CLAIMS

1. A method of transaction verification that is performed by a light node in a blockchain network, the method comprising:

acquiring a to-be-verified transaction transmitted by a user node in the blockchain network and transaction verification information corresponding to the to-be-verified transaction, the transaction verification information being determined by a full node in the blockchain network by executing a transaction service corresponding to the to-be-verified transaction, the transaction verification information including a state read set that includes state data associated with the to-be-verified transaction, a state write set that includes state data associated with the transaction service, an initial transaction execution result of the transaction service, and a block identifier of a target block with a maximum generation timestamp on a full blockchain of the full node, a first block header of the target block including the block identifier, the maximum generation timestamp indicating that the target block is a latest generated block in the full blockchain;

determining a second block header from a block header chain of the light node in the blockchain network, the second block header being matched with the block identifier in the first block header;

checking the state read set and the state write set based on a first state snapshot in the second block header to obtain a first check result of the to-be-verified transaction;

when the first check result indicates a successful check, executing the transaction service based on the state read set to obtain a target transaction execution result corresponding to the transaction service and to-be-written state data that is state data required to be written to a new block, and recording the to-be-written state data; and

checking legitimacy of the to-be-verified transaction based on the initial transaction execution result, the target transaction execution result, the state write set, and the to-be-written state data.

2. The method according to claim 1, wherein the state read set and the state write set are generated by the full node based on a Merkle Patricia tree (MPT) in the target block; and

the determining the second block header comprises:

determining a state snapshot in the second block header as the first state snapshot in the second block header;

acquiring a first proof of existence of first state data from the state read set;

checking the first proof of existence based on the first state snapshot to obtain a first proof check result, the first state data being state data obtained by the full node from the MPT of the target block and associated with the to-be-verified transaction;

acquiring a proof of modifiability of second state data from the state write set;

checking the proof of modifiability based on the first state snapshot to obtain a second proof check result, the second state data being state data that is recorded by the full node when the full node simulates to execute

the transaction service and written in a block subsequent to the target block; and

determining the first check result of the to-be-verified transaction based on the first proof check result and the second proof check result.

3. The method according to claim 2, wherein:

the acquiring the first proof of existence of the first state data from the state read set, and checking the first proof of existence based on the first state snapshot to obtain the first proof check result comprises:

acquiring the first proof of existence of the first state data that is a node path acquired by the full node from the MPT in the target block and associated with the first state data, the node path including a leaf node for storing the first state data, and a root node;

setting the leaf node as a child node and setting a node preceding the leaf node as a parent node corresponding to the child node based on a first node index relationship indicated by the node path;

comparing a hash value of the child node with a hash pointer of the parent node to obtain an initial comparison result;

in response to the initial comparison result indicating that the hash value being consistent with the hash pointer, updating the child node and the parent node based on the first node index relationship, and obtaining a first comparison result of the first proof of existence when the updated parent node is the root node;

comparing a hash value of the root node with the first state snapshot to obtain a second comparison result of the first proof of existence; and

obtaining the first proof check result based on the first comparison result and the second comparison result.

4. The method according to claim 1, wherein the checking the legitimacy of the to-be-verified transaction comprises:

comparing the initial transaction execution result with the target transaction execution result to obtain a third comparison result;

acquiring a second proof of existence of second state data from the state write set, and checking the second proof of existence based on the first state snapshot;

in response to the second proof of existence being verified, comparing the second state data with the to-be-written state data to obtain a fourth comparison result;

obtaining a second check result of the to-be-verified transaction based on the third comparison result and the fourth comparison result; and

checking the legitimacy of the to-be-verified transaction based on the second check result.

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

generating, based on second state data in the state write set, a second state snapshot associated with the

second state data in response to the legitimacy of the to-be-verified transaction being verified;

generating a to-be-processed block header based on the to-be-verified transaction, the target transaction execution result and the second state snapshot, the to-be-processed block header being applied as a block header subsequent to the second block header; and

writing the to-be-processed block header in the block header chain.

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

in response to the to-be-processed block header being written in the block header chain, acquiring a node public key of the full node, and encrypting the to-be-processed block header to obtain encrypted data information; and

transmitting the encrypted data information to the full node such that the full node decrypts the encrypted data information based on a node private key of the full node to obtain the to-be-processed block header.

7. The method according to claim 1, wherein the block identifier includes one of a block height and a hash value of the target block.

8. A method of transaction verification that is performed by a full node in a blockchain network, the method comprising:

receiving a to-be-verified transaction transmitted by a user node in the blockchain network;

setting a block with a maximum generation timestamp as a target block and setting a block header of the target block as a first block header on a full blockchain of the full node, the full blockchain including a plurality of blocks and the maximum generation timestamp indicating that the target block is a latest generated block of the plurality of blocks in the full blockchain;

acquiring a Merkle Patricia tree (MPT) in the target block, and executing a transaction service corresponding to the to-be-verified transaction based on the MPT in the target block to obtain an initial transaction execution result, a state read set and a state write set corresponding to the transaction service;

acquiring a block identifier in the first block header, and determining transaction verification information corresponding to the to-be-verified transaction, the transaction verification information including the initial transaction execution result, the state read set, the state write set, and the block identifier; ~~and~~

transmitting the transaction verification information to the user node in the blockchain network; and

generating a new block in the blockchain network based on a to-be-processed block header transmitted by a light node and the to-be-verified transaction.

9. The method according to claim 8, wherein the setting comprises:

receiving transaction signature information that corresponds to the to-be-verified transaction and is transmitted by the user node in the blockchain network, the transaction signature information being generated

the user node by signing the to-be-verified transaction based on a user private key;

acquiring a user public key corresponding to the user private key, and verifying the transaction signature information based on the user public key to obtain a verification result; and

acquiring the block with the maximum generation timestamp on the full blockchain of the full node in response to the verification result indicating that the transaction signature information is verified.

10. The method of claim 9, wherein the verifying the transaction signature information further comprises:

verifying the transaction signature information based on the user public key to obtain a first summary information of the to-be-verified transaction;

performing a hash calculation on the to-be-verified transaction to obtain a second summary information of the to-be-verified transaction; and

comparing the first summary information with the second summary information to obtain the verification result.

11. The method according to claim 8, wherein the acquiring the MPT in the target block comprises:

acquiring the MPT in the target block, setting the acquired MPT as a to-be-processed tree, retrieving state data associated with the to-be-verified transaction from the to-be-processed tree, and setting the state data as first state data;

executing the transaction service corresponding to the to-be-verified transaction based on the first state data, obtaining the initial transaction execution result corresponding to the transaction service, recording state data that is written in a block subsequent to the target block when the full node simulates to execute the transaction service, and setting the recorded state data as second state data;

collecting a first proof of existence of the first state data based on the to-be-processed tree, and generating the state read set based on the first proof of existence; and

collecting a proof of modifiability and a second proof of existence of the second state data based on the to-be-processed tree, and generating the state write set based on the proof of modifiability and the second proof of existence.

12. The method according to claim 11, wherein the collecting the proof of modifiability and the second proof of existence of the second state data comprises:

acquiring a key string associated with the second state data before the second state data is inserted into the to-be-processed tree, and acquiring the proof of modifiability of the second state data based on the key string;

acquiring the second proof of existence of the second state data after the second state data is inserted into the to-be-processed tree; and

generating the state write set based on the proof of modifiability and the second proof of existence.

13. The method according to claim 12, wherein the to-be-processed tree comprises a root node and a leaf node; and

the acquiring the key string associated with the second state data comprises:

acquiring the key string associated with the second state data before the second state data is inserted into the to-be-processed tree;

acquiring a second node index relationship indicated by the to-be-processed tree, and traversing from the root node to the leaf node to search for characters that match characters of the key string based on the second node index relationship; and

obtaining a to-be-processed string based on the matched characters, acquiring a proof of existence associated with the to-be-processed string, and setting the acquired proof of existence associated with the to-be-processed string as the proof of modifiability of the second state data.

14. A computer device, comprising a processor and a memory;

the processor being connected to the memory, the memory being configured to store a computer program, and the processor being configured to invoke the computer program to cause the computer device to perform the method according to any one of claims 1 to 7, or perform the method according to any one of claims 8 to 13.

15. A computer-readable storage medium, storing a computer program, the computer program, when being loaded and executed by a processor, causing a computer device with the processor to perform the method according to any one of claims 1 to 7, or perform the method according to any one of claims 8 to 13.