(19) **CONFÉDÉRATION SUISSE**
INSTITUT FÉDÉRAL DE LA PROPRIÉTÉ INTELLECTUELLE

(11) **CH 716 286 A2**

(51) Int. Cl.: ***G06F 21/62*** *(2013.01)*
H04L 9/06 *(2006.01)*
H04L 9/08 *(2006.01)*
G06K 9/00 *(2006.01)*
H04L 29/08 *(2006.01)*
G06F 21/57 *(2013.01)*

Demande de brevet pour la Suisse et le Liechtenstein
Traité sur les brevets, du 22 décembre 1978, entre la Suisse et le Liechtenstein

(12) **DEMANDE DE BREVET**

(21) Numéro de la demande: 00758/19

(22) Date de dépôt: 07.06.2019

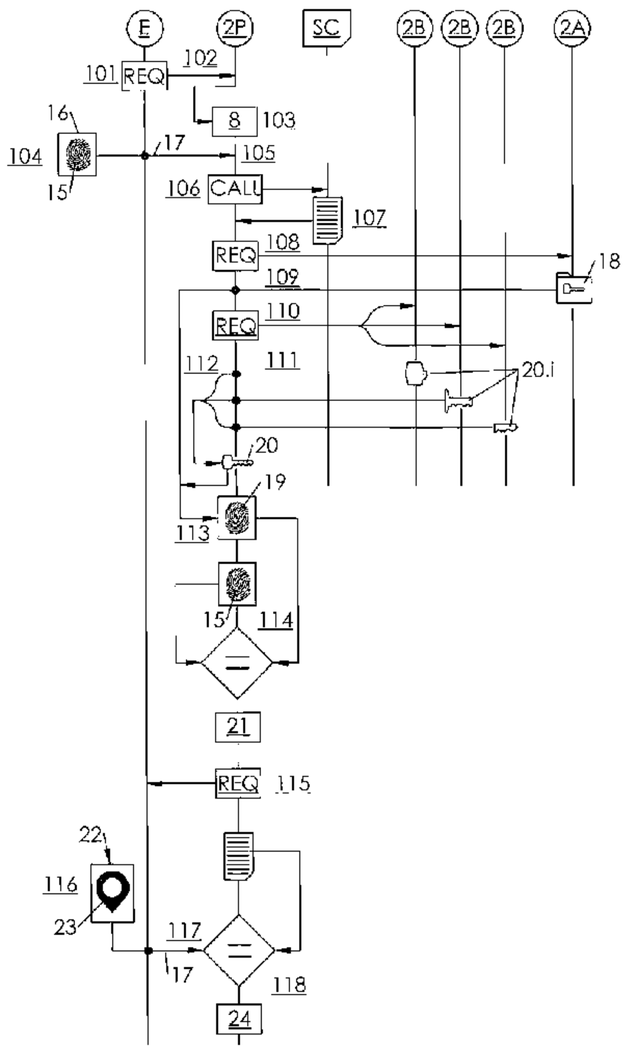
(43) Demande publiée: 15.12.2020

(71) Requérant:
Lapsechain SA C/O Leax Avocats,
Faubourg de l'Hôpital 18
2000 Neuchâtel (CH)

(72) Inventeur(s):
Jonathan Attia, 2000 Neuchâtel (CH)
Raphaël Louiset, 2072 Saint-Blaise (CH)

(54) Procédé de contrôle de données biométriques d'un individu sous condition de géolocalisation, avec inscription, dans une blockchain, d'un résultat d'analyse.

(57) L'invention concerne un procédé de contrôle des données biométriques d'un individu, à partir d'un émetteur (**E**) relié à un réseau pair-à-pair sur lequel est déployée une blockchain (**5**), ce procédé comprenant la comparaison de données (**15**) scannées au niveau de l'émetteur (**E**) avec des données (**19**) de référence contenues dans un conteneur (**18**) crypté mémorisé au préalable dans au moins un nœud (**2A**) du réseau ; après vérification d'une condition supplémentaire de géolocalisation, une trace du résultat de la comparaison et une trace du résultat de la vérification sont inscrites dans la blockchain (**5**).



Description

DOMAINE TECHNIQUE

[0001] L'invention a trait au domaine de l'informatique, et plus précisément au contrôle des données biométriques d'un individu.

ART ANTERIEUR

[0002] Les données biométriques des individus (typiquement une empreinte digitale ou de la paume, une empreinte rétinienne ou de l'iris, le réseau veineux de la main, une image du visage), tendent à se généraliser en tant que données d'authentification, notamment à l'usage des systèmes de contrôle d'accès à un environnement physique (par ex. local, coffre-fort) ou virtuel (typiquement une session sur un ordinateur, une tablette ou encore un smartphone).

[0003] Classiquement, des données biométriques de référence, capturées lors d'une session de paramétrage, sont stockées dans une base de données, puis, sur requête, sont appelées pour être comparées à des données biométriques instantanées, capturées localement à partir d'un terminal sur un individu souhaitant accéder à un environnement donné.

[0004] Lorsque plusieurs individus („utilisateurs“) sont présumés autorisés à accéder à un même environnement, une technique classique consiste à mémoriser, dans la base de données, autant de données biométriques que d'individus bénéficiant d'une autorisation d'accès.

[0005] Dans une première version, dite locale, la base de données est locale, c'est-à-dire qu'elle est stockée dans un espace mémoire équipant le (ou directement relié au) terminal à partir duquel est réalisée la capture. Cette technique peut paraître à l'abri des intrusions (et donc des usurpations d'identité) en raison de son caractère local, mais il est le plus souvent nécessaire de prévoir un accès réseau à la base de données, aux fins d'administration (y compris l'ajout ou le retrait d'utilisateurs. Il en résulte que la base de données peut être piratée.

[0006] Dans une deuxième version, dite délocalisée, la base de données est stockée dans un espace mémoire réservé dans un serveur distant auquel, à chaque sollicitation par un utilisateur, le terminal se connecte pour comparer les données biométriques capturées aux données biométriques de référence.

[0007] Cette deuxième version présente l'avantage de permettre une administration à distance des autorisations associées aux utilisateurs. Cependant, elle repose, d'une part, sur la politique de sécurité informatique à laquelle est soumis le serveur distant sur lequel est stockée la base de données ; d'autre part, sur la confiance que l'on peut accorder à l'administrateur dudit serveur.

[0008] Il existe par ailleurs de nombreux services pour lesquels les données biométriques des individus sont stockées sur des serveurs distants pour être utilisées en tant que moyen d'authentification sur des comptes utilisateurs.

[0009] Dans tous les cas, il n'est jamais certain que les données biométriques des utilisateurs soient protégées des intrusions, de la copie, d'une éventuelle exploitation commerciale, ou encore de l'effacement.

[0010] Il existe par conséquent un besoin d'améliorer la confidentialité avec laquelle sont traitées les données biométriques des individus.

RESUME DE L'INVENTION

[0011] Il est proposé un procédé de contrôle de données biométriques d'un individu, à partir d'une unité de traitement informatique, dite émetteur, reliée ou intégrée à un réseau pair-à-pair composé d'une pluralité de nœuds formant une base de données distribuée sur laquelle est mémorisée, par réplication sur chaque nœud, une chaîne de blocs, ce procédé comprenant les opérations suivantes :

- Etablir une session de communication entre l'émetteur et le réseau ;
- Sélectionner parmi le réseau un nœud dit de calcul, équipé d'une unité de traitement dans laquelle est implémenté un environnement d'exécution sécurisé par cryptographie, dit enclave ;
- Instancier l'enclave ;
- Au moyen d'un dispositif de capture équipant ou relié à l'émetteur, réaliser une capture de données biométriques de l'individu au niveau d'un membre ou d'un organe de cet individu ;
- Charger dans l'enclave, via une ligne de communication sécurisée, les données biométriques capturées ;
- Au moyen d'une balise de géolocalisation équipant ou reliée à l'émetteur, éditer des données instantanées de géolocalisation ;
- Charger dans l'enclave, via une ligne de communication sécurisée, les données instantanées de géolocalisation ;
- Sélectionner parmi le réseau un nœud de stockage sur lequel est mémorisé un conteneur crypté contenant des données biométriques de référence ;
- Charger le conteneur crypté dans l'enclave ;
- Déchiffrer dans l'enclave les données biométriques de référence, au moyen d'une clé de déchiffrement associée au conteneur crypté ;

– Dans l'enclave, comparer les données biométriques capturées et les données de référence, et vérifier si les données instantanées de géolocalisation satisfont une condition de géolocalisation prédéfinie ;
– Inscrire dans un bloc de la chaîne de blocs une transaction comprenant une trace du résultat de la comparaison, et une trace du résultat de la vérification.

[0012] Selon un mode préféré de réalisation, sont prévues les opérations suivantes :

- A partir de l'enclave du nœud de calcul, sélectionner parmi le réseau des nœuds sur lesquels sont stockés des fragments de la clé cryptographique de déchiffrement associée au conteneur ;

- Charger lesdits fragments dans l'enclave du nœud de calcul ;

- Dans l'enclave :

o Reconstituer la clé cryptographique de déchiffrement associée au conteneur, à partir des fragments ainsi chargés ;

o Déchiffrer les données biométriques du conteneur au moyen de la clé ainsi reconstituée.

BREVE DESCRIPTION DES FIGURES

[0013] D'autres objets et avantages de l'invention apparaîtront à la lumière de la description d'un mode de réalisation, faite ci-après en référence aux dessins annexés dans lesquels :

La **FIG.1** est un schéma fonctionnel simplifié illustrant un réseau pair-à-pair sur lequel est distribuée une chaîne de blocs ;

La **FIG.2** est un schéma fonctionnel simplifié illustrant différents composants d'une unité de traitement informatique impliqués dans la création et l'exploitation d'un environnement d'exécution sécurisé appelé enclave ;

La **FIG.3** est un schéma fonctionnel illustrant un procédé de contrôle des données biométriques d'un individu ;

La **FIG.4** est un diagramme fonctionnel prolongeant le diagramme de la **FIG.3** et illustrant des opérations d'inscription du résultat du contrôle dans la blockchain.

DESCRIPTION DETAILLEE DE L'INVENTION

[0014] Le procédé proposé vise à contrôler, de manière confidentielle, des données biométriques d'un individu.

[0015] Sans s'y restreindre, le procédé de contrôle proposé exploite, en les combinant, des fonctionnalités offertes par deux technologies relativement récentes dont il paraît utile de faire une description préalable avant d'entrer dans les détails du procédé, à savoir :

– La technologie de la chaîne de blocs ou, en terminologie anglo-saxonne, blockchain (dans ce qui suit, on préférera la terminologie anglo-saxonne, en raison de son emploi courant dans la plupart des langues, y compris en langue française) ;
– La technologie de l'environnement d'exécution sécurisé ou, en terminologie anglo-saxonne, du trusted execution environment (TEE).

[0016] La technologie blockchain est organisée en couches. Elle comprend :

– Une couche d'infrastructure matérielle, appelée „réseau blockchain“ ;
– Une couche protocolaire appelée „protocole blockchain“ ;
– Une couche informationnelle, appelée „registre blockchain“.

[0017] Le réseau blockchain est un réseau informatique décentralisé, dit réseau pair-à-pair (en terminologie anglo-saxonne Peer-to-Peer ou P2P), constitué d'une pluralité d'ordinateurs (au sens fonctionnel du terme : il s'agit d'un appareil pourvu d'une unité de traitement informatique programmable, qui peut se présenter sous forme d'un smartphone, d'une tablette, d'un ordinateur de bureau, d'une station de travail, d'un serveur physique ou virtuel, c'est-à-dire un espace de calcul et de mémoire alloué au sein d'un serveur physique et sur lequel tourne un système d'exploitation ou une émulation de système d'exploitation), appelés „nœuds“ en référence à la théorie des graphes, capables de communiquer entre eux (c'est-à-dire de s'échanger des données informatiques), deux à deux, au moyen de liaisons filaires ou sans fil.

[0018] Un réseau **1** blockchain comprenant des nœuds **2** communiquant par des liaisons **3** est illustré sur la **FIG.1**. Par souci de simplification et de conformité à la théorie des graphes, sur la **FIG.1**, les nœuds **2** du réseau **1** sont représentés par des cercles ; les liaisons **3**, par des arêtes reliant les cercles. Pour ne pas surcharger de traits le dessin, seules certaines liaisons **3** entre les nœuds **2** sont représentées.

[0019] Les nœuds **2** peuvent être disséminés sur de larges régions géographiques ; ils peuvent également être regroupés dans des régions géographiques plus restreintes.

[0020] Le protocole blockchain se présente sous forme d'un programme informatique implémenté dans chaque nœud **2** du réseau **1** blockchain, et qui inclut, outre des fonctions de dialogue - c'est-à-dire d'échange des données informatiques - avec les autres nœuds **2** du réseau **1**, un algorithme de calcul qui, à partir de données d'entrée appelées „transactions“ (qui sont des transcriptions d'interactions entre un ou plusieurs terminaux informatiques émetteurs et un ou plusieurs terminaux informatiques destinataires) :

- Élabore des fichiers **4** de données structurées appelés „blocs“, chaque bloc **4** comprenant un corps **4A** contenant des empreintes numériques de transactions, et un en-tête **4B** contenant :

o Un numéro d'ordre, ou rang, ou encore hauteur (height en anglais), sous forme d'un nombre entier qui désigne la position du bloc **4** au sein d'une chaîne dans l'ordre croissant à partir d'un bloc initial (Genesis block en anglais) ;

o Une empreinte numérique unique des données du corps **4A** ;

o Une empreinte numérique unique, appelée pointeur, de l'en-tête du bloc **4** précédent,

o Une donnée d'horodatage (timestamp en anglais) ;

- Met en œuvre un mécanisme de validation des blocs **4** par consensus entre tout ou partie des nœuds **2** ;

- Concatène les blocs **4** validés pour former un registre **5** (le registre blockchain) sous forme d'un agrégat dans lequel chaque bloc **4** est relié mathématiquement au précédent par son pointeur.

[0021] La moindre modification des données du corps **4A** ou de l'en-tête **4B** d'un bloc **4** affecte la valeur de son empreinte numérique et rompt par conséquent le lien existant entre ce bloc **4** ainsi modifié et le bloc **4** suivant dont le pointeur ne correspond plus.

[0022] Selon un mode particulier de réalisation, l'empreinte numérique de chaque bloc **4** est un condensé (ou condensat, en anglais hash) des données du bloc **4**, c'est-à-dire le résultat d'une fonction de hachage appliquée aux données du bloc **4** (y compris le corps **4A** et l'en-tête **4B** à l'exception de l'empreinte numérique elle-même). La fonction de hachage est typiquement SHA-256.

[0023] Pour un bloc **4** donné de rang N (N un entier), le pointeur assure avec le bloc **4** précédent de rang N-1 une liaison inaltérable. En effet, toute modification des données du bloc **4** de rang N-1 aboutirait à la modification de son empreinte, et donc à un défaut de correspondance entre cette empreinte (modifiée) du bloc **4** de rang N-1 et le pointeur mémorisé parmi les métadonnées du bloc **4** de rang N.

[0024] La succession des blocs **4** reliés entre eux deux à deux par correspondance du pointeur d'un bloc **4** donné de rang N avec l'empreinte numérique du bloc précédent de rang N-1 constitue par conséquent le registre **5** blockchain sous forme d'un agrégat de blocs **4** corrélés, dans lequel la moindre modification des données d'un bloc **4** de rang N-1 se traduit par une rupture du lien avec le bloc **4** suivant de rang N - et donc la rupture du registre blockchain.

[0025] C'est cette structure particulière qui procure aux données contenues dans le registre **5** blockchain une réputation d'immuabilité, garantie par le fait que le registre **5** blockchain est répliqué sur tous les nœuds **2** du réseau **1**, obligeant tout attaquant, non seulement à modifier tous les blocs **4** de rang supérieur au bloc **4** modifié, mais à déployer ces modifications (alors même que le registre **5** blockchain continue de se constituer par les nœuds **2** appliquant le protocole blockchain) à l'ensemble des nœuds **2**.

[0026] Quel que soit le type de consensus appliqué par le mécanisme de validation des blocs **4**, la plupart des technologies blockchain ont pour fonction primaire d'enregistrer, dans leur registre **5** blockchain, des transactions passées entre un ou plusieurs terminaux émetteurs, et un ou plusieurs terminaux récepteurs, indifféremment appelés „utilisateurs“.

[0027] A chaque utilisateur est associé un compte, appelé de manière simplificatrice „portefeuille électronique“ (en anglais digital wallet), qui contient une zone mémoire et une interface programmatique ayant des fonctions d'interaction avec le réseau **1** blockchain pour lui soumettre des transactions, et des fonctions de synchronisation avec le registre **5** blockchain pour inscrire, dans la zone mémoire, les transactions validées par inscription dans le registre **5** blockchain.

[0028] Sauf mention contraire, et par souci de simplification, l'expression simple „chaîne de blocs“ ou „blockchain“ désigne le registre **5** blockchain lui-même.

[0029] Certaines technologies blockchain récentes (Ethereum, typiquement) ajoutent aux trois couches matérielle (réseau blockchain), protocolaire (protocole blockchain) et informationnelle (registre blockchain) une couche applicative qui se présente sous forme d'un environnement de développement permettant de programmer des applications, appelées „contrats intelligents“ (en anglais Smart contracts), qui peuvent être déployées sur le registre **5** blockchain à partir des nœuds **2**.

[0030] On décrit à présent succinctement la technologie des contrats intelligents.

[0031] Un contrat intelligent comprend deux éléments :

– Un compte, appelé „compte de contrat“ (en anglais Contract account), dans la zone mémoire duquel est inscrit un code source contenant des instructions informatiques implémentant les fonctions attribuées au contrat intelligent ;
– Un code exécutable (en anglais Executable Bytecode) résultant d'une compilation du code source, ce code exécutable étant mémorisé ou déployé au sein du registre **5** blockchain, c'est-à-dire inséré en tant que transaction dans un bloc **4** du registre **5** blockchain.

[0032] Dans la technologie blockchain proposée par Ethereum, un smart contrat est activé par un appel (en anglais Call) adressé par un autre compte, dit compte initiateur (qui peut être un compte utilisateur ou un compte de contrat), cet appel se présentant sous forme d'une transaction contenant, d'une part, un fonds de réserve à transférer (c'est-à-dire un paiement) depuis le compte initiateur au compte de contrat et, d'autre part, des conditions initiales.

[0033] Cet appel est inscrit en tant que transaction dans le registre **5** blockchain. Il déclenche :

– Le transfert du fonds de réserve du compte initiateur au compte de contrat ;
– La désignation, parmi le réseau **1** blockchain, d'un nœud d'exécution associé à un compte utilisateur ;
– L'activation, dans une unité de traitement informatique du nœud d'exécution, d'un environnement d'exécution ou machine virtuelle (appelé Ethereum Virtual Machine ou EVM dans le cas d'Ethereum) ;
– L'exécution pas-à-pas des étapes de calcul du code exécutable par la machine virtuelle à partir des conditions initiales, chaque étape de calcul étant accompagnée d'un transfert d'une fraction (appelée gas dans le cas d'Ethereum) du fonds de réserve depuis le compte de contrat vers le compte utilisateur du nœud d'exécution, et ce jusqu'à épuisement des étapes de calcul, au terme desquelles est obtenu un résultat ;
– L'inscription (éventuellement sous forme d'une empreinte numérique) de ce résultat en tant que transaction dans le registre **5** blockchain.

[0034] Le compte initiateur récupère (c'est-à-dire, en pratique, télécharge) le résultat lors de sa synchronisation au registre **5** blockchain.

[0035] On introduit à présent brièvement les environnements d'exécution sécurisé.

[0036] Un environnement d'exécution sécurisé (Trusted execution environment ou TEE) est, au sein d'une unité **6** de traitement informatique pourvue d'un processeur ou CPU (Central Processing Unit) **7**, un espace temporaire de calcul et de stockage de données, appelé (par convention) enclave, ou encore enclave cryptographique, qui se trouve isolé, par des moyens cryptographiques, de toute action non autorisée résultant de l'exécution d'une application hors de cet espace, typiquement du système d'exploitation.

[0037] Intel® a, par exemple, revu à partir de 2013 la structure et les interfaces de ses processeurs pour y inclure des fonctions d'enclave, sous la dénomination Software Guard Extension, plus connue sous l'acronyme SGX. SGX équipe la plupart des processeurs de type XX86 commercialisés par Intel® depuis 2015, et plus précisément à partir de la sixième génération incorporant la microarchitecture dite Skylake. Les fonctions d'enclave proposées par SGX ne sont pas accessibles d'office : il convient de les activer via le système élémentaire d'entrée/sortie (Basic Input Output System ou BIOS).

[0038] Il n'entre pas dans les nécessités de la présente description de détailler l'architecture des enclaves, dans la mesure où :

– En dépit de sa relative jeunesse, cette architecture est relativement bien documentée, notamment par Intel® qui a déposé de nombreux brevets, cf. par ex., parmi les plus récents, la demande de brevet américain US 2019/0058696 ;
– Des processeurs permettant de les implémenter sont disponibles sur le marché - notamment les processeur Intel® précités ;
– Seules les fonctionnalités permises par l'enclave nous intéressent ici, ces fonctionnalités pouvant être mises en œuvre via des lignes de commande spécifiques. A ce titre, l'homme du métier pourra se référer au guide édité en 2016 par Intel® : Software Guard Extensions, Developer Guide.

[0039] Pour une description plus accessible des enclaves, et plus particulièrement d'Intel® SGX, l'homme du métier peut également se référer à A. Adamski, Overview of Intel SGX - Part 1, SGX Internal, ou à D. Boneh, Surnaming Schemes, Fast Verification, and Applications to SGX Technology, in Topics in Cryptology, CT - RSA 2017, The Cryptographers' Track at the RSA Conférence 2017, San Francisco, CA, USA, Feb.14-17, 2017, Proceedings, pp.149-164, ou encore à K. Severinsen, Secure Programming with Intel SGX and Novel Applications, Thesis submitted for the Degree of Master in Programming and Networks, Dept. Of Informatics, Faculty of Mathematics and Natural Science, University of Oslo, Autumn 2017.

[0040] Pour résumer, en référence à la **FIG.2**, une enclave **8** comprend, en premier lieu, une zone **9** mémoire sécurisée (dénommée Page Cache d'enclave, en anglais Enclave Page Cache ou EPC), qui contient du code et des données relatives à l'enclave elle-même, et dont le contenu est chiffré et déchiffré en temps réel par une puce dédiée dénommée Moteur de Chiffrement de Mémoire (en anglais Memory Encryption Engine ou MEE). L'EPC **9** est implémentée au sein d'une partie de la mémoire vive dynamique (DRAM) **10** allouée au processeur **7**, et à laquelle les applications ordinaires (notamment le système d'exploitation) n'ont pas accès.

[0041] L'enclave **8** comprend, en deuxième lieu, des clés cryptographiques employées pour chiffrer ou signer à la volée les données sortant de l'EPC **9**, ce grâce à quoi l'enclave **8** peut être identifiée (notamment par d'autres enclaves), et les

données qu'elle génère peuvent être chiffrées pour être stockées dans des zones de mémoire non protégées (c'est-à-dire hors de l'EPC **9**).

[0042] Pour pouvoir exploiter une telle enclave **8**, une application **11** doit être segmentée en, d'une part, une ou plusieurs parties **12** non sécurisées (en anglais untrusted part(s), et, d'autre part, une ou plusieurs parties **13** sécurisées (en anglais trusted part(s)).

[0043] Seuls les processus induits par la (les) partie(s) **13** sécurisée(s) de l'application **11** peuvent accéder à l'enclave **8**. Les processus induits par la (les) partie(s) **12** non sécurisée(s) ne peuvent pas accéder à l'enclave **8**, c'est-à-dire qu'ils ne peuvent pas dialoguer avec les processus induits par la (les) partie(s) **13** sécurisée(s).

[0044] La création (également dénommée instanciation) de l'enclave **8** et le déroulement de processus en son sein sont commandés via un jeu **14** d'instructions particulières exécutables par le processeur **7** et appelées par la (les) partie(s) **13** sécurisée(s) de l'application **11**.

[0045] Parmi ces instructions :

- ECREATE commande la création d'une enclave **8** ;
- EINIT commande l'initialisation de l'enclave **8** ;
- EADD commande le chargement de code dans l'enclave **8** ;
- EENTER commande l'exécution de code dans l'enclave **8** ;
- ERESUME commande une nouvelle exécution de code dans l'enclave **8** ;
- EEXIT commande la sortie de l'enclave **8**, typiquement à la fin d'un processus exécuté dans l'enclave **8**.

[0046] On a, sur la **FIG.2**, représenté de manière fonctionnelle l'enclave **8** sous la forme d'un bloc (en traits pointillés) englobant la partie **13** sécurisée de l'application **11**, le jeu **14** d'instructions du processeur **7**, et l'EPC **9**. Cette représentation n'est pas réaliste ; elle vise simplement à regrouper visuellement les éléments qui composent ou exploitent l'enclave **8**.

[0047] Nous expliquerons ci-après comment sont exploitées les enclaves.

[0048] Le procédé proposé vise à effectuer un contrôle de données **15** biométriques d'un individu personne physique, ci-après simplement dénommé utilisateur, à partir d'une unité **E** de traitement informatique (équipant par ex. un ordinateur personnel fixe ou portable, une tablette, un smartphone, ou encore un dispositif de contrôle d'accès), reliée ou intégrée au réseau **1** blockchain et appelée émetteur.

[0049] Le contrôle des données **15** biométriques est typiquement effectué aux fins de déverrouiller, pour l'utilisateur, un accès à un environnement physique (un logement, une pièce, un coffre-fort) ou virtuel (un environnement de travail dans un ordinateur, une tablette ou un smartphone).

[0050] A cet effet, l'émetteur **E** est équipé de (ou relié à) un dispositif **16** de capture biométrique ou scanneur. Le scanneur **16** est configuré pour réaliser une capture des données **15** biométriques de l'utilisateur au niveau d'un membre (par ex. un ou plusieurs doigt(s), une main) ou d'un organe (par ex. un oeil, le visage, une oreille, une partie du réseau veineux) de cet utilisateur.

[0051] Comme nous le verrons, ce contrôle est effectué de manière décentralisée, sur le réseau **1**.

[0052] Une première opération **101** consiste à établir, sur requête (**REQ**), une session de communication entre l'émetteur **E** et le réseau **1**.

[0053] Cette session est activée à partir de l'émetteur **E**, par exemple à la suite d'une action prédéterminée comme l'appui sur un bouton ou la détection (par ex. au moyen d'un capteur de proximité) de l'approche du membre ou de l'organe de l'utilisateur.

[0054] Une deuxième opération **102** consiste, parmi le réseau **1**, à sélectionner un nœud **2P** de calcul, équipé d'une unité de traitement dans laquelle est implémentée une enclave **8**.

[0055] Une troisième opération **103** consiste, au sein du nœud **2P** de calcul, à instancier l'enclave **8**.

[0056] Une quatrième opération **104** consiste, au niveau de l'émetteur **E**, à réaliser, au moyen de son scanneur **16**, une capture des données **15** biométriques de l'individu. Dans l'exemple illustré, ces données **15** biométriques sont issues d'une empreinte digitale.

[0057] Une cinquième opération **105** consiste, à partir de l'émetteur **E**, à charger, dans l'enclave **8** du nœud **2P** de calcul, via une ligne **17** de communication sécurisée (par ex. utilisant le protocole Transport Layer Security ou TLS), les données **15** biométriques ainsi capturées.

[0058] Selon un mode préféré de réalisation, le contrôle des données **15**, conduit au sein de l'enclave **8**, est effectué suivant les instructions d'un contrat intelligent **SC** déployé sur la blockchain **5**.

[0059] Comme illustré sur la **FIG.3**, une sixième opération **106** consiste, pour l'enclave **8**, à appeler (**CALL**) le contrat intelligent **SC**.

[0060] Une septième opération **107** consiste, en retour, à charger le code du contrat intelligent **SC** dans l'enclave **8** pour exécution. Est également chargée dans l'enclave **8** une machine virtuelle (EVM lorsque le contrat intelligent **SC** est programmé selon les spécifications d'Ethereum) destinée à permettre l'exécution du code du contrat intelligent **SC**.

[0061] Une huitième opération **108** consiste, pour l'enclave **8**, à sélectionner parmi le réseau **1** un nœud **2A** de stockage sur lequel est stocké un conteneur **18** crypté contenant des données **19** biométriques de référence, et à transmettre à ce nœud **2B** de stockage une requête (**REQ**) de communication de ce conteneur **19**.

[0062] Pour faciliter cette sélection, le conteneur **18** crypté peut être couplé à un identifiant associé à l'émetteur **E** (transmis à l'enclave **8** avec les données **15** biométriques scannées), ou à l'utilisateur (et qui peut être saisi par celui-ci sur une interface équipant ou reliée à l'émetteur **E**).

[0063] Une neuvième opération **109** consiste, à partir du nœud **2B** de stockage, à charger dans l'enclave **8** le conteneur **18** crypté.

[0064] Le déchiffrement des données **19** biométriques de référence du premier conteneur **18** crypté nécessite une clé **20** cryptographique de déchiffrement, stockée sur le réseau **1**.

[0065] Selon un mode préféré de réalisation, et comme illustré sur la **FIG.3**, la clé **20** est distribuée sur le réseau **1** en application des règles de Shamir (dites du Partage de Secret de Shamir, en anglais Shamir's Secret Sharing).

[0066] Plus précisément, selon les règles de Shamir, la clé **20** a, auparavant, été fragmentée en un ensemble de N fragments **20.i** (N un entier prédéterminé, $N>3$, i un indice entier, $1\leq i\leq N$), tel qu'un sousensemble de K fragments **20.i** (K un entier prédéterminé, $1<K<N$, avec $1\leq i\leq K$) est suffisant pour reconstituer la clé **20**, chaque fragment **20.i** étant stocké séparément dans un nœud **2C** de stockage du réseau **1**.

[0067] Une dixième opération **110** consiste par conséquent, pour l'enclave **8**, à sélectionner parmi le réseau **1** K nœuds **2B** de stockage sur lesquels sont stockés K fragments **20.i** respectifs, et à transmettre à chaque nœud **2B** de stockage une requête (**REQ**) de communication de son fragment **20.i**.

[0068] Comme illustré sur la **FIG.3**, une onzième opération **111** consiste, à partir des nœuds **2B** de stockage ainsi sélectionnés, à charger dans l'enclave **8** les K fragments **20.i**.

[0069] Une douzième opération **112** consiste, pour l'enclave **8**, à reconstituer la clé **20** à partir des K fragments **20.i** ainsi chargés.

[0070] Une treizième opération **113** consiste, pour l'enclave **8**, à déchiffrer les données **19** biométriques de référence du conteneur **18** en lui appliquant la clé **20** ainsi reconstituée.

[0071] Une quatorzième opération **114** consiste, pour l'enclave **8**, à comparer les données **15** biométriques issues de la capture effectuée par le scanneur **16** et les données **19** biométriques de référence ainsi déchiffrées. La comparaison peut être effectuée par une technique classique (typiquement par mesure des distances entre minuties dans le cas de l'empreinte digitale).

[0072] Si cette comparaison est un échec, les données **15**, **19** sont décrétées ne pas correspondre, et l'action pour laquelle le contrôle des données biométriques de l'utilisateur était requis (typiquement un déverrouillage) n'est pas autorisée. L'émetteur **E** en est informé. Les opérations de capture et de comparaison des données biométriques peuvent cependant être répétées, pour minimiser le risque de faux négatif.

[0073] Si au contraire la comparaison est un succès, les données **15**, **19** sont décrétées correspondre. La comparaison produit un résultat **21** (sous forme, par ex., d'un bit de valeur 0 en cas d'échec, et de valeur **1** en cas de succès).

[0074] Cependant la seule authentification biométrique de l'individu derrière l'émetteur **E** est décrétée insuffisante pour autoriser l'action requise : une condition supplémentaire de géolocalisation, prédéfinie, doit en effet être vérifiée. Cette condition est par exemple inscrite parmi les instructions du contrat intelligent **SC**. Cette condition spécifie par ex. une zone géographique prédéfinie, au sein de laquelle l'émetteur **E** est supposé se trouver (ou au contraire une ou plusieurs zones géographiques hors desquelles l'émetteur **E** est supposé se trouver).

[0075] A cet effet, l'émetteur **E** est équipé de (ou relié à) une balise **22** de géolocalisation. Cette balise **22** est configurée pour éditer des données **23** instantanées de géolocalisation, typiquement sous forme d'un tuple de coordonnées géographiques incluant latitude et longitude, et le cas échéant altitude.

[0076] Ainsi, une quinzième opération **115** consiste, pour l'enclave **8**, à transmettre à l'émetteur **E** une requête (**REQ**) de communication de données **23** instantanées de géolocalisation (à moins que les données **23** n'aient été communiquées préalablement, par ex. avec les données **15** biométriques.

[0077] Une seizième opération **116** consiste, au niveau de l'émetteur **E**, à éditer, à partir de la balise **22**, des données **23** instantanées de géolocalisation.

[0078] Une dix-septième opération **117**\consiste, à partir de l'émetteur **E**, à charger dans l'enclave **8** les données **23** instantanées de géolocalisation, via une ligne **17** de communication sécurisée.

[0079] Une dix-huitième opération **118** consiste, pour l'enclave **8**, à vérifier que les données **23** instantanées de géolocalisation satisfont la condition de géolocalisation. Dans l'exemple illustré, cette condition de géolocalisation est issue des instructions du contrat intelligent **SC**.

[0080] Cette vérification produit un résultat **24** (typiquement, un bit de valeur 1 lorsque la condition de géolocalisation est satisfaite, un bit de valeur 0 lorsque la condition de géolocalisation n'est pas satisfaite).

[0081] Si les données **23** instantanées de géolocalisation ne satisfont pas la condition de géolocalisation, il est mis fin au procédé de contrôle, et il n'est délivré aucune autorisation d'accès.

[0082] Si au contraire les données **23** instantanées de géolocalisation satisfont la condition de géolocalisation, l'action pour laquelle le contrôle des données biométriques de l'utilisateur était requis est autorisée.

[0083] Lorsque l'action (typiquement un déverrouillage) doit être appliquée au niveau de l'émetteur **E**, celui-ci doit être informé du résultat.

[0084] Dans ce cas, une dix-neuvième opération **119** consiste, pour l'enclave **8**, à transmettre le résultat **21** à l'émetteur **E**.

[0085] Par ailleurs, aux fins de traçabilité, le résultat **21** (ou, de préférence, une trace de ce résultat **21**) ainsi que le résultat **24** (ou de préférence une trace de ce résultat **24**), sont de préférence inscrits dans la blockchain **5**.

[0086] Dans ce cas, une vingtième opération **120** consiste, pour l'enclave **8**, à initier une transaction **TX** à destination de la blockchain **5**. Cette transaction peut être signée au moyen d'une clé privée associée à l'enclave **8**.

[0087] A cet effet, et selon un mode de réalisation illustré sur la **FIG.4**, une vingt-et-unième opération **121** consiste, pour l'enclave **8**, à établir une session de communication (**SESS**) avec au moins un nœud **2** du réseau **1**, une vingt-deuxième opération **122** consistant alors à transmettre à ce nœud **2** la transaction **TX** signée, en vue de son inscription dans la blockchain **5**. La transaction **TX** signée est alors distribuée sur plusieurs nœuds **2** validateurs du réseau **1**, aux fins de vérification préalable à l'inscription.

[0088] Après que la transaction **TX** signée a été vérifiée, une vingttroisième opération **123** consiste, pour un ou plusieurs nœuds **2** validateurs, à l'inscrire dans un nouveau bloc **4** destiné à la blockchain **5**. Une vingt-quatrième opération **124** consiste, pour l'un des nœuds **2** validateurs, à ajouter le nouveau bloc **4** contenant la transaction **TX** signée à la blockchain **5**, après l'achèvement d'un mécanisme de consensus tel que preuve de travail (en anglais proof-of-work ou PoW), preuve d'autorité (en anglais proof-of-authority ou PoA) ou preuve d'enjeu (en anglais proof-of-stake ou PoS).

[0089] Le procédé qui vient d'être décrit présente les avantages suivants.

[0090] Premièrement, les données **19** biométriques de référence, cryptées, ne sont exploitables par aucun tiers, y compris celui qui en assure la mémorisation sur un nœud **2A** de stockage.

[0091] Aucune utilisation (en particulier commerciale) ne peut donc en être faite, au bénéfice de la confidentialité.

[0092] Deuxièmement, la réplication du conteneur **18** sur plusieurs nœuds **2A** de stockage limite, par ailleurs, le risque d'effacement tout en augmentant la fiabilité du procédé.

[0093] Troisièmement, le caractère décentralisé du contrôle évite une éventuelle prise en main frauduleuse sur l'émetteur **E** par un tiers non autorisé.

[0094] Quatrièmement, la condition de géolocalisation ajoute encore à la difficulté pour un tiers non autorisé à accéder à l'environnement souhaité (typiquement un environnement virtuel), lorsque cette condition n'est pas satisfaite.

Revendications

1. Procédé de contrôle de données biométriques d'un individu, à partir d'une unité de traitement informatique, dite émetteur (**E**), reliée ou intégrée à un réseau (**1**) pair-à-pair composé d'une pluralité de nœuds (**2**) formant une base de données distribuée sur laquelle est mémorisée, par réplication sur chaque nœud (**2**), une chaîne (**5**) de blocs, ce procédé comprenant les opérations suivantes :
   - Etablir une session de communication entre l'émetteur (**E**) et le réseau (**1**) ;
   - Sélectionner parmi le réseau (**1**) un nœud (**2P**) dit de calcul, équipé d'une unité de traitement dans laquelle est implémenté un environnement d'exécution sécurisé par cryptographie, dit enclave (**8**) ;
   - Instancier l'enclave (**8**) ;
   - Au moyen d'un dispositif (**16**) de capture équipant ou relié à l'émetteur (**E**), réaliser une capture de données (**15**) biométriques de l'individu au niveau d'un membre ou d'un organe de cet individu ;
   - Charger dans l'enclave (**8**), via une ligne (**17**) de communication sécurisée, les données (**15**) biométriques capturées ;
   - Au moyen d'une balise (**22**) de géolocalisation équipant ou reliée à l'émetteur (**E**), éditer des données (**23**) instantanées de géolocalisation ;
   - Charger dans l'enclave (**8**), via une ligne (**17**) de communication sécurisée, les données (**23**) instantanées de géolocalisation ;
   - Sélectionner parmi le réseau (**1**) un nœud (**2A**) de stockage sur lequel est mémorisé un conteneur (**18**) crypté contenant des données (**19**) biométriques de référence ;

- Charger le conteneur (**18**) crypté dans l'enclave (**8**) ;
- Déchiffrer dans l'enclave (**8**) les données biométriques (**19**) de référence, au moyen d'une clé (**20**) de déchiffrement associée au conteneur (**18**) crypté ;
- Dans l'enclave (**8**), comparer les données (**15**) biométriques capturées et les données (**19**) de référence, et vérifier si les données (**23**) instantanées de géolocalisation satisfont une condition de géolocalisation prédéfinie ;
- Inscrire dans un bloc (**4**) de la chaîne (**5**) de blocs une transaction comprenant une trace du résultat (**21**) de la comparaison et une trace du résultat (**24**) de la vérification.

2. Procédé de contrôle selon la revendication 1, qui comprend les opérations suivantes :
- A partir de l'enclave (**8**) du nœud de calcul, sélectionner parmi le réseau (**1**) des nœuds (**2C**) sur lesquels sont stockés des fragments (**20.i**) de la clé (**20**) cryptographique de déchiffrement associée au conteneur (**18**) ;
- Charger lesdits fragments (**20.i**) dans l'enclave (**8**) du nœud de calcul ;
- Dans l'enclave :
o Reconstituer la clé (**20**) cryptographique de déchiffrement associée au conteneur (**18**), à partir des fragments (**20.i**) ainsi chargés ;
o Déchiffrer les données biométriques (**19**) du conteneur (**18**) au moyen de la clé (**20**) ainsi reconstituée.

FIG.1

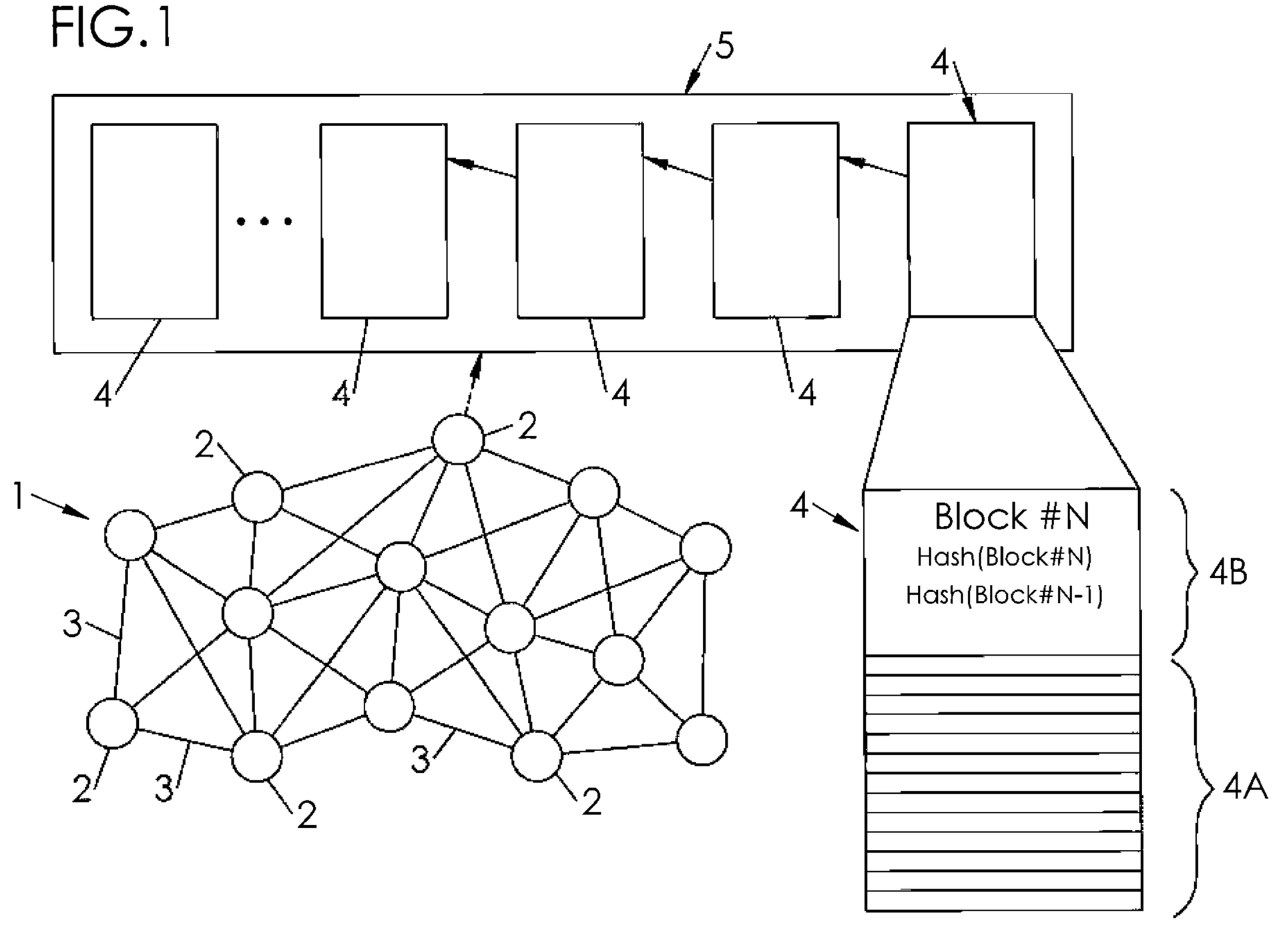
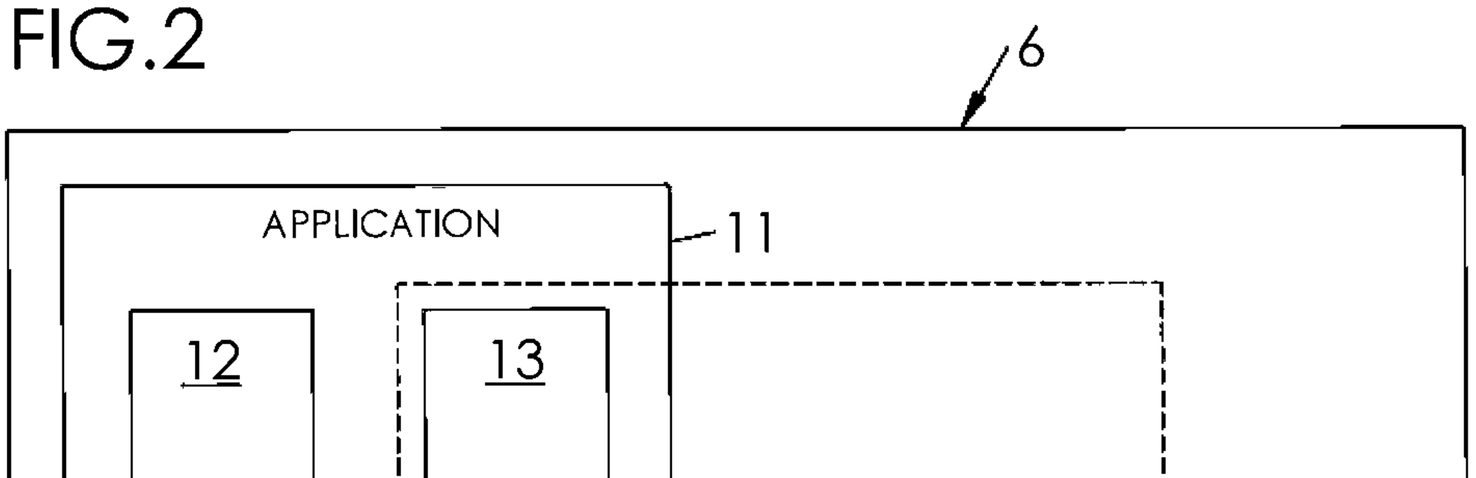

5
4
4
4
4
4
2
2
1
3
3
2
3
2
2
4
Block #N
Hash(Block#N)
Hash(Block#N-1)
4B
4A

FIG.2


6
APPLICATION
11
12
13
8
7
10
CPU
DRAM
ECREATE
EINIT
EADD
EENTER
ERESUME
EEXIT
14
EPC
9

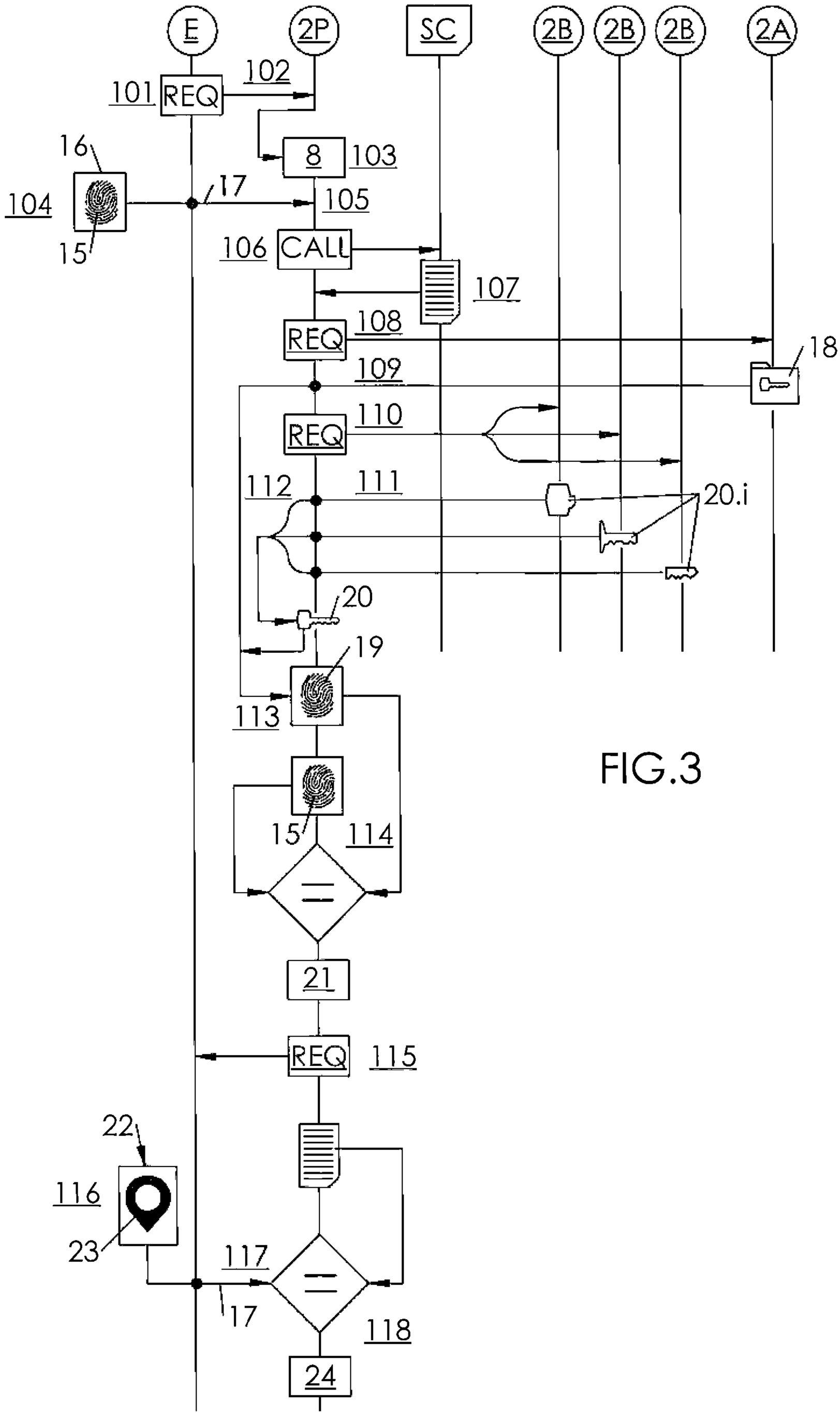

E
2P
SC
2B
2B
2B
2A
101
REQ
102
8
103
16
104
17
105
15
106
CALL
107
108
REQ
18
109
110
REQ
112
111
20.i
20
19
113
15
114
21
REQ
115
22
116
23
117
17
118
24

FIG.3

FIG.4