

# INVOICE ACCESS METHOD AND APPARATUS BASED ON BLOCKCHAIN, AND ELECTRONIC DEVICE

## TECHNICAL FIELD

[0001] One or more implementations of the present specification relate to the field  
5 of blockchain technologies, and in particular, to an invoice access method and  
apparatus based on a blockchain, and an electronic device.

## BACKGROUND

[0002] A blockchain technology, also known as a distributed ledger technology, is  
an emerging technology in which several computing devices participate in  
10 "bookkeeping" and jointly maintain a complete distributed database. Because the  
blockchain technology is characterized by being decentralized, open and transparent,  
enabling each computing device to participate in database recording, and enabling  
data synchronization to be performed quickly between the computing devices, the  
blockchain technology is used to build a decentralized system, and various executable  
15 programs are stored in the distributed database of the blockchain for automatic  
execution, which has been widely used in many fields.

## SUMMARY

[0003] The present specification provides an invoice access method based on a  
blockchain, including: receiving an access request initiated by an access user for a  
20 target invoice by using a client, the target invoice being encrypted and stored in the  
blockchain; determining, in response to the access request, whether the access user  
has an authority to access the target invoice; and in response to that the access user  
has the authority to access the target invoice, invoking decryption logic provided in a  
predetermined smart contract to decrypt ciphertext data of the target invoice stored in  
25 the blockchain, and returning decrypted plaintext data of the target invoice to the  
client.

[0004] Optionally, the smart contract further provides an access authority  
verification logic for the access user; and the determining whether the access user has

the authority to access the target invoice includes: invoking the access authority verification logic provided in the smart contract to determine whether the access user has the authority to access the target invoice.

**[0005]** Optionally, the method further includes: receiving a creation request initiated by a creation user for the target invoice by using a client; determining an invoice issuer and an invoice receiver of the target invoice in response to the creation request; and invoking an encryption logic provided in the smart contract, to separately encrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, and adding encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage.

**[0006]** Optionally, the separately encrypting the plaintext data of the target invoice based on the keys corresponding to the invoice issuer and the invoice receiver includes: encrypting plaintext data of the target invoice based on a first key corresponding to the invoice issuer to obtain first ciphertext data; and encrypting the plaintext data of the target invoice based on a second key corresponding to the invoice receiver to obtain second ciphertext data; and the adding the encrypted ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage includes: adding a corresponding relationship between the first ciphertext data and a user identifier of the invoice issuer to the blockchain for storage; and adding a corresponding relationship between the second ciphertext data and a user identifier of the invoice receiver to the blockchain for storage.

**[0007]** Optionally, the access request includes a user identifier of the access user; and the determining whether the access user has the authority to access the target invoice includes: obtaining the user identifiers of the invoice issuer and the invoice receiver of the target invoice; determining whether the user identifier of the access user matches the user identifiers of the invoice issuer or the invoice receiver; and in response to that the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver, determining that the access user has the authority to access the target invoice.

**[0008]** Optionally, the decrypting the ciphertext data of the target invoice stored in the blockchain includes: decrypting, based on a key corresponding to the access user, the ciphertext data of the target invoice stored in the blockchain.

[0009] Optionally, the smart contract further provides a key distribution logic; and the method further includes: receiving a registration request initiated by a user by using a client; and invoking the key distribution logic in the smart contract in response to the registration request, and writing a corresponding relationship between a user  
5 identifier of the user and a key distributed to the user into the smart contract.

[0010] Optionally, the invoice identifier is a taxpayer identification number.

[0011] The present specification further provides an invoice access apparatus based on a blockchain, including: a first receiving module, configured to receive an access request initiated by an access user for a target invoice by using a client, the target  
10 invoice being encrypted and stored in the blockchain; a first determining module, configured to determine, in response to the access request, whether the access user has an authority to access the target invoice; and a decryption module, configured to: when the access user has the authority to access the target invoice, invoke decryption logic provided in a predetermined smart contract to decrypt ciphertext data of the  
15 target invoice stored in the blockchain, and return decrypted plaintext data of the target invoice to the client.

[0012] Optionally, the smart contract further provides an access authority verification logic for the access user; and the first determining module is specifically configured to: invoke the access authority verification logic provided in the smart  
20 contract to determine whether the access user has the authority to access the target invoice.

[0013] Optionally, the apparatus further includes: a second receiving module, configured to receive a creation request initiated by a creation user for the target invoice by using a client; a second determining module, configured to determine an  
25 invoice issuer and an invoice receiver of the target invoice in response to the creation request; and an encryption module, configured to invoke an encryption logic provided in the smart contract, to separately encrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, and add encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer  
30 and the invoice receiver of the target invoice to the blockchain for storage.

[0014] Optionally, the encryption module is specifically configured to: encrypt plaintext data of the target invoice based on a first key corresponding to the invoice issuer to obtain first ciphertext data; and encrypt the plaintext data of the target invoice based on a second key corresponding to the invoice receiver to obtain second

ciphertext data; and the adding the encrypted ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage includes: adding a corresponding relationship between the first ciphertext data and a user identifier of the invoice issuer to the blockchain for storage; and adding a corresponding relationship between the second ciphertext data and a user identifier of the invoice receiver to the blockchain for storage.

**[0015]** Optionally, the access request includes a user identifier of the access user; and the first determining module is specifically configured to: obtain the user identifiers of the invoice issuer and the invoice receiver of the target invoice; determine whether the user identifier of the access user matches the user identifiers of the invoice issuer or the invoice receiver; and in response to that the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver, determine that the access user has the authority to access the target invoice.

**[0016]** Optionally, the decryption module is specifically configured to: decrypt, based on a key corresponding to the access user, the ciphertext data of the target invoice stored in the blockchain.

**[0017]** Optionally, the smart contract further provides a key distribution logic; and the apparatus further includes: a third receiving module, configured to receive a registration request initiated by a user by using a client; and a registration module, configured to invoke the key distribution logic in the smart contract in response to the registration request, and write a corresponding relationship between a user identifier of the user and a key distributed to the user into the smart contract.

**[0018]** Optionally, the invoice identifier is a taxpayer identification number.

**[0019]** The present specification further provides an electronic device, including: a processor; and a memory configured to store machine-executable instructions; where by reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a blockchain, the processor is enabled to: receive an access request initiated by an access user for a target invoice by using a client, the target invoice being encrypted and stored in the blockchain; determine, in response to the access request, whether the access user has an authority to access the target invoice; and in response to that the access user has the authority to access the target invoice, invoke decryption logic provided in a

predetermined smart contract to decrypt ciphertext data of the target invoice stored in the blockchain, and return decrypted plaintext data of the target invoice to the client.

[0020] In the above technical solutions, when a user needs to access an invoice encrypted and stored in a blockchain, access authority of the user can be verified by  
5 invoking corresponding logic in a smart contract deployed in the blockchain, and the user is allowed to access the invoice when it is determined that the user has the authority to access the invoice. As such, data security of invoices circulating in the blockchain can be ensured, and potential safety hazards caused by invoice data leakage are avoided.

## 10 BRIEF DESCRIPTION OF DRAWINGS

[0021] FIG. 1 is a flowchart illustrating an invoice access method based on a blockchain according to some example implementations of the present specification;

[0022] FIG. 2 is a flowchart illustrating another invoice access method based on a blockchain according to some example implementations of the present specification;

15 [0023] FIG. 3 is a diagram illustrating a hardware structure of an electronic device where an invoice access apparatus based on a blockchain is located according to some example implementations of the present specification; and

[0024] FIG. 4 is a block diagram illustrating an invoice access apparatus based on a blockchain according to some example implementations of the present specification.

## 20 DESCRIPTION OF IMPLEMENTATIONS

[0025] Example implementations are described in detail herein, and examples thereof are shown in the accompanying drawings. When the following description refers to the accompanying drawings, unless otherwise indicated, the same numbers in different accompanying drawings refer to the same or similar elements. The following  
25 example implementations do not represent all implementations consistent with one or more implementations of the present specification. In contrast, they are merely examples of apparatuses and methods consistent with some aspects of one or more implementations of the present specification as described in detail in the appended claims.

**[0026]** Terms used in the present specification are for the purpose of describing example implementations only and are not intended to limit the present specification. As used in the present specification and the appended claims, the singular forms "a (an)", "the" and "this" are also intended to include the plural forms, unless the context  
5 clearly indicates other meaning. It should also be understood that the term "and/or" as used herein refers to and includes any or all possible combinations of one or more associated listed items.

**[0027]** It should be understood that although the terms such as "first", "second", and "third" can be used to describe various types of information in the present  
10 specification, the information should not be limited to these terms. These terms are only used to differentiate information of a same type. For example, without departing from the scope of the present specification, the first information can also be called second information, and similarly, the second information can also be called first information. Depending on the context, the word "if" as used herein can be explained  
15 as "when" or "in response to determining".

**[0028]** The present specification provides technical solutions in which when a user needs to access an invoice encrypted and stored in a blockchain, access authority of the user is verified by invoking corresponding logic in a smart contract deployed in a blockchain network of the blockchain, and the user is allowed to access the invoice  
20 when it is determined that the user has the authority to access the invoice.

**[0029]** In example implementations, invoices are usually encrypted and stored in a blockchain, that is, invoices are usually stored in the blockchain in the form of ciphertext. When a user needs to access a certain invoice stored in the blockchain, the user can initiate a request for access to the invoice by using a client.

**[0030]** After receiving the access request, a node device in the blockchain network can respond to the access request.  
25

**[0031]** Specifically, it can be determined first whether the user has authority to access the invoice. Usually only an invoice issuer and an invoice receiver of a certain invoice are allowed to access the invoice, and other users are not allowed to access the  
30 invoice, so as to ensure the data security of the invoice and avoid leakage of invoice data. In this case, it can be first determined whether the user is the invoice issuer or invoice receiver of the invoice. In response to that the user is the invoice issuer or invoice receiver of the invoice, it can be determined that the user has the authority to access the invoice.

**[0032]** After it is determined that the user has the authority to access the invoice, the node device can invoke decryption logic provided, e.g., declared, in a smart contract deployed in the blockchain to decrypt ciphertext data of the invoice stored in the blockchain, to obtain plaintext data of the invoice.

5 **[0033]** After obtaining the plaintext data of the invoice, the node device can return the plaintext data of the invoice to the client, so that the client can perform subsequent service processing based on the plaintext data of the invoice. For example, the client can show the plaintext data of the invoice to the access user for viewing.

**[0034]** In the above technical solutions, when a user needs to access an invoice  
10 encrypted and stored in a blockchain, access authority of the user can be verified by invoking corresponding logic in a smart contract deployed in the blockchain, and the user is allowed to access the invoice when it is determined that the user has the authority to access the invoice. As such, data security of invoices circulating in the blockchain can be ensured, and potential safety hazards caused by invoice data  
15 leakage are avoided.

**[0035]** The present specification is further described below with reference to example implementations.

**[0036]** FIG. 1 is a flowchart illustrating an invoice access method based on a blockchain according to some example implementations of the present specification.

20 The method can be applied at a node device in a blockchain network, and includes the following steps.

**[0037]** Step 102: Receive an access request initiated by an access user for a target invoice by using a client, the target invoice being encrypted and stored in the blockchain.

25 **[0038]** Step 104: Determine, in response to the access request, whether the access user has an authority to access the target invoice.

**[0039]** Step 106: In response to that the access user has the authority to access the target invoice, invoke decryption logic provided in a predetermined smart contract to decrypt ciphertext data of the target invoice stored in the blockchain, and return  
30 decrypted plaintext data of the target invoice to the client.

**[0040]** The blockchain network described in the present specification can include any type of blockchain network. For example, in practice, any one of a public blockchain network, a private blockchain network, and a consortium blockchain network can be used.

[0041] In the implementations, invoices can be encrypted and stored in a blockchain, that is, invoices are usually stored in the blockchain in the form of ciphertext.

[0042] When an access user needs to access a target invoice stored in the blockchain, the access user can initiate, by using a client, a transaction for accessing the invoice, *i.e.*, initiate a request for access to the invoice by using the client.

[0043] Transactions in the blockchain are divided into transactions in narrow meanings and transactions in broad meanings. A transaction in narrow meanings refers to a value transfer added by a user to the blockchain. For example, in a conventional bitcoin blockchain network, a transaction can be a transfer initiated by a user in the blockchain. A transaction in broad meanings refers to a piece of service or business data with a service or business intention added by a user to the blockchain. For example, the operator can build a consortium blockchain based on actual service requirements, and deploy, based on the consortium blockchain, some other online services unrelated to value transfer (such as a rental service, a vehicle dispatching service, an insurance claims service, a credit service, and a medical service). In this consortium blockchain, the transaction can be a service/business message or service/business request with a service/business intention added by the user in the consortium blockchain.

[0044] It should be noted that the user can be an individual or an enterprise, which does not limit the scope of the present specification.

[0045] After receiving the access request, a node device in the blockchain can respond to the access request.

[0046] For example, it can be determined first whether the access user has an authority to access the target invoice.

[0047] After it is determined that the access user has the authority to access the target invoice, decryption logic provided in a smart contract deployed in the blockchain can be invoked to decrypt ciphertext data of the target invoice stored in the blockchain, so as to obtain plaintext data of the target invoice.

[0048] The decryption logic can be, for example, program code (such as some program methods or functions that can be invoked) that is provided in the smart contract and related to execution logic for decrypting ciphertext data of an invoice.

[0049] After the plaintext data of the target invoice is decrypted, the plaintext data of the target invoice can be returned to the client, so that the client can perform

subsequent service processing based on the plaintext data of the target invoice. For example, the client can sort out the plaintext data of the target invoice based on the standard format of an invoice, and show the sorted target invoice to the access user for viewing.

5   **[0050]**   In practice, the smart contract can further provide an access authority verification logic for the access user. In this case, the access authority verification logic in the smart contract can be invoked to determine whether the access user has an authority to access the target invoice.

10   **[0051]**   The access authority verification logic can be, for example, program code that is provided in the smart contract and related to execution logic for verifying access authority of an access user.

**[0052]**   In addition, referring to FIG. 2, the following steps can be performed to encrypt a target invoice and then store the encrypted target invoice in a blockchain.

15   **[0053]**   Step 202: Receive a creation request initiated by a creation user for the target invoice by using a client.

**[0054]**   Step 204: Determine an invoice issuer and an invoice receiver of the target invoice in response to the creation request.

20   **[0055]**   Step 206: Invoke an encryption logic provided in the smart contract, to separately encrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, and add encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage.

**[0056]**   In the implementations, the creation user can initiate the creation request for the above target invoice by using a client.

25   **[0057]**   After receiving the creation request, the node device in the blockchain can first determine an invoice issuer and an invoice receiver of the target invoice in response to the creation request. In practice, the invoice issuer of the target invoice is usually the creation user.

30   **[0058]**   After the invoice issuer and the invoice receiver of the target invoice are determined, encryption logic provided in a smart contract deployed in the blockchain can be invoked to decrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, so as to obtain ciphertext data of the target invoice.

[0059] The encryption logic can be, for example, program code that is provided in the smart contract and related to execution logic for encrypting plaintext data of an invoice.

5 [0060] After ciphertext data of the target invoice is encrypted, the ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice can be added to the blockchain for storage. The user identifier can be, for example, a taxpayer identification number.

10 [0061] For example, the ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice can be sent to a node device in the blockchain. After receiving the ciphertext data and the user identifiers, the node device in the blockchain can initiate consensus on the ciphertext data and the user identifiers, and store the ciphertext data and the user identifiers after the consensus is passed, so that the ciphertext data and the user identifiers are added to the blockchain for storage.

15 [0062] It should be noted that the smart contract for decrypting ciphertext data of an invoice stored in the blockchain and the smart contract for encrypting the plaintext data of the invoice and adding the encrypted plaintext data to the blockchain for storage can be integrated into one smart contract and deployed in the blockchain network, or can be deployed in the blockchain network as two different smart contracts, which does not limit the scope of the present specification.

[0063] In practice, the above smart contract can further provide user identity verification logic for the target invoice. In this case, the user identity verification logic in the smart contract can be invoked to determine the invoice issuer and the invoice receiver of the target invoice.

25 [0064] The user identity verification logic can be, for example, code that is provided in the smart contract and related to execution logic for determining the invoice issuer and the invoice receiver of the invoice.

30 [0065] For example, the creation request can include the user identifiers of the invoice issuer and the invoice receiver that is input by the creation user during creation of the target invoice, that is, the user identifier of the invoice issuer provided by the creation user can be determined as the user identifier of the invoice issuer of the target invoice, and the user identifier of the invoice receiver provided by the creation user can be determined as the user identifier of the invoice receiver of the

target invoice, so that the invoice issuer and the invoice receiver of the target invoice are determined.

[0066] In some illustration implementations, to encrypt the plaintext data of the target invoice, on the one hand, the plaintext data of the target invoice can be encrypted based on the key corresponding to the invoice issuer, to obtain ciphertext data (called first ciphertext data) of the target invoice corresponding to the invoice issuer; on the other hand, the plaintext data of the target invoice can be encrypted based on the key corresponding to the invoice receiver, to obtain ciphertext data (called second ciphertext data) of the target invoice corresponding to the invoice receiver.

[0067] Subsequently, a corresponding relationship between the first ciphertext data and the user identifier of the invoice issuer can be added to the blockchain for storage, and a corresponding relationship between the second ciphertext data and the user identifier of the invoice receiver can be added to the blockchain for storage.

[0068] For example, the storage form of the ciphertext data of the target invoice can be shown in Table 1:

Table 1

|                   |                   |                   |                   |
|-------------------|-------------------|-------------------|-------------------|
| Ciphertext data 1 | User identifier 1 | Ciphertext data 2 | User identifier 2 |
|-------------------|-------------------|-------------------|-------------------|

[0069] In Table 1, ciphertext data 1 is the ciphertext data of the target invoice corresponding to the invoice issuer, *e.g.*, the first ciphertext data, and user identifier 1 is the user identifier of the invoice issuer; and ciphertext data 2 is the ciphertext data of the target invoice corresponding to the invoice receiver, *e.g.*, the second ciphertext data, and user identifier 2 is the user identifier of the invoice receiver.

[0070] With reference to the implementation shown in FIG. 2 and continuing to refer to FIG. 1, in some illustrated implementations, the access request initiated by the access user for the target invoice by using the client can include the user identifier of the access user.

[0071] In this case, to determine whether the access user has the authority to access the target invoice, the user identifiers of the invoice issuer and the invoice receiver of the target invoice can be obtained first.

[0072] After the user identifiers of the invoice issuer and the invoice receiver are obtained, on the one hand, the user identifier of the access user can be compared with the user identifier of the invoice issuer; and on the other hand, the user identifier of the access user can be compared with the user identifier of the invoice receiver, so as

to determine whether the user identifier of the access user matches the user identifiers of the invoice issuer or the invoice receiver.

[0073] In response to that the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver, it can be  
5 determined that the access user has the authority to access the target invoice. For example, in response to that the user identifier of the access user is the same as the user identifier of the invoice issuer, it means that the access user is the invoice issuer of the target invoice and has the authority to access the target invoice; and in response to that the user identifier of the access user is the same as the user identifier of the  
10 invoice receiver, it means that the access user is the invoice receiver of the target invoice and also has the authority to access the target invoice.

[0074] After it is determined that the access user has the authority to access the target invoice, the ciphertext data of the target invoice stored in the blockchain can be decrypted based on the key corresponding to the access user, so as to obtain plaintext  
15 data of the target invoice.

[0075] In practice, the smart contract can further provide key distribution logic for a user. In this case, any user can initiate a registration request by using a client.

[0076] The key distribution logic can be, for example, program code that is provided in the smart contract and related to execution logic for distributing keys to  
20 users.

[0077] After receiving the registration request, a node device in the blockchain can invoke, in response to the registration request, the key distribution logic in the smart contract to distribute a key to the user, and write a corresponding relationship between a user identifier (which can usually be included in the registration request) of the user  
25 and the key distributed to the user into the smart contract.

[0078] That is, the corresponding relationship shown in Table 2 can be stored in the smart contract:

Table 2

|                   |       |
|-------------------|-------|
| User identifier 1 | Key 1 |
| User identifier 2 | Key 2 |
| User identifier 3 | Key 3 |
| ...               | ...   |

[0079] In Table 2, assuming that user identifier 1 is the user identifier of user 1, the key corresponding to user 1 is key 1; assuming that user identifier 2 is the user identifier of user 2, the key corresponding to user 2 is key 2; and so on.

[0080] The present specification is described below by taking an invoice issued by enterprise A to enterprise B as an example.

[0081] First, enterprise A and enterprise B can separately initiate registration requests by using their clients.

[0082] After receiving the registration requests, a node device in a blockchain can invoke, in response to the registration requests, key distribution logic in a smart contract deployed in the blockchain to distribute keys to enterprise A and enterprise B respectively.

[0083] Assuming that the user identifier of enterprise A is tax identification number A, and the key distributed to enterprise A is key A; the user identifier of enterprise B is tax identification number B, and the key distributed to enterprise B is key B, then the corresponding relationship between user identifiers and keys shown in Table 3 can be stored in the smart contract deployed in the blockchain:

Table 3

|                             |       |
|-----------------------------|-------|
| Tax identification number A | Key A |
| Tax identification number B | Key B |
| ...                         | ...   |

[0084] When enterprise A issues an invoice to enterprise B, enterprise A can initiate a creation request for the invoice by using a client.

[0085] After receiving the creation request, a node device in the blockchain can determine, in response to the creation request, the invoice issuer of the invoice as enterprise A (whose user identifier is tax identification number A), and the invoice receiver of the invoice as enterprise B (whose user identifier is tax identification number B).

[0086] After the invoice issuer and the invoice receiver of the invoice are determined, on the one hand, key A corresponding to enterprise A can be determined based on tax identification number A of enterprise A (invoice issuer), and plaintext data of the invoice can be encrypted based on key A to obtain ciphertext data A of the invoice corresponding to enterprise A; on the other hand, key B corresponding to enterprise B can be determined based on tax identification number B of enterprise B

(invoice receiver), and the plaintext data of the invoice can be encrypted based on key B to obtain ciphertext data B of the invoice corresponding to enterprise B.

[0087] Subsequently, the ciphertext data (ciphertext data A and ciphertext data B) of the invoice and the user identifiers (tax identification number A and tax identification number B) of the invoice issuer and invoice receiver of the invoice can be added to the blockchain for storage in the storage form shown in Table 4:

Table 4

|                   |                             |                   |                             |
|-------------------|-----------------------------|-------------------|-----------------------------|
| Ciphertext data A | Tax identification number A | Ciphertext data B | Tax identification number B |
|-------------------|-----------------------------|-------------------|-----------------------------|

[0088] When enterprise B needs to access the invoice, enterprise B can initiate an access request for the invoice by using a client. The access request can include tax identification number B.

[0089] After receiving the access request, the node device in the blockchain can first determine, in response to the access request, user identifiers of the invoice issuer and the invoice receiver of the invoice as tax identification number A and tax identification number B respectively. Subsequently, it can be determined whether the user identifier (tax identification number B) of enterprise B matches the user identifiers (tax identification number A and tax identification number B) of the invoice issuer and invoice receiver of the invoice.

[0090] Because the user identifier of enterprise B is the user identifier of the invoice receiver of the invoice, that is, enterprise B is the invoice receiver of the invoice, it can be determined that the user identifier of enterprise B matches the user identifier of the invoice receiver of the invoice, and thus it can be determined that enterprise B has the authority to access the target invoice.

[0091] Subsequently, key B corresponding to enterprise B can be determined based on tax identification number B of enterprise B, the ciphertext data of the invoice can be decrypted based on key B to obtain the plaintext data of the invoice, and the plaintext data of the invoice can be returned to the client used by enterprise B.

[0092] In the above technical solutions, when a user needs to access an invoice encrypted and stored in a blockchain, access authority of the user can be verified by invoking corresponding logic in a smart contract deployed in the blockchain, and the user is allowed to access the invoice when it is determined that the user has the authority to access the invoice. As such, data security of invoices circulating in the

blockchain can be ensured, and potential safety hazards caused by invoice data leakage are avoided.

**[0093]** Corresponding to the above implementations of the invoice access method based on a blockchain, the present specification further provides implementations of the invoice access apparatus based on a blockchain.

**[0094]** The implementations of the invoice access apparatus based on a blockchain according to the present specification can be applied to an electronic device. The apparatus implementations can be implemented by using software, hardware, or a combination thereof. Taking software implementation as an example, a logical apparatus is implemented by reading, by using a processor of an apparatus device where the apparatus is located, corresponding computer program instructions in a non-volatile memory into a memory for running. On the hardware level, FIG. 3 is a diagram illustrating a hardware structure of an electronic device where an invoice access apparatus based on a blockchain is located according to the present specification. In addition to a processor, a memory, a network interface, and a non-volatile memory shown in FIG. 3, the electronic device where the apparatus is located according to the implementation can generally further include other hardware on the basis of the actual invoice access function based on the blockchain, and details are omitted for simplicity.

**[0095]** FIG. 4 is a block diagram illustrating an invoice access apparatus based on a blockchain according to some example implementations of the present specification. The apparatus 40 can be applied to the electronic device shown in FIG. 3, and includes: a first receiving module 401, configured to receive an access request initiated by an access user for a target invoice by using a client, the target invoice being encrypted and stored in the blockchain; a first determining module 402, configured to determine, in response to the access request, whether the access user has an authority to access the target invoice; and a decryption module 403, configured to: when the access user has an authority to access the target invoice, invoke decryption logic provided in a predetermined smart contract to decrypt ciphertext data of the target invoice stored in the blockchain, and return decrypted plaintext data of the target invoice to the client.

**[0096]** In the implementations, the smart contract can further provide an access authority verification logic for the access user; and the first determining module 402 can be specifically configured to: invoke the access authority verification logic

provided in the smart contract to determine whether the access user has an authority to access the target invoice.

[0097] In the implementations, the apparatus 40 can further include: a second receiving module 404, configured to receive a creation request initiated by a creation user for the target invoice by using a client; a second determining module 405, configured to determine an invoice issuer and an invoice receiver of the target invoice in response to the creation request; and an encryption module 406, configured to invoke an encryption logic provided in the smart contract, to separately encrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, and add encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage.

[0098] In the implementations, the encryption module 406 can be specifically configured to: encrypt plaintext data of the target invoice based on a first key corresponding to the invoice issuer to obtain first ciphertext data; and encrypt the plaintext data of the target invoice based on a second key corresponding to the invoice receiver to obtain second ciphertext data; and the adding the encrypted ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage includes: adding a corresponding relationship between the first ciphertext data and a user identifier of the invoice issuer to the blockchain for storage; and adding a corresponding relationship between the second ciphertext data and a user identifier of the invoice receiver to the blockchain for storage.

[0099] In the implementations, the access request can include a user identifier of the access user; and the first determining module 402 can be specifically configured to: obtain the user identifiers of the invoice issuer and the invoice receiver of the target invoice; determine whether the user identifier of the access user matches the user identifiers of the invoice issuer or the invoice receiver; and in response to that the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver, determine that the access user has the authority to access the target invoice.

[0100] In the implementations, the decryption module 403 can be specifically configured to: decrypt, based on a key corresponding to the access user, the ciphertext data of the target invoice stored in the blockchain.

[0101] In the implementations, the smart contract can further provide key distribution logic; and the apparatus 40 can further include: a third receiving module 407, configured to receive a registration request initiated by a user by using a client; and a registration module 408, configured to invoke the key distribution logic in the smart contract in response to the registration request, and write a corresponding relationship between a user identifier of the user and a key distributed to the user into the smart contract.

[0102] In the implementations, the invoice identifier can be a taxpayer identification number.

10 [0103] The implementation process of functions of each module in the above apparatus is detailed in the implementation process of the corresponding steps in the above method. Details are omitted herein for simplicity.

[0104] Because the apparatus implementations basically correspond to the method implementations, for related parts, references can be made to partial description of the method implementation. The apparatus implementations described above are merely illustrative, where modules described as separate components can or cannot be physically separated. Components displayed as modules can or cannot be physical modules, that is, the components can be located in one place, or can be distributed to multiple network modules. Some or all of the modules can be selected based on actual needs to achieve the object of the solution of the present specification. A person of ordinary skill in the art can understand and implement the implementations without innovative efforts.

20 [0105] The systems, apparatuses, or modules illustrated in the above implementations can be specifically implemented by computer chips or entities, or by products with certain functions. A typical implementation device is a computer. The computer can be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email sending and receiving device, a game console, a tablet computer, a wearable device, or a combination thereof.

30 [0106] Corresponding to the above implementations of the invoice access method based on a blockchain, the present specification further provides implementations of an electronic device. The electronic device includes a processor and a memory configured to store machine-executable instructions, where the processor and the memory are usually connected to each other by using an internal bus. In other possible

implementations, the device can further include an external interface that can be communicated with other devices or components.

**[0107]** In the implementations, by reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a blockchain, the processor is enabled to: receive an access request initiated by an access user for a target invoice by using a client, the target invoice being encrypted and stored in the blockchain; determine, in response to the access request, whether the access user has an authority to access the target invoice; and in response to that the access user has the authority to access the target invoice, invoke decryption logic declared in a predetermined smart contract to decrypt ciphertext data of the target invoice stored in the blockchain, and return decrypted plaintext data of the target invoice to the client.

**[0108]** In the implementations, the smart contract can further provide an access authority verification logic for the access user; and by reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a blockchain, the processor is enabled to: invoke the access authority verification logic declared in the smart contract to determine whether the access user has an authority to access the target invoice.

**[0109]** In the implementations, by reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a blockchain, the processor is further enabled to: receive a creation request initiated by a creation user for the target invoice by using a client; determine an invoice issuer and an invoice receiver of the target invoice in response to the creation request; and invoke an encryption logic declared in the smart contract, to separately encrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, and add encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage.

**[0110]** In the implementations, by reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a blockchain, the processor is enabled to: encrypt plaintext data of the target invoice based on a first key corresponding to the invoice issuer to obtain first ciphertext data; and encrypt the plaintext data of the target invoice based on a second key corresponding to the invoice receiver to obtain second ciphertext data; and the

adding the encrypted ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage includes: adding a corresponding relationship between the first ciphertext data and a user identifier of the invoice issuer to the blockchain for storage; and adding a  
5 corresponding relationship between the second ciphertext data and a user identifier of the invoice receiver to the blockchain for storage.

**[0111]** In the implementations, the access request can include a user identifier of the access user; and by reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a  
10 blockchain, the processor is enabled to: obtain the user identifiers of the invoice issuer and the invoice receiver of the target invoice; determine whether the user identifier of the access user matches the user identifiers of the invoice issuer or the invoice receiver; and in response to that the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver, determine that  
15 the access user has the authority to access the target invoice.

**[0112]** By reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a blockchain, the processor is enabled to: decrypt, based on a key corresponding to the access user, the ciphertext data of the target invoice stored in the blockchain.

**[0113]** In the implementations, the smart contract can further declare key distribution logic; and by reading and executing the machine-executable instructions that are stored in the memory and correspond to control logic for invoice access based on a blockchain, the processor is further enabled to: receive a registration request initiated by a user by using a client; and invoke the key distribution logic in the smart  
20 contract in response to the registration request, and write a corresponding relationship between a user identifier of the user and a key distributed to the user into the smart contract.

**[0114]** In the implementations, the invoice identifier can be a taxpayer identification number.

**[0115]** A person skilled in the art will easily figure out other implementation solutions of the present specification after considering the present specification and practicing the application disclosed herein. The present specification is intended to cover any modification, use or adaptive change of the present specification, and these  
30 modifications, uses or adaptive changes follow the general principles of the present

specification and include common knowledge or conventional technical means in the technical field not disclosed in the present specification. The present specification and implementations are regarded as example only, with the true scope and spirit of the present specification being indicated by the following claims.

5   **[0116]**   It should be understood that the present specification is not limited to the precise structure already described above and shown in the accompanying drawings, and various modifications and changes can be performed without departing from its scope. The scope of the present specification is limited only by the appended claims.

10   **[0117]**   The above descriptions are merely the preferred implementations of the present specification, and are not intended to limit the present specification. Any modifications, equivalent replacements, improvements, and the like made within the spirit and principles of the present specification should fall within the protection scope of the present specification.

## CLAIMS

1. A method based on a blockchain, comprising:

receiving a creation request initiated by a creation user for a target invoice through a client of the creation user;

determining an invoice issuer and an invoice receiver of the target invoice in response to the creation request;

invoking an encryption logic provided in a smart contract to separately encrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, and adding encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain;

receiving an access request initiated by an access user for accessing the target invoice through a client of the access user;

determining, in response to the access request, whether the access user has an authority to access the target invoice; and

in response to the access user having the authority to access the target invoice, invoking a decryption logic provided in the smart contract to decrypt the ciphertext data of the target invoice stored in the blockchain, and returning decrypted plaintext data of the target invoice to the client.

2. The method according to claim 1, wherein:

the smart contract provides an access authority verification logic for the access user; and

the determining whether the access user has the authority to access the target invoice includes:

invoking the access authority verification logic provided in the smart contract to determine whether the access user has the authority to access the target invoice.

3. The method according to claim 1, wherein the separately encrypting the plaintext data of the target invoice based on the keys corresponding to the invoice issuer and the invoice receiver includes:

encrypting plaintext data of the target invoice based on a first key corresponding to the invoice issuer to obtain first ciphertext data;

encrypting the plaintext data of the target invoice based on a second key corresponding to the invoice receiver to obtain second ciphertext data; and

the adding the encrypted ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage includes:

adding a corresponding relationship between the first ciphertext data and a user identifier of the invoice issuer to the blockchain for storage; and

adding a corresponding relationship between the second ciphertext data and a user identifier of the invoice receiver to the blockchain for storage.

4. The method according to claim 1, wherein:

the access request includes a user identifier of the access user; and

the determining whether the access user has the authority to access the target invoice includes:

obtaining the user identifiers of the invoice issuer and the invoice receiver of the target invoice;

determining whether the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver; and

in response to the user identifier of the access user matching at least one of the user identifiers of the invoice issuer or the invoice receiver, determining that the access user has the authority to access the target invoice.

5. The method according to claim 1, wherein the decrypting the ciphertext data of the target invoice stored in the blockchain includes:

decrypting, based on a key corresponding to the access user, the ciphertext data of the target invoice stored in the blockchain.

6. The method according to claim 1, wherein:

the smart contract further provides a key distribution logic; and

the method comprises:

receiving a registration request initiated by a user by using a client; and

invoking the key distribution logic in the smart contract in response to the registration request, and writing a corresponding relationship between a user identifier of the user and a key distributed to the user into the smart contract.

7. An electronic device, comprising:

a processor; and

a memory storing machine-executable instructions,

wherein by reading and executing the machine-executable instructions that are stored in the memory, the processor is enabled to implement acts including:

receiving a creation request initiated by a creation user for a target invoice through a client of the creation user;

determining an invoice issuer and an invoice receiver of the target invoice in response to the creation request; and

separately encrypting plaintext data of the target invoice based on keys corresponding to the invoice issuer and the invoice receiver, and adding encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice to a blockchain;

receiving an access request initiated by an access user for accessing the target invoice through a client of the access user;

determining, in response to the access request, whether the access user has an authority to access the target invoice; and

in response to the access user having the authority to access the target invoice, decrypting the ciphertext data of the target invoice stored in the blockchain, and returning decrypted plaintext data of the target invoice to the client.

8. The electronic device according to claim 7, wherein the determining whether the access user has the authority to access the target invoice includes:

invoking a smart contract to determine whether the access user has the authority to access the target invoice.

9. The electronic device according to claim 7, wherein the separately encrypting the plaintext data of the target invoice based on the keys corresponding to the invoice issuer and the invoice receiver includes:

encrypting plaintext data of the target invoice based on a first key corresponding to the invoice issuer to obtain first ciphertext data;

encrypting the plaintext data of the target invoice based on a second key corresponding to the invoice receiver to obtain second ciphertext data; and

the adding the encrypted ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage includes:

adding a corresponding relationship between the first ciphertext data and a user identifier of the invoice issuer to the blockchain for storage; and

adding a corresponding relationship between the second ciphertext data and a user identifier of the invoice receiver to the blockchain for storage.

10. The electronic device according to claim 7, wherein:  
the access request includes a user identifier of the access user; and  
the determining whether the access user has the authority to access the target invoice includes:

obtaining the user identifiers of the invoice issuer and the invoice receiver of the target invoice;

determining whether the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver; and

in response to the user identifier of the access user matching at least one of the user identifiers of the invoice issuer or the invoice receiver, determining that the access user has the authority to access the target invoice.

11. The electronic device according to claim 7, wherein the decrypting the ciphertext data of the target invoice stored in the blockchain includes:

decrypting, based on a key corresponding to the access user, the ciphertext data of the target invoice stored in the blockchain.

12. The electronic device according to claim 7, wherein:  
the acts include:

receiving a registration request initiated by a user by using a client; and

in response to the registration request, invoking a smart contract to distribute a key to the user and writing a corresponding relationship between a user identifier of the user and the key distributed to the user into the smart contract.

13. A non-transitory computer-readable storage medium storing computer-executable instructions, which when executed by a processor, configure the processor to implement acts including:

receiving a creation request initiated by a creation user for a target invoice through a client of the creation user;

determining an invoice issuer and an invoice receiver of the target invoice in response to the creation request; and

invoking an encryption logic provided in a smart contract to separately encrypt plaintext data of the target invoice based on keys corresponding to the invoice issuer and the

invoice receiver, and adding encrypted ciphertext data of the target invoice and user identifiers of the invoice issuer and the invoice receiver of the target invoice to a blockchain;

receiving an access request initiated by an access user for accessing the target invoice through a client of the access user;

determining, in response to the access request, whether the access user has an authority to access the target invoice; and

in response to the access user having the authority to access the target invoice, invoking a decryption logic provided in the smart contract to decrypt the ciphertext data of the target invoice stored in the blockchain, and returning decrypted plaintext data of the target invoice to the client.

14. The non-transitory computer-readable storage medium according to claim 13, wherein:

the smart contract provides an access authority verification logic for the access user; and

the determining whether the access user has the authority to access the target invoice includes:

invoking the access authority verification logic provided in the smart contract to determine whether the access user has the authority to access the target invoice.

15. The non-transitory computer-readable storage medium according to claim 13, wherein the separately encrypting the plaintext data of the target invoice based on the keys corresponding to the invoice issuer and the invoice receiver includes:

encrypting plaintext data of the target invoice based on a first key corresponding to the invoice issuer to obtain first ciphertext data;

encrypting the plaintext data of the target invoice based on a second key corresponding to the invoice receiver to obtain second ciphertext data; and

the adding the encrypted ciphertext data of the target invoice and the user identifiers of the invoice issuer and the invoice receiver of the target invoice to the blockchain for storage includes:

adding a corresponding relationship between the first ciphertext data and a user identifier of the invoice issuer to the blockchain for storage; and

adding a corresponding relationship between the second ciphertext data and a user identifier of the invoice receiver to the blockchain for storage.

16. The non-transitory computer-readable storage medium according to claim 13, wherein:

the access request includes a user identifier of the access user; and

the determining whether the access user has the authority to access the target invoice includes:

obtaining the user identifiers of the invoice issuer and the invoice receiver of the target invoice;

determining whether the user identifier of the access user matches at least one of the user identifiers of the invoice issuer or the invoice receiver; and

in response to the user identifier of the access user matching at least one of the user identifiers of the invoice issuer or the invoice receiver, determining that the access user has the authority to access the target invoice.

17. The non-transitory computer-readable storage medium according to claim 13, wherein the decrypting the ciphertext data of the target invoice stored in the blockchain includes:

decrypting, based on a key corresponding to the access user, the ciphertext data of the target invoice stored in the blockchain.

18. The non-transitory computer-readable storage medium according to claim 13, wherein:

the smart contract further provides a key distribution logic; and

the acts include:

receiving a registration request initiated by a user by using a client; and

invoking the key distribution logic in the smart contract in response to the registration request, and writing a corresponding relationship between a user identifier of the user and a key distributed to the user into the smart contract.