HASH UPDATING METHODS AND APPARATUSES OF DEDICATED BLOCKCHAIN NODE DEVICE

TECHNICAL FIELD

[0001] The present disclosure relates to the field of blockchain technologies, and in particular to hash updating methods and apparatuses of a dedicated blockchain node device.

BACKGROUND

[0002] Blockchain technology (also called distributed ledger technology) is a decentralized distributed database technology having many characteristics such as decentralization, openness, transparency, immutability and trustability and the like, and thus it is applicable to many application scenarios with high demands for data reliability.

SUMMARY

[0003] In view of this, one or more embodiments of the present disclosure provide hash updating methods and apparatuses of a dedicated blockchain node device.

[0004] To achieve the above-mentioned object, one or more embodiments of the present disclosure provide the following technical solution.

[0005] According to a first aspect of one or more embodiments of the present disclosure, provided is a hash updating method of a dedicated blockchain node device, including:

[0006] sending, by a cryptographic acceleration card assembled on the dedicated blockchain node device, negotiation information to a provider of a new version of disk image, where the negotiation information is used respectively by the provider and the cryptographic acceleration card to determine a deployment key, and the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device;

[0007] receiving, by the cryptographic acceleration card, a new hash value encrypted by the provider using the deployment key, wherein the new hash value corresponds to the new version of disk image;

[0008] replacing, by the cryptographic acceleration card, a stored old hash value corresponding to the old version of disk image with the new hash value, wherein the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[0009] According to a second aspect of one or more embodiments of the present disclosure, provided is a hash updating method of a dedicated blockchain node device, including:

[0010] in response to negotiation information sent by a cryptographic acceleration card assembled on the dedicated blockchain node device, forwarding, by the dedicated blockchain node device, the negotiation information to a provider of a new version of disk image, such that the provider and the cryptographic acceleration card respectively determine a deployment key based on the negotiation information, wherein the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device;

[0011] forwarding, by the dedicated blockchain node device, a new hash value corresponding to the new version of disk image and encrypted by the provider using the deployment key to the cryptographic acceleration card, such that the cryptographic acceleration card replaces a stored old hash value corresponding to the old version of disk image with the new hash value, wherein the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[0012] According to a third aspect of one or more embodiments of the present disclosure, provided is a hash updating apparatus of a dedicated blockchain node device, including:

[0013] a negotiating module, configured to enable a cryptographic acceleration card assembled on the dedicated blockchain node device to send negotiation information to a provider of a new version of disk image, wherein the negotiation information is used respectively by the provider and the cryptographic acceleration card to determine a deployment key, and the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device;

[0014] a receiving module, configured to enable the cryptographic acceleration card to receive a new hash value encrypted by the provider using the deployment key, wherein the new hash value corresponds to the new version of disk image; and

[0015] a replacing module, configured to enable the cryptographic acceleration card to replace a stored old hash value corresponding to the old version of disk image with the new hash value, wherein the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[0016] According to a fourth aspect of one or more embodiments of the present disclosure, provided is a hash updating apparatus of a dedicated blockchain node device, including:

[0017] an information forwarding module, configured to, in response to negotiation information sent by a cryptographic acceleration card assembled on the dedicated blockchain node device, enable the dedicated blockchain node device to forward the negotiation

information to a provider of a new version of disk image, such that the provider and the cryptographic acceleration card respectively determine a deployment key based on the negotiation information, wherein the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device; and

[0018] a hash forwarding module, configured to enable the dedicated blockchain node device to forward a new hash value corresponding to the new version of disk image and encrypted by the provider using the deployment key to the cryptographic acceleration card, such that the cryptographic acceleration card replaces a stored old hash value corresponding to the old version of disk image with the new hash value, wherein the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[0019] According to a fifth aspect of one or more embodiments of the present disclosure, provided is a cryptographic acceleration card, including:

[0020] a processor; and

[0021] a memory for storing processor executable instructions;

[0022] where the processor implements the method according to the first aspect by running the executable instructions.

[0023] According to a sixth aspect of one or more embodiments of the present disclosure, provided is a dedicated blockchain node device, including:

[0024] a processor; and

[0025] a memory for storing processor executable instructions;

[0026] where the processor implements the method according to the second aspect by running the executable instructions.

[0027] According to a seventh aspect of one or more embodiments of the present disclosure, provided is a computer readable storage medium having computer instructions stored thereon, where the instructions are executed by a processor to implement steps in the method according to the first or second aspect.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] FIG. 1 is a flowchart of a hash updating method of a dedicated blockchain node device according to example embodiments of the present disclosure.

[0029] FIG. 2 is a flowchart of another hash updating method of a dedicated blockchain node device according to example embodiments of the present disclosure.

[0030] FIG. 3 is an interaction flowchart of a hash updating method of a dedicated blockchain node device according to example embodiments of the present disclosure.

[0031] FIG. 4 is an interaction flowchart of another hash updating method of a dedicated blockchain node device according to example embodiments of the present disclosure.

[0032] FIG. 5 is a structural schematic diagram of a cryptographic acceleration card according to example embodiments of the present disclosure.

[0033] FIG. 6 is a block diagram of a hash updating apparatus of a dedicated blockchain node device according to example embodiments of the present disclosure.

[0034] FIG. 7 is a structural schematic diagram of a dedicated blockchain node device according to example embodiments of the present disclosure.

[0035] FIG. 8 is a block diagram of another hash updating apparatus of a dedicated blockchain node device according to example embodiments of the present disclosure.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0036] Example embodiments will be described in detail herein with the example embodiments thereof expressed in the drawings. When the following descriptions involve the drawings, same numbers in different drawings represent same or similar elements unless stated otherwise. The implementations described in the following example embodiments do not represent all implementations consistent with one or more embodiments of the present disclosure. On the contrary, they are merely embodiments of apparatuses and methods consistent with some aspects of one or more embodiments of the present disclosure described in detail in the appended claims.

[0037] It should be noted that the steps of corresponding method are not necessarily performed according to the sequence shown in the present disclosure in other embodiments. In some other embodiments, the steps included in the corresponding method can be more or less than described in the specification. Further, a single step described in the specification may be divided into several steps for descriptions in other embodiments while several steps described in the specification may be combined into a single step for descriptions in other embodiments.

[0038] In the early stage of development of the blockchain technology, users mostly add their own personal computer (PC) and laptop computer and the like into a blockchain network to become a blockchain node in the blockchain network. At this time, the stage can be called 1.0 architecture era of blockchain network, in which the behaviors of users to participate in the blockchain network are autonomous and the users also need to perform autonomous

maintenance, for example, perform maintenance and configuration and so on for their devices (for example, PC) participating in the blockchain network. Along with continuous development of the blockchain technology, especially along with increasing needs of users for infrastructures with high performance and high availability, the blockchain network develops into 2.0 architecture era based on cloud service. In the 2.0 architecture era, Blockchain-as-a-Service (BaaS) provides fast and convenient solutions for fast blockchain deployment and technical implementation and supports a large number of blockchain service projects. Generally, BaaS is built on infrastructures such as public cloud or private cloud, which introduces heavy dependence on infrastructure as well as providing strong deployment capability. However, because blockchain is a typical distributed computing technology, not all nodes can be migrated to clouds but privatization deployment is needed. The additional technical migration and maintenance costs brought by the privatization deployment cause inconsistent technical interfaces and high deployment and maintenance costs during an actual implementation. Therefore, to satisfy the needs of users for privatization and security and the like of the blockchain network, it is necessary to perform further architecture upgrade to the blockchain network, thereby realizing 3.0 architecture era based on dedicated blockchain node device (blockchain all-in-one machine).

[0039] Software and hardware integration can be realized for the dedicated blockchain node device. When providing a dedicated blockchain node device, a provider will not only provide hardware devices of the dedicated blockchain node device to users but also provide software configurations for realizing deep optimizations of the hardware devices integrated into the dedicated blockchain node device, thereby realizing the above-mentioned software-hardware integration.

[0040] Hardware optimization can be realized for the dedicated blockchain node device. For example, a dedicated smart contract processing chip can be deployed on the dedicated blockchain node device. For example, the smart contract processing chip can be Field Programmable Gate Array (FPGA) chip, or another type of chip to increase the processing efficiency for a smart contract. A hardware root-of-trust key can be deployed on the smart contract processing chip, for example, the hardware root-of-trust key can be pre-programmed by the provider into the smart contract processing chip and the provider can also know a public key corresponding to the hardware root-of-trust key (for example, the key is disclosed). Therefore, the smart contract processing chip can send negotiation information to the provider and sign the negotiation information by using the hardware root-of-trust key, so that the provider can verify the signature based on the corresponding public key; and, after successful signature verification, it is ensured that the smart contract processing chip and the provider

obtain the same key through negotiation based on the above-mentioned negotiation information. The negotiated key can include an image deployment key, and thus the provider can encrypt and transmit a binary disk image needed by the blockchain node to the smart contract processing chip based on the image deployment key, and the smart contract processing chip can decrypt and deploy the binary disk image based on the image deployment key. The negotiated key can include a service secret deployment key, and thus the provider can encrypt and transmit a node private key of the blockchain node, a service root key of the blockchain node, etc., to the smart contract processing chip based on the service secret deployment key, and the smart contract processing chip can obtain and deploy the node private key and the service root key and the like based on the service secret deployment key to satisfy the privacy transaction needs in a blockchain scenario. For example, the node private key corresponds to a node public key, and thus a client device can perform encryption and transmission for a blockchain transaction by using the node public key, and the blockchain node can perform decryption by using the node private key. The service root key is a symmetric key which can be used to perform encrypted storage for service data such as contract codes and value of contract status and the like. The service root key may not be directly used, and the smart contract processing chip can perform encryption and decryption through a derivation key of the service root key to reduce the security risk of the service root key. Through reliable management for the node private key and the service root key (or its derivation key), data will be always in encrypted status unless processed by the smart contract processing chip. Therefore, the smart contract processing chip actually forms a Trusted Execution Environment (TEE) of hardware on the dedicated blockchain node device, so as to ensure the data requiring privacy protection such as transactions, contract codes, and contract status will not be leaked.

[0041] For another example, an intelligent network card can be deployed on the dedicated blockchain node device. In addition to realizing a traditional network card function, the intelligent network card also can replace or assist a CPU of the dedicated blockchain node device to perform partial functions so as to offload computation of the CPU. Especially, the operations with intensive network I/O can be transferred from CPU to the intelligent network card to perform, so that the CPU can process more computation-intensive operations, for example, transaction processing, and storage processing and the like. Compared with other components (for example, CPU) on the dedicated blockchain node device, the intelligent network card is closer to the network regardless of physical level or logical level, so the intelligent network card can always fetch data transmitted in the network preferentially. Therefore, with no storage access or a small amount of storage access is involved, the

intelligent network card can process these data with a relatively higher processing efficiency and a relatively smaller delay, and a relatively larger throughput, so as to achieve a higher performance benefit with a lower cost. For example, in consensus algorithm, there is almost no need to access storage except in the cases of change of network status, addition and deletion of node, change of consensus configuration and the like. Therefore, the consensus operation can be completed by the intelligent network card and only need to inform the CPU of a consensus result. Therefore, the CPU is not required to directly participate in the consensus process, thereby significantly improving the consensus efficiency. Similarly, the same effect can be achieved in forwarding transactions by the intelligent network card and achieving block synchronization by the intelligent network card on a newly-added blockchain node and the like and will not be repeated herein. Furthermore, after receiving transactions, the intelligent network card can identify or filter out a replay transaction by comparing the received transaction with historical transactions, for example, comparing data fields of sender information of transaction, destination address, time stamp, and hash value and the like. The intelligent network card can also perform content analysis for those received transactions, so as to filter out illegal transactions or predefined undesired transactions and the like as a supplementation to layer-2 or layer-3 packet filtering implemented by a switch.

[0042] For another example, a cryptographic acceleration card which is also called a high-speed cryptographic acceleration card can be deployed on the dedicated blockchain node device. The cryptographic acceleration card can realize total encrypted memory, defend against side-channel attacks by hardware reinforcement, and also realize physical protection against approaches such as probe, laser and the like, having very high security. For example, the cryptographic acceleration card used on the dedicated blockchain node device can have level-2 qualification from the State Cryptography Administration, level-3 qualification from the State Cryptography Administration and the like. When the cryptographic acceleration card is deployed, the hardware roof-of-trust key as described above-mentioned can be maintained in the cryptographic acceleration card, and the cryptographic acceleration card can perform signature operation based on the hardware roof-of-trust key and replace or assist the smart contract processing chip to complete the operations such as the key negotiation as described above-mentioned. Similarly, the cryptographic acceleration card can be used to maintain a public key so that the cryptographic acceleration card can realize signature verification operation based on the maintained public key. In short, at least part of operations relating to key management, encryption and decryption, and signature verification and the like on the dedicated blockchain node device can be handed over to the cryptographic acceleration card, so that very high security can be realized and task offloading can be realized for the CPU of

the dedicated blockchain node device or the above-mentioned smart contract processing chip, thereby improving the processing efficiency.

[0043] Software optimization can be realized for the dedicated blockchain node device. For example, a certificate authority service can be built in the dedicated blockchain node device to realize automatic certificate issuing, node identity authentication, automatic blockchain construction, and automatic adding of blockchain node, thereby realizing the plug and play of the dedicated blockchain node device. In this case, a user can realize fast deployment of the dedicated blockchain node device. In addition to quickly establishing a private blockchain network among a plurality of dedicated blockchain node devices, the dedicated blockchain node device can integrate a standardized on-cloud service interface to enable the dedicated blockchain node device to automatically connect to on-cloud service, thereby realizing hybrid deployment between the dedicated blockchain node device and the cloud-deployed blockchain node to construct a hybrid blockchain network. The dedicated blockchain node device can also integrate a standardized cross-chain service interface to enable the dedicated blockchain node device to realize cross-chain services based on a standardized cross-chain protocol or standardized cross-chain service, thereby greatly expanding the application scenarios of the dedicated blockchain node device, and satisfying the cross-chain needs of users. For example, cross-chain data interaction between different blockchain networks is achieved, and for another example, cross-chain data interaction between the blockchain network and an off-chain computing node and the like is achieved (for example, the off-chain computing node shares computation task for the blockchain node and the like).

[0044] Based on a unified software logic, the cryptographic acceleration card assembled on the dedicated blockchain node device can realize the process of updating a self-stored old hash value corresponding to an old version of disk image. In this updating process, the cryptographic acceleration card firstly determines a deployment key together with a provider of a new version of disk image (hereinafter called provider for short) through negotiation information, and then a new hash value corresponding to the new version of disk image is encrypted and transmitted to the cryptographic acceleration card from the provider and finally, the cryptographic acceleration card replaces the locally-stored old hash value with the new hash value obtained through decryption, thereby the old hash value is updated. The process will be described below in combination with accompanying drawings.

[0045] FIG. 1 is a flowchart of a hash updating method of a dedicated blockchain node device according to example embodiments of the present disclosure. The method is applied to a cryptographic acceleration card assembled on the dedicated blockchain node device. In order to distinguish from the cryptographic acceleration card, the dedicated blockchain node

device in the subsequent descriptions of the present disclosure can be understood as a CPU of the dedicated blockchain node device, or another component other than the cryptographic acceleration card assembled in the dedicated blockchain node device. As shown in FIG. 1, the method can include the following steps.

[0046] At step 102, the cryptographic acceleration card assembled on the dedicated blockchain node device sends negotiation information to a provider of a new version of disk image, where the negotiation information is respectively used by the provider and the cryptographic acceleration card to determine a deployment key, and the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device.

[0047] In this embodiment, a disk image (that has not been executed yet) is locally pre-deployed in the dedicated blockchain node device. The specific form of the disk image is not limited herein in the present disclosure. For example, the disk image can be an executable disk image, for example, an executable file of .exe format. At this time, the executable file can be pre-installed in an execution component of the dedicated blockchain node device, such as a hard disk drive. The disk image can also be a binary disk image, for example, a binary image of .bin format. At this time, the binary image can be pre-stored at a proper position of an execution component of the dedicated blockchain node device, such as a hard disk drive to be invoked and executed. Further, the disk image can be a binary disk image corresponding to a blockchain node and deployed in the dedicated blockchain node device. Correspondingly, when the binary disk image is executed, the dedicated blockchain node device is implemented as a blockchain node, for example, realize one or more functions of blockchain visualization, contract creation and deployment, contract execution, key management and privacy computing and the like. The disk image can also be a platform disk image including the above-mentioned binary disk image corresponding to the blockchain node and deployed in the dedicated blockchain node device. When the platform disk image is executed, the dedicated blockchain node device is implemented not only as a blockchain node, but also can realize one or more other functions such as image processing, node monitoring and service monitoring, which will not be redundantly described herein.

[0048] Correspondingly, the cryptographic acceleration card currently already stored an old hash value corresponding to the old version of disk image. The solution of the present disclosure aims to replace the old hash value with a new hash value corresponding to the above-mentioned new version of disk image so as to update the old hash value, thus ensuring the hash value stored in the cryptographic acceleration card always corresponds to the new

hash value.

[0049] The deployment key determined by the cryptographic acceleration card and the provider based on the above-mentioned negotiation information is used to update the old hash value corresponding to the old version of disk image and stored currently in the cryptographic acceleration card. Therefore, in an embodiment, the cryptographic acceleration card can send the above-mentioned negotiation information to the provider after determining that hash updating needs to be performed. For example, before updating a disk image, the cryptographic acceleration card can proactively send an update inquiry message to the provider and determine whether to perform hash updating based on a confirmation message returned by the provider for the update inquiry message. For example, the cryptographic acceleration card can send the above-mentioned update inquiry message to the provider according to a predetermined period when the dedicated blockchain node device is started for running, or before the disk image is executed, so as to know update needs for the disk image in time. For another example, before updating the disk image, the cryptographic acceleration card can determine hash updating is needed after receiving an update notification for the disk image sent by the provider. For another example, when or after updating the disk image, the cryptographic acceleration card can determine hash updating is needed according to a detected update event of the disk image, and then send the above-mentioned negotiation information to the provider.

[0050] In an embodiment, the cryptographic acceleration card can generate the above-mentioned negotiation information in a trusted execution environment (TEE), and sign it with its own corresponding private key. For example, the cryptographic acceleration card can generate one random number S1 as its own private key and obtain a public key by performing modular exponentiation, and then sign parameters of the modular exponentiation and the above-mentioned public key by selecting a key exchange algorithm such as RSA, ECDSA or DSA or the like and then send the above-mentioned parameters, the public key and the signature to the provider in an associated manner. At this time, the above-mentioned negotiation information includes the parameters for performing modular exponentiation for the above-mentioned random number and the public key obtained by modular exponentiation. Correspondingly, after receiving the above-mentioned negotiation information, the provider can generate a random number S2 based on the above-mentioned negotiation information in a case of successful signature verification, and return a public key encrypted by the random number S2 to the cryptographic acceleration card. In this way, the cryptographic acceleration card and the provider can respectively generate a same deployment key based on the above-mentioned process. The specific process of the above-mentioned negotiation can be

referred to the contents disclosed by the existing technology and will not be repeated herein.

[0051] In an embodiment, a root-of-trust key (Attestation Key) can be pre-deployed in the cryptographic acceleration card. In a key negotiation process, the cryptographic acceleration card can sign the negotiation information with the root-of-trust key. The root-of-trust key can be pre-programmed in the cryptographic acceleration card or the root-of-trust key can be deployed into the cryptographic acceleration card by a dedicated blockchain node device or another object in an offline secure environment. For example, when the cryptographic acceleration card is a cryptographic acceleration card based on FPGA chip, the root-of-trust key can be programmed into the FPGA chip. However, due to volatility of the FPGA chip, the stored root-of-trust key will be lost upon power down, so the cryptographic acceleration card needs to re-obtain and store the root-of-trust key after power up. To reduce the number of programming times of a client device, the FPGA structure can further include a memory connected with the FPGA chip, so that the root-of-trust key is deployed in the memory and the FPGA chip can read the root-of-trust key from the memory to realize relevant functions. Due to non-volatility of the memory, the memory can still store the root-of-trust key even upon power down, and the cryptographic acceleration card only needs to re-read the root-of-trust key from the memory to the FPGA chip after power up, without re-obtaining and re-programming. The above-mentioned memory can have several forms, for example, can be a re-writable non-volatile memory such as flash memory, or can be a non-rewritable memory such as fuse memory, which will not be limited herein.

[0052] The above-mentioned the root-of-trust key is an asymmetric key and a public key corresponding to the root-of-trust key is disclosed or maintained only by the provider. In this case, even if the root-of-trust key is not deployed (pre-built, or deployed by another object) by the provider, the provider can still verify, by using the disclosed public key, a signature generated by the root-of-trust key. The provider and the cryptographic acceleration card can realize key negotiation using the root-of-trust key: during the key negotiation process, the cryptographic acceleration card can sign the above-mentioned negotiation information using the above-mentioned root-of-trust key, so that the provider performs verification using the public key corresponding to the root-of-trust key maintained by itself, thus ensuring the privacy of the negotiation information during transmission process.

[0053] Further, the deployment key determined respectively by the cryptographic acceleration card and the provider can be directly obtained by the cryptographic acceleration card and the provider through key negotiation, that is, the deployment key is a secret value (or, master key); or, the deployment key can respectively be derived from the above-mentioned secret value by the cryptographic acceleration card and the

provider through a key derivation function. Based on the above-mentioned process, the provider and the cryptographic acceleration card can complete key negotiation such that the provider and the cryptographic acceleration card can respectively obtain the same deployment key.

[0054] Further, when the new hash value is encrypted and transmitted using the deployment key determined in the above-mentioned key negotiation process, the provider and the cryptographic acceleration card can store the above-mentioned deployment key in their respective TEEs after occurrence of the first negotiation, so that the deployment key can be used in subsequent transmission of the new hash value (corresponding to disk image update). Thus, the deployment key negotiated in one time can be used several times, thereby reducing the workload of the key negotiation and improving the usage efficiency of the key. When a same pair of deployment keys are used a plurality of times, the risk of leakage of the keys can be increased correspondingly. As a result, to avoid leakage of the keys, an upper limit number of usage times or a valid time length can be set for the negotiated deployment key. Key negotiation can also be performed temporarily every time the new hash value is transmitted in a scenario with high security requirements, and the negotiated deployment key is discarded after each use to ensure the deployment key is not leaked. In this embodiment, key negotiation is performed based on the pre-deployed root-of-trust key, which not only ensures the privacy of the new hash value in transmission process but also reduces the possibility that the root-of-trust key of the cryptographic acceleration card is leaked, thereby improving the security of the cryptographic acceleration card.

[0055] At step 104, the cryptographic acceleration card receives a new hash value encrypted by the provider using the deployment key, where the new hash value corresponds to the new version of disk image.

[0056] In an embodiment, an old hash value currently stored in the cryptographic acceleration card can be obtained through calculation by the provider in a TEE according to an old version of disk image provided by the provider. Similarly, the new hash value for replacing the old hash value can also be obtained through calculation by the provider according to the new version of disk image for updating the old version of disk image. The above-mentioned TEE corresponding to the provider can be constructed by adopting any technique of related technologies, and the old hash value and the new hash value can be calculated by the provider by adopting any hash algorithm of related technologies, which is not limited herein. A hash algorithm adopted to calculate the old hash value according to the old version of disk image can be identical to or different from a hash algorithm adopted to calculate the new hash value according to the new version of disk image, which is not limited

herein.

[0057] In an embodiment, the provider can encrypt the new hash value obtained in the above-mentioned calculation process by using the deployment key determined in the above-mentioned key negotiation process and then transmit the encrypted new hash value to the cryptographic acceleration card, thus ensuring the privacy of the new hash value in transmission process. For example, the provider can transmit the encrypted new hash value directly to the cryptographic acceleration card to reduce the possibility of information leakage of the new hash value in the transmission process; or, in a case of inability to perform direct communication with the cryptographic acceleration card due to unavailability of a network address of the cryptographic acceleration card and the like, the provider can also firstly send the encrypted new hash value to the dedicated blockchain node device which then forwards the encrypted new hash value to the cryptographic acceleration card.

[0058] At step 106, the cryptographic acceleration card replaces the stored old hash value corresponding to the old version of disk image with the new hash value, where the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[0059] In an embodiment, after receiving the new hash value encrypted using the deployment key, the cryptographic acceleration card can obtain the new hash value in plaintext status by performing decryption using the deployment key determined in the above-mentioned key negotiation process in the TEE corresponding to the cryptographic acceleration card, and then replace the currently-stored old hash value with the new hash value. Thus, with the pre-deployed root-of-trust key, the temporarily-determined deployment key can be used in the transmission process of the new hash value, which not only ensures the privacy of the new hash value in the transmission process but also reduces the possibility that the root-of-trust key of the cryptographic acceleration card is leaked, thereby improving the security of the cryptographic acceleration card. In a case that a plurality of old versions of disk image respectively correspond to different new versions of disk image in the dedicated blockchain node device at the same time, the cryptographic acceleration card can establish an association relationship between a new hash value corresponding to any new version of disk image and a file identifier of the new version of disk image, and encrypt and store various new hash values according to respective association relationships, so as to specifically update any old hash value.

[0060] In this embodiment, the above-mentioned old hash value stored in the cryptographic

acceleration card or the updated new hash value can both be used to verify whether the corresponding disk image is trusted before the corresponding disk image is executed. For example, before the old version of disk image and the old hash value are updated, a current hash value of the old version of disk image can be calculated and then compared with the old hash value: if a comparison result indicates that the current hash value is identical to the old hash value, it indicates that the old version of disk image locally deployed in the dedicated blockchain node device is the disk image provided by the provider, which is un-tampered and successfully deployed, and thus the old version of disk image is trusted. Therefore, the dedicated blockchain node device can execute the old version of disk image to form a blockchain node. On the contrary, if the comparison result indicates that the current hash value is different from the old hash value, it indicates that the old version of disk image locally deployed in the dedicated blockchain node device is not the disk image provided by the provider but can be illegally tampered or erroneously deployed, and thus the old version of disk image is not trusted. At this time, the dedicated blockchain node device can refuse to execute the old version of disk image. Similarly, after the old version of disk image and the old hash value are respectively updated to the new version of disk image and the new hash value, before the new version of disk image is executed, a current hash value of the new version of disk image can also be calculated to determine whether to execute, by the dedicated blockchain node device, the new version of disk image by determining whether the current hash value is identical to the above-mentioned new hash value in the same above-mentioned process.

[0061] Actually, since the cryptographic acceleration card is assembled in the dedicated blockchain node device, when the cryptographic acceleration card sends the negotiation information to the provider and when the provider sends the new hash value encrypted by the negotiated deployment key to the cryptographic acceleration card, the dedicated blockchain node device can forward the above-mentioned negotiation information and the new hash value in ciphertext status. FIG. 2 is a flowchart of another hash updating method of a dedicated blockchain node device according to example embodiments of the present disclosure. The method is applied to the dedicated blockchain node device. As shown in FIG. 2, the method can include the following steps.

[0062] At step 202, in response to negotiation information sent by a cryptographic acceleration card assembled on the dedicated blockchain node device, the dedicated blockchain node device forwards the negotiation information to a provider of a new version of disk image, such that the provider and the cryptographic acceleration card respectively determine a deployment key based on the negotiation information; where the new version of

disk image is used to update an old version of disk image deployed in the dedicated blockchain node device.

[0063] At step 204, the dedicated blockchain node device forwards a new hash value corresponding to the new version of disk image and encrypted by the provider using the deployment key to the cryptographic acceleration card, such that the cryptographic acceleration card replaces a stored old hash value corresponding to the old version of disk image with the new hash value, where the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[0064] In this embodiment, the cryptographic acceleration card forwards the negotiation information to the provider of the new version of disk image through the dedicated blockchain node device, such that a same deployment key is determined between the cryptographic acceleration card and the provider; after that, the provider forwards the new hash value encrypted by the deployment key to the cryptographic acceleration card through the dedicated blockchain node device, and the cryptographic acceleration card parses the new hash value with its own deployment key after receiving the new hash value and replaces the old hash value with the obtained new hash value, thereby realizing updating operation for the old hash value stored in the cryptographic acceleration card. The above-mentioned old hash value stored in the cryptographic acceleration card or the updated new hash value can both be used to verify whether the corresponding disk image is trusted before the corresponding disk image is executed. Reference can be made to the above-mentioned embodiments corresponding to FIG. 1 for the above-mentioned hash updating process and the updated image verification process and thus redundant descriptions will not be made herein.

[0065] In the above-mentioned embodiments of the present disclosure, the cryptographic acceleration card sends the negotiation information to the provider of the new version of disk image such that the cryptographic acceleration card and the provider determine a deployment key for encryption and transmission of the new hash value, to ensure the new hash value received by the cryptographic acceleration card is indeed a trusted hash value generated by the provider based on the new version of disk image and sent to the cryptographic acceleration card, thereby the old hash value corresponding to the old version of disk image and currently stored in the cryptographic acceleration card is updated. Further, it helps to realize trusted execution of the updated new version of disk image and trusted startup of the dedicated blockchain node device.

[0066] As mentioned above, the negotiation information and the new hash value transmitted

between the cryptographic acceleration card and the provider can be directly transmitted between the cryptographic acceleration card and the provider or can be forwarded by the dedicated blockchain node device, which will be described below in combination with FIGS. 3 and 4.

[0067] FIG. 3 is an interaction flowchart of a hash updating method of a dedicated blockchain node device. In combination with FIG. 3, the process in which the old hash value stored in the cryptographic acceleration card is updated by directly sending data between the cryptographic acceleration card and the provider will be described below. As shown in FIG. 3, the process can include the following steps.

[0068] At step 302, a root-of-trust key is locally pre-stored in the cryptographic acceleration card.

[0069] In this embodiment, the cryptographic acceleration card can pre-store a root-of-trust key in a TEE. For example, the root-of-trust key can be pre-stored in the cryptographic acceleration card when the dedicated blockchain node device leaves factory or the root-of-trust key can be deployed into the cryptographic acceleration card by the dedicated blockchain node device or another object in an offline secure environment. For example, when the cryptographic acceleration card is a cryptographic acceleration card based on FPGA chip, the root-of-trust key can be programmed to be deployed into the FPGA structure of the cryptographic acceleration card, for example, programmed into a secure key zone of the e cryptographic acceleration card. However, due to volatility of the FPGA chip, the stored root-of-trust key will be lost upon power down, so that the cryptographic acceleration card needs to re-deploy the root-of-trust key after power up. To reduce the number of deployments of a client device, the FPGA structure can further include a memory connected with the FPGA chip, so that the root-of-trust key is deployed in the memory and the FPGA chip can read the root-of-trust key from the memory to realize relevant functions. Due to non-volatility of the memory, the memory can still store the root-of-trust key even upon power down, and the cryptographic acceleration card only needs to re-read the root-of-trust key from the memory to the FPGA chip after power up, without re-deploying. The above-mentioned memory can be in several forms, for example, can be a re-writable non-volatile memory such as flash memory, or can be a non-rewritable memory such as fuse memory, which will not be limited herein.

[0070] The above-mentioned the root-of-trust key is an asymmetric key and a public key corresponding to the root-of-trust key is disclosed or maintained only by the provider. In this case, even if the root-of-trust key is not deployed (pre-built by the provider, or deployed by another object) by the provider, the provider can still verify, by using the disclosed public key,

a signature generated by the root-of-trust key. For example, the negotiation is realized based on SM2 algorithm (or another algorithm). In this case, the provider and the cryptographic acceleration card need to perform information interaction at least once during the negotiation process: when sending the negotiation information to the provider, the cryptographic acceleration card can sign the above-mentioned negotiation information using the above-mentioned root-of-trust key, so that the provider performs signature verification using the disclosed public key after receiving the signed negotiation information so as to determine the negotiation information is indeed sent by the cryptographic acceleration card and trusts the negotiation information; when the signature verification is unsuccessful, the provider can choose not to trust the received negotiation information. The above-mentioned verification process can be locally completed by the provider without assistance of any third party, which, on the one hand, reduces the number of verifications and improves the verification efficiency, and, on the other hand, avoids additional security risk caused by intervention of any third party or the increased number of interactions, thus ensuring the reliability of the verification result is not dependent on self-control and credibility of the third party.

[0071] At step 304, the provider of the new version of disk image calculates a new hash value corresponding to the new version of disk image in its own TEE.

[0072] In an embodiment, for the old version of disk image deployed in the dedicated blockchain node device, the provider can firstly determine the new version of disk image for updating the old version of disk image and then calculate the new hash value corresponding to the new version of disk image. The TEE can be locally deployed in the provider. At this time, a standard hash value can be calculated directly by the provider in the TEE. Alternatively, to reduce the calculation workload of the provider, the TEE can also be deployed in another component relating to the provider, and then the standard hash value is controlled to be calculated in the TEE by the provider and then encrypted and transmitted to the provider. Further, the above-mentioned new hash value can be calculated based on an algorithm such as SHA algorithm, MD4 algorithm, MD5 algorithm, ETHASH algorithm, and SCRYPT algorithm. The construction process of the above-mentioned TEE and the hash calculation process can be referred to the contents disclosed by the existing technology and thus will not be repeated herein.

[0073] Correspondingly, the old hash value stored in the cryptographic acceleration card can also be obtained by the provider by performing calculation based on the old version of disk image provided by the provider in the TEE. The hash algorithm adopted to calculate the old hash value according to the old version of disk image can be identical to or different from the hash algorithm adopted to calculate the new hash value according to the new version of disk

image, which is not limited herein. Furthermore, the TEE corresponding to the provider and the cryptographic acceleration card can be constructed in several manners, for example, can be constructed based on Intel Software Guard Extensions (SGX) or based on AMD TrustZone, which is not limited herein.

[0074] The above-mentioned steps 302 and 304 both are pre-executed steps, and a time interval between the two steps 302 and 304 and the subsequent step 306 is not limited in the present disclosure. Actually, the step 304 can be performed at any time before step 312, for example, the corresponding new hash value can be calculated after the new version of disk image is generated or determined or after the deployment key between the provider and the cryptographic acceleration card is determined. Thus, the time of performing the step 304 is not limited in the present disclosure.

[0075] At step 306, the cryptographic acceleration card sends the negotiation information and a signature performed using the root-of-trust key to the provider.

[0076] In an embodiment, the cryptographic acceleration card can send the above-mentioned negotiation information to the provider after determining that hash updating needs to be performed. For example, before updating a disk image, the cryptographic acceleration card can proactively send an update inquiry message to the provider and determine whether to perform hash updating based on a confirmation message returned by the provider for the update inquiry message. For example, the cryptographic acceleration card can send the above-mentioned update inquiry message to the provider when the dedicated blockchain node device is started for running, or before the disk image is executed or according to a predetermined period (e.g. one day, one week or one month or the like), so as to know update needs for the disk image in time. For another example, before updating the disk image, the cryptographic acceleration card can determine hash updating is required after receiving an update notification for the disk image sent by the provider. For another example, when or after updating the disk image, the cryptographic acceleration card can determine hash updating is needed according to a detected update event of the disk image, and then send the above-mentioned negotiation information to the provider.

[0077] In an embodiment, the cryptographic acceleration card can generate the above-mentioned negotiation information in a trusted execution environment (TEE) corresponding to the cryptographic acceleration card in case of determining that hash updating is needed, and sign the negotiation information with the above-mentioned root-of-trust key and then send the negotiation information to the provider. For example, the cryptographic acceleration card can generate one random number S1 as its own private key and obtain a public key by performing modular exponentiation, and then sign parameters of

the modular exponentiation and the above-mentioned public key by selecting a key exchange algorithm such as RSA, ECDSA or DSA or the like and then send the above-mentioned parameters, the public key and the signature to the provider in an associated manner. At this time, the negotiation information as above includes the parameters for performing modular exponentiation for the above-mentioned random number and the public key obtained by modular exponentiation.

[0078] At step 308, the provider verifies the signature of the negotiation information.

[0079] In an embodiment, the provider can verify the above-mentioned negotiation information based on a public key corresponding to the root-of-trust key maintained by the provider to ensure the privacy of the negotiation information during a transmission process. For example, the provider can realize verification of the signature without decrypting the signature (based on the feature of DH: only public keys are exchanged and encryption operation is performed) and the specific process can be referred to the contents disclosed by existing technology. Alternatively, the cryptographic acceleration card can first calculate a first information hash of the above-mentioned negotiation information using a particular hash algorithm, and then obtain the signature of the negotiation information by encrypting the first information hash using the above-mentioned root-of-trust key. Correspondingly, after receiving the above-mentioned negotiation information and the signature, the provider can firstly obtain a signature hash by decrypting the signature using a public key corresponding to the root-of-trust key maintained by the provider and calculate a second information hash of the above-mentioned negotiation information based on the same algorithm, and then determine, through comparison, whether the signature hash is identical to the second information hash: if the signature hash is identical to the second information hash, it is determined that the above-mentioned signature is successfully verified; otherwise, if the signature hash is different from the second information hash, it is determined that the above-mentioned signature is unsuccessfully verified. The above-mentioned comparison can be performed in a full text bit-by-bit comparison manner, that is, the bits of the signature hash and the second information hash are compared bit by bit along a predetermined direction: if the values of all bits of the signature hash are correspondingly equal to the values of all bits of the second information hash, it is determined that the signature hash and the second information hash are same; on the contrary, if the value of any bit of the signature hash is unequal to the value of the corresponding bit of the second information hash, it is determined that the signature hash and the second information hash are not same.

[0080] In a case of successful verification of the above-mentioned signature, it is determined that the negotiation information is indeed sent by the cryptographic acceleration card and

trusted, and the deployment key is further determined by performing step 310a. In a case of unsuccessful verification of the above-mentioned signature, it indicates that the negotiation information may not be sent by the cryptographic acceleration card or can be sent by the cryptographic acceleration card but tampered by any third party. Thus, the provider can choose not to trust the negotiation information. At this time, the negotiation information can be directly discarded and/or a warning is initiated for the negotiation information.

[0081] At step 310a, the provider determines a deployment key according to the negotiation information.

[0082] At step 310b, the cryptographic acceleration card determines a deployment key according to the negotiation information.

[0083] The provider and the cryptographic acceleration card can respectively determine their own deployment keys according to the negotiation information. Continued from the embodiment of the previous step 306, correspondingly, after receiving the above-mentioned negotiation information, the provider can generate a random number S2 based on the above-mentioned negotiation information in a case of successful signature verification, and return a public key encrypted by the random number S2 to the cryptographic acceleration card. In this way, the cryptographic acceleration card and the provider can respectively generate a same deployment key based on the above-mentioned process. The specific process of the above-mentioned negotiation can be referred to the contents disclosed by the existing technology and will not be repeated herein.

[0084] Eventually, the provider and the cryptographic acceleration card can respectively obtain the same deployment key, or obtain a private key and a public key of a pair of asymmetric keys, thus completing the key negotiation process.

[0085] At step 312, the provider sends a new hash value encrypted using the deployment key to the cryptographic acceleration card.

[0086] At step 314, the cryptographic acceleration card obtains the new hash value by performing decryption using its own deployment key and replaces an old hash value currently stored in the cryptographic acceleration card with the new hash value.

[0087] In this embodiment, after determining the deployment key, the provider can encrypt the new hash value obtained at step 304 by using the deployment key, and then send the encrypted new hash value to the cryptographic acceleration card. Correspondingly, after receiving the new hash value in ciphertext status, the cryptographic acceleration card can decrypt the new hash value in ciphertext status using the deployment key determined in the above-mentioned key negotiation process in the TEE corresponding to the cryptographic acceleration card so as to obtain the new hash value in plaintext status.

[0088] In this embodiment, the cryptographic acceleration card can replace the locally-stored old hash value with the new hash value in plaintext status. For example, the cryptographic acceleration card can write the new hash value into a storage space corresponding to the old hash value in its secure key zone, or FPGA chip or memory connected with the FPGA chip, so as to overwrite the old hash value in corresponding position to update the old hash value.

[0089] In an embodiment, after receiving the new hash value in ciphertext status, the cryptographic acceleration card does not directly decrypt the new hash value but stores the new hash value in the ciphertext status into the secure key zone, FPGA chip or the memory connected with the FPGA chip, and decrypts the new hash value using the deployment key only when the new hash value is invoked for use, thereby avoiding possible hash leakage caused by earlier decryption of the new hash value. Further, the new hash value in ciphertext status can be stored in a non-TEE storage space to reduce the occupation of the new hash value for the storage space corresponding to TEE.

[0090] The above-mentioned old hash value corresponds to the old version of disk image before update in the dedicated blockchain node device and the new hash value corresponds to the updated new version of disk image in the dedicated blockchain node device. The updated new hash value can be used to verify whether the new version of disk image in the dedicated blockchain node device is a trusted disk image provided by the provider so as to realize trusted execution of the new version of disk image and trusted startup of dedicated blockchain node device. For example, the dedicated blockchain node device can calculate the current hash value of the new version of disk image and send the current hash value to the cryptographic acceleration card for comparison with the new hash value in the cryptographic acceleration card: if a comparison result indicates that the current hash value is identical to the new hash value, it indicates that the new version of disk image deployed locally in the dedicated blockchain node device is the disk image provided by the provider, which is un-tampered and successfully deployed, and thus the new version of disk image is trusted. In this case, the dedicated blockchain node device can execute the new version of disk image to form a blockchain node. On the contrary, if the comparison result indicates that the current hash value is different from the new hash value, it indicates that the new version of disk image deployed locally in the dedicated blockchain node device is not the disk image provided by the provider but can be illegally-tampered or erroneously-deployed disk image and thus the new version of disk image is not trusted. At this time, the dedicated blockchain node device can refuse to execute the new version of disk image.

[0091] Since the cryptographic acceleration card is assembled in the dedicated blockchain

node device, when the cryptographic acceleration card sends the negotiation information to the provider and the provider sends the new hash value encrypted by the negotiated deployment key to the cryptographic acceleration card, the dedicated blockchain node device can forward the above-mentioned negotiation information and the new hash value in ciphertext status. FIG. 4 is an interaction flowchart of another hash updating method of a dedicated blockchain node device according to example embodiments of the present disclosure. In combination with FIG. 4, the process in which the old hash value stored in the cryptographic acceleration card is updated by forwarding data through the dedicated blockchain node device between the cryptographic acceleration card and the provider will be described below. As shown in FIG. 4, the process can include the following steps.

[0092] At step 402, the cryptographic acceleration card locally pre-stores a root-of-trust key.

[0093] At step 404, the provider of the new version of disk image calculates the new hash value in its own TEE according to the new version of disk image.

[0094] At step 406a, the cryptographic acceleration card sends negotiation information and a signature performed using the root-of-trust key to the dedicated blockchain node device.

[0095] At step 406b, the dedicated blockchain node device forwards the negotiation information and the signature performed using the root-of-trust key to the provider.

[0096] In the steps 406a-406b, the negotiation information and the signature performed using the root-of-trust key are forwarded through the dedicated blockchain node device to the provider, so that the cryptographic acceleration card and the provider are connected through the dedicated blockchain node device.

[0097] At step 408, the provider verifies the signature of the negotiation information.

[0098] At step 410 (i.e. 410a and 401b), the provider determines a deployment key according to the negotiation information.

[0099] At step 412a, the provider sends a new hash value encrypted by the deployment key to the dedicated blockchain node device.

[00100] At step 412b, the dedicated blockchain node device forwards the new hash value encrypted by the deployment key to the cryptographic acceleration card.

[00101] In the steps 412a-412b, the new hash value encrypted by the deployment key is forwarded through the dedicated blockchain node device to the cryptographic acceleration card, so that the cryptographic acceleration card and the provider are connected through the dedicated blockchain node device.

[00102] At step 414, the cryptographic acceleration card obtains the new hash value by performing decryption using its own deployment key and replaces a currently-stored old hash value with the new hash value.

[00103] The above-mentioned specific process of various steps in this embodiment is basically identical to step 314 of the embodiment shown in FIG. 3 and thus can referred to the previous descriptions and will not be repeated herein.

[00104] FIG. 5 is a structural schematic diagram of a cryptographic acceleration card according to example embodiments of the present disclosure. As shown in FIG. 5, from the hardware level, the device includes a processor 502, an internal bus 504, a network interface 506, a memory 508, a non-volatile memory 510, a crypto-operation unit 512 and a secure key zone 514. Of course, the device may further include hardware needed for other services. The crypto-operation unit 512 stores received or generated relevant keys in the secure key zone 514 so that the processor 502 invokes the keys to realize relevant functions such as encryption, decryption, signature obtaining and/or signature verification. The processor 502 reads corresponding computer programs from the non-volatile memory 510 to the memory 508 for running, so as to logically form a hash updating apparatus of a dedicated blockchain node device. Of course, in addition to the software implementation, one or more embodiments of the present disclosure do not preclude other implementations, for example, logic device or a combination of software and hardware or the like. That is, the executing subject of the following processing flow is not limited to each logic unit and can also be hardware or logic device.

[00105] As shown in FIG. 6, in a software implementation, a hash updating apparatus of a dedicated blockchain node device can include:

[00106] a negotiating module 601, configured to enable a cryptographic acceleration card assembled on the dedicated blockchain node device to send negotiation information to a provider of a new version of disk image, where the negotiation information is used respectively by the provider and the cryptographic acceleration card to determine a deployment key, and the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device;

[00107] a receiving module 602, configured to enable the cryptographic acceleration card to receive a new hash value encrypted by the provider using the deployment key, where the new hash value corresponds to the new version of disk image; and

[00108] a replacing module 603, configured to enable the cryptographic acceleration card to replace a stored old hash value corresponding to the old version of disk image with the new hash value, where the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[00109] Optionally, the disk image deployed in the dedicated blockchain node device

includes:

[00110] a binary disk image corresponding to a blockchain node and deployed in the dedicated blockchain node device; or,

[00111] a platform disk image deployed in the dedicated blockchain node device, wherein the platform disk image includes the binary disk image.

[00112] Optionally, the replacing module 603 is further configured to enable:

[00113] the cryptographic acceleration card to obtain the new hash value by performing decryption using the deployment key after receiving the new hash value encrypted using the deployment key;

[00114] the cryptographic acceleration card to replace the stored old hash value with the new hash value.

[00115] Optionally, the old hash value and the new hash value are calculated and generated in a trusted execution environment using the old version of disk image and new version of disk image provided by the provider, respectively.

[00116] Optionally, the negotiation information is signed by a root-of-trust key pre-stored in the cryptographic acceleration card, where a public key corresponding to the root-of-trust key is disclosed or the public key corresponding to the root-of-trust key is maintained only by the provider.

[00117] FIG. 7 is a structural schematic diagram of a dedicated blockchain node device according to example embodiments of the present disclosure. As shown in FIG. 7, from the hardware level, the device includes a processor 702, an internal bus 704, a network interface 706, a memory 708, and a non-volatile memory 710. Of course, the device may further include hardware needed for other services. The processor 702 reads corresponding computer programs from the non-volatile memory 710 to the memory 708 for running, so as to logically form a hash updating apparatus of a dedicated blockchain node device. Of course, in addition to the software implementation, one or more embodiments of the present disclosure do not preclude other implementations, for example, logic device or a combination of software and hardware or the like. That is, the executing subject of the following processing flow is not limited to each logic unit and can also be hardware or logic device.

[00118] As shown in FIG. 8, in a software implementation, a hash updating apparatus of a dedicated blockchain node device can include:

[00119] an information forwarding module 801, configured to, in response to negotiation information sent by a cryptographic acceleration card assembled on the dedicated blockchain node device, enable the dedicated blockchain node device to forward the negotiation information to a provider of a new version of disk image, such that the provider and the

cryptographic acceleration card respectively determine a deployment key based on the negotiation information; where the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device; and

[00120] a hash forwarding module 802, configured to enable the dedicated blockchain node device to forward a new hash value corresponding to the new version of disk image and encrypted by the provider using the deployment key to the cryptographic acceleration card, such that the cryptographic acceleration card replaces a stored old hash value corresponding to the old version of disk image with the new hash value, wherein the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

[00121] Optionally, the disk image deployed in the dedicated blockchain node device includes:

[00122] a binary disk image corresponding to a blockchain node and deployed in the dedicated blockchain node device; or,

[00123] a platform disk image deployed in the dedicated blockchain node device, where the platform disk image includes the binary disk image.

[00124] Optionally, the old hash value and the new hash value are calculated and generated in a trusted execution environment using the old version of disk image and new version of disk image provided by the provider, respectively.

[00125] Optionally, the negotiation information is signed by a root-of-trust key pre-stored in the cryptographic acceleration card, where a public key corresponding to the root-of-trust key is disclosed or the public key corresponding to the root-of-trust key is maintained by the provider.

[00126] The systems, apparatuses, modules or units described in the above-mentioned embodiments can be specifically implemented by a computer chip or an entity or can be implemented by a product with a particular function. A typical implementing device can be a computer and the computer can be specifically a personal computer, a laptop computer, a cellular phone, a camera phone, a smart phone, a personal digital assistant, a media player, a navigation device, an email transceiver, a game console, a tablet computer, a wearable device, or a combination of any several devices of the above-mentioned devices.

[00127] In a typical configuration, the computer includes one or more central processing units (CPU), one or more input/output interfaces, one or more network interfaces and one or more memories.

[00128] The memory may include a non-permanent memory, a random access memory

(RAM), and/or a non-volatile memory and the like in a computer readable medium, for example, read only memory (ROM), or flash RAM. The memory is one embodiment of the computer readable medium.

[00129] The computer readable medium includes transitory, non-transitory, portable and non-portable media, which can realize information storage by any method or technology. The information can be computer readable instructions, data structures, program modules and other data. The examples of the computer storage medium include but not limited to: a phase change random access memory (PRAM), a Static Random Access Memory (SRAM), a Dynamic Random Access Memory (DRAM), and other types of RAMs, Read-Only Memory (ROM), an Electrically-Erasable Programmable Read-Only Memory (EEPROM), a Flash Memory, or other memory technologies, CD-ROM, digital versatile disc (DVD) or other optical storage, cassette type magnetic tape, magnetic disk storage, quantum memory, storage medium based on graphene, or other magnetic storage device or other non-transmission medium for storing information accessible by computing devices. According to the definition of the specification, the computer readable medium does not include transitory computer readable media, for example, modulated data signal and carriers.

[00130] It should be noted that the term “including”, “containing” or any variation thereof is intended to encompass non-exclusive inclusion, so that a process, method, product or device including a series of elements includes not only those elements but also other elements not listed explicitly or those elements inherent to such a process, method, product or device. Without more limitations, an element defined by the statement “including a…” shall not be precluded to include additional same elements in a process, method, product or device including the elements.

[00131] The specific embodiments are described as above. Other embodiments be are within the scope of the appended claims. In some cases, the actions or steps recorded in the claims can be performed in a sequence different from the embodiments to achieve the desired result. Further, the processes shown in drawings do not necessarily require a particular sequence or a continuous sequence shown to achieve the desired result. In some embodiments, a multi-task processing and parallel processing are possible and can also be advantageous.

[00132] The terms used in one or more embodiments of the present disclosure are for the purpose of describing particular embodiments only, and are not intended to limit the one or more embodiments of the present disclosure. Terms “a”, “the” and “said” used in their singular forms in one or more embodiments of the present disclosure and the appended claims are also intended to include plurality, unless clearly indicated otherwise in the context. It should also be understood that the term “and/or” as used herein refers to and includes any and

all possible combinations of one or more of the associated listed items.

[00133] It should be understood that, although the terms "first," "second," "third," and the like may be used in one or more embodiments of the present disclosure to describe various information, such information should not be limited to these terms. These terms are only used to distinguish one category of information from another. For example, without departing from the scope of one or more embodiments of the present disclosure, first information can be referred as second information; and similarly, the second information can also be referred as the first information. Depending on the context, the term "if" as used herein can be interpreted as "when" or "upon" or "in response to determining".

[00134] The above-mentioned disclosure is merely illustrative of preferred embodiments of one or more embodiments of the present disclosure but not intended to limit the present disclosure, and any modifications, equivalent substitutions, adaptations thereof made within the spirit and principles of the disclosure shall be encompassed in the scope of protection of the present disclosure.

CLAIMS

1. A hash updating method of a dedicated blockchain node device, comprising:

sending, by a cryptographic acceleration card assembled on the dedicated blockchain node device, negotiation information to a provider of a new version of disk image, wherein the negotiation information is used respectively by the provider and the cryptographic acceleration card to determine a deployment key, and the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device;

receiving, by the cryptographic acceleration card, a new hash value encrypted by the provider using the deployment key, wherein the new hash value corresponds to the new version of disk image;

replacing, by the cryptographic acceleration card, a stored old hash value corresponding to the old version of disk image with the new hash value, wherein the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

2. The method according to claim 1, wherein the disk image deployed in the dedicated blockchain node device comprises:

a binary disk image corresponding to a blockchain node and deployed in the dedicated blockchain node device; or,

a platform disk image deployed in the dedicated blockchain node device, wherein the platform disk image comprises the binary disk image.

3. The method according to claim 1, wherein replacing, by the cryptographic acceleration card, the stored old hash value corresponding to the old version of disk image with the new hash value comprising:

after receiving the new hash value encrypted by the deployment key, obtaining, by the cryptographic acceleration card, the new hash value by decrypting using the deployment key;

replacing, by the cryptographic acceleration card, the stored old hash value with the new hash value.

4. The method according to claim 1, wherein the old hash value and the new hash value are calculated and generated in a trusted execution environment using the old version of disk image and the new version of disk image provided by the provider, respectively.

5. The method according to claim 1, wherein the negotiation information is signed by a root-of-trust key pre-stored in the cryptographic acceleration card, wherein a public key corresponding to the root-of-trust key is disclosed or the public key corresponding to the root-of-trust key is maintained only by the provider.

6. A hash updating method of a dedicated blockchain node device, comprising:

in response to negotiation information sent by a cryptographic acceleration card assembled on the dedicated blockchain node device, forwarding, by the dedicated blockchain node device, the negotiation information to a provider of a new version of disk image, such that the provider and the cryptographic acceleration card respectively determine a deployment key based on the negotiation information, wherein the new version of disk image is used to update an old version of disk image deployed in the dedicated blockchain node device;

forwarding, by the dedicated blockchain node device, a new hash value corresponding to the new version of disk image and encrypted by the provider using the deployment key to the cryptographic acceleration card, such that the cryptographic acceleration card replaces a stored old hash value corresponding to the old version of disk image with the new hash value, wherein the new hash value is used to be compared with a current hash value of a disk image deployed in the dedicated blockchain node device to determine whether the new version of disk image is deployed in the dedicated blockchain node device.

7. The method according to claim 6, wherein the disk image deployed in the dedicated blockchain node device comprises:

a binary disk image corresponding to a blockchain node and deployed in the dedicated blockchain node device; or,

a platform disk image deployed in the dedicated blockchain node device, wherein the platform disk image comprises the binary disk image.

8. The method according to claim 6, wherein the old hash value and the new hash value are calculated and generated in a trusted execution environment using the old version of disk image and the new version of disk image provided by the provider, respectively.

9. The method according to claim 6, wherein the negotiation information is signed by a root-of-trust key pre-stored in the cryptographic acceleration card, wherein a public key corresponding to the root-of-trust key is disclosed or the public key corresponding to the root-of-trust key is maintained only by the provider.

10. A cryptographic acceleration card, comprising:

a processor; and

a memory for storing processor executable instructions ;

wherein the processor implements a method according to any of claims 1-5 by running the executable instructions.

11. A dedicated blockchain node device, comprising:

a processor; and

a memory for storing processor executable instructions;

wherein the processor implements a method according to any of claims 6-9 by running the executable instructions.

12. A computer readable storage medium having computer instructions stored thereon, wherein the instructions are executed by a processor to implement steps in a method according to any of claims 1-9.