(19) CONFÉDÉRATION SUISSE
INSTITUT FÉDÉRAL DE LA PROPRIÉTÉ INTELLECTUELLE

(11) **CH 716 261 A2**

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

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: 00735/19

(22) Date de dépôt: 07.06.2019

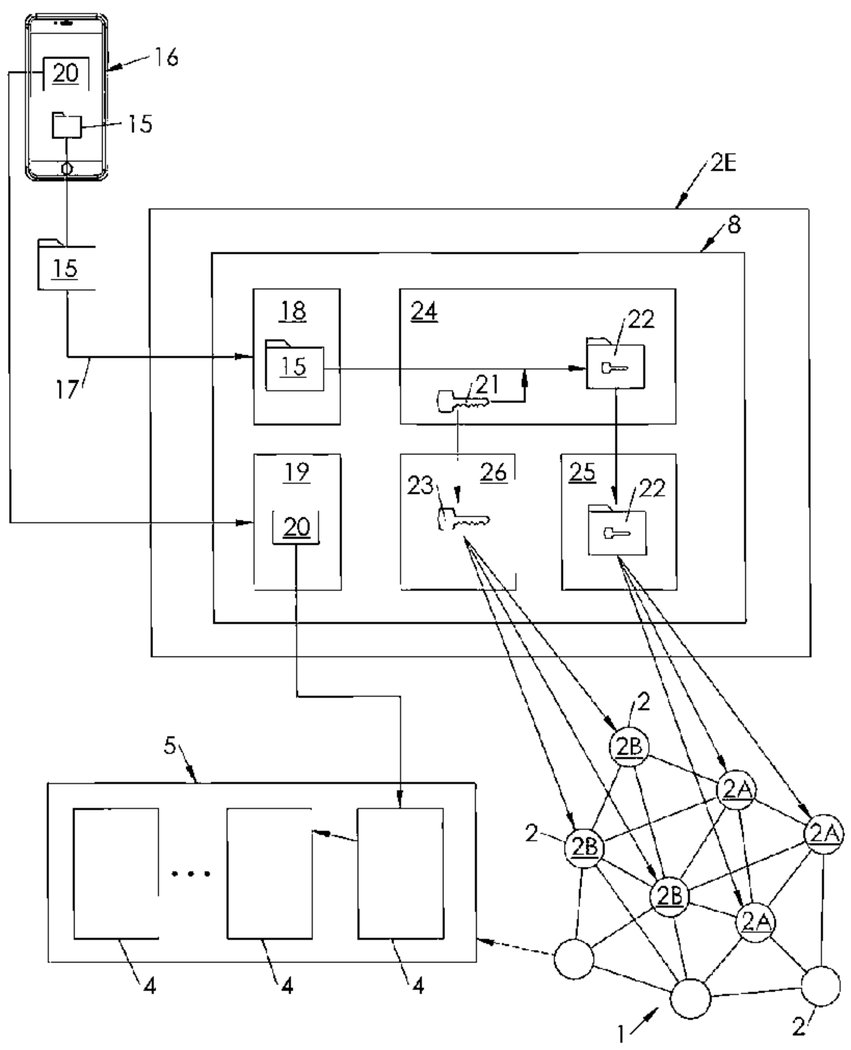
(43) Demande publiée: 15.12.2020

(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 stockage de données informatiques par distribution d'un conteneur crypté et de sa clé de déchiffrement sur un réseau blockchain.

(57) L'invention concerne un procédé de stockage sécurisé de données (**15**) informatiques au sein d'un réseau (**1**) pair-à-pair composé d'une pluralité de nœuds (**2**) sur lequel est distribué une chaîne (**5**) de blocs, procédé dans lequel les données sont cryptées au sein d'une enclave (**8**). Tant les données (**15**) ainsi cryptées qu'une clé (**23**) de déchiffrement de celles-ci sont distribuées sur les nœuds (**2**) du réseau pair-à-pair, et des empreintes numériques du chargement des données dans l'enclave (**8**) et de leur mémorisation sont stockées dans un bloc (**4**) de la chaîne (**5**) de blocs.



Description

DOMAINE TECHNIQUE

[0001] L'invention a trait au domaine de l'informatique, et plus précisément au domaine du stockage sécurisé des données informatiques en réseau.

ART ANTERIEUR

[0002] La sécurisation du stockage en réseau des données informatiques a pour double objectif d'éviter la perte des données (c'est-à-dire leur effacement intempestif) et leur exploitation (ce terme incluant la lecture ainsi que la copie partielle ou totale des données) par des tiers non autorisés. En d'autres termes, la sécurisation vise à garantir la pérennité et la confidentialité des données.

[0003] Pour minimiser le risque de perte des données (en d'autres termes, pour maximiser la pérennité des données), on procède généralement à des réplications, c'est-à-dire que l'on effectue une distribution des données parmi plusieurs espaces de stockage. Les données sont par conséquent stockées de manière redondante.

[0004] Pour garantir (autant que possible) la confidentialité des données stockées en réseau, on recourt généralement à deux méthodes :

– La première consiste à restreindre (typiquement via des mots de passe ou des certificats électroniques) l'accès à un serveur sur lequel sont stockées les données, ce qui minimise les risques de lecture ou de copie ;
– La deuxième consiste à chiffrer les données elles-mêmes au moyen de techniques cryptographiques.

[0005] La première méthode est efficace à deux conditions principales.

[0006] Première condition : le risque de faille dans les restrictions d'accès au serveur doit être nul ou, à tout le moins, minimum.

[0007] L'expérience montre toutefois que certaines attaques permettent de contourner ces restrictions, typiquement en récupérant les identifiants et mots de passe d'utilisateurs et en usurpant leur identité pour accéder à leurs données.

[0008] Deuxième condition : l'administrateur du serveur doit lui-même être digne de confiance.

[0009] L'expérience montre toutefois que certains fournisseurs de services en ligne (notamment de réseaux sociaux), qui administrent des serveurs sur lesquels sont stockées les données personnelles de nombreux utilisateurs, se permettent d'accéder à ces données et d'en faire une exploitation pour leur propre compte, typiquement en revendant les données à des sociétés commerciales ou à des agences gouvernementales, ou en analysant elles-mêmes les données.

[0010] La deuxième méthode résout les problèmes de la première, puisque les tiers non autorisés ne peuvent faire aucune exploitation des données, sauf à craquer les algorithmes de cryptage, ce qui, jusqu'à présent, s'est révélé infaisable pour les algorithmes les plus couramment utilisés, tels que l'algorithme RSA (Rivest, Shamir, Adleman) ou l'algorithme des courbes elliptiques.

[0011] Généralement, les données sont chiffrées localement au sein d'un terminal émetteur (typiquement un ordinateur personnel), puis les données chiffrées sont transmises au réseau pour y être stockées tandis qu'une clé cryptographique de déchiffrement des données est stockée localement au sein du terminal émetteur.

[0012] Cette méthode présente cependant un risque : la perte de la clé cryptographique de déchiffrement des données, ce qui rend les données définitivement inexploitables (et ce par quiconque, y compris leur propriétaire), car indéchiffrables.

[0013] Il est envisageable de déléguer au réseau (typiquement à un serveur mandataire distant) le soin de procéder au chiffrement des données et d'administrer la clé cryptographique de déchiffrement.

[0014] Dans ce cas se pose à nouveau le problème d'un potentiel accès aux données par l'administrateur du serveur mandataire, soit que celui-ci les copie avant leur chiffrement, soit qu'il prenne la liberté de les déchiffrer en exploitant la clé cryptographique générée. Se pose également le problème, du point de vue du terminal émetteur, d'obtenir une preuve, de la part du réseau, que le chiffrement a bien été effectué.

[0015] L'invention vise à offrir une solution efficace à ces problèmes, en particulier au problème de la confidentialité.

RESUME DE L'INVENTION

[0016] Il est proposé un procédé de stockage sécurisé de données informatiques au sein d'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 consistant à :

- Emettre, à partir d'un terminal émetteur dans lequel sont stockées les données, une requête de stockage à destination du réseau ;

- A réception de la requête de stockage par au moins un nœud du réseau, sélectionner au sein du réseau au moins un nœud, dit nœud d'entrée, équipé d'une unité de traitement informatique dans laquelle est implémenté un environnement d'exécution sécurisé par cryptographie, dit enclave ;

- Instancier l'enclave ;

- Charger les données, à partir du terminal émetteur et via une ligne de communication sécurisée, vers l'enclave du nœud d'entrée ;

- Inscrire dans un bloc de la chaîne de blocs, par un ou plusieurs nœuds du réseau, au moins une transaction contenant une empreinte numérique de la requête de stockage et/ou du chargement ainsi effectué ;

- Dans l'enclave :

o Chiffrer les données pour former un conteneur crypté, en lui associant une clé cryptographique de déchiffrement ;

o Désigner, parmi le réseau, un ou plusieurs nœuds primaires de stockage et un ou plusieurs nœuds secondaires de stockage ;

o Distribuer le conteneur crypté vers le ou les nœuds primaires de stockage ;

o Distribuer la clé cryptographique de déchiffrement vers le ou les nœuds secondaires de stockage ;

- Hors de l'enclave :

o Mémoriser le conteneur crypté au sein de chaque nœud primaire de stockage ;

o Mémoriser la clé cryptographique de déchiffrement au sein de chaque nœud secondaire de stockage ;

- Inscrire dans un bloc de la chaîne de blocs, par un ou plusieurs nœuds du réseau, au moins une transaction contenant une empreinte numérique des distributions ou des mémorisations ainsi effectuées.

BREVE DESCRIPTION DES FIGURES

[0017] 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 pour partie une architecture réseau, pour partie des étapes d'un procédé de stockage, et pour partie des fichiers produits, échangés ou stockés au sein du réseau pour les besoins (ou en application) de ce procédé.

DESCRIPTION DETAILLEE DE L'INVENTION

[0018] Sans s'y restreindre, le procédé de stockage 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).

[0019] 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“.

[0020] 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.

[0021] 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.

[0022] Les nœuds **2** peuvent être disséminés sur de larges régions géographiques ; ils peuvent également être regroupés dans des régions géographiques plus restreintes.

[0023] 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.

[0024] 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.

[0025] 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.

[0026] 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.

[0027] 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.

[0028] 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**.

[0029] 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“.

[0030] 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.

[0031] Sauf mention contraire, et par souci de simplification, l'expression simple „chaîne de blocs“ ou „blockchain“ désigne le registre **5** blockchain lui-même.

[0032] 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**.

[0033] On décrit à présent succinctement la technologie des contrats intelligents.

[0034] 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 Exécutable 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.

[0035] 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.

[0036] 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.

[0037] Le compte initiateur récupère (c'est-à-dire, en pratique, télécharge) le résultat lors de sa synchronisation au registre **5** blockchain.

[0038] On introduit à présent brièvement les environnements d'exécution sécurisé.

[0039] Un environnement d'exécution sécurisé (Trusted exécution 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.

[0040] 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).

[0041] 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.

[0042] 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.

[0043] 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.

[0044] 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**).

[0045] 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)).

[0046] 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).

[0047] 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**.

[0048] 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**.

[0049] 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**.

[0050] Nous expliquerons ci-après comment sont exploitées les enclaves.

[0051] Les nœuds **2** du réseau **1** blockchain étant tous équipés de zones mémoires, celles-ci peuvent être exploitées en tant qu'espace de stockage pour des données **15** issues d'un terminal **16** émetteur (ici représenté sous forme d'un smartphone) relié au réseau **1**. Pour minimiser le risque de perte des données **15**, il est avantageux de procéder à une réplication de celles-ci, c'est-à-dire d'effectuer une copie des données **15**, et de distribuer une copie à plusieurs nœuds **2** du réseau **1** blockchain. La traçabilité des données **15** peut être réalisée par inscription, dans le registre **5** blockchain, d'une ou plusieurs empreintes numériques des mémorisations ainsi effectuées.

[0052] Ce stockage distribué des données **15** est avantageusement piloté par un contrat intelligent, activé par exemple par le terminal **16** émetteur qui, tout en transmettant les données **15** à stocker au réseau **1**, transmet au contrat intelligent un appel par la procédure décrite précédemment.

[0053] Si la réplication des données **15** au sein du réseau **1** blockchain résout le problème de la pérennité des données **15**\ il ne résout pas le problème de leur confidentialité.

[0054] Procéder à un chiffrement des données **15** au niveau du terminal **16** émetteur (lequel stockerait une clé de déchiffrement des données) résoudrait le problème de la confidentialité vis-à-vis des tiers non autorisés (sauf à admettre que la clé de déchiffrement pourrait être copiée par un tiers non autorisé à partir du terminal **16** émetteur), mais ne garantirait cependant pas que les données demeureraient accessibles par le terminal **16** émetteur lui-même, dans l'hypothèse - qui est réaliste en pratique - où la clé cryptographique serait perdue ou corrompue.

[0055] Il est ici proposé de distribuer sur le réseau **1** non seulement les données **15** (sous forme cryptée), mais également une clé de déchiffrement de celles-ci, via une enclave **8** instanciée sur un nœud **2E**, dit nœud d'entrée, du réseau **1** (**FIG.3**).

[0056] A cet effet, une opération préliminaire consiste, à partir du terminal **16** émetteur dans lequel sont stockées les données **15**, à transmettre une requête de stockage des données **15** à destination du réseau **1**, typiquement en activant un contrat intelligent déployé sur la blockchain **5**.

[0057] A réception de la requête de stockage par au moins un nœud **2** du réseau **1**, le contrat intelligent sélectionne, au sein du réseau **1** au moins un nœud **2E** d'entrée, équipé d'une unité de traitement informatique dans laquelle est implémentée une enclave **8**.

[0058] Cette enclave **8** est alors instanciée en application des instructions du contrat intelligent, et les données **15** y sont chargées à partir du terminal **16** émetteur et via une ligne **17** de communication sécurisée (par ex. utilisant le protocole Transport Layer Security ou TLS). A cet effet, l'enclave **8** peut être pourvue, dès son instanciation, d'une émulation d'interface **18** de communication supportant le protocole choisi (ici TLS) pour l'échange des données sécurisées.

[0059] Le protocole blockchain est avantageusement chargé dans l'enclave **8**, pour former un module **19** blockchain apte à interroger le registre **5** blockchain et/ou à participer au processus de création et validation des blocs **4**.

[0060] Une transaction **20** contenant une empreinte numérique du chargement des données ainsi effectué est inscrite dans un bloc **4** de la blockchain **5**, aux fins de traçabilité.

[0061] Cette transaction **20** peut être transmise au module **19** blockchain par le terminal **16** émetteur, pour être inscrite par un ou plusieurs nœuds **2** du réseau (par le processus décrit plus haut) dans un nouveau bloc **4** du registre **5** blockchain. En variante, le module **19** blockchain émet de lui-même une transaction **20** pour inscription dans un nouveau bloc **4**.

[0062] Dans l'enclave **8** se déroule alors un processus qui comprend les opérations suivantes.

[0063] Une première opération consiste à chiffrer les données **15** au moyen d'une clé **21** cryptographique de chiffrement pour former un conteneur **22** crypté, en lui associant une clé **23** cryptographique de déchiffrement. A cet effet, les données **15** reçues du terminal **16** émetteur sont relayées par l'interface **18** de communication à un module **24** de chiffrement implémenté dans l'enclave **8**.

[0064] Selon un mode préféré de réalisation, le chiffrement est symétrique. Dans ce cas, la clé **21** de chiffrement et la clé **23** de déchiffrement sont une seule et même clé.

[0065] Une deuxième opération consiste à désigner, parmi le réseau **1**, un ou plusieurs nœuds **2A** primaires de stockage et un ou plusieurs nœuds **2B** secondaires de stockage.

[0066] Une troisième opération consiste à distribuer le conteneur **22** crypté vers le ou les nœuds **2A** primaires de stockage.

[0067] A cet effet, l'enclave **8** est avantageusement pourvue d'un module **25** de distribution de données, auquel le module **24** de chiffrement communique le conteneur **22** crypté pour distribution aux nœuds **2A** primaires de stockage.

[0068] Une quatrième opération consiste à distribuer la clé **23** cryptographique de déchiffrement vers les nœuds **2B** secondaires de stockage.

[0069] A cet effet, l'enclave **8** est avantageusement pourvue d'un module **26** de distribution de clé, auquel le module **24** de chiffrement communique la clé **23** de déchiffrement pour distribution aux nœuds **2B** secondaires de stockage.

[0070] Ces opérations achevées, l'enclave **8** peut être refermée.

[0071] Hors de l'enclave **8**, les opérations suivantes sont réalisées :

- o Le conteneur **22** crypté est mémorisé au sein de chaque nœud **2A** primaire de stockage ;
- o La clé **23** cryptographique de déchiffrement est mémorisée au sein de chaque nœud **2B** secondaire de stockage.

[0072] Pour assurer la traçabilité de ces opérations, au moins une transaction contenant une empreinte numérique des distributions ou des mémorisations ainsi effectuées est inscrite dans un bloc **4** de la chaîne **5** de blocs, par un ou plusieurs nœuds **2** du réseau **1**. Cette transaction peut être initiée par les nœuds **2A** primaires et **2B** secondaires de stockage, mais elle peut également être initiée par l'enclave **8** elle-même (et plus précisément par le module **19** blockchain) avant sa fermeture.

[0073] Le procédé qui vient d'être décrit présente les avantages suivants.

[0074] La clé **23** de déchiffrement des données n'étant pas stockée localement au niveau du terminal **16** émetteur, elle ne risque pas d'être perdue avec celui-ci. On peut observer que le fait qu'elle soit stockée sur des nœuds **2B** distants ne la rend pas plus vulnérable aux appropriations indues que si elle était stockée dans le terminal **16** émetteur.

[0075] Les données étant cryptées au sein du conteneur **22**, leur accès est impossible sans obtenir (indument) la clé **23** de déchiffrement.

[0076] Quant à la réplication du conteneur **22** sur le réseau **1** blockchain, elle permet de stocker les données avec une bonne assurance qu'elles pourront être récupérées le moment venu par les personnes autorisées à récupérer également la clé **23** de déchiffrement depuis au moins un nœud **2B** secondaire de stockage.

Revendications

1. Procédé de stockage sécurisé de données (**15**) informatiques au sein d'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 consistant à :

- Emettre, à partir d'un terminal (**16**) émetteur dans lequel sont stockées les données (**15**), une requête de stockage à destination du réseau (**1**) ;
- A réception de la requête de stockage par au moins un nœud (**2**) du réseau, sélectionner au sein du réseau (**1**) au moins un nœud (**2**), dit nœud (**2E**) d'entrée, équipé d'une unité de traitement informatique dans laquelle est implémenté un environnement d'exécution sécurisé par cryptographie, dit enclave (**8**) ;
- Instancier l'enclave (**8**) ;
- Charger les données (**15**), à partir du terminal (**16**) émetteur et via une ligne (**17**) de communication sécurisée, vers l'enclave (**8**) du nœud (**2E**) d'entrée ;
- Inscrire dans un bloc (**4**) de la chaîne (**5**) de blocs, par un ou plusieurs nœuds (**2**) du réseau (**1**), au moins une transaction (**20**) contenant une empreinte numérique de la requête de stockage et/ou du chargement ainsi effectué ;
- Dans l'enclave (**8**) :
o Chiffrer les données (**15**) pour former un conteneur (**22**) crypté, en lui associant une clé (**23**) cryptographique de déchiffrement ;
o Désigner, parmi le réseau (**1**), un ou plusieurs nœuds (**2A**) primaires de stockage et un ou plusieurs nœuds (**2B**) secondaires de stockage ;
o Distribuer le conteneur (**22**) crypté vers le ou les nœuds (**2A**) primaires de stockage ;
o Distribuer la clé (**23**) cryptographique de déchiffrement vers le ou les nœuds (**2B**) secondaires de stockage ;
- Hors de l'enclave (**8**) :
o Mémoriser le conteneur (**22**) crypté au sein de chaque nœud (**2**) primaire de stockage ;
o Mémoriser la clé (**23**) cryptographique de déchiffrement au sein de chaque nœud (**2B**) secondaire de stockage ;
- Inscrire dans un bloc (**4**) de la chaîne (**5**) de blocs, par un ou plusieurs nœuds (**2**) du réseau (**1**), au moins une transaction contenant une empreinte numérique des distributions ou des mémorisations ainsi effectuées.

FIG.1

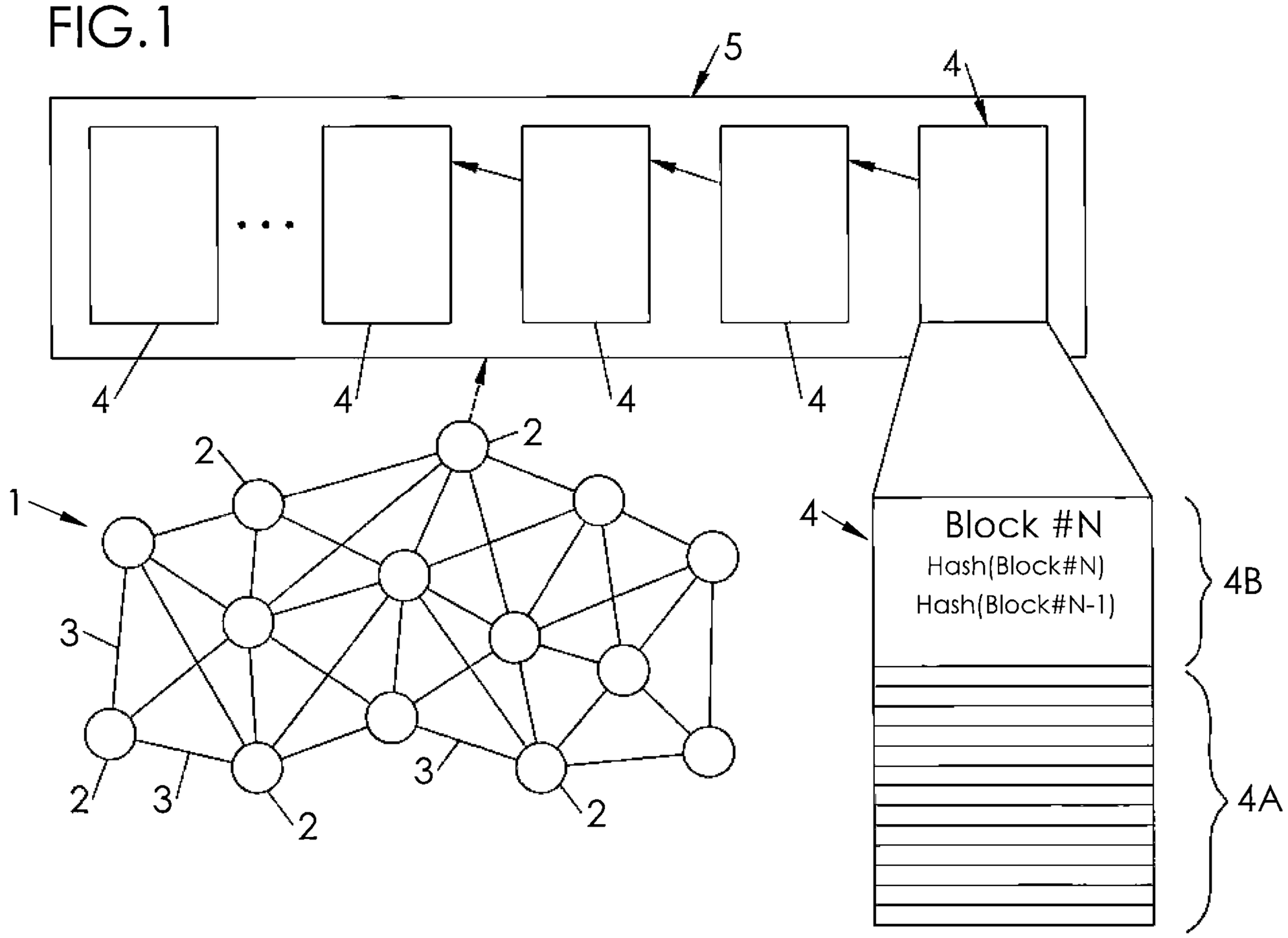
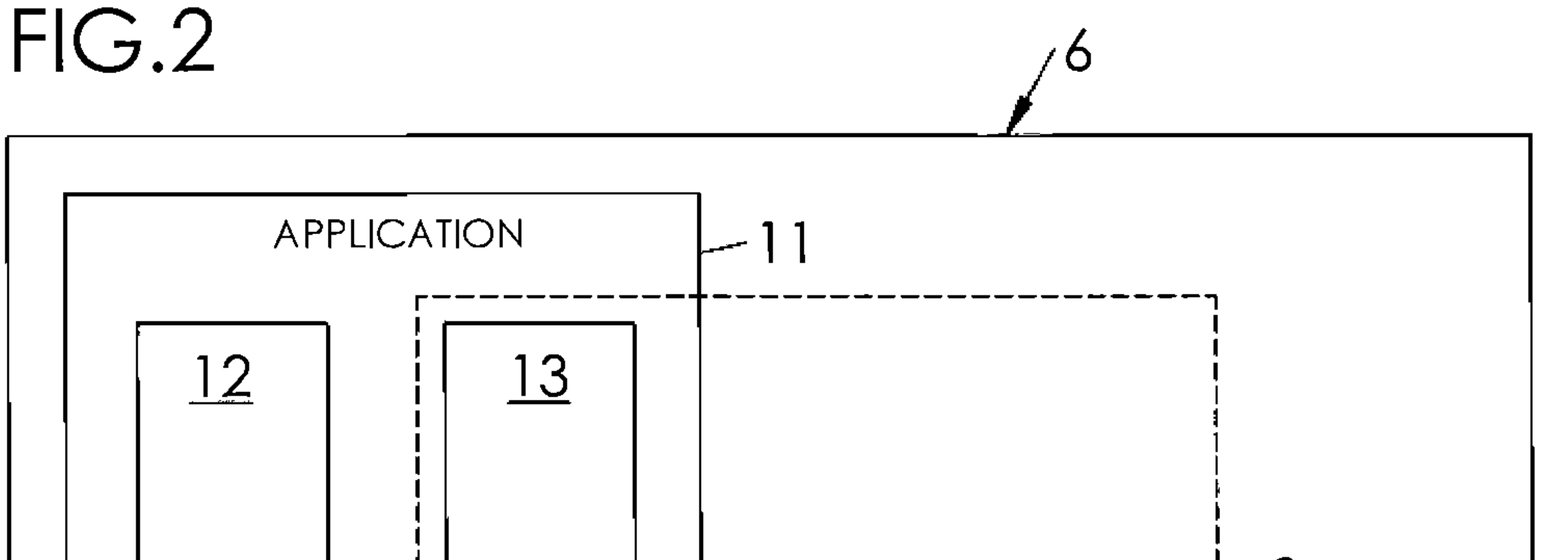

5
4
4
4
4
4
4
2
2
1
3
2
3
2
3
2
4
Block #N
Hash(Block#N)
Hash(Block#N-1)
4B
4A

FIG.2


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

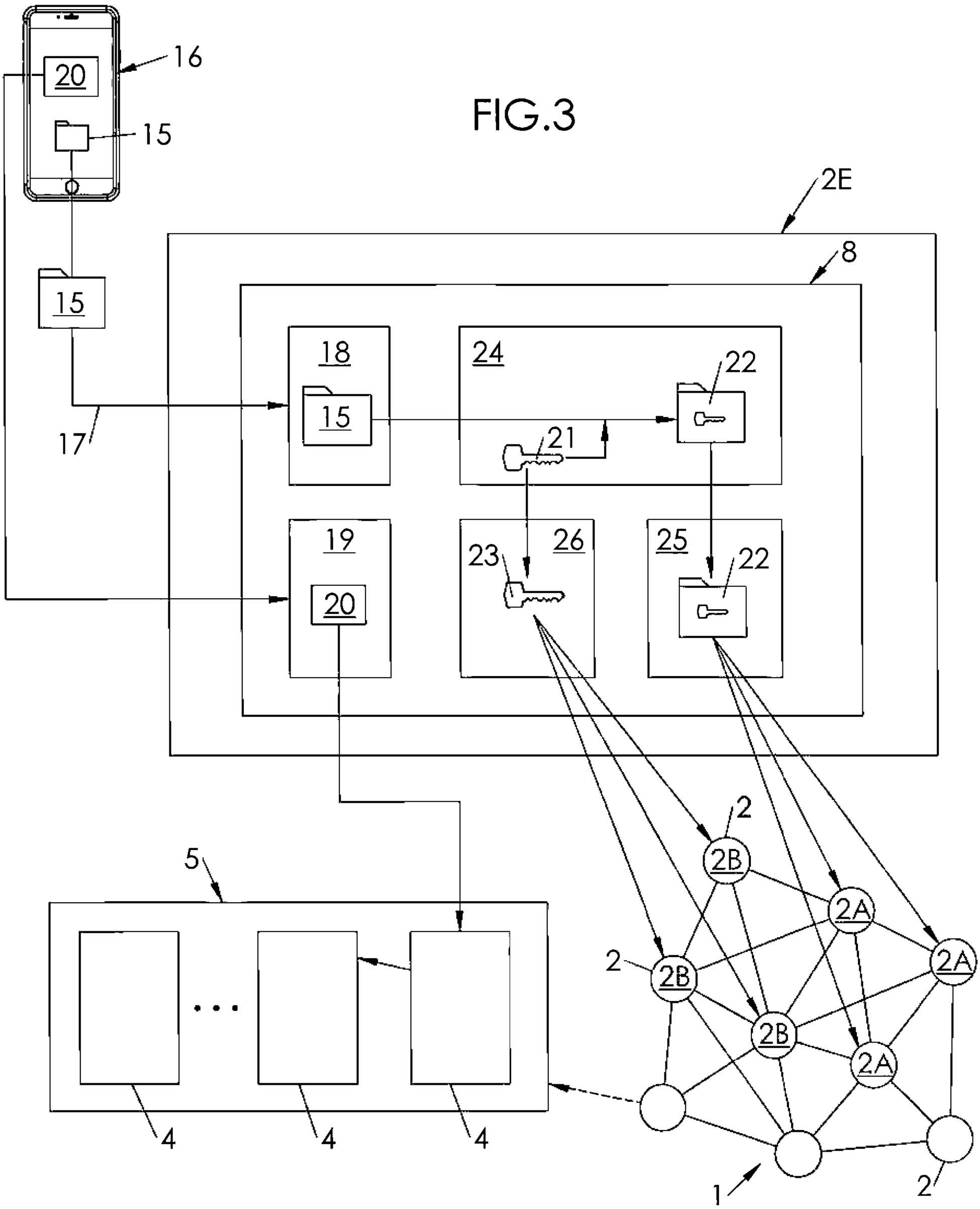

16
20
15
15
17
8
18
15
24
22
21
19
20
26
23
25
22
2E
2
2B
2A
2
2B
2A
2B
2A
2
1
5
4
4
4

FIG.3