METHOD, APPARATUS AND ELECTRONIC DEVICE FOR BLOCKCHAIN TRANSACTIONS

CROSS REFERENCE TO RELATED APPLICATION

[0001] The present application is based on and claims priority to Chinese Patent Application No. 201811003743.9, filed on August 30, 2018, which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] One or more of embodiments of the application are related to the technical field of blockchain, particularly to a method, apparatus, and electronic device for implementing blockchain-based transactions.

BACKGROUND

[0003] Blockchain nodes of a blockchain network may jointly maintain unified blockchain ledgers through a consensus reached among the blockchain nodes, to record data of transactions. The blockchain ledgers can be made public for viewing and verification of the historical data of the transactions at any time. Therefore, current blockchain ledgers practically have no privacy protection.

SUMMARY

[0004] One or more of embodiments of the specification provide a method for implementing blockchain-based transactions. The method may comprise: determining a transaction amount to be remitted from a blockchain account of a remitter into a blockchain account of a receiver, wherein the blockchain account of the remitter records a homomorphic encryption ciphertext of the remitter's balance in a blockchain, and the blockchain account of the receiver records a homomorphic encryption ciphertext of the receiver's balance in the blockchain; generating a homomorphic encryption ciphertext of the transaction amount with respect to the remitter and a homomorphic encryption ciphertext of the transaction amount with respect to the receiver; and submitting to the blockchain a transaction comprising information of the blockchain account of the remitter, information of the blockchain account of the receiver, the homomorphic encryption ciphertext of the transaction amount with respect to the remitter and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver, for the homomorphic encryption ciphertext of the transaction amount with respect to the remitter to be subtracted from the homomorphic encryption ciphertext of the remitter's balance and for the homomorphic encryption ciphertext of the transaction amount with

respect to the receiver to be added to the homomorphic encryption ciphertext of the receiver's balance.

[0005] In some embodiments, the homomorphic encryption ciphertext of the remitter's balance is a commitment of the remitter's balance, and the homomorphic encryption ciphertext of the receiver's balance is a commitment of the receiver's balance.

[0006] In some embodiments, the homomorphic encryption ciphertext of the remitter's balance is a Pedersen Commitment of the remitter's balance, and the homomorphic encryption ciphertext of the receiver's balance is a Pedersen Commitment of the remitter's balance.

[0007] In some embodiments, the homomorphic encryption ciphertext of the remitter's balance is obtained by applying a homomorphic encryption algorithm to the remitter's balance based on a public key of the remitter and a remitter random number, and the homomorphic encryption ciphertext of the receiver's balance is obtained by applying a homomorphic encryption algorithm to the receiver's balance based on a public key of the receiver and a receiver random number.

[0008] In some embodiments, the method further comprises: generating a range proof related to the remitter's balance and the transaction amount with respect to the remitter; and adding the range proof to the transaction for the one or more blockchain nodes in the blockchain to verify whether the transaction amount is no greater than the remitter's balance.

[0009] In some embodiments, the homomorphic encryption ciphertext of the transaction amount with respect to the remitter is a commitment of the transaction amount with respect to the remitter based on the public key of the remitter, and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver is a commitment of the transaction amount with respect to the receiver based on the public key of the receiver.

[0010] In some embodiments, the method further comprises: applying a homomorphic encryption algorithm to generate the commitment of the transaction amount with respect to the remitter based on a transaction amount random number with respect to the remitter, the transaction amount, and the public key of the remitter; and applying the homomorphic encryption algorithm to generate the commitment of the transaction amount with respect to the receiver based on a transaction amount random number with respect to the receiver, the transaction amount, and the public key of the receiver.

[0011] In some embodiments, the method further comprises: determining a difference

between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver; and adding the difference between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the commitment of the transaction amount with respect to the remitter is equal to the commitment of the transaction amount with respect to the receiver.

[0012] In some embodiments, the method is implementable by a first computing device associated with the remitter, and the method further comprises: sending the transaction amount random number with respect to the receiver to a second computing device associated with the receiver via an off-chain channel.

[0013] In some embodiments, the method further comprises: generating a public key proof associated with the receiver according to a homomorphic encryption algorithm and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver; and adding the public key proof associated with the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the homomorphic encryption ciphertext of the transaction amount with respect to the receiver and the homomorphic encryption ciphertext of the receiver's balance are both encrypted based on the public key of the receiver.

[0014] In some embodiments, wherein generating a public key proof associated with the receiver according to a homomorphic encryption algorithm and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver comprises: generating a first verification random number and a second verification random number; applying the homomorphic encryption algorithm to generate a random-number homomorphic encryption ciphertext based on the first verification random number, the second verification random number, and the public key of the receiver; obtaining a hash digest by conducting a hash operation on the homomorphic encryption ciphertext of the transaction amount with respect to the receiver and the random-number homomorphic encryption ciphertext; and generating a first verification element associated with the first verification random number and a second verification element associated with the second verification random number according to the hash digest.

[0015] In some embodiments, the method further comprises: generating a digital signature of

the remitter endorsing the homomorphic encryption ciphertext of the transaction amount with respect to the remitter and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver based on a private key of the remitter; and adding the digital signature of the remitter to the transaction for one or more blockchain nodes in the blockchain to verify the digital signature of the remitter.

[0016] One or more embodiments of the specification also provide an apparatus for implementing blockchain-based transactions. The apparatus comprises one or more processors and one or more non-transitory computer-readable memories coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the apparatus to perform operations comprising: determining a transaction amount to be remitted from a blockchain account of a remitter into a blockchain account of a receiver, wherein the blockchain account of the remitter records a homomorphic encryption ciphertext of the remitter's balance in a blockchain, and the blockchain account of the receiver records a homomorphic encryption ciphertext of the receiver's balance in the blockchain; generating a homomorphic encryption ciphertext of the transaction amount with respect to the remitter and a homomorphic encryption ciphertext of the transaction amount with respect to the receiver; and submitting to the blockchain a transaction comprising information of the blockchain account of the remitter, information of the blockchain account of the receiver, the homomorphic encryption ciphertext of the transaction amount with respect to the remitter and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver, for the homomorphic encryption ciphertext of the transaction amount with respect to the remitter to be subtracted from the homomorphic encryption ciphertext of the remitter's balance and for the homomorphic encryption ciphertext of the transaction amount with respect to the receiver to be added to the homomorphic encryption ciphertext of the receiver's balance.

[0017] One or more embodiments of the specification further provide a non-transitory computer-readable storage medium configured with instructions executable by one or more processors to cause the one or more processors to perform operations comprising: determining a transaction amount to be remitted from a blockchain account of a remitter into a blockchain account of a receiver, wherein the blockchain account of the remitter records a homomorphic encryption ciphertext of the remitter's balance in a blockchain, and the blockchain account of the receiver records a homomorphic encryption ciphertext of the receiver's balance in the blockchain; generating a homomorphic encryption ciphertext of the

transaction amount with respect to the remitter and a homomorphic encryption ciphertext of the transaction amount with respect to the receiver; and submitting to the blockchain a transaction comprising information of the blockchain account of the remitter, information of the blockchain account of the receiver, the homomorphic encryption ciphertext of the transaction amount with respect to the remitter and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver, for the homomorphic encryption ciphertext of the transaction amount with respect to the remitter to be subtracted from the homomorphic encryption ciphertext of the remitter's balance and for the homomorphic encryption ciphertext of the transaction amount with respect to the receiver to be added to the homomorphic encryption ciphertext of the receiver's balance.

[0018] The method, apparatus, and computing device for implementing blockchain-based transactions described in the application provide a non-interactive transfer which allows a remitter to transfer money or assets to a receiver without the participation of the receiver, while protecting privacies such as account balances and transaction amounts. Through the embodiments described in the application, the balance of an account is visible to the owner of the account and may be hidden from other users. The transaction amount is visible to the remitter and the receiver (also referred to as a remittee) of a transaction and may be hidden from other users. Blockchain nodes can verify the validity of the transaction based on the encrypted transaction amount and update the account balances of the remitter and the remittee according to the verified transaction. The non-interactive transaction method, apparatus, and computing device have the following advantages: transactions may be initiated by a remitter without relying on a receiver's input or a network transmission of the receiver, and therefore can protect transaction execution from interferences, such as disconnection of the receiver, a delay in response, network failure and latency, etc.

BRIEF DESCRIPTION OF THE DRAWING

[0019] FIG. 1 is a flowchart of a method for implementing blockchain-based transactions according to some embodiments of the specification.

[0020] FIG. 2 is a schematic diagram of a method for implementing a remittance transaction in a blockchain network according to some embodiments of the specification.

[0021] FIG. 3 is a flowchart of a method for implementing a remittance transaction in a blockchain network according to some embodiments of the specification.

[0022] FIG. 4 is a schematic structural diagram of a device for implementing blockchain-

based transactions according to some embodiments of the specification.

[0023] FIG. 5 is a block diagram of an apparatus for implementing blockchain-based transactions according to some embodiments of the specification.

DETAILED DESCRIPTION

[0024] Embodiments will be described in detail, with examples shown in the accompanying drawings. When the description below involves the accompanying drawings, unless otherwise indicated, the same numeral in different accompanying drawings stands for the same element or similar elements. The implementation manners described in the following embodiments do not represent all the implementation manners consistent with the present specification. Rather, they are mere examples of the apparatuses and methods described in detail in the attached claims and consistent with some aspects of one or more embodiments of the present specification.

[0025] In some embodiments, it is not necessary to perform steps in the methods according to the sequence illustrated and described in the present specification. In some other embodiments, the steps of the methods may be more or fewer than those described in the present specification. Further, a single step described in the present specification may be split into multiple steps for description in other embodiments, while multiple steps described in the present specification may be merged into one single step for description in yet other embodiments.

[0026] A blockchain described in the embodiments may refer to a computer network system of blockchain nodes based on a consensus mechanism and a distributed data storage structure. The data in the blockchain is stored in multiple temporally connected "blocks," with one block containing a data digest of the preceding block in a chain. The consensus mechanism may comprise, for example, Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPOS), or Practical Byzantine Fault Tolerance (PBFT). Since blockchain systems operate under a corresponding consensus mechanism, the data recorded in the blockchain's database cannot be easily tampered with by individual blockchain nodes. Thus, the consensus mechanism may ensure a unified version of the stored data. In some embodiments, the data recorded in the distributed databases of the blockchain cannot be tampered with, thereby achieving persistent storage of blockchain-based transactions.

[0027] FIG. 1 is a flowchart of a method for implementing blockchain-based transactions according to some embodiments of the specification. As shown in FIG. 1, the method is

implementable by a computing device associated with a remitter (also referred to as a remitter device) and may comprise the following steps 102-106.

[0028] Step 102, determining a transaction amount to be remitted from a blockchain account of a remitter into a blockchain account of a receiver, wherein the blockchain account of the remitter records a homomorphic encryption ciphertext of the remitter's balance in a blockchain, and the blockchain account of the receiver records a homomorphic encryption ciphertext of the receiver's balance in the blockchain.

[0029] In some embodiments, one or more remitter users (also referred to as remitters) may negotiate with one or more receiver users (also referred to as receivers) to determine a transaction amount of a transaction (e.g., a financial transaction). The transaction may be executed in a blockchain network maintaining a blockchain and recorded in the blockchain. Thus, the transaction may be blockchain-based. A remitter and receiver may refer to an individual or entity participating the blockchain-based transaction. For example, a remitter may be an individual or entity remitting money to a receiver through a transaction. A receiver may be an individual or entity receiving money from a remitter through the transaction. The remitter may be associated with remitter devices and the receiver may be associated with receiver devices. For example, the remitter devices and receiver devices may be computing devices such as a personal computer, a laptop computer, a cell phone, a camera phone, a smart phone, a PDA (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 any combination thereof. For convenience and conciseness, a remitter or receiver described in the specification may also refer to a computing device associated with the human or entity remitter or receiver, respectively.

[0030] A blockchain-based transaction of the application can be achieved between a remitter device associated with a remitter and a receiver device associated with a receiver by remitting (or transferring) an asset certificate representing the transaction amount to be remitted from the remitter's blockchain account to the receiver's blockchain account. The asset certificate may be associated with a token, a digital asset, or any other intelligent asset in a blockchain. Alternatively, the asset certificate may also represent cash, security, discount coupon, real estate or any other off-chain asset.

[0031] In some embodiments, the homomorphic encryption ciphertext of the remitter's balance described in the foregoing embodiments may be a commitment of the remitter's

balance, and may be calculated by applying a homomorphic encryption algorithm to the remitter's balance based on a public key of the remitter (also referred to as the remitter's public key) and a remitter random number. Similarly, the homomorphic encryption ciphertext of the receiver's balance described in the foregoing embodiments may be a commitment of the receiver's balance, and may be calculated by applying a homomorphic encryption algorithm to the receiver's balance based on a public key of the receiver (also referred to as the receiver's public key) and a receiver random number. The blockchain nodes in a blockchain may respectively maintain unified blockchain ledgers based on a consensus. Therefore, by recording the ciphertext of the remitter's balance and the ciphertext of the receiver's balance in the blockchain, the blockchain ledgers maintained by the blockchain nodes may not need to record the remitter's balance and the receiver's balance. As a result, the blockchain balances held by remitter users and the blockchain balances held by receiver users are concealed as private data. Thus, evidence of the account balances of the remitters and the receivers can be stored in the blockchain without storing the account balances themselves. Further, the evidence of the account balances is difficult to be tampered with, thereby protecting the privacy of the blockchain accounts.

[0032] There are no restrictions on the methods for recording the ciphertext of the remitter's balance or the ciphertext of the receiver's balance in the blockchain. The ciphertext of the remitter's balance or the ciphertext of the receiver's balance may be published in the blockchain in the form of blockchain-based transactions, or recorded in the account attributes of the remitter account or the receiver account, or in any other forms that a person of ordinary skill in the art can come up with without creative efforts.

[0033] In some embodiments, a random number may be generated according to the homomorphic encryption algorithm, so that the random number and unencrypted data are collectively converted into corresponding ciphertext data (such as a commitment). Accordingly, corresponding decrypted data may be obtained by acquiring the random number to decrypt the ciphertext data. In addition, it may be verified whether the ciphertext data corresponds to the unencrypted data based on the random number. For example, the homomorphic encryption algorithm may be a Pedersen Commitment (PC) mechanism. The ciphertext data may be a Pedersen Commitment.

[0034] For example, the homomorphic encryption algorithm involving Pedersen Commitment (PC) is used to encrypt the remitter's account balance t. The common parameters (G, H) are two generators of an elliptic curve, and the remitter has a pair of public

and private keys (Pk, Sk) based on the elliptic curve, where Pk=SkG; the commitment of the remitter's balance t is (T, T')=(rG+tH, rPk), where rG+tH is Pedersen Commitment, and r is a random number used in the homomorphic encryption. Since the part containing the random number r, i.e., rG, can be eliminated when the remitter uses the private key Sk for decryption to inspect its account balance, the random number r does not have to be recorded. In decrypting (T, T')=(rG+tH, rPk), the key Sk may be used to obtain $rG=T'Sk^{-1}$ from T'=rPk=rSkG, thereby eliminating the part containing a random number r, i.e., rG, to obtain tH=T−rG. In addition, in order to decrypt the ciphertext tH of the remitter's account balance amount t, the following setting is implemented, i.e., the amount t has a range of validity, for example, $[0, 2^{32}]$. Many methods are available for decrypting the ciphertext tH of amount t, for example, Pollard's Kangaroo method. Alternatively, a corresponding ciphertext set of the range is calculated and stored in advance, and the value of t can be determined by looking up in a table according to the value of tH.

[0035] Similarly, the ciphertext of the receiver's balance may be generated by applying the same homomorphic encryption algorithm as that used for generating the ciphertext of the remitter's balance. For example, the commitment of the receiver's balance may be generated by using the generators G, H of the same elliptic curve and based on a public key of the receiver (also referred to as the receiver's public key) and the receiver's account balance, so that the receiver's balance can be obtained by decrypting the commitment of the receiver's balance by using the receiver's private key, thereby protecting the privacy of the user accounts in the blockchain.

[0036] Step 104, generating a homomorphic encryption ciphertext of the transaction amount with respect to the remitter and a homomorphic encryption ciphertext of the transaction amount with respect to the receiver.

[0037] In some embodiments, the homomorphic encryption ciphertext of the transaction amount with respect to the remitter is a commitment of the transaction amount with respect to the remitter based on the public key of the remitter, and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver is a commitment of the transaction amount with respect to the receiver based on the public key of the receiver.

[0038] In some embodiments, the same homomorphic encryption algorithm may be used for the commitment of the transaction amount with respect to the remitter and the commitment of the remitter's balance, as well as for the commitment of the transaction amount with respect

to the receiver and the commitment of the receiver's balance. Therefore, the commitment of the transaction amount with respect to the remitter can be subtracted from the commitment of the remitter's balance, and the commitment of the transaction amount with respect to the receiver can be subtracted from the commitment of the receiver's balance.

[0039] In some embodiments, the commitment of the transaction amount with respect to the remitter is calculated by applying a homomorphic encryption algorithm (such as the homomorphic encryption algorithm described above) according to a transaction amount random number with respect to the remitter, r_A, the transaction amount t_A, and the remitter's public key Pk_A. For example, the commitment of the transaction amount with respect to the remitter is (T_A, T'_A)=(r_AG+t_AH, r_APk_A). Similarly, the commitment of the transaction amount with respect to the receiver is calculated by applying the homomorphic encryption algorithm according to a transaction amount random number with respect to the receiver, r_B, the transaction amount t_B, and the receiver's public key Pk_B. For example, the commitment ciphertext of the transaction amount with respect to the receiver is (T_B, T'_B)= (r_BG+t_BH, r_BPk_B), where t_A=t_B. In addition, in order to enable the receiver to generate a range proof for a new transaction based on an updated account balance after receiving the transaction amount, the receiver may need to obtain the value of r_B. In order to prevent the other nodes in the blockchain from learning the value of r_B to decrypt the transaction amount, the remitter may send r_B in the transaction to the receiver via an off-chain channel. Such an action does not affect the ongoing transaction, so the random number r_B can be sent after the transaction is submitted by the remitter and implemented in the blockchain. The random number r_B can be sent via an email or other ways agreed with by the receiver in advance.

[0040] Since both the commitment of the transaction amount sent by the remitter and the commitment of the transaction amount with respect to the receiver are ciphertexts, the other nodes can learn neither the value of the transaction amount remitted by the remitter t_A, nor whether the amount t_A sent by the remitter is the same as the amount t_B received by the receiver. The receiver can know the value of t_B by decrypting the receiver's transaction commitment sent by the remitter based on the private key of the receiver. Further, in some embodiments of the application, the remitter may publish the difference between the random number r_A and the random number r_B in the blockchain, so that the blockchain nodes in the blockchain can verify whether the value of rG is equal to the value of T_A-T_B. If they are equal, it proves that t_A=t_B.

[0041] The foregoing contains a scenario where r=r_A=r_B, and t, t_A, and t_B are used to calculate the commitment of the transaction amount with respect to the remitter (T_A, T'_A)=(rG+tH, rPk_A) and the commitment of the transaction amount with respect to the receiver (T_B, T'_B)=(rG+tH, rPk_B). In such a scenario, the transaction amount does not have to be categorized as the remitter's transaction amount and the receiver's transaction amount, and the corresponding commitments may be merged as one, e.g., (rG+tH, rPk_A, rPk_B). For such a scenario, the following technical details may be simplified.

[0042] In some embodiments, the ciphertext of the transaction amount with respect to the remitter is calculated by the homomorphic encryption algorithm according to the transaction amount and the remitter's public key, and the ciphertext of the transaction amount with respect to the receiver is calculated by the homomorphic encryption algorithm according to the transaction amount and the receiver's public key. Therefore, the other nodes in the blockchain (such as blockchain nodes other than those associated with the remitter and the receiver) may not know from the ciphertext of the transaction amount with respect to the receiver whether the ciphertext of the transaction amount with respect to the receiver is based on the receiver's public key (or whether it is the same as the public key used in the ciphertext of the receiver's balance), not to mention whether the ciphertext of the transaction amount with respect to the receiver can be decrypted with the receiver's private key to obtain the transaction amount. In order to help the other nodes verify the ciphertext of the transaction amount with respect to the receiver released by the remitter and prevent transaction fraud potentially committed by the remitter, the remitter may also send a public key proof associated with the receiver to the blockchain, so that the blockchain nodes in the blockchain can verify that the ciphertext of the transaction amount with respect to the receiver and the ciphertext of the receiver's balance are calculated by the homomorphic encryption algorithm based on the same public key. In some embodiments, the public key proof associated with the receiver is generated based on the homomorphic encryption algorithm and the ciphertext of the transaction amount with respect to the receiver.

[0043] In some embodiments, the remitter generates the public key proof Pf in the following manner, to prove that the commitment of the transaction amount with respect to the receiver (T_B, T'_B)= (r_BG+t_BH, r_BPk_B) is encrypted with the public key Pk_B of the receiver: generating a pair of random numbers (r', t'), where r' is a first verification random number and t' is a second verification random number, and calculating the random number commitment (T, T')= (r' G+ t' H, r' Pk_B) by using the same homomorphic encryption algorithm used in

generating a commitment of the transaction amount with respect to the receiver, where Pk_B is the public key used in the commitment of the receiver's balance; calculating a hash u=Hash(T_B, T'_B, T, T') and v=u r_B+ r', w=u t_B+ t', where Hash represents a hash operation (such as an operation based on a hash function which returns a hash digest), v is a first verification element corresponding to the first verification random number r', and w is a second verification element corresponding to the second verification random number t'; and generating Pf= (T, T'; v, w).

[0044] In verifying whether the public key used in the commitment of the transaction amount with respect to the receiver sent by the remitter is the same as the public key used in the commitment of the receiver's balance, the other nodes in the blockchain acquire the foregoing Pf= (T, T'; v, w) to verify whether (vG+wH, v Pk_B)=(uT_B+ T, uT'_B+ T') is correct.

[0045] If correct, it indicates that the commitment of the transaction amount with respect to the receiver and the commitment of the receiver's balance are calculated by the homomorphic encryption algorithm based on the same public key, and the commitment of the transaction amount with respect to the receiver can be decrypted with a private key corresponding to the foregoing public key to obtain the transaction amount. If not correct, the transaction may be rejected. There may be serious consequences if not rejecting such transaction. For example, the remitter may have encrypted the transaction amount with the public key which is not the receiver's public key, to generate the commitment of the transaction amount with respect to the receiver, or have used a number to falsify the foregoing commitment of the transaction amount with respect to the receiver. In this case, if the commitment of the transaction amount with respect to the receiver is updated to the commitment of the receiver's balance, the receiver cannot decrypt the commitment of his or her balance and cannot remit assets from his or her account, and the receiver's account can be impaired.

[0046] The implementation manner of the foregoing public key proof associated with the receiver can prevent accounts from being impaired, without requiring the remittee's participation, which is therefore a non-interactive way.

[0047] As described in the foregoing embodiments, the commitment of the remitter's balance is calculated by the homomorphic encryption algorithm based on the remitter random number, the remitter's balance, and the remitter's public key. The commitment of the remitter's balance is (S_A, S'_A) = (r'_A G+s_A H, r'_A Pk_A), where s_A is the remitter's balance.

The commitment of the transaction amount with respect to the remitter is calculated by the homomorphic encryption algorithm according to the transaction amount random number with respect to the remitter, the transaction amount and the remitter's public key, i.e., the commitment of the transaction amount with respect to the remitter is (T_A, T'_A)=(r_AG+t_AH, r_APk_A), where t_A is the transaction amount. A remitter device may generate a range proof according to the remitter random number r'_A, the remitter's balance s_A, the commitment of the remitter's balance (S_A, S'_A), the transaction amount random number with respect to the remitter r_A, the transaction amount t_A, and the commitment of the transaction amount with respect to the remitter (T_A, T'_A). For instance, range proof technologies such as a Bulletproofs method or a Borromean ring signature method in related technologies may be used. The remitter device may add the range proof in the transaction, so that the blockchain nodes in the blockchain can verify whether the transaction amount t_A is no smaller than zero and no greater than the remitter's balance s_A.

[0048] A digital signature may provide validation or authentication of the data that the digital signature is signed against. A digital signature may achieved by asymmetric cryptography. In some embodiments, a remitter device may generate a digital signature endorsing the ciphertext of the transaction amount with respect to the remitter and the ciphertext of the transaction amount with respect to the receiver via a private key of the remitter, and add the digital signature to the transaction. The remitter device may submit the transaction to the blockchain so that the blockchain nodes in the blockchain can verify the digital signature. In some embodiments, the remitter's digital signature may also endorse the data that has been signed against, such as a difference between the transaction amount random number with respect to the remitter and a transaction amount random number with respect to the receiver, the public key proof associated with the receiver, and/or the range proof. When a transaction does not contain this remitter's digital signature, the blockchain nodes may determine that consensus fails and refuse to proceed with this transaction.

[0049] Step 106, submitting to the blockchain a transaction comprising information of the blockchain account of the remitter, information of the blockchain account of the receiver, the homomorphic encryption ciphertext of the transaction amount with respect to the remitter and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver, for the homomorphic encryption ciphertext of the transaction amount with respect to the remitter to be subtracted from the homomorphic encryption ciphertext of the remitter's balance and for the homomorphic encryption ciphertext of the transaction amount with

respect to the receiver to be added to the homomorphic encryption ciphertext of the receiver's balance.

[0050] For example, the subtraction and addition in Step 106 may be implemented according to the following equations:

$$(S\_A, S'\_A) - (T\_A, T'\_A) = (S\_A-T\_A, S'\_A-T'\_A),$$

$$(S\_B, S'\_B) + (T\_B, T'\_B) = (S\_B+T\_B, S'\_B+T'\_B).$$

That is, subtracting the commitment of the transaction amount with respect to the remitter (T_A, T'_A) from the commitment of the remitter's balance (S_A, S'_A) equals to the commitment of the result of subtracting the transaction amount with respect to the remitter from the remitter's balance; similarly, adding the commitment of the transaction amount with respect to the receiver to the commitment of the receiver's balance equals to the commitment of the sum of the transaction amount with respect to the receiver and the receiver's balance.

[0051] In the foregoing embodiments, the ciphertext of the transaction amount with respect to the remitter and the ciphertext of the transaction amount with respect to the receiver are used in a transaction, so that the blockchain ledgers may record the transaction ciphertexts, and may not record the corresponding transaction amount, thereby concealing the transaction amount. In addition, the ciphertext of the remitter's balance corresponding to a remitter's balance, a ciphertext of the receiver's balance corresponding to a receiver's balance, and a ciphertext of the transaction amount with respect to the remitter and a ciphertext of the transaction amount with respect to the receiver corresponding to a transaction amount are generated by the same homomorphic encryption algorithm. Accordingly, the other blockchain nodes may subtract the ciphertext of the transaction amount with respect to the remitter from the ciphertext of the remitter's balance, and add the ciphertext of the receiver's balance and the receiver's transaction ciphertext, without the need to acquire the remitter's balance, the receiver's balance and the transaction amount, so as to ensure protection of the private data.

[0052] In some embodiments, after the blockchain nodes receive the foregoing transaction, the blockchain nodes may use double-spending prevention mechanism or replay attack prevention mechanism to verify if the transaction has been executed. If the transaction has been executed, the transaction may be rejected.

[0053] In some embodiments, a remittance transaction in a blockchain network is taken below as an example to describe the application in detail. FIG. 2 is a schematic diagram of implementing a remittance transaction in a blockchain network according to some

embodiments of the specification. As shown in FIG. 2, assuming that a user A remits to a user B in a blockchain. A "user" in the application may be manifested as a signed-in user account, while the user account may belong to an individual or an organization.

[0054] It is assumed that a remitter device used by the user A is a user device 1, for example, a user account associated with the user A is logged in on the user device 1. Similarly, a receiver device used by the user B is a user device 2. A client program of blockchain may be run on the user device 1, so that the user device 1 runs a corresponding blockchain node in the blockchain network, e.g., a node 1 shown in FIG. 2. Similarly, a client program of blockchain may be run on the user device 2 so that the user device 2 runs a corresponding blockchain node in the blockchain network, e.g., a node 2 shown in FIG. 2. The blockchain network also has other blockchain nodes, such as node i shown in FIG. 2, which will not be enumerated herein. Through the foregoing node 1, node 2, etc., the remittance transaction between the user A and the user B may be implemented via the blockchain network, and related transaction information may be recorded to the blockchain ledgers maintained by blockchain nodes respectively, thereby preventing the recorded transaction information from being tampered with, and also facilitating subsequent inspection.

[0055] With respect to the scenario of the remittance transaction shown in FIG. 2, FIG. 3 is a flowchart of implementing a remittance transaction in a blockchain network according to some embodiments of the specification. As shown in FIG. 3, the process of interaction among the remitter and blockchain nodes may comprise the following steps 301-316.

[0056] Step 301, determining a transaction amount t_A to be remitted from a remitter's blockchain account into a receiver's blockchain account.

[0057] In some embodiments, a remitter refers to the individual or entity that remits a fund and other resources in a remittance transaction, and correspondingly a receiver refers to the individual or entity that receives the fund and other resources in the remittance transaction. For example, in the embodiments shown in FIG. 2, the user device 1 may be configured as the remitter, while the user device 2 may be configured as the receiver.

[0058] The commitment of the remitter's balance is recorded by the remitter's blockchain account in the blockchain:

$(S_A, S'_A)=(r'_A G+s_A H, r'_A Pk_A)$.

[0059] The commitment of the receiver's balance is recorded by the receiver's blockchain account in the blockchain:

(S_B, S'_B)=(r'_B G+s_B H, r'_B Pk_B).

[0060] The commitment of the remitter's balance (S_A, S'_A) is calculated by the homomorphic encryption algorithm according to the remitter's balance s_A and the remitter's public key Pk_A. The commitment of the receiver's balance (S_B, S'_B) is calculated by the homomorphic encryption algorithm according to the receiver's balance s_B and the receiver's public key Pk_B. A balance random number with respect to the remitter r'_A and a balance random number with respect to the receiver r'_B are random numbers used by the homomorphic encryption algorithm, and G, H are the two generators of the elliptic curve algorithm.

[0061] Step 302, generating a commitment of the transaction amount with respect to the remitter (T_A, T'_A)=(r_AG+t_AH, r_APk_A) and the commitment of the transaction amount with respect to the receiver (T_B, T'_B)= (r_BG+t_BH, r_BPk_B).

[0062] The commitment of the transaction amount with respect to the remitter (T_A, T'_A) is calculated by the homomorphic encryption algorithm according to the transaction amount t_A and the remitter's public key Pk_A. The commitment of the transaction amount with respect to the receiver (T_B, T'_B) is calculated by the homomorphic encryption algorithm according to the transaction amount t_B and the receiver's public key Pk_B, and t_B=t_A. The transaction amount random number with respect to the remitter r_A and the transaction amount random number with respect to the receiver r_B are the random numbers used by the foregoing homomorphic encryption algorithm. The two generators G, H of the foregoing elliptic curve algorithm are still used in the generation of the foregoing ciphertexts.

[0063] Further, in order to enable the other nodes in the blockchain to verify t_B=t_A, r=r_A-r_B is to be generated in the next step 303.

[0064] Step 303, generating a difference between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver: r=r_A-r_B.

[0065] In some embodiments, in order to ensure that a remittance transaction is implemented smoothly, blockchain nodes may determine that the values of the remitted amount t_A and the account balance s_A meet the following conditions: $t\_A \geq 0$ and $s_A - t\_A \geq 0$. In some embodiments, a range proof technology enables the blockchain nodes to verify whether the transaction meets preset conditions in a ciphertext state. For example, the Bulletproofs or the Borromean ring signature may be used to achieve this goal. The remitter may generate a

range proof PR related to (r_A, s_A, S_A, r, t_A, T) for the blockchain nodes to verify if t_A≥0 and s_A-t_A≥0 are met in the subsequent process. For example, as shown in FIG. 3:

[0066] Step 304, generating a range proof RP1 to prove t_A≥0.

[0067] Step 305, generating a range proof RP2 to prove s_A-t_A≥0.

[0068] For the nodes in the blockchain to verify that the commitment of the transaction amount with respect to the receiver (T_B, T'_B) is encrypted in the same encryption mode as the commitment of the receiver's balance (s_B, s'_B), and that the commitment of the transaction amount with respect to the receiver (T_B, T'_B) can be decrypted with the receiver's private key to correctly obtain the transaction amount, and for the nodes to update the account balance by using the transaction amount, the remitter may also generate a public key proof associated with the receiver, as shown in FIG. 3.

[0069] Step 306, generating a proof Pf to prove that (T_B, T'_B) is encrypted by using receiver's public key Pk_B.

[0070] In some embodiments, generating the proof Pf may comprise: generating a pair of random numbers (r', t') where r' is a first verification random number and t' is a second verification random number; calculating the random number commitment (T, T')= (r' G+ t' H, r' Pk_B) by the same homomorphic encryption algorithm as used in generating the commitment of the transaction amount with respect to the receiver, where Pk_B is the public key used in the commitment of the receiver's balance; calculating a hash u=Hash(T_B, T'_B, T, T'), v=u r_B+ r', w=u t_B+ t', where Hash represents a hash operation, v is the a verification element corresponding to the first verification random number r', and w is a second verification element corresponding to the second verification random number t'; generating Pf= (T, T'; v, w).

[0071] Step 307, the remitter uses the private key corresponding to the remitter's public key Pk_A to sign (A, T_A, T'_A; B, T_B, T'_B) to generate a digital signature Sig_A, where A, B represent the identification information of the remitter and receiver respectively, such as the remitter's and the receiver's blockchain accounts or blockchain account addresses, respectively, to enable the other nodes in the blockchain to verify the remitter and the receiver involved in this transaction.

[0072] In some embodiments, the remitter's digital signature may endorse not only A, T_A, T'_A, B, T_B, T'_B, but also one or more of r, RP1, RP2, and Pf.

[0073] Step 308, the remitter submits the foregoing generated ciphertexts (T_A, T'_A; T_B, T'_B) as well as the identification information, such as the remitter's and the receiver's accounts or account addresses, in the format of a transaction to the blockchain network for the nodes in the blockchain to conduct consensus verification on this transaction. If the transaction is consensus-verified, it may be added to a new block of the blockchain, thus added in the distributed ledgers of the blockchain. In some embodiments, for the nodes in the blockchain to conduct consensus verification on this transaction, the generated proof parameters (r, RP1, RP2, Pf) and the digital signature Sig_A may be added into the content of the transaction.

[0074] In some embodiments, the remitter submits the transaction to the blockchain via a node 1 to execute the remittance. The transaction may be transmitted to all or some blockchain nodes in the blockchain network and the all or some blockchain nodes may verify the content and format of the transaction respectively and execute the remittance transaction when the verification is successful, or reject the remittance transaction when the verification is not successful.

[0075] Each of the blockchain nodes may be any blockchain node in a blockchain network. Every blockchain node in the blockchain network may receive the transaction, for example, (A, T_A, T'_A; B, T_B, T'_B; r, RP1, RP2, Pf; Sig_A), and conduct verification and other operations through Steps 309-312.

[0076] Step 309, the blockchain nodes check if the transaction has been executed.

[0077] In some embodiments, after the blockchain nodes receive the remittance transaction (A, T_A, T'_A; B, T_B, T'_B; r, RP1, RP2, Pf; Sig_A), the blockchain nodes may use the anti-double spending or anti-replay mechanism in blockchain-related technologies to verify if this remittance transaction has been executed. If this remittance transaction has been executed, the execution of this remittance transaction is repetitive and may be rejected. Otherwise the process moves to Step 310.

[0078] Step 310, the blockchain nodes check the signature Sig_A.

[0079] In some embodiments, the blockchain nodes may check if the signature Sig_A contained in the remittance transaction is correct. If the signature is not correct, the execution of this remittance transaction may be rejected. Otherwise the process proceeds to Step 311.

[0080] Step 311, the blockchain nodes verify whether t_A is equal to t_B.

[0081] In some embodiments, whether t_A is equal to t_B can be verified by verifying whether rG= T_A-T_B is correct. If not correct, the execution of this remittance transaction may be rejected. Otherwise, the process proceeds to Step 312.

[0082] Step 312, the blockchain nodes verify t_A≥ 0;

[0083] In some embodiments, the blockchain nodes may check the range proof PR1 contained in this remittance transaction based on a range proof technology to determine whether t_A≥ 0 is met. If the condition t_A≥ 0 is not met, the execution of this remittance transaction may be rejected. Otherwise, the process proceeds to Step 313.

[0084] Step 313, the blockchain nodes verify s_A-t_A≥0;

[0085] In some embodiments, the blockchain nodes may check the range proof PR2 contained in this remittance transaction based on a range proof technology to determine whether s_A-t_A≥0 is met. If the condition s_A-t_A≥0 is not met, the execution of this remittance transaction may be rejected. Otherwise, the process proceeds to Step 314.

[0086] Step 314, the blockchain nodes verify whether the amount ciphertext (T_B, T'_B) is calculated by the homomorphic encryption algorithm based on the same public key Pk_B as the commitment of the receiver's balance (s_B, s'_B).

[0087] In some embodiments, the blockchain nodes may verify if (vG+wH, v Pk_B) = (uT_B+ T, uT'_B+ T') is correct for all parameters in the Pf contained in the remittance transaction. If not correct, the execution of this remittance transaction may be rejected. Otherwise, the process proceeds to Step 315.

[0088] Step 315, in the maintained blockchain ledgers, the blockchain nodes update the balances of the blockchain accounts associated with the user A and the user B respectively.

[0089] In some embodiments, after the verification in Steps 309 ~ 314 is successful, the blockchain nodes may deduct the commitment of the transaction amount with respect to the remitter (T_A, T'_A) from the balance commitment (S_A, S'_A) corresponding to the remitter's account balance s_A recorded in the blockchain ledger, and add the commitment of the transaction amount with respect to the receiver (T_B, T'_B) to the balance commitment (S_B, S'_B) corresponding to the receiver's account balance s_B, so that the balance commitment in the blockchain account 1 associated with the user A is updated to s_A-t_A, and the balance commitment in the blockchain account 2 associated with the user B is updated to s_B+t_B, where t_A= t_B.

[0090] Based on the nature of homomorphic encryption algorithms, when the balance commitment (S_A, S'_A) is updated to [(S_A, S'_A) - (T_A, T'_A)], since (S_A, S'_A)-(T_A, T'_A)= (S_A-T_A, S'_A-T'_A), and (S_A-T_A, S'_A-T'_A) = (r'_A- r_A)G+ (s_A-t_A)H, (r'_A-r_A)Pk_A), the remitter's account balance is updated to (s_A-t_A) after the consensus verification by the blockchain nodes succeeds, while the other nodes in the blockchain do not know the value of the remitter's account balance s _A and the transaction amount t_A remitted by the remitter.

[0091] Similarly, based on the nature of homomorphic encryption algorithms, when the balance commitment (S_B, S'_B) is updated to [(S_B, S'_B)+(T_B, T'_B)], since (S_B, S'_B)+(T_B, T'_B)= (S_B+T_B, S'_B+T'_B), and (S_B+T_B, S'_B+T'_B)= (r'_B+ r_B)G+ (s_B+t_B)H, (r'_B+r_B)Pk_B), the receiver's account balance is updated to (s_B+t_B) after the consensus verification by the blockchain nodes succeeds, while the other nodes in the blockchain do not know the value of the receiver's account balance s _B and the transaction amount t_B remitted by the remitter.

[0092] In some embodiments, in order to ensure that the receiver can conduct transfer transaction based on the receiver's account balance after accepting this transaction, the foregoing embodiments may comprise Step 316: the remitter may send the foregoing transaction amount random number with respect to the receiver r_B to the receiver via an off-chain channel.

[0093] In summary, the use of the homomorphic encryption mechanism allows the encryption of the balances in blockchain accounts and the encryption of the remitted amounts in the process of remittance transactions. The encrypted balance ciphertexts may be recorded in the blockchain accounts, and the encrypted remittance ciphertexts may be used in the implementation of remittance transactions. Therefore, the remittance transactions can be implemented successfully via the blockchain network without revealing the account balances and remitted amounts, and without affecting the blockchain nodes' verification of transaction conditions, thereby enabling the blockchain network to provide a privacy protection function. In the foregoing embodiments provided in the application, the remitter may generate a remittance transaction along with verification parameters and submits the transaction to the blockchain network, the other nodes in the blockchain may conduct consensus verification on the transaction content, and then the remittance operation can be completed. The entire process may not require the involvement of the receiver's node device, and may not rely on the receiver device and the network transmission of the receiver device, thereby eliminating

the interference caused by an unconnected status of the receiver, a delay in response, network failure and latency, etc.

[0094] In some embodiments, the proofs or digital signatures (such as r, RP1, RP2, Pf and Sig_A) may be generated in an order different from the one described in FIG. 3. Similarly, the contents and the proofs or digital signatures in the transaction to the remitter may be verified by the nodes in the blockchain in a different order than the one described in FIG. 3.

[0095] FIG. 4 is a schematic structural diagram of a device for implementing blockchain-based transactions according to some embodiments of the specification. As shown in FIG. 4, at the level of hardware, the device comprises a processor 402, an internal bus 404, a network interface 406, an internal memory 408, and a non-volatile memory (NVM) 410. The device may further comprise other pieces of hardware for transactions. The processor 402 reads a corresponding computer program from the NVM 410 to the internal memory 408 and then executes the computer program to form a blockchain transaction apparatus at a logical level. In addition to the software implementation manner, one or more of embodiments of the application do not exclude other implementation manners, such as a logic device or a combination of software and hardware. That is, the executive entities of the following processes are not limited to logical units, and may also be hardware or logic devices.

[0096] Referring to FIG. 5, a block diagram of an apparatus for implementing blockchain-based transactions is provided according to some embodiments of the specification. As shown in FIG. 5, in the software implementation manner, the apparatus for implementing blockchain-based transactions is implemented as a remitter device 50 and may comprise: a determination unit 502 for determining a transaction amount to be remitted from a blockchain account of a remitter into a blockchain account of a receiver, wherein the blockchain account of the remitter records a homomorphic encryption ciphertext of the remitter's balance in a blockchain, the blockchain account of the receiver records a homomorphic encryption ciphertext of the receiver's balance in the blockchain, the homomorphic encryption ciphertext of the remitter's balance may be a commitment of the remitter's balance based on a public key of the remitter, and the homomorphic encryption ciphertext of the receiver's balance may be a commitment of a receiver's balance based on a public key of the receiver; a generation unit 504 for generating a homomorphic encryption ciphertext of the transaction amount with respect to the remitter and a homomorphic encryption ciphertext of the transaction amount with respect to the receiver, wherein the homomorphic encryption ciphertext of the transaction amount with respect to the remitter may be a commitment of the transaction

amount based on the public key of the remitter, and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver may be a commitment of the transaction amount based on the public key of the receiver; a submission unit 506 for submitting to the blockchain a transaction comprising information of the blockchain account of the remitter, information of the blockchain account of the receiver, the homomorphic encryption ciphertext of the transaction amount with respect to the remitter and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver, for the homomorphic encryption ciphertext of the transaction amount with respect to the remitter to be subtracted from the homomorphic encryption ciphertext of the remitter's balance and for the homomorphic encryption ciphertext of the transaction amount with respect to the receiver to be added to the homomorphic encryption ciphertext of the receiver's balance.

[0097] In some embodiments, the commitment of the transaction amount with respect to the remitter is calculated by applying a homomorphic encryption algorithm according to the transaction amount random number with respect to the remitter, the transaction amount and the remitter's public key; and the commitment of the transaction amount with respect to the receiver is calculated by applying the homomorphic encryption algorithm according to the transaction amount random number with respect to the receiver, the transaction amount and the receiver's public key.

[0098] In some embodiments, the generation unit 504 may be configured to: determine a difference between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver; add the difference between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the transaction amount encrypted as the commitment of the transaction amount with respect to the remitter is equal to the transaction amount encrypted as the commitment of the transaction amount with respect to the receiver.

[0099] In some embodiments, the generation unit 504 may further be configured to: generate a public key proof associated with the receiver based on a homomorphic encryption algorithm and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver; and add the public key proof associated with the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the homomorphic encryption ciphertext of the transaction amount with respect to the receiver and the

homomorphic encryption ciphertext of the receiver's balance are calculated by the homomorphic encryption algorithm based on a same public key.

[00100] In some embodiments, the generation unit 504 may further be configured to: generate a first verification random number and a second verification random number; generate a random-number homomorphic encryption ciphertext by the homomorphic encryption algorithm based on the first verification random number, the second verification random number and the public key of the receiver; obtain a hash digest by conducting a hash operation on the homomorphic encryption ciphertext of the transaction amount with respect to the receiver and the random-number homomorphic encryption ciphertext; generate a first verification element associated with the first verification random number and a second verification element associated with the second verification random number according to the hash digest, wherein the public key proof associated with the receiver comprises the random number commitment ciphertext, the first verification element and the second verification element.

[00101] In some embodiments, the commitment of the remitter's balance is calculated by a homomorphic encryption algorithm according to the remitter's balance, the remitter's public key and a remitter random number.

[00102] In some embodiments, the generation unit 504 may further be configured to: generating a range proof related to the remitter's balance and the transaction amount with respect to the remitter; and add the range proof to the transaction for the one or more blockchain nodes in the blockchain to verify whether the transaction amount is no smaller than zero and no greater than the remitter's balance.

[00103] In some embodiments, the generation unit 504 may further be configured to: generate a digital signature of the remitter endorsing the homomorphic encryption ciphertext of the transaction amount with respect to the remitter and the homomorphic encryption ciphertext of the transaction amount with respect to the receiver based on a private key of the remitter; add the digital signature of the remitter to the transaction for one or more blockchain nodes in the blockchain to verify the digital signature of the remitter.

[00104] In some embodiments, the foregoing apparatus further comprises: an off-chain sending unit for sending the transaction amount random number with respect to the receiver to the receiver via an off-chain channel.

[00105] For details of how the functions and roles of the units of the foregoing apparatus are

implemented, the embodiments of the foregoing method may be referenced to.

[00106] The systems, apparatuses, modules or units described in the foregoing embodiments may be achieved by computer chips or entities, or by products with certain functions. A typical achievement device is a computer. An example of the computer may be a personal computer, a laptop computer, a cell phone, a camera phone, a smart phone, a PDA (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 any combination of a few of these devices.

[00107] The embodiments of the application also provide a computer device which comprises a memory and a processor, where the memory is configured with computer programs which can be executed by the processor, and the processor executes the computer programs to achieve the steps of the methods of blockchain transactions in the embodiments of the application. The steps of the method for implementing blockchain-based transactions are detailed in the foregoing description.

[00108] The foregoing descriptions are embodiments of the application and are not intended to limit the application. All modifications, equivalent replacements and improvements made without departing from the spirit and principle of the application shall fall within the scope of the application.

[00109] In a typical configuration, a computing device comprises one or more of processors (CPU), I/O interfaces, network interfaces and internal memories. The internal memory may be in a form of volatile memory, random access memory (RAM) and/or non-volatile memory such as read-only memory (ROM) or flash memory (flash RAM) in computer readable media. Internal memory is an example of the computer readable media.

[00110] Computer readable media include non-volatile and volatile, movable and non-movable media and may achieve information storage by any method or technology. Information may be computer readable instructions, data structures, program modules or other data.

[00111] Examples of computer storage media include, but are not limited to, phase change random access memory (PRAM), static random access memory (SRAM), dynamic random access memory (DRAM), other types of random access memory (RAM), read only memory (ROM), electrically erasable programmable read only memory (EEPROM), flash memory or other memory technologies, compact disc – read only memory (CD-ROM), digital versatile

disc (DVD) or other optical memory, cassette type magnetic tape, magnetic disk memory, magnetic tape, magnetic disc storage or other magnetic storage devices or any other non-transmission media, which may be used to store information that is accessible to computer devices. According to the specification, computer readable media do not include transitory media, such as modulated data signals and carriers.

[00112] It should be further noted that the terms "include," "contain" or any variants thereof are intended to cover non-exclusive inclusion so that a process, method, product or device that includes a set of elements not only includes those elements but also includes other elements that are not expressly listed, or also includes the elements inherent to such process, method, product or device. In the absence of more limitations, the elements defined by the sentence "include one..." do not exclude that other equivalent elements are included in the process, method, product or device.

[00113] One of ordinary skill in the art should understand that the embodiments of the application may be provided as a method, a system or a computer program product. Therefore, the embodiments of the application may be entire hardware embodiments, entire software embodiments or embodiments combining hardware and software. Further, the embodiments of the application may adopt the form of a computer program product implemented on one or more of computer usable storage media (including, but not limited to, magnetic disk memory, CD-ROM and optical memory) containing computer readable program codes.

What is claimed is:

1. A blockchain-based transaction method, applicable at a remitter's equipment, the method comprising:

determining a transaction amount to be remitted from a blockchain account of the remitter into a blockchain account of a receiver, wherein the blockchain account of the remitter records a commitment ciphertext of the remitter's balance in a blockchain, and the blockchain account of the receiver records a commitment ciphertext of the receiver's balance in the blockchain, wherein the commitment ciphertext of the remitter's balance is obtained by applying a homomorphic encryption algorithm based on the remitter's balance and a public key of the remitter, and the commitment ciphertext of the receiver's balance is obtained by applying the homomorphic encryption algorithm based on the receiver's balance and a public key of the receiver;

generating a commitment ciphertext of the transaction amount with respect to the remitter and a commitment ciphertext of the transaction amount with respect to the receiver, wherein the commitment ciphertext of the transaction amount with respect to the remitter is generated by applying the homomorphic encryption algorithm based on the transaction amount and the public key of the remitter, and the commitment ciphertext of the transaction amount with respect to the receiver is generated by applying the homomorphic encryption algorithm based on the transaction amount and the public key of the receiver; and

submitting to the blockchain a transaction comprising information of the blockchain account of the remitter, information of the blockchain account of the receiver, the commitment ciphertext of the transaction amount with respect to the remitter, and the commitment ciphertext of the transaction amount with respect to the receiver, for the commitment ciphertext of the transaction amount with respect to the remitter to be subtracted from the commitment ciphertext of the remitter's balance and for the commitment ciphertext of the transaction amount with respect to the receiver to be added to the commitment ciphertext of the receiver's balance;

the method further comprising:

generating a public key proof associated with the receiver, wherein the public key proof associated with the receiver is generated by applying the homomorphic encryption algorithm

based on the commitment ciphertext of the transaction amount with respect to the receiver; and

adding the public key proof associated with the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the commitment ciphertext of the transaction amount with respect to the receiver and the commitment ciphertext of the receiver's balance are both obtained by applying the homomorphic encryption algorithm based on the public key of the receiver.

2. The method of claim 1, wherein:

the commitment ciphertext of the transaction amount with respect to the remitter is generated by applying a homomorphic encryption algorithm based on a transaction amount random number with respect to the remitter, the transaction amount, and the public key of the remitter; and

the commitment ciphertext of the transaction amount with respect to the receiver is generated by applying the homomorphic encryption algorithm based on a transaction amount random number with respect to the receiver, the transaction amount, and the public key of the receiver,

the method further comprising:

generating a difference between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver; and

adding the difference between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the transaction amount encrypted in the commitment ciphertext of the transaction amount with respect to the remitter is equal to the transaction amount encrypted in the commitment ciphertext of the transaction amount with respect to the receiver.

3. The method of claim 1, wherein generating the public key proof associated with the receiver comprises:

generating a first verification random number and a second verification random number;

applying the homomorphic encryption algorithm to generate a random-number commitment ciphertext based on the first verification random number, the second verification random number, and the public key of the receiver;

obtaining a hash digest by conducting a hash operation on the commitment ciphertext of the transaction amount with respect to the receiver and the random-number commitment ciphertext; and

generating a first verification element associated with the first verification random number and a second verification element associated with the second verification random number according to the hash digest,

wherein the public key proof associated with the receiver comprises the random-number commitment ciphertext, the first verification element, and the second verification element.

4. The method of claim 3, wherein the commitment ciphertext of the remitter's balance is obtained by applying the homomorphic encryption algorithm based on the remitter's balance, a public key of the remitter, and a remitter random number;

the method further comprising:

generating a range proof based on the remitter random number, the transaction amount random number with respect to the remitter, the remitter's balance, the commitment ciphertext of the remitter's balance, the transaction amount, and the commitment ciphertext of the transaction amount with respect to the remitter; and

adding the range proof to the transaction for the one or more blockchain nodes in the blockchain to verify whether the transaction amount satisfies: the transaction amount being no less than 0 and no greater than the remitter's balance.

5. The method of any one of claims 1-4, further comprising:

generating, based on a private key of the remitter, a digital signature of the remitter related to the commitment ciphertext of the transaction amount with respect to the remitter and

the commitment ciphertext of the transaction amount with respect to the receiver; and

adding the digital signature to the transaction for one or more blockchain nodes in the blockchain to verify the digital signature.

6. The method of claim 2, further comprising:

sending the transaction amount random number with respect to the receiver to the receiver via an off-chain channel.

7. A blockchain-based transaction device, applicable at a remitter's equipment, the device comprising:

a determining unit determining a transaction amount to be remitted from a blockchain account of the remitter into a blockchain account of a receiver, wherein the blockchain account of the remitter records a commitment ciphertext of the remitter's balance in a blockchain, and the blockchain account of the receiver records a commitment ciphertext of the receiver's balance in the blockchain, wherein the commitment ciphertext of the remitter's balance is obtained by applying a homomorphic encryption algorithm based on the remitter's balance and a public key of the remitter, and the commitment ciphertext of the receiver's balance is obtained by applying the homomorphic encryption algorithm based on the receiver's balance and a public key of the receiver;

a generating unit generating a commitment ciphertext of the transaction amount with respect to the remitter and a commitment ciphertext of the transaction amount with respect to the receiver, wherein the commitment ciphertext of the transaction amount with respect to the remitter is generated by applying the homomorphic encryption algorithm based on the transaction amount and the public key of the remitter, and the commitment ciphertext of the transaction amount with respect to the receiver is generated by applying the homomorphic encryption algorithm based on the transaction amount and the public key of the receiver; and

a submitting unit submitting to the blockchain a transaction comprising information of the blockchain account of the remitter, information of the blockchain account of the receiver, the commitment ciphertext of the transaction amount with respect to the remitter, and the

commitment ciphertext of the transaction amount with respect to the receiver, for the commitment ciphertext of the transaction amount with respect to the remitter to be subtracted from the commitment ciphertext of the remitter's balance and for the commitment ciphertext of the transaction amount with respect to the receiver to be added to the commitment ciphertext of the receiver's balance;

wherein the generating unit:

generating a public key proof associated with the receiver, wherein the public key proof associated with the receiver is generated by applying the homomorphic encryption algorithm based on the commitment ciphertext of the transaction amount with respect to the receiver; and

adding the public key proof associated with the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the commitment ciphertext of the transaction amount with respect to the receiver and the commitment ciphertext of the receiver's balance are both obtained by applying the homomorphic encryption algorithm based on the public key of the receiver.

8. The device of claim 7, wherein:

the commitment ciphertext of the transaction amount with respect to the remitter is generated by applying a homomorphic encryption algorithm based on a transaction amount random number with respect to the remitter, the transaction amount, and the public key of the remitter; and

the commitment ciphertext of the transaction amount with respect to the receiver is generated by applying the homomorphic encryption algorithm based on a transaction amount random number with respect to the receiver, the transaction amount, and the public key of the receiver,

the generating unit:

generating a difference between the transaction amount random number with respect to the remitter and the transaction amount random number with respect to the receiver; and

adding the difference between the transaction amount random number with respect to the

remitter and the transaction amount random number with respect to the receiver in the transaction, for one or more blockchain nodes in the blockchain to verify that the transaction amount encrypted in the commitment ciphertext of the transaction amount with respect to the remitter is equal to the transaction amount encrypted in the commitment ciphertext of the transaction amount with respect to the receiver.

9. The device of claim 7, wherein the generating unit:

generating a first verification random number and a second verification random number;

applying the homomorphic encryption algorithm to generate a random-number commitment ciphertext based on the first verification random number, the second verification random number, and the public key of the receiver;

obtaining a hash digest by conducting a hash operation on the commitment ciphertext of the transaction amount with respect to the receiver and the random-number commitment ciphertext; and

generating a first verification element associated with the first verification random number and a second verification element associated with the second verification random number according to the hash digest,

wherein the public key proof associated with the receiver comprises the random-number commitment ciphertext, the first verification element, and the second verification element.

10. The device of claim 9, wherein the commitment ciphertext of the remitter's balance is obtained by applying the homomorphic encryption algorithm based on the remitter's balance, a public key of the remitter, and a remitter random number;

the generating unit:

generating a range proof based on the remitter random number, the transaction amount random number with respect to the remitter, the remitter's balance, the commitment ciphertext of the remitter's balance, the transaction amount, and the commitment ciphertext of the transaction amount with respect to the remitter; and

adding the range proof to the transaction for the one or more blockchain nodes in the blockchain to verify whether the transaction amount satisfies: the transaction amount being no less than 0 and no greater than the remitter's balance.

11. The device of any one of claims 7-10, wherein the generating unit:

generating, based on a private key of the remitter, a digital signature of the remitter related to the commitment ciphertext of the transaction amount with respect to the remitter and the commitment ciphertext of the transaction amount with respect to the receiver; and

adding the digital signature to the transaction for one or more blockchain nodes in the blockchain to verify the digital signature.

12. The device of claim 8, further comprising:

an off-chain sending unit sending the transaction amount random number with respect to the receiver to the receiver via an off-chain channel.

13. A computer device, comprising: a storage device and a processor, the storage device storing computer programs executable by the processor which, when executing the computer programs, performs the method of any of claims 1 to 6.