(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 606 185**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **86 15324**

(51) Int Cl⁴ : G 06 F 12/14; G 06 K 19/02.

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 4 novembre 1986.

(30) **Priorité :**

(43) **Date de la mise à disposition du public de la demande :** BOPI « Brevets » n° 18 du 6 mai 1988.

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

(71) **Demandeur(s) :** *Société dite : EUROTECHNIQUE.* — FR.

(72) **Inventeur(s) :** Gilles Lisimaque.

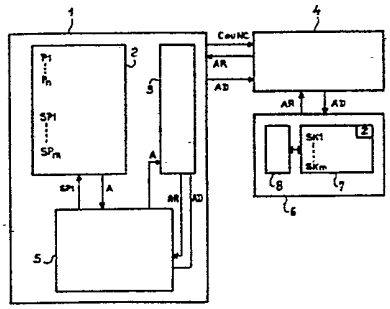
(73) **Titulaire(s) :**

(74) **Mandataire(s) :** Lydie Borin, Thomson-CSF.

(54) Procédé et dispositif de stockage d'un logiciel livré par un fournisseur à un utilisateur.

(57) L'invention concerne un procédé et un dispositif de stockage d'un logiciel, le procédé consistant à stocker ledit logiciel c'est-à-dire les programmes (P1,... Pn) et sous-programmes (SP1,... SPm) dans la mémoire de programmes d'un calculateur 1, les sous-programmes étant des versions incomplètes mais fonctionnelles; et à stocker des sous-programmes version complète dans la mémoire non volatile d'un circuit intégré 6 extérieur dont la lecture est interdite, seule l'exécution des sous-programmes étant autorisée pour un utilisateur habilité; à déclencher l'exécution des sous-programmes incomplets dans le cas où le circuit intégré n'est pas couplé au calculateur et dans le cas où l'utilisateur n'est pas habilité ou à déclencher l'exécution des sous-programmes version complète pour le circuit intégré dans le cas où le circuit 6 est couplé au calculateur 1 et où l'utilisateur est habilité.

Application à la protection des logiciels.



FR 2 606 185 - A1

Vente des fascicules à l'IMPRIMERIE NATIONALE, 27, rue de la Convention — 75732 PARIS CEDEX 15

PROCEDE ET DISPOSITIF DE STOCKAGE D'UN LOGICIEL LIVRE PAR UN FOURNISSEUR A UN UTILISATEUR

L'invention concerne un procédé et un dispositif de stockage d'un logiciel livré par un fournisseur à un utilisateur.

Généralement, lorsqu'un logiciel est livré par un fournisseur à un utilisateur, le fournisseur entend rester maître de ce logiciel en le rendant inexécutable dans sa forme telle que livré. L'exécution du logiciel est subordonnée à la livraison d'une clé de validation.

Cette protection que le fournisseur cherche à obtenir ne vise pas le contenu même du logiciel mais vise à donner au détenteur d'un logiciel l'assurance qu'il pourra utiliser ce logiciel alors que des utilisateurs potentiels non habilités ne le pourront pas.

Pour cela, une solution consiste à rendre inexécutables les programmes en l'état tels que livrés aux utilisateurs ; à délivrer à chaque utilisateur au moins un support annexe qui pourrait être une carte comprenant au moins des circuits de traitement et une zone mémoire où est enregistré un code secret seul connu du fournisseur et propre à chaque utilisateur ; et pour chaque utilisateur, à associer à chaque programme une clé de validation prédéterminée définie en fonction du programme et du code secret contenu dans la carte de l'utilisateur pour rendre ce programme exécutable une fois la carte couplée au dispositif susceptible d'exécuter le programme.

Une telle solution ne satisfait pas entièrement les fournisseurs de logiciel. En effet, certains fournisseurs préféreraient pendant la phase de négociation d'une location ou d'une vente d'un logiciel, laisser la possibilité à l'utilisateur potentiel de disposer de ce logiciel pour que celui-ci puisse bien connaître ses fonctions sans toutefois lui remettre la carte.

Or cela est impossible avec la solution connue, puisque le logiciel est inexécutable lorsque la carte (support annexe)

comportant le code secret n'est pas couplée au dispositif susceptible d'exécuter le ou les programmes du logiciel. Par ailleurs, si le fournisseur prête le logiciel et la carte, il prend le risque de ne plus être maître de ce logiciel et d'avoir programmé une carte inutilement si le client, après avoir pris connaissance du logiciel, n'est plus intéressé.

Il se peut en effet que le client se dise inintéressé et qu'il ait cependant, dupliqué les programmes pour les exécuter après avoir simulé les clés de validation qui se trouvaient sur la carte.

La présente invention permet de remédier à ces inconvénients en proposant un procédé de stockage de logiciel qui consiste ;

- à stocker un logiciel comportant des programmes (P1,... Pi,... Pn), faisant appel à des sous programmes complets (SK1,... SKj,... SKm), dans la mémoire de programme d'un calculateur ;

- à stocker des sous programmes (SP1,... SPj,... SPm) dans la mémoire de programmes du calculateur, ces sous programmes correspondant à des versions incomplètes mais fonctionnelles des sous programmes appelés ;

- à stocker dans un circuit intégré extérieur les sous programmes complets (SK1,... SKJ,... SKm), ce circuit étant apte à être couplé au calculateur et à stocker dans ce circuit un code d'habilitation destiné à être comparé à un code d'accès fourni par l'utilisateur ;

- seuls les sous programmes incomplets étant exécutés dans le cas où le code d'accès ne correspond pas au code d'habilitation :

- les sous programmes complets étant susceptibles d'être exécutés par le circuit intégré, dans le cas où le code d'accès correspond au code d'habilitation.

La présente invention permet donc de proposer un logiciel de démonstration comportant un ou des programmes principaux (P1,... Pi,... Pn) et des sous programmes (SP1,... SPj,...

SPm) exécutables par un organe de traitement tel qu'un calculateur dont dispose un client potentiel. Ce même logiciel est complété lorsque le client a loué ou acheté le logiciel, par la remise d'un circuit intégré (par exemple d'une carte à mémoire) et d'un organe périphérique apte à permettre le dialogue entre le calculateur et le circuit intégré. Le circuit intégré comporte les sous programmes (SK1,... SKm) appelés par le ou les programmes principaux et qui correspondent à une version complète des sous programmes (SP1,... SPm).

Le circuit intégré est apte à exécuter les sous programmes qui sont stockés dans sa mémoire et à transmettre les résultats au programme en cours d'exécution dans le calculateur du client. Ce circuit est dit "de sécurité" car le contenu de sa mémoire n'est pas accessible en lecture (ni en écriture). En effet ce contenu ne peut pas être transmis de la mémoire vers les bornes extérieures du circuit intégré. Seule par conséquent, l'exécution des sous-programmes par le circuit est autorisée pour un utilisateur habilité.

D'autres particularités et avantages de l'invention apparaîtront mieux à la lecture de la description détaillée qui suit et se réfère à la figure annexée donnée à titre d'exemple non limitatif et qui illustre schématiquement le principe du procédé conforme à l'invention.

Dans la description qui suit, on entend par fournisseur le titulaire d'un logiciel comprenant un ou plusieurs programmes P1,... Pi,... Pn, faisant appel chacun, à un ou plusieurs sous programmes.

En référence à la figure, l'utilisateur doit posséder un calculateur 1 qui comprend au moins une mémoire de programmes 2, un circuit interface 3 qui permet des échanges avec un ou plusieurs périphériques 4 couplés au calculateur, et un circuit de commande et de traitement 5 d'informations stockées dans la mémoire ou provenant des périphériques.

Le circuit de commande et de traitement 5 ainsi que le circuit interface 3 sont classiques en soi et ont des spécificités propres au type de calculateur utilisé.

Le périphérique 4 permet d'adapter le mode de communication du circuit intégré au calculateur. Ce périphérique n'est indispensable que dans la mesure où le client dispose d'un circuit intégré 6 sur lequel sont enregistrés les sous programmes SK1,... SKn, qui comprend un circuit de traitement apte à effectuer directement les échanges avec le calculateur en plus du traitement des informations stockées.

Le circuit intégré de sécurité peut se présenter sous forme d'une carte comprenant une mémoire non volatile 7, programmable sous le contrôle d'un organe de traitement d'informations 8 capable de lancer l'exécution du sous programme sélectionné dans la mémoire 7.

Le fournisseur délivre le logiciel à un utilisateur sous forme d'informations stockées sur un support, du type, disque dur ou bande magnétique.

L'utilisateur procède à une première étape de chargement du logiciel dans la mémoire de programmation de son calculateur.

Conformément au procédé selon l'invention, les programmes P1,... Pi,..., Pn sont chargés dans cette mémoire ainsi que les sous programmes SP1,... SPj,... SPm.

L'utilisateur procède ensuite à des procédures classiques et inhérentes au type de calculateur, pour lancer l'exécution d'un programme du logiciel, par exemple P1.

Ce programme P1 comporte une instruction A d'entrée-sortie qui est envoyée au circuit d'interface 3 lequel est apte à échanger des informations avec le périphérique 4 qui est couplé au calculateur 1.

L'instruction d'entré-sortie A permet de savoir si la carte à mémoire est présente ou non dans le périphérique.

Dans le cas où la carte 6 n'est pas présente dans le périphérique 4 ou dans le cas où le périphérique n'est pas couplé au calculateur 1, l'interface 3 transmet un paramètre NC qui traduit l'état du couplage et qui provoque un déroutement du programme P1 vers le sous programme SP1 enregistré dans la mémoire du calculateur et qui permet de lancer son exécution.

Dans le cas où la carte 6 est présente dans le périphérique 4, le périphérique étant couplé au calculateur, l'interface 3 transmet un paramètre C qui traduit l'état du couplage et qui provoque un déroutement vers le sous programme SK1 enregistré sur la carte. Le paramètre C permet alors au programme P1 en cours, d'envoyer les arguments AD nécessaires à l'exécution du sous programme SK1.

Le circuit de traitement 8 de la carte reçoit ces arguments AD et lance l'exécution du sous programme SK1. Le circuit de traitement 8 transmet les arguments résultats AR au calculateur 1 via le périphérique 4 et le circuit interface 3.

Le programme principal P1 reçoit les arguments AR traités par le circuit 5 qui lui permettent, à partir de l'instruction qui suit l'appel au sous programme, de poursuivre son exécution jusqu'au prochain appel de sous programme.

Au prochain appel de sous programme on procède de la même façon que précédemment pour provoquer l'exécution soit du sous programme enregistré dans la mémoire du calculateur, soit du sous programme enregistré dans la mémoire de la carte.

Le périphérique est classique en soi. Il s'agit d'un dispositif de transfert d'informations. Il est apte à transmettre les arguments de sortie du calculateur vers le circuit de traitement de la carte et inversement à transmettre les arguments de sortie de la carte vers le calculateur.

Le circuit de traitement de la carte est par exemple constitué par un microprocesseur et de circuits annexes connus.

Les sous programmes stockés dans la mémoire de programme du calculateur représentent des versions incomplètes des possibilités de services que peut procurer le programme principal lorsqu'il est dérouté vers les sous programmes stockés sur la carte.

La carte à mémoire utilisée est avantageusement protégée en lecture/écriture une fois délivrée par le fournisseur.

La carte comporte pour cela un code secret stocké dans une zone Z de la mémoire.

Pour pouvoir utiliser la carte, le détenteur habilité devra connaître un code d'accès propre à cette carte et introduire ce code dans le calculateur. Le calculateur va alors transmettre ce code d'accès à la carte, par exemple, avec l'instruction d'entrée-sortie A. Le circuit de traitement 8 de la carte va prendre en compte le code d'accès transmis par le calculateur 1 et le code secret (également appelé code d'habilitation) enregistré pour les comparer. En fonction du résultat de comparaison entre ces deux codes, le circuit 8 va transmettre le paramètre C ou NC pour autoriser ou non l'utilisation des programmes contenu dans la carte.

Une personne non habilitée n'ayant pas connaissance du code d'accès ne pourra donc pas disposer de toutes les fonctionalités du logitiel mais seulement de celles qui sont prévues par la forme simplifiée.

Par ailleurs, le code d'accès d'un détenteur habilité peut être prévu pour donner une habilitation partielle. En effet, le code d'accès peut par exemple n'autoriser le déroutement que pour certains sous programmes contenus dans la mémoire de la carte et interdire le déroutement pour d'autres sous programmes dans la mesure où ces sous programmes correspondent à des options qui n'intéressent pas ce détenteur.

REVENDICATIONS

1. Procédé de stockage d'un logiciel livré par un fournisseur à un utilisateur, caractérisé en ce qu'il consiste à :

- à stocker un logiciel comportant des programmes (P1,... Pi,... Pn), faisant appel à des sous programmes complets (SK1,... SKj,... SKm), dans la mémoire de programmes d'un calculateur ;

- à stocker des sous.programmes (SP1,... SPj,... SPm) dans la mémoire de programmes du calculateur, ces sous programmes correspondant à des versions incomplètes mais fonctionnelles des sous programmes appelés ;

- à stocker dans un circuit intégré extérieur les sous programmes complets (SK1,... SKj,... SKm), ce circuit étant apte à être couplé au calculateur et à stocker dans ce circuit un code d'habilitation destiné à être comparé à un code d'accès fourni par l'utilisateur ;

- seuls les sous programmes incomplets étant exécutés dans le cas où le code d'accès ne correspond pas au code d'habilitation ;

- les sous programmes complets étant susceptibles d'être exécutés par le circuit intégré, dans le cas où le code d'accès correspondant au code d'habilitation.

2. Procédé de stockage selon la revendication 1, caractérisé en ce que le contenu de la mémoire non volatile du circuit intégré extérieur ne peut être transmis de cette mémoire vers les bornes d'accès du circuit et en ce que seule l'exécution des sous programmes stockés dans ce circuit intégré extérieur est autorisée pour un utilisateur habilité ;

3. Dispositif pour la mise en oeuvre du procédé selon la revendication 1 ou 2, caractérisé en ce qu'il comprend :

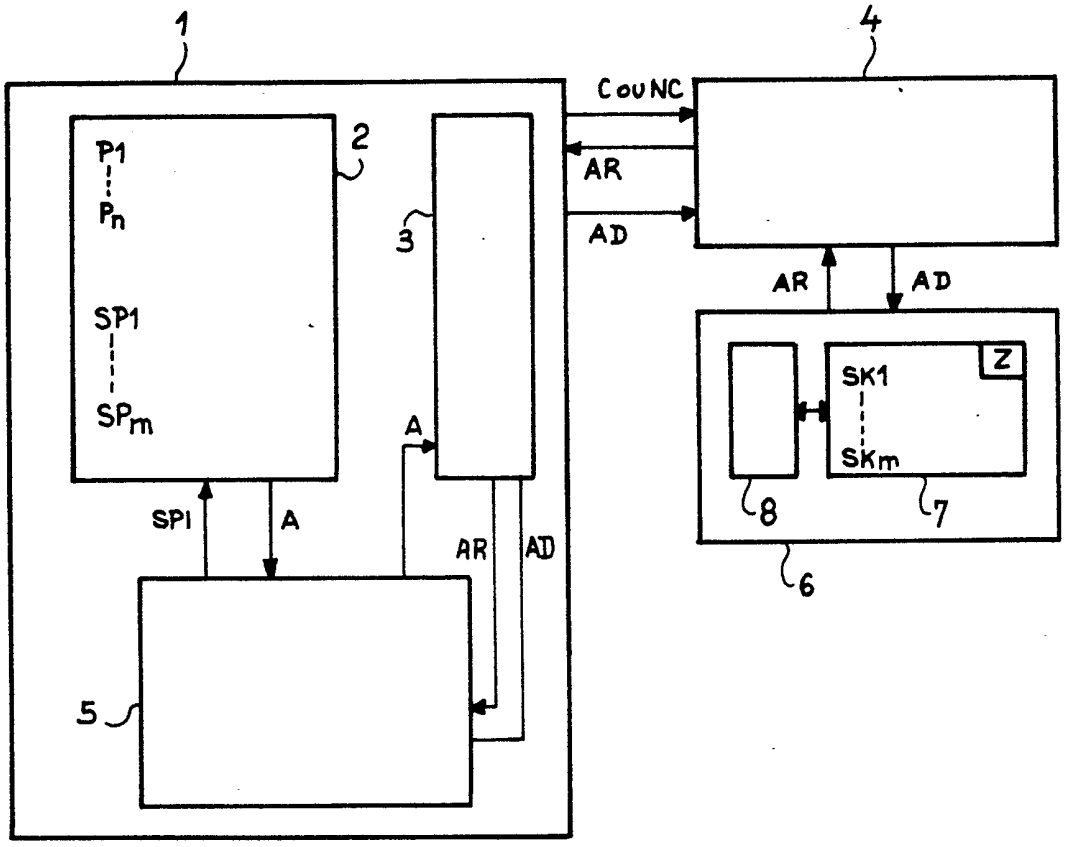
- un circuit intégré (6) comportant une mémoire non volatile (7) programmable, un circuit de traitement (8), le circuit intégré étant susceptible d'être couplé au calculateur ;

- la mémoire (2) du calculateur (1) comportant les programmes (P1,... Pi,... Pn) et les sous programmes (SP1,... SPj,... SPm) ;

- le circuit intégré (6) comportant les sous programmes (SK1,... SKj,... SKm) correspondant à des versions complètes les sous programmes stockés dans la mémoire de programmes et le code d'habilitation destiné à être comparé au code d'accès fourni par l'utilisateur ;

- le calculateur (1) étant apte à envoyer un ordre d'entrée-sortie (A) destiné à savoir si le circuit intégré (6) est couplé ou non au calculateur (1) et ; à lancer l'exécution d'un programme qui fera appel aux sous programmes stockés dans sa mémoire de programmes (2) dans le cas où le circuit intégré (6) n'est pas couplé ou lorque le code d'accès ne correspond pas au code d'habilitation ; ou à lancer l'exécution d'un programme qui fera appel aux sous programmes stockés dans le circuit intégré (6), dans le cas où le circuit intégré est couplé et ou le code d'accès corrrespond au code d'habilitation ; puis à transmettre un ordre qui provoquera l'exécutionde ce sous programme par le circuit intégré (6).

4. Dispositif selon la revendication 3, caractérisé en ce qu'il comporte un organe périphérique (4) susceptible de recevoir le circuit intégré (6), d'être couplé au calculateur (1) et d'adapter le mode communication du circuit intégré (6) au calculateur (1).


1
4
CouNC
AR
AD
2
P1
Pn
SP1
SPm
3
A
SPI
A
AR
AD
5
AR
AD
Z
SK1
SKm
8
7
6