DEVICE AND METHOD FOR SELF-SOVEREIGN IDENTITY AUTHENTICATION

BACKGROUND OF THE INVENTION

1. Field of the Invention

[01] The present invention relates to a blockchain-based identity authentication technique, and more specifically, to a self-sovereign identity authentication device and a method thereof, which can update identity information using a registration blockchain and a revocation blockchain when self-sovereign identity authentication is performed based on a blockchain, and securely manage self-sovereign identity information by cryptographically splitting and storing the information.

2. Description of Related Art

[02] Generally, in a blockchain environment, a public key is used to generate a blockchain transaction address (e-wallet). Typically, a private key is managed by a user and stored in the form of a file in a user platform. For example, the user platform is a mobile terminal, a PC terminal device, or the like. Since storage or management of a private key for the blockchain transaction address (e-wallet) is entirely left to the user, an occasion incapable of using a transaction balance remaining in the blockchain transaction address (e-wallet) occurs due to loss or damage of the private key, and this technical limitation has been

solved using a medium such as a hardware wallet, a paper wallet, or the like.

[03] Meanwhile, new identity authentication techniques using blockchain-based self-sovereign identity (decentralized ID) are introduced, and according to the '2018 GDPR (Privacy Regulations of the European Union)', information for authenticating one's identity is based on a new identity authentication paradigm that is based on 'identity information under self-sovereignty'.

[04] As the blockchain is decentralized without relying on a central institution, authentication of self-sovereign identity works in a way the same as that of identity in real life. A user possesses all the rights for issuance, submission, deletion, and restoration of an identity certificate, and the blockchain proves that the identity certificate is owned by the user and used for record management such as issuance, submission, update, and the like of the identity certificate. That is, the self-sovereign identity authentication mechanism is a mechanism for generating identity information based on one's own information that can prove one's own identity, registering the generated identity information in the blockchain, generating an identity ID for inquiring the registered identity information, and using the ID to confirm the identity.

[05] However, as identity information according to identity ID is registered in the blockchain, there is a limitation in that

registered identity information may not be changed due to the technical characteristics of the blockchain. To overcome this technical limitation, the Sovrin foundation, bitcoin, Ethereum, and the like employ a method of storing procedure information such as system log or the like related to registration of identity information in the blockchain, without storing the identity information in the blockchain, and storing real data in a variable DB.

[06] (Patent Document 0001) 1. Korean Patent Publication No. 10-2019-0114435 "Blockchain-based authority authentication method, terminal, and server using the same" (published date: October 10, 2019)

SUMMARY OF THE INVENTION

[07] The present invention provides a self-sovereign identity authentication device and a method thereof for authenticating self-sovereign identity using integrity of a blockchain.

[08] The present invention provides a self-sovereign identity authentication device and a method thereof, which can change and update identity authentication information by managing information on registration and revocation of identity authentication in a blockchain.

[09] The present invention may generate and store several pieces of private split key data using a secret split algorithm

and restore a private key when split conditions are satisfied although part of the private key is lost or damaged.

[10] According to one aspect of the present invention, there is provided a self-sovereign identity authentication device.

[11] The self-sovereign identity authentication device according to an embodiment of the present invention includes a key management unit for managing a private key associated with a decentralized identifier, an identity information unit for generating and registering the decentralized identifier in a blockchain, and an authentication unit for verifying whether the decentralized identifier is valid.

[12] According to another aspect of the present invention, there is provided a self-sovereign identity authentication method and a computer program executing the same.

[13] A self-sovereign identity authentication method and a computer program executing the same according to an embodiment of the present invention may include the steps of receiving a request for inquiring information on a decentralized identifier, inquiring revocation information of the decentralized identifier, inquiring registration information of the decentralized identifier, and transmitting an inquiry result.

BRIEF DESCRIPTION OF THE DRAWINGS

[14] FIG. 1 is a view related to conventional self-sovereign identity authentication using a decentralized identifier on a blockchain.

[15] FIGS. 2 to 5 are views briefly describing a method of managing a private key by a self-sovereign identity authentication device according to an embodiment of the present invention.

[16] FIGS. 6 to 8 are views briefly describing a self-sovereign identity authentication device according to an embodiment of the present invention.

[17] FIGS. 9 to 12 are exemplary views showing management of a decentralized identifier by a self-sovereign identity authentication device according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[18] Since the present invention can be diversely modified and may have various embodiments, specific embodiments are illustrated in the drawings and will be described in detail through the detailed description. However, it should be understood that this is not intended to limit the present invention to the specific embodiments, but to include all modifications, equivalents, and substitutes included in the spirit and scope of the present invention. When it is determined in describing the present

invention that a detailed description of related known techniques may unnecessarily obscure the gist of the present invention, the detailed description will be omitted. In addition, singular expressions used in this specification and claims should be generally interpreted to mean "one or more" unless mentioned otherwise.

[19] Hereinafter, preferred embodiments of the present invention will be described in detail with reference to the accompanying drawings, and in describing with reference to the accompanying drawings, the same reference numerals are given to the same or corresponding components, and duplicated description thereof will be omitted.

[20] FIG. 1 is a view related to conventional self-sovereign identity authentication using a decentralized identifier on a blockchain.

[21] Referring to FIG. 1, a user generates and registers a decentralized identifier (DID) and a decentralized identifier document (DID Document) in a self-sovereign identity (SSI) environment.

[22] Describing in detail, a user generates an asymmetric key that uses a private key and a public key for proof of ownership of the decentralized identifier.

[23] The user securely stores the private key of the asymmetric key, and generates and stores a decentralized identifier

and a decentralized identifier document in the blockchain using the public key.

[24] The user transfers the decentralized identifier to an institution, and the institution may register the decentralized identifier or issue a requested certificate after verifying whether the user is the owner of the received decentralized identifier.

[25] Therefore, management of the asymmetric key pair is important in a self-sovereign identity (SSI) environment. This is since that there is no way to infer the relationship between the private key and the decentralized identifier due to the characteristics of the blockchain. When the private key is lost or changed, all previously signed proofs using the public key, as well as proof of identity, may be invalidated.

[26] The self-sovereign identity authentication device 10 may manage the private key to be securely stored in preparation for loss or damage of the user platform in which the private key is stored, loss or damage of the stored private key, or loss or damage caused by a human error.

[27] The self-sovereign identity authentication device 10 may effectively handle threats such as hacking or the like attacked through access of others, other than the owner, through a secret split algorithm.

[28] The self-sovereign identity authentication device 10 may determine whether it is a valid request for processing reissuance

or revocation of the private key or an attack by an illegal hacking attack.

[29] FIGS. 2 to 5 are views briefly describing a method of managing a private key by a self-sovereign identity authentication device according to an embodiment of the present invention.

[30] Referring to FIG. 2, the self-sovereign identity authentication device 10 includes a key management unit 100, an identity information unit 200, and an authentication unit 300.

[31] The key management unit 100 generates and manages a private key used when a decentralized identifier (DID) is generated.

[32] The key management unit 100 may distinguish a valid request for processing reissuance or revocation of a private key from an attack by an illegal hacking attack, not a valid request.

[33] Describing in detail with reference to FIG. 3, the key management unit 100 may perform dual management by dividing key management into two steps of a private key encryption and management step and a backup step.

[34] The key management unit 100 may encrypt the private key and store the encrypted private key. Describing in detail, the key management unit 100 splits the private key through a secret split algorithm and stores it in the user platform.

[35] The key management unit 100 may back up encrypted private key data in a backend server platform. For example, the backend server platform may be a hardware security module (HSM) storage.

[36] Referring to FIG. 4, the key management unit 100 includes a split unit 110, a restoration unit 120, and a backup unit 130.

[37] The split unit 110 may split a private key into a plurality of cryptographically secure private split key data using a secret split algorithm, and store them in a mobile device. The split unit 110 may set a split condition or a restoration condition, and may set a split password (credential). For example, the split unit 110 may set a split condition such as the number of split pieces of a private key, and may set a restoration condition such as the number of private split keys required for restoration. Since the split unit 110 utilizes a secret split algorithm, only the user knows the split condition or the restoration condition, and since the location of the user platform where the private key is stored or the location of the backup server cannot be known, it is possible to cope with threats such as hacking or the like through access of other people, such as a system manager/service manager or the like, not the user.

[38] The restoration unit 120 may restore a private key when the restoration condition is satisfied, although part of the private split key data is lost or damaged, according to the split condition or the restoration condition set by the split unit 110.

[39] The backup unit 130 may back up the private split key data in the backend server platform to additionally cope with loss

or damage of data. For example, the backend server platform is a hardware security module (HSM) storage.

[40] According to an embodiment of the present invention, the backup unit 330 may stores the private split key data in the user platform, and then selectively or randomly transmit as many private split key data as a predetermined number to the server to be stored in the server so that a backup/restoration step may be performed in preparation for the loss or damage of data.

[41] According to another embodiment of the present invention, the backup unit 330 may back up all the split and generated private split key data in the backend server platform.

[42] FIG. 5 is a view briefly describing a method of managing an asymmetric key by a self-sovereign identity authentication device according to an embodiment of the present invention. Although each process described below is a process performed by each functional unit constituting the self-sovereign identity authentication device in a step, for brief and clear explanation of the present invention, the subject of each step is commonly referred to as a self-sovereign identity authentication device.

[43] Referring to FIG. 5, the self-sovereign identity authentication device 10 uses an asymmetric key algorithm using a private key and a public key.

[44] The self-sovereign identity authentication device 10 may manage the private key in two steps of splitting and storing the

private key and backing up the private key in a backend server platform.

[45] The self-sovereign identity authentication device 10 generates an asymmetric key at step S510.

[46] At step S520, the self-sovereign identity authentication device 10 splits the generated private key using a secret split algorithm. The self-sovereign identity authentication device 10 may set a split condition such as the number of private split keys, or the like. The self-sovereign identity authentication device 10 may set a restoration condition such as the minimum number of private split keys when the private key is restored.

[47] At step S530, the self-sovereign identity authentication device 10 may store the private split key data in the user platform and then selectively or randomly transmit as many private split key data as a predetermined number to the server to be stored in the server. The self-sovereign identity authentication device 10 may store all the private split keys or only some of the private split keys according to the restoration condition. The self-sovereign identity authentication device 10 may restore the private key using the split key data stored in the backend server platform even when the user platform, as well as data of the private key, is lost or damaged.

[48] At step S540, the self-sovereign identity authentication device 10 generates a decentralized identifier (DID) using the public key.

[49] At step S550, the self-sovereign identity authentication device 10 generates a decentralized identifier document (DID Document) including the public key.

[50] At step S560, the self-sovereign identity authentication device 10 stores the decentralized identifier (DID) and the decentralized identifier document (DID Document) in the blockchain. Describing in detail, the self-sovereign identity authentication device 10 may register the decentralized identifier (DID) information in a registration blockchain.

[51] FIGS. 6 and 7 are views briefly describing a self-sovereign identity authentication device according to an embodiment of the present invention.

[52] Referring to FIG. 6, the self-sovereign identity authentication device 10 may simultaneously keep a registration blockchain and a revocation blockchain based on a blockchain.

[53] ① The issuer in FIG. 6 may be an institution that issues certificates.

[54] ② The holder in FIG. 6 may be a user who can generate and own a decentralized identifier and perform self-sovereign identity authentication.

[55] ③ The verifier in FIG. 6 may be an institution or a system that can provide identity authentication service by verifying a decentralized identifier.

[56] ④ The universal resolver in FIG. 6 may interconnect the functions of the entire system using APIs. The self-sovereign identity authentication device 10 may provide a DIF standard for storing decentralized identifiers in the blockchain and inquiring decentralized identifiers through the APIs provided by the universal resolver. The self-sovereign identity authentication device 10 may verify the decentralized identifiers stored in various blockchains through the universal resolver.

[57] The self-sovereign identity authentication device 10 requests registration of a decentralized identifier by transmitting decentralized identifier information generated by the user using the public key to the institution in response to a user's request.

[58] The self-sovereign identity authentication device 10 authenticates the user possessing the decentralized identifier, and registers the decentralized identifier in the blockchain through the universal resolver.

[59] At this point, the self-sovereign identity authentication device 10 first checks whether the decentralized identifier is registered in the revocation blockchain, and when the decentralized identifier is not registered in the revocation blockchain, it searches for the decentralized identifier in the

registration blockchain and stores the decentralized identifier therein.

[60] The self-sovereign identity authentication device 10 may store the decentralized identifier in the blockchain to maintain and use the technical characteristics of the blockchain as they are. The self-sovereign identity authentication device 10 has a 'double blockchain structure' of storing revocation information according to revocation of identity information in a blockchain dedicated to revocation.

[61] As the registration blockchain and the revocation blockchain are separately maintained, the self-sovereign identity authentication device 10 may reduce the response time, depend against forgery and falsification attacks by using the characteristic of the blockchain incapable of reversing or changing information that is once registered, and update information overcoming the characteristic of the blockchain incapable of changing or deleting registered information. In addition, as the registration and revocation procedures are separated, the self-sovereign identity authentication device 10 may easily defend against social engineering attacks such as a human error.

[62] The self-sovereign identity authentication device 10 may record and manage revocation of a decentralized identifier in the revocation blockchain according to the revocation condition of the decentralized identifier. The revocation condition of the

decentralized identifier may be a case in which the decentralized identifier is changed due to occurrence of a change in a registered decentralized identifier, an invalid decentralized identifier, expiration of a decentralized identifier, expiration of a private key used for generation of a decentralized identifier, exposure of a key, forgery and falsification attack on a key, or the like.

[63] Although the method of managing identity information using an existing database may easily cope with an attack technically since only a corresponding field value of the database needs to be changed in a situation of revoking identity information, the method may be vulnerable to attacks disguising as normal from the viewpoint that it can be changed by a third party such as a manager or the like. However, since the self-sovereign identity authentication device 10 manages revocation information in a separate blockchain, forgery and falsification are impossible, and the identity information can be updated.

[64] Referring to FIG. 2 again, the identity information unit 200 may generate a decentralized identifier (DID) on the basis of the user's own personal information that can prove the user's identity in response to a user's request.

[65] The identity information unit 200 manages revocation information of a decentralized identifier (DID) that cannot be used as an identity proof any longer as the identity information is updated. Describing in detail, since the identity information unit

200 may not delete or change a decentralized identifier (DID) that cannot be used as an identity proof any longer as the identity information is updated, the decentralized identifier (DID) for revocation may be registered in the blockchain. Since information once registered cannot be changed due to the technical characteristics of the blockchain, it has integrity. However, as an existing decentralized identifier (DID) should be deleted and a new decentralized identifier (DID) should be generated when the private key of the decentralized identifier (DID) is changed, the decentralized identifier (DID) information should be changed. Accordingly, the identity information unit 200 may register a decentralized identifier (DID) that cannot be used as it is updated, changed, or lost in the revocation blockchain and manage it as a revoked decentralized identifier.

[66] Referring to FIG. 7, the identity information unit 200 includes a registration management unit 210 and a revocation management unit 220.

[67] The registration management unit 210 manages registration information of a decentralized identifier (DID). Describing in detail, the registration management unit 210 registers a decentralized identifier (DID) in the registration blockchain.

[68] The revocation management unit 220 manages revocation information of a changed or no longer valid decentralized

identifier (DID). Describing in detail, the revocation management unit 220 registers changed or no longer valid decentralized identifiers (DID) in the revocation blockchain. Describing in detail, the revocation management unit 220 registers a decentralized identifier (DID) that may not prove identity any longer in the revocation blockchain including revocation information. When there is an authentication request using a decentralized identifier (DID) that cannot be used any longer, the revocation management unit 220 may confirm that the decentralized identifier (DID) is revoked information and transfer information indicating that the decentralized identifier (DID) is revoked.

[69] The authentication unit 300 may verify whether the decentralized identifier (DID) information is valid information.

[70] Referring to FIG. 8, when information on the generated decentralized identifier is updated due to various reasons, the self-sovereign identity authentication device 10 may register the decentralized identifier (DID) in the revocation blockchain and confirm that the information is not valid any longer.

[71] Since the self-sovereign identity authentication device 10 cannot delete or change a decentralized identifier (DID) that cannot be used as an identity proof any longer as identity information has been updated, it may receive a decentralized identifier revocation request at step S810.

[72] At step S820, the self-sovereign identity authentication device 10 registers information on the decentralized identifier (DID) requested to be revoked in the revocation blockchain.

[73] At step S830, the self-sovereign identity authentication device 10 generates a new decentralized identifier (DID) in response to a request.

[74] At step S840, the self-sovereign identity authentication device 10 may register the generated decentralized identifier (DID) in the registration blockchain.

[75] When a decentralized identifier authentication request is generated, the self-sovereign identity authentication device 10 may search the revocation blockchain and the registration blockchain to verify whether the decentralized identifier is a valid decentralized identifier (DID).

[76] FIGS. 9 to 12 are exemplary views showing management of a decentralized identifier (DID) by a self-sovereign identity authentication device according to an embodiment of the present invention.

[77] FIG. 9 is a sequence diagram illustrating an example of processing a decentralized identifier registration request by the self-sovereign identity authentication device 10.

[78] Referring to FIG. 9, the self-sovereign identity authentication device 10 receives a decentralized identifier registration request from a user or an institution at step S910.

[79] At step S920, the self-sovereign identity authentication device 10 inquires the decentralized identifier information from the revocation blockchain through the universal resolver.

[80] The universal resolver may be configured of APIs, a DID converter, blockchain (B/C) management, blockchain APIs, and the like.

[81] At step S930, the self-sovereign identity authentication device 10 inquires the decentralized identifier information from the registration blockchain through the universal resolver.

[82] When the decentralized identifier information is not registered in the revocation blockchain and also not registered in the registration blockchain, the self-sovereign identity authentication device 10 transmits an inquiry result informing that the decentralized identifier (DID) can be registered.

[83] When the decentralized identifier information is not registered in the revocation blockchain and is registered in the registration blockchain, the self-sovereign identity authentication device 10 may transmit an inquiry result informing that the decentralized identifier is an already registered valid decentralized identifier (DID).

[84] When the decentralized identifier information is registered in both the revocation blockchain and the registration blockchain, the self-sovereign identity authentication device 10 may transmit an inquiry result informing that the decentralized

identifier is an already revoked decentralized identifier (DID). The revoked decentralized identifier (DID) may not be used any longer.

[85] When the decentralized identifier can be registered as a result of the inquiry, the self-sovereign identity authentication device 10 may register the decentralized identifier (DID) in the registration blockchain at step S940.

[86] At step S950, the self-sovereign identity authentication device 10 registers the decentralized identifier (DID) in the registration blockchain and transmits a result.

[87] FIG. 10 is a sequence diagram illustrating an example of processing a decentralized identifier revocation request by the self-sovereign identity authentication device 10.

[88] Referring to FIG. 10, the self-sovereign identity authentication device 10 receives a decentralized identifier revocation request from a user or an institution at step S1010.

[89] At step S1020, the self-sovereign identity authentication device 10 inquires the decentralized identifier information from the revocation blockchain through the universal resolver.

[90] At step S1030, the self-sovereign identity authentication device 10 inquires the decentralized identifier information from the registration blockchain through the universal resolver.

[91] When the decentralized identifier information is not registered in the revocation blockchain and registered in the registration blockchain, the self-sovereign identity authentication device 10 may transmit an inquiry result informing that the decentralized identifier (DID) can be revoked.

[92] When the decentralized identifier information is not registered in the revocation blockchain and also not registered in the registration blockchain, the self-sovereign identity authentication device 10 transmits an inquiry result informing that the decentralized identifier is not a registered decentralized identifier (DID).

[93] When the decentralized identifier information is registered in both the revocation blockchain and the registration blockchain, the self-sovereign identity authentication device 10 may transmit an inquiry result informing that the decentralized identifier is an already revoked decentralized identifier (DID).

[94] When the decentralized identifier can be revoked as a result of the inquiry, the self-sovereign identity authentication device 10 may register the decentralized identifier (DID) in the revocation blockchain at step S1040.

[95] At step S1050, the self-sovereign identity authentication device 10 registers the decentralized identifier (DID) in the revocation blockchain and transmits a result.

[96] FIG. 11 is a sequence diagram illustrating an example of processing a decentralized identifier inquiry request by the self-sovereign identity authentication device 10.

[97] Referring to FIG. 11, the self-sovereign identity authentication device 10 receives a decentralized identifier inquiry request from a user or an institution at step S1110.

[98] At step S1120, the self-sovereign identity authentication device 10 inquires the decentralized identifier information from the revocation blockchain through the universal resolver.

[99] At step S1130, the self-sovereign identity authentication device 10 inquires the decentralized identifier information from the registration blockchain through the universal resolver.

[100] The self-sovereign identity authentication device 10 transmits a result of inquiring the decentralized identifier (DID) at step S1140.

[101] Describing in detail, when the decentralized identifier information is not registered in the revocation blockchain and is registered in the registration blockchain, the self-sovereign identity authentication device 10 may transmit an inquiry result informing that the decentralized identifier is a valid decentralized identifier (DID).

[102] When the decentralized identifier information is not registered in the revocation blockchain and also not registered in the registration blockchain, the self-sovereign identity authentication device 10 transmits an inquiry result informing that the decentralized identifier (DID) can be registered.

[103] When the decentralized identifier information is registered in both the revocation blockchain and the registration blockchain, the self-sovereign identity authentication device 10 may transmit an inquiry result informing that the decentralized identifier is an already revoked decentralized identifier (DID).

[104] FIG. 12 is a sequence diagram illustrating an example of processing a decentralized identifier verification request by the self-sovereign identity authentication device 10.

[105] Referring to FIG. 12, the self-sovereign identity authentication device 10 receives a request for verifying a decentralized identifier (DID) or a decentralized identifier document from a user or an institution at step S1210.

[106] At step S1220, the self-sovereign identity authentication device 10 inquires the decentralized identifier information from the revocation blockchain through the universal resolver.

[107] At step S1230, the self-sovereign identity authentication device 10 inquires the decentralized identifier

information or the decentralized identifier document from the registration blockchain through the universal resolver.

[108] When there is a valid decentralized identifier, the self-sovereign identity authentication device 10 transmits decentralized identifier information or decentralized identifier document information at step S1240.

[109] According to an embodiment of the present invention, as the technical characteristics of the blockchain can be maintained as they are when an identity is authenticated, forgery and falsification of information are impossible.

[110] According to an embodiment of the present invention, as the registration and revocation procedures of the self-sovereign identity authentication information are separated from each other, self-sovereign identity authentication information registered in the blockchain can be updated.

[111] According to an embodiment of the present invention, it is easy to defend against social engineering attacks such as a human error through secure split, restoration, and backup management of a private key, and the private key may be restored although the user platform in which the private key is stored is lost or damaged.

[112] According to an embodiment of the present invention, as restoration conditions/split conditions are set through a secret split algorithm for the sake of secure restoration of the private

key, it is possible to fundamentally respond to threats such as hacking or the like attacked through access of other people, such as a system manager/service manager or the like, not the owner of the private key.

[113] The self-sovereign identity authentication method described above may be implemented as a computer-readable code on a computer-readable medium. The computer-readable recording medium may be, for example, a transferable recording medium (CD, DVD, Blu-ray disk, USB storage device, transferable hard disk) or a fixed recording medium (ROM, RAM, computer-equipped hard disk). The computer program recorded on the computer-readable recording medium may be transmitted to another computing device through a network such as the Internet or the like and installed in another computing device, and therefore may be used in another computing device.

[114] Although all the components configuring the embodiments of the present invention are described as being combined in one piece or operating in combination, the technical spirit of the present invention is not necessarily limited to the embodiments. That is, within the scope of the objects of the present invention, all the components may operate by selectively combining one or more.

[115] Although the operations are shown in a particular order in the drawings, it should not be understood that the operations should be performed in a specific order or a sequential order as

shown in the drawings or all the operations shown in the drawings should be performed to obtain a desired result. In a specific situation, multitasking and parallel processing may be advantageous. Moreover, separation of the various components in the embodiments described above should not be construed as being necessarily required, and it should be understood that the described program components and systems may be generally integrated together into a single software product or packaged into a plurality of software products.

[116] Until now, the present invention has been described focusing on the embodiments thereof. Those skilled in the art will understand that the present invention can be implemented in a modified form without departing from the essential characteristics of the present invention. Therefore, the disclosed embodiments should be considered in an illustrative viewpoint rather than a restrictive viewpoint. The scope of the present invention is shown in the claims rather than the above description, and all differences within the scope equivalent thereto should be construed as being included in the present invention.

DESCRIPTION OF SYMBOLS

10: Self-sovereign identity authentication device

100: Key management unit

110: Split unit

120: Restoration unit

130: Backup unit

200: Identity information unit

210: Registration management unit

220: Revocation management unit

300: Authentication unit

WE CLAIM:

1. A self-sovereign identity authentication device comprising:

a key management unit for managing a private key associated with a decentralized identifier;

an identity information unit for generating and registering the decentralized identifier in a blockchain; and

an authentication unit for verifying whether the decentralized identifier is valid.

2. The device according to claim 1, wherein the identity information unit includes:

a registration management unit for managing registration information of the decentralized identifier; and

a revocation management unit for managing revocation information of the decentralized identifier.

3. The device according to claim 1, wherein integrity is guaranteed using the blockchain.

4. The device according to claim 1, wherein the key management unit includes:

a split unit for splitting a private key into a plurality of cryptographically secure private split key data using a secret split algorithm;

a restoration unit for restoring the private key according to a split condition or a restoration condition set in advance; and

a backup unit for backing up the private split key data in a backend server platform.

5. The device according to claim 4, wherein the backend server platform is a hardware security module storage.

6. A self-sovereign identity authentication method executed by a self-sovereign identity authentication device, the method comprising the steps of:

receiving a request for inquiring information on a decentralized identifier;

inquiring revocation information of the decentralized identifier;

inquiring registration information of the decentralized identifier; and

transmitting an inquiry result.

7. The method according to claim 6, wherein information on the decentralized identifier is updated by registering the registration

information and the revocation information of the decentralized identifier in blockchains, respectively.

8. A self-sovereign identity authentication method executed by a self-sovereign identity authentication device, the method comprising the steps of:

splitting a private key into a plurality of cryptographically secure private split key data using a secret split algorithm;

restoring the private key according to a split condition or a restoration condition set in advance; and

backing up the private split key data in a backend server platform.

9. A self-sovereign identity authentication device according to claim 8, wherein the backend server platform is a hardware security module storage.

10. A computer program for executing any one of the self-sovereign identity authentication methods of claims 6 to 9 and recorded in a computer-readable recording medium.