(19) CONFÉDÉRATION SUISSE
INSTITUT FÉDÉRAL DE LA PROPRIÉTÉ INTELLECTUELLE

(11) **CH 716 301 A2**

(51) Int. Cl.: ***G06F 21/62*** *(2013.01)*
G06F 21/57 *(2013.01)*
H04L 29/08 *(2006.01)*
H04L 9/06 *(2006.01)*
H04L 9/32 *(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: 00773/19

(22) Date de dépôt: 07.06.2019

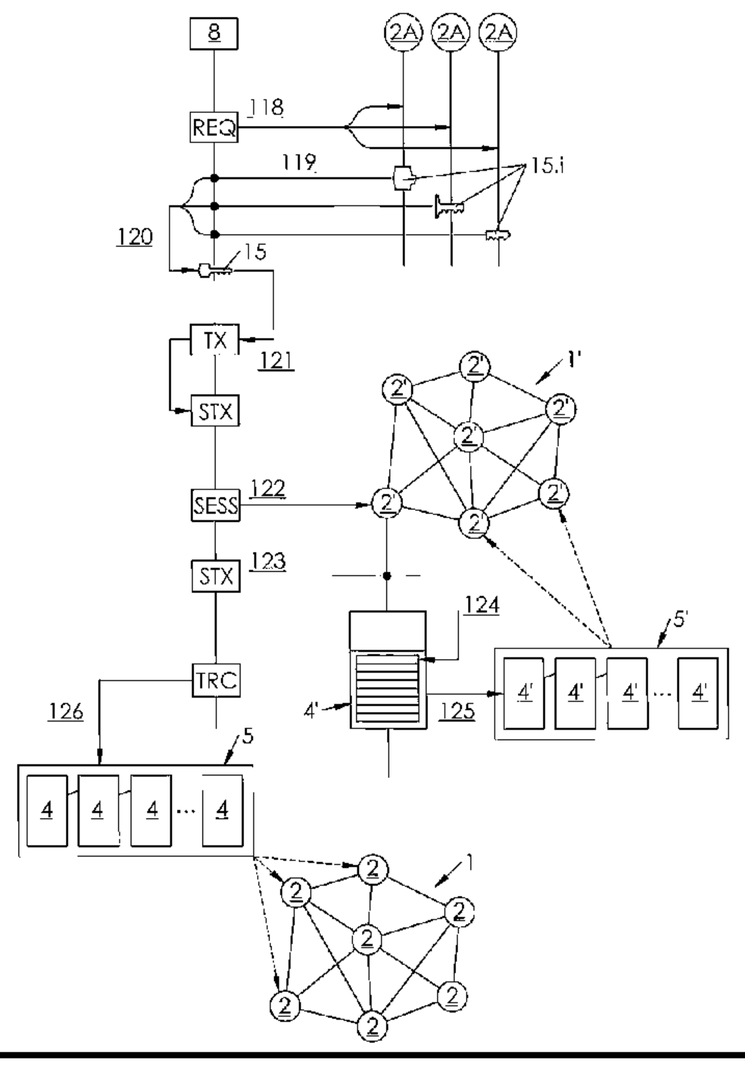
(43) Demande publiée: 15.12.2020

(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 signature d'une transaction destinée à une blockchain déployée sur un réseau pair-à-pair, au moyen d'une clé cryptographique distribuée parmi les noeuds d'un autre réseau pair-à-pair.

(57) L'invention concerne un procédé de signature numérique d'une transaction (**TX**) destinée à être inscrite dans un nouveau bloc (**4'**) d'une blockchain (**5'**) déployée sur un réseau (**1'**) pair-à-pair, dans lequel la transaction (**TX**) est initiée par un émetteur et transmise à une enclave (**8**) cryptographique d'un nœud de calcul d'un autre réseau (**1**) pair-à-pair ; des fragments (**15.i**) de la clé (**15**) privée associée à l'émetteur, et stockés sur des nœuds de cet autre réseau (**1**), sont chargés dans l'enclave (**8**), la clé (**15**) est reconstituée à partir de ces fragments (**15.i**) et appliquée à la transaction (**TX**) pour former une transaction (**STX**) signée, laquelle est ensuite inscrite dans un nouveau bloc (**4'**) de la blockchain (**5'**).



Description

DOMAINE TECHNIQUE

[0001] L'invention a trait au domaine de l'informatique, et plus précisément à 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).

ART ANTERIEUR

[0002] 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“.

[0003] 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.

[0004] 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.

[0005] Les nœuds **2** peuvent être disséminés sur de larges régions géographiques ; ils peuvent également être regroupés dans des régions géographiques plus restreintes.

[0006] 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.

[0007] 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.

[0008] 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.

[0009] 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.

[0010] 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.

[0011] 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**.

[0012] Sauf mention contraire, et par souci de simplification, l'expression simple „chaîne de blocs“ ou „blockchain“ désigne le registre **5** blockchain lui-même.

[0013] 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 la blockchain **5**, des transactions passées entre un ou plusieurs terminaux émetteurs, et un ou plusieurs terminaux récepteurs, indifféremment appelés „utilisateurs“.

[0014] 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 la blockchain **5** pour inscrire, dans la zone mémoire, les transactions validées par inscription dans la blockchain **5**.

[0015] Le portefeuille comprend en outre un jeu de clés, et plus précisément au moins une paire de clés appariées, à savoir une clé publique et une clé privée associée. La clé publique est généralement utilisée (directement en tant que telle, ou indirectement par l'intermédiaire d'un condensat) en tant qu'identifiant de l'utilisateur. Quant à la clé privée, elle est généralement utilisée pour signer les transactions émises à partir du portefeuille.

[0016] On comprend dès lors que la clé privée est une donnée critique, dont il convient, pour l'utilisateur, de maintenir une confidentialité stricte vis-à-vis des tiers qui, en possession de la clé privée, pourraient usurper l'identité de l'utilisateur et signer des transactions à sa place. Ces transactions étant inscrites de manière immuable dans la blockchain **5**, il n'est possible, pour l'utilisateur authentique, ni de les annuler, ni de les inverser.

[0017] Perdre un portefeuille électronique revient, pour l'utilisateur, à perdre sa capacité de signer des transactions. Se faire subtiliser ou pirater un portefeuille électronique revient, pour l'utilisateur, à permettre à un tiers (généralement malveillant) de signer les transactions à sa place.

[0018] Il existe différentes versions de portefeuilles électroniques, selon leur mode d'installation.

[0019] Dans une première version purement logicielle, le portefeuille électronique est installé localement, sur un terminal communiquant (typiquement un ordinateur de bureau, fixe ou portable, ou encore un smartphone).

[0020] Cette première version n'est pas sans défaut : comme le terminal est en réseau, elle nécessite de protéger celui-ci des intrusions, en installant des filtres et des passerelles dont l'infaillibilité ne saurait cependant être garantie. Les cas de perte de clés (consécutive à une panne matérielle ou logicielle, ou encore à une désinstallation intempestive du portefeuille) ne sont pas rares. De nombreux cas de vol de clés sont également à déplorer.

[0021] Dans une deuxième version physique, le portefeuille électronique est installé localement encore, sur un dispositif électronique dédié (de type clé USB).

[0022] Ce portefeuille ne devient actif que lorsque le dispositif est branché sur un terminal assurant la connexion au réseau blockchain.

[0023] Cependant les portefeuilles physiques ne vont pas non plus sans défaut. Ils souffrent d'une grande sensibilité aux dégradations (notamment du fait de l'exposition à l'humidité et aux chocs), sont susceptibles d'être perdus, ou d'être volés (avec les mêmes conséquences que la version purement logicielle du portefeuille).

[0024] Dans une troisième version, le portefeuille électronique est installé sur un serveur distant, administré par un tiers (qui en détient par conséquent les clés cryptographiques) et auquel l'utilisateur se connecte via un terminal.

[0025] Si cette version est présumée mieux sécurisée que la première version (purement logicielle et locale) contre la perte et le vol par des tiers, elle présente cependant une faille majeure : le vol des clés par l'administrateur lui-même (et de fait, des cas ont été répertoriés).

[0026] En résumé, la première version est pratique, mais peu fiable.

[0027] La deuxième version est un peu plus fiable que la première, mais beaucoup moins pratique.

[0028] Quant à la troisième version, elle paraît plus pratique que les deux premières, tout en étant plus fiable que la première mais moins, toutefois, que la seconde.

[0029] Quoiqu'il en soit, aucune des versions des portefeuilles électroniques n'est à la fois pratique et fiable.

[0030] Dans tous les cas, pour se prémunir notamment de la perte des clés, les utilisateurs sont incités à en conserver une version imprimée, qu'il convient de garder dans un endroit discret et protégé, tel qu'un coffre-fort. Certains préconisent même de garder cette version imprimée dans un coffre-fort bancaire. Ces techniques anachroniques de conservation du secret ralentissent considérablement l'adoption à grande échelle des technologies de blockchain.

[0031] L'invention vise à proposer une technique alternative de portefeuille électronique dédié aux transactions de blockchain, qui soit à la fois pratique, fiable, sûr, et qui dispense les utilisateurs de devoir effectuer des sauvegardes matérielles de leurs clés.

[0032] Elle vise également à proposer une technique de portefeuille électronique qui permettre d'inscrire des transactions dans des blockchains distinctes.

RESUME DE L'INVENTION

[0033] Il est proposé un procédé de signature numérique d'une transaction initiée par une unité de traitement informatique, dite émetteur, reliée à un réseau pair-à-pair primaire 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 primaire, cette transaction numérique étant destinée à être inscrite dans un nouveau bloc d'une chaîne de blocs secondaire mémorisée par réplication sur chaque nœud d'un réseau pair-à-pair secondaire, ce procédé comprenant les opérations suivantes :

- Générer la transaction par l'émetteur, cette transaction contenant un identifiant d'émetteur ;

- Sélectionner parmi le réseau primaire 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 ;

- Charger la transaction numérique dans l'enclave, via une ligne de communication sécurisée ;

- Sélectionner parmi le réseau primaire des nœuds sur lesquels sont stockés des fragments d'une clé cryptographique privée associée à l'identifiant d'émetteur contenu dans la transaction ;

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

- Dans l'enclave du nœud de calcul :

o Reconstituer la clé cryptographique privée à partir des fragments ainsi chargés ;

o Signer numériquement la transaction par chiffrement de tout ou partie des informations de celle-ci au moyen de la clé cryptographique privée ainsi reconstituée, pour former une transaction signée ;

o Transmettre au réseau secondaire la transaction signée ;

- Au sein d'au moins un nœud du réseau secondaire :

o Vérifier la transaction signée ;

o La transaction signée vérifiée, l'inscrire dans un nouveau bloc destiné à la chaîne de blocs secondaire ;

- Inscrire une trace de la transaction signée dans un nouveau bloc destiné à la chaîne de blocs primaire.

BREVE DESCRIPTION DES FIGURES

[0034] 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 diagramme fonctionnel illustrant des étapes de contrôle préalable à une signature d'une transaction à destination d'une blockchain ;

La **FIG.4** est un diagramme fonctionnel prolongeant le diagramme de la **FIG.3** et illustrant des étapes de signature et d'inscription de la transaction dans la blockchain.

DESCRIPTION DETAILLEE DE L'INVENTION

[0035] L'environnement dans lequel est exécuté le procédé proposé est un réseau **1** blockchain comprenant une pluralité de nœuds **2** interconnectés par des liaisons **3**. Sur ce réseau **1** est déployé un registre **5** contenant des blocs **4** constitué en chaîne, ou chaîne de blocs (blockchain), dont les caractéristiques sont conformes à la description faite ci-dessus en introduction.

[0036] Sur le réseau **1** est implémentée 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 la blockchain **5** à partir des nœuds **2**.

[0037] 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.

[0038] 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.

[0039] 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.

[0040] Le compte initiateur récupère (c'est-à-dire, en pratique, télécharge) le résultat lors de sa synchronisation au registre **5** blockchain.

[0041] Par ailleurs, certains au moins des nœuds **2** sont équipés d'unités **6** de traitement sur lesquelles est implémenté un environnement d'exécution sécurisé ou, en terminologie anglo-saxonne, trusted execution environment (TEE).

[0042] 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.

[0043] 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).

[0044] 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.

[0045] 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.

[0046] 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.

[0047] 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**).

[0048] 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)).

[0049] 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).

[0050] 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**.

[0051] 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**.

[0052] 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**.

[0053] Nous expliquerons ci-après comment sont exploitées les enclaves.

[0054] Le procédé proposé vise à permettre, au sein du réseau **1**, dit réseau primaire, sur lequel est déployée une blockchain **5** primaire, la signature numérique décentralisée d'une transaction **TX** numérique à destination d'une blockchain **5'** dite secondaire, distincte de la blockchain **5** primaire, cette blockchain **5'** secondaire étant déployée sur un réseau **1'** pair-à-pair secondaire distinct du réseau **1** primaire.

[0055] On note **2'** les nœuds du réseau **1'** secondaire. Certains nœuds **2**, **2'** peuvent être communs aux deux réseaux **1**, **1'**.

[0056] Cette signature doit être effectuée par chiffrement de la transaction **TX** au moyen d'une clé **15** cryptographique privée, stockée sur le réseau **1** primaire en application des règles de Shamir (dites du Partage de Secret de Shamir, en anglais Shamir's Secret Sharing).

[0057] Selon les règles de Shamir, la clé **15** privée a, auparavant, été fragmentée en un ensemble de N fragments **15.i** (N un entier prédéterminé, $N>3$, i un indice entier, $1\leq i\leq N$), tel qu'un sous-ensemble de K fragments **15.i** (K un entier prédéterminé, $1<K<N$, avec $1\leq i\leq K$) est suffisant pour reconstituer la clé **15**.

[0058] Chaque fragment **15.i** est stocké séparément dans un nœud **2A** de stockage du réseau **1** primaire.

[0059] La transaction **TX** est initiée par une unité **E** de traitement informatique (équipant par ex. un ordinateur personnel fixe ou portable, une tablette ou encore un smartphone), reliée au réseau **1** primaire et appelée émetteur.

[0060] Ce procédé comprend plusieurs phases, à savoir :

- Une phase de chargement ;
- Une phase de contrôle ;
- Une phase de signature ;
- Une phase de déploiement.

[0061] La phase de chargement comprend une première opération **101** qui consiste, à l'initiative de l'émetteur E, à établir une connexion au réseau **1**, via une requête (**REQ**) adressée à celui-ci.

[0062] Une deuxième opération **102** consiste, parmi le réseau **1** primaire, à sélectionner un nœud **2P** dit de calcul, équipé d'une unité de traitement dans laquelle est implémentée une enclave **8**.

[0063] Une troisième opération **103** consiste, au sein du nœud **2P** de calcul, à instancier l'enclave **8**.

[0064] Une quatrième opération **104** consiste, au sein de l'émetteur **E**, à générer la transaction **TX**.

[0065] Cette transaction **TX** contient typiquement un identifiant associé à l'émetteur **E** ou à son utilisateur personne physique (cette identité se présente par ex. sous forme d'une clé cryptographique publique, ou d'un dérivé de cette clé cryptographique publique, par ex. un condensat de cette clé, notamment par la méthode SHA-256).

[0066] Dans l'hypothèse où la transaction **TX** est émise à l'intention d'un destinataire, elle contient également un identifiant associé à ce destinataire (par ex. sous forme d'une clé publique ou d'un dérivé de cette clé publique, par ex. un condensat).

[0067] Elle contient par ailleurs un actif (qui peut se présenter sous forme de données quelconques - il peut s'agir d'un nombre, correspondant par ex. à un montant exprimé dans une devise).

[0068] Une cinquième opération **105** consiste à charger la transaction **TX** dans l'enclave, via une ligne **16** de communication sécurisée (par ex. utilisant le protocole Transport Layer Security ou TLS).

[0069] Les phases de contrôle, de signature et de déploiement sont conduites au sein ou à partir de l'enclave **8**.

[0070] Selon un mode préféré de réalisation, les opérations relatives au contrôle et à la signature sont effectuées suivant les instructions d'un contrat intelligent **SC** déployé sur la blockchain **5** primaire.

[0071] Comme illustré sur la **FIG.3**, une sixième opération **106** consiste, pour l'enclave **8**, à appeler (**CALL**) le contrat intelligent **SC**.

[0072] 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**.

[0073] La phase de contrôle a pour objectif d'authentifier la (ou les) individu(s) personne(s) physique(s) derrière l'émetteur E, pour minimiser les risques d'usurpation d'identité.

[0074] Cette authentification est ici réalisée ici par comparaison de données biométriques.

[0075] A cet effet, l'émetteur **E** est équipé de (ou relié à) un dispositif **17** de capture biométrique ou scanneur. Le scanneur **17** est configuré pour réaliser une capture de données **18** biométriques de l'individu 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 individu.

[0076] Ainsi, une huitième opération **108** consiste, pour l'enclave **8**, à transmettre à l'émetteur **E** une requête d'authentification.

[0077] Une neuvième opération **109** consiste, au niveau de l'émetteur **E**, à réaliser, au moyen de son scanneur **17**, une capture des données 18 biométriques de l'individu. Dans l'exemple illustré, ces données **18** biométriques sont issues d'une empreinte digitale.

[0078] Une dixième opération **110** consiste, à partir de l'émetteur **E**, à charger, dans l'enclave **8** du nœud **2P** de calcul via une ligne **16** de communication sécurisée (par ex. utilisant le protocole Transport Layer Security ou TLS), les données **18** biométriques ainsi capturées.

[0079] Parallèlement, ou subséquemment, une onzième opération **111** consiste, pour l'enclave **8**, à sélectionner parmi le réseau **1** primaire un nœud **2B** de stockage sur lequel est stocké un conteneur **19** crypté contenant des données **20** biométriques de référence, et à transmettre à ce nœud **2B** de stockage une requête (**REQ**) de communication de ce conteneur **19**.

[0080] Une douzième opération **112** consiste, à partir du nœud **2B** de stockage, à charger dans l'enclave **8** le conteneur **19** crypté.

[0081] Le déchiffrement des données **20** biométriques de référence du premier conteneur **19** crypté nécessite une clé **21** cryptographique de déchiffrement, stockée sur le réseau **1**.

[0082] Selon un mode préféré de réalisation, et comme illustré sur la **FIG.3**, la clé **21** est distribuée sur le réseau **1** primaire en application des règles de Shamir (dites du Partage de Secret de Shamir, en anglais Shamir's Secret Sharing).

[0083] Plus précisément, selon les règles de Shamir, la clé **21** a, auparavant, été fragmentée en un ensemble de N fragments **21.i** (N un entier prédéterminé, $N>3$, i un indice entier, $1\leq i\leq N$), tel qu'un sous-ensemble de K fragments **21.i** (K un entier prédéterminé, $1<K<N$, avec $1\leq i\leq K$) est suffisant pour reconstituer la clé **21**, chaque fragment **21.i** étant stocké séparément dans un nœud **2C** de stockage du réseau **1** primaire.

[0084] Une treizième opération **113** consiste par conséquent, pour l'enclave **8**, à sélectionner parmi le réseau **1** primaire K nœuds **2C** de stockage sur lesquels sont stockés K fragments **21.i** respectifs, et à transmettre à chaque nœud **2C** de stockage une requête (**REQ**) de communication de son fragment **21.i**.

[0085] Comme illustré sur la **FIG.3**, une quatorzième opération **114** consiste, à partir des nœuds **2C** de stockage ainsi sélectionnés, à charger dans l'enclave **8** les K fragments **21.i**.

[0086] Une quinzième opération **115** consiste, pour l'enclave **8**, à reconstituer la clé **21** à partir des K fragments **21.i** ainsi chargés.

[0087] Une seizième opération **116** consiste, pour l'enclave **8**, à déchiffrer les données **20** biométriques de référence du conteneur **19** en lui appliquant la clé **21** ainsi reconstituée.

[0088] Une dix-septième opération **117** consiste, pour l'enclave **8**, à comparer les données **18** biométriques issues de la capture effectuée par le scanneur **17** et les données **20** 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).

[0089] Si cette comparaison est un échec, les données **18**, **20** sont décrétées ne pas correspondre, et il est mis fin au processus de signature. L'émetteur **E** en est informé. Les opérations de capture et de comparaison des données biométriques peuvent être répétées, pour minimiser le risque de faux négatif.

[0090] Si au contraire la comparaison est un succès, les données **18**, **20** sont décrétées correspondre, et la phase de signature est initiée. Pour conduire celle-ci, l'enclave **8** doit disposer de la clé **15** privée.

[0091] Une dix-huitième opération **118** consiste, pour l'enclave **8**, à sélectionner parmi le réseau **1** primaire K nœuds **2A** de stockage sur lesquels sont stockés K fragments **15.i** respectifs, et à transmettre à chaque nœud **2A** de stockage une requête (**REQ**) de communication de son fragment **15.i** (**FIG.4**).

[0092] Comme illustré sur la **FIG.4**, une dix-neuvième opération **119** consiste, à partir des nœuds **2A** de stockage ainsi sélectionnés, à charger dans l'enclave **8** les K fragments **15.i**.

[0093] Une vingtième opération **120** consiste, pour l'enclave **8**, à reconstituer la clé **15** à partir des K fragments **21.i** ainsi chargés.

[0094] Une vingt-et-unième opération **121** consiste à signer la transaction **TX** en chiffrant tout ou partie des informations de celle-ci au moyen de la clé **15** privée ainsi reconstituée, le résultat étant une transaction **STX** signée.

[0095] La phase de déploiement peut alors être initiée : il s'agit de soumettre la transaction **STX** signée au réseau **1'** secondaire pour validation et inscription dans la blockchain **5'** secondaire.

[0096] A cet effet, et selon un mode de réalisation illustré sur la **FIG.4**, une vingt-deuxième opération **122** consiste, pour l'enclave **8**, à établir une session de communication (**SESS**) avec au moins un nœud **2'** du réseau **1'** secondaire, une vingt-troisième opération **123** consistant alors à transmettre à ce nœud **2'** la transaction **STX** signée, en vue de son inscription dans la blockchain **5'** secondaire. La transaction **STX** signée est alors distribuée sur plusieurs nœuds **2'** validateurs du réseau **1** secondaire, aux fins de vérification préalable à l'inscription.

[0097] La vérification de la transaction **STX** comprend au moins la vérification (c'est-à-dire la reconnaissance) de la signature ; elle peut en outre comprendre la reconnaissance, sur la blockchain **5'** secondaire, des identifiants de l'émetteur **E** et du destinataire, et le constat que l'émetteur **E** dispose effectivement d'un solde suffisant.

[0098] Après que la transaction **STX** signée a été vérifiée, une vingt-quatrième opération **124** consiste, pour un ou plusieurs nœuds **2'** validateurs du réseau **1'** secondaire, à l'inscrire dans un nouveau bloc **4'** destinée à la blockchain **5'** secondaire. Une vingt-cinquième opération **125** consiste, pour l'un des nœuds **2'** validateurs du réseau **1'** secondaire, à ajouter le nouveau bloc **4'** contenant la transaction **STX** signée à la blockchain **5'** secondaire, 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).

[0099] Une vingt-sixième opération **126** consiste à inscrire une trace TRC (en pratique, il s'agit d'un condensat) de la transaction **STX** dans la blockchain **5** primaire - et plus exactement dans un nouveau bloc **4** destiné à la blockchain **5** primaire.

[0100] Le procédé qui vient d'être décrit présente les avantages suivants.

[0101] Premièrement, il permet d'offrir une nouvelle version, distribuée, de portefeuille électronique. L'émetteur **E** ne dispose pas de la clé **15** privée utilisée pour signer les transactions **TX** : cette clé **15** est stockée de manière distribuée sur des nœuds **2A** du réseau **1** primaire. Le portefeuille électronique est ainsi indépendant de l'émetteur **E**, tout en demeurant associé à son utilisateur en tant que personne physique.

[0102] Il en résulte une praticité, une fiabilité et une sécurité accrues. L'utilisateur est dispensé d'effectuer des sauvegardes de sa clé **15** privée.

[0103] La clé **15** étant fragmentée, aucun nœud **2** du réseau **1** primaire n'en dispose seul, et ne peut par conséquent reconstituer la clé **15** pour usurper l'identité de l'utilisateur et valider des transactions à sa place, au bénéfice de la confidentialité.

[0104] En outre, le procédé proposé permet, à partir d'un réseau **1** primaire sur lequel est déployée une blockchain **5** primaire, d'effectuer la signature décentralisée d'une transaction **TX** à destination d'une blockchain **5'** secondaire distincte de la blockchain **5** primaire, et déployée sur un réseau **1'** secondaire distinct du réseau **1** primaire.

[0105] La blockchain **5** primaire sert notamment de support au contrat intelligent **SC** dont les instructions, chargées dans l'enclave **8** du nœud **2P** de calcul, permettent d'effectuer les opérations nécessaires à la signature de la transaction **TX**. Comme évoqué, la blockchain **5** primaire sert également à l'inscription d'une trace **TRC** de cette transaction (une fois signée), bien que celle-ci soit destinée à la blockchain secondaire **5'**.

[0106] En d'autres termes, le procédé proposé permet d'offrir un service de portefeuille électronique décentralisé utilisable sur une blockchain **5'** déployée sur un réseau **1'** auquel l'émetteur **E** n'est pas nécessairement connecté.

Revendications

1. Procédé de signature numérique d'une transaction (**TX**) initiée par une unité de traitement informatique, dite émetteur (**E**), reliée à un réseau (**1**) pair-à-pair primaire 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 (**4**) primaire, cette transaction (**TX**) numérique étant destinée à être inscrite dans un nouveau bloc (**4'**) d'une chaîne (**5'**) de blocs secondaire mémorisée par réplication sur chaque nœud (**2'**) d'un réseau (**1'**) pair-à-pair secondaire, ce procédé comprenant les opérations suivantes :
- Générer la transaction (**TX**) par l'émetteur (**E**), cette transaction (**TX**) contenant un identifiant d'émetteur ;
- Sélectionner parmi le réseau (**1**) primaire 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**) ;
- Charger la transaction (**TX**) numérique dans l'enclave (**8**), via une ligne (**16**) de communication sécurisée ;
- Sélectionner parmi le réseau (**1**) primaire des nœuds (**2A**) sur lesquels sont stockés des fragments (**15.i**) d'une clé (**15**) cryptographique privée associée à l'identifiant d'émetteur contenu dans la transaction (**TX**) ;
- Charger lesdits fragments (**15.i**) dans l'enclave (**8**) du nœud (**2P**) de calcul:
- Dans l'enclave (**8**) du nœud (**2P**) de calcul :
o Reconstituer la clé (**15**) cryptographique privée à partir des fragments (**15.i**) ainsi chargés ;
o Signer numériquement la transaction (**TX**) par chiffrement de tout ou partie des informations de celle-ci au moyen de la clé (**15**) cryptographique privée ainsi reconstituée, pour former une transaction (**STX**) signée ;
o Transmettre au réseau (**1'**) secondaire la transaction (**STX**) signée ;
- Au sein d'au moins un nœud (**2'**) du réseau (**1'**) secondaire :
o Vérifier la transaction (**STX**) signée ;
o La transaction (**STX**) signée vérifiée, l'inscrire dans un nouveau bloc (**4'**) destiné à la chaîne (**5'**) de blocs secondaire ;
- Inscrire une trace de la transaction (**STX**) signée dans un nouveau bloc (**4**) destiné à la chaîne (**5**) de blocs primaire.

FIG.1

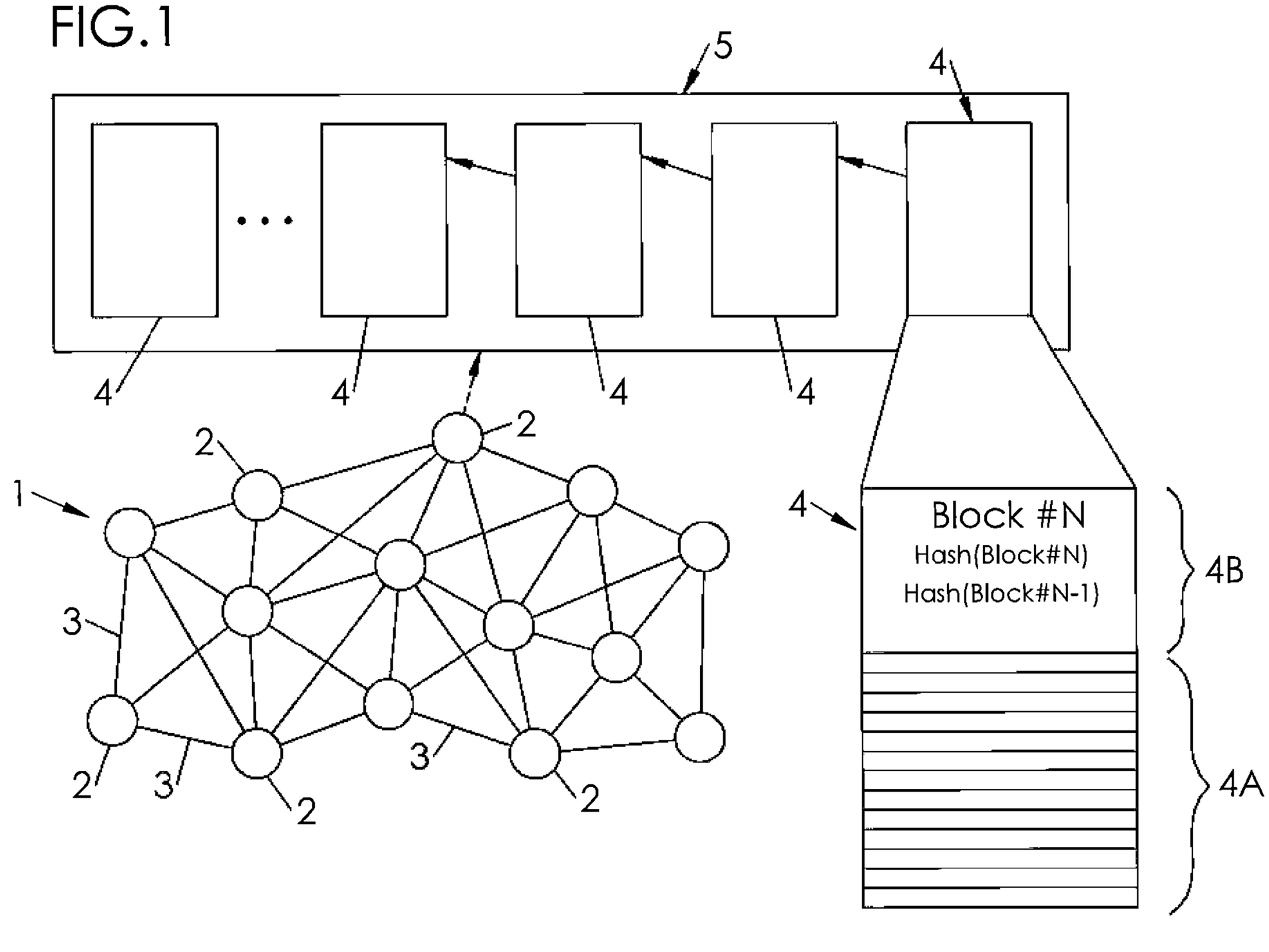
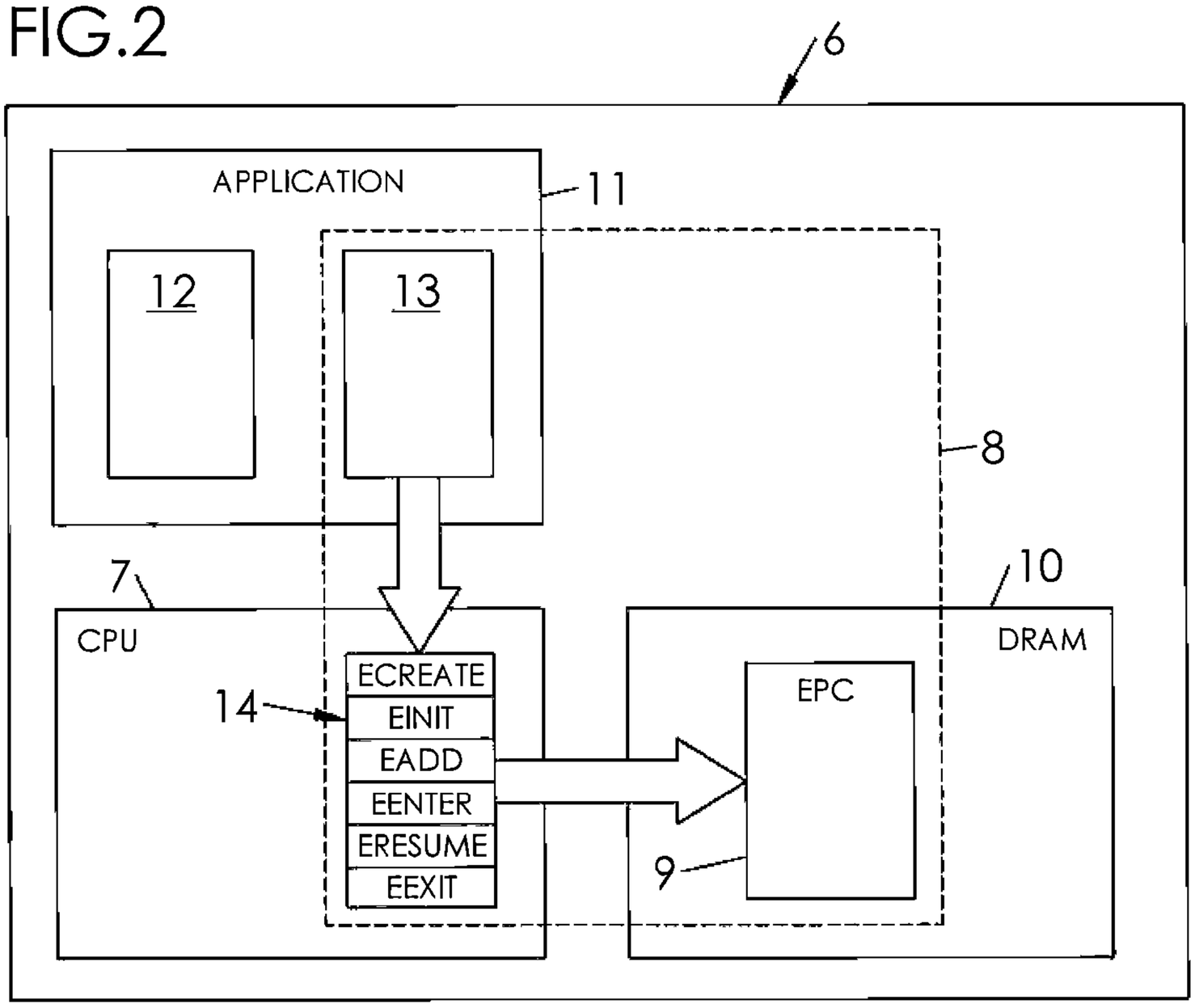

5
4
4
4
4
4
2
2
1
3
2
3
2
3
2
Block #N
Hash(Block#N)
Hash(Block#N-1)
4B
4A

FIG.2


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

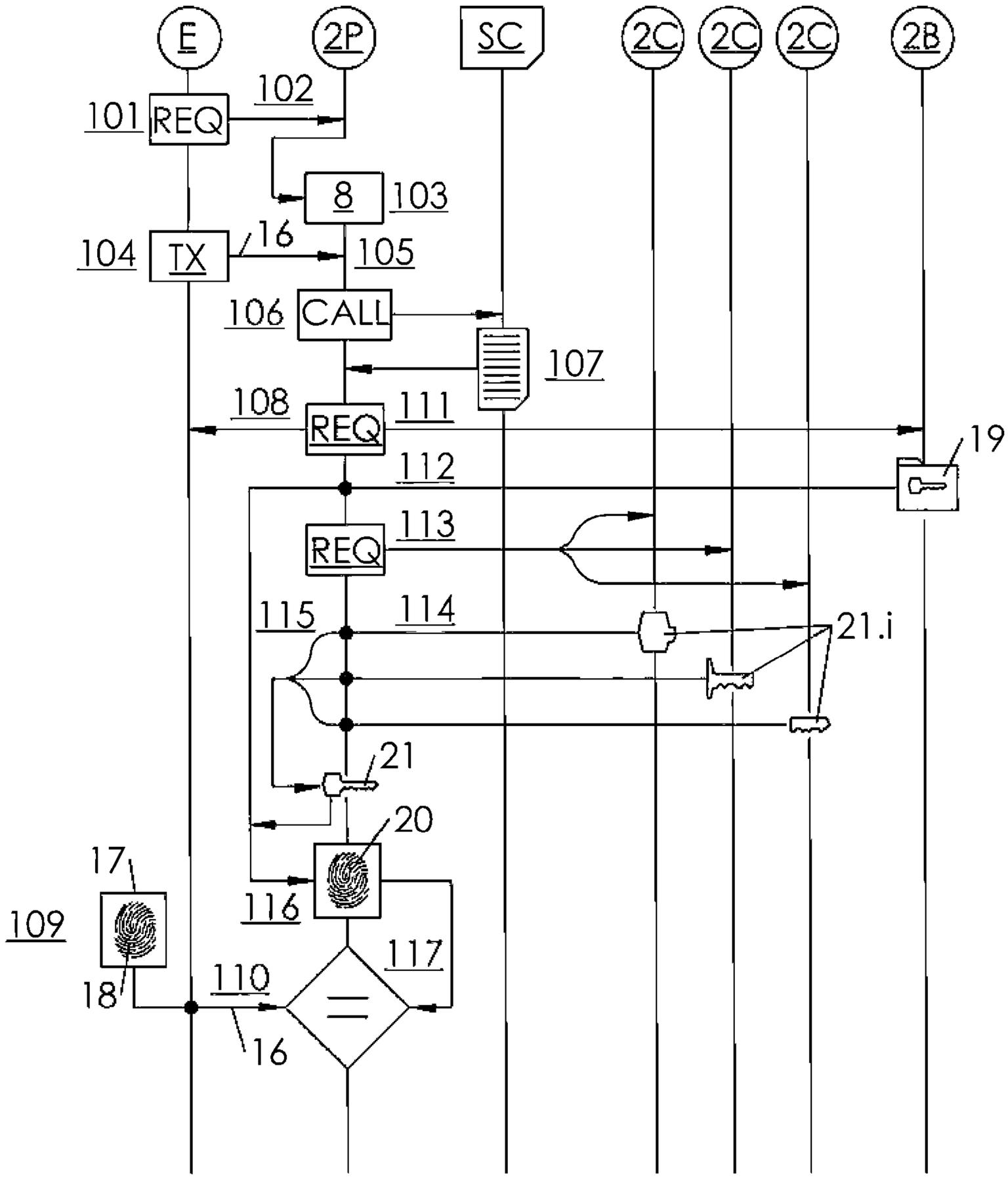

E
2P
SC
2C
2C
2C
2B
101
REQ
102
8
103
104
TX
16
105
106
CALL
107
108
REQ
111
19
112
113
REQ
115
114
21.i
21
20
17
109
116
117
110
18
16

FIG.3

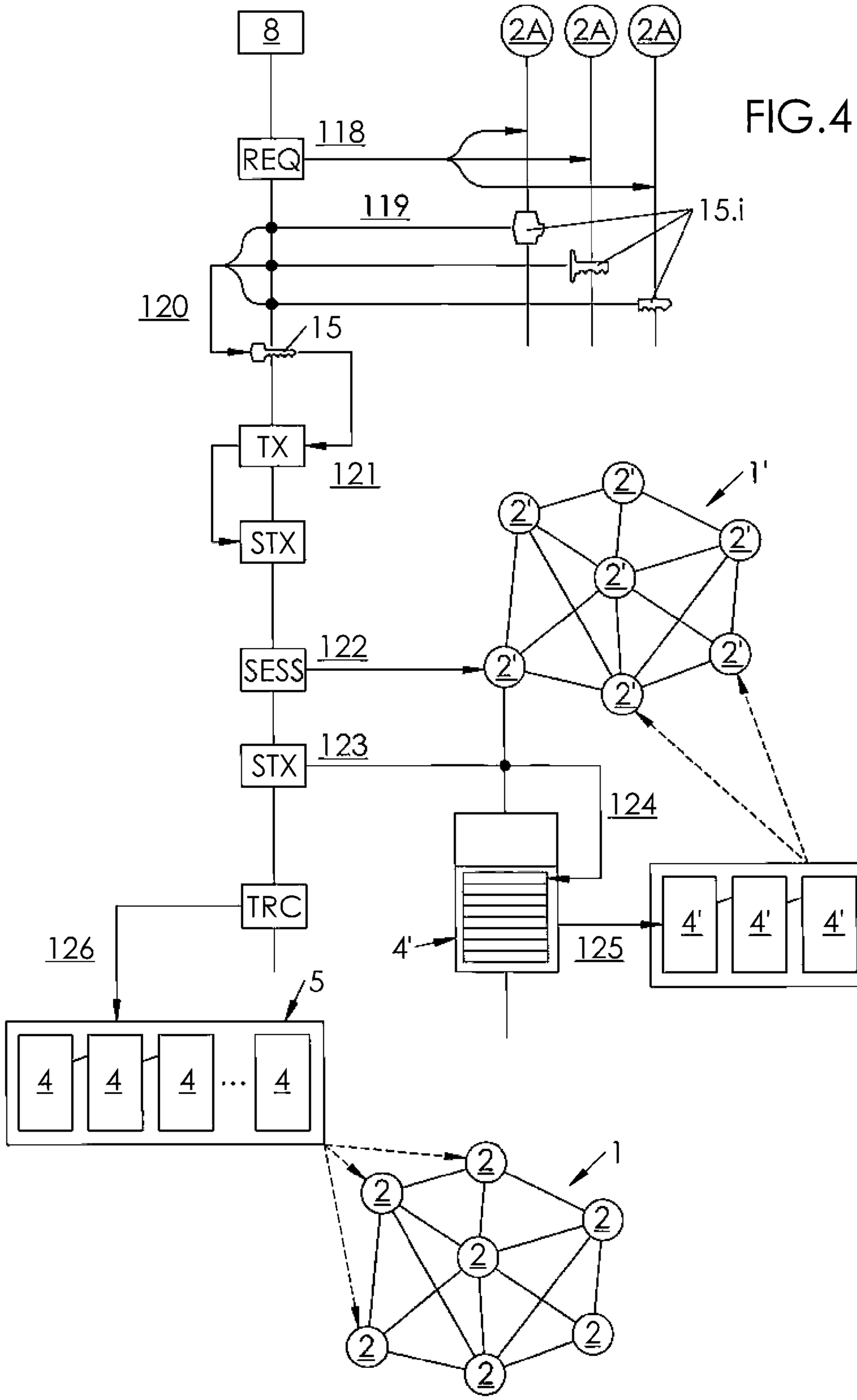

8
2A
2A
2A
118
REQ
119
15.i
120
15
TX
121
STX
2'
1'
122
SESS
123
STX
124
5'
TRC
126
4'
125
4'
4'
4'
4'
5
4
4
4
4
1
2

FIG.4