaction is an instantaneous transaction.

[00316] When this embodiment is specifically implemented, the instantaneous-ness information is generated in the following manner:

[00317] obtaining a child type of the local transaction of the first type; and

[00318] determining instantaneous-ness information according to the child type.

[00319] In this embodiment, the child type is obtained in a manner similar to the manner in which the child type is determined in the above embodiment.

[00320] The child type may be the foregoing invoice reimbursement type, invoicing type, resource exchange type, and resource statistics type. According to the instantaneous-ness requirements of each child type, a child type-to-instantaneous-ness information comparison table is made, for example:

Table 6

Child type	Instantaneous-ness information
------------	--------------------------------

Invoice reimbursement type	Instant
Invoicing type	Instant
Resource exchange type	Non-instant
Resource statistics type	Non-instant

[00321] As shown in Table 6, if the child type is the invoice reimbursement type or invoice, the instantaneous-ness requirement is high, and the instantaneous-ness information may be set to instant (so that the uploading verification result of the local transaction with the child type being the invoicing type can be instantly returned to the first relay node). For the resource exchange type or the resource statistics type, the instantaneous-ness requirement is lower than that for the invoice reimbursement type. Therefore, the instantaneous-ness information is set to non-instant.

[00322] After the comparison table is made, the comparison table may be looked up according to the child type to determine the instantaneous-ness information corresponding to the child type.

[00323] For example, the local transaction of the first type is "Electronic invoice reimbursement 0001, amount 500 for accommodation". Then the child type of the local transaction is first obtained as the invoice reimbursement type. Secondly, the above comparison table is looked up in response to the child type being the invoicing type to obtain the instant information being instant, which means that "Electronic invoice reimbursement 0001, amount 500 for accommodation" is an instantaneous transaction. Then, after the verification of the third consensus node is successful, the third consensus node immediately returns the uploading verification result to the first relay node, so as to improve the instantaneous-ness of the instantaneous transaction processing.

[00324] The advantage of this embodiment is that the uploading verification result is returned instantly for the instant local transaction, thus ensuring the instant processing of the instantaneous transaction.

[00325] In some embodiments, an uploading verification result request is transmitted to the third consensus node and the uploading verification result is obtained from a response of the third consensus node to the uploading verification result request when the instantaneous-ness information indicates that the local transaction is a non-instantaneous transaction.

[00326] For example, the local transaction of the first type is "100 first resources are exchanged with 10 second resources". Then the child type of the local transaction is first obtained

as the resource exchange type. Secondly, the above comparison table is looked up in response to the child type being the resource exchange type to obtain the instant information being non-instant, which means that "100 first resources are exchanged with 10 second resources" is a non-instantaneous transaction. In this case, after the verification of the third consensus node is successful, the third consensus node is not required to return the uploading verification result to the first relay node so urgently, and may wait for the first relay node to transmit the uploading verification result request to the third consensus node and make a response to the request. The first relay node obtains the uploading verification result from the response of the third consensus node to the uploading verification result request.

[00327] The advantage of this embodiment is that a non-instant local transaction waits for the request of the first relay node, thus reducing the occupation of transmission resources.

[00328] The first relay node transmits an uploading verification result request to the third consensus node, so as to obtain the uploading verification result from a response of the third consensus node to the uploading verification result request when the instantaneous-ness information indicates that the local transaction is a non-instantaneous transaction. Next, the uploading verification result request is described.

[00329] Reference is made to FIG. 18. The uploading verification result request shown in FIG. 18 includes a block identifier, a source address, and a destination address. The source address refers to an address of the first relay node and the destination address refers to an address of the third consensus node. The block identifier is configured to indicate a block where the uploading verification result is located (the block is located on the third blockchain maintained by the third consensus node).

[00330] The block identifier is generated by the third consensus node and transmitted by the third consensus node to the first relay node. For example, step 330d includes that: the third consensus node verifies the uploading of the local transaction according to the first uploading verification information and generates an uploading verification result; and the third consensus node records the uploading verification result to the third blockchain and returns the block identifier recorded to the third blockchain to the first relay node.

[00331] In the specific implementation of this embodiment, before the third consensus node transmits the block identifier to the first relay node, the third consensus node generates a block according to the uploading verification result and records the block to the third blockchain. The block includes a block body and a block header. The block body is divided into multiple blocks. The uploading verification result may only occupy one block, multiple blocks, or all blocks. When

only one or multiple blocks are occupied, the remaining blocks record uploading verification results of local transactions of other third consensus nodes. That is, the third consensus node packages the blocks occupied by the uploading verification results and the blocks corresponding to the uploading verification results indicating that other third consensus nodes need to upload into the block body, adds the block header to form a block, and records the block on the blockchain. The block header indicates relevant information of this block, including: a version number, a feature value, a timestamp, a difficulty value, and the like. The block identifier is an identifier that can distinguish a block from other blocks, such as the height of the block and the feature value of the block. The feature value of the block is generally the digest value generated by the block body of the block via the digest algorithm. As the block bodies are different, the digest values are different. Therefore, the digest values can uniquely distinguish the blocks. The height of the block refers to the position of the block recorded on the blockchain. The height of the block can also uniquely distinguish the block from other blocks. After the block is recorded to the third blockchain, the block identifier recorded to the third blockchain is returned to the first relay node.

[00332] After the first relay node receives the block identifier, in some embodiments, step 340 includes: transmitting an uploading verification result request to the third consensus node, the uploading verification result request having the block identifier, whereby the third consensus node obtains the uploading verification result from the third blockchain according to the block identifier; and receiving the uploading verification result returned by the third consensus node. The advantage is that the third consensus node records the uploading verification results to the third blockchain, which is convenient for other nodes to check, and the uploading verification results are not easy to be tampered, thus improving the security of the uploading verification results.

[00333] The manner of transmitting the uploading verification result request includes, but is not limited to, the following three cases:

[00334] (1) After the first relay node receives the block identifier, the first relay node immediately transmits an uploading verification result request to the third consensus node.

[00335] (2) The first relay node presets a second transmitting period, and after the block identifier is received and the second transmitting period has passed, the first relay node transmits an uploading verification result request to the third consensus node.

[00336] (3) After the first relay node receives the block identifier, the first relay node transmits a prompt message "whether to obtain a first uploading verification result" to the local consensus network. When receiving indication information "yes" returned by the first consensus

node, the first relay node transmits an uploading verification result request to the third consensus node in response to the indication information returned by the first consensus node.

[00337] Considering generalized uploading requirements of global transactions

[00338] According to the transaction uploading method in the above embodiment, the local transaction is executed and recorded to the corresponding blockchain according to the parent type of the local transaction only for the uploading demand of the local transaction. For example, the second consensus node records the local transaction of the first type to the second blockchain, and the second blockchain is not accessible by nodes other than the second consensus node. For another example, the first consensus node records the local transaction of the second type to the first blockchain, and the first blockchain can be accessed by nodes other than the first consensus node. That is, the local transactions of the second type are recorded on the first consensus node in the local consensus network. However, in the blockchain hierarchical consensus system, the local transactions recorded by each local consensus network are required to be summarized into the global consensus network, and the generalized global transactions formed by the local transactions of the second type will be recorded on the third blockchain by the second global consensus network in the global consensus network. Therefore, in one embodiment, after the local transaction is executed by the first consensus node and recorded to the first blockchain in response to determining that the parent type of the local transaction is the second type, it is further necessary to record the generalized global transaction generated for the local transaction to the third blockchain.

[00339] As shown in FIG. 19, after step 320, the transaction uploading method according to one embodiment of this application may further include the following steps:

[00340] Step 1910: Obtain a global transaction generalized from multiple local transactions of the second type.

[00341] Step 1920: Transmit the global transaction to the third consensus node, the global transaction being used for instructing the third consensus node to execute the global transaction and to record the global transaction to a third blockchain.

[00342] Steps 1910 to 1920 are described in detail below.

[00343] In step 1910, a global transaction may be generalized from multiple local transactions in a local consensus network or from local transactions in multiple local consensus networks.

[00344] The manner of generalizing the global transaction includes the following:

[00345] (1) After the first consensus node receives a predetermined number of local transactions, a global transaction is generated according to the predetermined number of local transactions.

[00346] (2) When a time difference between a current time of the first consensus node and a previous transmitting time of the generalized global transaction exceeds a threshold, a new global transaction is generated according to the received local transaction.

[00347] (3) After the first consensus node receives the predetermined number of local transactions, the global transaction is generated according to the predetermined number of local transactions. If the predetermined number is not reached but when the time difference between the current time of the first consensus node and the previous transmitting time of the generalized global transaction exceeds the threshold, the global transaction is generated according to the received local transaction.

[00348] For example, it is assumed that there are three local transactions of the second type: "Order 10001: purchase 50 office supplies", "Order 10002: purchase 150 office supplies", and "Order 10003: purchase 200 office supplies". According to the three local transactions, the generalized global transaction may be "The total number of orders is 3" or "The total number of orders purchased is 400 office supplies".

[00349] With continued reference to FIG. 19, step 1920 may be specifically divided into 1920a: transmitting the generalized global transaction to the third consensus node, and 1920b (executed by the third consensus node): executing the generalized global transaction and recording the generalized global transaction to the third blockchain.

[00350] In step 1920a, the manner in which the first consensus node transmits the generalized global transaction includes the following:

[00351] (1) After the first consensus node obtains the local transaction, the generalized global transaction is immediately transmitted to the third consensus node.

[00352] (2) The first consensus node presets a third transmitting period, and after the first consensus node obtains the local transaction and the third transmitting period has passed, the first consensus node transmits the generalized global transaction to the third consensus node.

[00353] In step 1920b, the third consensus node executes the generalized global transaction and records the generalized global transaction to the third blockchain. The third blockchain records the generalized global transaction and can be accessed by nodes other than the third consensus node.

[00354] The specific process of executing the local transaction by the third consensus node and recording the local transaction to the third blockchain may be referred to the description of executing the local transaction by the first consensus node and recording the local transaction to the first blockchain in the above embodiment.

[00355] The advantage of the above embodiment is that based on the hierarchical consensus system architecture (as shown in FIG. 10), each local consensus network may be guaranteed to run local transactions independently, which is convenient for hierarchical service governance. It is unnecessary to collect all the traffic of the service alliance in the whole witness network into the global consensus network. On the premise of data isolation, service security isolation, and legal compliance, steps 1910 to 1920 can ensure that the second global consensus network is not required to carry a large number of subdivided services, and may focus on the governance of key tasks and global transactions.

[00356] In addition, in the above embodiment, after the generalized global transaction is recorded to the third blockchain by the third consensus node, the first relay node may obtain second uploading verification information from the third consensus node and transmit the second uploading verification information to the first consensus node, whereby the second uploading verification information is verified by the first consensus node to ensure that the generalized global transaction has been truly executed and truly uploaded.

[00357] As shown in FIG. 20, after step 1920, the transaction uploading method according to one embodiment of this application may further include the following steps:

[00358] Step 2010: Receive, when the third consensus node records the global transaction to the third blockchain to generate second uploading verification information, the second uploading verification information from the third consensus node.

[00359] Step 2020: Transmit the second uploading verification information to the first consensus node, the second uploading verification information being used for instructing the first consensus node to verify uploading of the generalized global transaction.

[00360] For example, it is assumed that the second uploading verification information includes a digest value of the generalized global transaction and a digest value recorded in the third bypass node. The third bypass node is obtained by the third consensus node in the following manner.

[00361] A third tree is generated. A leaf node of the third tree records a digest value of a transaction recorded on the third blockchain. The recorded transaction includes the generalized

global transaction. In two adjacent layers in the third tree, a node of the previous layer is generated by at least two nodes of the next layer, and the same node in the next layer is only configured to generate one node in the previous layer.

[00362] A third bypass node of a third path is determined. The third path is a path between the leaf node corresponding to the digest value of the generalized global transaction in the third tree and a root node in the third tree. The third bypass node is a sibling node of nodes in the third tree other than the root node in the third path.

[00363] Correspondingly, step 2020 includes:

[00364] determining a third comparative digest value according to the digest value of the generalized global transaction and the digest value recorded in the third bypass node; and

[00365] determining that the verification is successful if the third comparative digest value is consistent with the digest value recorded in the root node in the third tree.

[00366] A specific method of verifying according to the second uploading verification information, reference is referred to a specific method of verifying according to the first uploading verification information.

[00367] Through steps 2010 to 2020, this embodiment of this application ensures that the first consensus node can accurately verify the generalized global transaction, and can verify that the generalized global transaction is truly recorded without cheating.

[00368] Implementation Detail Diagram of Transaction Uploading Method of this Embodiment of this Application

[00369] Referring to FIG. 21 the implementation details of the transaction uploading method of this embodiment of this application will be illustrated in detail.

[00370] In step 310, the first relay node determines that a parent type of a local transaction is a first type or determines that the parent type of the local transaction is a second type.

[00371] In step 320, in response to determining that the parent type of the local transaction is the second type, the local transaction is executed by the first consensus node and recorded to the first blockchain.

[00372] Step 330 is divided into steps 330a to 330d. In step 330a, the first relay node places the local transactions in a queue for queuing (step 410), and packages and transmits the local transactions in the queue to the second consensus node in response to the queue satisfying a predetermined condition (step 420). In step 330b, the second consensus node executes the local

transaction, and records the local transaction to the second blockchain. In step 330c, a first tree is generated in step 1110. A first bypass node of a first path is determined in step 1111. A digest value of the local transaction and a digest value recorded in the first bypass node are transmitted to the third consensus node in step 1112. In step 330d, a first comparative digest value is determined according to the digest value of the local transaction and the digest value recorded in the first bypass node (step 1120). The verification is successful when the digest value recorded in the root node in the first tree and the first comparative digest value are consistent (step 1121).

[00373] In step 340, the uploading verification result of the local transaction is obtained from the third consensus node. For example, when the digest value recorded in the root node in the first tree is consistent with the first comparative digest value, the uploading verification result is verification success. When the digest value recorded in the root node in the first tree is inconsistent with the first comparative digest value, the uploading verification result is verification failure.

[00374] In addition to local transactions required to be uploaded, there are also generalized global transactions required to be recorded to the third blockchain. Therefore, with continued reference to FIG. 21, the generalized global transaction generated for the local transaction is obtained in step 1910. The generalized global transaction is transmitted to the third consensus node (1920a) in step 1920. The third consensus node executes the generalized global transaction and records the generalized global transaction to the third blockchain (1920b). Next, after step 1920, second uploading verification information of the generalized global transaction is received from the third consensus node in step 2010. The second uploading verification information is transmitted to the first consensus node in step 2020a, and the first consensus node verifies uploading of the generalized global transaction in step 2020b.

[00375] The advantages of this embodiment include, but are not limited to: it can be distinguished whether a local transaction is a first type or a second type, and an uploading security requirement of a target transaction is met; by packaging and transmitting multiple local transactions together, the network overhead can be reduced; by using a digest value and a digest value recorded in a bypass node as first uploading verification information, the uploading authenticity of local transactions can be verified and the leakage of other related transaction true information can be reduced; by executing and recording a generalized global transaction by a third consensus node to a third blockchain, the third consensus node can process the global transaction more centrally; and the uploading authenticity of the generalized global transaction can be verified through second uploading verification information, thus improving the security.

[00376] The first relay node is mainly used as a core processing module in the above embodiment, and the second consensus node is used as a core processing module to describe the transaction uploading method of another embodiment.

[00377] Referring to FIG. 22, a transaction uploading method performed by a second consensus node includes the following steps:

[00378] Step 2210: Receive a local transaction forwarded by a first relay node, a parent type of the local transaction being a first type.

[00379] The local transaction comes from a local consensus network.

[00380] Step 2220: Execute the local transaction, and record the local transaction to a second blockchain, the second blockchain being not accessible by nodes other than the second consensus node.

[00381] Step 2230: Obtain first uploading verification information of the local transaction.

[00382] Step 2240: Transmit the first uploading verification information to a third consensus node in a second global consensus network (step 2240a), the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result (step 2240b).

[00383] In this embodiment, for the local transaction of the first type, the local transaction is transmitted to the second consensus node to instruct the second consensus node to record the local transaction to the second blockchain. Since the second blockchain is not allowed to be accessed by nodes other than the second consensus node, the security of the local transaction of the first type is guaranteed. In the meantime, after completing the uploading of the local transaction, the second consensus node transmits the first uploading verification information after the uploading to the third consensus node. The third consensus node performs uploading verification to obtain an uploading verification result, and the uploading verification result is obtained from the third consensus node, thus ensuring that the uploading of the local transaction is authentic although the local transaction cannot be viewed by the outside world, thereby improving the uploading security of the local transaction.

[00384] In some embodiments, the second consensus node has a TEE. The second blockchain is encrypted with a key of the TEE. The process of transmitting the first uploading verification information to the third consensus node includes: receiving a first uploading verification information request transmitted by the second relay node; and transmitting the first uploading

verification information to the second relay node in response to the first uploading verification information request. The first uploading verification information is used for instructing the second relay node to forward the first uploading verification information to the third consensus node.

[00385] Specifically, the second relay node transmits the first uploading verification information request to the second consensus node in response to adding a block encrypted with the key of the TEE to the second blockchain. The second consensus node transmits the first uploading verification information to the second relay node in response to the first uploading verification information request transmitted by the second relay node. The second relay node forwards the first uploading verification information to the third consensus node in the second global consensus network.

[00386] In this embodiment, the local transactions recorded by the second blockchain on the first global consensus network are not visible to the first relay node, and therefore, key encryption in the TEE is employed. However, a block added to the second blockchain is visible. Therefore, the second relay node may identify whether there is an additional block encrypted with the key in the second blockchain. After discovering that there is an additional block encrypted with the key in the second blockchain, the first uploading verification information request is transmitted to the second consensus node. After receiving the first uploading verification information request, the second consensus node cannot directly return the first uploading verification information to the first relay node, but returns the first uploading verification information to the second relay node, which is forwarded to the third consensus node by the second relay node. Thus, the third consensus node verifies according to the first uploading verification information.

[00387] In this embodiment of this application, the block is encrypted with the key of the TEE, and the first uploading verification information is forwarded by the second relay node, thereby further improving the uploading security of the target transaction of the first global consensus network.

[00388] Description of Apparatus and Device of this Embodiment of this Application

[00389] It is to be appreciated that although the various steps in the flowcharts are shown in sequence as indicated by the arrows, these steps are not necessarily performed in the order indicated by the arrows. These steps are performed in no strict order unless explicitly stated in this embodiment, and these steps may be performed in other orders. Moreover, at least some of the steps in the flowcharts may include multiple steps or multiple stages. These steps or stages are not necessarily performed at the same time, but may be performed at different times. These steps or

stages are not necessarily performed in sequence, but may be performed in turn or in alternation with other steps or at least some of the steps or stages in other steps.

[00390] In each specific implementation of this application, when data related to the characteristics of a target object, such as attribute information or attribute information set of the target object, needs to be processed, permission or consent of the target object will be obtained first, and the collection, use, and processing of these data will comply with relevant laws, regulations, and standards of relevant countries and regions. Furthermore, when this embodiment of this application needs to obtain the attribute information of the target object, the individual permission or individual consent of the target object is obtained through pop-up or jumping to a confirmation page, etc. After explicitly obtaining the individual permission or individual consent of the target object, necessary target object related data for enabling this embodiment of this application to operate normally is obtained.

[00391] FIG. 23 is a schematic structural diagram of a transaction uploading apparatus 2300 according to an embodiment of this application. The transaction uploading apparatus 2300 includes the following units.

[00392] A first determining unit 2310 is configured to determine, when a first consensus node receives a local transaction, a parent type of the local transaction.

[00393] A first transmitting unit 2330 is configured to transmit the local transaction to a second consensus node when the parent type of the local transaction is a first type. The local transaction is used for instructing the second consensus node to execute the local transaction for recording the local transaction to a second blockchain, generating first uploading verification information of the local transaction, and transmitting the first uploading verification information of the local transaction to a third consensus node. The first uploading verification information is used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result of the local transaction. The second blockchain is not accessible by nodes other than the second consensus node.

[00394] A first obtaining unit 2340 is configured to obtain the uploading verification result of the local transaction from the third consensus node.

[00395] In some embodiments, the first consensus node belongs to a local consensus network. The second consensus node belongs to a first global consensus network in global consensus networks. The third consensus node belongs to a second global consensus network in the

global consensus networks. The first relay node is configured to transmit the local transaction of the first type to the second consensus node in the first global consensus network.

[00396] In some embodiments, the transaction uploading apparatus 2300 further includes a first uploading unit 2320, configured to instruct, when the parent type is a second type, the first consensus node to execute the local transaction and to record the local transaction to a first blockchain. The first blockchain is allowed to be accessed by nodes other than the first consensus node.

[00397] In some embodiments, the transaction uploading apparatus 2300 further includes the following units.

[00398] A third obtaining unit is configured to obtain a global transaction generalized from multiple local transactions of the second type.

[00399] A third transmitting unit is configured to transmit the global transaction to the third consensus node. The global transaction is used for instructing the third consensus node to execute the global transaction and to record the global transaction to a third blockchain. The third blockchain is allowed to be accessed by nodes other than the third consensus node.

[00400] In some embodiments, the transaction uploading apparatus 2300 further includes the following units.

[00401] A second receiving unit is configured to receive, when the third consensus node records the global transaction to the third blockchain to generate second uploading verification information, the second uploading verification information from the third consensus node.

[00402] A third transmitting unit is configured to transmit the second uploading verification information to the first consensus node. The second uploading verification information is used for instructing the first consensus node to verify uploading of the global transaction.

[00403] In some embodiments, the first uploading verification information includes a digest value of the local transaction and a digest value that is determined according to a first path in a first tree and recorded in a first bypass node. A leaf node in the first tree records a digest value of a transaction recorded on the second blockchain. The first path is a path from a leaf node in the first tree to a root node in the first tree, and the leaf node in the first tree records the digest value of the local transaction. The first bypass node is a sibling node in the first tree of nodes other than the root node in the first path. The first uploading verification information is further used for instructing the third consensus node to determine a first comparative digest value according to the digest value of the local transaction and the digest value recorded in the first bypass node. The verification is

successful when the digest value recorded in the root node in the first tree and the first comparative digest value are consistent.

[00404] In some embodiments, the first uploading verification information further includes a digest value of associated data obtained after execution of the local transaction and a digest value recorded in a second bypass node in a second tree. A leaf node in the second tree records a digest value of associated data obtained after execution of a transaction recorded on the second blockchain. A second path is a path from a leaf node in the second tree to a root node in the second tree, and the leaf node in the second tree records the digest value of the associated data obtained after execution of the local transaction. The second bypass node is a sibling node in the second tree of nodes other than the root node on the second path. The first uploading verification information is further used for instructing the third consensus node to determine, when the digest value recorded in the root node in the first tree and the first comparative digest value are consistent, a second comparative digest value according to the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node. The verification is successful when the digest value recorded in the root node in the second tree and the second comparative digest value are consistent.

[00405] In some embodiments, the first uploading verification information further includes transaction desensitization information. The transaction desensitization information is obtained by desensitizing the local transaction by the second consensus node. The desensitization is performed according to a desensitization field in a target contract. The target contract corresponds to a child type of the local transaction.

[00406] The first obtaining unit 2340 is further specifically configured to:

[00407] obtain the transaction desensitization information from the third consensus node as the uploading verification result.

[00408] In some embodiments, the first uploading verification information further includes instantaneous-ness information of the local transaction. The instantaneous-ness information is determined by the second consensus node according to a child type of the local transaction.

[00409] The first obtaining unit 2340 is further specifically configured to:

[00410] receive the uploading verification result returned by the third consensus node after the verification of the third consensus node is successful when the instantaneous-ness information indicates that the local transaction is an instantaneous transaction.

[00411] In some embodiments, the first obtaining unit 2340 is further configured to transmit an uploading verification result request to the third consensus node and obtain the uploading verification result from a response of the third consensus node to the uploading verification result request when the instantaneous-ness information indicates that the local transaction is a non-instantaneous transaction.

[00412] In some embodiments, the first uploading verification information further includes a first signature of the digest value recorded in the second bypass node. The first signature is obtained by encrypting a first digest by the second consensus node with a first private key of the second consensus node. The first digest is generated by the second consensus node according to the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node. The first signature is used for instructing the third consensus node to decrypt, when the root node in the second tree is consistent with a second to-be-compared root node, the first signature with a first public key of the second consensus node to obtain a second digest, and to generate a third digest according to the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node. The verification is successful when the third digest and the second digest are consistent.

[00413] In some embodiments, the transaction uploading apparatus 2300 is further configured to receive a block identifier returned by the third consensus node after the third consensus node records a block including the uploading verification result to a third blockchain. The block identifier refers to the block including the uploading verification result.

[00414] The first obtaining unit 2340 is further configured to:

[00415] transmit an uploading verification result request to the third consensus node, the uploading verification result request carrying the block identifier, and the uploading verification result request being used for instructing the third consensus node to obtain the uploading verification result from the third blockchain according to the block identifier; and receive the uploading verification result returned by the third consensus node.

[00416] In some embodiments, the first transmitting unit 2330 is further configured to:

[00417] place the local transaction in a queue for queuing; and

[00418] package local transactions in the queue in response to the queue satisfying a predetermined condition, and transmit the local transaction to the second consensus node.

[00419] In some embodiments, the first transmitting unit 2330 is further specifically configured to:

[00420] package local transactions in the queue in response to the number of local transactions in the queue reaching a first threshold.

[00421] In some embodiments, the first transmitting unit 2330 is further specifically configured to: determine a time difference between a current time and a previous packaging time in response to the number of local transactions in the queue not reaching a first threshold; and

[00422] package local transactions in the queue in response to the time difference reaching a second threshold.

[00423] In some embodiments,

[00424] the first transmitting unit 2330 is further specifically configured to:

[00425] determine a child type of the local transaction; determine a queue corresponding to the child type of the local transaction; and place the local transaction in the queue corresponding to the child type for queuing.

[00426] In some embodiments, the first transmitting unit 2330 is further configured to: package local transactions in the queue in response to the number of local transactions in the queue reaching a third threshold.

[00427] In some embodiments, the first transmitting unit 2330 is further specifically configured to:

[00428] determine a child type of the local transaction;

[00429] determine a priority of the local transaction according to the child type;

[00430] place the local transaction in a queue for queuing in a descending order of priority; and

[00431] sequentially take out the local transactions from the queue, and transmit the local transactions to the second consensus node.

[00432] In some embodiments, the second consensus node has a TEE. The second blockchain is encrypted with a key of the TEE. The first uploading verification information is transmitted by the second consensus node to the second relay node in response to a first uploading verification information request transmitted by the second relay node. The first uploading verification information request is transmitted by the second relay node in response to adding an encrypted

block on the second blockchain. The encrypted block is encrypted with the key of the TEE. The first uploading verification information is used for instructing the second relay node to forward the first uploading verification information to the third consensus node.

[00433] FIG. 24 is a schematic structural diagram of a transaction uploading apparatus 2400 according to an embodiment of this application. The transaction uploading apparatus 2400 includes the following units.

[00434] A first receiving unit 2410 is configured to receive a local transaction forwarded by a first relay node. A parent type of the local transaction is a first type.

[00435] A second uploading unit 2420 is configured to execute the local transaction, and record the local transaction to a second blockchain. The second blockchain is not accessible by nodes other than the second consensus node.

[00436] A second obtaining unit 2430 is configured to obtain first uploading verification information of the local transaction.

[00437] A second transmitting unit 2440 is configured to transmit the first uploading verification information to a third consensus node. The first uploading verification information is used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result.

[00438] In some embodiments, the second consensus node has a TEE. The second blockchain is encrypted with a key of the TEE. The second transmitting unit 2440 is further configured to: receive a first uploading verification information request transmitted by a second relay node, the first uploading verification information request being transmitted by the second relay node in response to adding an encrypted block on the second blockchain, and the encrypted block being encrypted with the key of the TEE; and transmit the first uploading verification information to the second relay node in response to the first uploading verification information request, the first uploading verification information being used for instructing the second relay node to forward the first uploading verification information to the third consensus node.

[00439] Reference is made to FIG. 25. FIG. 25 is a block diagram of a partial structure of a terminal implementing a transaction uploading method according to an embodiment of this application. The terminal includes: a radio frequency (RF) circuit 2510, a memory 2515, an input unit 2530, a display unit 2540, a sensor 2550, an audio circuit 2560, a wireless fidelity (WiFi) module 2570, a processor 2580, a power supply 2590, and other components. A person skilled in the art may understand that the structure of the terminal shown in FIG. 25 does not constitute a

limitation on a mobile phone or a computer, and the mobile phone may include more components or fewer components than those shown in the figure, or some components may be combined, or a different component deployment may be used.

[00440] The RF circuit 2510 may be configured to receive and transmit signals during an information receiving and transmitting process or a call process. Specifically, downlink information from a base station is received and then delivered to the processor 2580 for processing. In addition, designed uplink data is transmitted to the base station.

[00441] The memory 2515 may be configured to store a software program and module. The processor 2580 runs the software program and module stored in the memory 2515, to implement various functional applications and data processing of the terminal object.

[00442] The input unit 2530 may be configured to receive input digit or character information, and generate a keyboard signal input related to the user setting and function control of the terminal object. Specifically, the input unit 2530 may include a touch panel 2531 and another input apparatus 2532.

[00443] The display unit 2540 may be configured to display input information or provided information, and various menus of the terminal object. The display unit 2540 may include a display panel 2541.

[00444] The audio circuit 2560, a loudspeaker 2561, and a microphone 2562 may provide audio interfaces.

[00445] In this embodiment, the processor 2580 included in the terminal may perform the transaction uploading method according to the foregoing embodiment.

[00446] The terminal according to this embodiment of this application includes, but is not limited to, a mobile phone, a computer, an intelligent voice interaction device, an intelligent household appliance, a vehicle-mounted terminal, an aircraft, and the like. This embodiment of the present invention may be applied to various scenarios, including but not limited to data security, blockchain, data storage, information technology, and the like.

[00447] FIG. 26 is a block diagram of a partial structure of a server implementing a transaction uploading method according to an embodiment of this application. The server may greatly differ due to different configurations or performances, and may include one or more CPUs 2622 (for example, one or more processors), a memory 2632, and one or more storage media 2130 for storing applications 2642 or data 2644 (for example, one or more mass storage apparatuses). The memory 2632 and the storage medium 2630 may be configured for transient storage or

permanent storage. A program stored in the storage medium 2630 may include one or more modules (which are not marked in the figure), and each module may include a series of instruction operations on the server. Furthermore, the CPU 2622 may be configured to communicate with the storage medium 2630, and perform, on the server, the series of instruction operations in the storage medium 2630.

[00448] The server may further include one or more power supplies 2626, one or more wired or wireless network interfaces 2650, one or more input/output interfaces 2658, and/or one or more operating systems 2641, for example, Windows ServerTM, Mac OS XTM, UnixTM, LinuxTM, FreeBSDTM, and the like.

[00449] The CPU 2622 in the server may be configured to performing the transaction uploading method according to this embodiment of this application.

[00450] This embodiment of this application further provides a computer-readable storage medium. The computer-readable storage medium is configured to store computer-readable instructions. The computer-readable instructions are used for performing the transaction uploading method according to each of the foregoing embodiments.

[00451] This embodiment of this application further provides a computer program product. The computer program product includes computer programs. A processor of a computer device reads and executes the computer programs, whereby the computer device implements the transaction uploading.

[00452] The terms "first", "second", "third", "fourth", and the like in the specification of this application and in the above drawings are used for distinguishing between similar objects and not necessarily for describing a particular order or sequential order. It is to be understood that such used data is interchangeable where appropriate whereby the embodiments of this application described here can be, for example, implemented in an order other than those illustrated or described here. Furthermore, the terms "include", "have", and any other variants mean to cover the non-exclusive inclusion. For example, a process, method, system, product, or apparatus that includes a list of steps or units is not necessarily limited to those steps or units, but may include other steps or units not expressly listed or inherent to such a process, method, product, or apparatus.

[00453] It is to be understood that "at least one" means one or more and "multiple" means two or more in this application. "And/or" describes an association relationship between associated objects, and represents that there are three relationships. For example, "A and/or B" may mean that there are only A, only B, and both A and B, where A and B may be singular or plural. The character

"/" generally represents that contextual objects are in an "or" relationship. "At least one of the following" or a similar expression thereof means any combination of these items, including any combination of single items or plural items. For example, at least one of a, b, or c may represent a, b, c, "a and b," "a and c," "b and c," or "a and b and c," where a, b, and c may be single or multiple.

[00454] It is to be understood that in the description of this embodiment of this application, the term multiple (or plural) means more than two, the terms greater than, less than, and exceed are understood to exclude the current number, and the terms above, below, and within are understood to include the current number.

[00455] In the several embodiments provided in this application, it is to be understood that the disclosed system, apparatus, and method may be implemented in other manners. For example, the apparatus embodiments described above are merely examples. For example, division into the units is merely logical function division, and may be another division in an actual implementation. For example, multiple units or assemblies may be combined or may be integrated into another system, or some features may be ignored or not be performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented by using some interfaces. The indirect couplings or communication connections between the apparatuses or units may be electrical, mechanical, or in other forms.

[00456] The units described as separate components may or may not be physically separated, and components displayed as units may or may not be physical units, may be located in one position, or may be distributed over multiple network units. Some or all of the units may be selected based on actual requirements to achieve the objects of the solutions of this embodiment.

[00457] In addition, functional units in the various embodiments of this application may be integrated into one processing unit, the units may exist alone physically, or two or more units may be integrated into one unit. The integrated unit may be implemented in the form of hardware, or may be implemented in the form of a software functional unit.

[00458] If the integrated unit is implemented in the form of a software functional unit and sold or used as an independent product, the integrated unit may be stored in a computer-readable storage medium. Based on this understanding, the technical solution of this application, either inherently or in any part contributing to the related art, or all or part of the technical solution, may be embodied in the form of a software product. The computer software product is stored in a storage medium, and includes several instructions for enabling a computer apparatus (which may be a personal computer, a server, or a network apparatus) to perform all or part of the steps of the method according to the various embodiments of this application. The foregoing storage medium

includes any medium capable of storing computer-readable instructions, such as a USB flash drive, a removable hard disk, a read-only memory (ROM), a random access memory (RAM), a magnetic disk, or an optical disc.

[00459] It is also to be understood that the various implementations provided by this embodiment of this application may be arbitrarily combined to achieve different technical effects.

[00460] The above is a specific description of the implementations of this application, but this application is not limited to the above implementations. A person skilled in the art may make various equivalent modifications or substitutions without violating the spirit of this application, and these equivalent modifications or substitutions are all included in the scope defined by the claims of this application.

CLAIMS

What is claimed is:

1. A transaction uploading method, performed by a first relay node, the method comprising:
determining, when a first consensus node receives a local transaction, a parent type of the local transaction;
transmitting the local transaction to a second consensus node when the parent type is a first type, the local transaction being used for instructing the second consensus node to execute the local transaction for recording the local transaction to a second blockchain, generating first uploading verification information of the local transaction, and transmitting the first uploading verification information to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result of the local transaction, and the second blockchain being not accessible by nodes other than the second consensus node; and
obtaining the uploading verification result of the local transaction from the third consensus node.
2. The method according to claim 1, wherein the first consensus node belongs to a local consensus network, the second consensus node belongs to a first global consensus network in global consensus networks, the third consensus node belongs to a second global consensus network in the global consensus networks, and the first relay node is configured to transmit the local transaction of the first type to the second consensus node in the first global consensus network.
3. The method according to claim 1 or 2, further comprising:
instructing, when the parent type is a second type, the first consensus node to execute the local transaction and to record the local transaction to a first blockchain, the first blockchain being accessible by nodes other than the first consensus node.
4. The method according to claim 3, further comprising:
obtaining a global transaction generalized from a plurality of local transactions of the second type; and
transmitting the global transaction to the third consensus node, the global transaction being used for instructing the third consensus node to execute the global transaction and to record the global transaction to a third blockchain, and the third blockchain being accessible by nodes other

than the third consensus node.

5. The method according to claim 4, further comprising:

receiving, when the third consensus node records the global transaction to the third blockchain to generate second uploading verification information, the second uploading verification information from the third consensus node; and

transmitting the second uploading verification information to the first consensus node, the second uploading verification information being used for instructing the first consensus node to verify uploading of the global transaction.

6. The method according to any one of claims 1 to 5, wherein the first uploading verification information comprises a digest value of the local transaction and a digest value that is determined according to a first path in a first tree and recorded in a first bypass node; a leaf node in the first tree records a digest value of a transaction recorded on the second blockchain; the first path is a path from a leaf node in the first tree to a root node in the first tree, and the leaf node in the first tree records the digest value of the local transaction; the first bypass node is a sibling node in the first tree of nodes other than the root node in the first path; the first uploading verification information is further used for instructing the third consensus node to determine a first comparative digest value according to the digest value of the local transaction and the digest value recorded in the first bypass node; and the verification is successful when the digest value recorded in the root node in the first tree and the first comparative digest value are consistent.

7. The method according to claim 6, wherein the first uploading verification information further comprises a digest value of associated data obtained after execution of the local transaction and a digest value recorded in a second bypass node in a second tree; a leaf node in the second tree records a digest value of associated data obtained after execution of a transaction recorded on the second blockchain; a second path is a path from a leaf node in the second tree to a root node in the second tree, and the leaf node in the second tree records the digest value of the associated data obtained after execution of the local transaction; the second bypass node is a sibling node in the second tree of nodes other than the root node on the second path; the first uploading verification information is further used for instructing the third consensus node to determine, when the digest value recorded in the root node in the first tree and the first comparative digest value are consistent, a second comparative digest value according to the digest value of the associated data obtained after execution of the local transaction and the digest value recorded in the second bypass node; and the verification is successful when the digest value recorded in the root node in the second tree and the second comparative digest value are consistent.

8. The method according to claim 7, wherein the first uploading verification information further comprises transaction desensitization information, the transaction desensitization information is obtained by desensitizing the local transaction by the second consensus node, the desensitization is performed according to a desensitization field in a target contract, and the target contract corresponds to a child type of the local transaction; and

the obtaining the uploading verification result of the local transaction from the third consensus node comprises:

obtaining the transaction desensitization information from the third consensus node as the uploading verification result.

9. The method according to claim 7, wherein the first uploading verification information further comprises instantaneous-ness information of the local transaction, and the instantaneous-ness information is determined by the second consensus node according to a child type of the local transaction; and

the obtaining the uploading verification result of the local transaction from the third consensus node comprises:

receiving the uploading verification result returned by the third consensus node after the verification of the third consensus node is successful when the instantaneous-ness information indicates that the local transaction is an instantaneous transaction.

10. The method according to claim 9, further comprising:

transmitting an uploading verification result request to the third consensus node and obtaining the uploading verification result from a response of the third consensus node to the uploading verification result request when the instantaneous-ness information indicates that the local transaction is a non-instantaneous transaction.

11. The method according to claim 7, wherein the first uploading verification information further comprises a first signature of the second bypass node, the first signature is obtained by encrypting a first digest by the second consensus node with a first private key of the second consensus node, and the first digest is generated by the second consensus node according to the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node; and

the first signature is used for instructing the third consensus node to decrypt, when the root node in the second tree is consistent with a second to-be-compared root node, the first signature

with a first public key of the second consensus node to obtain a second digest, and to generate a third digest according to the digest value of the local transaction, the digest value recorded in the first bypass node, the digest value of the associated data obtained after execution of the local transaction, and the digest value recorded in the second bypass node, and the verification is successful when the third digest and the second digest are consistent.

12. The method according to claim 1, further comprising:

receiving a block identifier returned by the third consensus node after the third consensus node records a block comprising the uploading verification result to a third blockchain, the block identifier referring to the block comprising the uploading verification result,

wherein the obtaining the uploading verification result of the local transaction from the third consensus node comprises:

transmitting an uploading verification result request to the third consensus node, the uploading verification result request carrying the block identifier, and the uploading verification result request being used for instructing the third consensus node to obtain the uploading verification result from the third blockchain according to the block identifier; and

receiving the uploading verification result returned by the third consensus node.

13. The method according to any one of claims 1 to 12, wherein the transmitting the local transaction to a second consensus node comprises:

placing the local transaction in a queue for queuing; and

packaging local transactions in the queue in response to the queue satisfying a predetermined condition, and transmitting the local transaction to the second consensus node.

14. The method according to claim 13, wherein the packaging local transactions in the queue in response to the queue satisfying a predetermined condition comprises:

packaging local transactions in the queue in response to the number of local transactions in the queue reaching a first threshold.

15. The method according to claim 13, wherein the packaging local transactions in the queue in response to the queue satisfying a predetermined condition comprises:

determining a time difference between a current time and a previous packaging time in response to the number of local transactions in the queue not reaching a first threshold; and

packaging local transactions in the queue in response to the time difference reaching a second

threshold.

16. The method according to claim 13, wherein the placing the local transaction in a queue for queuing comprises:

- determining a child type of the local transaction;
- determining a queue corresponding to the child type of the local transaction; and
- placing the local transaction in the queue corresponding to the child type for queuing.

17. The method according to claim 16, wherein the packaging local transactions in the queue in response to the queue satisfying a predetermined condition comprises:

packaging local transactions in the queue in response to the number of local transactions in the queue reaching a third threshold.

18. The method according to any one of claims 1 to 17, wherein the transmitting the local transaction to a second consensus node comprises:

- determining a child type of the local transaction;
- determining a priority of the local transaction according to the child type;
- placing the local transaction in a queue for queuing in a descending order of priority; and
- sequentially taking out the local transactions from the queue, and transmitting the local transactions to the second consensus node.

19. The method according to any one of claims 1 to 18, wherein the second consensus node has a trusted execution environment (TEE), and the second blockchain is encrypted with a key of the TEE; the first uploading verification information is transmitted by the second consensus node to the second relay node in response to a first uploading verification information request transmitted by the second relay node, the first uploading verification information request is transmitted by the second relay node in response to adding an encrypted block on the second blockchain, the encrypted block is encrypted with the key of the TEE, and the first uploading verification information is used for instructing the second relay node to forward the first uploading verification information to the third consensus node.

20. A transaction uploading method, performed by a second consensus node, the method comprising:

receiving a local transaction forwarded by a first relay node, a parent type of the local transaction being a first type;

executing the local transaction, and recording the local transaction to a second blockchain, the second blockchain being not accessible by nodes other than the second consensus node;

obtaining first uploading verification information of the local transaction; and

transmitting the first uploading verification information to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result.

21. The method according to claim 20, wherein the second consensus node has a trusted execution environment (TEE), and the second blockchain is encrypted with a key of the TEE; and

the transmitting the first uploading verification information to a third consensus node comprises:

receiving a first uploading verification information request transmitted by a second relay node, the first uploading verification information request being transmitted by the second relay node in response to adding an encrypted block on the second blockchain, and the encrypted block being encrypted with the key of the TEE; and

transmitting the first uploading verification information to the second relay node in response to the first uploading verification information request, the first uploading verification information being used for instructing the second relay node to forward the first uploading verification information to the third consensus node.

22. A transaction uploading apparatus, comprising:

a first determining unit, configured to determine, when a first consensus node receives a local transaction, a parent type of the local transaction;

a first transmitting unit, configured to transmit the local transaction to a second consensus node when the parent type is a first type, the local transaction being used for instructing the second consensus node to execute the local transaction for recording the local transaction to a second blockchain, generating first uploading verification information of the local transaction, and transmitting the first uploading verification information of the local transaction to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result of the local transaction, and the second blockchain being not accessible by nodes other than the second consensus node; and

a first obtaining unit, configured to obtain the uploading verification result of the local transaction from the third consensus node.

23. A transaction uploading apparatus, comprising:

a first receiving unit, configured to receive a local transaction forwarded by a first relay node, a parent type of the local transaction being a first type;

a second uploading unit, configured to execute the local transaction, and record the local transaction to a second blockchain, the second blockchain being not accessible by nodes other than the second consensus node;

a second obtaining unit, configured to obtain first uploading verification information of the local transaction; and

a second transmitting unit, configured to transmit the first uploading verification information to a third consensus node, the first uploading verification information being used for instructing the third consensus node to verify uploading of the local transaction according to the first uploading verification information to obtain an uploading verification result.

24. An electronic device, comprising a memory and one or more processors, the memory storing computer-readable instructions, the processor, when executing the computer-readable instructions, implementing the transaction uploading method according to any one of claims 1 to 21.

25. One or more readable storage media, storing computer-readable instructions, the computer-readable instructions, when executed by one or more processors, implementing the transaction uploading method according to any one of claims 1 to 21.

26. A computer program product, comprising computer-readable instructions, the computer-readable instructions being read and executed by one or more processors of a computer device, whereby the computer device performs the transaction uploading method according to any one of claims 1 to 21.