(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** **3 143 244**

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **22 12896**

(51) Int Cl[8] : ***H 04 L 9/32*** *(2023.01), G 06 F 21/33, 8/61*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 07.12.22.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 14.06.24 Bulletin 24/24.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Se reporter à la fin du présent fascicule*

(60) **Références à d'autres documents nationaux apparentés :**

◯ **Demande(s) d'extension :**

(71) **Demandeur(s) :** *ELECTRICITE DE FRANCE Société anonyme* — FR.

(72) **Inventeur(s) :** NIAMKE Richard et JOSIEN Eric.

(73) **Titulaire(s) :** ELECTRICITE DE FRANCE Société anonyme.

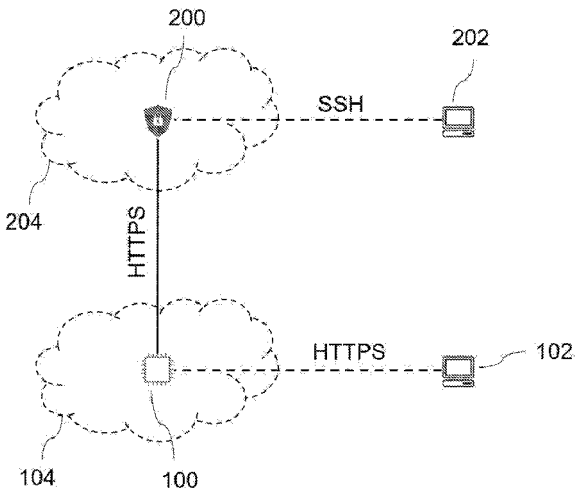
(74) **Mandataire(s) :** Plasseraud IP.

(54) Mécanisme d'autorisation pour l'utilisation d'un procédé logiciel avec sécurisation du code source.

(57) Il est proposé un procédé d'exécution d'un code logiciel protégé, le procédé étant mis en œuvre par une application contenant un régénérateur d'un certificat client préalablement signé par une autorité de certification et un code offusqué correspondant au code logiciel protégé, le procédé comprenant, sur exécution de l'application :

- régénérer, à l'aide du régénérateur, le certificat client et émettre une requête comprenant le certificat client, et
- lorsque, suite à l'émission de la requête, une connexion est établie avec un serveur et une réponse indicative de validité du certificat client est reçue via la connexion établie, exécuter le code offusqué dans le conteneur logiciel, sinon, arrêter l'exécution de l'application sans exécuter le code offusqué.

Figure de l'abrégé : Figure 1



FR 3 143 244 - A1

Description

Titre de l'invention : Mécanisme d'autorisation pour l'utilisation d'un procédé logiciel avec sécurisation du code source

Domaine technique

[0001] La présente divulgation relève du domaine de la sécurité des systèmes d'information.

[0002] Plus particulièrement, la présente divulgation porte sur un procédé d'exécution d'un code logiciel protégé, sur un programme informatique et sur un support d'enregistrement correspondants.

Technique antérieure

[0003] En matière d'autorisation d'utilisation de logiciel propriétaire, l'état de l'art repose sur deux principes :

- le code logiciel protégé, c'est-à-dire sécurisé, peut être installé sur des serveurs sur site, autrement dit en local, et l'autorisation d'utilisation est alors soumise à licence, c'est-à-dire qu'une clé ou un fichier permet de débloquer le fonctionnement du code logiciel, ou
- le code logiciel protégé est accessible sous forme de service hébergé par exemple dans un réseau informatique en nuage, ou « cloud », et l'opérateur offre tous les mécanismes d'autorisation nécessaires pour permettre son exécution sans donner l'accès au code en lui-même.

[0004] De nombreux travaux de recherche, en particulier dans le domaine de l'apprentissage automatique, conduisent à produire des codes informatiques à l'aide de langages de programmation qui ne se compilent pas, tels que Python ou R.

[0005] Pour sécuriser de tels codes, une solution possible est de ne pas les diffuser. Par exemple, il peut être prévu que le propriétaire d'un code logiciel protégé recueille des données qui lui sont transmises par un utilisateur, traite ces données à l'aide du code logiciel protégé qu'il détient, et transmette uniquement le résultat de ce traitement à l'utilisateur.

[0006] Cette façon de procéder sous forme de service présente l'inconvénient pour le propriétaire du code protégé de maintenir une infrastructure de stockage et une infrastructure de calcul pour pouvoir traiter les données de ses partenaires et clients. De plus, s'il s'agit de données sensibles, le partage peut s'avérer fastidieux au vu de la réglementation en vigueur.

[0007] Il existe donc un besoin pour un mécanisme permettant d'autoriser l'usage par un utilisateur d'un code logiciel protégé qui peut être écrit en un langage non compilable, sans pour autant divulguer à l'utilisateur le contenu des lignes de code et sans requérir le maintien d'infrastructures de stockage et de calcul.

Résumé

[0008] La présente divulgation vient améliorer la situation.

[0009] Il est proposé un procédé d'exécution d'un code logiciel protégé, le procédé étant mis en œuvre par une application contenant un régénérateur d'un certificat client préalablement signé par une autorité de certification et un code offusqué correspondant au code logiciel protégé, le procédé comprenant, sur exécution de l'application :

- régénérer, à l'aide du régénérateur, le certificat client et émettre une requête comprenant le certificat client, et
- lorsque, suite à l'émission de la requête, une connexion est établie avec un serveur et une réponse indicative de validité du certificat client est reçue via la connexion établie, exécuter le code offusqué dans le conteneur logiciel, sinon, arrêter l'exécution de l'application sans exécuter le code offusqué.

[0010] Le certificat client est régénéré une seule fois par exécution de l'application, et peut être à nouveau régénéré et réutilisé lors d'une exécution ultérieure de l'application. Le certificat client permet d'accorder une autorisation d'utilisation du code protégé qui peut être limitée en particulier dans le temps et une communication sécurisée avec le serveur d'autorisation. L'accès au code logiciel protégé est empêché par son offuscation. L'application peut être, par exemple, un conteneur logiciel qui peut aussi bien être stocké dans une infrastructure locale que dans un « cloud » et qui ne nécessite pas d'être fourni en tant que service.

[0011] Il a été constaté que le procédé proposé n'entraîne aucun impact de performance sur l'exécution du code offusqué.

[0012] La mise en place d'un mécanisme d'autorisation sécurisé et fiable, avec la possibilité de contrôler l'exécution du code protégé à distance, permet de mettre en place une gestion fine de l'autorisation d'exécution. Par exemple il est possible d'accorder une autorisation pour un nombre limité d'exécutions sur une période limitée, le premier arrivant à échéance. De plus un serveur dispose généralement d'un journal de connexions qui permet de recenser, au niveau du serveur, toutes les exécutions du code protégé ayant été mises en œuvre au niveau d'applications distribuées à un ensemble de clients.

[0013] Optionnellement, le procédé comprend en outre :

- effacer le certificat client régénéré immédiatement après l'émission de la requête.

[0014] Ceci renforce la sécurité du mécanisme d'autorisation en empêchant que le client ou qu'une tierce partie puisse accéder au certificat client.

[0015] Optionnellement, le régénérateur de certificat est offusqué.

[0016] Ceci renforce la sécurité du mécanisme d'autorisation en empêchant que le client ou qu'une tierce partie puisse, par l'analyse du régénérateur de certificat, se procurer une copie du certificat client.

[0017] Optionnellement, l'application est gérée par un conteneur logiciel dans un réseau informatique en nuage.

[0018] Contrairement aux solutions connues, le mécanisme d'autorisation proposé ne dépend pas d'un composant matériel particulier et est donc applicable à une utilisation du code protégé dans un réseau décentralisé.

[0019] Optionnellement, la requête est une requête HTTPS.

[0020] De manière générale, il est souhaitable que chaque liaison de données utile à la mise en œuvre du procédé soit une liaison sécurisée, de manière à empêcher toute interception de flux de données et, plus particulièrement, toute interception du certificat client. Ces liaisons sécurisées incluent évidemment la liaison de données entre l'application et le serveur, mais aussi, potentiellement, des liaisons de données au sein du système informatique du client ou au sein du système informatique auquel est rattaché le serveur.

[0021] Optionnellement, l'exécution de l'application est déclenchée par un logiciel de planification de tâches.

[0022] Ceci permet notamment de forcer un renouvellement périodique de l'autorisation d'exécution du code protégé, par exemple tous les jours, et éventuellement de manière transparente pour l'utilisateur final.

[0023] Il est également proposé un programme informatique comportant des instructions pour la mise en œuvre du procédé ci-avant lorsque ce programme est exécuté par un processeur.

[0024] Il est également proposé un support d'enregistrement non transitoire lisible par un ordinateur sur lequel est enregistré un programme pour la mise en œuvre du procédé ci-avant lorsque ce programme est exécuté par un processeur.

Brève description des dessins

[0025] D'autres caractéristiques, détails et avantages apparaîtront à la lecture de la description détaillée ci-après, et à l'analyse des dessins annexés, sur lesquels :

Fig. 1

[0026] [Fig.1] représente une architecture réseau mettant en communication des systèmes informatiques pour une mise en œuvre d'un code protégé.

Fig. 2

[0027] [Fig.2] illustre par un ordinogramme un procédé d'exécution d'un code logiciel protégé, selon un exemple de réalisation.

Fig. 3

[0028] [Fig.3] illustre par un ordinogramme un procédé de validation et d'autorisation pouvant être implémenté côté serveur pour une autorisation d'exécution d'un code logiciel protégé par une application côté client, selon un exemple de réalisation.

Description des modes de réalisation

[0029] L'offuscation de code est une technique connue qui permet de rendre du code informatique illisible par l'humain tout en gardant son « exécutabilité » par la machine.

[0030] Pour ce qui est de l'autorisation d'utilisation d'un code offusqué, plusieurs mécanismes de licence sont connus : par NTP (Network Time Protocol), ou par numéro de série de disque, ou encore par adresse MAC. Ces mécanismes de licence ne sont pas pleinement satisfaisants. Le protocole NTP n'est pas un protocole sécurisé. De plus, il est facile, avec peu de connaissance réseau, de détourner l'adresse d'un serveur NTP vers un autre serveur et de faire en sorte de pouvoir utiliser le procédé mis en œuvre par logiciel indéfiniment. Des autorisations liées à un numéro de série de disque particulier ou à une adresse MAC particulière ne sont pas non plus adaptées dans le « cloud ». En effet il n'est pas possible de connaitre le numéro de série du disque dur sous-jacent, ou l'adresse MAC de l'interface réseau mais surtout il n'y a pas la garantie que ces composants soient fixes.

[0031] L'invention se distingue de l'art antérieur et vise à permettre d'accorder une autorisation limitée dans le temps d'utilisation d'un procédé mis en œuvre par logiciel, aussi bien de manière locale que dans un « cloud », sans nécessiter d'implémentation du type d'une fourniture de service et sans permettre l'accès au code source.

[0032] Pour cela, il est proposé un procédé mis en œuvre par une application contenant un régénérateur d'un certificat client préalablement signé par une autorité de certification et un code offusqué correspondant au code logiciel protégé, le procédé comprenant, sur exécution de l'application :

- régénérer, à l'aide du régénérateur, le certificat client et émettre une requête comprenant le certificat client, et
- lorsque, suite à l'émission de la requête, une connexion est établie avec un serveur et une réponse indicative de validité du certificat client est reçue via la connexion établie, exécuter le code offusqué dans l'application, sinon, arrêter l'exécution de l'application sans exécuter le code offusqué.

[0033] La protection du code est assurée par une combinaison de son offuscation avec un mécanisme d'autorisation d'utilisation sécurisée dont l'exécution est compatible aussi bien avec une utilisation sur site que dans un « cloud ».

[0034] Un exemple particulier de réalisation est à présent décrit en référence à la [Fig.1] qui représente une architecture réseau mettant en communication des systèmes informatiques pour une mise en œuvre d'un code protégé.

[0035] Un premier système informatique (104) d'un client permet à un terminal utilisateur (102) du client d'accéder à une application (100) destinée à commander l'exécution d'un code protégé. Le code protégé est offusqué pour le rendre inaccessible au client.

[0036] Un deuxième système informatique (204) d'un fournisseur met en œuvre un serveur (200) de validation et d'autorisation. Les systèmes informatiques (104, 204) sont adaptés pour pouvoir mettre en œuvre, sous conditions, une communication entre l'application (100) et le serveur (200) de validation et d'autorisation avant que l'application (100) ne déclenche l'exécution du code protégé.

[0037] A titre d'exemple, dans la [Fig.1], les systèmes informatiques (104, 204) sont représentés sous la forme de clouds et le serveur (200) de validation et d'autorisation est considéré comme ayant été préalablement configuré depuis un terminal utilisateur (202) du fournisseur.

[0038] Afin de garantir une sécurisation des échanges entre l'application (100) et le serveur (200) de validation et d'autorisation, une possibilité est de mettre en œuvre une mini infrastructure à clé publique ou « mini-pki » reposant sur un mécanisme SSL.

[0039] Le mécanisme SSL se présente comme suit. A l'aide d'un outil de cryptographie, par exemple OpenSSL, il est possible de créer une autorité de certification et de générer un certificat de l'autorité de certification ainsi qu'un certificat serveur et des certificats clients signés par l'autorité de certification. Chaque certificat client ainsi généré peut être destiné à un client cible particulier.

[0040] Le certificat serveur est fourni au serveur (200) de validation et d'autorisation pour mettre en œuvre une authentification bidirectionnelle de type « SSL two-way ».

[0041] La mini infrastructure à clé publique comprend, au niveau de l'application (100), un générateur de code d'intégration. Un code d'intégration est un code technique de sécurisation. Dans l'exemple du mécanisme SSL évoqué, le code technique de sécurisation est configuré pour régénérer, lors de son exécution, un certificat client prévu pour un client cible. Ce certificat client est signé par l'autorité de certification et peut être reconnu par le serveur (200) de validation et d'autorisation qui dispose du certificat serveur.

[0042] Pour établir une connexion sécurisée avec le serveur d'autorisation, le mécanisme SSL a besoin d'écrire les certificats sur disque. Pour répondre à une problématique de ne pas laisser de trace de ces certificats après la connexion avec le serveur (200), il est possible de prévoir une écriture volatile sur disque des certificats juste au moment de la connexion, les fichiers temporaires générés disparaissant ensuite instantanément. En complément de cette précaution, le code technique de sécurisation peut être offusqué pour assurer que ni le client ni un tiers malveillant n'ait accès à son contenu.

[0043] Pour concevoir une version de l'application (100) pour un client cible, une possibilité est à présent détaillée. Un certificat client signé par l'autorité de certification est généré pour le client cible. Un module logiciel est ensuite créé de manière à incorporer le certificat client pour le client cible et le certificat de l'autorité de certification. Le module logiciel peut par exemple être rédigé en langage python. Le module logiciel et

le code à protéger peuvent être combinés et être offusqués ensemble.

[0044] L'ensemble ainsi offusqué prend la forme d'une application qui peut être installée par exemple sur une machine virtuelle. Une option particulière concerne l'installation de l'application dans un conteneur logiciel, c'est-à-dire un environnement d'exécution léger qui regroupe tous les composants système nécessaires à l'exécution de son contenu. En l'espèce, le contenu de l'application comporte au moins le module logiciel et le code protégé. Un tel conteneur peut être indifféremment distribué en tant que logiciel sur site ou comme un service hébergé sur abonnement.

[0045] Un exemple particulier de réalisation est à présent décrit en référence à la [Fig.2] qui représente un ordinogramme d'un programme informatique convenant pour la mise en œuvre d'un procédé d'exécution du code logiciel protégé par l'application (100) dans une version donnée pour un client cible.

[0046] L'exécution de l'application (100) peut être déclenchée de diverses manières, actives ou passives. Un exemple de déclenchement actif est une interaction homme-machine au niveau d'un terminal utilisateur (102) du client. Un exemple de déclenchement passif est le passage d'un repère temporel prédéfini dans un logiciel de planification de tâches.

[0047] A l'exécution, l'application (100) crée, ou régénère (300), par le biais du module logiciel, le certificat de l'autorité de certification et le certificat client pour le client cible signé par l'autorité de certification. Ces certificats sont régénérés de manière temporaire sous la forme de fichiers de certificats. Il existe de nombreuses méthodes connues pour entraîner l'effacement de fichiers de manière active ou passive après leur utilisation prévue.

[0048] L'application émet (302) une requête à destination du serveur (200) de validation et d'autorisation. La requête présente le certificat client pour le client cible.

[0049] L'application agit comme lanceur du code protégé offusqué et conditionne l'exécution (310) du code protégé offusqué à une double vérification :

- la première vérification (304) concerne l'établissement de la connexion sécurisée entre l'application (100) et le serveur (200), et
- la deuxième vérification (306) concerne l'acceptation par le serveur (200) du certificat client fourni par l'application (100) dans la requête.

[0050] A l'inverse, n'importe quelle erreur SSL ou mauvaise réponse du serveur (200) a pour effet de quitter l'application (100).

[0051] Dès lors que les certificats ne sont plus nécessaires, c'est-à-dire, dès l'émission (302) de la requête, ils ne sont plus nécessaires et sont automatiquement effacés (308). Cet effacement survient indépendamment du fait que la connexion sécurisée soit ensuite établie ou qu'une erreur SSL ou qu'une mauvaise réponse du serveur occasionne une sortie de l'application.

[0052] En procédant ainsi, la version de l'application (100) remise au client cible a la certitude de s'adresser au bon serveur. En fait, toute tentative de connexion à un autre serveur entraîne une erreur SSL car seule l'autorité de certification est considérée de confiance. De façon réciproque, le serveur (200) n'accepte que des certificats clients qui ont été signés par l'autorité de certification.

[0053] La validité du certificat client correspond à une limitation dans le temps de l'autorisation d'utilisation pouvant être accordée. Cette durée peut être modifiée, c'est-à-dire prolongée ou réduite, par des actions entreprises au niveau du serveur (200). En particulier, il est possible de mettre fin à une autorisation d'utilisation pour un client cible en révoquant le certificat client pour ce client cible.

[0054] L'infrastructure serveur peut se mettre en œuvre de différentes manières avec des logiciels du marché ou avec des logiciels spécifiques.

[0055] Un exemple particulier de réalisation est à présent décrit en référence à la [Fig.3] qui représente un ordinogramme d'un programme informatique convenant pour la mise en œuvre d'un procédé de validation et d'autorisation par le serveur (200) de validation et d'autorisation.

[0056] Le serveur (200) reçoit (400) une requête, émise par l'application (100) et présentant un certificat client. Le serveur obtient (402) ainsi le certificat client présenté et vérifie (404) si ce certificat a été signé par l'autorité de certification.

[0057] Lorsque le résultat de cette vérification est positif, c'est-à-dire lorsque le serveur reconnaît le certificat client présenté, une communication sécurisée est établie (408) avec l'application (100).

[0058] Le serveur, après avoir ensuite vérifié (410) la validité du certificat client présenté, émet (412) un signal indicatif d'une autorisation à exécuter le code protégé. L'exécution du code protégé peut de ce fait être déclenchée par l'application (100).

[0059] Si le certificat client présenté n'est pas valide, par exemple parce que sa date de validité a expiré, le client cible n'est pas autorisé à exécuter le code protégé. Le serveur émet (414) alors un signal indicatif d'une absence d'autorisation à exécuter le code protégé. L'exécution du code protégé par l'application (100) est de ce fait prohibée, ce qui, en combinaison avec le procédé décrit en référence à la [Fig.2], entraîne la fermeture de l'application (100).

[0060] Si le serveur ne reconnaît pas le certificat client présenté, la connexion n'est pas établie (406), ce qui, en combinaison avec le procédé décrit en référence à la [Fig.2], entraîne la fermeture de l'application (100).

Revendications

[Revendication 1] Procédé d'exécution d'un code logiciel protégé, le procédé étant mis en œuvre par une application contenant un régénérateur d'un certificat client préalablement signé par une autorité de certification et un code offusqué correspondant au code logiciel protégé, le procédé comprenant, sur exécution de l'application :
- régénérer, à l'aide du régénérateur de certificat, le certificat client et émettre une requête comprenant le certificat client, et
- lorsque, suite à l'émission de la requête, une connexion est établie avec un serveur et une réponse indicative de validité du certificat client est reçue via la connexion établie, exécuter le code offusqué dans l'application, sinon, arrêter l'exécution de l'application sans exécuter le code offusqué.

[Revendication 2] Procédé selon la revendication 1, comprenant en outre :
- effacer le certificat client régénéré immédiatement après l'émission de la requête.

[Revendication 3] Procédé selon la revendication 1 ou 2, dans lequel le régénérateur de certificat est offusqué.

[Revendication 4] Procédé selon l'une des revendications 1 à 3, dans lequel l'application est gérée par un conteneur logiciel dans un réseau informatique en nuage.

[Revendication 5] Procédé selon l'une des revendications 1 à 4, dans lequel la requête est une requête HTTPS.

[Revendication 6] Procédé selon l'une des revendications 1 à 5, dans lequel l'exécution de l'application est déclenchée par un logiciel de planification de tâches.

[Revendication 7] Programme informatique comportant des instructions pour la mise en œuvre du procédé selon l'une des revendications 1 à 6 lorsque ce programme est exécuté par un processeur.

[Revendication 8] Support d'enregistrement non transitoire lisible par un ordinateur sur lequel est enregistré un programme pour la mise en œuvre du procédé selon l'une des revendications 1 à 6 lorsque ce programme est exécuté par un processeur.

[Fig. 1]

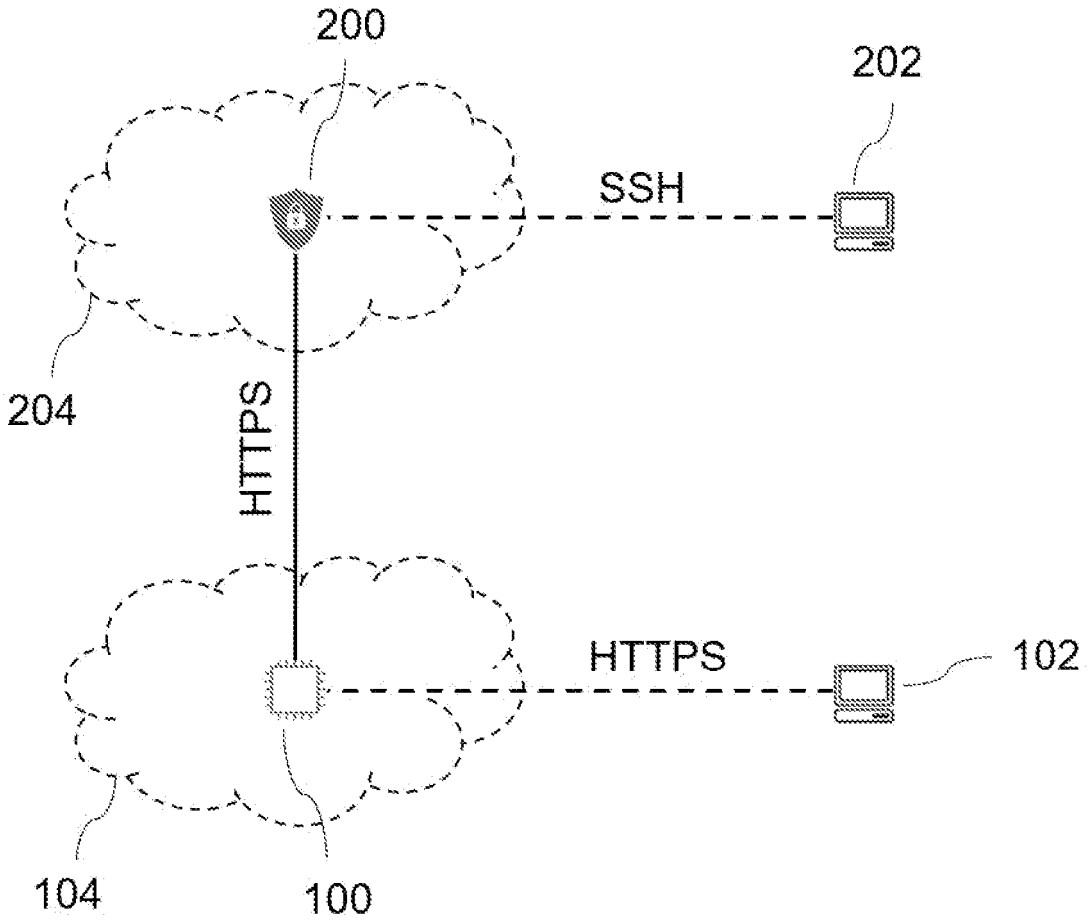


FIG. 1

[Fig. 2]

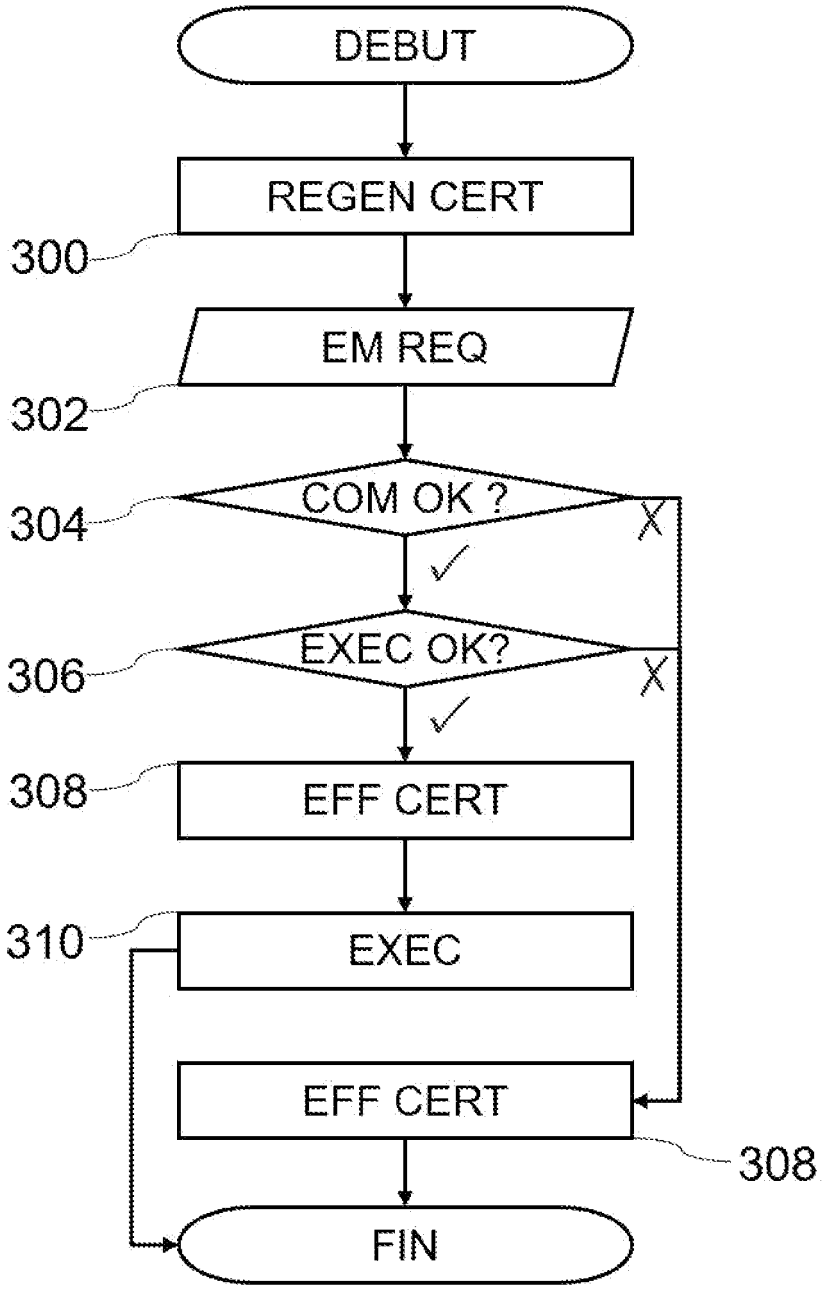

DEBUT
REGEN CERT
300
EM REQ
302
COM OK ?
304
X
EXEC OK?
306
X
EFF CERT
308
EXEC
310
EFF CERT
308
FIN

FIG. 2

[Fig. 3]

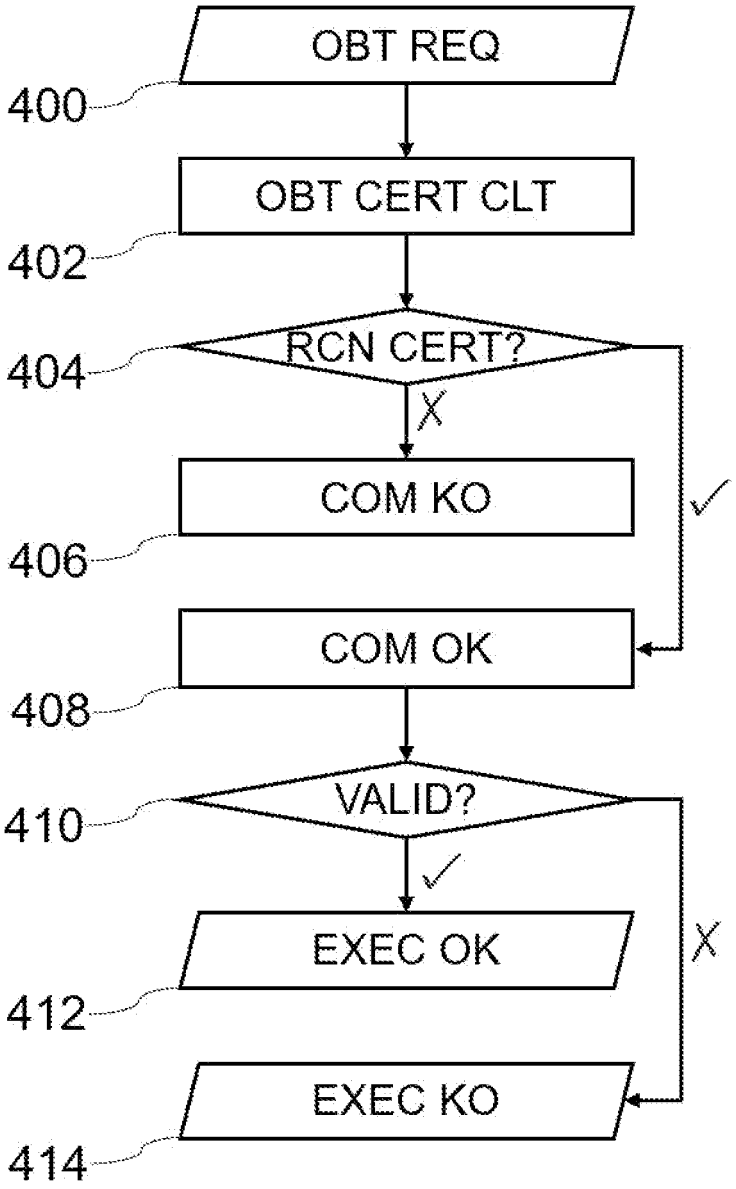

OBT REQ
400
OBT CERT CLT
402
RCN CERT?
404
X
COM KO
406
✓
COM OK
408
VALID?
410
✓
EXEC OK
412
X
EXEC KO
414

FIG. 3

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

N° d'enregistrement national

FA 916865
FR 2212896

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	WO 2020/098377 A1 (ALIBABA GROUP HOLDING LTD [CN]) 22 mai 2020 (2020-05-22) * alinéa [0002] - alinéa [0119] *	1-8	G06F 21/33 G06F 8/61 H04L 9/32
A	US 2005/044359 A1 (ERIKSSON THOMAS [SE] ET AL) 24 février 2005 (2005-02-24) * alinéa [0026] - alinéa [0058] *	1-8	
A	US 2011/145568 A1 (CLEMENT JEAN-YVES [FR] ET AL) 16 juin 2011 (2011-06-16) * alinéa [0023] - alinéa [0040]; figure 2 *	1-8	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
			G06F H04L

Date d'achèvement de la recherche: 29 septembre 2023

Examinateur: Jardak, Christine

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons
.....
& : membre de la même famille, document correspondant

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 2212896 FA 916865

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du **29-09-2023**
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 2020098377 A1	22-05-2020	CN 110011801 A	12-07-2019
		CN 112468473 A	09-03-2021
		TW 202021306 A	01-06-2020
		WO 2020098377 A1	22-05-2020
US 2005044359 A1	24-02-2005	US 2005044359 A1	24-02-2005
		US 2010212028 A1	19-08-2010
		WO 2004099952 A2	18-11-2004
US 2011145568 A1	16-06-2011	AUCUN	

EPO FORM P0465