(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** **3 068 152**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **17 55893**

(51) Int Cl[8] : ***G 06 F 21/12*** *(2017.01), G 06 F 21/52, G 06 F 21/53, G 06 F 21/64*

(12) **BREVET D'INVENTION** **B1**

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

(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.

(45) **Date de la mise à disposition du public du brevet d'invention :** 02.08.19 Bulletin 19/31.

(56) **Liste des documents cités dans le rapport de recherche :**

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, BENHAMMAM ABDELGHANI et BAILLY ALEXIS.

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

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

FR 3 068 152 - B1

DOMAINE TECHNIQUE GENERAL

L'invention concerne un procédé de protection d'un dispositif électronique exécutant un programme contre des attaques par injection de faute de type mises en œuvre sur une variable destinée à être utilisée par le programme.

ETAT DE LA TECHNIQUE

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.

Pour protéger une variable contre une attaque par injection de faute, plusieurs types de contremesures existent.

Une première contremesure pour protéger une variable contre une attaque par injection de faute consiste à simplement dupliquer le contenu de la pile d'exécution dans une pile de sauvegarde. Pour vérifier l'intégrité d'une variable, on compare la valeur de la variable mémorisée dans la pile d'exécution et la valeur de sa copie dans la pile de sauvegarde. Si les deux valeurs comparées sont différentes, une faute est détectée.

Une deuxième contremesure, décrite dans le document EP 1 960 934 B1, consiste à calculer une donnée de contrôle d'intégrité Q sur la variable et mémoriser cette donnée de contrôle d'intégrité dans une pile de contrôle dédiée. Pour appliquer ce principe de protection, deux piles indépendantes sont allouées dans la mémoire du dispositif, comme cela est représenté sur la **figure 1** :

- une pile d'exécution P1 contenant la valeur de chaque variable V destinée à être utilisée par le programme, et
- une pile de contrôle P2 contenant les données de contrôle d'intégrité Q sur chaque variable.

Sont également prévus deux registres d'index :

- un premier registre d'index ind1 contenant des données d'adressage A1 adaptées pour localiser les variables contenues dans la pile d'exécution P1, et

- un deuxième registre d'index ind2 contenant des données d'adressage A2 adaptées pour localiser les données de contrôle d'intégrité contenues dans la pile de contrôle P2.

Pour vérifier l'intégrité d'une variable, on lit la valeur courante de cette variable présente dans la pile d'exécution P1 (cette valeur étant localisée grâce à une donnée d'adressage associée à la variable, présente dans le premier registre d'index ind1), on met en œuvre un calcul d'intégrité sur la base de la valeur lue et on compare le résultat de ce calcul à la donnée d'intégrité associée à la variable qui est présente dans la pile de contrôle (la donnée d'intégrité étant localisée grâce à une donnée d'adressage correspondante dans le deuxième registre d'index ind2). Si les deux valeurs comparées sont différentes, une faute est détectée.

On constate donc que pour protéger l'intégrité de données deux piles sont mises œuvre. Une telle méthode est consommatrice de mémoire ce qui n'est pas souhaitable pour une carte à puce dont la mémoire est limitée.

PRESENTATION DE L'INVENTION

Un but de l'invention est de protéger une variable mémorisée dans une pile d'exécution contre des attaques par injection de faute moyennant une consommation de mémoire supplémentaire réduite.

A ce titre, l'invention concerne, un procédé de protection d'un dispositif électronique exécutant un programme contre des attaques par injection de faute de type susceptibles d'affecter le programme, l'exécution du programme mettant en œuvre une pile dans laquelle des variables utilisées par le programme sont lues et/ou mémorisées, le procédé comprenant, au cours de l'exécution dudit programme, une étape de calcul d'une donnée de contrôle d'intégrité par application d'au moins une fonction prédéterminée à au moins une variable d'intégrité, la variable d'intégrité et la donnée de contrôle d'intégrité ayant une valeur mémorisée dans la pile, le calcul de la donnée d'intégrité étant mis en œuvre à intervalle aléatoire au cours de l'exécution du programme.

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

- le calcul de la donnée de contrôle d'intégrité comprend des étapes de mémorisation dans la pile d'au moins une valeur d'une variable d'intégrité ;

lecture dans la pile de ladite valeur mémorisée ; application de la fonction prédéterminée à l'au moins valeur mémorisée dont le résultat est la donnée de contrôle d'intégrité ; mémorisation de la donnée de contrôle d'intégrité ainsi obtenue ; lecture dans la pile de la donnée de contrôle d'intégrité obtenue ;

- le procédé comprend en outre, la comparaison de la donnée d'intégrité obtenue à un résultat attendu et un signalement ou non d'une erreur en fonction du résultat de la comparaison ;
- la fonction prédéterminée est une opération d'addition d'une pluralité de variables d'intégrité ; une opération de soustraction d'une pluralité de variables d'intégrité ; une opération de multiplication d'une pluralité de variables d'intégrité ; une opération logique du type « OR », « AND », « XOR », « NOR ».

L'invention concerne également un produit programme d'ordinateur comprenant des instructions de code de programme pour l'exécution des étapes du procédé selon l'une des revendications précédentes, lorsque ce procédé est exécuté par au moins un processeur.

L'invention concerne enfin un dispositif électronique, tel qu'une carte à puce, comprenant : au moins un processeur configuré pour exécuter un programme, au moins une mémoire adaptée pour contenir une pile d'exécution, le dispositif étant caractérisé en ce que le processeur est configuré pour mettre en œuvre un procédé selon l'invention.

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, outre la figure 1 déjà discutée :

- la figure 2 illustre schématiquement un dispositif électronique selon un mode de réalisation de l'invention ;

- la figure 3 illustre schématiquement le contenu des mémoires au cours de la mise en œuvre d'un procédé selon l'invention ;

- la figure 4 illustre schématiquement des étapes d'un procédé selon l'invention.

Sur l'ensemble des figures les éléments similaires portent des références identiques.

DESCRIPTION DETAILLEE DE L'INVENTION

En référence à la **figure 2**, 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 non volatile 8, par exemple de type disque dur, SSD, flash, EEPROM, etc. La mémoire non volatile 8 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 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, 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.

On suppose dans la suite que la machine virtuelle est mémorisée dans la mémoire non volatile 8, de même qu'un programme interprétable par la machine virtuelle ainsi qu'un programme de contrôle.

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

En relation avec les **figures 3** et **4**, on décrit ci-après un procédé de protection d'un dispositif électronique 1 exécutant un programme contre des attaques par injection de faute de type susceptibles d'affecter le programme.

Le processeur 2 démarre 100 l'exécution de la machine virtuelle.

La machine virtuelle alloue 200 dans la mémoire volatile 6 une pile d'exécution P associée à un programme. Cette pile d'exécution P a pour vocation à contenir des variables dont le processus du programme a besoin au cours de son exécution.

On considère que le programme comprend des instructions de code pour mettre en œuvre des étapes ci-dessous.

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

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

Au cours de l'exécution 300 du programme par la machine virtuelle, cette dernière commande la mémorisation ou la lecture 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 programme est quant à lui stockée dans la mémoire non volatile 8 sous forme d'instructions de code Inst_1, Inst_2, …, Inst_K. Ces instructions de code commandent au besoin la lecture/mémorisation des variables dans la mémoire volatile 6.

Au cours de l'exécution de ce programme, en outre, un calcul 400 d'une donnée de contrôle d'intégrité Q par application d'au moins une fonction prédéterminée à au moins une variable d'intégrité q1, q2 est mis en œuvre. Ce calcul est stockée sous la forme d'une instructions de code Inst_f(q1, q2)=Q et utilise les variables q1, q2 dont le résultat est la donne de contrôle d'intégrité Q.

Chaque variable d'intégrité q1, q2 et la donnée de contrôle d'intégrité Q ont une valeur mémorisée dans la pile P, respectivement Vq1, Vq2, VQ.

Un tel calcul est exécuté de manière aléatoire au cours de l'exécution du programme dont les instructions sont stockées dans la mémoire non volatile 8.

En particulier, la fréquence d'exécution du calcul est déterminée par un intervalle dans lequel une valeur aléatoire R est prise afin de de lancer l'exécution du calcul au bout de R instruction(s). Cette valeur R est rafraichie de manière aléatoire dans cet intervalle à la fin de chaque calcul.

De manière préférée, le calcul 400 de la donnée de contrôle d'intégrité Q comprend des étapes de

- mémorisation 401 dans la pile (P) d'au moins une valeur Vq1, Vq2 d'une variable d'intégrité q1, q2 ;
- lecture 402 dans la pile P de ladite valeur mémorisée ;
- application 403 de la fonction prédéterminée à l'au moins valeur mémorisée dont le résultat est la donnée de contrôle d'intégrité Q de valeur VQ ;
- mémorisation 404, dans la pile P de la donnée de contrôle d'intégrité Q ainsi obtenue ;
- lecture 405 dans la pile de la donnée de contrôle d'intégrité Q obtenue.

Ces étapes sont en soit classique dans la l'utilisation de la pile au cours du programme.

Toutefois, dans une étape ultérieure, on compare 500 la donnée de contrôle d'intégrité Q obtenue à un résultat attendu RES stockée dans la mémoire non volatile 8.

En effet, le calcul de la donnée de contrôle d'intégrité doit retourner un résultat déterminé à l'avance, la fonction prédéterminée, et les variables d'intégrité étant fixées à l'avance.

De fait, cette étape de comparaison 500 permet de signaler 600 ou pas une erreur en fonction du résultat de la comparaison.

En particulier une erreur est signalée si la valeur VQ de la donnée de contrôle d'intégrité Q est différente du résultat attendu.

Une erreur est significative d'une attaque sur la pile P. En effet une attaque sur la pile peut entrainer une modification des données d'entrée et/ou du résultat attendu, etc. qui conduira systématiquement à un résultat erroné suite à l'application de la fonction prédéterminée.

Une fonction prédéterminée peut consister à l'une des opérations suivantes :

- une opération d'addition d'une pluralité de variables d'intégrité ;
- une opération de soustraction d'une pluralité de variables d'intégrité ;
- une opération de multiplication d'une pluralité de variables d'intégrité ;
- une opération logique du type « OR », « AND », « XOR », « NOR ».

D'autres fonctions prédéterminées utilisant l'environnement de la machine virtuelle peuvent aussi être mises en œuvre :

- opération de lecture objet en mémoire volatile ;
- opération d'écriture objet en mémoire volatile ;
- opération de lecture objet en mémoire non volatile ;
- opération d'écriture objet en mémoire non volatile.

REVENDICATIONS

1. Procédé de protection d'un dispositif électronique (1) exécutant un programme contre des attaques par injection de faute de type susceptibles d'affecter le programme, l'exécution du programme mettant en œuvre une pile (P) dans laquelle des variables (var1, var2, var3) utilisées par le programme sont lues et/ou mémorisées, le procédé comprenant, au cours de l'exécution dudit programme, une étape de calcul (400) d'une donnée (Q) de contrôle d'intégrité par application d'au moins une fonction prédéterminée à au moins une variable d'intégrité (q1, q2), la variable d'intégrité et la donnée de contrôle d'intégrité ayant une valeur mémorisée dans la pile (P), le calcul de la donnée d'intégrité étant mis en œuvre à intervalle aléatoire au cours de l'exécution du programme.

2. Procédé selon la revendication 1, dans lequel le calcul (400) de la donnée de contrôle d'intégrité comprend des étapes de

- mémorisation (401) dans la pile (P) d'au moins une valeur (V) d'une variable d'intégrité ;
- lecture (402) dans la pile (P) de ladite valeur mémorisée ;
- application (403) de la fonction prédéterminée à l'au moins valeur mémorisée dont le résultat est la donnée de contrôle d'intégrité;
- mémorisation (404) de la donnée de contrôle d'intégrité ainsi obtenue ;
- lecture (405) dans la pile de la donnée de contrôle d'intégrité obtenue.

3. Procédé selon l'une des revendications précédentes, comprenant en outre, la comparaison de la donnée d'intégrité obtenue à un résultat attendu (RES) et un signalement ou non d'une erreur en fonction du résultat de la comparaison.

4. Procédé selon l'une des revendications précédentes, dans lequel ladite fonction prédéterminée est

- une opération d'addition d'une pluralité de variables d'intégrité ;
- une opération de soustraction d'une pluralité de variables d'intégrité ;
- une opération de multiplication d'une pluralité de variables d'intégrité ;
- une opération logique du type « OR », « AND », « XOR », « NOR ».

5. Produit programme d'ordinateur comprenant des instructions de code de programme pour l'exécution des étapes du procédé selon l'une des revendications précédentes, lorsque ce procédé est exécuté par au moins un processeur (2).

6. Dispositif électronique, tel qu'une carte à puce, comprenant :

- au moins un processeur (2) configuré pour exécuter un programme,
- au moins une mémoire (4) adaptée pour contenir une pile d'exécution (P),

le dispositif étant caractérisé en ce que le processeur (2) est configuré pour mettre en œuvre un procédé selon l'une des revendications 1 à 4.

FIG. 1

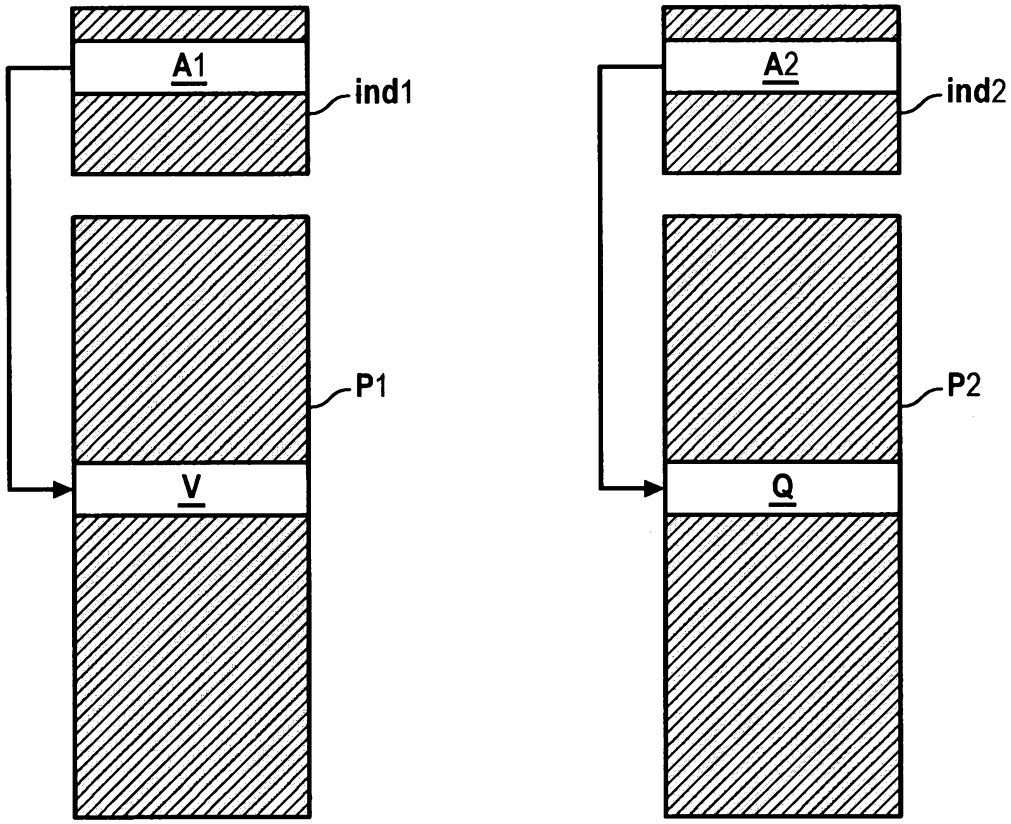


FIG. 2

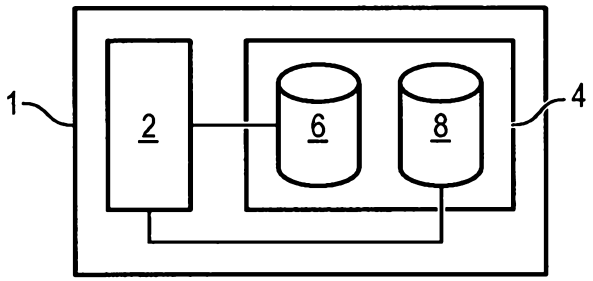


FIG. 3

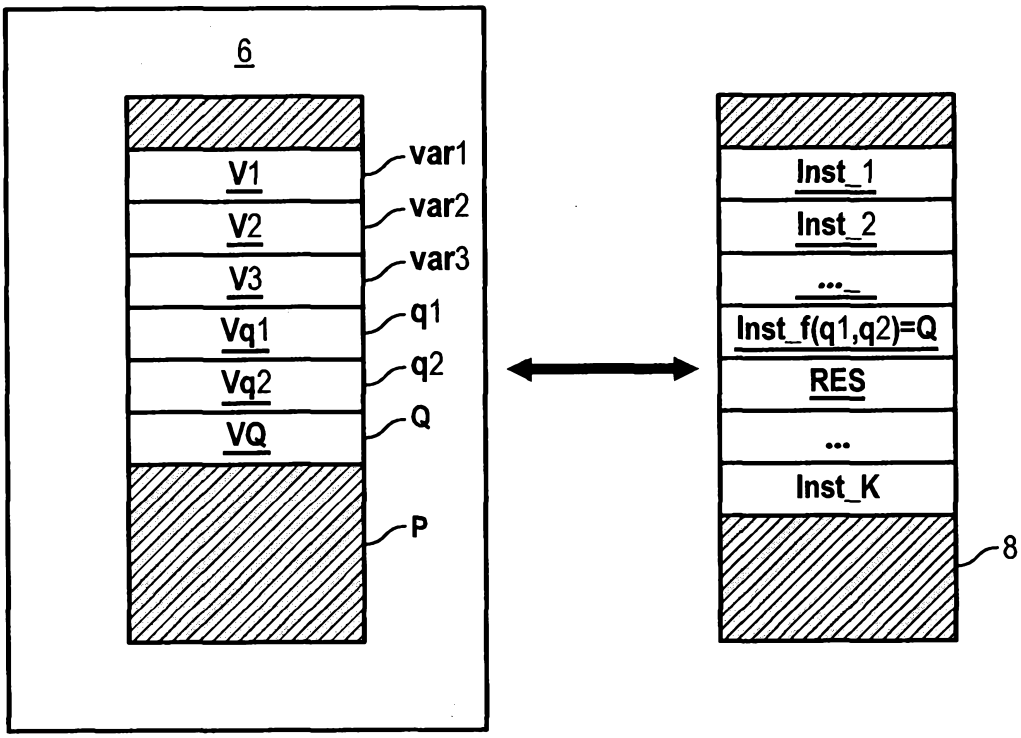

6
V1
V2
V3
Vq1
Vq2
VQ
P
var1
var2
var3
q1
q2
Q
Inst_1
Inst_2
Inst_f(q1,q2)=Q
RES
Inst_K
8

FIG. 3

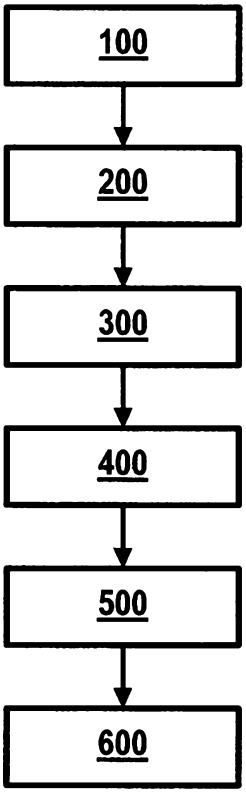
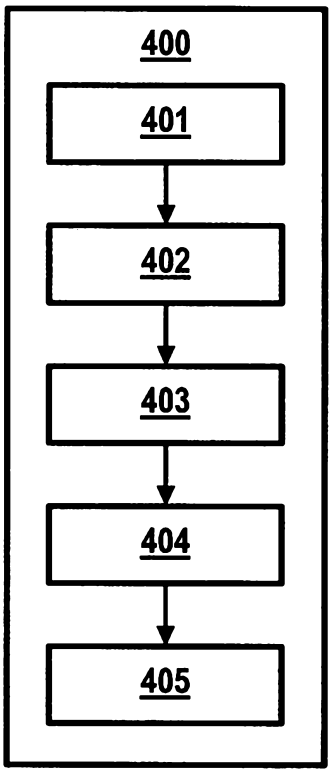


FIG. 4



N° d'enregistrement national : FR1755893 N° de publication : FR3068152

RAPPORT DE RECHERCHE

articles L.612-14, L.612-53 à 69 du code de la propriété intellectuelle

OBJET DU RAPPORT DE RECHERCHE

L'I.N.P.I. annexe à chaque brevet un "RAPPORT DE RECHERCHE" citant les éléments de l'état de la technique qui peuvent être pris en considération pour apprécier la brevetabilité de l'invention, au sens des articles L. 611-11 (nouveauté) et L. 611-14 (activité inventive) du code de la propriété intellectuelle. Ce rapport porte sur les revendications du brevet qui définissent l'objet de l'invention et délimitent l'étendue de la protection.
Après délivrance, l'I.N.P.I. peut, à la requête de toute personne intéressée, formuler un "AVIS DOCUMENTAIRE" sur la base des documents cités dans ce rapport de recherche et de tout autre document que le requérant souhaite voir prendre en considération.

CONDITIONS D'ETABLISSEMENT DU PRESENT RAPPORT DE RECHERCHE

[x] Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.

[x] Le demandeur a maintenu les revendications.

☐ Le demandeur a modifié les revendications.

☐ Le demandeur a modifié la description pour en éliminer les éléments qui n'étaient plus en concordance avec les nouvelles revendications.

☐ Les tiers ont présenté des observations après publication du rapport de recherche préliminaire.

☐ Un rapport de recherche préliminaire complémentaire a été établi.

DOCUMENTS CITES DANS LE PRESENT RAPPORT DE RECHERCHE

La répartition des documents entre les rubriques 1, 2 et 3 tient compte, le cas échéant, des revendications déposées en dernier lieu et/ou des observations présentées.

[x] Les documents énumérés à la rubrique 1 ci-après sont susceptibles d'être pris en considération pour apprécier la brevetabilité de l'invention.

☐ Les documents énumérés à la rubrique 2 ci-après illustrent l'arrière-plan technologique général.

☐ Les documents énumérés à la rubrique 3 ci-après ont été cités en cours de procédure, mais leur pertinence dépend de la validité des priorités revendiquées.

☐ Aucun document n'a été cité en cours de procédure.

1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION
EP 1 881 404 A1 (GEMPLUS CARD INT [FR]) 23 janvier 2008 (2008-01-23) WO 2008/125479 A1 (GEMPLUS [FR]; GAUTERON LAURENT [FR]) 23 octobre 2008 (2008-10-23)
2. ELEMENTS DE L'ETAT DE LA TECHNIQUE ILLUSTRANT L'ARRIERE-PLAN TECHNOLOGIQUE GENERAL
NEANT
3. ELEMENTS DE L'ETAT DE LA TECHNIQUE DONT LA PERTINENCE DEPEND DE LA VALIDITE DES PRIORITES
NEANT