(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** **3 060 807**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **16 62945**

(51) Int Cl[8] : ***G 06 F 21/50*** *(2017.01)*

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

(54) PROCEDE DE VERIFICATION DE L'INTEGRITE D'UN PROGRAMME, ENTITE ELECTRONIQUE ASSOCIEE ET APPAREIL ELECTRONIQUE COMPRENANT UNE TELLE ENTITE ELECTRONIQUE.

(22) **Date de dépôt :** 20.12.16.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 22.06.18 Bulletin 18/25.

(45) **Date de la mise à disposition du public du brevet d'invention :** 24.05.19 Bulletin 19/21.

(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) :** *OBERTHUR TECHNOLOGIES Société anonyme* — FR.

(72) **Inventeur(s) :** DOTTAX EMMANUELLE.

(73) **Titulaire(s) :** OBERTHUR TECHNOLOGIES Société anonyme.

(74) **Mandataire(s) :** JACOBACCI CORALIS HARLE Société par actions simplifiée.

FR 3 060 807 - B1

DOMAINE TECHNIQUE AUQUEL SE RAPPORTE L'INVENTION

La présente invention concerne la vérification de l'intégrité des programmes mémorisés et exécutables au sein d'un appareil électronique.

Elle concerne plus particulièrement un procédé de vérification de l'intégrité d'un programme, une entité électronique associée et un appareil électronique comprenant une telle entité électronique.

L'invention s'applique particulièrement avantageusement dans le cas où l'entité électronique est intégrée (par exemple soudée) à l'appareil électronique.

ARRIERE-PLAN TECHNOLOGIQUE

Dans le but de vérifier l'intégrité d'un programme mémorisé dans une mémoire d'un appareil électronique et exécutable par un processeur de l'appareil électronique, il a déjà été proposé qu'un serveur distant transmette un nombre aléatoire à l'appareil électronique afin que celui-ci retourne une preuve d'intégrité déterminée en fonction du nombre aléatoire et des données constituant le programme mémorisé.

Une telle solution est par exemple décrite dans l'article "*FPGA-Based Remote-Code Integrity Verification of Programs in Distributed Embedded Systems*", de C. Basile, S. Di Carlo et A. Scionti, in "*IEEE Transactions on Systems, Man, and Cybernetics – Part C : Applications and Reviews", Vol. 42, No. 2, March 2012*".

OBJET DE L'INVENTION

Dans ce contexte, la présente invention propose un procédé de vérification de l'intégrité d'un programme mémorisé dans une mémoire d'un appareil électronique et exécutable par un processeur de l'appareil électronique, caractérisé en ce qu'il comprend les étapes suivantes :

- réception, en provenance d'un serveur distant, d'une valeur de défi par une entité électronique (par exemple une entité électronique sécurisée) équipant l'appareil électronique ;

- comparaison, par l'entité électronique, d'une première donnée mémorisée dans l'entité électronique et d'une seconde donnée déterminée en fonction d'une partie au moins du programme ;

- en cas d'égalité entre la première donnée et la seconde donnée à

l'étape de comparaison, détermination, par l'entité électronique, d'une valeur de réponse en fonction de la valeur de défi et d'une clé mémorisée dans l'entité électronique ;

- émission de la valeur de réponse à destination du serveur distant.

Le serveur est ainsi assuré que la comparaison assurant la vérification d'intégrité a bien été effectuée par l'entité électronique associée à l'appareil électronique. En outre, la vérification d'intégrité étant effectuée par l'entité électronique, ceci allège le fonctionnement du serveur et évite la mémorisation au sein du serveur de données nécessaires à la vérification (qu'il s'agisse de paires défi/réponse, ou du programme à vérifier et d'une donnée cryptographique permettant le calcul de la réponse). De ce fait, cette solution est d'ailleurs pratique en particulier lorsque le serveur n'est pas lié à l'organisme fournissant le programme.

Selon un premier mode de réalisation envisageable, le procédé peut comprendre les étapes suivantes :

- réception, par l'entité électronique, de ladite partie au moins du programme ;

- détermination de la seconde donnée par l'entité électronique en fonction de ladite partie au moins du programme.

La seconde donnée est par exemple déterminée par application d'une fonction de hachage à ladite partie au moins du programme.

Selon un second mode de réalisation envisageable, le procédé peut comprendre les étapes suivantes :

- transmission d'un élément de diversification de l'entité électronique au processeur ;

- détermination de la seconde donnée par le processeur en fonction de l'élément de diversification et de ladite partie au moins du programme.

La seconde donnée est ainsi déterminée par le processeur, ce qui évite de transférer le programme (généralement volumineux) à l'entité électronique.

La seconde donnée est par exemple déterminée par application d'une fonction de hachage à la concaténation de l'élément de diversification et de ladite partie au moins du programme.

Selon d'autres caractéristiques envisageables de manière optionnelle (et donc non limitative) :

- la valeur de réponse est déterminée par application, à la valeur de défi reçue, d'un algorithme cryptographique (par exemple de signature) utilisant la clé mémorisée dans l'entité électronique ;

- l'entité électronique est un élément sécurisé ;

- l'entité électronique est intégrée (par exemple soudée) à l'appareil électronique.

L'invention propose également une entité électronique destinée à équiper un appareil électronique comprenant un processeur et une mémoire, et conçue pour mettre en œuvre les étapes suivantes :

- réception d'une valeur de défi ;

- comparaison d'une première donnée mémorisée dans l'entité électronique et d'une seconde donnée déterminée en fonction d'une partie au moins d'un programme mémorisé dans ladite mémoire et exécutable par ledit processeur ;

- en cas d'égalité entre la première donnée et la seconde donnée à l'étape de comparaison, détermination d'une valeur de réponse en fonction de la valeur de défi et d'une clé mémorisée dans l'entité électronique ;

- émission de la valeur de réponse.

Lorsque l'entité électronique comprend un processeur et une mémoire, la mémoire de l'entité électronique mémorise par exemple des instructions de programme exécutables par le processeur de l'entité électronique et conçues de sorte que l'entité électronique mette en œuvre les étapes de réception, de comparaison, de détermination et d'émission mentionnées ci-dessus lorsque ces instructions sont exécutées par le processeur.

L'invention propose enfin un appareil électronique comprenant un processeur, une mémoire et une entité électronique telle que définie ci-dessus.

DESCRIPTION DETAILLEE D'UN EXEMPLE DE REALISATION

La description qui va suivre en regard des dessins annexés, donnés à titre d'exemples non limitatifs, fera bien comprendre en quoi consiste l'invention et comment elle peut être réalisée.

Sur les dessins annexés :

- la figure 1 représente schématiquement les éléments principaux d'un système dans lequel peut être mise en œuvre l'invention ;

- la figure 2 est un logigramme représentant un procédé de chargement

d'un programme dans un appareil électronique ;

- la figure 3 est un logigramme représentant un premier exemple de procédé de vérification de l'intégrité d'un tel programme ; et

- la figure 4 est un logigramme représentant un second exemple de procédé de vérification de l'intégrité d'un tel programme.

La figure 1 représente schématiquement les éléments principaux d'un système dans lequel peut être mise en œuvre l'invention.

Dans un tel système, un appareil électronique 2 comprend un processeur 4 (par exemple un microprocesseur), une mémoire 6, un circuit de communication 8 et une entité électronique 10.

La mémoire 6 est une mémoire non-volatile réinscriptible, par exemple de type mémoire flash.

Comme cela sera expliqué plus en détail dans la suite, la mémoire 6 mémorise un programme P_n formé d'instructions exécutables par le processeur 4. Par mesure de concision, on écrira ainsi dans la suite que le programme P_n est exécutable par le processeur 4.

Lorsqu'il est exécuté par le processeur 4, le programme P_n permet la mise en œuvre de fonctionnalités par l'appareil électronique 2, par exemple des fonctionnalités prévues dans le cadre du fonctionnement normal de cet appareil électronique 2. Le programme P_n peut en outre permettre la mise en œuvre, par le processeur 4 de l'appareil électronique 2, de l'un au moins des procédés décrits ci-dessous en référence aux figures 2 à 4.

Le circuit de communication 8 est conçu de manière à établir une connexion de l'appareil électronique 2 à un réseau de communication, ici un réseau de communication public I tel que le réseau Internet.

Par souci de simplification, on a représenté sur la figure 1 une connexion directe entre le circuit de communication 8 et le réseau de communication I. On peut toutefois prévoir en pratique que cette connexion s'établisse par l'intermédiaire d'autres dispositifs électroniques. Par ailleurs, la connexion entre le circuit de communication 8 et le réseau de communication I pourra se faire au moyen de liaisons de différents types, par exemple des liaisons sans fil et/ou des liaisons filaires.

Du fait de la connexion établie par le circuit de communication 8 au réseau de communication I, le processeur 4 peut échanger des données avec

d'autres équipements connectés au réseau de communication I, en particulier avec un serveur distant 20, comme décrit dans la suite.

Dans l'exemple décrit, l'entité électronique 10 est une entité électronique sécurisée, en l'occurrence un élément sécurisé (ou SE pour "*Secure Element*"), réalisé ici sous la forme d'un microcontrôleur. Un tel élément sécurisé peut éventuellement être intégré à l'appareil électronique 2, par exemple en étant soudé au sein de l'appareil électronique : l'entité électronique 10 est alors de type eSE (pour "*embedded Secure Element*").

En variante, l'entité électronique 10 pourrait être une carte à microcircuit (par exemple une carte à microcircuit universelle ou UICC pour "*Universal Integrated Circuit Card*", telle que visée dans la spécification technique ETSI TS 102 221), ou une carte à microcircuit universelle soudée ou eUICC pour "*embedded Universal Circuit Card*" (telle que décrite dans la norme ETSI TS 103 383).

L'entité électronique 10 comprend un processeur (par exemple un microprocesseur) et au moins une mémoire, notamment ici une mémoire non-volatile réinscriptible. Cette mémoire mémorise en particulier des instructions conçues de telle sorte que l'entité électronique 10 puisse mettre en œuvre l'un au moins des procédés décrits ci-dessous en références aux figures 2 à 4 lorsque ces instructions sont exécutées par le processeur de l'entité électronique 10.

La mémoire de l'entité électronique 10 mémorise également des données utilisées au cours de son fonctionnement, en particulier une clé publique PK_S associée au serveur 20, une clé privée SK spécifique à l'entité électronique 10 et un condensat H_n associé au programme P_n mémorisé dans la mémoire 6 de l'appareil électronique 2 comme mentionné ci-dessus. Dans certains modes de réalisation décrits ci-dessous, l'entité électronique 10 peut également mémoriser une clé secrète de déchiffrement S_{DEC}.

Comme déjà indiqué, l'entité électronique 10 est ici sécurisée, c'est-à-dire que cette entité électronique 10 est conçue, du fait de sa construction physique et de la conception des programmes d'ordinateur qu'elle mémorise, de façon à rendre très difficile, voire impossible, pour un attaquant l'accès (par lecture et/ou modification) aux données confidentielles qu'elle mémorise. Ainsi, l'entité électronique 10 présente un niveau de sécurité conforme aux critères communs EAL (pour "*Evaluation Assurance Level*"), correspondant à la norme ISO 15408,

avec un niveau supérieur ou égal à 4, ou à la norme FIPS (pour "*Federal Information Processing Standard*") 140-2.

Lors d'une phase de préparation de l'appareil électronique 2 en vue de son fonctionnement (ou phase de personnalisation), une version initiale P_0 du programme embarqué dans l'appareil électronique 2 est écrite dans la mémoire 6 et un condensat correspondant H_0 est écrit dans la mémoire de l'entité électronique 10.

Le condensat H_0 est par exemple produit par application d'une fonction de hachage G à la version initiale P_0 : $H_0 = G(P_0)$. Cette fonction de hachage G peut être par exemple une fonction de la famille SHA-2 (NIST FIPS 180-4), ou de la famille SHA-3 (NIST FIPS 202). On pourrait également utiliser une fonction de type MAC (acronyme pour "*Message Authentication Code*"), par exemple de type HMAC (NIST FIPS 198-1) ou CMAC (NIST SP800-38B). Une telle fonction de type MAC utilise alors dans ce cas une clé secrète mémorisée dans l'entité électronique 10

La figure 2 est un logigramme représentant un procédé de chargement d'un programme P_n dans l'appareil électronique 2 afin de mettre à jour le programme P_{n-1} embarqué dans cet appareil électronique 2.

Ce procédé débute à l'étape E2 par l'émission, par le serveur 20 et à destination du processeur 4 de l'appareil électronique 2, du programme P_n et d'une signature (électronique) S_n de ce programme P_n. (Une telle signature S_n est par exemple générée au niveau du serveur 20 par application aux données formant le programme P_n d'un algorithme cryptographique de signature utilisant une clé privée associée à la clé publique PK_S déjà mentionnée.)

Le processeur 4 reçoit le programme P_n et la signature S_n et les transmet à l'entité électronique 10 à l'étape E4.

L'entité électronique 10 reçoit ainsi le programme P_n et la signature S_n à l'étape E6.

L'entité électronique 10 vérifie la signature S_n à l'étape E8 : pour ce faire, l'entité électronique 10 met en œuvre un algorithme cryptographique de vérification de signature impliquant le programme reçu P_n, la signature reçue S_n et la clé publique PK_S mémorisée dans la mémoire de l'entité électronique 10 (comme déjà indiqué).

Si la vérification de signature échoue à l'étape E8, le procédé se poursuit

par une étape E10 de traitement d'erreur : l'entité électronique 10 renvoie par exemple un message d'erreur au processeur 4 et le processeur 4 (sur lequel s'exécute le programme embarqué P_{n-1}) ne procède pas dans ce cas à l'installation du programme P_n reçu.

Si la vérification de signature réussit à l'étape E8, l'entité électronique 10 détermine un condensat H_n associé au programme P_n, ici par application de la fonction de hachage G (*i.e.* $H_n=G(P_n)$), et mémorise ce condensat H_n dans la mémoire de l'entité électronique 10 (étape E12).

On peut prévoir dans certains modes de réalisation de l'étape E12 (dont celui permettant la mise en œuvre ultérieure du procédé décrit plus bas en référence à la figure 4) que l'entité électronique 10 génère (par exemple par tirage aléatoire) une pluralité d'éléments de diversification r_i, puis détermine un condensat $H_{n,i}$ associé au programme P_n pour chaque élément de diversification r_i et mémorise les paires (r_i, $H_{n,i}$) formées chacune d'un élément de diversification r_i et du condensat correspondant $H_{n,i}$.

Le condensat $H_{n,i}$ est par exemple déterminé en appliquant une fonction de hachage F à la concaténation de l'élément de diversification associé r_i et du programme P_n concerné : $H_{n,i} = F(r_i \,||\, P_n)$, où || est l'opérateur de concaténation. La fonction F est ici résistante aux attaques par recherche de second antécédent (en anglais "*second-preimage resistant*"), c'est-à-dire que connaissant x, il est impossible en pratique de trouver un x' différent de x et tel que F(x) = F(x').

La fonction de hachage F peut être par exemple une fonction de la famille SHA-2 (NIST FIPS 180-4), ou de la famille SHA-3 (NIST FIPS 202). On pourrait également utiliser une fonction de type MAC (acronyme pour "*Message Authentication Code*"), par exemple de type HMAC (NIST FIPS 198-1) ou CMAC (NIST SP800-38B).

En variante, le condensat $H_{n,i}$ peut être déterminé en appliquant une fonction de hachage F' de type MAC, par exemple de type HMAC (NIST FIPS 198-1) ou CMAC (NIST SP800-38B), au programme P_n concerné en utilisant l'élément de diversification r_i concerné en tant que clé sécrète dans cette fonction de type MAC : $H_{n,i} = F'_{ri}(P_n)$.

Selon une autre variante envisageable, le condensat H_n (ou les paires élément de diversification r_i – condensat $H_{n,i}$) peu(ven)t être déterminé(es) au préalable au sein du serveur 20 et transmis(es) à l'entité électronique 10 (par

exemple annexé(es) au programme Pn) sous forme chiffrée, par exemple par application (au niveau du serveur 20) d'un algorithme cryptographique de chiffrement utilisant la clé publique associée à la clé secrète de déchiffrement S_{DEC} (dans le cas où cette clé secrète de déchiffrement S_{DEC} est une clé privée de chiffrement asymétrique ; en variante, la clé secrète de déchiffrement S_{DEC} pourrait être une clé symétrique partagée entre l'entité électronique 10 et le serveur 20, auquel cas l'algorithme cryptographique de chiffrement précité utiliserait lui aussi cette clé secrète de déchiffrement S_{DEC}). Le condensat H_n reçu (ou les paires r_i, $H_{n,i}$ reçues) est (sont) dans ce cas déchiffré(es) par l'entité électronique 10 au moyen d'un algorithme cryptographique de déchiffrement utilisant la clé secrète de déchiffrement S_{DEC}, puis mémorisé(es) dans la mémoire de l'entité électronique 10.

On peut prévoir en particulier dans ce cas que le condensat H_n (ou les condensats $H_{n,i}$) soi(en)t obtenu(s) par application d'une fonction de hachage à l'ensemble du contenu prévu de la mémoire 6 après mémorisation du programme P_n (et non au seul programme P_n).

Par ailleurs, dans ces cas où le condensat H_n (ou les condensats $H_{n,i}$) est (sont) transmi(s) (chiffré(s)) du serveur 20 à l'appareil électronique 2 avec le programme Pn, il(s) est (sont) également couvert(s) par la signature S_n susmentionnée.

Lorsque le condensat H_n (ou les condensats $H_{n,i}$) est (sont) mémorisé(s) dans l'entité électronique 10, l'entité électronique 10 émet un message OK indicatif de bon fonctionnement à destination du processeur 4 (étape E14).

Le processeur 4 reçoit ce message OK à l'étape E16 et procède alors à la mise à jour du programme embarqué au moyen du programme reçu P_n (étape E18), par exemple en mémorisant le programme reçu P_n dans la mémoire 6 susmentionnée.

La figure 3 est un logigramme représentant un premier exemple de procédé de vérification de l'intégrité du programme P_n embarqué dans l'appareil électronique 2.

Ce procédé débute à l'étape E20 à laquelle le serveur 20 envoie une valeur de défi CH à destination de l'appareil électronique 2. Le serveur 20 envoie par exemple la valeur de défi CH au sein d'une requête de vérification d'intégrité du programme embarqué P_n destinée au processeur 4.

Le processeur 4 reçoit cette requête et la valeur de défi CH et initie alors le processus de vérification d'intégrité du programme embarqué P_n en transmettant la valeur de défi reçue CH et les données formant le programme embarqué P_n (mémorisées dans la mémoire 6) à l'entité électronique 10 (étape E22).

L'entité électronique 10 reçoit les données formant le programme embarqué P_n et la valeur de défi CH à l'étape E24.

L'entité électronique 10 vérifie alors à l'étape E26 que l'application de la fonction de hachage G aux données formant le programme embarqué P_n produit bien un résultat identique au condensat H_n mémorisé dans la mémoire de l'entité électronique 10, ce qui est le cas si le programme embarqué P_n n'a pas été altéré pas un attaquant puisqu'on a normalement comme indiqué ci-dessus :

$H_n = G(P_n)$.

Pour que la vérification d'intégrité du programme P_n assure l'absence d'un programme malveillant au sein de l'appareil électronique 2, il convient de s'assurer que la mémoire 6 n'a pas une capacité suffisante pour mémoriser un tel programme malveillant en plus des données formant le programme P_n dont on vérifie l'intégrité. En effet, si cette condition n'était pas respectée, le programme malveillant pourrait en théorie recevoir la requête de vérification d'intégrité et transmettre à l'entité électronique 10 les données formant le programme P_n (ce qui entraînerait une vérification de l'intégrité).

Si la vérification échoue à l'étape E26, l'entité électronique 10 effectue un traitement d'erreur à l'étape E28, par exemple en renvoyant un message d'échec au processeur 4. Le processeur 4 peut éventuellement transmettre à son tour dans ce cas un message d'erreur au serveur 20 (cela n'étant toutefois pas nécessaire, puisqu'en l'absence de réception d'une réponse correcte à l'étape E36 décrite ci-dessous, le serveur 20 considère qu'un problème est survenu).

Si la vérification réussit à l'étape E30, l'entité électronique 10 détermine une valeur de réponse R en fonction de la valeur de défi CH et d'une clé secrète mémorisée dans l'entité électronique 10, ici la clé privée SK. La valeur de réponse R est par exemple obtenue par application à la valeur de défi CH d'un algorithme cryptographique (ici de signature) utilisant la clé secrète (ici la clé privée SK).

L'entité électronique 10 transmet à l'étape E32 la valeur de réponse R au processeur 4. Le processeur 4 peut ainsi envoyer à l'étape E34 la valeur de

réponse R (ici en tant que réponse à la requête de vérification mentionnée plus haut) au serveur 20.

Le serveur 20 reçoit la valeur de réponse R à l'étape E36.

Le serveur 20 peut donc vérifier que cette valeur de réponse R a bien été émise par l'entité électronique 10 sur la base de la valeur de défi CH (étape E38), ici en appliquant à la valeur de défi CH et à la valeur de réponse R un algorithme de vérification de signature utilisant la clé publique associée à la clé privée SK.

Si la valeur de réponse R est la réponse attendue (*i.e.* associée à la valeur de défi CH envoyée à l'étape E20), ceci indique que l'entité électronique 10 a effectivement vérifié l'intégrité du programme P_n à l'étape E26 et l'appareil électronique 2 peut donc continuer à être utilisé sans risque. Le serveur 20 peut ultérieurement lancer (à l'étape E20) une nouvelle vérification de l'intégrité du programme embarqué P_n.

Si la valeur de réponse R n'est pas la valeur attendue (ou si aucune réponse n'est reçue de l'appareil électronique 20 après un laps de temps donné par rapport à l'étape E20), ceci indique que l'entité électronique 10 n'a pas vérifié l'intégrité du programme embarqué P_n et on considère donc alors que le programme embarqué P_n est altéré (que ce soit fortuitement ou suite à une attaque).

On procède donc dans ce cas à une étape E40 de traitement du défaut d'intégrité, qui peut comprendre une ou plusieurs des actions suivantes selon les circonstances :

- lancement d'une mise à jour du programme embarqué (comme décrit ci-dessus à la figure 2) ;
- annulation de droits ou autorisations associé(s) à l'appareil électronique 2 ;
- signalisation d'un risque d'attaque à un utilisateur de l'appareil électronique 2.

La figure 4 est un logigramme représentant un second exemple de procédé de vérification de l'intégrité du programme P_n embarqué dans l'appareil électronique 2.

Ce procédé débute à l'étape E50 à laquelle le serveur 20 envoie une valeur de défi CH à destination de l'appareil électronique 2. Le serveur 20 envoie par exemple la valeur de défi CH au sein d'une requête de vérification d'intégrité

du programme embarqué P_n destinée au processeur 4.

Le processeur 4 reçoit cette requête et la valeur de défi CH et initie alors le processus de vérification d'intégrité du programme embarqué P_n en transmettant la valeur de défi reçue CH à l'entité électronique 10 (étape E52).

L'entité électronique 10 reçoit la valeur de défi CH à l'étape E54.

L'entité électronique 10 choisit alors (étape E55) un élément de diversification r_i parmi la pluralité d'éléments de diversification mémorisés dans la mémoire de l'entité électronique (voir ci-dessus, dans la description relative à l'étape E12, l'utilisation possible de tels éléments de diversification r_i).

L'élément de diversification r_i est par exemple choisi aléatoirement parmi les éléments de diversification mémorisés. En variante, les éléments de diversification sont successivement choisis au fur et à mesure des différentes mises en œuvre du procédé de vérification d'intégrité décrit ici.

L'entité électronique 10 transmet l'élément de diversification choisi r_i au processeur 4 à l'étape E56.

Le processeur 4 reçoit l'élément de diversification r_i à l'étape E58 et peut ainsi déterminer à l'étape E60 le condensat $H'_{n,i}$ associé à cet élément de diversification r_i et à la version courante P_n du programme embarqué. Dans l'exemple décrit ici, le condensat $H'_{n,i}$ est déterminé comme suit en utilisant la fonction de hachage F mentionnée plus haut (dans le cadre de la détermination du condensat correspondant $H_{n,i}$ mémorisé dans l'entité électronique 10) :

$H'_{n,i} = F(r_i \,||\, P_n)$.

Le processeur 4 transmet alors à l'entité électronique 10 le condensat $H'_{n,i}$ ainsi déterminé (étape E62).

L'entité électronique 10 reçoit à l'étape E64 le condensat $H'_{n,i}$ déterminé par le processeur 4 et compare à l'étape E66 ce condensat $H'_{n,i}$ au condensat $H_{n,i}$ mémorisé (lors de l'étape E12 décrite plus haut) dans la mémoire de l'entité électronique 10 en association avec l'élément de diversification r_i (choisi à l'étape E55).

Si le condensat $H'_{n,i}$ reçu par l'entité électronique 10 diffère du condensat $H_{n,i}$ mémorisé dans l'entité électronique 10, cela indique que le programme P_n a été altéré et on procède alors à un traitement d'erreur à l'étape E68, par exemple en renvoyant un message d'échec au processeur 4. Le processeur 4 peut éventuellement transmettre à son tour dans ce cas un message d'erreur au

serveur 20 (ce qui n'est toutefois pas nécessaire, puisqu'en l'absence de réception d'une réponse correcte à l'étape E76 décrite ci-dessous, le serveur 20 considère qu'un problème est survenu).

Si le condensat $H'_{n,i}$ reçu par l'entité électronique 10 est identique au condensat $H_{n,i}$ mémorisé dans l'entité électronique 10, l'entité électronique 10 détermine une valeur de réponse R en fonction de la valeur de défi CH (reçue à l'étape E54) et d'une clé secrète mémorisée dans l'entité électronique 10, ici la clé privée SK. La valeur de réponse R est par exemple obtenue par application à la valeur de défi CH d'un algorithme cryptographique (ici de signature) utilisant la clé secrète (ici la clé privée SK).

On remarque que cette solution permet une vérification d'intégrité du programme P_n par l'entité électronique 10 sans avoir à transférer l'intégralité du programme P_n à l'entité électronique 10.

L'entité électronique 10 transmet à l'étape E72 la valeur de réponse R au processeur 4. Le processeur 4 peut ainsi envoyer à l'étape E74 la valeur de réponse R (ici en tant que réponse à la requête de vérification mentionnée plus haut) au serveur 20.

Le serveur 20 reçoit la valeur de réponse R à l'étape E76.

Le serveur 20 peut donc vérifier que cette valeur de réponse R a bien été émise par l'entité électronique 10 sur la base de la valeur de défi CH (étape E78), ici en appliquant à la valeur de défi CH et à la valeur de réponse R un algorithme de vérification de signature utilisant la clé publique associée à la clé privée SK.

Si la valeur de réponse R est la réponse attendue (*i.e.* associée à la valeur de défi CH envoyée à l'étape E50), ceci indique que l'entité électronique 10 a effectivement vérifié l'intégrité du programme P_n à l'étape E66 et l'appareil électronique 2 peut donc continuer à être utilisé sans risque. Le serveur 20 peut ultérieurement lancer (à l'étape E50) une nouvelle vérification de l'intégrité du programme embarqué P_n.

Si la valeur de réponse R n'est pas la valeur attendue (ou si aucune réponse n'est reçue de l'appareil électronique 20 après un laps de temps donné par rapport à l'étape E50), ceci indique que l'entité électronique 10 n'a pas vérifié l'intégrité du programme embarqué P_n et on considère donc alors que le programme embarqué P_n est altéré (que ce soit fortuitement ou suite à une attaque).

On procède donc dans ce cas à une étape E80 de traitement du défaut d'intégrité, qui peut comprendre une ou plusieurs des actions suivantes selon les circonstances :

- lancement d'une mise à jour du programme embarqué (comme décrit ci-dessus à la figure 2) ;
- annulation de droits ou autorisations associé(s) à l'appareil électronique 2 ;
- signalisation d'un risque d'attaque à un utilisateur de l'appareil électronique 2.

REVENDICATIONS

1. Procédé de vérification de l'intégrité d'un programme (P_n) mémorisé dans une mémoire (6) d'un appareil électronique (2) et exécutable par un processeur (4) de l'appareil électronique (2), caractérisé en ce qu'il comprend les étapes suivantes :

- réception (E24 ; E54), en provenance d'un serveur distant (20), d'une valeur de défi (CH) par une entité électronique (10) équipant l'appareil électronique (2) ;
- comparaison (E26 ; E66), par l'entité électronique (10), d'une première donnée (H_n) mémorisée dans l'entité électronique (10) et d'une seconde donnée ($H'_{n,i}$) déterminée en fonction d'une partie au moins du programme (P_n) ;
- en cas d'égalité entre la première donnée (H_n) et la seconde donnée ($H'_{n,i}$) à l'étape de comparaison (E26 ; E66), détermination (E30 ; E70), par l'entité électronique (10), d'une valeur de réponse (R) en fonction de la valeur de défi (CH) et d'une clé (SK) mémorisée dans l'entité électronique (10) ;
- émission de la valeur de réponse (R) à destination du serveur distant (20).

2. Procédé de vérification selon la revendication 1, comprenant les étapes suivantes :

- réception (E24), par l'entité électronique (10), de ladite partie au moins du programme (P_n) ;
- détermination (E26) de la seconde donnée par l'entité électronique (10) en fonction de ladite partie au moins du programme (P_n).

3. Procédé de vérification selon la revendication 2, dans lequel la seconde donnée est déterminée par application d'une fonction de hachage à ladite partie au moins du programme (P_n).

4. Procédé de vérification selon la revendication 1, comprenant les étapes suivantes :

- transmission (E56) d'un élément de diversification (r_i) de l'entité électronique (10) au processeur (4) ;

- détermination (E60) de la seconde donnée ($H'_{n,i}$) par le processeur (4) en fonction de l'élément de diversification (r_i) et de ladite partie au moins du programme (P_n).

5. Procédé de vérification selon la revendication 4, dans lequel la seconde donnée ($H'_{n,i}$) est déterminée par application d'une fonction de hachage à la concaténation de l'élément de diversification (r_i) et de ladite partie au moins du programme (P_n).

6. Procédé de vérification selon l'une des revendications 1 à 5, dans lequel la valeur de réponse (R) est déterminée par application, à la valeur de défi (CH) reçue, d'un algorithme cryptographique utilisant la clé mémorisée (SK) dans l'entité électronique (10).

7. Procédé de vérification selon l'une des revendications 1 à 6, dans lequel l'entité électronique (10) est un élément sécurisé.

8. Procédé de vérification selon l'une des revendications 1 à 7, dans lequel l'entité électronique (10) est intégrée à l'appareil électronique (2).

9. Entité électronique (10) destinée à équiper un appareil électronique (2) comprenant un processeur (4) et une mémoire (6), et conçue pour mettre en œuvre les étapes suivantes :

- réception (E24 ; E54) d'une valeur de défi (CH) ;
- comparaison (E26 ; E66) d'une première donnée (H_n) mémorisée dans l'entité électronique (10) et d'une seconde donnée ($H'_{n,i}$) déterminée en fonction d'une partie au moins d'un programme (P_n) mémorisé dans ladite mémoire (6) et exécutable par ledit processeur (4) ;
- en cas d'égalité entre la première donnée et la seconde donnée à l'étape de comparaison (E26 ; E66), détermination (E30 ; E70) d'une valeur de réponse (R) en fonction de la valeur de défi (CH) et d'une clé (SK) mémorisée dans l'entité électronique (10) ;
- émission de la valeur de réponse (R).

10. Appareil électronique (2) comprenant un processeur (4), une mémoire (6) et une entité électronique (10) selon la revendication 9.

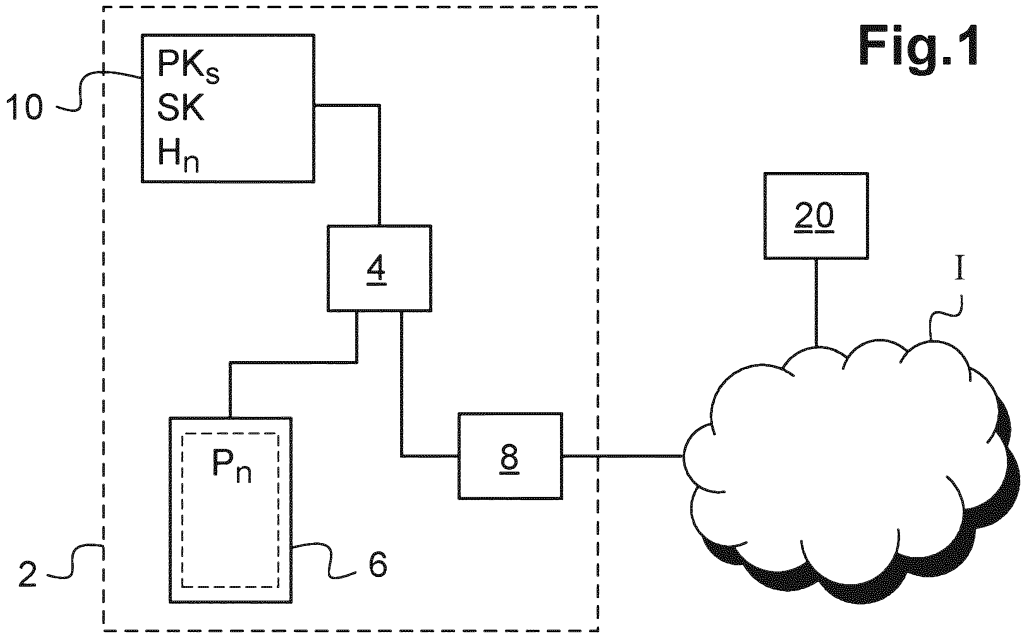
10
PKs
SK
Hn
4
20
I
Pn
8
2
6

Fig.1

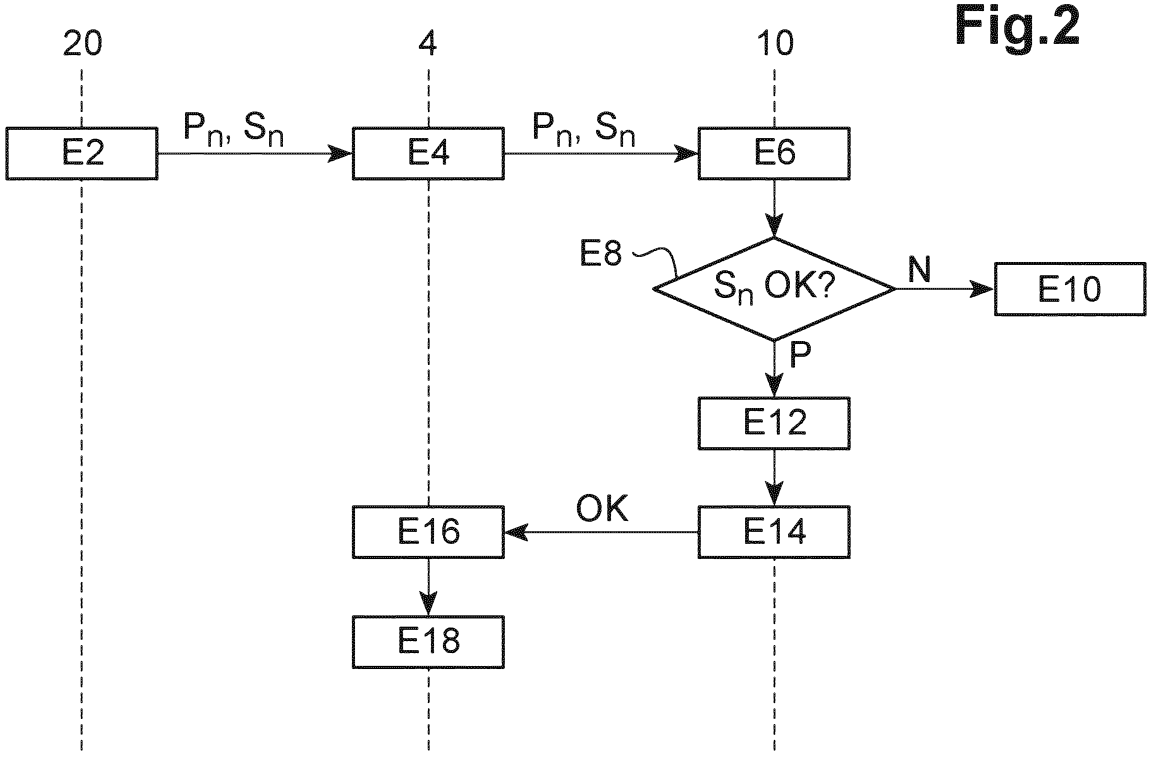
20
4
10
E2
Pn, Sn
E4
Pn, Sn
E6
E8
Sn OK?
N
E10
P
E12
OK
E16
E14
E18

Fig.2

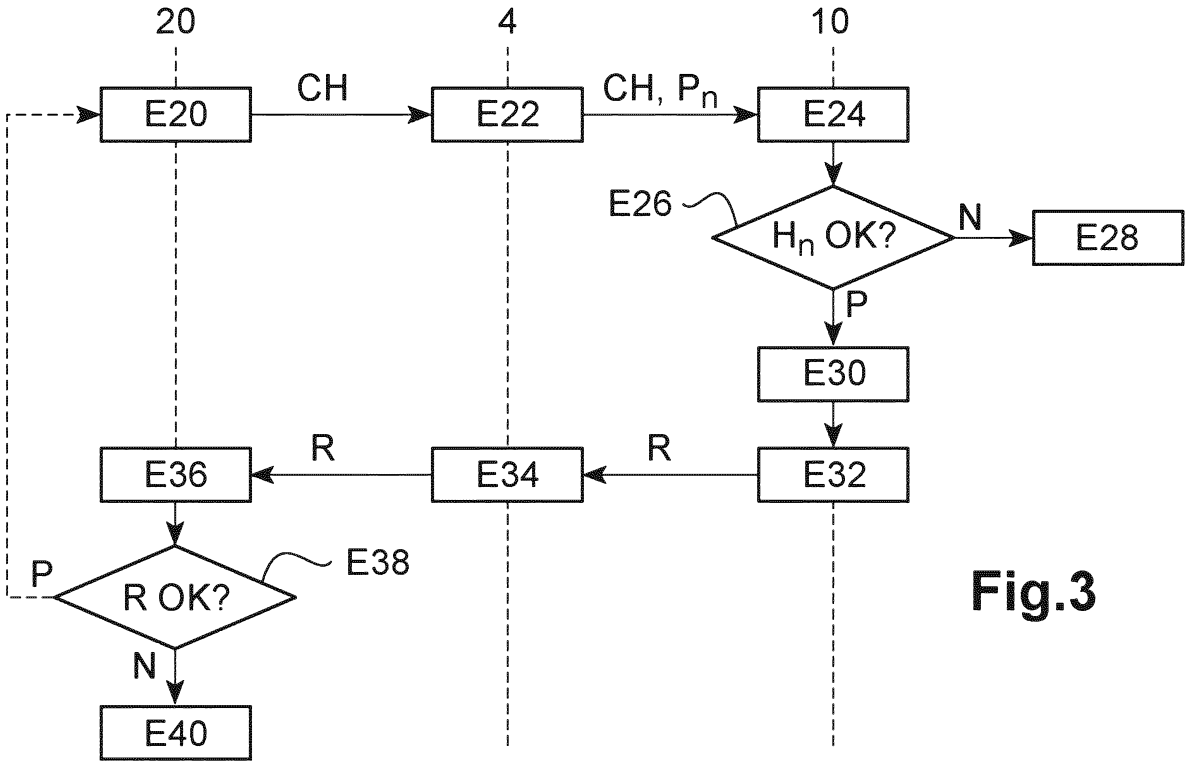
20
4
10
E20
CH
E22
CH, Pn
E24
E26
Hn OK?
N
E28
P
E30
E36
R
E34
R
E32
P
R OK?
E38
N
E40

Fig.3

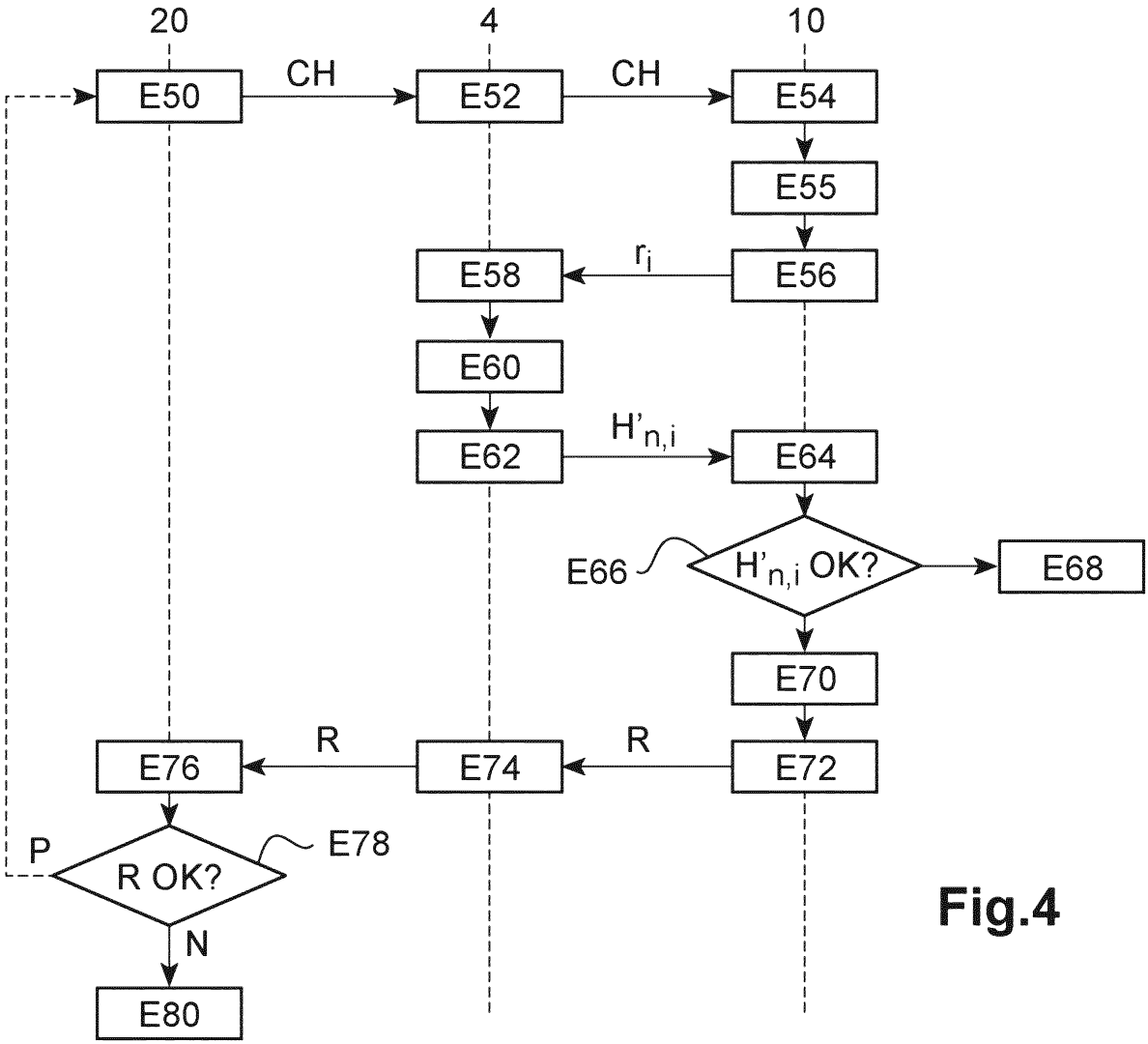
20
4
10
E50
CH
E52
CH
E54
E55
E58
ri
E56
E60
E62
H'n,i
E64
E66
H'n,i OK?
E68
E70
E76
R
E74
R
E72
P
R OK?
E78
N
E80

Fig.4

N° d'enregistrement national : 1662945 N° de publication : 3060807

RAPPORT DE RECHERCHE

articles L.612-14, L.612-17 et R.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'ÉTABLISSEMENT DU PRÉSENT RAPPORT DE RECHERCHE

- ☒ Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.
- ☒ 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 CITÉS DANS LE PRÉSENT 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.

- ☒ 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.

N° d'enregistrement national : 1662945 N° de publication : 3060807

1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION

BASILE C ET AL: "FPGA-Based Remote-Code Integrity Verification of Programs in Distributed Embedded Systems", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART C:APPLICATIONS AND REVIEWS, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 42, no. 2, 1 mars 2012 (2012-03-01), pages 187-200, XP011469391,
ISSN: 1094-6977, DOI: 10.1109/TSMCC.2011.2106493

EP 2 840 492 A1 (BRITISH TELECOMM [GB])
25 février 2015 (2015-02-25)

FR 3 030 831 A1 (OBERTHUR TECH [FR])
24 juin 2016 (2016-06-24)

WO 2016/181152 A1 (CRITICAL BLUE LTD [GB])
17 novembre 2016 (2016-11-17)

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