(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** **3 068 151**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **17 55892**

(51) Int Cl[8] : ***G 06 F 21/12*** *(2013.01), G 06 F 21/52, 21/53, 21/64*

# (12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 27.06.17.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 28.12.18 Bulletin 18/52.

(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) :** *SAFRAN IDENTITY & SECURITY* *Société par actions simplifiée* — FR.

(72) **Inventeur(s) :** BRUGNON MARC, BENCHETTRIT MICHEL, BAILLY ALEXIS et BENHAMMAM ABDELGHANI.

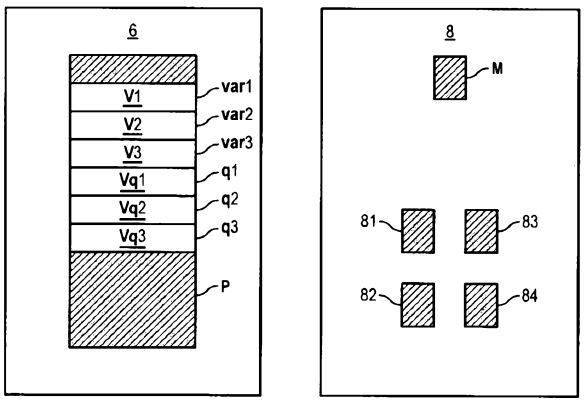
(73) **Titulaire(s) :** SAFRAN IDENTITY & SECURITY Société par actions simplifiée.

(74) **Mandataire(s) :** REGIMBEAU.

## (54) PROCEDE DE PROTECTION D'UN DISPOSITIF ELECTRONIQUE EXECUTANT UN PROGRAMME CONTRE DES ATTAQUES PAR INJECTION DE FAUTE.

(57) L'invention concerne un procédé de protection d'un dispositif électronique, exécutant une application par l'intermédiaire d'une machine virtuelle, contre des attaques par injection de faute, l'exécution de l'application mettant en œuvre une pile dans laquelle des variables utilisées par le programme sont lues et/ou mémorisées, le dispositif électronique comprenant une mémoire non volatile dans laquelle est stockée une pluralité d'applications, le procédé comprenant les étapes suivantes :

- détermination d'au moins une application dite générique parmi les applications stockées ;
- application d'un mode d'exécution sécurisé sur ladite application générique, comprenant une sécurisation des variables lue(s)/mémorisée(s) dans la pile.




FR 3 068 151 - A1

## DOMAINE TECHNIQUE GENERAL

L'invention concerne les logiciels et environnements d'exécution embarqués dans un dispositif électronique et concerne, en particulier, un procédé pour protéger l'exécution d'une application compilée en langage intermédiaire contre des attaques par injection de faute, lors de son exécution sur un dispositif électronique muni d'une machine virtuelle.

## ETAT DE LA TECHNIQUE

L'exécution d'un code Java (et les données manipulées par ce dernier) est sujette à des attaques par injection de fautes.

De façon connue, une attaque par injection de faute consiste à perturber l'environnement physique d'un dispositif électronique qui exécute un programme, de sorte à modifier la valeur mémorisée par le dispositif d'une variable destinée à être utilisée par le programme. De telles perturbations peuvent être produites de différentes manières : variation d'une tension d'alimentation, variation d'une fréquence d'horloge du dispositif, émission de rayonnement électromagnétique ou laser, etc.

Afin de protéger les données notamment dites sensibles, il convient de les protéger en intégrité au niveau de l'application et non pas systématiquement au moment de l'exécution ce qui est consommateur de mémoire et non souhaitable compte tenu de la mémoire souvent limitée.

Aussi, de façon classique, pour une machine virtuelle donnée, les développeurs d'applications doivent suivre un certain nombre de recommandations relatives à la protection des données au niveau de l'application (Guide de développement, règles sécuritaires).

Toutefois, sur un dispositif électronique des applications dites « génériques » peuvent être chargées sur le dispositif électronique. Ces applications sont génériques en ce que les développeurs n'ont pas nécessairement suivi les recommandations précitées.

## PRESENTATION DE L'INVENTION

Un but de l'invention est de protéger l'exécution d'applications contre des attaque par injection de faute.

A cet effet, l'invention propose un procédé de protection d'un dispositif électronique, exécutant une application par l'intermédiaire d'une machine virtuelle, contre des attaques par injection de faute, l'exécution de l'application mettant en œuvre une pile dans laquelle des variables utilisées par le programme sont lues et/ou mémorisées, le dispositif électronique comprenant une mémoire non volatile dans laquelle est stockée une pluralité d'applications, le procédé comprenant les étapes suivantes :

- détermination d'au moins une application dite générique parmi les applications stockées ;
- application d'un mode d'exécution sécurisé sur ladite application générique, comprenant une sécurisation des variables lue(s)/mémorisée(s) dans la pile.

L'invention est avantageusement complétée par les caractéristiques suivantes, prises seules ou en une quelconque de leur combinaison techniquement possible :

- la sécurisation des variables comprend, lorsqu'une variable doit être mémorisée dans la pile, les étapes suivantes : calcul d'une fonction d'intégrité de la variable à mémoriser ; mémorisation de ladite variable et de la fonction d'intégrité associée ;
- la sécurisation des variables comprend, lorsqu'une variable doit être lue dans la pile, les étapes suivantes : lecture d'une variable et de la fonction d'intégrité correspondante ; calcul d'une fonction d'intégrité de la variable lue selon la fonction d'intégrité utilisée à la mémorisation de la variable lue ; comparaison de la fonction d'intégrité lue à la fonction d'intégrité ainsi calculée ; signalement ou non d'une erreur en fonction du résultat de la comparaison ;
- le procédé comprend un stockage dans la mémoire non volatile d'une pluralité d'applications ;
- la fonction d'intégrité est : la fonction identité, une fonction de hachage, une fonction générant un code de redondance longitudinal, une fonction générant un code de redondance cyclique ;
- le dispositif électronique est une carte à puce comprenant une mémoire non volatile dans laquelle est stockée une machine virtuelle.

L'invention concerne également une machine virtuelle d'exécution d'une application compilée en code intermédiaire sur un dispositif électronique, ladite

machine virtuelle étant stockée dans une mémoire non volatile du dispositif électronique et est configurée pour mettre en œuvre un procédé selon l'invention.

Et l'invention concerne également un dispositif électronique, tel qu'une carte à puce, comprenant une mémoire non volatile dans laquelle est stockée la machine virtuelle selon l'invention.

L'invention permet d'avoir un comportement dynamique de la machine virtuelle en fonction du type d'applications. En outre, l'invention évite d'avoir à sécuriser le contenu complet de la mémoire volatile ce qui limite la consommation de mémoire supplémentaire nécessaire à la sécurisation. L'invention a donc un impact léger sur les performances du dispositif électronique.

PRESENTATION DES FIGURES

D'autres caractéristiques, buts et avantages de l'invention ressortiront de la description qui suit, qui est purement illustrative et non limitative, et qui doit être lue en regard des dessins annexés sur lesquels :

- la figure 1 illustre schématiquement un dispositif électronique selon un mode de réalisation de l'invention ;
- la figure 2 illustre schématiquement une mémoire volatile et une mémoire non volatile d'un dispositif électronique selon l'invention ;
- les figures 3 à 5 illustrent des organigrammes d'étapes d'un procédé selon un mode de réalisation de l'invention.

DESCRIPTION DETAILLEE DE L'INVENTION

En référence à la **figure 1**, un dispositif électronique 1 comprend au moins un processeur 2 et au moins une mémoire 4.

La mémoire 4 comprend au moins une mémoire volatile 6, par exemple de type RAM. La mémoire volatile 6 a pour fonction de mémoriser temporairement des données, par exemple des données calculées par le processeur 2. Le contenu de la mémoire volatile 6 est effacé lors d'une mise hors tension du dispositif électronique 1.

La mémoire 4 comprend en outre au moins une mémoire 8 non volatile, par exemple de type disque dur, SSD, flash, EEPROM, etc. La mémoire 8 non volatile a pour fonction de mémoriser des données de manière persistante, au sens où une

mise hors tension du dispositif électronique 1 n'efface pas le contenu de la mémoire 8 non volatile.

Le processeur 2 est adapté pour exécuter des instructions de code de programmes appartenant à un jeu d'instructions prédéterminé.

Par extension, le processeur 2 est adapté pour exécuter des programmes se présentant sous la forme d'un binaire compilé comprenant des instructions de codes appartenant à ce jeu d'instructions prédéterminé. On prendra l'exemple dans la suite des programmes de type machine virtuelle.

La machine virtuelle est configurée pour interpréter des programmes ou applications, les programmes se présentant sous la forme d'un binaire comprenant des instructions de code dans un format différent du jeu d'instruction précité, lorsque la machine virtuelle est exécutée par le processeur 2.

Par exemple, une machine virtuelle JavaCard est configurée pour interpréter un « *bytecode* » issu d'un code source dans le langage de programmation JavaCard, qui est un langage de programmation orienté objet.

En relation avec la **figure 2**, la machine virtuelle M est mémorisée dans la mémoire non volatile 8, de même que des applications 81, 82, 83, 84 destinées à être exécutées par la machine virtuelle M.

Le dispositif électronique 1 est par exemple une carte à puce, telle qu'une carte SIM.

En relation avec les **figures 3** à **5**, un procédé de protection du dispositif électrique 1 contre des attaques par injection de faute comprend les étapes suivantes.

Dans une étape préliminaire 100, des applications 81, 82, 83, 84 sont stockées à divers emplacements de la mémoire 8 non volatile. La machine virtuelle M a connaissance des adresses de stockage des différentes applications.

Le processeur 2 démarre 200 l'exécution de la machine virtuelle M, par exemple lors d'une mise sous tension du dispositif électronique 1.

Dans une phase d'initialisation 300, la machine virtuelle M détermine parmi les applications 81, 82, 83, 84 stockées dans la mémoire non volatile 8, lesquelles sont dites génériques c'est-à-dire en ce qu'elles n'ont pas été sécurisées directement au niveau de l'application elle-même selon des recommandations spécifiques. Cette discrimination (applet sécurisée ou non) est indiquée au niveau du package (entête). Le package est le container dans lequel se trouve l'applet. De manière plus générale

ici il s'agit de sélectionner un certain nombre d'applications parmi celles stockées dans la mémoire non volatile.

Puis, la machine virtuelle exécute 400 selon un mode d'exécution dit sécurisé au moins une de ces applications génériques. A noter que pour les applications qui ne sont pas génériques on a un mode d'exécution dit nominal. En fonction du type d'application à exécuter la machine virtuelle mettra en œuvre soit le mode d'exécution nominal soit le mode d'exécution sécurisé.

Quel que ce soit le mode d'exécution, la machine virtuelle M alloue 500 dans la mémoire volatile 6, une pile d'exécution P associé à une application.

Cette pile d'exécution P a pour vocation à contenir des variables dont le processus de l'application a besoin au cours de son exécution.

La pile d'exécution P est définie par une adresse de pile dans la mémoire volatile 6, une taille, et une taille de mot adressable dans la pile P.

L'adresse de pile AP est une adresse de début de la pile P, ou bien une adresse de fin de la pile P.

Au cours de l'exécution 400 du programme par la machine virtuelle M, cette dernière commande la mémorisation 410 ou la lecture 420 de plusieurs variables var1, var2, var3 dans la mémoire volatile 6. Chaque variable a une valeur V1, V2, V3. Sur la figure 3 trois variables sont représentées à titre d'exemple.

Le mode d'exécution sécurisé se distingue du mode nominal comme suit.

A la mémorisation 410, une fonction d'intégrité de la variable à mémoriser est calculée 411 ayant pour résultat une donnée d'intégrité qui est mémorisée 412 avec la variable dans la mémoire volatile.

La donnée d'intégrité peut être mémorisée directement à la suite de la variable ou bien à une adresse différente (comme cela est illustré sur la figure 3 où q1, q2, q3 sont les données d'intégrité ayant pour valeur Vq1, Vq2, Vq3 des variables var1, var2, var3).

A la lecture 420, outre la variable lue 421, sa donnée d'intégrité est également lue 422. Puis, on calcule 423 alors une fonction d'intégrité de la variable lue ayant pour résultat une donnée d'intégrité de contrôle. Ensuite, on compare 424 la donnée d'intégrité lue à la donnée d'intégrité de contrôle qui vient d'être calculée. Bien entendu, la fonction d'intégrité est identique à celle mise en œuvre à la mémorisation de la variable considérée.

La fonction d'intégrité peut être : la fonction identité, une fonction de hachage, une fonction générant un code de redondance longitudinal (« *longitudinal redundancy check* » ou LRC en anglais), une fonction générant un code de redondance cyclique (« *cyclic redundancy check* » ou CRC en anglais), etc.

En fonction du résultat de la comparaison, une erreur est éventuellement signalée 500.

En particulier, lorsque la donnée d'intégrité lue et la donnée d'intégrité de contrôle sont différentes, une erreur est signalée (étape 510) par la machine virtuelle M. Dans ce cas, on peut considérer qu'une attaque a été effectuée à l'encontre de la mémoire volatile.

A contrario, lorsque la donnée d'intégrité lue et la donnée d'intégrité de contrôle sont identiques on ne signale pas (étape 520) d'erreur.

Lorsque l'étape de signalement d'erreur est mise en œuvre, diverses mesures peuvent être prises : par exemple un arrêt de l'exécution de l'application, ou bien une mesure plus radicale, par exemple un arrêt du dispositif électronique. De manière alternative, on peut décider de l'arrêt de l'application ou bien du dispositif électronique 1 après un certain nombre de signalement.

Lorsqu'il n'est pas pris de décision d'arrêter l'exécution de l'application, la machine virtuelle M poursuit l'exécution de l'application.

REVENDICATIONS

1. Procédé de protection d'un dispositif électronique (1), exécutant une application par l'intermédiaire d'une machine (M) virtuelle, contre des attaques par injection de faute, l'exécution de l'application mettant en œuvre une pile (P) dans laquelle des variables (var1, var2, var3) utilisées par le programme sont lues et/ou mémorisées, le dispositif électronique comprenant une mémoire non volatile dans laquelle est stockée une pluralité d'applications, le procédé comprenant les étapes suivantes

- détermination (300) d'au moins une application dite générique parmi les applications stockées ;
- application (400) d'un mode d'exécution sécurisé sur ladite application générique, comprenant une sécurisation des variables lue(s)/mémorisée(s) dans la pile (P).

2. Procédé selon la revendication 1, dans lequel la sécurisation des variables comprend, lorsqu'une variable doit être mémorisée dans la pile, les étapes suivantes :

- calcul d'une fonction d'intégrité de la variable à mémoriser ;
- mémorisation de ladite variable et de la fonction d'intégrité associée.

3. Procédé selon la revendication 2, dans lequel la sécurisation des variables comprend, lorsqu'une variable doit être lue dans la pile, les étapes suivantes :

- lecture d'une variable et de la fonction d'intégrité correspondante ;
- calcul d'une fonction d'intégrité de la variable lue selon la fonction d'intégrité utilisée à la mémorisation de la variable lue ;
- comparaison de la fonction d'intégrité lue à la fonction d'intégrité ainsi calculée ;
- signalement ou non d'une erreur en fonction du résultat de la comparaison.

4. Procédé selon l'une quelconque des revendications précédentes, comprenant un stockage dans la mémoire non volatile d'une pluralité d'applications.

5. Procédé selon l'une quelconque des revendications 2 à 4, dans lequel la fonction d'intégrité est : la fonction identité, une fonction de hachage, une fonction générant

un code de redondance longitudinal, une fonction générant un code de redondance cyclique.

6. Procédé selon l'une quelconque des revendications précédentes, dans lequel le dispositif électronique est une carte à puce comprenant une mémoire (8) non volatile dans laquelle est stockée une machine virtuelle.

7. Machine virtuelle d'exécution d'une application compilée en code intermédiaire sur un dispositif électronique, ladite machine virtuelle étant stockée dans une mémoire non volatile (8) du dispositif électronique et est configurée pour mettre en œuvre un procédé selon l'une des revendications précédentes.

8. Dispositif électronique, tel qu'une carte à puce, comprenant une mémoire (8) non volatile dans laquelle est stockée la machine virtuelle selon la revendication précédente.

FIG. 1

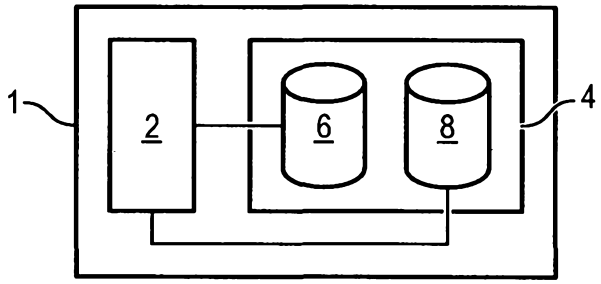

1
2
6
8
4

FIG. 2

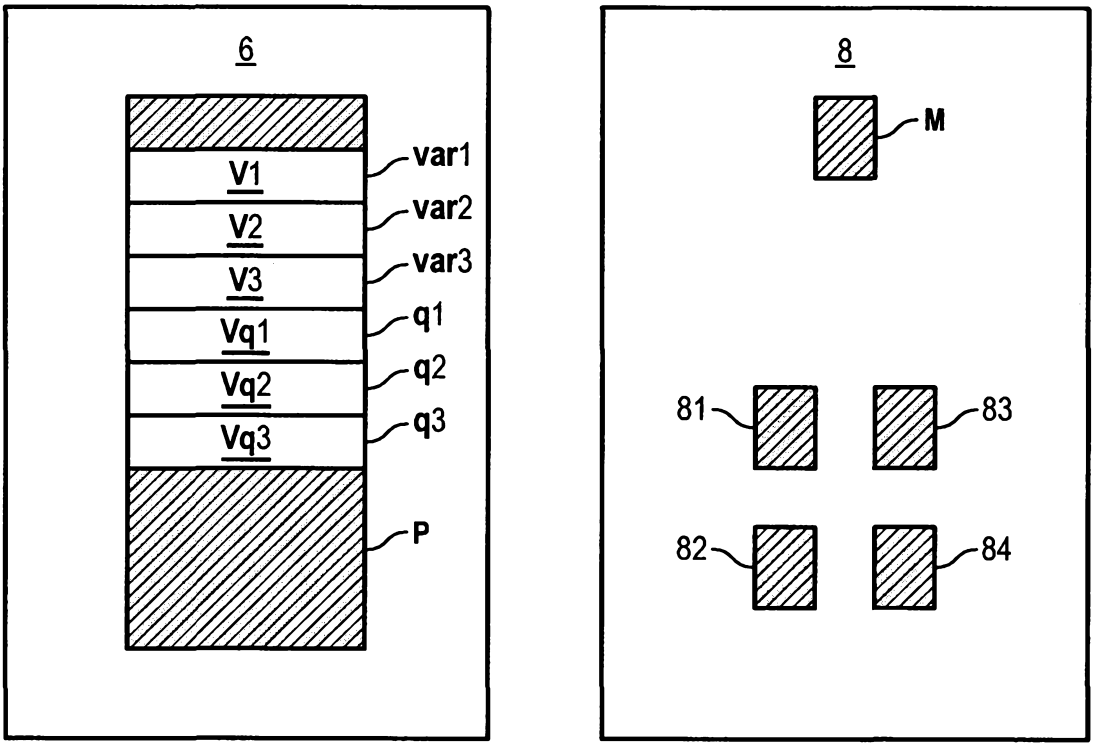

6
V1
V2
V3
Vq1
Vq2
Vq3
var1
var2
var3
q1
q2
q3
P
8
M
81
83
82
84

FIG. 3

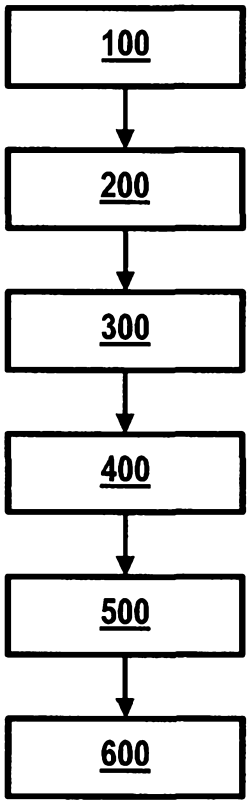



FIG. 4

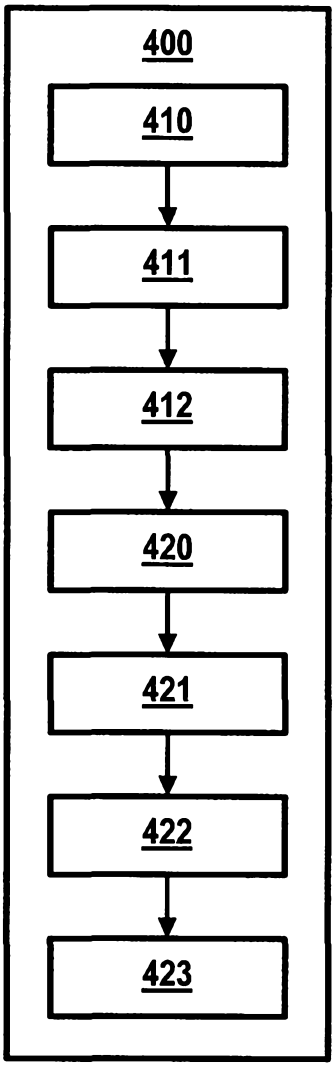
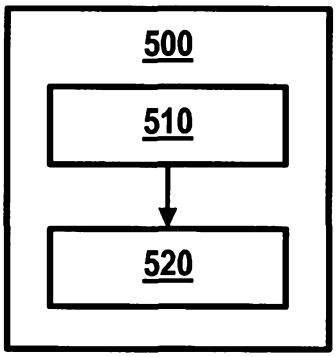



FIG. 5



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 843632
FR 1755892

| 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 | EP 1 881 404 A1 (GEMPLUS CARD INT [FR])<br>23 janvier 2008 (2008-01-23)<br>* abrégé *<br>* alinéa [0001] *<br>* alinéa [0013] - alinéa [0015] *<br>* alinéa [0017] *<br>* alinéa [0022] *<br>* alinéa [0035] *<br>* alinéa [0038] *<br>* alinéa [0042] *<br>* alinéa [0065] - alinéa [0070] *<br>----- | 1-8 | G06F21/12<br>G06F21/64<br>G06F21/52<br>G06F21/53 |
| X | WO 2007/068706 A1 (GEMPLUS CARD INT [FR]; GIRARD PIERRE [FR]; GONZALVO BENOIT [FR])<br>21 juin 2007 (2007-06-21)<br>* abrégé *<br>* page 1, ligne 1 - ligne 7 *<br>* page 6, ligne 6 - ligne 30 *<br>* page 10, ligne 28 - ligne 31 *<br>----- | 1-8 | |
| | | | DOMAINES TECHNIQUES RECHERCHÉS (IPC) |
| | | | G06F |

| Date d'achèvement de la recherche | Examinateur |
|---|---|
| 17 avril 2018 | Oliveira, Joel |

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 1755892 FA 843632

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 17-04-2018
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 |
|---|---|---|---|
| EP 1881404 A1 | 23-01-2008 | EP 1881404 A1 | 23-01-2008 |
| | | EP 2047366 A1 | 15-04-2009 |
| | | US 2009328231 A1 | 31-12-2009 |
| | | WO 2008009697 A1 | 24-01-2008 |
| WO 2007068706 A1 | 21-06-2007 | AT 550725 T | 15-04-2012 |
| | | CN 101366035 A | 11-02-2009 |
| | | EP 1960934 A1 | 27-08-2008 |
| | | US 2009165149 A1 | 25-06-2009 |
| | | WO 2007068706 A1 | 21-06-2007 |

EPO FORM P0465

Pour tout renseignement concernant cette annexe : voir Journal Officiel de l'Office européen des brevets, No.12/82