

BLOCKCHAIN-BASED TRANSACTION METHOD AND APPARATUS, AND REMITTER DEVICE

TECHNICAL FIELD

5 [0001] Implementations of the present specification relate to the field of computer technologies, and particularly, to a blockchain-based transaction method and apparatus, and a remitter device.

BACKGROUND

10 [0002] The blockchain technology is a distributed database technology that protects data from being tampered with and forged by using cryptography and consensus mechanisms. With the development of computer and Internet technologies, thanks to advantages such as decentralization, openness and transparency, tamper-resistance, and trustiness, the blockchain technology is highly favored and is widely applied to many fields such as smart contracts, securities transactions, E-commerce, the Internet of Things, social communication, file storage, proof of existence, identity verification, and equity crowdfunding.

15 [0003] Currently, when the blockchain technology is applied to transaction scenarios, because transaction information needs to be sent to the blockchain for verification, implementation, and chaining-on, the transaction information is at risk of being leaked to a third party unrelated to the transaction.

20 [0004] There is a need of a technical solution for realizing privacy protection in a transaction process.

SUMMARY

25 [0005] The objective of implementations of the present specification is to provide a blockchain-based transaction method and apparatus, and a remitter device, to achieve privacy protection without interaction with a remittee device.

[0006] To achieve the above objective, an implementation of the present specification provides a blockchain-based transaction method, including: calculating a transaction amount commitment based on a commitment random number and a

transaction amount; obtaining a first commitment random number ciphertext and a first transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitter account using a homomorphic encryption algorithm; obtaining a second
5 commitment random number ciphertext and a second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remittee account using the homomorphic encryption algorithm; and submitting transaction data to the blockchain, the transaction data including the transaction amount commitment, the first
10 commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment
15 random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account.

[0007] To achieve the above objective, an implementation of the present specification provides a blockchain-based transaction apparatus, including: a calculation unit, configured to calculate a transaction amount commitment based on a
20 commitment random number and a transaction amount; a first encryption unit, configured to obtain a first commitment random number ciphertext and a first transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitter account using a homomorphic encryption algorithm; a second encryption unit,
25 configured to obtain a second commitment random number ciphertext and a second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remittee account using the homomorphic encryption algorithm; and a submission unit, configured to submit transaction data to the blockchain, the transaction data including
30 the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount

commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account.

[0008] To achieve the above objective, an implementation of the present specification further provides a remitter device, including: a memory, configured to store computer instructions; and a processor, configured to execute the computer instructions to implement the method described in any of the implementations of the present specification.

[0009] It can be understood from the technical solutions provided in the implementations of the present specification, the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext can be obtained by using the commitment mechanism and the homomorphic encryption mechanism. The transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext can be submitted to the blockchain, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account. On the one hand, privacy protection can be realized by using a commitment mechanism and a homomorphic encryption mechanism. On the other hand, the transaction can be smoothly completed without the remittee device by submitting the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext to the blockchain.

BRIEF DESCRIPTION OF DRAWINGS

[0010] To describe the technical solutions in the implementations of the present specification or in the existing technologies more clearly, the following is a brief description of the accompanying drawings for illustrating such technical solutions. Clearly, the accompanying drawings described below are merely some implementations of the present specification, and a person skilled in the art can derive

other drawings from such accompanying drawings without making innovative efforts.

[0011] FIG. 1 is a schematic diagram illustrating blockchain-based transaction implementation according to an implementation of the present specification;

5 [0012] FIG. 2 is a flowchart illustrating a blockchain-based transaction method according to an implementation of the present specification;

[0013] FIG. 3 is a flowchart illustrating blockchain-based transaction implementation according to an implementation of the present specification;

[0014] FIG. 4 is a function structural diagram illustrating a blockchain-based transaction apparatus according to an implementation of the present specification; and

10 [0015] FIG. 5 is a function structural diagram illustrating a remitter device according to an implementation of the present specification.

DESCRIPTION OF IMPLEMENTATIONS

[0016] The following clearly and comprehensively describes the technical solutions in the implementations of the present specification with reference to the accompanying
15 drawings in the implementations of the present specification. Clearly, the described implementations are merely some rather than all of the implementations of the present specification. All other implementations obtained by a person of ordinary skill in the art based on the implementations of the present specification without creative efforts shall fall within the protection scope of the present specification.

20 [0017] Referring to FIG. 1 and FIG. 2, an implementation of the present specification provides a blockchain-based transaction method.

[0018] In this implementation, the blockchain can be a distributed ledger that organizes multiple pieces of block data in a chain structure based on a time sequence, and ensures security, traceability, and tamper-resistance by using a cryptography
25 algorithm. The blockchain can include a public blockchain, a consortium blockchain, a private blockchain, etc. The blockchain can be implemented based on a blockchain network. The blockchain network can include a peer-to-peer network (P2P network), etc. The blockchain network can include a plurality of blockchain nodes. Unified blockchain ledgers are maintained among blockchain nodes together.

30 [0019] In the implementations, the transaction method can be executed by a remitter device. The remitter device can join the blockchain network as a blockchain node. A remitter account can be logged into on the remitter device. The remitter account can be

an account of a remitter in the blockchain. The remitter account can have a signature public-private key pair, and an encryption public-private key pair. The signature public-private key pair can include a signature public key and a signature private key that are associated. The signature private key can be used to sign transaction data to be submitted to the blockchain. The signature public key can be disclosed to other blockchain nodes in the blockchain network for the other blockchain nodes to verify the signed transaction data. The encryption public-private key pair can include a homomorphic encryption public key and a homomorphic encryption private key that are associated. The homomorphic encryption public key can be disclosed to other blockchain nodes in the blockchain network, so the other blockchain nodes can encrypt data (e.g., a remitter balance, a transaction amount, a remitter random number, and a commitment random number) by using a homomorphic encryption algorithm. The homomorphic encryption private key can be used to decrypt an encrypted ciphertext. The remitter account can also register a remitter balance commitment, a remitter random number ciphertext, and a remitter balance ciphertext in the blockchain.

[0020] The remitter balance commitment can be calculated by the remitter device based on the remitter random number and the remitter balance. The remitter random number can be a random number corresponding to the remitter account and used for constructing the remitter balance commitment. The remitter balance can be a balance of any type of resource. The resource can include, for example, securities, coupons, property, virtual currency, and funds. Specifically, the remitter balance commitment can be implemented based on any type of commitment scheme with homomorphism, such as a Pedersen commitment mechanism. Taking the Pedersen commitment mechanism as an example, the remitter balance commitment can be calculated based on equation $PC(r_A, s_A) = g^{r_A} h^{s_A}$, where, $PC(r_A, s_A)$ is the remitter balance commitment; r_A is the remitter random number; s_A is the remitter balance; g and h are pre-determined parameters, respectively. Certainly, the remitter balance commitment can also be implemented by using a homomorphic encryption algorithm. That is, a ciphertext obtained by encrypting the remitter balance based on the homomorphic encryption algorithm is used as the remitter balance commitment. The homomorphic encryption algorithm will be described in detail subsequently. As such, on the one hand, direct registration of the remitter balance can be avoided by registering the remitter balance commitment in the blockchain, so the

remitter balance can be hidden and kept secret. On the other hand, because the commitment mechanism can support a high-efficiency zero-knowledge proof, the remitter balance commitment can make a zero-knowledge proof (e.g., a third zero-knowledge proof in the subsequent process) process involving the remitter balance in the subsequent process more efficient.

[0021] The remitter random number ciphertext can be obtained by the remitter device by encrypting the remitter random number based on the homomorphic encryption algorithm. The homomorphic encryption algorithm can include, for example, the Paillier algorithm, the Okamoto-Uchiyama algorithm, and the Damgard-Jurik algorithm. In particular, the Paillier algorithm and the Damgard-Jurik algorithm are more efficient and the encrypted ciphertext is shorter than the Okamoto-Uchiyama algorithm. Taking the Paillier algorithm as an example, the remitter random number ciphertext can be calculated based on equation $PA_A(r_A) = u1^{r_A} d1^{n1}$, where $PA_A(r_A)$ is the remitter random number ciphertext; r_A is the remitter random number; each of $u1$ and $n1$ is a portion of a homomorphic encryption public key of the remitter account; $d1$ is a random number used in the homomorphic encryption algorithm. On the one hand, the burden caused by locally keeping the remitter random number off the chain can be avoided by registering the remitter random number in the blockchain. On the other hand, direct registration of the remitter random number can be avoided by registering the remitter random number ciphertext in the blockchain, so the remitter random number can be hidden and kept secret.

[0022] The remitter balance ciphertext can be obtained by the remitter device by encrypting the remitter balance based on the homomorphic encryption algorithm. Taking the Paillier algorithm as an example, the remitter balance ciphertext can be calculated based on equation $PA_A(s_A) = u1^{s_A} d2^{n1}$, where $PA_A(s_A)$ is the remitter balance ciphertext; s_A is the remitter balance; each of $u1$ and $n1$ is a portion of the homomorphic encryption public key of the remitter account; $d2$ is a random number used in the homomorphic encryption algorithm. As such, direct registration of the remitter balance can be avoided by registering the remitter balance ciphertext in the blockchain, so the remitter balance can be hidden and kept secret.

[0023] In the implementations, the transaction method can transfer the transaction amount from the remitter account to a remittee account. The remittee account can be an account of a remittee in the blockchain. The remittee account can be logged into on a

remittee device. The remittee device can join the blockchain network as a blockchain node. The transaction amount can be negotiated between the remitter and the remittee. Similar to the remitter account, the remittee account can have a signature public-private key pair, and an encryption public-private key pair. The remittee account can also register a remittee balance commitment, a remittee random number ciphertext, and a remittee balance ciphertext in the blockchain. The remittee balance commitment can be calculated by the remittee device based on the remittee random number and the remittee balance. The remittee random number ciphertext can be obtained by the remittee device by encrypting the remittee random number based on the homomorphic encryption algorithm. The remittee balance ciphertext can be obtained by the remittee device by encrypting the remittee balance based on the homomorphic encryption algorithm.

[0024] The transaction method can include the following steps. It should be noted that although the present specification provides the method operation steps in the implementations or the flowcharts, more or fewer operation steps can be included based on conventional or non-creative efforts. In addition, a sequence of the steps listed in the implementation is merely one of numerous execution sequences of the steps, and does not represent a unique execution sequence. In actual execution of an apparatus or a product, execution can be performed based on the method sequence shown in the implementation or the accompanying drawing, or performed in parallel (*e.g.*, a parallel processor or a multi-thread processing environment).

[0025] Step S10: Calculate a transaction amount commitment based on a commitment random number and a transaction amount.

[0026] In the implementations, the commitment random number can be generated by the remitter device. The transaction amount can be negotiated between the remitter and the remittee. The transaction amount commitment can be implemented based on any type of commitment scheme with homomorphism, such as a Pedersen commitment mechanism. Taking the Pedersen commitment mechanism as an example, the transaction amount commitment can be calculated based on equation $PC(r, t) = g^r h^t$, where, $PC(r, t)$ is the transaction amount commitment; r is the commitment random number; t is the transaction amount; g and h are pre-determined parameters, respectively. Certainly, the transaction amount commitment can also be implemented by using the homomorphic encryption algorithm. That is, a ciphertext obtained by encrypting the transaction amount based on the homomorphic encryption algorithm is

used as the transaction amount commitment. As such, the transaction amount can be hidden and kept secret by using the transaction amount commitment.

[0027] Step S12: Obtain a first commitment random number ciphertext and a first transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitter account using a homomorphic encryption algorithm.

[0028] In the implementations, the remitter device can obtain the first commitment random number ciphertext by encrypting the commitment random number based on the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm. Specifically, the remitter device can generate a random number for homomorphic encryption as a first encryption random number; and obtain the first commitment random number ciphertext by encrypting the commitment random number based on the homomorphic encryption public key of the remitter account and the first encryption random number using the homomorphic encryption algorithm. Taking the Paillier algorithm as an example, the first commitment random number ciphertext can be calculated based on equation $PA_A(r) = u1^r \cdot y1^{n1}$, where $PA_A(r)$ is the first commitment random number ciphertext; r is the commitment random number; each of $u1$ and $n1$ is a portion of the homomorphic cryptographic public key of the remitter account, respectively; $y1$ is the first encryption random number.

[0029] In the implementations, the remitter device can obtain the first transaction amount ciphertext by encrypting the transaction amount based on the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm. Specifically, the remitter device can generate a random number for homomorphic encryption as a second encryption random number; and obtain the first transaction amount ciphertext by encrypting the transaction amount based on the homomorphic encryption public key of the remitter account and the second encryption random number using the homomorphic encryption algorithm. Taking the Paillier algorithm as an example, the first transaction amount ciphertext can be calculated based on equation $PA_A(t) = u1^t \cdot y2^{n1}$, where $PA_A(t)$ is the first transaction amount ciphertext; t is the transaction amount; each of $u1$ and $n1$ is a portion of the homomorphic cryptographic public key of the remitter account, respectively; $y2$ is the second encryption random number.

[0030] Step S14: Obtain a second commitment random number ciphertext and a

second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remittee account using the homomorphic encryption algorithm.

5 [0031] In the implementations, the remitter device can obtain the second commitment random number ciphertext by encrypting the commitment random number based on the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm. Specifically, the remitter device can generate a random number for homomorphic encryption as a third encryption random number; and obtain the second commitment random number ciphertext by encrypting the
10 commitment random number based on the homomorphic encryption public key of the remittee account and the third encryption random number using the homomorphic encryption algorithm. Taking the Paillier algorithm as an example, the second commitment random number ciphertext can be calculated based on equation $PA_B(r) = u^r z_1^{n^2}$, where $PA_B(r)$ is the second commitment random number
15 ciphertext; r is the commitment random number; each of u^2 and n^2 is a portion of the homomorphic cryptographic public key of the remittee account, respectively; z_1 is the third encryption random number.

[0032] In the implementations, the remitter device can obtain the second transaction amount ciphertext by encrypting the transaction amount based on the homomorphic
20 encryption public key of the remittee account using the homomorphic encryption algorithm. Specifically, the remitter device can generate a random number for homomorphic encryption as a fourth encryption random number; and obtain the second transaction amount ciphertext by encrypting the transaction amount based on the homomorphic encryption public key of the remittee account and the fourth encryption
25 random number using the homomorphic encryption algorithm. Taking the Paillier algorithm as an example, the second transaction amount ciphertext can be calculated based on equation $PA_B(t) = u^t z_2^{n^2}$, where $PA_B(t)$ is the second transaction amount ciphertext; t is the transaction amount; each of u^2 and n^2 is a portion of the homomorphic cryptographic public key of the remittee account, respectively; z_2
30 is the fourth encryption random number.

[0033] Step S16: Submit transaction data to the blockchain.

[0034] In the implementations, the transaction data can include the transaction amount commitment, the first commitment random number ciphertext, the first

transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext. The remitter device can submit the transaction data to the blockchain, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account.

[0035] Specifically, after the transaction data is submitted to the blockchain, a consensus blockchain node in the blockchain can update the remitter balance commitment based on the transaction amount commitment; update the remitter random number ciphertext based on the first commitment random number ciphertext; update the remitter balance ciphertext based on the first transaction amount ciphertext; update the remittee balance commitment based on the transaction amount commitment; update the remittee random number ciphertext based on the second commitment random number ciphertext; and update the remittee balance ciphertext based on the second transaction amount ciphertext. The consensus blockchain node can be a blockchain node determined based on a consensus mechanism of the blockchain. As such, the transaction data can be recorded into a blockchain ledger maintained by each blockchain node in the blockchain network, so as to avoid tampering.

[0036] An updated remitter balance commitment can be a quotient of the remitter balance commitment and the transaction amount commitment. Because the remitter balance commitment and the transaction amount commitment are implemented by using a homomorphic commitment mechanism, the transaction amount can be deducted from the remitter balance and the commitment random number can be deducted from the remitter random number. For example, the remitter balance commitment can be

$$PC(r_{-A}, s_{-A}) = g^{r_{-A}} h^{s_{-A}} ; \text{ the transaction amount commitment can be } PC(r, t) = g^r h^t ; \text{ the updated remitter balance commitment can be } PC(r_{-A} - r, s_{-A} - t) = PC(r_{-A}, s_{-A}) / PC(r, t) = g^{(r_{-A} - r)} h^{(s_{-A} - t)} .$$

[0037] An updated remitter random number ciphertext can be a quotient of the remitter random number ciphertext and the first commitment random number ciphertext. Because the remitter random number ciphertext and the first commitment random number ciphertext are calculated by using the homomorphic encryption algorithm, the commitment random number can be deducted from the remitter random number. For

example, the remitter random number ciphertext can be $PA_A(r_A) = u1^{r-A}d1^{n1}$; the first commitment random number ciphertext can be $PA_A(r) = u1^r y1^{n1}$; the updated remitter random number ciphertext can be $PA_A(r_A - r) = PA_A(r_A) / PA_A(r) = u1^{r-A-r}(d1 / y1)^{n1}$.

5 **[0038]** An updated remitter balance ciphertext can be a quotient of the remitter balance ciphertext and the first transaction amount ciphertext. Because the remitter balance ciphertext and the first transaction amount ciphertext are calculated by using the homomorphic encryption algorithm, the transaction amount can be deducted from the remitter balance. For example, the remitter balance ciphertext can be
 10 $PA_A(s_A) = u1^{s-A}d2^{n1}$; the first transaction amount ciphertext can be $PA_A(t) = u1^t y2^{n1}$; the updated remitter balance ciphertext can be $PA_A(s_A - t) = PA_A(s_A) / PA_A(t) = u1^{s-A-t}(d2 / y2)^{n1}$.

[0039] An updated remitee balance commitment can be a product of the remitee balance commitment and the transaction amount commitment. Because the remitee balance commitment and the transaction amount commitment are implemented by
 15 using the homomorphic commitment mechanism, the transaction amount can be added to the remitee balance and the commitment random number can be added to the remitee random number. For example, the remitee balance commitment can be $PC(r_B, s_B) = g^{r-B}h^{s-B}$; the transaction amount commitment can be
 20 $PC(r, t) = g^r h^t$; the updated remitee balance commitment can be $PC(r_B + r, s_B + t) = PC(r_B, s_B)PC(r, t) = g^{(r-B+r)}h^{(s-B+t)}$. r_B is the remitee random number; s_B is the remitee balance; g and h are pre-determined parameters, respectively.

[0040] An updated remitee random number ciphertext can be a product of the
 25 remitee random number ciphertext and the second commitment random number ciphertext. Because the remitee random number ciphertext and the second commitment random number ciphertext are calculated by using the homomorphic encryption algorithm, the commitment random number can be added to the remitee random number. For example, the remitee random number ciphertext can be
 30 $PA_B(r_B) = u2^{r-B}f1^{n2}$; the second commitment random number ciphertext can be $PA_B(r) = u2^r z1^{n2}$; the updated remitee random number ciphertext can be

$PA_B(r_B + r) = PA_B(r_B)PA_B(r) = u2^{r-B+r}(f1z1)^{n^2}$. r_B is the remittee random number; $f1$ is a random number used by the homomorphic encryption algorithm.

[0041] An updated remittee balance ciphertext can be a product of the remittee balance ciphertext and the second transaction amount ciphertext. Because the remittee balance ciphertext and the second transaction amount ciphertext are calculated by using the homomorphic encryption algorithm, the transaction amount can be added to the remittee balance. For example, the remittee balance ciphertext can be $PA_B(s_B) = u2^{s-B}f2^{n^2}$; the second transaction amount ciphertext can be $PA_B(t) = u2^t z2^{n^2}$; the updated remittee balance ciphertext can be $PA_B(s_B + t) = PA_B(s_B)PA_B(t) = u2^{s-B+t}(f2z2)^{n^2}$. s_B is the remittee balance; $f2$ is a random number used by the homomorphic encryption algorithm.

[0042] In an implementation, the remitter device can further generate a first zero-knowledge proof based on a zero-knowledge proof technique; and add the first zero-knowledge proof to the transaction data for the consensus blockchain node to verify that the transaction amount is not less than 0. The zero-knowledge proof technique can be implemented, for example, based on the zkSNARK scheme. Further, the zero-knowledge proof technique can include a range proof technique. The remitter device can generate a first range proof based on the range proof technique; and add the first range proof to the transaction data for the consensus blockchain node to verify that the transaction amount is not less than 0. The range proof technique can be implemented, for example, based on the Bulletproofs scheme or the Borromean ring signature scheme.

[0043] In an implementation, the remitter device can further generate a second zero-knowledge proof based on the zero-knowledge proof technique; and add the second zero-knowledge proof to the transaction data for the consensus blockchain node to verify that the transaction amount is not greater than the remitter balance. Further, the zero-knowledge proof technique can include the range proof technique. The remitter device can generate a second range proof based on the range proof technique; and add the second range proof to the transaction data for the consensus blockchain node to verify that the transaction amount is not greater than the remitter balance.

[0044] In an implementation, the remitter device can further generate a third zero-knowledge proof based on the zero-knowledge proof technique; and add the third zero-

knowledge proof to the transaction data for the consensus blockchain node to verify that: the commitment random number for calculating the transaction amount commitment is consistent with the commitment random number for calculating the second commitment random number ciphertext, and verify that: the transaction amount for calculating the transaction amount commitment is consistent with the transaction amount for calculating the second transaction amount ciphertext, so as to avoid recording inconsistent commitment random numbers and inconsistent transaction amounts into the remitter account and the remittee account.

[0045] For example, the third zero-knowledge proof can be (C, D, E, a, b, c, d).

$C = g^{r^*} h^{t^*}$; $D = u 2^{r^*} z1^{t^* n^2}$; $E = u 2^{t^*} z2^{r^* n^2}$; $x = Hash(C, D, E)$; $a = r^* + xr$; $b = t^* + xt$; $c = z1^{r^*} z1^{t^*}$; $d = z2^{r^*} z2^{t^*}$. r^* , t^* , $z1^*$, and $z2^*$ are random numbers generated by the remitter device and respectively correspond to the commitment random number r , the transaction amount t , the third encryption random number $z1$, and the fourth encryption random number $z2$; Hash represents a hash operation.

[0046] When verifying that all $g^a h^b == C T^x$, $u 2^a c^{n^2} == D M^x$, and $u 2^b d^{n^2} == E N^x$ are valid, the consensus blockchain node considers that: the commitment random number for calculating the transaction amount commitment is consistent with the commitment random number for calculating the second commitment random number ciphertext, and that: the transaction amount for calculating the transaction amount commitment is consistent with the transaction amount for calculating the second transaction amount ciphertext. T is the transaction amount commitment $PC(r, t)$; M is the second commitment random number ciphertext $PA_B(r)$; N is the second transaction amount ciphertext $PA_B(t)$.

[0047] In an implementation, the remitter device can calculate the first transaction amount commitment based on the first commitment random number and the transaction amount; and calculate the second transaction amount commitment based on the second commitment random number and the transaction amount. The first commitment random number and the second commitment random number can be generated by the remitter device. The first transaction amount commitment can correspond to a transaction amount to be transferred out of the remitter account. The second transaction amount commitment can correspond to a transaction amount to be transferred to the remittee account. As such, the transaction amount to be transferred out of the remitter account

and the transaction amount to be transferred to the remittee account can be distinguished.

[0048] The remitter device can separately obtain the first commitment random number ciphertext and the first transaction amount ciphertext by encrypting the first commitment random number and the transaction amount based on the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm; separately obtain the second commitment random number ciphertext and the second transaction amount ciphertext by encrypting the second commitment random number and the transaction amount based on the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm; and submit the transaction data to the blockchain. The transaction data can include the first transaction amount commitment, the second transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext, for the first transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the second transaction amount commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account.

[0049] Specifically, after the transaction data is submitted to the blockchain, the consensus blockchain node can update the remitter balance commitment based on the first transaction amount commitment; update the remitter random number ciphertext based on the first commitment random number ciphertext; update the remitter balance ciphertext based on the first transaction amount ciphertext; update the remittee balance commitment based on the second transaction amount commitment; update the remittee random number ciphertext based on the second commitment random number ciphertext; and update the remittee balance ciphertext based on the second transaction amount ciphertext.

[0050] An updated remitter balance commitment can be a quotient of the remitter balance commitment and the first transaction amount commitment; an updated remitter random number ciphertext can be a quotient of the remitter random number ciphertext and the first commitment random number ciphertext; an updated remitter balance ciphertext can be a quotient of the remitter balance ciphertext and the first transaction amount ciphertext; an updated remittee balance commitment can be a product of the remittee balance commitment and the second transaction amount commitment; an

updated remittee random number ciphertext can be a product of the remittee random number ciphertext and the second commitment random number ciphertext; an updated remittee balance ciphertext can be a product of the remittee balance ciphertext and the second transaction amount ciphertext.

5 **[0051]** Further, the remitter device can further generate a fourth zero-knowledge proof based on the zero-knowledge proof technique; and add the fourth zero-knowledge proof to the transaction data for the consensus blockchain node to verify that the transaction amount for calculating the first transaction amount commitment is consistent with the transaction amount for calculating the second transaction amount
10 commitment, so as to prevent the transaction amount transferred out of the remitter account from being inconsistent with the transaction amount transferred to the remittee account.

[0052] In an implementation, the remitter device can further obtain signature data by signing the transaction data using a signature private key of the remitter account; and
15 add the signature data to the transaction data for the consensus blockchain node to perform signature verification.

[0053] In the implementations, the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount
20 ciphertext can be obtained by using the commitment mechanism and the homomorphic encryption mechanism. The transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext can be submitted to the blockchain, for the transaction amount commitment, the first
25 commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account. On the one hand, privacy protection can be realized for transaction information such as the transaction amount by using the
30 commitment mechanism and the homomorphic encryption mechanism. On the other hand, the transaction can be smoothly completed in the blockchain without the remittee device by submitting the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext to the

blockchain. Therefore, the implementation can realize privacy protection under a non-interactive condition.

[0054] Further, the remitter account can register the remitter balance commitment, the remitter random number ciphertext, and the remitter balance ciphertext in the blockchain. The remittee account can register the remittee balance commitment, the remittee random number ciphertext, and the remittee balance ciphertext in the blockchain. On the one hand, privacy protection can be realized for both the account balance and the transaction amount. On the other hand, by registering the remitter random number ciphertext and the remittee random number ciphertext on the blockchain, the burden caused by locally keeping the remitter random number and the remittee random number off the chain can be avoided, and the risk of loss can be avoided.

[0055] Refer to FIG. 1, FIG. 2 and FIG. 3. For ease of understanding, an example scenario of an implementation of the present specification is described below.

[0056] In this implementation, assume that the remitter account can be Account_A. The remitter account Account_A can have a signature public-private key pair and an encryption public-private key pair. The remitter account Account_A can also register a remitter balance commitment $PC(r_A, s_A) = g^{r_A} h^{s_A}$, a remitter random number ciphertext $PA_A(r_A) = u1^{r_A} d1^{n1}$, and a remitter balance ciphertext $PA_A(s_A) = u1^{s_A} d2^{n1}$ in the blockchain. r_A is the remitter random number; s_A is the remitter balance; g and h are pre-determined parameters; each of $u1$ and $n1$ is a portion of the homomorphic encryption public key of the remitter account; $d1$ and $d2$ are random numbers used in the homomorphic encryption algorithm.

[0057] Assume that the remittee account can be Account_B. The remittee account Account_B can have a signature public-private key pair and an encryption public-private key pair. The remittee account Account_B can also register a remittee balance commitment $PC(r_B, s_B) = g^{r_B} h^{s_B}$, a remittee random number ciphertext $PA_B(r_B) = u2^{r_B} f1^{n2}$, and a remittee balance ciphertext $PA_B(s_B) = u2^{s_B} f2^{n2}$ in the blockchain. r_B is the remittee random number; s_B is the remittee balance; g and h are pre-determined parameters; each of $u2$ and $n2$ is a portion of the homomorphic encryption public key of the remittee account; $f1$ and $f2$ are random numbers used in the homomorphic encryption

algorithm.

[0058] In the implementations, the example transaction of the scenario can implement the transfer of transaction amount t from the remitter account Account_A to the remittee account Account_B. Specifically, the remitter device can generate a commitment random number r , a first encryption random number $y1$, a second encryption random number $y2$, a third encryption random number $z1$, and a fourth encryption random number $z2$. The remitter device can calculate the transaction amount commitment $PC(r, t) = g^r h^t$ based on the commitment random number r and transaction amount t ; obtain the first commitment random number ciphertext $PA_A(r) = u1^r y1^{n1}$ by encrypting the commitment random number r based on the first encryption random number $y1$ and the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm; obtain the first transaction amount ciphertext $PA_A(t) = u1^t y2^{n1}$ by encrypting transaction amount t based on the second encryption random number $y2$ and the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm; obtain the second commitment random number ciphertext $PA_B(r) = u2^r z1^{n2}$ by encrypting the commitment random number r based on the third encryption random number $z1$ and the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm; obtain the second transaction amount ciphertext $PA_B(t) = u2^t z2^{n2}$ by encrypting transaction amount t based on the fourth encryption random number $z2$ and the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm.

[0059] In this implementation, the remitter device can generate a first zero-knowledge proof $RP1$, a second zero-knowledge proof $RP2$, and a third zero-knowledge proof $RP3$. The first zero-knowledge proof $RP1$ can be used to verify: $t \geq 0$. The second zero-knowledge proof $RP2$ can be used to verify: $s_A - t \geq 0$. The third zero-knowledge proof $RP3$ can be used to verify that the commitment random number r in $PC(r, t)$ is consistent with the commitment random number r in $PA_B(r)$, and verify that transaction amount t in $PC(r, t)$ is consistent with transaction amount t in $PA_B(t)$.

[0060] In this implementation, the remitter device can obtain signature data SIG by

signing $[PC(r, t), PA_A(r), PA_A(t), PA_B(r), PA_B(t), RP1, RP2, RP3]$ using a signature private key of the remitter account; and submit transaction data $[PC(r, t), PA_A(r), PA_A(t), PA_B(r), PA_B(t), RP1, RP2, RP3, SIG]$ to the blockchain.

5 **[0061]** In this implementation, the blockchain can determine a consensus blockchain node based on the consensus mechanism. By using the anti-double-spending or anti-replay mechanism of the related technologies, the consensus blockchain node can verify whether the transaction has been executed. If the transaction has been executed, the consensus blockchain node can reject the transaction.

10 **[0062]** If the transaction has not been executed, the consensus blockchain node can verify whether the signature data SIG in the transaction data is correct. If the signature data SIG is incorrect, the consensus blockchain node can reject the transaction.

[0063] If the signature data SIG is correct, the consensus blockchain node can check the first zero-knowledge proof $RP1$ in the transaction data to verify whether $t \geq 0$ is satisfied. If not satisfied, the consensus blockchain node can reject the transaction.

15 **[0064]** If satisfied, the consensus blockchain node can check the second zero-knowledge proof $RP2$ in the transaction data to verify whether $s_A - t \geq 0$ is satisfied. If not satisfied, the consensus blockchain node can reject the transaction.

[0065] If satisfied, the consensus blockchain node can check the third zero-knowledge proof $RP3$ in the transaction data to verify whether the commitment random number r in $PC(r, t)$ is consistent with the commitment random number r in $PA_B(r)$, and verify whether transaction amount t in $PC(r, t)$ is consistent with transaction amount t in $PA_B(t)$. If not satisfied, the consensus blockchain node can reject the transaction.

25 **[0066]** If satisfied, the consensus blockchain node can update the remitter balance commitment $PC(r_A, s_A)$, the remitter random number ciphertext $PA_A(r_A)$, the remitter balance ciphertext $PA_A(s_A)$, the remittee balance commitment $PC(r_B, s_B)$, the remittee random number ciphertext $PA_B(r_B)$, and the remittee balance ciphertext $PA_B(s_B)$. Specifically, an updated remitter balance commitment can be $PC(r_A - r, s_A - t) = PC(r_A, s_A) / PC(r, t) = g^{(r_A - r)} h^{(s_A - t)}$;
30 an updated remitter random number ciphertext can be

$PA_A(r_A - r) = PA_A(r_A) / PA_A(r) = u1^{r-A-r} (d1 / y1)^{n1}$; an updated remitter
 balance ciphertext can be $PA_A(s_A - t) = PA_A(s_A) / PA_A(t) = u1^{t-A-t} (d2 / y2)^{n1}$;
 an updated remittee balance commitment can be
 $PC(r_B + r, s_B + t) = PC(r_B, s_B)PC(r, t) = g^{(r-B+r)} h^{(s-B+t)}$; an updated
 5 remittee random number ciphertext can be
 $PA_B(r_B + r) = PA_B(r_B)PA_B(r) = u2^{r-B+r} (f1z1)^{n2}$; an updated remittee
 balance ciphertext can be $PA_B(s_B + t) = PA_B(s_B)PA_B(t) = u2^{s-B+t} (f2z2)^{n2}$.

[0067] Referring to FIG. 4, an implementation of the present specification further
 provides a blockchain-based transaction apparatus. The transaction apparatus can
 10 include the following units: a calculation unit 20, configured to calculate a transaction
 amount commitment based on a commitment random number and a transaction amount;
 a first encryption unit 22, configured to obtain a first commitment random number
 ciphertext and a first transaction amount ciphertext by encrypting the commitment
 random number and the transaction amount separately based on a homomorphic
 15 encryption public key of a remitter account using a homomorphic encryption algorithm;
 a second encryption unit 24, configured to obtain a second commitment random number
 ciphertext and a second transaction amount ciphertext by encrypting the commitment
 random number and the transaction amount separately based on a homomorphic
 encryption public key of a remittee account using the homomorphic encryption
 20 algorithm; and a submission unit 26, configured to submit transaction data to the
 blockchain, the transaction data including the transaction amount commitment, the first
 commitment random number ciphertext, the first transaction amount ciphertext, the
 second commitment random number ciphertext, and the second transaction amount
 ciphertext, for the transaction amount commitment, the first commitment random
 25 number ciphertext, and the first transaction amount ciphertext to be recorded into the
 remitter account, and the transaction amount commitment, the second commitment
 random number ciphertext, and the second transaction amount ciphertext to be recorded
 into the remittee account.

[0068] Referring to FIG. 5, an implementation of the present specification further
 30 provides a remitter device. The remitter device can include a memory and a processor.

[0069] In this implementation, the memory can be implemented in any suitable way.
 For example, the memory can be a read-only memory, a mechanical hard disk, a solid-
 state drive, a USB flash drive, etc. The memory can be configured to store computer

instructions.

[0070] In this implementation, the processor can be implemented in any suitable way. For example, the processor can be a microprocessor, a processor, or a computer readable medium storing computer readable program code (such as software or firmware) that can be executed by the microprocessor or the processor, a logic gate, a switch, an application-specific integrated circuit (ASIC), a programmable logic controller, or a built-in microcontroller. The processor can execute the computer instructions to perform the steps of: calculating a transaction amount commitment based on a commitment random number and a transaction amount; obtaining a first commitment random number ciphertext and a first transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitter account using a homomorphic encryption algorithm; obtaining a second commitment random number ciphertext and a second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitee account using the homomorphic encryption algorithm; and submitting transaction data to the blockchain, the transaction data including the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remitee account.

[0071] The implementations in the present specification are described in a progressive method. For the same or similar parts in the implementations, references can be made to each other. Each implementation focuses on a difference from other implementations. Especially, an apparatus implementation and a device implementation are basically similar to a method implementation, and therefore are described briefly. For related parts, refer to descriptions in the method implementation.

[0072] After reading the present specification, a person skilled in the art can think of any combination between some or all of the implementations enumerated in the present specification without creative efforts. These combinations are also within the scope disclosed and protected by the present specification.

[0073] In the 1990s, whether technology improvement is hardware improvement (for example, improvement of a circuit structure, such as a diode, a transistor, or a switch) or software improvement (improvement of a method procedure) can be clearly distinguished. However, as technologies develop, the current improvement for many method procedures can be considered as a direct improvement of a hardware circuit structure. A designer usually programs an improved method procedure to a hardware circuit, to obtain a corresponding hardware circuit structure. Therefore, a method procedure can be improved by using a hardware entity module. For example, a programmable logic device (PLD) (for example, a field programmable gate array (FPGA)) is such an integrated circuit, and a logical function of the programmable logic device is determined by a user through device programming. A designer performs programming to "integrate" a digital system to a single PLD, without requiring a chip manufacturer to design and manufacture a dedicated integrated circuit chip 2. In addition, at present, instead of manually manufacturing an integrated chip, this type of programming is mostly implemented by using "logic compiler" software. The programming is similar to a software compiler used to develop and write a program. Original code needs to be written in a particular programming language for compilation. The language is referred to as a hardware description language (HDL). There are many HDLs, such as the Advanced Boolean Expression Language (ABEL), the Altera Hardware Description Language (AHDL), Confluence, the Cornell University Programming Language (CUPL), HDcAl, the Java Hardware Description Language (JHDL), Lava, Lola, MyHDL, PALASM, and the Ruby Hardware Description Language (RHDL). The very-high-speed integrated circuit hardware description language (VHDL) and Verilog2 are most commonly used. A person skilled in the art should also understand that a hardware circuit that implements a logical method procedure can be readily obtained once the method procedure is logically programmed by using the several described hardware description languages and is programmed into an integrated circuit.

[0074] The system, apparatus, module, or unit illustrated in the implementations can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical implementation device is a computer. A specific form of the computer can be a personal computer, a laptop computer, a cellular phone, a camera phone, an intelligent phone, a personal digital assistant, a media player, a navigation device, an email transceiver device, a game console, a tablet computer, a

wearable device, or any combination thereof.

[0075] It can be understood from the descriptions of the implementations that a person skilled in the art can clearly understand that the present specification can be implemented by using software and a necessary general hardware platform. Based on
5 such an understanding, the technical solutions in the present specification essentially or the part contributing to the existing technology can be implemented in a form of a software product. The computer software product can be stored in a storage medium, such as a ROM/RAM, a magnetic disk, or an optical disc, and includes several instructions for instructing a computer device (which can be a personal computer, a
10 server, a network device, etc.) to execute the method described in the implementations of the present specification or in some parts of the implementations of the present specification.

[0076] The present specification can be applied to many general-purpose or dedicated computer system environments or configurations, for example, a personal
15 computer, a server computer, a handheld device or a portable device, a tablet device, a multi-processor system, a microprocessor-based system, a set-top box, a programmable consumption electronic device, a network PC, a minicomputer, a mainframe computer, and a distributed computing environment including any one of the previous systems or devices.

[0077] The present specification can be described in the general context of computer executable instructions executed by a computer, for example, a program module. Generally, the program module includes a routine, a program, an object, a component, a data structure, etc., executing a specific task or implementing a specific abstract data
20 type. The present specification can also be practiced in distributed computing environments. In the distributed computing environments, tasks are performed by remote processing devices connected through a communications network. In a distributed computing environment, the program module can be located in both local and remote computer storage media including storage devices.

[0078] Although the present specification is described by using the implementations,
30 a person of ordinary skill in the art knows that many variations of the present specification can be made without departing from the spirit of the present specification. It is expected that the appended claims include these variations without departing from the spirit of the present specification.

CLAIMS

What is claimed is:

1. A blockchain-based transaction method, comprising:

calculating a transaction amount commitment based on a commitment random number and a transaction amount involving a remitter and a remittee;

obtaining a first commitment random number ciphertext and a first transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitter account of the remitter using a homomorphic encryption algorithm;

obtaining a second commitment random number ciphertext and a second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remittee account of the remittee using the homomorphic encryption algorithm; and

submitting transaction data to the blockchain, the transaction data including the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account.

2. The method according to claim 1, wherein the remitter account registers a remitter balance commitment, a remitter random number ciphertext, and a remitter balance ciphertext in the blockchain; the remitter balance commitment is calculated based on a remitter random number and a remitter balance; the remitter random number ciphertext is calculated based on the remitter random number by using the homomorphic encryption algorithm; the remitter balance ciphertext is calculated based on the remitter balance by using the homomorphic encryption algorithm; and

wherein the remittee account registers a remittee balance commitment, a remittee random number ciphertext, and a remittee balance ciphertext in the blockchain; the

remittee balance commitment is calculated based on a remittee random number and a remittee balance; the remittee random number ciphertext is calculated based on the remittee random number by using the homomorphic encryption algorithm; the remittee balance ciphertext is calculated based on the remittee balance by using the homomorphic encryption algorithm.

3. The method according to claim 2, wherein the transaction amount commitment, the first transaction amount ciphertext, and the first commitment random number ciphertext are recorded into the remitter account by:

updating the remitter balance commitment based on the transaction amount commitment, updating the remitter random number ciphertext based on the first commitment random number ciphertext, and updating the remitter balance ciphertext based on the first transaction amount ciphertext; and

wherein the transaction amount commitment, the second transaction amount ciphertext, and the second commitment random number ciphertext are recorded into the remittee account by:

updating the remittee balance commitment based on the transaction amount commitment, updating the remittee random number ciphertext based on the second commitment random number ciphertext, and updating the remittee balance ciphertext based on the second transaction amount ciphertext.

4. The method according to claim 3, wherein an updated remitter balance commitment is a quotient of the remitter balance commitment and the transaction amount commitment; an updated remitter random number ciphertext is a quotient of the remitter random number ciphertext and the first commitment random number ciphertext; and an updated remitter balance ciphertext is a quotient of the remitter balance ciphertext and the first transaction amount ciphertext; and

wherein an updated remittee balance commitment is a product of the remittee balance commitment and the transaction amount commitment; an updated remittee random number ciphertext is a product of the remittee random number ciphertext and the second commitment random number ciphertext; and an updated remittee balance ciphertext is a product of the remittee balance ciphertext and the second transaction amount ciphertext.

5. The method according to claim 1, wherein the obtaining the first commitment random number ciphertext and the first transaction amount ciphertext includes:

obtaining the first commitment random number ciphertext by encrypting the commitment random number based on a first encryption random number and the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm; and

obtaining the first transaction amount ciphertext by encrypting the transaction amount based on a second encryption random number and the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm.

6. The method according to claim 1, wherein the obtaining the second commitment random number ciphertext and the second transaction amount ciphertext includes:

obtaining the second commitment random number ciphertext by encrypting the commitment random number based on a third encryption random number and the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm; and

obtaining the second transaction amount ciphertext by encrypting the transaction amount based on a fourth encryption random number and the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm.

7. The method according to claim 1, comprising:

before the submitting the transaction data to the blockchain,

generating a first zero-knowledge proof, the first zero-knowledge proof configured to verify that the transaction amount is not less than 0;

wherein the transaction data includes the first zero-knowledge proof.

8. The method according to claim 1, comprising:

before the submitting the transaction data to the blockchain,

generating a second zero-knowledge proof, the second zero-knowledge proof configured to verify that the transaction amount is not greater than the remitter balance;

wherein the transaction data includes the second zero-knowledge proof.

9. The method according to claim 1, comprising:

before the submitting the transaction data to the blockchain,

generating a third zero-knowledge proof, the third zero-knowledge proof configured to verify that the commitment random number for calculating the transaction amount commitment is consistent with the commitment random number for calculating the second commitment random number ciphertext, and verify that the transaction amount for calculating the transaction amount commitment is consistent with the transaction amount for calculating the second transaction amount ciphertext;

wherein the transaction data includes the third zero-knowledge proof.

10. The method according to claim 1, comprising:

before the submitting the transaction data to the blockchain,

obtaining signature data by signing the transaction data using a signature private key of the remitter account;

wherein the transaction data includes the signature data.

11. The method according to claim 1, wherein the calculating the transaction amount commitment based on the commitment random number and the transaction amount includes:

obtaining a first transaction amount commitment based on a first commitment random number and the transaction amount, and obtaining a second transaction amount commitment based on a second commitment random number and the transaction amount;

wherein the obtaining the first commitment random number ciphertext and the first transaction amount ciphertext by encrypting the transaction amount includes:

obtaining the first transaction amount ciphertext and the first commitment random number ciphertext by encrypting the transaction amount and the first commitment random number separately based on the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm;

wherein the obtaining the second commitment random number ciphertext and the second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on the homomorphic encryption public key of the remitee account using the homomorphic encryption algorithm includes:

obtaining the second commitment random number ciphertext and the second transaction amount ciphertext by encrypting the second commitment random number and the transaction amount separately based on the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm; and

wherein the transaction data includes the first transaction amount commitment, the second transaction amount commitment, the first transaction amount ciphertext, the first commitment random number ciphertext, the second transaction amount ciphertext, and the second commitment random number ciphertext, for the first transaction amount commitment, the first transaction amount ciphertext, and the first commitment random number ciphertext to be recorded into the remitter account, and the second transaction amount commitment, the second transaction amount ciphertext, and the second commitment random number ciphertext to be recorded into the remittee account.

12. The method according to claim 11, further comprising:

before the submitting the transaction data to the blockchain,

generating a fourth zero-knowledge proof, the fourth zero-knowledge proof configured to verify that the transaction amount for calculating the first transaction amount commitment is consistent with the transaction amount for calculating the second transaction amount commitment;

wherein the transaction data further includes the fourth zero-knowledge proof.

13. A blockchain-based transaction apparatus, comprising:

a calculation unit, configured to calculate a transaction amount commitment based on a commitment random number and a transaction amount involving a remitter and a remittee;

a first encryption unit, configured to obtain a first commitment random number ciphertext and a first transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitter account of the remitter using a homomorphic encryption algorithm;

a second encryption unit, configured to obtain a second commitment random number ciphertext and a second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a

homomorphic encryption public key of a remittee account of the remittee using the homomorphic encryption algorithm; and

a submission unit, configured to submit transaction data to the blockchain, the transaction data including the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment random number ciphertext, and the second transaction amount ciphertext to be recorded into the remittee account.

14. A system, comprising:

a memory, configured to store computer instructions; and

a processor, configured to execute the computer instructions to implement acts including:

calculating a transaction amount commitment based on a commitment random number and a transaction amount involving a remitter and a remittee;

obtaining a first commitment random number ciphertext and a first transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remitter account of the remitter using a homomorphic encryption algorithm;

obtaining a second commitment random number ciphertext and a second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on a homomorphic encryption public key of a remittee account of the remittee using the homomorphic encryption algorithm; and

submitting transaction data to the blockchain, the transaction data including the transaction amount commitment, the first commitment random number ciphertext, the first transaction amount ciphertext, the second commitment random number ciphertext, and the second transaction amount ciphertext, for the transaction amount commitment, the first commitment random number ciphertext, and the first transaction amount ciphertext to be recorded into the remitter account, and the transaction amount commitment, the second commitment random number ciphertext, and the second

transaction amount ciphertext to be recorded into the remittee account.

15. The system according to claim 14, wherein the remitter account registers a remitter balance commitment, a remitter random number ciphertext, and a remitter balance ciphertext in the blockchain; the remitter balance commitment is calculated based on a remitter random number and a remitter balance; the remitter random number ciphertext is calculated based on the remitter random number by using the homomorphic encryption algorithm; the remitter balance ciphertext is calculated based on the remitter balance by using the homomorphic encryption algorithm; and

wherein the remittee account registers a remittee balance commitment, a remittee random number ciphertext, and a remittee balance ciphertext in the blockchain; the remittee balance commitment is calculated based on a remittee random number and a remittee balance; the remittee random number ciphertext is calculated based on the remittee random number by using the homomorphic encryption algorithm; the remittee balance ciphertext is calculated based on the remittee balance by using the homomorphic encryption algorithm.

16. The system according to claim 15, wherein the transaction amount commitment, the first transaction amount ciphertext, and the first commitment random number ciphertext are recorded into the remitter account by:

updating the remitter balance commitment based on the transaction amount commitment, updating the remitter random number ciphertext based on the first commitment random number ciphertext, and updating the remitter balance ciphertext based on the first transaction amount ciphertext; and

wherein the transaction amount commitment, the second transaction amount ciphertext, and the second commitment random number ciphertext are recorded into the remittee account by:

updating the remittee balance commitment based on the transaction amount commitment, updating the remittee random number ciphertext based on the second commitment random number ciphertext, and updating the remittee balance ciphertext based on the second transaction amount ciphertext.

17. The system according to claim 16, wherein an updated remitter balance

commitment is a quotient of the remitter balance commitment and the transaction amount commitment; an updated remitter random number ciphertext is a quotient of the remitter random number ciphertext and the first commitment random number ciphertext; and an updated remitter balance ciphertext is a quotient of the remitter balance ciphertext and the first transaction amount ciphertext; and

wherein an updated remittee balance commitment is a product of the remittee balance commitment and the transaction amount commitment; an updated remittee random number ciphertext is a product of the remittee random number ciphertext and the second commitment random number ciphertext; and an updated remittee balance ciphertext is a product of the remittee balance ciphertext and the second transaction amount ciphertext.

18. The system according to claim 14, wherein the obtaining the first commitment random number ciphertext and the first transaction amount ciphertext includes:

obtaining the first commitment random number ciphertext by encrypting the commitment random number based on a first encryption random number and the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm; and

obtaining the first transaction amount ciphertext by encrypting the transaction amount based on a second encryption random number and the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm.

19. The system according to claim 14, wherein the obtaining the second commitment random number ciphertext and the second transaction amount ciphertext includes:

obtaining the second commitment random number ciphertext by encrypting the commitment random number based on a third encryption random number and the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm; and

obtaining the second transaction amount ciphertext by encrypting the transaction amount based on a fourth encryption random number and the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm.

20. The system according to claim 14, wherein the calculating the transaction amount commitment based on the commitment random number and the transaction amount includes:

obtaining a first transaction amount commitment based on a first commitment random number and the transaction amount, and obtaining a second transaction amount commitment based on a second commitment random number and the transaction amount;

wherein the obtaining the first commitment random number ciphertext and the first transaction amount ciphertext by encrypting the transaction amount includes:

obtaining the first transaction amount ciphertext and the first commitment random number ciphertext by encrypting the transaction amount and the first commitment random number separately based on the homomorphic encryption public key of the remitter account using the homomorphic encryption algorithm;

wherein the obtaining the second commitment random number ciphertext and the second transaction amount ciphertext by encrypting the commitment random number and the transaction amount separately based on the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm includes:

obtaining the second commitment random number ciphertext and the second transaction amount ciphertext by encrypting the second commitment random number and the transaction amount separately based on the homomorphic encryption public key of the remittee account using the homomorphic encryption algorithm; and

wherein the transaction data includes the first transaction amount commitment, the second transaction amount commitment, the first transaction amount ciphertext, the first commitment random number ciphertext, the second transaction amount ciphertext, and the second commitment random number ciphertext, for the first transaction amount commitment, the first transaction amount ciphertext, and the first commitment random number ciphertext to be recorded into the remitter account, and the second transaction amount commitment, the second transaction amount ciphertext, and the second commitment random number ciphertext to be recorded into the remittee account.