(19) CONFÉDÉRATION SUISSE
INSTITUT FÉDÉRAL DE LA PROPRIÉTÉ INTELLECTUELLE

(11) **CH 716 282 A2**

(51) Int. Cl.: ***G06F 21/62*** *(2013.01)*
H04L 9/08 *(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: 00754/19

(22) Date de dépôt: 07.06.2019

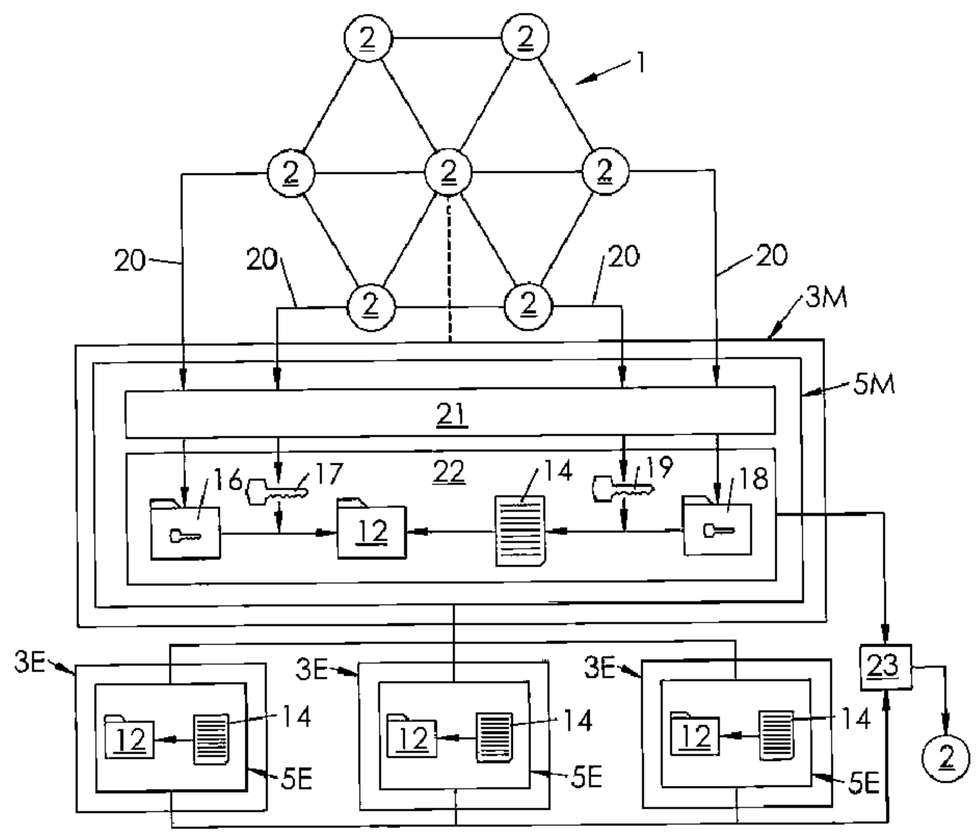
(43) Demande publiée: 15.12.2020

(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 traitement distribué, sous enclaves, de données informatiques chiffrées au moyen d'une application chiffrée.

(57) L'invention concerne un procédé de traitement de données (**12**) informatiques contenues dans un premier conteneur (**16**) chiffré, au moyen d'une application (**14**) tierce contenue dans un deuxième conteneur (**18**) chiffré, ce traitement étant conduit de manière distribuée au sein d'enclaves (**5M**, **5E**) cryptographiques reliées en réseau sécurisé.



Description

DOMAINE TECHNIQUE

[0001] L'invention a trait au domaine de l'informatique, et plus précisément au domaine du traitement des données informatiques au moyen d'une application.

ART ANTERIEUR

[0002] Un programme d'ordinateur comprend des instructions au moyen desquelles, par l'intermédiaire d'une unité de traitement, des opérations sont réalisées sur des données d'entrée (en anglais input), pour fournir, à partir de celles-ci, des données de sortie ou résultat (en anglais output).

[0003] Le traitement de données en masse (en anglais big data) tend à s'industrialiser. Des programmes spécialisés scrutent les flux de données (tout particulièrement les données personnelles des usagers) circulant sur la Toile, procèdent à des analyses de ces données, effectuent des calculs (notamment statistiques) puis stockent leurs résultats sur des bases de données aux fins d'une exploitation ultérieure (notamment commerciale).

[0004] Pour éviter que leurs données personnelles soient exploitées sans leur autorisation, ou pour conserver un relatif anonymat sur la Toile, certains usagers prennent la précaution de les chiffrer.

[0005] Certaines données font d'ailleurs (ou devraient faire) l'objet d'un chiffrement systématique, en raison de leur caractère confidentiel (typiquement les données de santé ou les données financières), les instances chargées de leur gestion (hôpitaux, banques) étant soumises à des obligations professionnelles de secret, et responsables à ce titre de l'usage qui pourrait en être fait par des tiers non autorisés.

[0006] Les données chiffrées sont cependant inaccessibles - et donc inexploitables - pour qui ne dispose pas d'une clé cryptographique appropriée permettant leur déchiffrement.

[0007] Pourtant il existe de nombreux cas où le propriétaire des données (typiquement un patient d'un institut médical ou un client d'une banque) peut souhaiter, sous conditions, autoriser un tiers intéressé à procéder à un traitement des données, sans pour autant communiquer à ce tiers la clé de déchiffrement, ni l'autoriser à accéder pleinement aux données lorsque seule une partie de ces données est susceptible de l'intéresser.

[0008] Inversement, le tiers intéressé peut souhaiter conserver secrètes, vis-à-vis d'autres tiers encore, ses propres routines d'analyse, et chiffrer les données programmatiques de ses applications, dont le développement est le fruit d'un savoir-faire précieux.

[0009] L'invention vise à offrir une solution à ces problèmes, en proposant un procédé de traitement de données chiffrées qui permette à la fois à une application de traiter ces données sans affecter leur confidentialité, et à l'application elle-même de demeurer confidentielle.

RESUME DE L'INVENTION

[0010] Il est proposé un procédé de traitement de données informatiques au moyen d'une application tierce se présentant sous forme d'un ou plusieurs fichier(s) de données programmatiques, ce procédé comprenant les opérations consistant à :

- Chiffrer les données à traiter pour former un premier conteneur crypté de données ;
- Mémoriser séparément le premier conteneur crypté et une première clé cryptographique de déchiffrement de celui-ci ;
- Chiffrer le(s) fichier(s) de données programmatiques de l'application tierce pour former un deuxième conteneur crypté ;
- Mémoriser séparément le deuxième conteneur et une deuxième clé cryptographique de déchiffrement associée ;
- A réception d'une requête de traitement des données par l'application tierce, sélectionner une unité de traitement informatique, dite maîtresse, dans laquelle est implémenté un environnement d'exécution sécurisé par cryptographie, dit enclave ;
- Instancier cette enclave, dite enclave maîtresse ;
- Charger dans l'enclave maîtresse le premier conteneur et la première clé cryptographique de déchiffrement associée, ainsi que le deuxième conteneur et la deuxième clé cryptographique de déchiffrement associée ;
- Sélectionner une ou plusieurs unités de traitement informatique, dites esclaves, reliées en réseau à l'unité de traitement maîtresse, et dans chacune desquelles est implémentée une enclave dite esclave ;
- Instancier les enclaves esclaves ;
- Etablir une connexion cryptée entre l'enclave maîtresse et les enclaves esclaves ;
- Dans l'enclave maîtresse, déchiffrer les données du premier du premier conteneur au moyen de la première clé cryptographique de déchiffrement associée, et déchiffrer le(s) fichier(s) de données programmatiques de l'application ou des applications tierce(s) contenu(s) dans le deuxième conteneur au moyen de la deuxième clé cryptographique de déchiffrement associée ;
- Partager, à partir de l'enclave maîtresse, tout ou partie des données à traiter, issues du premier conteneur, vers les enclaves esclaves ;
- Partager, à partir de l'enclave maîtresse, tout ou partie du ou des fichier(s) de données programmatiques de l'application vers les enclaves esclaves ;

– Partager, à partir de l'enclave maîtresse et vers les enclaves esclaves, l'exécution de l'application tierce sur les données à traiter ;
– Hors des enclaves, mémoriser le résultat de cette exécution sur un emplacement mémoire prédéfini.
– Hors de l'enclave, mémoriser le résultat de cette exécution sur un emplacement mémoire prédéfini.

[0011] Il peut être prévu, préalablement à l'opération de mémorisation du résultat hors enclave, une opération d'agrégation du résultat dans l'enclave maîtresse.

BREVE DESCRIPTION DES FIGURES

[0012] 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 simplifié illustrant un réseau informatique dont au moins un nœud est équipé d'une unité de traitement sur laquelle est implémentée une enclave ;

La **FIG.2** est un schéma fonctionnel illustrant le stockage de données chiffrées à traiter et d'une application également chiffrée ;

La **FIG.3** est un schéma fonctionnel illustrant le traitement des données par l'application.

DESCRIPTION DETAILLEE DE L'INVENTION

[0013] Le procédé proposé vise à traiter, de manière confidentielle, des données informatiques au moyen d'une application tierce (c'est-à-dire provenant d'un éditeur distinct de l'émetteur des données).

[0014] Ce traitement peut être conduit au sein d'un réseau **1** informatique (qui, comme illustré, peut être du type pair-à-pair) comprenant une pluralité de nœuds **2** (ordinateurs, serveurs).

[0015] Le procédé de traitement proposé exploite notamment la technologie de l'environnement d'exécution sécurisé ou, en terminologie anglo-saxonne, du trusted execution environment (TEE).

[0016] Par référence à la **FIG.1**, un environnement d'exécution sécurisé (Trusted execution environment ou TEE) est, au sein d'une unité **3** de traitement informatique équipant un nœud **2** et pourvue d'un processeur ou CPU (Central Processing Unit) **4**, un espace **5** 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.

[0017] 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).

[0018] 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.

[0019] 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.

[0020] Pour résumer, en référence à la **FIG.1**, une enclave **5** comprend, en premier lieu, une zone **6** 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 **6** est implémentée au sein d'une partie de la mémoire vive dynamique (DRAM) **7** allouée au processeur **4**, et à laquelle les applications ordinaires (notamment le système d'exploitation) n'ont pas accès.

[0021] L'enclave **5** comprend, en deuxième lieu, des clés cryptographiques employées pour chiffrer ou signer à la volée les données sortant de l'EPC **6**, ce grâce à quoi l'enclave **5** 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 **6**).

[0022] Pour pouvoir exploiter une telle enclave **5**, une application **8** doit être segmentée en, d'une part, une ou plusieurs parties **9** non sécurisées (en anglais untrusted part(s)), et, d'autre part, une ou plusieurs parties **10** sécurisées (en anglais trusted part(s)).

[0023] Seuls les processus induits par la (les) partie(s) **10** sécurisée(s) de l'application **8** peuvent accéder à l'enclave **5**. Les processus induits par la (les) partie(s) **9** non sécurisée(s) ne peuvent pas accéder à l'enclave **5**, c'est-à-dire qu'ils ne peuvent pas dialoguer avec les processus induits par la (les) partie(s) **10** sécurisée(s).

[0024] La création (également dénommée instanciation) de l'enclave **5** et le déroulement de processus en son sein sont commandés via un jeu **11** d'instructions particulières exécutables par le processeur **4** et appelées par la (les) partie(s) **10** sécurisée(s) de l'application **8**.

[0025] Parmi ces instructions :

- ECREATE commande la création d'une enclave **5** ;
- EINIT commande l'initialisation de l'enclave **5** ;
- EADD commande le chargement de code dans l'enclave **5** ;
- ERESUME commande une nouvelle exécution de code dans l'enclave **5** ;
- EEXIT commande la sortie de l'enclave **5**, typiquement à la fin d'un processus exécuté dans l'enclave **5**.

[0026] On a, sur la **FIG.1**, représenté de manière fonctionnelle l'enclave **5** sous la forme d'un bloc (en traits pointillés) englobant la partie **10** sécurisée de l'application **8**, le jeu **11** d'instructions du processeur **4**, et l'EPC **6**. Cette représentation n'est pas réaliste ; elle vise simplement à regrouper visuellement les principaux éléments qui composent ou exploitent l'enclave **5**.

[0027] Nous expliquerons ci-après comment sont exploitées les enclaves.

[0028] Comme cela est visible sur la **FIG.1**, on a représenté des données **12** à traiter sous forme d'un dossier. Ces données **12** peuvent se présenter sous forme d'un ou plusieurs fichiers informatiques. Les données **12** peuvent être structurées, par ex. sous forme de fichiers de bases de données, de formulaires, de tableurs, de documents. Il peut également s'agir de données brutes, non structurées.

[0029] Ces données **12** sont créées, ou temporairement stockées, en clair (c'est-à-dire sous forme non chiffrée), au sein d'un terminal **13** dit émetteur, relié à un réseau **1**.

[0030] Le traitement des données est destiné à être réalisé au moyen d'une application **14** tierce qui se présente sous forme d'un ou plusieurs fichiers de données programmatiques (c'est-à-dire de données représentant des instructions d'un programme d'ordinateur).

[0031] Les données programmatiques peuvent être des lignes de code en langage haut niveau (code source). Dans ce cas, les données programmatiques se présentent typiquement sous forme de fichiers texte. Elles peuvent également se présenter sous forme d'un ou plusieurs fichiers de code intermédiaire (bytecode), ou encore sous forme d'un ou plusieurs fichiers de code exécutable directement par un processeur.

[0032] L'application **14** est créée (ou éditée), ou temporairement stockée, en clair (c'est-à-dire sous forme non chiffrée), au sein d'un terminal **15** dit éditeur, relié à un réseau **1** (qui peut être le même que celui auquel est relié l'émetteur **13**).

[0033] Avant d'être stockées sur le réseau **1** en vue d'un futur traitement, les données **12** sont d'abord chiffrées pour former un premier conteneur **16** crypté de données. Le déchiffrement du premier conteneur **16** crypté ne peut être réalisé qu'au moyen d'une première clé **17** cryptographique de déchiffrement. Selon un mode particulier de réalisation, une seule clé **17** est utilisée à la fois pour le chiffrement et le déchiffrement. Dans ce cas, le chiffrement est dit symétrique.

[0034] Dans l'exemple illustré, le chiffrement est réalisé au sein du terminal **13** émetteur.

[0035] Une fois les données **12** chiffrées, le premier conteneur **16** et la première clé **17** de déchiffrement sont mémorisés séparément (ici sur des nœuds **2** distincts du réseau **1**).

[0036] De même, avant d'être stockée sur le réseau **1** en vue d'une future utilisation, l'application **14** est d'abord chiffrée pour former un deuxième conteneur **18** crypté. Le déchiffrement du deuxième conteneur **18** crypté ne peut être réalisé qu'au moyen d'une deuxième clé **19** cryptographique de déchiffrement. Selon un mode particulier de réalisation, une seule clé **19** est utilisée à la fois pour le chiffrement et le déchiffrement. Dans ce cas, le chiffrement est dit symétrique.

[0037] Dans l'exemple illustré, le chiffrement est réalisé au sein du terminal **15** éditeur.

[0038] Une fois l'application **14** chiffrée, le deuxième conteneur **18** et la deuxième clé **19** de déchiffrement sont mémorisés séparément (ici sur des nœuds **2** distincts du réseau **1**).

[0039] Le traitement des données **12** par l'application **14** peut être sollicité, sur requête, par un utilisateur (qui peut être l'éditeur **15**). Dans ce cas, une requête de traitement des données **12** par l'application **14** est émise par l'utilisateur.

[0040] Pour réaliser ce traitement, une première opération consiste, à réception (par le réseau **1**) de la requête de traitement des données **12**, à sélectionner une unité **3M** de traitement informatique (équipant par ex. un nœud **2** du réseau **1**), dite maîtresse, dans laquelle est implémentée une enclave **5M**, dite maîtresse.

[0041] Une deuxième opération consiste à instancier l'enclave **5** maîtresse.

[0042] Une troisième opération consiste à charger dans l'enclave **5** maîtresse (de préférence via des lignes **20** de communication sécurisées, typiquement utilisant le protocole Transport Layer Security ou TLS) :

– Le premier conteneur **16** et la première clé **17** cryptographique de déchiffrement qui lui est associée ;
– Le deuxième **18** conteneur et la deuxième clé **19** cryptographique de déchiffrement qui lui est associée.

[0043] A cet effet, l'enclave **5** maîtresse est avantageusement pourvue, dès son instanciation, d'une émulation d'interface **21** de communication supportant le protocole choisi (ici TLS) pour l'échange des données sécurisées.

[0044] Une quatrième opération consiste à sélectionner une ou plusieurs unités **3E** de traitement informatique (équipant par ex. un ou plusieurs nœuds **2** du réseau **1**), dites esclaves, dans chacune desquelles est implémentée une enclave **5E**, dite esclave.

[0045] Une cinquième opération consiste à instancier les enclaves **5E** esclaves, une sixième opération consistant à établir une connexion cryptée entre l'enclave **5M** maîtresse et les enclaves **5E** esclaves.

[0046] Une septième opération consiste, dans l'enclave **5M** maîtresse, à déchiffrer les données du premier conteneur **16** au moyen de la première clé **17** de déchiffrement. A l'issue de cette opération, les données **12** sont accessibles en clair dans l'enclave **5** (mais pas depuis l'extérieur de celle-ci).

[0047] A cet effet, l'enclave **5M** maîtresse est avantageusement pourvue d'un module **22** de déchiffrement.

[0048] Une huitième opération consiste, dans l'enclave **5**, à déchiffrer le(s) fichier(s) de données programmatiques de l'application **14** tierce contenu(s) dans le deuxième conteneur **18**, au moyen de la deuxième clé **19** de déchiffrement. Cette opération est ici réalisée par le module **22** de déchiffrement. A l'issue de cette opération, l'application **14** est accessible en clair dans l'enclave **5** (mais pas depuis l'extérieur de celle-ci).

[0049] Une neuvième opération consiste à partager, à partir de l'enclave **5M** maîtresse, tout ou partie des données **12** à traiter, issues du premier conteneur **16**, vers les enclaves **5E** esclaves.

[0050] Une dixième opération consiste à partager, à partir de l'enclave **5M** maîtresse, tout ou partie du ou des fichier(s) de données programmatiques de l'application **14** tierce, vers les enclaves **5E** esclaves.

[0051] Ces partages s'effectuent via les liaisons sécurisées établies entre les enclaves **5M**, **5E**.

[0052] Une onzième opération consiste à partager, à partir de l'enclave **5M** et vers les enclaves **5E** esclaves, l'exécution de l'application **14** tierce sur les données **12** à traiter.

[0053] Cette exécution est avantageusement conduite, au sein de chaque enclave **5M**, **5E**, par un module de calcul.

[0054] Chaque enclave **5E** esclave peut exécuter une partie des instructions de l'application **14**, en sorte qu'ensemble toutes les enclaves **5M**, **5E** exécutent toutes les instructions de l'application **14**.

[0055] La division des tâches (correspondant aux instructions de l'application **14**) est avantageusement pilotée par l'enclave **5M** maîtresse, qui sous-traite aux enclaves **5E** esclaves l'exécution de toute ou partie des tâches.

[0056] Lorsque l'application **14** se présente sous forme d'un (ou plusieurs) fichier(s) exécutable(s), cette exécution peut être conduite directement. Lorsque l'application **14** se présente sous forme de code source ou de bytecode, il convient de le compiler au préalable. Dans ce cas, on peut charger dans l'enclave **5M** maîtresse, préalablement à l'exécution, un programme de compilation.

[0057] L'exécution de l'application **14** sur les données produit un résultat **23**. Ce résultat **23** est mémorisé, hors des enclaves **5M**, **5E**, sur un emplacement mémoire prédéfini (au préalable, il peut être prévu une opération d'agrégation du résultat **23** dans l'enclave **8M** maîtresse). Dans l'exemple illustré, le résultat **23** est transmis par un nœud **2** du réseau.

[0058] Grâce à ce procédé, les données **12** peuvent être analysées sans être disponibles en clair hors des enclaves **5M**, **5E**, auxquelles les tiers n'ont pas accès. En outre, les données **12** ne sont pas non plus communiquées à l'éditeur **15** pour qu'il procède lui-même à l'exécution de son application **14**. L'éditeur **15** ne dispose donc pas des données **12** : il ne dispose que du résultat fourni par l'exécution de son application **14**. Il en résulte le maintien de la confidentialité des données **12** y compris vis-à-vis des tiers qui souhaitent (et peuvent être autorisés à) les analyser.

[0059] De même, l'application **14** n'est pas non plus connue des tiers (ou copiable par eux), en dépit de sa disponibilité pour le traitement des données **12** dans les enclaves **5M**, **5E**. Il en résulte que l'éditeur **15** est protégé de la copie tout en pouvant traiter les données **12** dès lors qu'il y est autorisé.

[0060] Le fait de faire traiter les données **12** de manière distribuée, au sein d'un réseau d'enclaves **5M**, **5E**, permet en outre de disposer de ressources de calcul et/ou de stockage (temporaire) suffisantes (tout en maintenant la confidentialité) lorsque les données **12** et/ou l'application **14** sont de taille importante.

Revendications

1. Procédé de traitement de données (**12**) informatiques au moyen d'une application (**14**) tierce se présentant sous forme d'un ou plusieurs fichier(s) de données programmatiques, ce procédé comprenant les opérations consistant à :
   - Chiffrer les données (**12**) à traiter pour former un premier conteneur (**16**) crypté de données ;
   - Mémoriser séparément le premier conteneur (**16**) crypté et une première clé (**17**) cryptographique de déchiffrement de celui-ci ;
   - Chiffrer le(s) fichier(s) de données programmatiques de l'application (**14**) tierce pour former un deuxième conteneur (**18**) crypté ;
   - Mémoriser séparément le deuxième conteneur (**18**) et une deuxième clé (**19**) cryptographique de déchiffrement associée ;
   - A réception d'une requête de traitement des données (**12**) par l'application (**14**) tierce, sélectionner une unité (**3M**) de traitement informatique, dite maîtresse, dans laquelle est implémenté un environnement d'exécution sécurisé par cryptographie, dit enclave (**5**) ;
   - Instancier cette enclave (**5**), dite enclave maîtresse ;
   - Charger dans l'enclave (**5**) maîtresse le premier conteneur (**16**) et la première clé (**17**) cryptographique de déchiffrement associée, ainsi que le deuxième conteneur (**18**) et la deuxième clé (**19**) cryptographique de déchiffrement associée ;
   - Sélectionner une ou plusieurs unités (**3E**) de traitement informatique, dites esclaves, reliées en réseau à l'unité de traitement (**3M**) maîtresse, et dans chacune desquelles est implémentée une enclave (**5E**) dite esclave ;
   - Instancier les enclaves (**5E**) esclaves ;
   - Etablir une connexion cryptée entre l'enclave (**5M**) maîtresse et les enclaves (**5E**) esclaves ;
   - Dans l'enclave (**5M**) maîtresse, déchiffrer les données du premier du premier conteneur (**16**) au moyen de la première clé (**17**) cryptographique de déchiffrement associée, et déchiffrer le(s) fichier(s) de données programmatiques de l'application ou des applications tierce(s) contenu(s) dans le deuxième conteneur (**18**) au moyen de la deuxième clé (**19**) cryptographique de déchiffrement associée ;
   - Partager, à partir de l'enclave (**5M**) maîtresse, tout ou partie des données (**12**) à traiter, issues du premier conteneur (**16**), vers les enclaves (**5E**) esclaves ;
   - Partager, à partir de l'enclave (**5M**) maîtresse, tout ou partie du ou des fichier(s) de données programmatiques de l'application vers les enclaves (**5E**) esclaves ;
   - Partager, à partir de l'enclave (**5M**) maîtresse et vers les enclaves (**5E**) esclaves, l'exécution de l'application tierce sur les données (**12**) à traiter ;
   - Hors des enclaves (**5M**, **5E**), mémoriser le résultat (**24**) de cette exécution sur un emplacement mémoire prédéfini.
2. Procédé selon la revendication 1, qui comprend, préalablement à l'opération de mémorisation du résultat (**24**) hors enclave, une opération d'agrégation du résultat (**24**) dans l'enclave (**5M**) maîtresse.

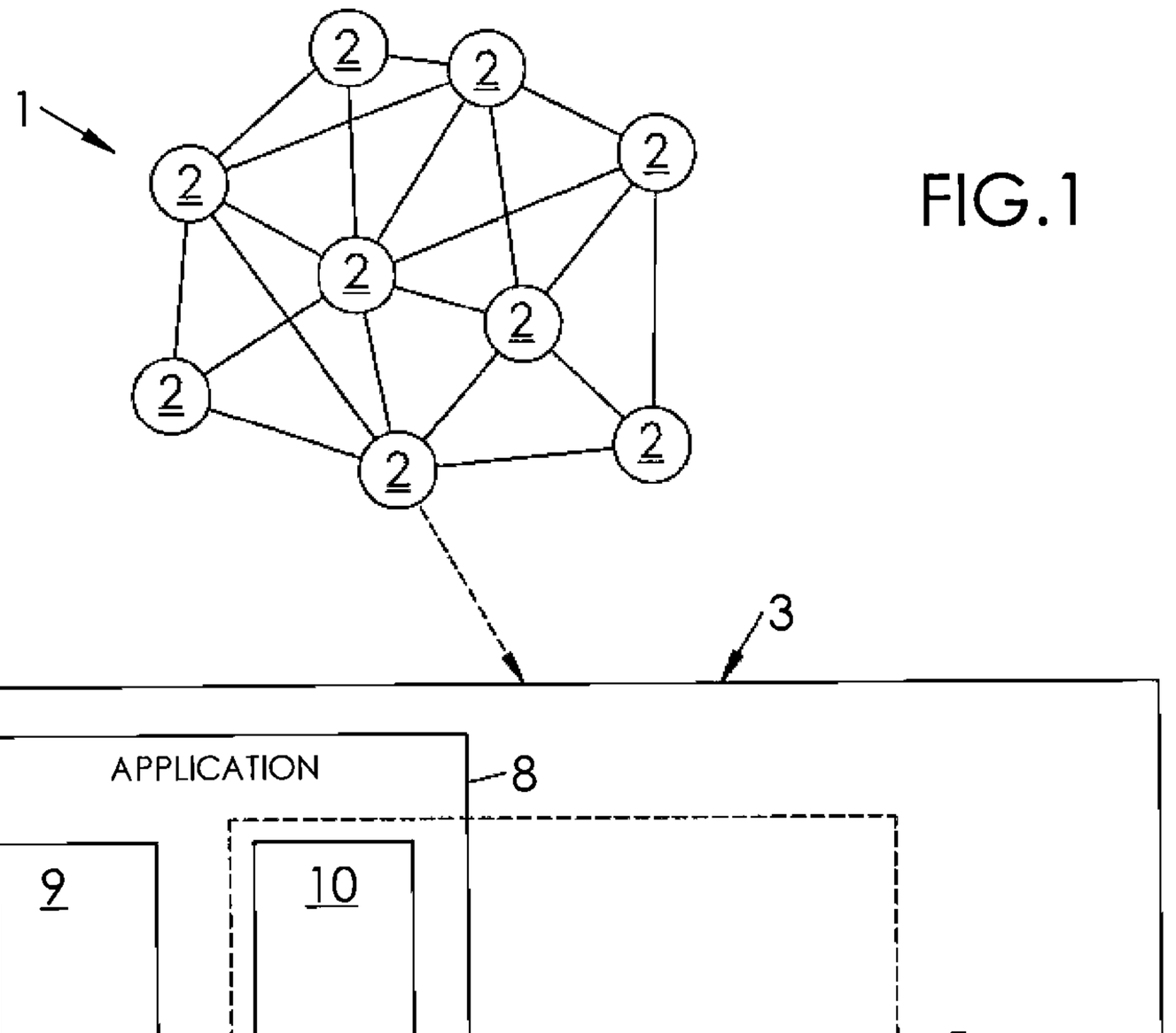

1
2
2
2
2
2
2
2
2
3
APPLICATION
8
9
10
5
4
7
CPU
DRAM
ECREATE
EINIT
EADD
EENTER
ERESUME
EEXIT
11
EPC
6

FIG.1

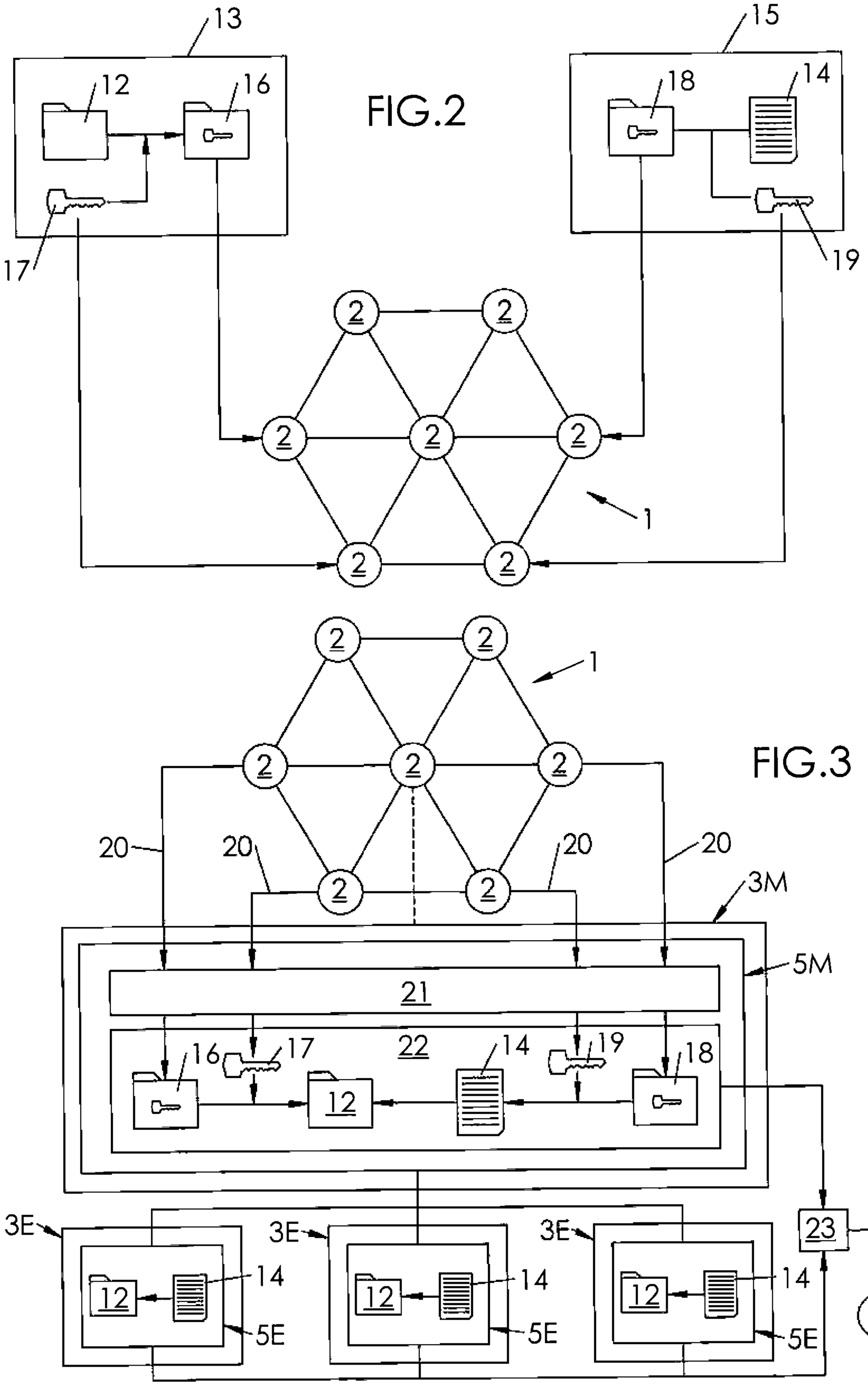


FIG.2

FIG.3