

CLAIMS

1. A method for controlling a data access right, wherein the method comprises:

receiving, by a first proxy node, a first request message from a request node, wherein the first request message comprises an identity of the request node and an identifier of to-be-accessed data;

determining, by the first proxy node, a first encrypted ciphertext on a blockchain-based on the identifier of the to-be-accessed data, and determining, based on the identity of the request node, whether the request node has a right to read the first encrypted ciphertext; and

if the request node has the right to read the first encrypted ciphertext-, initiating a right verification request for the request node to at least one second proxy node, and determining, based on a feedback result of the at least one second proxy node, provisioning of the first encrypted ciphertext,

wherein the determining, based on a feedback result of the at least one second proxy node, provisioning of the first encrypted ciphertext comprises:

receiving, by the first proxy node, partial keys sent by at least k second proxy nodes, wherein the partial keys are generated and sent by the second proxy nodes after the second proxy nodes verify, based on the identity of the request node, that the request node has the right to read the first encrypted ciphertext, and $k \leq m$;

recovering the at least k partial keys to generate a conversion key;

converting, by using the conversion key and an encryption algorithm, the first encrypted ciphertext into a second encrypted ciphertext encrypted by a public key of the request node;
and

sending the second encrypted ciphertext to the request node.

2. The method according to claim 1, wherein the determining, based on the identity of the request node, whether the request node has a right to read the first encrypted ciphertext comprises:

reading, on the blockchain, authorization information of the request node based on the identity of the request node, wherein the authorization information is closest to an end of a chained storage structure of the blockchain, and the authorization information comprises allowing or disallowing.

3. The method according to claim 1 or 2, wherein the initiating a right verification request

for the request node to at least one second proxy node comprises:

determining, by the first proxy node, m second proxy nodes, wherein m is a positive integer greater than or equal to 1;

generating the right verification request, wherein the right verification request comprises the identity of the request node and the first encrypted ciphertext, and the first encrypted ciphertext is generated by encrypting the to-be-accessed data; and

sending the right verification request to each of the m second proxy nodes.

~~4. The method according to any one of claims 1 to 3, wherein the determining, based on a feedback result of the at least one second proxy node, provisioning of the first encrypted ciphertext comprises:~~

~~receiving, by the first proxy node, partial keys sent by at least k second proxy nodes, wherein the partial keys are generated and sent by the second proxy nodes after the second proxy nodes verify, based on the identity of the request node, that the request node has the right to read the first encrypted ciphertext, and $k \leq m$;~~

~~recovering the at least k partial keys to generate a conversion key;~~

~~converting, by using the conversion key and an encryption algorithm, the first encrypted ciphertext into a second encrypted ciphertext encrypted by a public key of the request node; and~~

~~sending the second encrypted ciphertext to the request node.~~

~~54. The method according to claim 3, wherein the determining, based on a feedback result of the at least one second proxy node, provisioning of the first encrypted ciphertext comprises:~~

~~if the first proxy node receives partial keys sent by less than k second proxy nodes, or receives a denial message fed back by at least one second proxy node, sending an access denial message to the request node.~~

~~65. The method according to any one of claims 1 to 54, wherein the method further comprises:~~

~~determining, by the first proxy node based on the identity of the request node, that the request node has no right to read the first encrypted ciphertext, denying the first request message of the request node.~~

6. The method according to claim 3, further comprising:

receiving, by a second proxy node, the right verification request from the first proxy node, wherein the second proxy node is one of the m second proxy nodes;

determining, by the second proxy node based on the identity of the request node, whether the request node has a right to read the first encrypted ciphertext; and

if the request node has the right to read the first encrypted ciphertext, generating a partial key, and sending the generated partial key to the first proxy node.

7. The method according to claim 6, wherein the generating a partial key comprises:

obtaining, by the second proxy node, an encrypted partial key on a blockchain, wherein the encrypted partial key is generated by encryption using a public key of the second proxy node; and

decrypting, by the second proxy node, the encrypted partial key by using a private key of the second proxy node, to generate the partial key.

78. A data transmission method, wherein the method comprises:

obtaining, by a first node, an identity of a second node, wherein the identity is used to determine authorization information of the second node;

if the authorization information is allowing, generating, by the first node, a conversion key by using a private key of the first node, a public key of the second node, and an encryption algorithm;

encrypting, by the first node, the conversion key by using a public key of a proxy node on a blockchain network, to generate an encrypted partial key; and

generating, by the first node, first transaction information, and recording the first transaction information on the blockchain, wherein the first transaction information comprises the identity of the second node, the authorization information of the second node, and the encrypted partial key.

89. The method according to claim 78, wherein the encrypting, by the first node, the conversion key by using a public key of a proxy node on a blockchain network, to generate an encrypted partial key comprises:

splitting, by the first node, the conversion key into n partial keys by using a secret sharing algorithm, wherein n is a positive integer greater than or equal to 1; and

selecting, by the first node, m proxy nodes from the blockchain network, and encrypting the n partial keys by using public keys of the m proxy nodes, to generate n encrypted partial

keys, wherein $m \leq n$.

~~9~~10. The method according to claim ~~7~~8 or ~~8~~9, wherein
the first transaction information further comprises: identities of the m proxy nodes, and a minimum quantity k , of partial keys for recovering the conversion key.

~~10~~11. The method according to claim ~~7~~8, wherein the method further comprises:
if the authorization information is disallowing, generating second transaction information, wherein the second transaction information comprises the identity of the second node, and the authorization information that is disallowing and that is of the second node; and recording the second transaction information on the blockchain.

~~11~~12. The method according to any one of claims ~~7~~8 to ~~9~~10, wherein before the obtaining, by a first node, an identity of a second node, the method further comprises:

encrypting, by the first node, to-be-accessed data by using a public key of the first node and a proxy re-encryption algorithm, to generate a first encrypted ciphertext; and
storing, by the first node, the first encrypted ciphertext as a transaction on the blockchain.

~~12. A data transmission method, wherein the method comprises:
receiving, by a second proxy node, a right verification request from a first proxy node, wherein the right verification request comprises an identity of a request node and a first encrypted ciphertext, and the first encrypted ciphertext is generated by encrypting to-be-accessed data;~~

~~determining, by the second proxy node based on the identity of the request node, whether the request node has a right to read the first encrypted ciphertext; and~~

~~if the request node has the right to read the first encrypted ciphertext, generating a partial key, and sending the partial key to the first proxy node.~~

~~13. The method according to claim 12, wherein the generating a partial key comprises:
obtaining, by the second proxy node, an encrypted partial key on a blockchain, wherein the encrypted partial key is generated by encryption using a public key of the second proxy node; and~~

~~decrypting, by the second proxy node, the encrypted partial key by using a private key of the second proxy node, to generate the partial key.~~

~~14~~13. A network node, applied to a first proxy node and comprising:

a receiving unit, configured to receive a first request message from a request node, wherein the first request message comprises an identity of the request node and an identifier of to-be-accessed data;

a processing unit, configured to: determine a first encrypted ciphertext on a blockchain based on the identifier of the to-be-accessed data, and determine, based on the identity of the request node, whether the request node has a right to read the first encrypted ciphertext; and

a sending unit, configured to initiate a right verification request for the request node to at least one second proxy node if the request node has the right to read the first encrypted ciphertext, wherein

the processing unit is further configured to: determine, based on a feedback result of the at least one second proxy node, provisioning of the first encrypted ciphertext if the request node has the right to read the first encrypted ciphertext,

wherein the receiving unit is further configured to receive partial keys sent by at least k second proxy nodes, wherein the partial keys are generated and sent by the second proxy nodes after the second proxy nodes verify, based on the identity of the request node, that the request node has the right to read the first encrypted ciphertext, and $k \leq m$;

the processing unit is configured to: recover the at least k partial keys to generate a conversion key; and convert, by using the conversion key and an encryption algorithm, the first encrypted ciphertext into a second encrypted ciphertext encrypted by a public key of the request node; and

the sending unit is further configured to send the second encrypted ciphertext to the request node.

~~1514~~. The network node according to claim ~~1413~~, wherein

the sending unit is configured to read, on a blockchain, authorization information of the request node based on the identity of the request node, wherein the authorization information is closest to an end of a chained storage structure on the blockchain, and the authorization information comprises allowing or disallowing.

~~1615~~. The network node according to claim ~~14-13~~ or ~~1514~~, wherein

the processing unit is configured to: determine m second proxy nodes, and generate the right verification request, wherein the right verification request comprises the identity of the request node and the first encrypted ciphertext, the first encrypted ciphertext is generated by encrypting the to-be-accessed data, and m is a positive integer greater than or equal to 1; and

the sending unit is configured to send the right verification request to each of the m second proxy nodes.

~~17. The network node according to any one of claims 14 to 16, wherein~~

~~the receiving unit is further configured to receive partial keys sent by at least k second proxy nodes, wherein the partial keys are generated and sent by the second proxy nodes after the second proxy nodes verify, based on the identity of the request node, that the request node has the right to read the first encrypted ciphertext, and $k \leq m$;~~

~~the processing unit is configured to: recover the at least k partial keys to generate a conversion key; and convert, by using the conversion key and an encryption algorithm, the first encrypted ciphertext into a second encrypted ciphertext encrypted by a public key of the request node; and~~

~~the sending unit is further configured to send the second encrypted ciphertext to the request node.~~

~~1816. The network node according to claim 1615, wherein~~

~~the processing unit is configured to: if receiving, by using the receiving unit, partial keys sent by less than k second proxy nodes, or receiving, by using the receiving unit, a denial message fed back by at least one second proxy node, send an access denial message to the request node by using the sending unit.~~

~~1917. The network node according to any one of claims 1413 to 1816, wherein~~

~~the processing unit is further configured to: determine, based on the identity of the request node, whether the request node has the right to read the first encrypted ciphertext; and if the request node has no right to read the first encrypted ciphertext, deny the first request message of the request node.~~

18. The network node according to claim 15, wherein

the receiving unit is configured to receive the right verification request from the first proxy node;

the processing unit is configured to: determine, based on the identity of the request node, whether the request node has a right to read the first encrypted ciphertext; and if the request node has the right to read the first encrypted ciphertext, generate a partial key; and

the sending unit is configured to send the partial key to the first proxy node.

19. The network node according to claim 18, wherein the processing unit is configured to obtain an encrypted partial key on a blockchain, wherein the encrypted partial key is generated after encryption is performed by using a public key of the second proxy node; and decrypt the encrypted partial key by using a private key of the second proxy node, to generate the partial key.

20. A network node, applied to a first node and comprising:
a receiving unit, configured to obtain an identity of a second node, wherein the identity is used to determine authorization information of the second node; and
a processing unit, configured to: if the authorization information is allowing, generate a conversion key by using a private key of the network node, a public key of the second node, and an encryption algorithm; encrypt the conversion key by using a public key of a proxy node on a blockchain network, to generate an encrypted partial key; and generate first transaction information, and record the first transaction information on the blockchain, wherein the first transaction information comprises the identity of the second node, the authorization information of the second node, and the encrypted partial key.

21. The network node according to claim 20, wherein the processing unit is configured to: split the conversion key into n partial keys by using a secret sharing algorithm, wherein n is a positive integer greater than or equal to 1; and select m proxy nodes from the blockchain network, and encrypt the n partial keys by using public keys of the m proxy nodes, to generate n encrypted partial keys, wherein $m \leq n$.

22. The network node according to claim 20 or 21, wherein the first transaction information further comprises: identities of the m proxy nodes, and a minimum quantity k , of partial keys for recovering the conversion key.

23. The network node according to claim 20, wherein the processing unit is further configured to: if the authorization information is disallowing, generate second transaction information, and record the second transaction information on the blockchain, wherein the second transaction information comprises the identity of the second node, and the authorization information that is disallowing and that is of the second node.

24. The network node according to any one of claims 20 to 22, wherein the processing unit is further configured to: before obtaining the identity of the second node, encrypt to-be-accessed data by using a public key of the first node and a proxy re-encryption algorithm, to generate a first encrypted ciphertext; and store the first encrypted ciphertext as a transaction on the blockchain.

~~25. A network node, applied to a second proxy node and comprising:
a receiving unit, configured to receive a right verification request from a first proxy node, wherein the right verification request comprises an identity of a request node and a first encrypted ciphertext, and the first encrypted ciphertext is generated by encrypting to-be-accessed data;
a processing unit, configured to: determine, based on the identity of the request node, whether the request node has a right to read the first encrypted ciphertext; and if the request node has the right to read the first encrypted ciphertext, generate a partial key; and
a sending unit, configured to send the partial key to the first proxy node.~~

~~26. The network node according to claim 25, wherein
the processing unit is configured to obtain an encrypted partial key on a blockchain, wherein the encrypted partial key is generated after encryption is performed by using a public key of the second proxy node; and decrypt the encrypted partial key by using a private key of the second proxy node, to generate the partial key.~~

~~27~~25. A network node, comprising: a transceiver, a memory and a processor configured to execute code stored in the memory to perform one of the methods according to claims 1 to ~~13~~12.

~~28~~26. A computer readable storage medium, storing an instruction, when the instruction is run on a computer, the computer is caused to perform one of the methods according to claims 1 to ~~13~~12.

~~29~~27. A computer program product, comprising an instruction, when the instruction is run on a computer, the computer is caused to perform one of the methods according to claims 1 to ~~13~~12.