

CROSS-CHAIN DATA PROCESSING METHOD, APPARATUS, AND CLIENT DEVICE, AND BLOCKCHAIN SYSTEM

TECHNICAL FIELD

5 **[0001]** Solutions in embodiments of the present specification are related to the technical field of blockchain data processing, and in particular, to cross-chain data processing methods, apparatuses, and client devices, and blockchain systems.

BACKGROUND

10 **[0002]** The blockchain technology is also referred to as a distributed ledger technology. It is a decentralized distributed database technology that features decentralization, openness and transparency, tamper-resistance, and trustworthiness. Each piece of data of a blockchain is broadcast to all blockchain nodes of the entire network so that each node has full and consistent data. With the development and application of the blockchain technology, in addition to public blockchains that are
15 open to the entire network, private blockchains, consortium blockchain, etc. also emerge. Cross-chain data exchanges between blockchain systems have become an important factor that needs to be considered in technical or service solutions. In existing cross-chain technologies that are disclosed, relay chains and other technical solutions can be used to implement cross-chain data exchanges between blockchains.

20 **[0003]** For blockchain systems to exchange data with each other, a blockchain usually needs to be able to authenticate data on other blockchains to guarantee the security of data transmission between the blockchain systems and protect the blockchain data from being tampered with. Currently, multiple cross-chain network implementations are available, for example, Polkadot (a technology that implements
25 cross-chain communication). In Polkadot, a validator role is designed. The role checks whether data is from a blockchain ((by using data authentication logic of the blockchain). Validators use the proof of stake (POS) algorithm (a consensus algorithm) to reach a consensus on a result and validate the result. It can be confirmed that data that a consensus is reached on is from one blockchain, and the other blockchains can

use the cross-chain data. Currently, when a relay chain is used to authenticate cross-chain data, a consensus needs to be reached on the cross-chain data on the relay chain based on POS. A throughput of a consensus algorithm of the relay chain is subject to an upper limit of cross-chain data traffic, resulting in a throughput bottleneck. Moreover, when authenticating data on the relay chain, a blockchain needs to verify multiple signatures. For example, in the POS mechanism, a blockchain needs to verify hundreds of signatures, resulting in low execution efficiency.

[0004] Therefore, the industry is in dire need of cross-chain data authentication solutions that are more efficient and highly scalable.

SUMMARY

[0005] An objective of embodiments of the present specification is to provide cross-chain data processing methods, apparatuses, and client devices, and blockchain systems, to implement more efficient data authentication between blockchains, achieve better scalability and high efficiency, and extend applicability to more blockchain application scenarios.

[0006] The cross-chain data processing methods, apparatuses, and client devices, and blockchain systems provided in the embodiments of the present specification are implemented in the following ways:

[0007] A cross-chain data processing method is provided, where the method includes: sending, by a first blockchain, a cross-chain data request to a trusted attester; obtaining, by the trusted attester, cross-chain data from a second blockchain based on the cross-chain data request; verifying, by the trusted attester, the cross-chain data; and when the verification is passed, signing the cross-chain data by using a private key corresponding to a trusted execution environment (TEE) policy adopted by the trusted attester; returning, by the trusted attester, a signed request result to the first blockchain; and verifying, by the first blockchain, the request result by using a public key corresponding to the TEE policy; and when the verification is passed, determining that the cross-chain data in the request result is from the trusted attester and has been verified by the trusted attester.

[0008] A cross-chain data processing method is provided, where the method includes: receiving a cross-chain data request sent by a first blockchain; obtaining cross-chain data from a second blockchain based on the cross-chain data request;

verifying the cross-chain data; and when the verification is passed, signing the cross-chain data by using a private key corresponding to a TEE policy adopted; and returning a signed request result to the first blockchain, where the request result includes the cross-chain data and the signature.

- 5 **[0009]** A cross-chain data processing apparatus is provided, where the apparatus includes: a request receiving module, configured to receive a cross-chain data request sent by a first blockchain; a data acquisition module, configured to obtain cross-chain data from a second blockchain based on the cross-chain data request; a data verification module, configured to verify the cross-chain data; and when the
10 verification is passed, sign the cross-chain data by using a private key corresponding to a TEE policy adopted; and an attestation returning module, configured to return a signed request result to the first blockchain, where the request result includes the cross-chain data and the signature.

- [0010]** A cross-chain data processing client device is provided, including a
15 processor and a memory that is configured to store a processor-executable instruction, where when executing the instruction, the processor implements the steps of: receiving a cross-chain data request sent by a first blockchain; obtaining cross-chain data from a second blockchain based on the cross-chain data request; verifying the cross-chain data; and when the verification is passed, signing the cross-chain data by
20 using a private key corresponding to a TEE policy adopted; and returning a signed request result to the first blockchain, where the request result includes the cross-chain data and the signature.

[0011] A blockchain system is provided, where the system includes the apparatus or the client device according to any embodiments of the present specification.

- 25 **[0012]** A cross-chain system is provided, including at least two blockchains and a relay that corresponds to a target blockchain that returns cross-chain data, where a blockchain that requests for data obtains the cross-chain data from the target blockchain by using the corresponding relay, and the relay includes the apparatus according to any one of the embodiments of the present specification or any one of the
30 method implementations of the present specification.

[0013] According to the cross-chain data processing methods, apparatuses, and client devices, and the blockchain systems provided in the embodiments of the present specification, data exchanges between blockchains can be implemented based on a TEE, to rapidly and efficiently implement cross-chain data authentication, implement

high scalability of cross-chain data processing, and extend applicability to more cross-chain data exchange scenarios. With the solutions in the embodiments of the present specification, an upper-limit of cross-chain data throughput traffic can be increased efficiently with ease, effectively alleviating the problems that a conventional POS-based consensus relay has limited application scenarios and is relatively low in efficiency and poor in performance.

BRIEF DESCRIPTION OF DRAWINGS

- [0014] To describe the technical solutions in the embodiments of the present specification or in the existing technology more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments or the existing technology. Clearly, the accompanying drawings in the following description merely show some embodiments of the present specification, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.
- [0015] FIG. 1 is a schematic diagram illustrating a cross-chain data processing process and a system structure, according to the present specification;
- [0016] FIG. 2 is a schematic flowchart illustrating an embodiment of a cross-chain data processing method, according to the present specification;
- [0017] FIG. 3 is a schematic flowchart illustrating an embodiment of another cross-chain data processing method, according to the present specification;
- [0018] FIG. 4 is a block diagram illustrating a hardware structure of a client device that executes a cross-chain data processing method, according to an embodiment of the present specification;
- [0019] FIG. 5 is a schematic diagram illustrating a modular structure of an embodiment of a cross-chain data processing apparatus that can serve as a cross-chain relay, according to the present specification;
- [0020] FIG. 6 is a schematic diagram illustrating a modular structure of another embodiment of the apparatus, according to the present specification; and
- [0021] FIG. 7 is a schematic diagram illustrating another cross-chain data processing process and system structure, according to the present specification.

DESCRIPTION OF EMBODIMENTS

[0022] To help a person skilled in the art better understand the technical solutions of the present specification, the following clearly and comprehensively describes the technical solutions in the embodiments of the present specification with reference to the accompanying drawings in the embodiments of the present invention. Clearly, the described embodiments are merely some but not all of the embodiments of the present specification. All other embodiments obtained by a person of ordinary skill in the art based on one or more embodiments of the present specification without creative efforts shall fall within the protection scope of the present application.

[0023] Data in a blockchain system usually supports attestation by a third party. Usually, when interacting with a blockchain system, an application system can choose to obtain data from a blockchain node in a trusted domain because the application system can trust the data on the node. As a member of a blockchain network, the blockchain node can authenticate data on the network. A near-end client device can be deployed in the application system. The client device can be designed to implement a blockchain protocol and can authenticate data in the blockchain system. Therefore, the application system can interact with a blockchain by using the client device. In the embodiments of the present specification, data exchanges between blockchain systems can also be implemented by using a client device, same as the data exchanges between the application system and the blockchain system.

[0024] In the solutions provided in the embodiments of the present specification, a client device that implements cross-chain data authentication can be implemented by using core logic of a trusted execution environment (TEE) so that the client device itself can authenticate data on blockchains. The TEE is usually a secure environment provided in a hardware device to ensure that sensitive data is stored, processed, and protected in an isolated and trusted environment. The TEE can represent an environment that is isolated from other modules in the hardware device and that can execute trusted application programs (code). The TEE can include a CPU and other elements, such as a memory and an I/O device. The TEE can be a runtime environment that coexists with the Rich operating system (Rich OS, which is usually ANDROID etc.) on the device and provides security services for the Rich OS. The TEE has its own execution environment and has a higher security level than the Rich OS. Software and hardware resources that the TEE can access are separated from the

Rich OS. The TEE provides a secure execution environment for authorized software (which is a trusted application, for example, a blockchain verification application on the client device in the embodiments of the present specification), and also protects the confidentiality, integrity, and access permissions of resources and data. To
5 guarantee a root of trust of the TEE, the TEE needs to pass verification and be separated from the Rich OS in a secure boot process.

[0025] Hardware technical solutions supporting TEEs can include the AMD Platform Security Processor (PSP), the ARM TrustZone technology (applicable to all the ARM processors supporting TrustZone), the x86-64 instruction set, etc. In the
10 some embodiments provided in the present specification, Intel Software Guard Extensions (SGX) can be selected to implement a TEE. Intel SGX is a processor technology developed by Intel, a chip processor developer. SGX provides a trusted execution environment called an enclave. The enclave can prevent other application programs, including the OS, BIOS system, etc. from infiltration into and tampering
15 with a state of a protected application program and data in the trusted environment. Once software and data are located in an enclave, even the operating system cannot affect the code and data in the enclave. The security boundary of the enclave includes only a CPU and the enclave. SGX can further introduce an attestation-based authentication mechanism. Logic of a remote verification mechanism can be packaged
20 into a chip, and whether a program and a result of the program can be trusted is authenticated by Intel or an Intel agent in response to a request.

[0026] Specifically, FIG. 1 is a schematic flowchart illustrating a cross-chain data processing method, according to the present specification, and can also serve as a schematic diagram of a system architecture. Although the present specification
25 provides the method operation steps, apparatuses, or systems structure illustrated in the following embodiments or accompanying drawings, the methods or apparatuses can include more operation steps or modular units, or include fewer operation steps or modular units after some operation steps or modular units are combined, based on conventions or non-creative efforts. For steps or structures logically having no
30 necessary causal relationships, an execution sequence of the steps or the modular structure of the apparatus is not limited to the execution sequence or modular structure illustrated in the embodiments or accompanying drawings of the present specification. When being applied to an actual apparatuses, servers, or end-user device product, the method or modular structure can be executed sequentially or in parallel (for example,

by parallel processors or in a multi-thread processing environment, or even in an implementation environment including a distributed processing server cluster) based on the methods or modular structures illustrated in the embodiments or accompanying drawings.

5 **[0027]** FIG. 1 shows a specific embodiment. Intel SGX can be selected as the core logic in a TEE to implement a client device of a blockchain system. In this way, a TEE program (or referred to as a TEE application) itself can authenticate data on a blockchain. The client device can publish the following information:

10 D01. a code measurement value of the TEE program, used to attest to the external that the TEE program will honestly authenticate data on blockchains;

 D02. a public key of an asymmetric key pair in the TEE, where the private key corresponding to the public key is generated during internal initialization of the TEE and cannot be read from the external, and the public key is used by the TEE to sign cross-chain data to attest to the external that the data is from the client device of
15 the trusted TEE;

 D03. parameters for the initialization of the TEE program, for example, information about a blockchain that the client device of the TEE is connected to; and

 D04. remote attestation information of the previous three types of information of the TEE, where an attester can request a corresponding remote attester
20 to verify the attestation information, to verify whether the previous three types of information are from the client device of the TEE.

[0028] In FIG. 1, blockchain A requests for data of blockchain B through the TEE application. The TEE program can include a simplified payment verification (SPV) client device connected to a blockchain. In SPV, an SPV node does not store all the
25 blockchain data, but stores block head data so that the SPV client device needs a smaller overall storage capacity. Therefore, the SPV client device is also usually referred to as a lightweight client device or a light client device. Blockchain A that needs to read data across chains verifies and anchors the previous data D01 to D04, and uses the remote attestation information of the TEE to verify whether the previous
30 data is from the TEE. Blockchain A compares the code measurement value of the TEE program to determine whether the TEE is the needed cross-chain TEE. Blockchain A also stores the public key of the asymmetric key pair in the TEE. The public key is used to verify and sign cross-chain data from the TEE. Blockchain A further stores the parameters for the initialization of the TEE program, the information

about the blockchain that the client device of the TEE is connected to, so as to learn of the blockchain to which the TEE is bridged. The TEE program (TEE application) illustrated above includes a light client device. In specific implementations, it can be understood as that the TEE application includes a light client program module. It is a method that can be implemented in an application program by a person skilled in the art.

[0029] In an execution process, blockchain A that needs to read data across chains requests the TEE program to read cross-chain data. The TEE program uses the built-in client device (for example, an SPV client device) to request to step across to target blockchain B and verify data of target blockchain B. Then the TEE program can sign, by using the private key generated by Intel SGX, a request result that has been verified by the light client; and return the signed request result to blockchain A. The request result can include the cross-chain data or other related data information.

[0030] Blockchain A can use the previous anchored public key to verify the cross-chain data obtained from the TEE program, to verify that the cross-chain data has actually been verified by the trusted TEE program. In this way, blockchain A obtains the cross-chain data from blockchain B. In addition, the cross-chain data is from blockchain B that has been verified by using the built-in client device, and has been signed by the chip in the TEE. This ensures that the cross-chain data is sent after verified by the trusted TEE. Blockchain A that requests for authentication can use the corresponding public key for verification, and can use the cross-chain data after the cross-chain data passes the verification, to rapidly and efficiently implement authentication of the blockchain data across chains.

[0031] In the previous embodiment, one TEE program can be connected to one target blockchain, and the blockchain that requests for authentication can read the cross-chain data through the TEE program. When there are multiple blockchains, a corresponding TEE program can be configured for each blockchain. Or when a blockchain is added, a corresponding TEE program can be configured for the blockchain. In this way, the cross-chain data exchange processing method, provided in the present specification, that uses the TEE for attestation and conversion can achieve better scalability, effectively increase the throughput of cross-chain data processing, and improve the data processing performance between blockchain systems.

[0032] Therefore, based on the descriptions of the previous implementation

solution, in an embodiment of a cross-chain data processing method provided in the present specification, specifically as shown in FIG. 2, the method can include the following steps:

[0033] S0: A first blockchain sends a cross-chain data request to a trusted attester.

5 **[0034]** S2: The trusted attester obtains cross-chain data from a second blockchain based on the cross-chain data request.

[0035] S4: The trusted attester verifies the cross-chain data; and when the verification is passed, signs the cross-chain data by using a private key corresponding to a TEE policy adopted.

10 **[0036]** S6: The trusted attester returns a signed request result to the first blockchain.

[0037] S8: The first blockchain verifies the request result by using a public key corresponding to the TEE policy; and when the verification is passed, determines that the cross-chain data in the request result is from the trusted attester and has been
15 verified by the trusted attester.

[0038] In the previous embodiment, an implementation of the trusted attester can include the previous TEE application. For example, an apparatus that is implemented by using program code, an end-user device, or a program module can serve as a relay for data exchanges between blockchains. The first blockchain can be the blockchain
20 that requests for the cross-chain data from the trusted attester, and the second blockchain can be the target blockchain that returns the cross-chain data. The descriptions "first" and "second" in the first blockchain and the second blockchain do not limit the first blockchain and the second blockchain to specifically indicating fixed blockchains. In different trusted attesters, the first blockchain and the second
25 blockchain can correspond to different blockchains.

[0039] The first blockchain sends a cross-chain data request to a trusted attester can be specifically a blockchain node on the first blockchain initiating the cross-chain data request. Accordingly, the obtaining cross-chain data from a second blockchain can be specifically obtaining the data from the second blockchain by using a blockchain node
30 on the second blockchain.

[0040] It is worthwhile to note that in the previous embodiment, the TEE policy adopted by the trusted attester can include Intel SGX. However, other TEE policies such as the AMD PSP, the ARM TrustZone technology (applicable to all the ARM processors supporting TrustZone), and the x86-64 instruction set are not excluded

from the present specification. In an implementation provided in the present specification, the TEE policy may be as follows:

[0041] S40: The TEE policy includes Intel SGX.

[0042] The trusted attester can use various methods to verify the data obtained
5 from the second blockchain, for example, verify the integrity of the blockchain data or
verify the blockchain data by using other public blockchains, private blockchains, or
methods agreed in contracts. In an implementation provided in the present
specification, SPV can be used, as described above. Specifically, a light SPV client
device can be provided in the trusted attester, and blockchain data is verified by using
10 blockchain head data. In SPV, an SPV node does not store all the blockchain data, but
stores only block head data so that the SPV client device needs a smaller overall
storage capacity. Therefore, the light SPV client device is also usually referred to as a
lightweight client device, and implements rapid blockchain data verification.
Therefore, in another embodiment provided in the present specification, the verifying
15 the cross-chain data includes the following step:

[0043] S42: Verify the cross-chain data using SPV.

[0044] According to the cross-chain data processing method provided in this
embodiment of the present specification, cross-chain data authentication can be
implemented rapidly and efficiently based on a trusted bridge between TEE
20 blockchains, to achieve high scalability of cross-chain data processing, extend
applicability to more cross-chain data exchange scenarios, and efficiently increase the
upper-limit of cross-chain data throughput traffic with ease, effectively alleviating the
problems that a conventional POS-based consensus relay has limited application
scenarios and is relatively low in efficiency and poor in performance.

[0045] In an implementation of the solution of the previous embodiment, one TEE
program can correspond to one target blockchain, and one SPV client device
corresponding to the target blockchain can be provided in the one TEE program. The
present specification further provides another embodiment. In the another
embodiment, multiple SPV client devices can be provided in one TEE program, and
25 configuration information of a target blockchain corresponding to each SPV client
device is stored, to implement a mapping relationship between the one TEE program
and the multiple SPV client devices. Certainly, multiple TEE programs can be
provided in parallel. For example, when there are 10 target blockchains, three TEE
30 programs can be provided, for example, three processing apparatuses or relays, which

are referred to as T1, T2, and T3. T1 is provided with three SPV client devices, T2 is provided with three SPV client devices, and T3 is provided with four SPV client devices. Each SPV client device corresponds to one target blockchain. Therefore, the present specification further provides an embodiment of another cross-chain data processing method. After the cross-chain data request sent by the first blockchain is received, the method further includes: determining a corresponding SPV client device (or referred to as an SPV application) based on the cross-chain data request; and accordingly, the obtaining cross-chain data from a second blockchain and verifying the cross-chain data includes: using the corresponding SPV client device (SPV application) to obtain the cross-chain data from the second blockchain and verify the cross-chain data.

[0046] In this way, by using the solution of this embodiment, cross-chain data exchanges can be extended more flexibly, the system design can be optimized, more proper layout settings support can be provided, and cross-chain data throughput traffic can be increased.

[0047] FIG. 7 is a schematic diagram illustrating another cross-chain data processing process and system structure, according to the present specification.

[0048] Based on the descriptions of the method in the previous embodiments of data exchanges between multiple blockchains and a trusted attester, the present specification further provides a method embodiment that can be applied to the trusted attester (a relay for data exchanges between blockchains). A specific embodiment is shown in FIG. 3 and may include the following steps:

[0049] S200: Receive a cross-chain data request sent by a first blockchain.

[0050] S202: Obtain cross-chain data from a second blockchain based on the cross-chain data request.

[0051] S204: Verify the cross-chain data; and when the verification is passed, sign the cross-chain data by using a private key corresponding to a TEE policy adopted, where a public key corresponding to the TEE policy is published.

[0052] S206: Return a signed request result to the first blockchain, where the request result includes the cross-chain data and the signature.

[0053] The first blockchain verifies the request result by using the public key corresponding to the TEE policy; and when the verification is passed, determines that the cross-chain data in the request result is from the trusted attester and has been verified by the trusted attester.

[0054] With reference to the descriptions of the previous embodiments about multiple-side data exchanges, in another embodiment of the method, the TEE policy may be as follows:

[0055] S2040: The TEE policy includes Intel SGX.

5 **[0056]** With reference to the descriptions of the previous embodiments about multiple-side data exchanges, in another embodiment of the method, the verifying the cross-chain data includes the following step:

[0057] S2042: Verify the cross-chain data using SPV.

[0058] The previous method embodiments of the present specification are
10 described in a progressive way. For same or similar parts of the embodiments, references can be made to the embodiment mutually. Each embodiment focuses on a difference from other embodiments. For related parts, references can be made to related descriptions in the method embodiments.

[0059] The method embodiments provided in the embodiments of the present
15 application can be executed in fixed end-user devices, mobile devices, servers, or similar computational apparatuses. For example, the method embodiments are executed on a server. FIG. 4 is a block diagram illustrating a hardware structure of a client device that executes a cross-chain data processing method, according to an embodiment of the present specification. Certainly, the method embodiments can also
20 be applied to a blockchain node server. The hardware structure of the blockchain node server can be a hardware structure same as or similar to the hardware structure of the client device, or can have more or fewer hardware structures depending on an end-user device and a server device. Specifically, as shown in FIG. 4, the client device 10 can include one or more processors 102 (where one processor 102 is
25 illustrated in the figure, and the processor 102 can include but is not limited to a processing apparatus such as a microcontroller unit (MCU) or a field programmable gate array (FPGA)), a memory 104 configured to store data, and a transmission module 106 configured to perform communication. A person with ordinary skills in the art can understand that the structure shown in FIG. 4 is merely intended for
30 illustration but does not constitute a limitation on the structure of the previous electronic apparatus. For example, a server can alternatively include more or fewer components than the components shown in FIG. 4, for example, can further include other processing hardware or have a different configuration than the configuration shown in FIG. 4.

[0060] The memory 104 can be configured to store a software program and module of application software, for example, a program instruction or module corresponding to a cross-chain data processing method in the embodiments of the present invention. The processor 102 runs the software program and module stored in the memory 104 to execute various functional applications and data processing. The memory 104 can include a high-speed random access memory, and can further include a nonvolatile memory such as one or more magnetic storage apparatuses, flash memories, or other nonvolatile solid state memories. In some embodiments, the memory 104 can further include memories disposed remotely relatively to the processor 102. The remote memories can be connected to the client device 10 through a network. Embodiments of the network include but are not limited to the Internet, an enterprise intranet, a local area network (LAN), a mobile communications network, and a combination of such networks.

[0061] The transmission module 106 is configured to receive or send data through a network. Specific embodiments of the network can include a wireless network provided by a communications provider of the client device 10. In an embodiment, the transmission module 106 includes a network interface controller (NIC). The NIC can be connected to other network devices through a base station so as to communicate with the Internet. In an embodiment, the transmission module 106 can be a radio frequency (Radio Frequency, RF) module, and is configured to wirelessly communicate with the Internet.

[0062] Based on the cross-chain data processing method described above, the present specification further provides a cross-chain data processing apparatus. The apparatus can include a system (including a distributed system), software (an application), a module, a component, a server, a client device etc. that use the method provided in the embodiments of the present specification and necessary hardware for implementation. Based on the same inventive idea, a processing apparatus in an embodiment provided in the present specification is described in the following embodiments. Implementations of the apparatus for alleviating problems are similar to the implementations of the method. Therefore, for specific implementations of the processing apparatus in this embodiment of the present specification, references can be made to the previous implementations of the method. Details are omitted here for simplicity. Although the apparatus described in the following embodiments can be preferably implemented by software, implementations by hardware or a combination

of software and hardware are also possible and conceived. Specifically, FIG. 5 is a schematic diagram illustrating a modular structure of an embodiment of a cross-chain data processing apparatus provided in the present specification, where the apparatus can serve as a cross-chain relay. The apparatus can be implemented as a standalone client apparatus or server, a processor or server cluster, or a distributed system. Specifically, the apparatus can include: a request receiving module 200, which can be configured to receive a cross-chain data request sent by a first blockchain; a data acquisition module 202, which can be configured to obtain cross-chain data from a second blockchain based on the cross-chain data request; a data verification module 204, which can be configured to verify the cross-chain data; and when the verification is passed, sign the cross-chain data by using a private key corresponding to a TEE policy adopted, where a public key corresponding to the TEE policy is published; and an attestation returning module 206, which can be configured to return a signed request result to the first blockchain, where the request result includes the cross-chain data and the signature.

[0063] With reference to the previous descriptions of the method embodiments, in another embodiment of the apparatus, the TEE policy adopted by the data verification module 204 includes Intel SGX.

[0064] With reference to the previous descriptions of the method embodiments, in another embodiment of the apparatus, the data verification module 204 can include: an SPV unit 2041, which can be configured to verify the cross-chain data using SPV.

[0065] FIG. 6 is a schematic diagram illustrating a modular structure of another embodiment of the apparatus, according to the present specification.

[0066] In another embodiment, the apparatus further includes: a handler determining module, which can be configured to determine a corresponding SPV application based on the cross-chain data request after the cross-chain data request sent by the first blockchain is received; and accordingly, the data acquisition module obtains cross-chain data from a second blockchain and the data verification module verifies the cross-chain data, including: using the corresponding SPV application to obtain the cross-chain data from the second blockchain and verify the cross-chain data.

[0067] It is worthwhile to note that, based on the descriptions of the related method embodiments, the apparatus described in the previous embodiments of the present specification can further include other implementations. For specific implementations,

references can be made to the descriptions of the method embodiments. Details are omitted here for simplicity.

5 **[0068]** The embodiments in the present specification are described in a progressive way. For same or similar parts of the embodiments, references can be made to the embodiments mutually. Each embodiment focuses on a difference from other embodiments. An apparatus embodiment is basically similar to a method embodiment, and therefore is described briefly. For related parts, references can be made to related descriptions in the method embodiment.

10 **[0069]** The cross-chain data processing method and apparatus provided in the embodiments of the present specification can be implemented by a processor executing related program instructions in a computer, for example, implemented on a personal computer (PC) by using the c++ language of the WINDOWS OS, implemented by using application design languages corresponding to other OSs such as LINUX, ANDRIOD, and IOS in combination with necessary hardware,
15 implemented based on multi-server processing of a distributed system, or implemented based on processing logic of a quantum computer. Specifically, in a client device embodiment for implementing the previous method or apparatus provided in the present specification, the client device can include a processor and a memory that is configured to store a processor-executable instruction, where when
20 executing the instruction, the processor implements the steps of: receiving a cross-chain data request sent by a first blockchain; obtaining cross-chain data from a second blockchain based on the cross-chain data request; verifying the cross-chain data; and when the verification is passed, signing the cross-chain data by using a private key corresponding to a TEE policy adopted; and returning a signed request
25 result to the first blockchain, where the request result includes the cross-chain data and the signature.

[0070] As described above, in other embodiments of the client device, the TEE policy includes Intel SGX. Alternatively, when executing the instruction to verify the cross-chain data, the processor verifies the cross-chain data using SPV.

30 **[0071]** The instruction described above can be stored in various types of computer-readable storage media. The computer-readable storage media can include a physical apparatus that is configured to store information and that can digitalize the information and store the information in a medium in an electrical, magnetic, or optical form. The computer-readable storage media in this embodiment can further

include an apparatus that stores information by using electric energy, for example, various memories such as a random access memory (RAM) and a read-only memory (ROM); an apparatus that stores information by using magnetic energy, for example, a hard disk, a floppy disk, a magnetic tape, a magnetic-core memory, a magnetic bubble memory, or a USB flash drive; and an apparatus that stores information in an optical form, for example, a compact disc (CD) or a digital versatile disc (DVD). Certainly, readable storage media in other forms are also available, for example, a quantum memory and a graphene memory. The instruction in the following apparatus, server, client device, or system is the same as the instruction described above.

[0072] The apparatus or client device described above can be deployed in a blockchain system, can implement blockchain protocols, can authenticate data in the blockchain system, and can send blockchain data to other corresponding blockchains in response to requests from the blockchains while interacting with connected blockchains, to implement data exchanges across the blockchains. Therefore, the present specification further provides a blockchain system. The system includes the apparatus or client device according any embodiments of the present specification.

[0073] In an embodiment of the cross-chain system provided in the present specification, the cross-chain system can include multiple blockchains. A target blockchain that is allowed to provide data to other blockchains can be provided with a corresponding relay. The relay can be the client device, trusted attester, or apparatus described above. In this way, highly efficient and highly scalable cross-chain data exchanges can be implemented in the entire cross-chain system. Specifically, a cross-chain system is provided, including at least two blockchains and a relay that corresponds to a target blockchain that returns cross-chain data, where a blockchain that requests for data obtains the cross-chain data from the target blockchain by using the corresponding relay, and the relay includes the apparatus according to any one of the embodiments of the present specification or any one of the method implementations of the present specification.

[0074] When performance of a device for cross-chain data exchanges needs to be scaled up because a new blockchain is connected to the blockchain system, a relay can be added in some embodiments of the present specification, to efficiently and conveniently implement the scale-up. Therefore, in another embodiment of the cross-chain system, when a new blockchain is connected to the cross-chain system and the new blockchain is allowed to exchange cross-chain data with at least one

blockchain in the cross-chain system, a relay is added and configured for the new blockchain.

[0075] It is worthwhile to note that, based on the descriptions of the apparatuses, client devices, system etc. described in the previous embodiments of the present specification can further include other implementations. For specific implementations, references can be made to the descriptions of the method embodiments. Details are omitted here for simplicity.

[0076] The embodiments in the present specification are described in a progressive way. For same or similar parts of the embodiments, references can be made to the embodiments mutually. Each embodiment focuses on a difference from other embodiments. Particularly, a hardware + program type embodiment is basically similar to a method embodiment, and therefore is described briefly. For related parts, references can be made to related descriptions in the method embodiment.

[0077] Specific embodiments of the present specification are described above. Other embodiments fall within the scope of the appended claims. In some situations, the actions or steps described in the claims can be performed in an order different from the order in the embodiments and the desired results can still be achieved. In addition, the process depicted in the accompanying drawings does not necessarily need a particular execution order to achieve the desired results. In some implementations, multi-tasking and concurrent processing is feasible or can be advantageous.

[0078] According to the cross-chain data processing methods, apparatuses, and client devices, and the blockchain systems provided in the embodiments of the present specification, data exchanges between blockchains can be implemented based on a TEE, to rapidly and efficiently implement cross-chain data authentication, implement high scalability of cross-chain data processing, and extend applicability to more cross-chain data exchange scenarios. With the solutions in the embodiments of the present specification, an upper-limit of cross-chain data throughput traffic can be increased efficiently with ease, effectively alleviating the problems that a conventional POS-based consensus relay has limited application scenarios and is relatively low in efficiency and poor in performance.

[0079] Although the present application provides the method operation steps illustrated in the embodiments or flowcharts, more or fewer operation steps can be included, based on conventions or non-creative efforts. A sequence of steps listed in

an embodiment is merely one of various step execution sequences and does not indicate a sole execution sequence. In practice, when being executed by an apparatus or a client product, the steps can be executed sequentially or in parallel (for example, by parallel processors or in a multi-thread processing environment) based on the method illustrated in the embodiments or the accompanying drawings.

[0080] Although data descriptions and data acquisition, storage, exchange, computation, determining, and other operations such as using the SPV client device to verify the data of the target blockchain, selecting Intel SGX to implement the TEE, and using a relay for data exchanges are illustrated in the content of the embodiments of the present specification, it is not limited that the embodiments of the present specification must conform to industrial communications standards, standard blockchain protocols, communications protocols, and standard network modules or templates, or the situations described in the embodiments of the present specification. Some industrial standards, customized implementation solutions, or implementation solutions obtained by slightly modifying the implementations described in the embodiments can also achieve same, equivalent, or similar implementation effects as the previous embodiments or achieve predictable variant implementation effects. Embodiments obtained by applying such modifications or variations to the data acquisition, storage, determining, and processing methods can still fall within the scope of optional implementation solutions of the present specification.

[0081] In the 1990s, whether a technical improvement is a hardware improvement (for example, an improvement to a circuit structure, such as a diode, a transistor, or a switch) or a software improvement (an improvement to a method procedure) can be clearly distinguished. However, as technologies develop, current improvements to many method procedures can be considered as direct improvements to hardware circuit structures. A designer usually programs an improved method procedure into 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, an FPGA) is such an integrated circuit, and a logical function of the PLD is determined by a user through device programming. The designer performs programming to "integrate" a digital system to a PLD without requesting a chip manufacturer to design and produce an application-specific integrated circuit (ASIC) chip. In addition, at present, instead of manually manufacturing an integrated circuit chip, this type of programming is mostly

implemented by using "logic compiler" software. The software 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 Verilog 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.

[0082] A controller can be implemented by using any appropriate method. For example, the controller can be a microprocessor or a processor, or a computer-readable medium that stores 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 ASIC, a programmable logic controller, or a built-in microprocessor. Examples of the controller include but are not limited to the following microprocessors: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. The memory controller can also be implemented as a part of the control logic of the memory. A person skilled in the art also knows that, in addition to implementing the controller by using the computer readable program code, logic programming can be performed on method steps to allow the controller to implement the same function in forms of the logic gate, the switch, the ASIC, the programmable logic controller, and the built-in microcontroller. Therefore, the controller can be considered as a hardware component, and an apparatus configured to implement various functions in the controller can also be considered as a structure in the hardware component. Or the apparatus configured to implement various functions can even be considered as both a software module implementing the method and a structure in the hardware component.

[0083] The system, apparatus, module, or unit illustrated in the previous embodiments 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. Specifically, the computer can be a personal computer, a laptop computer, an on-board human-computer interaction device, a cellular phone, a camera phone, a smartphone, a personal digital assistant (PDA), a media player, a navigation device, an email device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[0084] Although the embodiments of the present specification provide the method operation steps illustrated in the embodiments or flowcharts, more or fewer operation steps can be included, based on conventions or non-creative means. A sequence of steps listed in an embodiment is merely one of various step execution sequences and does not indicate a sole execution sequence. In practice, when being executed by an apparatus or an end-user device product, the steps can be executed sequentially or in parallel (for example, by parallel processors or in a multi-thread processing environment, or even in a distributed data processing environment) based on the method illustrated in the embodiments or the accompanying drawings. The terms "include", "contain", or their any other variants are intended to cover a non-exclusive inclusion, so a process, a method, a product, or a device that includes a list of elements not only includes those elements but also includes other elements which are not expressly listed, or further includes elements inherent to such process, method, product, or device. Without more constraints, the existence of additional identical or equivalent elements in the process, method, product or device that includes the elements is not excluded.

[0085] For ease of description, the apparatus above is described by dividing functions into various modules. Certainly, during implementation of the embodiments of the present specification, the functions of the modules can be implemented in the same one or more pieces of software or hardware or the same combination of one or more pieces of software and hardware, modules implementing the same function can be implemented by using a combination of multiple sub-modules or sub-units, etc. The described apparatus embodiments are merely examples. For example, the unit division is merely logical function division and there can be other division methods in actual implementation. For example, multiple units or components may be combined or integrated into another system, or some features can be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented by using some interfaces. The

indirect couplings or communication connections between the apparatuses or units can be implemented in electronic, mechanical, or other forms.

5 **[0086]** A person skilled in the art also knows that, in addition to implementing the controller by using the computer readable program code, logic programming can be performed on method steps to allow the controller to implement the same function in forms of the logic gate, the switch, the ASIC, the programmable logic controller, and the built-in microcontroller. Therefore, the controller can be considered as a hardware component, and an apparatus configured to implement various functions in the controller can also be considered as a structure in the hardware component. Or the
10 apparatus configured to implement various functions can even be considered as both a software module implementing the method and a structure in the hardware component.

[0087] The present specification is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program
15 product based on the embodiments of the present specification. It is worthwhile to note that computer program instructions can be used to implement each process and/or each block in the flowcharts and/or the block diagrams and a combination of a process and/or a block in the flowcharts and/or the block diagrams. These computer program instructions can be provided for a general-purpose computer, a dedicated
20 computer, an embedded processor, or a processor of another programmable data processing device to generate a machine, so the instructions executed by the computer or the processor of the another programmable data processing device generate an apparatus for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

25 **[0088]** These computer program instructions can be stored in a computer readable memory that can instruct the computer or the another programmable data processing device to work in a specific way, so the instructions stored in the computer readable memory generate an artifact that includes an instruction apparatus. The instruction apparatus implements a specific function in one or more processes in the flowcharts
30 and/or in one or more blocks in the block diagrams.

[0089] These computer program instructions can be loaded onto the computer or another programmable data processing device, so a series of operations and steps are performed on the computer or the another programmable device, thereby generating computer-implemented processing. Therefore, the instructions executed on the

computer or the another programmable device provide steps for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

5 **[0090]** In a typical configuration, a computing device includes one or more processors (CPU), an input/output interface, a network interface, and a memory.

[0091] The memory can include a non-persistent memory, a RAM, a non-volatile memory, and/or another form that are in a computer readable medium, for example, a ROM or a flash memory (flash RAM). The memory is an example of the computer readable medium.

10 **[0092]** The computer readable medium includes persistent, non-persistent, movable, and unmovable media that can store information by using any method or technology. The information can be a computer readable instruction, a data structure, a program module, or other data. Examples of the computer storage medium include but are not limited to a phase change random access memory (PRAM), a static RAM (SRAM), a
15 dynamic RAM (DRAM), a RAM of another type, a ROM, an electrically erasable programmable ROM (EEPROM), a flash memory or another memory technology, a compact disc ROM (CD-ROM), a DVD, or another optical storage, a cassette, a cassette magnetic disk storage, or another magnetic storage device or any other non-transmission medium. The computer storage medium can be configured to store
20 information that can be accessed by a computing device. Based on the definition in the present specification, the computer readable medium does not include transitory computer readable media such as a modulated data signal and carrier.

[0093] A person skilled in the art should understand that an embodiment of the present specification can be provided as a method, a system, or a computer program
25 product. Therefore, the embodiments of the present specification can use a form of hardware only embodiments, software only embodiments, or embodiments with a combination of software and hardware. Moreover, the embodiments of the present specification can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk
30 memory, a CD-ROM, an optical memory, etc.) that include computer-usable program code.

[0094] The embodiments of the present specification can be described in the general context of computer-executable instructions, 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 type. The embodiments of the present specification can alternatively be practiced in a distributed computing environment. In the distributed computing environment, tasks are performed by remote processing devices that are connected through a communications network. In the distributed computing environment, the program module can be located in both local and remote computer storage media including storage devices.

[0095] The embodiments in the present specification are described in a progressive way. For same or similar parts of the embodiments, references can be made to the embodiments mutually. Each embodiment focuses on a difference from other embodiments. Particularly, a system embodiment is similar to a method embodiment, and therefore is described briefly. For related parts, references can be made to related descriptions in the method embodiment. In the descriptions of the present specification, descriptions provided with reference to terms such as "an embodiment", "some embodiments", "an example", "a specific example", or "some examples" intend to mean that a specific feature, structure, material, or characteristic described with reference to the embodiment or example is included in at least one embodiment or example of the embodiments of the present specification. In the present specification, illustrative expressions of the previous terms are not necessarily intended for the same embodiment or example. In addition, the described specific feature, structure, material, or characteristic can be combined in a proper way in any one or more embodiments or examples. Moreover, a person skilled in the art can combine and associate different embodiments or examples and features of different embodiments or examples described in the present specification, provided that the embodiments or examples and the features do not conflict with each other.

[0096] The previous descriptions are merely embodiments of the present specification and are not intended for limiting the embodiments of the present specification. A person skilled in the art knows that the embodiments of the present specification can have various modifications and changes. Any modifications, equivalent replacements, and improvements made within the spirit and principle of the embodiments of the present specification shall fall within the scope of the claims in the present specification.

CLAIMS

What is claimed is:

1. A cross-chain data processing method, wherein the method comprises:
sending, by a first blockchain, a cross-chain data request to a trusted attester;
5 obtaining, by the trusted attester, cross-chain data from a second blockchain based on the cross-chain data request;
verifying, by the trusted attester, the cross-chain data; and when the verification is passed, signing the cross-chain data by using a private key corresponding to a trusted execution environment (TEE) policy adopted;
10 returning, by the trusted attester, a signed request result to the first blockchain; and
verifying, by the first blockchain, the request result by using a public key corresponding to the TEE policy; and when the verification is passed, determining that the cross-chain data in the request result is from the trusted attester and has been
15 verified by the trusted attester.
2. A cross-chain data processing method, wherein the method comprises:
receiving a cross-chain data request sent by a first blockchain;
obtaining cross-chain data from a second blockchain based on the cross-chain data request;
20 verifying the cross-chain data; and when the verification is passed, signing the cross-chain data by using a private key corresponding to a TEE policy adopted; and
returning a signed request result to the first blockchain, wherein the request result comprises the cross-chain data and the signature.
3. The cross-chain data processing method according to claim 2, wherein the
25 TEE policy comprises Intel Software Guard Extensions (SGX).
4. The cross-chain data processing method according to claim 2, wherein the verifying the cross-chain data comprises:
verifying the cross-chain data using Simplified Payment Verification (SPV).
5. The cross-chain data processing method according to claim 4, wherein after
30 the receiving a cross-chain data request sent by a first blockchain, the method further comprises:
determining a corresponding SPV application based on the cross-chain data

request; and

accordingly, the obtaining cross-chain data from a second blockchain and verifying the cross-chain data comprises: using the corresponding SPV application to obtain the cross-chain data from the second blockchain and verify the cross-chain data.

6. A cross-chain data processing apparatus, where the apparatus comprises:

a request receiving module, configured to receive a cross-chain data request sent by a first blockchain;

a data acquisition module, configured to obtain cross-chain data from a second blockchain based on the cross-chain data request;

a data verification module, configured to verify the cross-chain data; and when the verification is passed, sign the cross-chain data by using a private key corresponding to a TEE policy adopted; and

an attestation returning module, configured to return a signed request result to the first blockchain, wherein the request result includes the cross-chain data and the signature.

7. The apparatus according to claim 6, wherein the TEE policy adopted by the data verification module comprises Intel SGX.

8. The apparatus according to claim 6, wherein the data verification module comprises:

an SPV unit, configured to verify the cross-chain data using SPV.

9. The apparatus according to claim 8, wherein the apparatus further comprises:

a handler determining module, configured to determine a corresponding SPV application based on the cross-chain data request after the cross-chain data request sent by the first blockchain is received; and accordingly, the data verification module obtains cross-chain data from a second blockchain and verifies the cross-chain data, comprising: using the corresponding SPV application to obtain the cross-chain data from the second blockchain and verify the cross-chain data.

10. A cross-chain data processing client device, comprising a processor and a memory that is configured to store a processor-executable instruction, wherein when executing the instruction, the processor implements the steps of:

receiving a cross-chain data request sent by a first blockchain;

obtaining cross-chain data from a second blockchain based on the cross-chain data request;

verifying the cross-chain data; and when the verification is passed, signing the cross-chain data by using a private key corresponding to a TEE policy adopted; and returning a signed request result to the first blockchain, wherein the request result includes the cross-chain data and the signature.

5 11. The client device according to claim 10, wherein the TEE policy comprises Intel SGX.

12. The client device according to claim 10, wherein when executing the instruction to verify the cross-chain data, the processor verifies the cross-chain data using SPV.

10 13. The client device according to claim 10, wherein the processor executes the instruction to further implement the step of:

determining a corresponding SPV application based on the cross-chain data request; and

15 accordingly, the obtaining cross-chain data from a second blockchain and verifying the cross-chain data comprises: using the corresponding SPV application to obtain the cross-chain data from the second blockchain and verify the cross-chain data.

14. A blockchain system, comprising the apparatus according to any one of claims 6 to 9 or comprising the client device according to any one of claims 10 to 13.

20 15. A cross-chain system, comprising at least two blockchains and a relay that corresponds to a target blockchain that returns cross-chain data, wherein a blockchain that requests for data obtains the cross-chain data from the target blockchain by using the corresponding relay, and the relay comprises the apparatus according to any one of claims 6 to 9 or an implementation of the method according to any one of claims 2 to

25 5.

16. The cross-chain system according to claim 15, wherein when a new blockchain is connected to the cross-chain system and the new blockchain is allowed to exchange cross-chain data with at least one blockchain in the cross-chain system, a relay is added and configured for the new blockchain.