

IL.



Industria y Comercio

SUPERINTENDENCIA

A/PCT

SOLICITUD

PATENTE DE INVENCION

04-3757

21. EXPEDIENTE N°

54. TÍTULO

METODO PARA PROTEGER UN SOFTWARE UTILIZANDO UN PRONCIPIO LLAMADO
DE VARIABLE CONTRA SU USO NO AUTORIZADO

51. CLASIFICACIÓN INTERNACIONAL G06 F 1/00

71. SOLICITANTE VALIDITY

DOMICILIO Romans Sur Isere, Francia

74. APODERADO EMILIO FERRERO
CG 1go 1092

22. BOGOTÁ, D. C. 2004 - I - 20

202040


Industria y Comerc
 SUPERINTENDENCI



04 003757 00

PETITORIO

SUPERINTENDENCIA Y COMERCIO

Radicación : 04003757 CCCCCC00

Folios: 125

Fecha (A/C): 2004-01-23 17:02:57

Trámite : 011 PCT-PATENT 379 FASE NACI 411 PRESENTA

Dependencia: ESSO DIVISION DE NUEVAS CREACIONES

**FORMULARIO UNICO DE SOLICITUD DE PATENTE
 PCT
 ENTRADA EN FASE NACIONAL**

1

SOLICITUD DE:

Patente de Invención.



Patente de Modelo de Utilidad.



Capítulo I.



Capítulo II.

2

**SOLICITANTE
(71)**

Nombre: **VALIDITY**

sociedad constituida y existente bajo las leyes de Francia.

Dirección: Zone Industrielle, 5,
Rue Jean Charcot,
F-26100 Romans Sur Isere,
Francia

Domicilio: Romans Sur Isere, Francia

IDENTIFICACION

C.C.



NIT



C.E.



Otro



Cual _____

Número: _____

3

**REPRESENTANTE
O APODERADO**

Nombre: **EMILIO FERRERO**

Dirección: Carrera 4ª. No. 72-35

Teléfono: 347 38 11

Fax: 217 82 11

E-mail: clientes@camyp.com.

Domicilio: Bogotá, Colombia.

IDENTIFICACION

C.C.



NIT



C.E.



Otro



Cual _____

Número: 438.019 T.P. 31.929

4

**INVENTOR (ES)
(72)**

1- **Jean-Christophe CUENOD**
11, Rue Edgar Degas,
F-78360 Montesson,
Francia.

2-

Gilles SGRO
3, Impasse Jules Ferry,
F-26300 Bourg de Peage,
Francia.

5

PCT (86)Solicitud Internacional No. PCT/FR02/02343Publicación Internacional No. WO 03/012571 A2Fecha: 4 de julio de 2002Fecha : 13 de febrero de 2003

6 Título (54) **PROCEDIMIENTO PARA PROTEGER UN SOFTWARE CON LA AYUDA DE UN PRINCIPIO DE "DETECCIÓN Y COERCION" CONTRA SU USO NO AUTORIZADO**

7 Clasificación Internacional (51) G06F 1/00

8 Prioridad	(33) País de Origen	(32) Fecha	(31) Número de Solicitud
SI ☒ NO ☐	<u>FR</u>	<u>31 de julio de 2001</u>	<u>01/10244</u>

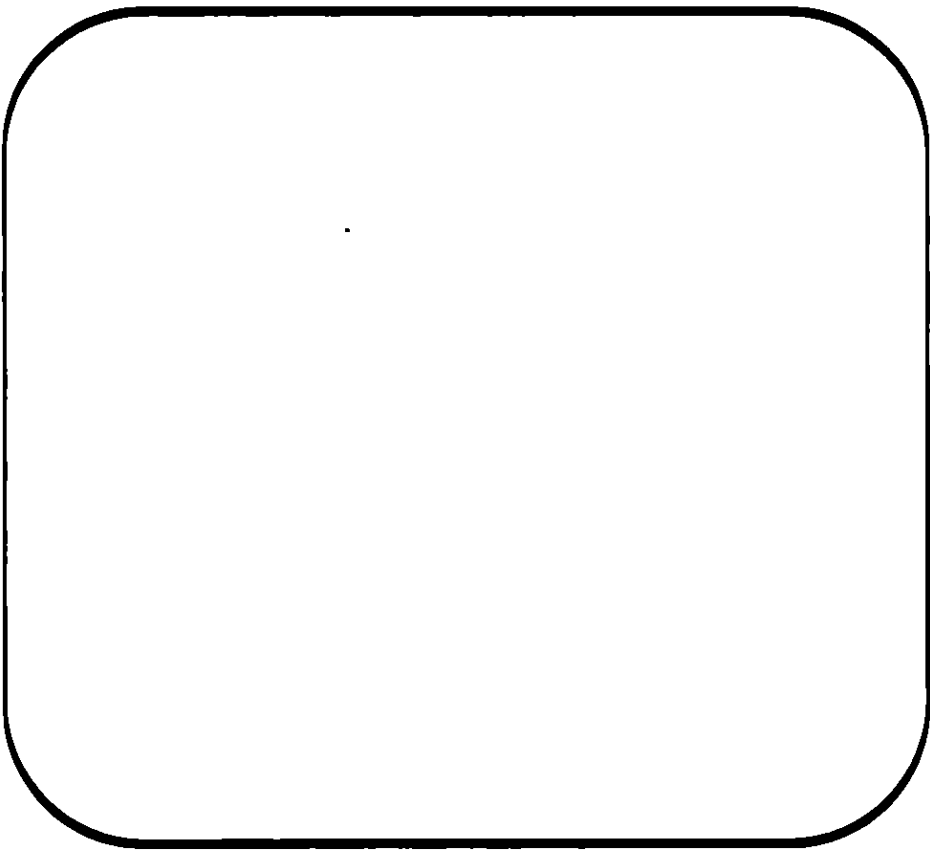
9 Comprobante de pago No. 4-3.590 Fecha: 19 de enero de 2004

10 **Anexos:**

- ☒ Comprobante de pago de la tasa de presentación de la solicitud por \$ 422.000
- ☐ Comprobante de pago por reivindicación de prioridad.
- ☐ Documento que acredita la existencia y representación legal cuando el solicitante sea persona jurídica.
- ☐ Poderes, si fuere el caso.
- ☐ Documento de cesión del inventor al solicitante o a su causante.
- ☐ Declaraciones.
- ☒ Copia de la solicitud internacional. (Resumen, descripción, reivindicaciones, dibujos)
- ☒ Traducción de la solicitud internacional al castellano. (Resumen, descripción, reivindicaciones, dibujos)
- ☐ Copia del examen preliminar efectuado por la IPEA.
- ☐ Traducción del examen preliminar efectuado por la IPEA.
- ☐ De ser el caso, copia del contrato de acceso.
- ☐ De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico.
- ☐ Arte final 12x12
- ☐ De ser el caso, información sobre otras solicitudes de patente o títulos obtenidos en el extranjero por el mismo titular o su causante, relacionadas parcial o totalmente con la invención de esta solicitud.
- ☐ Otros

11

FIGURA CARACTERISTICA:



12

Solicito la concesión de la patente,

NOMBRE: EMILIO FERRERO

FIRMA : 
C.C. 438.019 de Usaquén T.P. 31.929

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
13 février 2003 (13.02.2003)

PCT

(10) Numéro de publication internationale
WO 03/012571 A2

(51) Classification internationale des brevets⁷ : G06F
(21) Numéro de la demande internationale : PCT/FR02/02343
(22) Date de dépôt international : 4 juillet 2002 (04.07.2002)
(25) Langue de dépôt : français
(26) Langue de publication : français
(30) Données relatives à la priorité :
01/10244 31 juillet 2001 (31.07.2001) FR
(71) Déposant (pour tous les États désignés sauf US) : VALIDY
[FR/FR]; Zone Industrielle, 5, rue Jean Charcot, F-26100
Romans sur Isère (FR).

(81) États désignés (national) : AE, AG, AL, AM, AT, AU, AZ,
BA, BB, BG, BR, BY, BZ, CA, CH, CN, CQ, CR, CU, CZ,
DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM,
HR, HU, ID, IL, IN, IS, JP, KI, KG, KP, KR, KZ, LC, LK,
LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX,
MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI,
SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN,
YU, ZA, ZM, ZW.

(84) États désignés (régional) : brevet ARIPO (GH, GM, KI,
LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet
eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet
européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), brevet
OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML,
MR, NR, SN, TD, TG).

(72) Inventeurs; et
(75) Inventeurs/Déposants (pour US seulement) : CUENOD,
Jean-Christophe [FR/TR]; 11, rue Edgar Degas, F-78360
Montesson (FR). SGRO, Gilles [FR/FR]; 3, impasse Jules
Perry, F-26300 Bourg de Peage (FR).
(74) Mandataire : THIBAUT, Jean-Marc; Cabinet Bean de
Loménie, 51, avenue Jean-Jaurès, B.P. 7073, F-69301 Lyon
Cedex 07 (FR).

Publiée :
— sans rapport de recherche internationale, sera republiée
dès réception de ce rapport

En ce qui concerne les codes à deux lettres et autres abrévia-
tions, se référer aux "Notes explicatives relatives aux codes et
abréviations" figurant au début de chaque numéro ordinaire de
la Gazette du PCT.

(54) Title: METHOD FOR PROTECTING A SOFTWARE USING A SO-CALLED DETECTION AND COERCION PRINCIPLE
AGAINST ITS UNAUTHORISED USE

(54) Titre : PROCÉDÉ POUR PROTÉGER UN LOGICIEL A L'AIDE D'UN PRINCIPE DIT DE "DETECTION ET COERCI-
TION" CONTRE SON UTILISATION NON AUTORISÉE

(57) Abstract: The invention concerns a method for protecting, from a unit, a vulnerable software against its unauthorised use, said vulnerable software operating on a data processing system. The method consists in defining: at least one executing characteristic of the software, capable of being monitored at least partly in a unit; at least one criterion to be observed for at least one executing characteristic of the software; detection means to be implemented in a unit and enabling to detect whether at least one software executing characteristic does not observe at least one associated criterion; and coercion means to be implemented in a unit for informing the data processing system and/or modifying the execution of a software when at least one criterion is not observed.

(57) Abrégé : L'invention concerne un procédé pour protéger, à partir d'une unité, un logiciel vulnérable contre son utilisation non autorisée, ledit logiciel vulnérable fonctionnant sur un système de traitement de données. Le procédé consiste à définir : - au moins une caractéristique d'exécution de logiciel, susceptible d'être surveillée au moins en partie dans une unité, - au moins un critère à respecter pour au moins une caractéristique d'exécution de logiciel, - des moyens de détection à mettre en œuvre dans une unité et permettant de détecter qu'au moins une caractéristique d'exécution de logiciel ne respecte pas au moins un critère associé, - et des moyens de coercition à mettre en œuvre dans une unité et permettant d'informer le système de traitement de données et/ou de modifier l'exécution d'un logiciel lorsqu'au moins un critère n'est pas respecté.

WO 03/012571 A2

1/5

REQUETE PCT

71331-C7-JMT

Original (pour PRESENTATION) - Imprimé le 04.07.2002 12:37:02 PM

0	Réservé à l'Office récepteur	
0-1	Demande internationale No.	PCT/FR 02/02343
0-2	Date du dépôt international	
0-3	Nom de l'office récepteur et "Demande internationale PCT"	
0-4	Formulaire - PCT/ROHM1 Requête PCT	
0-4-1	Préparé avec	PCT-EASY Version 2.92 (mis à jour 01.06.2002)
0-5	Pédiction Le soussigné requiert que la présente demande internationale soit traitée conformément au Traité de coopération en matière de brevets	
0-6	Office récepteur (choix par le déposant)	Institut national de la propriété industrielle (France) (RO/FR)
0-7	Référence du dossier du déposant ou du mandataire	71331-C7-JMT
I	Titre de l'invention	PROCEDURE POUR PROTEGER UN LOGICIEL A L'AIDE D'UN PRINCIPE DIT DE "DETECTION ET CORRECTION" CONTRE SON UTILISATION NON AUTORISEE
II	Déposant	
II-1	Cette personne est :	Déposant seulement.
II-2	Déposant pour	Tous les Etats désignés sauf US
II-4	Nom	VALIDY
II-5	Adresse:	Zone Industrielle 5, Rue Jean Charcot F-26100 ROMANS SUR ISERE France
II-6	Nationalité (nom de l'Etat)	FR
II-7	Résidence (nom de l'Etat)	FR
III-1	Déposant et/ou Inventeur	
III-1-1	Cette personne est :	Déposant et inventeur
III-1-2	Déposant pour	US seulement
III-1-4	Nom (NOM DE FAMILLE, Prénom)	CUENOD, Jean-Christophe
III-1-5	Adresse:	11, Rue Edgar Degas F-78360 MONTESSON France
III-1-6	Nationalité (nom de l'Etat)	FR
III-1-7	Résidence (nom de l'Etat)	FR

2/5

REQUETE PCT

71331-C7-JMT

Original (pourPRESENTATION) - Imprimé le 04.07.2002 12:37:02 PM

II-2	Déposant et/ou inventeur	Déposant et inventeur
III-2-1	Cette personne est :	US seulement
III-2-2	Déposant pour	SGRO, Gilles
III-2-4	Nom (NOM DE FAMILLE, Prénom)	3, Impasse Jules Ferry
III-2-5	Adresse:	F-26300 BOURG DE PEAGE
		France
III-2-6	Nationalité (nom de l'Etat)	FR
III-2-7	Résidence (nom de l'Etat)	FR
IV-1	Mandataire ; Représentant commun ou adresse pour la correspondance. La personne nommée ci-dessous est/à été désignée pour agir au nom du ou des déposants auprès des autorités internationales compétentes, comme Nom (NOM DE FAMILLE, Prénom)	mandataire
IV-1-1	Nom (NOM DE FAMILLE, Prénom)	THIBAUT, Jean-Marc
IV-1-2	Adresse:	Cabinet Beau de Loménie 51, Avenue Jean-Jaurès B.P. 7073 F-69301 LYON CEDEX 07 France
IV-1-3	No. de téléphone	04 72 76 85 30
IV-1-4	No de télécopieur:	04 78 69 86 82
IV-1-5	Courrier électronique:	contact@cabinetbeaudelomenie.fr
V	Désignation d'Etats	
V-1	Brevet régional (d'autres formes de protection ou de traitement, le cas échéant, sont spécifiées entre parenthèses pour les Etats désignés concernés)	AP: GR GM KE LS MW NZ SD SL SZ TE UG ZM ZW et tout autre Etat qui est un Etat contractant du Protocole de Harare et du PCT EA: AM AZ BY KG KZ MD RU TJ TM et tout autre Etat qui est un Etat contractant de la Convention sur le brevet eurasién et du PCT EP: AT BE BG CH LI CY CZ DE DK EE ES FI FR GB GR IE IT LU MC NL PT SE SK TR et tout autre Etat qui est un Etat contractant de la Convention sur le brevet européen et du PCT OA: BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG et tout autre Etat qui est un Etat membre de l'OAPI et un Etat contractant du PCT

3/5

REQUETE PCT

71331-C7-JMT

Original (pour PRESENTATION) - Imprimé le 04.07.2002 12:37:02 PM

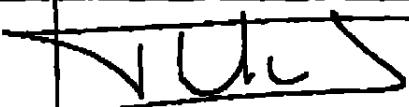
V-2	Brevet national (d'autres formes de protection ou de traitement, le cas échéant, sont spécifiées entre parenthèses pour les Etats désignés concernés)	AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH LI CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HN HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX ME NO NZ OM PH PL PT RO RU SD SE SG SI SK SL TJ TM TN TR TT TZ UA UG US UZ VN YU ZA ZM ZW
V-5	Déclaration concernant les désignations de préséance Outre les désignations faites sous les rubriques V-1, V-2 et V-3, le déposant fait aussi, conformément à la règle 4.8.b), toutes les désignations qui seraient autorisées en vertu du PCT, à l'exception de toute désignation(s) indiquée(s) dans la rubrique V-6 ci-dessous. Le déposant déclare que ces désignations additionnelles sont faites sous réserve de confirmation et que toute désignation qui n'est pas confirmée avant l'expiration d'un délai de 18 mois à compter de la date de priorité sera considérée comme retirée par le déposant à l'expiration de ce délai.	
V-6	Exclusion(s) des désignations de préséance	NEANT
VI-1	Réclamation de priorité d'une demande nationale antérieure	
VI-1-1	Date de dépôt	31 juillet 2001 (31.07.2001)
VI-1-2	Numéro	01 10 244
VI-1-3	Pays	FR
VI-2	Requête pour le document de priorité L'office récepteur est prié de préparer et de transmettre au Bureau International une copie certifiée conforme de la ou des demandes antérieures mentionnées ci-dessus sous la/les rubrique(s):	VI-1
VI-1	Administration chargée de la recherche internationale choisie	Office européen des brevets (OEB) (ISA/EP)
VII-2	Demande d'utilisation des résultats d'une recherche antérieure; mention de cette recherche	
VII-2-1	Date	23 avril 2002 (23.04.2002)
VII-2-2	Numéro	PA 606021
VII-2-3	Pays (ou office régional)	EP

4/5

REQUÊTE PCT

71331-C7-JMT

Original (pour PRÉSENTATION) - Imprimé le 04.07.2002 12:37:02 PM

VII	Déclarations	Nombre de déclarations	
VII-1	Déclaration relative à l'identité de l'inventeur	-	
VII-2	Déclaration relative au droit du déposant, à la date du dépôt international, de demander et d'obtenir un brevet	-	
VII-3	Déclaration relative au droit du déposant, à la date du dépôt international, de revendiquer la priorité de la demande antérieure	-	
VII-4	Déclaration relative à la qualité d'inventeur (seulement aux fins de la désignation des États-Unis d'Amérique)	-	
VII-5	Déclaration relative à des divulgations non opposables ou à des exceptions au défaut de nouveauté	-	
IX	Bordereau	Nombre de feuilles	Document(s) électronique(s) joint(s)
IX-1	Requête (y compris les feuilles pour déclaration)	5	-
IX-2	Description	65	-
IX-3	Revendications	18	-
IX-4	Abrégé	1	WEABST00.TXT
IX-5	Dessins	11	-
IX-7	TOTAL	100	
	Éléments joints	Document(s) papier joint(s)	Document(s) électronique(s) joint(s)
IX-8	Feuille de calcul des taxes	✓	-
IX-17	Disquette PCT-EASY	-	Disquette
IX-18	Figure des dessins qui doit accompagner l'abrégé		
IX-20	Langue de dépôt de la demande internationale	français	
X-1	Signature du déposant, du mandataire ou du représentant commun		
X-1-1	Nom (NOM DE FAMILLE, Prénom)	TRIBAULT, Jean-Marc	

RESERVE A L'OFFICE RECEPTEUR

10-1	Date effective de réception des pièces supposées constituer la demande internationale	4 JUL. 2002
10-2	Dessins:	
10-2-1	Reçus	
10-2-2	non reçus	
10-3	Date effective de réception, rectifiée en raison de la réception ultérieure, mais dans les délais, de documents ou de dessins complétant ce qui est supposé constituer la demande internationale:	
10-4	Date de réception, dans les délais, des corrections demandées selon l'article 11.2) du PCT	

5/5

REQUETE PCT

T1331-C7-JMT

Original (pourPRESENTATION) - imprimé le 04.07.2002 12:37:02 PM

10-5	Administration chargée de la recherche internationale	ISA/EP
10-8	Transmission de la copie de recherche différée jusqu'au paiement de la taxe de recherche	

RESERVE AU BUREAU INTERNATIONAL

11-1	Date de réception de l'exemplaire original par le Bureau International	
------	--	--

PROCEDIMIENTO PARA PROTEGER UN SOFTWARE CON LA AYUDA DE
UN PRINCIPIO DE "DETECCION Y COERCION" CONTRA SU USO NO
AUTORIZADO

5

MEMORIA DESCRIPTIVA

La presente invención se refiere al campo técnico de los sistemas de procesamiento de datos en el sentido general y contempla, de manera más precisa, los medios para proteger, contra su uso no autorizado, un software que funciona sobre dichos sistemas de procesamiento de datos.

El objeto de la invención contempla, de manera más particular, los medios para proteger un software contra su uso no autorizado, a partir de una unidad de procesamiento y de memoria, accesando a dicha unidad por lo general mediante una tarjeta virgen o mediante una llave conectable a un puerto USB.

En el campo técnico anterior, el principal inconveniente se refiere al empleo no autorizado de software por parte de usuarios que no han adquirido los derechos de licencia. Este uso ilícito de software causa un daño declarado por los editores de software, los distribuidores de software y/o cualquier persona que integre dicho software en productos. Para evitar dichas copias ilícitas, se ha propuesto en el estado de la técnica, diversas soluciones para proteger softwares.

De esta forma, se conoce una solución de protección que consiste en aplicar un sistema material de protección, tal como un elemento físico llamado llave de protección o "dongle" en terminología anglosajona. Dicha llave de protección debe garantizar la ejecución del software únicamente en presencia de la llave. Ahora bien, se debe constatar que dicha solución no es eficaz porque presenta el inconveniente de ser fácilmente

eludida. Una persona mal intencionada o pirata, con la ayuda de herramientas especializadas, tales como desarmadores, puede suprimir las instrucciones de control de la llave de protección. Podría ser entonces posible realizar copias ilícitas que corresponden a las versiones modificadas de software que ya no tengan protección alguna. Además, esta solución no puede ser generalizada a todos los softwares, en la medida en que es difícil conectar más de dos llaves de protección sobre un mismo sistema.

El objeto de la invención contempla justamente remediar los inconvenientes antes mencionados proponiendo un método para proteger un software contra su uso no autorizado, a partir de una unidad de procesamiento y memoria similar, en la medida en la que la presencia de dicha unidad es necesaria para que el software pueda ser completamente funcional.

Para lograr dicho objetivo, el propósito de la invención involucra un procedimiento para proteger, a partir de al menos una unidad virgen que comprende al menos medios de procesamiento y medios de memoria, un software vulnerable contra su utilización no autorizada, funcionando dicho software vulnerable en un sistema de procesamiento de datos. El procedimiento de la invención consiste:

- 20 > En una fase de protección:
 - Definir:
 - al menos una característica de ejecución del software susceptible de ser monitoreada al menos en parte en una unidad;
 - al menos un criterio a acatar por al menos una característica
- 25 de ejecución de software;
 - medios de detección a ser ejecutados en una unidad y que permiten detectar que al menos una característica de ejecución del software no acata al menos un criterio relacionado; y

- medios de coerción a ser ejecutados por una unidad y que permiten informar al sistema de procesamiento de datos y/o modificar la ejecución de un software cuando no se acata por lo menos un criterio;
 - construir medios de aprovechamiento que permitan
- 5 transformar la unidad virgen en una unidad capaz de poner en marcha los medios de detección y los medios de coerción;
- crear un software protegido:
 - al elegir por lo menos una característica de ejecución de software para ser monitoreado, entre las características de ejecución
- 10 susceptibles de ser monitoreadas,
- elegir por lo menos un criterio a acatar por al menos una característica de ejecución de software que se haya elegido,
 - elegir al menos un procesamiento algorítmico, el cual, cuando se ejecuta un software vulnerable, utiliza al menos un operando y permite
- 15 obtener al menos un resultado, y por el cual se monitorea al menos una característica de ejecución de software elegido,
- elegir por lo menos una porción de una fuente de software vulnerable que contiene al menos un procesamiento algorítmico elegido,
 - producir la fuente de software protegido a partir de la fuente de
- 20 software vulnerable, que modifica al menos una porción elegida de la fuente de software vulnerable para obtener al menos una porción modificada de una fuente de software protegido; dicha modificación es tal que:
- cuando se ejecuta el software protegido se ejecuta una primera parte de ejecución en el sistema de procesamiento de datos y una
- 25 segunda parte de ejecución se ejecuta en una unidad, que se obtiene a partir de la unidad virgen después de cargar informaciones,
- la segunda parte de ejecución ejecuta al menos la funcionalidad de por lo menos un procesamiento algorítmico elegido,

➤ y cuando se ejecuta el software protegido, por lo menos una característica de ejecución elegida es monitoreada por la segunda parte de ejecución y no se acata un criterio que termina en una modificación de la ejecución del software protegido;

5 - y producir:

➤ una primera parte objeto del software protegido, a partir de la fuente del software protegido, dicha primera parte objeto es tal que cuando se ejecuta el software protegido aparece una primera parte de ejecución que se ejecuta en el sistema de procesamiento de datos y en el
10 cual al menos una porción detecta que al menos se monitorea una característica de ejecución de software,

➤ y una segunda parte objeto del software protegido, que contiene los medios de aprovechamiento que ponen en marcha los medios de detección y los medios de coerción, esta segunda parte objeto es tal que,
15 después de ser cargada en la unidad virgen y cuando se ejecuta el software protegido aparece la segunda parte de ejecución en medio de la cual al menos se monitorea una característica de ejecución de software, y en medio de la cual el no acatamiento de un criterio termina en una modificación de la ejecución del software protegido,

20 • y cargar la segunda parte objeto en la unidad virgen en virtud de obtener la unidad;

➤ y en una fase de utilización en el curso de la misma se ejecuta el software protegido:

• en presencia de la unidad :

25 - siempre que se acaten todos los criterios que corresponden a todas las características de ejecución monitoreadas de todas las porciones modificadas del software protegido, se permite el funcionamiento nominal de

estas porciones del software protegido y en consecuencia se permite el funcionamiento nominal del software protegido,

- y si al menos no se acata uno de los criterios que corresponde a una característica de ejecución monitoreada de una porción del software protegido, informar al sistema de procesamiento de datos y/o modificar el funcionamiento de la porción de software protegido, de manera que se modifica el funcionamiento del software protegido;

- y en ausencia de la unidad a pesar de la demanda de una porción de la primera parte de ejecución (2pes) de iniciar la ejecución en la unidad (6) de la funcionalidad de un procesamiento algorítmico elegido, no poder responder correctamente a esa demanda, de manera que al menos esta porción no se ejecuta correctamente y que, en consecuencia, el software protegido (2p) no sea completamente funcional.

De acuerdo con otra modalidad preferida, el método de acuerdo con la invención consiste:

en la fase de protección:

en definir:

un juego de funciones elementales en donde las funciones elementales son susceptibles a ser ejecutadas en la unidad que comprende también los medios de procesamiento,

y un juego de comandos elementales para este juego de funciones elementales, los comandos elementales son susceptibles de ser ejecutados en el sistema de procesamiento de datos y de iniciar la ejecución en la unidad, de funciones elementales,

en construir medios de aprovechamiento que permitan transformar la unidad virgen en la unidad capaz de ejecutar las funciones elementales de dicho juego, la ejecución de estas funciones elementales son iniciadas por la ejecución en el sistema de procesamiento de datos, de

comandos elementales,

en modificar el software protegido:

seleccionando en la fuente del software protegido:

por lo menos un procesamiento algorítmico que durante la
5 ejecución del software protegido, utilice por lo menos una variable
seleccionada, y permita obtener por lo menos una variable resultante,

y por lo menos una porción que contiene al menos un
procesamiento algorítmico seleccionado,

modificando por lo menos una porción seleccionada de la fuente
10 del software protegido, esta modificación siendo tal que:

durante la ejecución del software protegido la primera parte de
ejecución es ejecutada en el sistema de procesamiento de datos y una
segunda parte de ejecución es ejecutada en la unidad,

por lo menos la funcionalidad de al menos un procesamiento
15 algorítmico seleccionado es ejecutada por medio de la segunda parte de
ejecución,

por lo menos un procesamiento algorítmico seleccionado se
fragmenta de manera que durante la ejecución del software protegido, este
procesamiento algorítmico es ejecutado por medio de la segunda parte de
20 ejecución, utilizando las funciones elementales,

para al menos un procesamiento algorítmico seleccionado, se
integran comandos elementales en la fuente del software protegido, de
manera que durante la ejecución del software protegido, cada comando
elemental es ejecutado por la primera parte de ejecución y inicia en la unidad,
25 la ejecución por medio de la segunda parte de ejecución, de una función
elemental,

y se selecciona un ordenamiento de comandos elementales
entre el conjunto de ordenamientos que permiten la ejecución del software

protegido,

y produciendo:

la primera parte objeto del software protegido, esta primera parte
objeto siendo tal que durante la ejecución del software protegido, los
5 comandos elementales son ejecutados de acuerdo con el ordenamiento
seleccionado,

y una segunda parte objeto del software protegido que contiene
los medios de aprovechamiento, esta segunda parte objeto siendo tal que
después de la carga en la unidad virgen y durante la ejecución del software
10 protegido, aparece la segunda parte de ejecución por medio de la cual se
ejecutan las funciones elementales iniciadas por la primera parte de
ejecución,

y en cargar la segunda parte objeto de la unidad virgen, con la
idea de obtener la unidad,

15 y en la fase de uso:

en presencia de la unidad cada vez que lo impone un comando
elemental contenido en una porción de la primera parte de ejecución, ejecutar
la función elemental correspondiente en la unidad, de modo que esta porción
es ejecuta correctamente y que en consecuencia, el software protegido es
20 completamente funcional,

y en ausencia de la unidad, a pesar de la petición de una
porción de la primera parte de ejecución, iniciar la ejecución de una función
elemental en la unidad, de no poder responder correctamente a esta
demanda, de modo que por lo menos esta porción no es ejecuta
25 correctamente y que en consecuencia, el software protegido no es
completamente funcional.

De acuerdo con otra modalidad preferida, el método de acuerdo
con la invención consiste:

en la fase de protección:

en definir:

un juego de funciones elementales en donde las funciones elementales son susceptibles de ser ejecutadas en la unidad,

5 y un juego de comandos elementales para este juego de funciones elementales, estos comandos elementales son susceptibles de ser ejecutados en el sistema de procesamiento de datos y de iniciar la ejecución en la unidad, de funciones elementales,

en construir medios de aprovechamiento que permitan a la
10 unidad, ejecutar las funciones elementales de dicho juego, la ejecución de estas funciones elementales es iniciada por la ejecución en el sistema de procesamiento de datos, de comandos elementales,

y en modificar el software protegido:

seleccionando en la fuente del software protegido, por lo menos
15 una etapa que durante la ejecución del software protegido, realice la funcionalidad de un procesamiento algorítmico,

modificando por lo menos una porción seleccionada de la fuente de software protegido, esta modificación siendo tal que:

por lo menos una etapa seleccionada sea fragmentada de
20 manera que durante la ejecución del software protegido, esta etapa sea ejecutada por medio de la segunda parte de ejecución, utilizando las funciones elementales,

para al menos una etapa seleccionada, los comandos elementales son integrados en la fuente del software protegido, de manera
25 que durante la ejecución del software protegido, cada comando elemental sea ejecutado por la primera parte de ejecución y inicie en la unidad, la ejecución por medio de la segunda parte de ejecución, de una función elemental,

y se selecciona un ordenamiento de comandos elementales

entre el conjunto de ordenamientos que permiten la ejecución del software protegido,

y produciendo:

la primera parte objeto del software protegido, esta primera parte
5 objeto siendo tal que durante la ejecución del software protegido, los comandos elementales son ejecutados de acuerdo con el ordenamiento seleccionado,

y la segunda parte objeto del software protegido que contiene también los medios de aprovechamiento, esta segunda parte objeto siendo tal
10 que después de cargar en la unidad y durante la ejecución del software protegido, aparece la segunda parte de ejecución por medio de la cual se ejecutan las funciones elementales iniciadas por la primera parte de ejecución,

y en la fase de uso:

15 en presencia de la unidad y cada vez que lo impone un comando elemental contenido en una porción de la primera parte de ejecución, ejecutar la función elemental que corresponde en la unidad, de modo que esta porción se ejecuta correctamente y que en consecuencia, el software protegido es completamente funcional,

20 y en ausencia de la unidad, a pesar de la petición de una porción de la primera parte de ejecución, iniciar la ejecución de una función elemental en la unidad, en no poder responder correctamente a esta petición, de modo que por lo menos esta porción no sea ejecutada correctamente y que en consecuencia, el software protegido no sea completamente funcional.

25 De acuerdo con otra modalidad preferida, el método de acuerdo con la invención consiste:

en la fase de protección:

en definir:

por lo menos una característica de ejecución de software, susceptible de ser vigilada por lo menos en parte en la unidad,

por lo menos un criterio a obedecer para al menos una característica de ejecución de software,

5 medios de detección a aplicar en la unidad y que permiten detectar que por lo menos una característica de ejecución de software obedece por lo menos un criterio asociado,

 y medios de coerción para aplicar en la unidad y que permiten informar al sistema de procesamiento de datos y/o de modificar la ejecución
10 de un software, cuando por lo menos un criterio no es obedecido,

 en construir los medios de aprovechamiento que permitan a la unidad, aplicar también los medios de detección y los medios de coerción,

 y en modificar el software protegido:

 seleccionando por lo menos una característica de ejecución de
15 software a vigilar, entre las características de ejecución susceptibles de ser vigilada,

 seleccionando por lo menos un criterio a obedecer para al menos una característica de ejecución de software seleccionado,

 seleccionando en la fuente del software protegido, funciones
20 elementales para las cuales al menos se vigilará una característica de ejecución de software seleccionado,

 modificando la menos una porción seleccionada de la fuente del software protegido, esta modificación siendo tal que durante la ejecución del software protegido, se vigila por lo menos una característica de ejecución
25 seleccionada por medio de la segunda parte de ejecución, y el no obedecer un criterio conduce a una información del sistema de procesamiento de datos y/o a una modificación de la ejecución del software protegido,

 y produciendo la segunda parte objeto del software protegido

que contiene los medios de aprovechamiento aplicando también los medios de detección y los medios de coerción, esta segunda parte objeto siendo tal que, después de carga en la unidad y durante la ejecución del software protegido, se vigila por lo menos una característica de ejecución de software y

5 el no obedecer un criterio conduce a una información del sistema de procesamiento de datos y/o a una modificación de la ejecución del software protegido,

y en la fase de uso:

en presencia de la unidad:

10 mientras se obedezcan todos los criterios correspondientes a todas las características de ejecución vigiladas de todas las porciones modificadas del software protegido, permitir el funcionamiento nominal de estas porciones del software protegido y en consecuencia, permitir el funcionamiento nominal del software protegido,

15 y si por lo menos uno de los criterios correspondientes a una característica de ejecución vigilada de una porción del software protegido no es obedecido, informar al sistema de procesamiento de datos y/o modificar el funcionamiento de la porción del software protegido, de modo que el funcionamiento del software protegido sea modificado.

20 De acuerdo con una variante de modalidad, el método de acuerdo con la invención consiste:

en la fase de protección:

en definir:

25 como característica de ejecución de software susceptible de ser vigilada, una variable de medición del uso de una funcionalidad de un software,

como criterio a obedecer, por lo menos un umbral asociado con cada variable de medición,

y medios de actualización que permiten actualizar por lo menos una variable de medición,

en construir los medios de aprovechamiento que permiten a la unidad aplicar también los medios de actualización,

5 y en modificar el software protegido:

seleccionando como característica de ejecución de software a vigilar, por lo menos una variable de medición del uso de una funcionalidad de un software,

seleccionando:

10 por lo menos una funcionalidad del software protegido en donde el uso es susceptible de ser vigilado gracias a una variable de medición,

por lo menos una variable de medición que sirve para cuantificar el uso de dicha funcionalidad,

15 por lo menos un umbral asociado con una variable de medición seleccionado que corresponde a un límite de uso de dicha funcionalidad,

y por lo menos un método para actualizar una variable de medición seleccionada en función del uso de dicha funcionalidad,

y modificando por lo menos una porción seleccionada de la fuente del software protegido, esta modificación siendo tal que, durante la ejecución del software protegido, la variable de medición es actualizada por medio de la segunda parte de ejecución, en función del uso de dicha funcionalidad y por lo menos se toma en cuenta un exceso de umbral,

20 y en la fase de uso, en presencia de la unidad, y en el caso en donde se detecte por lo menos un exceso de umbral correspondiente a por lo menos un límite de uso, en informar al sistema de procesamiento de datos y/o modificar el funcionamiento de la porción del software protegido, de modo que el funcionamiento de software protegido sea modificado.

De acuerdo con una variante de modalidad, el método de

acuerdo con la invención consiste:

en la fase de protección:

en definir:

para al menos una variable de medición, varios umbrales

5 asociados,

y medios de coerción diferentes que correspondan a cada uno

de estos umbrales,

y en modificar el software protegido:

seleccionando en la fuente del software protegido, por lo menos

10 una variable de medición seleccionada a la cual deben estar asociados varios umbrales correspondientes a los límites diferentes de uso de la funcionalidad,

seleccionando por lo menos dos umbrales asociados con la

variable de medición seleccionada,

y modificando por lo menos una porción seleccionada de la

15 fuente de software protegido, esta modificación siendo tal que, durante la ejecución del software protegido, se tomen en cuenta los excesos de diversos umbrales, por medio de la segunda parte de ejecución, de manera diferente,

y en la fase de uso:

en presencia de la unidad:

20 en el caso en donde se detecta el exceso de un primer umbral, en ordenar al software protegido de no utilizar más la funcionalidad correspondiente,

y en el caso en donde se detecte el exceso de un segundo

umbral, en hacer inoperante la funcionalidad correspondiente y/o por lo

25 menos una porción del software protegido.

De acuerdo con una variante de modalidad, el método de acuerdo con la invención consiste:

en la fase de protección:

en definir medios de recarga que permitan acreditar por lo menos un uso complementario para al menos una funcionalidad de software vigilada por una variable de medición,

en construir los medios de aprovechamiento que permitan
5 también a la unidad aplicar los medios de recarga,

y en modificar el software protegido:

seleccionando la fuente de software protegido, por lo menos una variable de medición seleccionada que permita limitar al uso de una funcionalidad a la cual por lo menos un uso complementario debe poder ser
10 acreditado,

y modificando al menos una porción seleccionada, esta modificación siendo tal que en una fase llamada de recarga, por lo menos se puede acreditar un uso complementario de por lo menos una funcionalidad que corresponde a una variable de medición seleccionada,

15 y en la fase de recarga:

en reactualizar por lo menos una variable de medición seleccionada y/o al menos un umbral asociado, a manera de permitir por lo menos un uso complementario de la funcionalidad.

De acuerdo con una variante de modalidad, el método de
20 acuerdo con la invención consiste:

en la fase de protección:

en definir:

como característica de ejecución de software susceptible de ser vigilada, un perfil de uso de software,

25 y como criterio a obedecer, por lo menos una característica de ejecución de software,

y en modificar el software protegido:

seleccionando como característica de ejecución de software a

vigilar por lo menos un perfil de uso de software,

seleccionando por lo menos una característica de ejecución que debe obedecer por lo menos un perfil de uso seleccionado,

5 y modificando al menos una porción seleccionada de la fuente de software protegido, esta modificación siendo tal que durante la ejecución del software protegido, la segunda parte de ejecución obedece todas las características de ejecución seleccionadas,

y en la fase de uso en presencia de la unidad, y en el caso en donde se detecta que por lo menos una característica de ejecución no es
10 obedecida, en informar al sistema de procesamiento de datos y/o en modificar el funcionamiento de la porción del software protegido, de modo que el funcionamiento del software protegido sea modificado.

De acuerdo con una variante de modalidad, el método de acuerdo con la invención consiste:

15 en la fase de protección:

en definir:

un juego de instrucciones en donde las instrucciones son susceptibles de ser ejecutadas en la unidad,

un juego de comandos de instrucciones para este juego de
20 instrucciones, estos comandos de instrucciones son susceptibles de ser ejecutados en el sistema de procesamiento de datos y de iniciar en la unidad la ejecución de las instrucciones,

como perfil de uso, el encadenamiento de las instrucciones,

como característica de ejecución, un encadenamiento deseado
25 para la ejecución de las instrucciones,

como medios de detección, medios que permitan detectar que el encadenamiento de las instrucciones no corresponde a aquel deseado,

y como medios de coerción, medios que permitan informar al

sistema de procesamiento de datos y/o de modificar el funcionamiento de la porción de software protegido cuando el encadenamiento de instrucciones no corresponda a aquel deseado,

5 en construir los medios de aprovechamiento que permitan también a la unidad ejecutar las instrucciones del juego de instrucciones, la ejecución de estas instrucciones son iniciadas por la ejecución en el sistema de procesamiento de datos, de comandos de instrucciones,

y en modificar el software protegido:

- la modificación al menos de una porción elegida de la fuente
10 del software protegido, siendo dicha modificación tal que:

> al menos un procesamiento algorítmico elegido se desglosa de manera que cuando se ejecuta el software protegido, se ejecuta este procesamiento algorítmico en la segunda parte de ejecución al utilizar instrucciones,

15 > para al menos un proceso algorítmico elegido, se integran los comandos de instrucciones en la fuente de software protegido de manera que cuando se ejecuta el software protegido se ejecuta cada comando de instrucciones en la primera parte de ejecución e inicia en la unidad la ejecución de al menos la segunda parte de ejecución de una instrucción,

20 > se elige una orden de comandos de instrucciones entre el conjunto de órdenes que permiten la ejecución del software protegido,

> y el enlace de que deben acatar por lo menos ciertas instrucciones cuando se especifica su ejecución en la unidad,

> y en la fase de utilización , en presencia de la unidad, en
25 el caso que se detecte que el enlace de las instrucciones ejecutadas en la unidad no corresponden a aquel que se desea, informar al sistema de procesamiento de datos y/o modificar el funcionamiento de la porción de

software protegido, de manera que se modifique el funcionamiento del software protegido.

De acuerdo con una variante de modalidad, el método de acuerdo con la invención consiste:

5 en la fase de protección:

 en definir:

 como juego de instrucciones, un juego de instrucciones en donde por lo menos ciertas instrucciones trabajan sobre registros y utilizan al menos un operando con el fin de arrojar un resultado,

10 por lo menos una parte de las instrucciones que trabajan sobre los registros:

 una parte que define a la funcionalidad de la instrucción,

 y una parte que define el encadenamiento deseado para la ejecución de instrucciones y que comprende campos de bits que

15 corresponden a:

 un campo de identificación de la instrucción, y para cada operando de la instrucción:

 un campo bandera,

 y un campo de identificación prevista del operando,

20 para cada registro que pertenece a los medios de aprovechamiento y utilizado por el juego de instrucciones, un campo de identificación generada en la cual se memoriza automáticamente la identificación de la última instrucción que haya arrojado su resultado en este registro,

25 como medios de detección, medios que permiten, durante la ejecución de una instrucción, para cada operando, cuando lo impone el campo bandera, controlar la igualdad entre el campo de identificación generada que corresponde al registro utilizado por este operando, y el campo

de identificación prevista del origen de este operando,

y como medios de coerción, medios que permiten modificar el resultado de las instrucciones, si por lo menos una de las igualdades controladas es falsa.

5 En una modalidad preferida, el procedimiento para la invención consiste:

➤ en la fase de protección:

• modificar el software protegido:

- al elegir al menos una variable que se utilice en al menos un

10 proceso algorítmico elegido, que cuando ejecuta el software protegido define parcialmente el estado de software protegido,

- al modificar al menos una porción elegida de la fuente del software protegido siendo esta modificación tal que cuando se ejecuta el software protegido, al menos una variable elegida o al menos una copia de la

15 variable elegida reside dentro de la unidad,

- y que produce:

➤ la primera parte objeto del software protegido, esta primera parte objeto es tal que cuando se ejecuta el software protegido, al menos una porción de la primera parte de ejecución se hace cargo también al
20 menos de una variable o al menos de una copia de variable que reside en la unidad,

➤ y la segunda parte objeto del software protegido, siendo segunda parte objeto tal que, después de cargar lo que la unidad y cuando se ejecuta el software protegido, aparece la segunda parte de ejecución, en
25 medio del cual al menos una variable elegida o al menos una copia de variable elegida reside también dentro de la unidad,

➤ y en la fase de utilización:

- en presencia de la unidad cada vez que se le imponga una porción de la primera parte de ejecución, utilizar una variable o una copia de variable que resida en la unidad, de suerte que esa porción se ejecute correctamente y que, en consecuencia, sea completamente funcional el software protegido,
5
- y en ausencia de la unidad, a pesar de la demanda de una porción de la primera parte de ejecución de utilizar una variable o una copia de variable que reside la unidad, no poder responder correctamente a esa demanda, de suerte que al menos no se ejecute esta porción
10 correctamente y que, en consecuencia, no sea completamente funcional el software protegido.

En otra modalidad preferida, el procedimiento de la invención consiste:

- > en la fase de protección:
15
 - definir:
 - en cuanto a un comando iniciador, un comando de instrucción,
 - en cuanto a una función dependiente, una instrucción,
- como una orden, por lo menos un argumento para un comando iniciador, que
20 corresponde por lo menos en parte a la información transmitida por el sistema de procesamiento de datos a la unidad, con el fin de iniciar la ejecución de la función dependiente correspondiente,
- un método de renombramiento de órdenes que permita renombrar las órdenes con el fin de obtener comandos iniciadores en órdenes
25 renombradas,
- y medios de restablecimiento destinados a ser aplicados en la unidad en el curso de la fase de uso, y que permitan encontrar la función dependiente a ejecutar, a partir de la orden renombrada,

en construir medios de aprovechamiento que permitan a la unidad aplicar también los medios de restablecimiento,

y en modificar el software protegido:

seleccionando en la fuente del software protegido, comandos

5 iniciadores,

modificando por lo menos una porción seleccionada de la fuente del software protegido renombrando las órdenes de comandos iniciadores seleccionados, a fin de disimular la identidad de las funciones dependientes correspondientes,

10 y produciendo:

la primera parte objeto del software protegido, esta primera parte objeto siendo tal que durante la ejecución del software protegido, se ejecutan los comandos iniciadores en órdenes renombradas,

y la segunda parte objeto del software protegido que contiene
15 los medios de aprovechamiento que aplican también los medios de restablecimiento, esta segunda parte objeto siendo tal que, después de carga en la unidad y durante la ejecución del software protegido, se reestablece la identidad de las funciones dependientes en donde la ejecución es iniciada por la primera parte de ejecución, por medio de la segunda parte de ejecución, y
20 las funciones dependientes son ejecutadas por medio de la segunda parte de ejecución,

y en la fase de uso:

en presencia de la unidad y cada vez que un comando iniciador en orden renombrada, contenido en una porción de la primera parte de
25 ejecución lo imponga, reestablecer en la unidad, la identidad de la función dependiente correspondiente y en ejecutar la misma, de modo que esta porción se ejecuta correctamente y que en consecuencia, el software protegido es completamente funcional,

y en ausencia de la unidad, a pesar de la petición de una porción de la primera parte de ejecución, iniciar la ejecución de una función dependiente en la unidad, en no poder responder correctamente a esta petición, de modo que por lo menos esta porción no se ejecuta correctamente y que en consecuencia, el software protegido no es completamente funcional.

De acuerdo con una variante de modalidad, el método de acuerdo con la invención consiste:

en la fase de protección:

en definir para al menos una función dependiente, una familia de funciones dependientes algorítmicamente equivalentes, pero iniciadas por comandos iniciadores en donde las órdenes renombradas son diferentes,

y en modificar el software protegido:

seleccionando en la fuente del software protegido por lo menos una comando iniciador en orden renombrada,

y modificando al menos una porción seleccionada de la fuente del software protegido reemplazando por lo menos la orden renombrada de un comando de iniciador en orden renombrada seleccionada, por otra orden renombrada, que inicie una función dependiente de la misma familia.

De acuerdo con una variante de modalidad, el método de acuerdo con la invención consiste:

en la fase de protección, en definir, para al menos una función dependiente, una familia de funciones dependientes algorítmicamente equivalentes:

concatenando un campo de ruido con la información que define la parte funcional de la función dependiente a ejecutar en la unidad,

o utilizando el campo de identificación de la instrucción y los campos de identificación prevista de los operandos.

De acuerdo con una variante de modalidad, el método de

acuerdo con la invención consiste:

en la fase de protección:

en definir:

como método de renombramiento de órdenes, un método de

5 codificación para codificar las órdenes,

y como medios de restablecimiento, medios que aplican un método de decodificación para decodificar las órdenes renombradas y reestablecer así la identidad de las funciones dependientes que se ejecutarán en la unidad.

10 De acuerdo con otra modalidad preferida, el método de acuerdo con la invención consiste:

en la fase de protección:

en modificar el software protegido:

seleccionando en la fuente del software protegido, por lo menos

15 una derivación condicional efectuada en por lo menos un procesamiento algorítmico seleccionado,

modificando por lo menos una porción seleccionada de la fuente del software protegido, esta modificación es tal que durante la ejecución del software protegido, se ejecuta la funcionalidad de por lo menos una derivación condicional seleccionada, por medio de la segunda parte de ejecución, en la
20 unidad,

y produciendo:

la primera parte objeto del software protegido, esta primera parte objeto siendo tal que durante la ejecución del software protegido, se ejecuta la
25 funcionalidad de por lo menos una derivación condicional seleccionada en la unidad,

y la segunda parte objeto del software protegido, esta segunda parte objeto siendo tal que después de carga en la unidad y durante la

ejecución del software protegido, aparezca la segunda parte de ejecución por medio de la cual se ejecuta la funcionalidad de por lo menos una derivación condicional seleccionada,

y en la fase de uso:

5 en presencia de la unidad y cada vez que lo impone una porción de la primera parte de ejecución, ejecutar la funcionalidad de por lo menos una derivación condicional en la unidad, de modo que esta porción se ejecuta correctamente y que en consecuencia, el software protegido es complementamente funcional,

10 y en ausencia de la unidad y a pesar de la petición de una porción de la primera parte de ejecución, ejecutar la funcionalidad de una derivación condicional en la unidad, en no poder responder correctamente a esta petición, de modo que por lo menos esta porción no se ejecuta correctamente y que en consecuencia, el software protegido no es
15 completamente funcional.

De acuerdo con una modalidad, el método de acuerdo con la invención consiste, en la fase de protección, en modificar el software protegido:

seleccionando, en la fuente del software protegido por lo menos
20 una serie de derivaciones condicionales seleccionadas,

modificando la menos una porción seleccionada de la fuente del software protegido, esta modificación es tal que durante la ejecución del software protegido, se ejecuta la funcionalidad global de por lo menos una serie seleccionada de derivaciones condicionales por medio de la segunda
25 parte de ejecución, en la unidad,

y produciendo:

la primera parte objeto del software protegido, esta primera parte objeto es tal que durante la ejecución del software protegido, se ejecuta la

funcionalidad de por lo menos una serie seleccionada de derivaciones condicionales en la unidad,

y la segunda parte objeto del software protegido, esta segunda parte objeto siendo tal que después de carga en la unidad y durante la
5 ejecución del software protegido, aparece la segunda parte de ejecución por medio de la cual se ejecuta la funcionalidad global de por lo menos una serie seleccionada de derivaciones condicionales.

El procedimiento de la invención permite así proteger la utilización de un software al poner en marcha una unidad de procesamiento y
10 memoria que tiene la particularidad de contener una parte de software ejecutable. Posteriormente toda versión derivada del software que intente funcionar sin la unidad de procesamiento y memoria impone la recreación de la parte del software contenida en la unidad de procesamiento y memoria cuando se lleva a cabo la ejecución, siempre que esta versión derivada del
15 software no sea completamente funcional.

Otras diversas características resultan de la descripción anterior con referencia a los dibujos anexos que muestran, a manera de ejemplos no limitativos, modalidades y aplicaciones del objeto de la invención.

Las figuras 1a y 1b son diagramas de bloques funcionales que
20 ilustran las diversas representaciones de un software respectivamente no protegido y protegido a través del método de acuerdo con la invención.

Las figuras 2a a 2c ilustran a manera de ejemplos, diversas modalidades de un dispositivo de aplicación del método conforme a la invención.

25 Las figuras 3a y 3b son diagramas de bloques funcionales que explican el principio general del método de acuerdo con la invención.

Las figuras 4a a 4d son esquemas que ilustran el método de
protección de acuerdo con la invención aplicando el principio de protección
por variable. mcb

Las figuras 5a a 5e son esquemas que ilustran el método de
5 protección de acuerdo con la invención aplicando el principio de protección
por detección y coerción.

Las figuras 6a a 6f son esquemas que ilustran el método de
protección de acuerdo con la invención aplicando el principio de protección
por renombramiento.

10 Las figuras 7a a 7c son esquemas que ilustran el método de
protección de acuerdo con la invención aplicando el principio de protección
por derivación condicional.

La figura 8 es un esquema que ilustra las diferentes fases de
aplicación del objeto de la invención.

15 La figura 9 ilustra un ejemplo de modalidad de un sistema que
permite la aplicación del período de construcción de la fase de protección de
acuerdo con la invención.

La figura 10 ilustra un ejemplo de modalidad de una unidad de
prepersonalización utilizada en el método de protección de acuerdo con la
20 invención.

La figura 11 ilustra un ejemplo de modalidad de un sistema que
permite la aplicación del período de confección de herramientas de la fase de
protección de acuerdo con la invención.

La figura 12 ilustra un ejemplo de modalidad de un sistema que
25 permite la aplicación del método de protección de acuerdo con la invención.

La figura 13 ilustra un ejemplo de modalidad de una unidad de
personalización utilizada en el método de protección de acuerdo con la
invención.

En el desarrollo de la descripción, se utilizarán las definiciones siguientes:

Un sistema de procesamiento de datos 3 es un sistema capaz de ejecutar un programa.

5 Una unidad de procesamiento y de memoria es una unidad capaz:

de aceptar datos provistos por un sistema de procesamiento de datos 3,

de restituir datos al sistema de procesamiento de datos 3,

de almacenar datos por lo menos en parte de manera secreta y

10 de conservar al menos una parte de éstos cuando la unidad está fuera de tensión,

y de efectuar el procesamiento algorítmico sobre los datos, una parte o la totalidad de este procesamiento es secreto.

• Una unidad 6 es una unidad (que) procesamiento y de ?
15 memoria que pone en marcha el procedimiento de conformidad con la invención.

Una unidad virgen 60 es una unidad que no aplica el método de acuerdo con la invención, pero que puede recibir informaciones que la transformen en una unidad 6.

20 Una unidad virgen 60 puede eventualmente convertirse en una unidad 6 durante la ejecución de un software protegido a través del método de acuerdo con la invención y de volver a convertirse después de la ejecución, en una unidad virgen 60.

Una unidad prepersonalizada 66 es una unidad virgen 60 que
25 haya recibido una parte de la información que le permita, después de la recepción de información complementaria, de ser transformada en una unidad 6.

La carga de información en una unidad virgen 60 o una unidad

prepersonalizada 66 corresponde a una transferencia de información en la unidad virgen 60 o la unidad prepersonalizada 66, y a un almacenamiento de dicha información transferida. Eventualmente, la transferencia puede comprender un cambio de formato de la información.

- 5 Una variable, un dato o una función contenida en el sistema de procesamiento de datos 3 será indicado con una mayúscula, mientras que una variable, un dato o una función contenida en la unidad 6 será indicado con una minúscula.

10 Un "software protegido", es un software que ha sido protegido por lo menos por un principio de protección aplicado a través del método de acuerdo con la invención.

 Un "software vulnerable", es un software que no ha sido protegido por ningún principio de protección aplicado a través del método de acuerdo con la invención.

- 15 En el caso en donde la diferenciación entre un software vulnerable y un software protegido no sea de importancia, se utiliza el término "software".

 Un software se presenta bajo diversas representaciones de acuerdo con el momento considerado en su ciclo de vida:

- 20 una representación fuente,
 una representación objeto,
 una distribución,
 o una representación dinámica.

- 25 Una representación fuente de un software está comprendida como una representación que después de transformación, da una representación objeto. Una representación fuente se puede presentar según diferentes niveles, desde un nivel conceptual abstracto hasta un nivel ejecutable directamente por un sistema de procesamiento de datos o una

unidad de procesamiento y de memoria.

Una representación objeto de un software corresponde a un nivel de representación que después de transferencia en una distribución y de carga posterior en un sistema de procesamiento de datos o una unidad de procesamiento y de memoria, puede ser ejecutado. Se puede tratar, por ejemplo, de un código binario, de un código interpretado, etc.

Una distribución es un soporte físico o virtual que contiene la representación objeto, esta distribución debe estar puesta a disposición del usuario para permitirle usar el software.

Una representación dinámica corresponde a la ejecución del software a partir de su distribución.

Una porción de software corresponde a una parte cualquiera de software y puede corresponder, por ejemplo, a una o más instrucciones consecutivas o no, y/o a uno o más bloques funcionales consecutivos o no, y/o a una o más funciones, y/o uno o más subprogramas, y/o uno o más módulos. Una porción de un software puede corresponder también a la totalidad de este software.

Las figuras 1a y 1b ilustran las diversas representaciones respectivamente de un software vulnerable 2v en sentido general, de un software protegido 2p de acuerdo con el método de la invención.

La figura 1a ilustra diversas representaciones de un software vulnerable 2v que aparece en el transcurso de su ciclo de vida. El software vulnerable 2v puede aparecer bajo una de las siguientes representaciones:

una representación fuente 2vs,
una representación objeto 2vo,
una distribución 2vd. Esta distribución se puede presentar comúnmente bajo la forma de un medio de distribución físico tal como un CDROM o bajo la forma de ficheros distribuidos a través de una red (GSM,

Internet,...),

- o una representación dinámica 2ve que corresponde a la ejecución del software vulnerable 2v en un sistema de procesamiento de datos 3 de cualquier tipo conocido, que comprende normalmente, por lo
- 5 menos un procesador 4.

La figura 1b ilustra diversas representaciones de un software protegido 2p que aparece durante su ciclo de vida. El software protegido 2p puede aparecer bajo una de las siguientes representaciones:

- una representación fuente 2ps que comprende una primera
- 10 parte fuente destinada al sistema de procesamiento de datos 3 y una segunda parte fuente destinada para la unidad 6, y una parte de las partes fuente puede estar contenida generalmente en registros comunes,

- una representación objeto 2po que comprende una primera
- 15 parte objeto 2pos destinada al sistema de procesamiento de datos 3 y una segunda parte objeto 2pou destinada para la unidad 6,
- una distribución 2pd que comprende:

- una primera parte de distribución 2pds que contiene la primera
- parte objeto 2pos, esta primera parte de distribución 2pds está destinada al sistema de procesamiento de datos 3 y se puede presentar comúnmente bajo
- 20 la forma de un medio de distribución físico tal como un CDROM, o bajo la forma de ficheros distribuidos a través de una red (GSM, Internet, ...),

- y una segunda parte de distribución 2pdu que se presenta
- bajo la forma: de al menos una unidad ~~pre~~ personalizada 66 en la cual fue carga una parte de la segunda parte objeto 2pou y por la que el usuario debe
- 25 terminar la personalización al cambiar las informaciones complementarias, para obtener una unidad 6, obteniéndose dichas informaciones complementarias, pro ejemplo, mediante la carga o carga a distancia a través de una red,

> o de al menos una unidad 6 en la cual fue cargada la segunda parte objeto 2pou,

o una representación dinámica 2pe que corresponde a la ejecución del software protegido 2p. Esta representación dinámica 2pe
5 comprende una primera parte de ejecución 2pes que es ejecutada en el sistema de procesamiento de datos 3 y una segunda parte de ejecución 2peu que es ejecutada en la unidad 6.

En el caso en donde la diferenciación entre las diferentes representaciones del software protegido 2p no tenga importancia, se utilizan
10 las expresiones de primera parte del software protegido y segunda parte del software protegido.

La aplicación del método de acuerdo con la invención conforme a la representación dinámica de la figura 1b, utiliza un dispositivo 1p que comprende un sistema de procesamiento de datos 3 unido a través de un
15 enlace químico a una unidad 6. El sistema de procesamiento de datos 3 es de cualquier tipo y comprende, normalmente, por lo menos un procesador 4. El sistema de procesamiento de datos 3 puede ser una computadora o puede ser parte, por ejemplo, de diversas máquinas, dispositivos, productos fijos o móviles, o vehículos en el sentido general. El enlace 5 se puede realizar de
20 cualquier manera posible, tal como por ejemplo a través de una línea de serie, un bus USB, un enlace de radio, un enlace óptico, un enlace de red o una conexión eléctrica directa sobre un circuito del sistema de procesamiento de datos 3, etc. Cabe señalar que la unidad 6 se puede encontrar eventualmente de manera física en el interior del mismo circuito integrado que el procesador
25 4 del sistema de procesamiento de datos 3. En este caso, la unidad 6 puede ser considerada como un coprocesador con relación al procesador 4 del sistema de procesamiento de datos 3 y el enlace 5 es interno al circuito integrado.

Las figuras 2a a 2c muestran de manera ilustrativa y a título no limitativo, diversas modalidades del dispositivo 1p que permiten la aplicación del método de protección de acuerdo con la invención.

En el ejemplo de modalidad ilustrado en la figura 2a, el
5 dispositivo de protección 1p comprende, como sistema de procesamiento de datos 3, una computadora y, como una unidad 6, una tarjeta de chip 7 y su interfaz 8 comúnmente llamada lector de tarjeta. La computadora 3 está conectada a la unidad 6 mediante un enlace 5. Durante la ejecución de un software protegido 2p, la primera parte de ejecución 2pes que es ejecutada
10 en la computadora 3 y la segunda parte de ejecución 2peu que es ejecutada en la tarjeta de chip 7 y su interfaz 8, deben ser funcionales a manera que el software protegido 2p sea completamente funcional.

En el ejemplo de modalidad ilustrado en la figura 2b, el dispositivo de protección 1p provee un producto 9 en sentido general, que
15 comprende diversos elementos 10 adaptados a la o las funciones asumidas por dicho producto 9. El dispositivo de protección 1p comprende, por una parte, un sistema de procesamiento de datos 3 metido en el producto 9 y, por otra parte, una unidad 6 asociada al producto 9. Para que el producto 9 sea completamente funcional, el software protegido 2p debe ser completamente
20 funcional. Así, durante la ejecución del software protegido 2p, la primera parte de ejecución 2pes que es ejecutada en el sistema de procesamiento de datos 3 y la segunda parte de ejecución 2peu que es ejecutada en la unidad 6, deben ser funcionales. Este software protegido 2p permite entonces de manera indirecta, proteger contra un uso no autorizado, el producto 9 o una
25 de sus funcionalidades. Por ejemplo, el producto 9 puede ser una instalación, un sistema, una máquina, un juguete, un aparato electrodoméstico, un teléfono, etc.

En el ejemplo de modalidad ilustrado en la figura 2c, el

dispositivo de protección 1p incluye varias computadoras, así como una parte de una red de comunicación. El sistema de procesamiento de datos 3 es una primera computadora conectada mediante un enlace 5 de tipo red, a una unidad 6 constituida por una segunda computadora. Para aplicar la invención, se utiliza la segunda computadora 6 como un servidor de licencias para un software protegido 2p. Durante la ejecución del software protegido 2p, la primera parte de ejecución 2pes que es ejecutada en la primera computadora 3 y la segunda parte de ejecución 2peu que es ejecutada en la segunda computadora 6, deben ser funcionales de manera que el software protegido 2p sea completamente funcional.

La figura 3a permite explicar de manera más precisa el método de protección de acuerdo con la invención. Cabe señalar que un software vulnerable 2v se considera siendo ejecutado totalmente en un sistema de procesamiento de datos 3. Por el contrario, en el caso de la aplicación de un software protegido 2p, el sistema de procesamiento de datos 3 comprende medios de transferencia 12 conectados mediante el enlace 5, a medios de transferencia 13 que forman parte de la unidad 6 que permiten comunicar entre ellas, la primera parte de ejecución 2pes y la segunda parte de ejecución 2peu del software protegido 2p.

Se debe considerar que los medios transferencia 12, 13 tienen naturaleza de software y/o material y son aptos para asegurar y eventualmente, optimizar la comunicación de datos entre el sistema de procesamiento de datos 3 y la unidad 6. Estos medios transferencia 12, 13 están adaptados para permitir disponer de un software protegido 2p que de preferencia, es independiente del tipo del enlace 5 utilizado. Estos medios transferencia 12, 13 no forman parte del objeto de la invención y no son descritos de manera más precisa, ya que son conocidos por los expertos en la técnica. La primera parte del software protegido 2p comprende comandos.

Durante la ejecución del software protegido 2p, la ejecución de estos comandos mediante la primera parte de ejecución 2pes permite la comunicación entre la primera parte de ejecución 2pes y la segunda parte de ejecución 2peu. En el desarrollo de la descripción, estos comandos son representados por IN, OUT o TRIG.

Tal como se ilustra en la figura 3b, para permitir la aplicación de la segunda parte de ejecución 2peu del software protegido 2p, la unidad 6 comprende medios de protección 14. Los medios de protección 14 comprenden medios de memoria 15 y medios de procesamiento 16.

Con el objeto de simplificar el desarrollo de la descripción se prefiere considerar, durante la ejecución del software protegido 2p, la presencia de la unidad 6 o la ausencia de la unidad 6. En realidad, una unidad 6 que presenta medios de protección 14 inadaptados para la ejecución de la segunda parte de ejecución 2peu del software protegido 2p también es considerada como ausente, cada vez que la ejecución del software protegido 2p no sea correcta. En otros términos:

una unidad 6 físicamente presente y que comprende medios de protección 14 adaptados para la ejecución de la segunda parte de ejecución 2pe del software protegido 2p, siempre es considerada como presente,

una unidad 6 físicamente presente pero que comprende medios de protección 14 inadaptados, es decir, que no permiten la aplicación correcta de la segunda parte de ejecución 2peu del software protegido 2p es considerada como presente, cuando funcione correctamente, y como ausente cuando no funcione correctamente,

y una unidad 6 físicamente ausente es siempre considerada como ausente.

En el caso en donde la unidad 6 esté constituida por una tarjeta de chip 7 y su interfaz 8, los medios de transferencia 13 son fragmentos en

dos partes en donde una se encuentra sobre la interfaz 8 y la otra se encuentra sobre la tarjeta de chip 7. En este ejemplo de modalidad, la ausencia de la tarjeta de chip 7 es considerada como equivalente a la ausencia de la unidad 6. En otras palabras, en ausencia de la tarjeta de chip 7 y/o de su interfaz 8, los medios de protección 14 no son accesibles y no permiten la ejecución de la segunda parte de ejecución 2peu del software protegido, de modo que el software protegido 2p no es completamente funcional.

De conformidad con la invención, el procedimiento de protección pretende poner en marcha un principio de protección, llamado de "detección y coerción", con el cual se efectúa una descripción en relación con las figuras 5a a 5e.

Para la aplicación del principio de protección por detección y coerción, se define:

por lo menos una característica de ejecución de software susceptible de ser vigilada por lo menos en parte en la unidad 6,

por lo menos un criterio a obedecer para al menos una característica de ejecución de software,

medios de detección 17 para aplicar en la unidad 6 y que permiten detectar que por lo menos una característica de ejecución de software no obedece al menos un criterio asociado,

y medios de coerción 18 para aplicar en la unidad 6 y que permiten informar al sistema de procesamiento de datos 3 y/o de modificar la ejecución de un software, cuando por lo menos un criterio no es obedecido.

Para la aplicación del principio de protección por detección y coerción, también se construyen medios de aprovechamiento que permiten transformar una unidad virgen 60 en una unidad 6 que ponga en marcha al menos los medios de detección 17 y los medios de coerción 18.

La figura 5a ilustra los medios de necesarios para la aplicación de este principio de protección por detección y coerción. La unidad 6 comprende medios de detección 17 y los medios de coerción 18 que pertenecen a los medios de procesamiento 16. Los medios de coerción 18 son informados del no obedecer un criterio por parte de los medios de detección 17.

De una forma más precisa, los medios de detección 17 utilizan informaciones procedentes de los medios de transferencia 13 y/o de los medios de memoria 15 y/o de los medios de procesamiento 16, a fin de vigilar una o más características de ejecución de software. A cada característica de ejecución de software está fijo por lo menos un criterio a obedecer.

En el caso en donde se detecta que por lo menos una característica de ejecución de software no obedece al menos un criterio, los medios de detección 17 informan a los medios de coerción 18. Estos medios de coerción 18 están adaptados para modificar, de la manera adecuada, el estado de la unidad 6.

Para la aplicación del principio de protección por detección y coerción, también se selecciona:

por lo menos una característica de ejecución de software a vigilar, entre las características de ejecución susceptibles de ser vigiladas,

por lo menos un criterio a obedecer para al menos una característica de ejecución de software seleccionado,

en la fuente del software vulnerable 2vs, por lo menos un procesamiento algorítmico para el cual al menos se vigile una característica de ejecución de software,

y en la fuente del software vulnerable 2vs, al menos una porción que contiene por lo menos un procesamiento algorítmico seleccionado.

Al menos una porción seleccionada de la fuente del software

vulnerable 2vs es entonces modificada, a manera de obtener la fuente del software protegido 2ps. Esta modificación es tal que especialmente durante la ejecución del software protegido 2p:

al menos una porción de la primera parte de ejecución 2pes, que
5 es ejecutada en el sistema de procesamiento de datos 3, toma en cuenta que por lo menos se vigilará una característica de ejecución de software seleccionado, al menos en parte en la unidad 6,

y la segunda parte de ejecución 2peu, que es ejecutada en la unidad 6, vigila por lo menos en parte, una característica de ejecución de
10 software seleccionado.

Durante la ejecución del software protegido 2p, protegido por este principio de protección por detección y coerción, en presencia de la unidad 6:

como todos los criterios correspondientes a todas las
15 características de ejecución vigiladas de todas las porciones modificadas del software protegido 2p son obedecidos, estas porciones modificadas del software protegido 2p funcionan de manera nominal, y en consecuencia, el software protegido 2p funciona de manera nominal,

y si por lo menos uno de los criterios que corresponden a una
20 característica de ejecución vigilada de una porción del software protegido 2p no es obedecido, el sistema de procesamiento de datos 3 es informado y/o el funcionamiento de la porción del software protegido 2p es modificado, de modo que el funcionamiento del software protegido 2p sea modificado.

Desde luego, en ausencia de la unidad 6, por lo menos una
25 petición de una porción de la primera parte de ejecución 2pes del software protegido 2p de utilizar la unidad 6 no puede ser satisfecha correctamente, de modo que por lo menos esta porción no se ejecute correctamente y que en consecuencia, el software protegido 2p no sea completamente funcional.

Para la aplicación del principio de protección por detección y coerción, se utilizan de preferencia dos tipos de características de ejecución de software.

El primer tipo de característica de ejecución de software
5 corresponde a una variable de medición de la ejecución de un software y el segundo tipo corresponde a un perfil de uso de un software. Estos dos tipos de características pueden ser utilizados de manera independiente o en combinación.

Para la aplicación del principio de protección por detección y
10 coerción que utiliza como característica de ejecución, una variable de medición de la ejecución de software, se define:

en los medios de memoria 15, la posibilidad de memorizar por lo menos una variable de medición que sirva para cuantificar el uso de por lo menos una funcionalidad de software,

15 en los medios de detección 17, la posibilidad de vigilar al menos un umbral asociado con cada variable de medición,

y medios de actualización que permiten actualizar cada variable de medición en función del uso de la funcionalidad a la cual está asociada.

También se construyen medios de aprovechamiento que
20 aplican, además de los medios de detección 17 y de los medios de coerción 18, los medios de actualización.

También se selecciona, en la fuente del software vulnerable 2vs:
por lo menos una funcionalidad del software vulnerable 2v en
donde el uso es susceptible de ser vigilado gracias a una variable de
25 medición,

por lo menos una variable de medición que sirve para cuantificar el uso de dicha funcionalidad,

por lo menos un umbral asociado con la variable de medición

que corresponde a un límite de uso de dicha funcionalidad,

y por lo menos un método de actualización de la variable de medición en función del uso de dicha funcionalidad.

La fuente del software vulnerable 2vs es entonces modificada, a
5 manera de obtener la fuente de software protegido 2ps, esta modificación
siendo tal que durante la ejecución del software protegido 2p, la segunda
parte de ejecución 2peu:

actualiza la variable de medición en función del uso de dicha
funcionalidad,

10 y toma en cuenta por lo menos un exceso de umbral.

En otros términos, durante la ejecución del software protegido
2p, la variable de medición es actualizada en función del uso de dicha
funcionalidad, y mientras el umbral es excedido, los medios de detección 17
informan a los medios de coerción 18 que toman una decisión adaptada para
15 informar al sistema de procesamiento de datos 3 y/o modificar los
procesamientos efectuados por lo medios de procesamiento 16 que permiten
modificar el funcionamiento de la porción de software protegido 2p, de modo
que se modifique el funcionamiento del software protegido 2p.

Para la aplicación de una primera variante preferida de
20 modalidad del principio de protección por detección y coerción que utilizan,
como característica, una variable de medición, se define:

para al menos una variable de medición, varios umbrales
asociados,

y medios de coerción diferentes que corresponden a cada uno
25 de estos umbrales.

También se selecciona, en la fuente del software vulnerable 2vs:
por lo menos una variable de medición que sirve para cuantificar
el uso de al menos una funcionalidad del software y a la cual deben estar

asociados varios umbrales que corresponden a los límites diferentes de uso de dicha funcionalidad,

y por lo menos dos umbrales asociados a la variable de medición.

5 La fuente del software vulnerable 2vs es entonces modificada, a manera de obtener la fuente del software protegido 2ps, esta modificación siendo tal que durante la ejecución del software protegido 2p, la segunda parte de ejecución 2peu:

10 actualiza la variable de medición en función del uso de dicha funcionalidad,

y toma en cuenta, de manera diferente, los excedentes de diversos umbrales.

En otros términos, por lo regular, durante la ejecución del software protegido 2p, mientras el primer umbral es excedido, la unidad 6
15 informa al sistema de procesamiento de datos 3 ordenando al software protegido 2p de no utilizar más esta funcionalidad. Si el software protegido 2p continúa utilizando esta funcionalidad, el segundo umbral podrá ser excedido. En el caso en donde el segundo umbral es excedido, los medios de coerción
18 pueden volver inoperante la funcionalidad seleccionada y/o volver
20 inoperante el software protegido 2p.

Para la aplicación de una segunda variante preferida de modalidad del principio de protección por detección y coerción utilizando, como característica, una variable de medición, se definen medios de recarga que permiten acreditar por lo menos un uso complementario para al menos
25 una funcionalidad de software vigilado por una variable de medición.

También se construyen medios de aprovechamiento que aplican, además los medios de detección 17, medios de coerción 18 y medios de actualización, los medios de recarga.

También se selecciona, en la fuente del software vulnerable 2vs, por lo menos una variable de medición que sirve para limitar el uso de al menos una funcionalidad del software y a la cual se debe poder acreditar por lo menos un uso complementario.

5 La fuente del software vulnerable 2vs es entonces modificada, a manera de obtener la fuente del software protegido 2ps, esta modificación siendo tal que en una fase llamada de recarga, por lo menos un uso complementario de al menos una funcionalidad que corresponde a una variable de medición seleccionada puede ser acreditado.

10 En la fase de recarga, se procede a la reactualización de por lo menos una variable de medición seleccionada y/o de por lo menos un umbral asociado, a manera de permitir al menos un uso complementario de la funcionalidad correspondiente. En otros términos, es posible, en la fase de recarga, acreditar los usos complementarios de por lo menos una
15 funcionalidad del software protegido 2p.

Para la aplicación del principio de protección por detección y coerción utilizando, como característica, un perfil de uso de software, se define como criterio a obedecer para este perfil de uso, por lo menos una característica de ejecución de software.

20 También se selecciona, en la fuente del software vulnerable 2vs: por lo menos un perfil de uso a vigilar,
y por lo menos una característica de ejecución que debe obedecer al menos un perfil de uso seleccionado.

La fuente del software vulnerable 2vs es entonces modificada, a
25 manera de obtener la fuente del software protegido 2ps, esta modificación siendo tal que durante la ejecución del software protegido 2p, la segunda parte de ejecución 2peu, obedece todas las características de ejecución seleccionadas. En otros términos, la unidad 6 vigila la manera en la cual la

segunda parte de ejecución 2peu es ejecutada y puede informar al sistema de procesamiento de datos 3 y/o modificar el funcionamiento del software protegido 2p, en el caso en donde no se obedezca por lo menos una característica de ejecución.

5 Durante la ejecución del software protegido 2p, protegido por este principio, en presencia de la unidad 6:

 mientras que se obedecen todas las características de ejecución de todas las porciones modificadas del software protegido 2p, estas porciones modificadas del software protegido 2p funcionan de manera nominal y en
10 consecuencia, el software protegido 2p funciona de manera nominal,

 y si no se obedece al menos una característica de ejecución de una porción de software protegido 2p, el sistema de procesamiento de datos 3 es informado y/o el funcionamiento de la porción del software protegido 2p es modificado, de modo que el funcionamiento del software protegido 2p sea
15 modificado.

 Se puede contemplar la vigilancia de diferentes características de ejecución, como por ejemplo la vigilancia de la presencia de instrucciones que comprende un marcador o la vigilancia del encadenamiento de ejecución por lo menos para una parte de las instrucciones.

20 Para la aplicación del principio de protección por detección y coerción que utilizan como característica de ejecución a obedecer, la vigilancia del encadenamiento de ejecución al menos para una parte de las instrucciones, se define:

 un juego de instrucciones en donde las instrucciones son
25 susceptibles de ser ejecutadas en la unidad 6,

 un juego de comandos de instrucción para este juego de instrucciones, estos comandos de instrucciones son susceptibles de ser ejecutados en el sistema de procesamiento de datos 3. La ejecución de cada

uno de estos comandos de instrucciones en el sistema de procesamiento de datos 3 inicia en la unidad 6, la ejecución de la instrucción correspondiente,

medios de detección 17 que permiten detectar que el encadenamiento de las instrucciones no corresponde a aquel deseado,

5 y medios de coerción 18 que permiten informar al sistema de procesamiento de datos 3 y/o de modificar la ejecución de un software cuando el encadenamiento de las instrucciones no corresponde a aquel deseado.

También se construyen medios de aprovechamiento que permiten, en la unidad 6, ejecutar además las instrucciones del juego de
10 instrucciones, la ejecución de estas instrucciones es iniciada por la ejecución en el sistema de procesamiento de datos 3 de comandos de instrucciones.

También se selecciona, en la fuente del software vulnerable 2vs, por lo menos un procesamiento algorítmico que llega a ser deportado a la unidad 6 y para el cual se va a vigilar el encadenamiento de por lo menos una
15 parte de las instrucciones.

La fuente del software vulnerable 2vs es entonces modificada a manera de obtener la fuente del software protegido 2ps, esta modificación es tal que durante la ejecución del software protegido 2p:

la segunda parte de ejecución 2peu ejecuta por lo menos la
20 funcionalidad del procesamiento algorítmico seleccionado,

el procesamiento algorítmico seleccionado es fragmentado en instrucciones,

se especifica el encadenamiento que deben obedecer por lo menos algunas de las instrucciones durante su ejecución en la unidad 6,

25 y la primera parte de ejecución 2pes del software protegido 2p ejecuta comandos de instrucciones que inician la ejecución de instrucciones en la unidad 6.

Durante la ejecución del software protegido 2p, protegido por

este principio, en presencia de la unidad 6:

mientras que el encadenamiento de las instrucciones de todas las porciones modificadas del software protegido 2p corresponde a aquel deseado, estas porciones modificadas del software protegido 2p funcionan de manera nominal y en consecuencia, el software protegido 2p funciona de manera nominal,

y si el encadenamiento de las instrucciones de una porción de software protegido 2p ejecutadas en la unidad 6 no corresponde a aquel deseado, el sistema de procesamiento de datos 3 es informado y/o el funcionamiento de la porción del software protegido 2p es modificado, de modo que el funcionamiento del software protegido 2p sea modificado.

La figura 5b ilustra un ejemplo de aplicación del principio de protección por detección y coerción que utilizan, como característica de ejecución a obedecer la vigilancia del encadenamiento de ejecución de por lo menos una parte de las instrucciones, en el caso en donde se obedezca el encadenamiento deseado.

La primera parte de ejecución 2pes del software protegido 2p, ejecutada en el sistema de procesamiento de datos 3, ejecuta comandos de instrucciones CI_i que inician, en la unidad 6, la ejecución de instrucciones i_i que pertenecen al juego de instrucciones. En el juego de instrucciones, por lo menos algunas de las instrucciones comprenden cada una, una parte que define la funcionalidad de la instrucción y una parte que permite verificar el encadenamiento deseado para la ejecución de instrucciones. En este ejemplo, los comandos de instrucciones CI_i son representados por $TRIG(i_i)$ y el encadenamiento deseado para la ejecución de instrucciones es i_n, i_{n+1} y i_{n+2} . La ejecución en la unidad 6, de la instrucción i_n da el resultado a, y la ejecución de la instrucción i_{n+1} da el resultado b. La instrucción i_{n+2} utiliza como operando, los resultados a y b de las instrucciones i_n y i_{n+1} y su ejecución da el

resultado c.

Teniendo en cuenta que este encadenamiento de instrucciones ejecutadas en la unidad 6 corresponde a aquel deseado, se provee un funcionamiento normal o nominal del software protegido 2p.

5 La figura 5c ilustra un ejemplo de aplicación del principio de protección por detección y coerción que utiliza, como característica de ejecución a obedecer, la vigilancia del encadenamiento de ejecución de por lo menos una parte de las instrucciones, en el caso en donde no se obedezca el encadenamiento deseado.

10 De acuerdo con este ejemplo, el encadenamiento deseado para la ejecución de instrucciones siempre es i_n , i_{n+1} y i_{n+2} . Sin embargo, el encadenamiento de ejecución de las instrucciones es modificado por el reemplazo de la instrucción i_n por la instrucción i'_n , de modo que el encadenamiento efectivamente ejecutado es i'_n , i_{n+1} y i_{n+2} . La ejecución de la

15 instrucción i'_n da el resultado a, es decir, el mismo resultado que la ejecución de la instrucción i_n . Sin embargo, posteriormente durante la ejecución de la instrucción i_{n+2} , los medios de detección 17 detectan que la instrucción i'_n no corresponde a la instrucción deseada para generar el resultado a utilizado como operando de la instrucción i_{n+2} . Los medios de detección 17 informan a

20 los medios de coerción 18 que modifican en consecuencia, el funcionamiento de la instrucción i_{n+2} , de modo que la ejecución de la instrucción i_{n+2} da el resultado c' que puede ser diferente a c. Desde luego, si la ejecución de la instrucción i'_n da un resultado a' diferente al resultado a de la instrucción i_n , es claro que el resultado de la instrucción i_{n+2} también puede ser diferente a c.

25 En la medición en que el encadenamiento de ejecución de las instrucciones ejecutadas en la unidad 6 no corresponda a aquel deseado, se puede obtener entonces una modificación del funcionamiento del software protegido 2p.

Las figuras 5d y 5e ilustran una variante preferida de modalidad del principio de protección por detección y coerción que utilizan, como característica de ejecución a obedecer, la vigilancia del encadenamiento de ejecución de por lo menos una parte de las instrucciones. De acuerdo con

5 esta variante preferida, se define un juego de instrucciones en donde por lo menos algunas instrucciones trabajan sobre registros y utilizan por lo menos un operando con el objeto de arrojar un resultado.

Como se ilustra en la figura 5d, se define para al menos algunas de las instrucciones que trabajan sobre registros, una parte PF que define la

10 funcionalidad de la instrucción y una parte PE que define al encadenamiento deseado para la ejecución de las instrucciones. La parte PF corresponde al código operación conocido por el experto en la técnica. La parte PE que define el encadenamiento deseado, comprende campos de bits que corresponden a:

15 un campo de identificación de la instrucción CII,

y para cada operando k de la instrucción, con k variando de 1 a K , y K número de operandos de la instrucción:

un campo bandera CD_k que indica si conviene verificar la procedencia del operando k ,

20 y un campo de identificación previsto CIP_k del operando, que indique la identidad esperada de la instrucción que haya generado el contenido del operando k .

Como se ilustra en la figura 5e, el juego de instrucciones comprende V registros que pertenecen a los medios de procesamiento 16,

25 cada registro es nombrado R_v , con v variando de 1 a V . Para cada registro R_v , se definen dos campos, a saber:

un campo funcional CF_v , conocido por el experto en la técnica y que permite almacenar el resultado de la ejecución de instrucciones,

y un campo de identificación generado CIG_v , que permite memorizar la identidad de la instrucción que haya generado el contenido del campo funcional CF_v . Este campo de identificación generada CIG_v es actualizado automáticamente con el contenido del campo de identificación de la instrucción CII habiendo generado el campo funcional CF_v . Este campo de identificación generada CIG_v no es accesible, ni modificable por alguna instrucción y sirve únicamente para los medios de detección 17.

Durante la ejecución de una instrucción, los medios de detección 17 efectúan para cada operando k las siguientes operaciones:

se lee el campo bandera CD_k ,

si el campo bandera CD_k lo impone, el campo de identificación prevista CIP_k y el campo de identificación generada CIG_v que corresponden al registro utilizado para el operando k son ambos leídos,

la igualdad de los dos campos CIP_k y CIG es controlada,

y si la igualdad es falsa, los medios de detección 17 consideran que el encadenamiento de ejecución de instrucciones no ha sido obedecido.

Los medios de coerción 18 permiten modificar el resultado de las instrucciones cuando los medios de detección 17 les informan de un encadenamiento de instrucciones no obedecido. Una modalidad preferida consiste en modificar la parte funcional PF de la instrucción durante la ejecución o la parte funcional PF de instrucciones posteriores.

En otra característica conveniente de la invención, el procedimiento de protección busca poner en marcha un principio de protección llamado "variable" con el cual se efectúa una descripción en relación a las figuras 4a a 4d.

Para la aplicación del principio de protección por variable, se selecciona en la fuente del software vulnerable $2vs$, por lo menos una variable que durante la ejecución del software vulnerable $2v$, defina parcialmente el

estado de ésta. Por estado de un software, se debe comprender el conjunto de informaciones, en un momento dado, necesarias para la ejecución completa de este software, de modo que la ausencia de dicha variable seleccionada perjudica la ejecución completa de este software. También se
 5 selecciona por lo menos una porción de fuente del software vulnerable 2vs que contiene al menos una variable seleccionada.

Por lo menos una porción seleccionada de la fuente del software vulnerable 2vs es entonces modificada, a manera de obtener la fuente del software protegido 2ps. Esta modificación es tal que durante la ejecución del
 10 software protegido 2p, por lo menos una porción de la primera parte de ejecución 2pes que es ejecutada en el sistema de procesamiento de datos 3, toma en cuenta que por lo menos una variable seleccionada o al menos una copia de variable elegida que reside en la unidad 6.

La figura 4a ilustra un ejemplo de ejecución de un software vulnerable 2v. En este ejemplo, aparece en el transcurso de la ejecución del software vulnerable 2v en el sistema de procesamiento de datos 3:

en el momento t_1 , la asignación del dato X a la variable V_1 , representado por $V_1 \leftarrow X$,

en el momento t_2 , la asignación del valor de la variable V_1 a la
 20 variable Y, representada por $Y \leftarrow V_1$,

en el momento t_3 , la asignación del valor de la variable V_1 a la variable Z, representada por $Z \leftarrow V_1$.

La figura 4b ilustra un ejemplo de una primera forma de aplicación de la invención para la cual la variable reside en la unidad 6. En
 25 este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes de software protegido 2p, y en presencia de la unidad 6, resulta:

en el momento t_1 , la ejecución de un comando de transferencia

que inicia la transferencia del dato X desde el sistema de procesamiento de datos 3 hacia la variable v_1 situada en los medios de memoria 15 de la unidad 6, este comando de transferencia es representado por OUT (v_1 , X) y corresponde al final a la asignación del dato X a la variable v_1 ,

5 en el momento t_2 , la ejecución de un comando de transferencia que inicia la transferencia del valor de la variable v_1 que reside en la unidad 6 hacia el sistema de procesamiento de datos 3 con el fin de asignar a la variable Y, este comando de transferencia es representado por IN (v_1) y corresponde al final a la asignación del valor de la variable v_1 a la variable Y,

10 y en el momento t_3 , la ejecución de un comando de transferencia que inicia la transferencia del valor de la variable v_1 que reside en la unidad 6 hacia el sistema de procesamiento de datos 3 con el fin de asignarlo a la variable Z, este comando de transferencia es representado por IN (v_1) y corresponde al final a la asignación del valor de la variable v_1 a la variable Z.

15 Cabe señalar que durante la ejecución del software protegido 2p, por lo menos una variable reside en la unidad 6. Así, cuando una porción de la primera parte de ejecución 2pes del software protegido 2p lo impone, y en presencia de la unidad 6, el valor de esta variable que reside en la unidad 6 es transferido hacia el sistema de procesamiento de datos 3 para ser
20 utilizado por la primera parte de ejecución 2pes del software protegido 2p, de modo que esta porción sea ejecutada correctamente y que en consecuencia, el software protegido 2p sea completamente funcional.

La figura 4c ilustra un ejemplo de una segunda forma de aplicación de la invención para la cual una copia de la variable reside en la
25 unidad 6. En este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes del software protegido 2p, y en presencia de la unidad 6, aparece:

en el momento t_1 , la asignación del dato X a la variable V_1

situado en el sistema de procesamiento de datos 3, así como la ejecución de un comando de transferencia que inicia la transferencia del dato X desde el sistema de procesamiento de datos 3 hacia la variable v_1 situada en los medios de memoria 15 de la unidad 6, este comando de transferencia siendo

5 representado por $\text{OUT}(v_1, X)$,

en el momento t_2 , la asignación del valor de la variable V_1 a la variable Y,

y en el momento t_3 , la ejecución de un comando de transferencia que inicia la transferencia del valor de la variable v_1 que reside en la unidad 6

10 hacia el sistema de datos 3 con el fin de asignarlo a la variable Z, este comando de transferencia siendo representado por $\text{IN}(v_1)$.

Cabe señalar que durante la ejecución del software protegido 2p, por lo menos una copia de una variable reside en la unidad 6. Así, cuando una porción de la primera parte de ejecución 2pes del software protegido 2p lo

15 impone, y en presencia de la unidad 6, el valor de esta copia de variable que reside en la unidad 6 es transferido hacia el sistema de procesamiento de datos 3 para ser utilizado por la primera parte de ejecución 2pes del software protegido 2p, de modo que esta porción es ejecutada correctamente y que en consecuencia, el software protegido 2p es completamente funcional.

20 La figura 4d ilustra un ejemplo de intento de ejecución del software protegido 2p, mientras la unidad 6 está ausente. En este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes del software protegido 2p:

en el momento t_1 , la ejecución del comando de transferencia

25 $\text{OUT}(v_1, X)$ no puede iniciar la transferencia del dato X hacia la variable v_1 , teniendo en cuenta la ausencia de la unidad 6,

en el momento t_2 , la ejecución del comando de transferencia $\text{IN}(v_1)$ no puede iniciar la transferencia del valor de la variable v_1 hacia el

sistema de procesamiento de datos 3, teniendo en cuenta la ausencia de la unidad 6,

y en el momento t_1 , la ejecución del comando de transferencia $IN(v_1)$ no puede iniciar la transferencia del valor de la variable v_1 hacia el sistema de procesamiento de datos 3, teniendo en cuenta la ausencia de la unidad 6.

Resulta entonces que en ausencia de la unidad 6, por lo menos una petición de una porción de la primera parte de ejecución 2pes para utilizar una variable o una copia de variable que reside en la unidad 6, no puede ser satisfecha correctamente, de modo que por lo menos esta porción no se ejecuta correctamente y en consecuencia, el software protegido 2p no es completamente funcional.

Cabe señalar que las transferencias de datos entre el sistema de procesamiento de datos 3 y la unidad 6 ilustrados en los ejemplos anteriores solamente utilizan asignaciones simples, pero el experto en la técnica sabrá combinarla con otras operaciones para llegar a operaciones complejas tales como por ejemplo $OUT(v_1, 2 \cdot X + 3)$ o bien $Z \leftarrow (5 \cdot v_1 + v_2)$.

De acuerdo con otra característica ventajosa de la invención, el método de protección contempla aplicar un principio de protección, llamado por "renombramiento", del cual se efectúa una descripción con relación a las figuras 6a a 6f.

Para la aplicación del principio de protección por renombramiento, se define:

un conjunto de funciones dependientes, en donde las funciones dependientes son susceptibles de ser ejecutadas, por medio de la segunda parte de ejecución 2peu, en la unidad 6, y eventualmente de transferir datos entre el sistema de procesamiento de datos 3 y la unidad 6, este conjunto de funciones dependientes puede ser finito o infinito,

un conjunto de comandos iniciadores para estas funciones dependientes, estos comandos iniciadores son susceptibles de ser ejecutados en el sistema de procesamiento de datos 3 y de iniciar en la unidad 6, la ejecución de funciones dependientes correspondientes,

5 para cada comando iniciador, una orden correspondiente por lo menos en parte a la información transmitida por la primera parte de ejecución 2pes, a la segunda parte de ejecución 2peu, con el fin de iniciar la ejecución de la función dependiente correspondiente, esta orden se presenta bajo la forma de por lo menos un argumento del comando iniciador,

10 un método de renombramiento de órdenes destinado a ser aplicado durante la modificación del software vulnerable, dicho método permite renombrar las órdenes con el fin de obtener comandos iniciadores para órdenes renombradas que permiten disimular la identidad de las funciones dependientes correspondientes,

15 y medios de restablecimiento 20 destinados a ser aplicados en la unidad 6 durante la fase de uso y que permiten encontrar la orden inicial, a partir de la orden renombrada, con el fin de encontrar la función dependiente a ejecutar.

20 Para la aplicación del principio de protección por renombramiento, también se construyeron medios de aprovechamiento que permiten transformar una unidad virgen 60 en una unidad 6 que pone al menos en marcha los medios de restablecimiento 20.

25 Para la aplicación del principio de protección por renombramiento, también se selecciona, en la fuente del software vulnerable 2vs:

por lo menos un procesamiento algorítmico que utiliza al menos un operando y que arroja al menos un resultado,

y por lo menos una porción de la fuente del software vulnerable

2vs, que contiene por lo menos un procesamiento algorítmico seleccionado.

La fuente del software vulnerable 2vs es entonces modificada, a manera de obtener la fuente del software protegido 2ps. Esta modificación es tal que especialmente:

5 durante la ejecución del software protegido 2p, por lo menos una porción de la primera parte de ejecución 2pes, que es ejecutada en el sistema de procesamiento de datos 3, toma en cuenta que la funcionalidad de por lo menos un procesamiento algorítmico seleccionado es ejecutada en la unidad 6,

10 durante la ejecución del software protegido 2p, la segunda parte de ejecución 2peu, que es ejecutada en la unidad 6, ejecuta por lo menos la funcionalidad de al menos un procesamiento algorítmico seleccionado,

 cada procesamiento algorítmico seleccionado es fragmentado de manera que durante la ejecución del software protegido 2p, cada
15 procesamiento algorítmico seleccionado sea ejecutado, por medio de la segunda parte de ejecución 2peu, utilizando funciones dependientes. De preferencia, cada procesamiento algorítmico seleccionado es fragmentado en funciones dependientes fd_n (con n variando de 1 a N), a saber:

 eventualmente una o más funciones dependientes que permitan
20 la disposición de uno de más operandos para la unidad 6,

 funciones dependientes en donde algunas utilizan el o los operandos y que en combinación, ejecutan la funcionalidad del procesamiento algorítmico seleccionado, utilizando este o estos operandos,

 y eventualmente una o más funciones dependientes que
25 permitan la disposición para la unidad 6, para el sistema de procesamiento de datos 3 del resultado del procesamiento algorítmico seleccionado,

 durante la ejecución del software protegido 2p, la segunda parte de ejecución 2peu ejecuta las funciones dependientes fd_n ,

durante la ejecución de software protegido 2p, las funciones dependientes son iniciadas por comandos iniciadores en órdenes renombradas,

y se selecciona un ordenamiento de estos comandos iniciadores entre el conjunto de ordenamientos que permiten la ejecución del software protegido 2p.

La primera parte de ejecución 2pes del software protegido 2p, ejecutada en el sistema de procesamiento de datos 3, ejecuta comandos iniciadores a órdenes renombradas transfiriendo hacia la unidad 6 órdenes renombradas, y que inicia en la unidad 6, el restablecimiento a través de medios de restablecimiento 20, de órdenes, posteriormente la ejecución por medio de la segunda parte de ejecución 2peu, de cada una de las funciones anteriores fd_n previamente definidas.

En otros términos, el principio de protección por renombramiento consiste en renombrar las órdenes de comandos iniciadores, a manera de obtener comandos iniciadores en órdenes renombradas en donde la ejecución en el sistema de procesamiento de datos 3, inicia en la unidad 6, la ejecución de funciones dependientes que habrán sido iniciadas por los comandos iniciadores en órdenes no renombradas, no obstante, sin que el examen del software protegido 2p no permita determinar la identidad de funciones dependientes ejecutadas.

La figura 6a ilustra un ejemplo de ejecución de un software vulnerable 2v. En este ejemplo, aparece en el transcurso de la ejecución del software vulnerable 2v en el sistema de procesamiento de datos 3, en un momento dado, el cálculo de $Z \leftarrow F(X, Y)$ que corresponde a la asignación a una variable Z del resultado de un procesamiento algorítmico representado por una función F y que utiliza los operandos X y Y.

Las figuras 6b y 6c ilustran un ejemplo de aplicación de la

invención.

La figura 6b ilustra la aplicación parcial de la invención. En este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes del software protegido 2p y en presencia de

5 la unidad 6, aparece:

en los momentos t_1 , t_2 , la ejecución de los comandos iniciadores CD_1 , CD_2 que inician en la unidad 6, la ejecución por medio de la segunda parte de ejecución 2peu, de funciones dependientes fd_1 , fd_2 correspondientes que aseguran la transferencia de datos X , Y desde el sistema de procesamiento de datos 3 hacia zonas de memoria respectivamente x , y situadas en los medios de memoria 15 de la unidad 6, estos comandos iniciadores CD_1 , CD_2 están representados respectivamente por $OUT(x, X)$, $OUT(y, Y)$,

en los momentos t_3 a t_{N-1} , la ejecución de comandos iniciadores CD_3 a CD_{N-1} , que inician en la unidad 6, la ejecución por medio de la segunda parte de ejecución 2peu, de funciones dependientes fd_3 a fd_{N-1} correspondientes, estos comandos iniciadores CD_3 a CD_{N-1} están representados, respectivamente, por $TRIG(fd_3)$ a $TRIG(fd_{N-1})$. El desarrollo de las funciones dependientes fd_3 a fd_{N-1} ejecutadas en combinación es algorítmicamente equivalente a la función F . De manera más precisa, la ejecución de estos comandos iniciadores conduce a la ejecución en la unidad 6, de funciones dependientes fd_3 a fd_{N-1} que se sirven del contenido de las zonas de memoria x , y y arrojan el resultado en una zona de memoria z de la unidad 6,

25 y en el momento t_N , la ejecución de un comando iniciador CD_N que inicia en la unidad 6, la ejecución por medio de la segunda parte de ejecución 2peu, de la función dependiente fd_N que asegura la transferencia del resultado del procesamiento algorítmico contenido en la zona de memoria

z de la unidad 6 hacia el sistema de procesamiento de datos 3, con el fin de asignarlo a la variable Z, este comando es representado por IN(z).

En este ejemplo, para aplicar completamente la invención, se selecciona como orden, el primer argumento de los comandos iniciadores OUT y el argumento de los comandos iniciadores TRIG e IN. Las órdenes así seleccionadas son renombradas por el método de renombramiento de órdenes. De esta manera, las órdenes de comandos iniciadores CD_1 a CD_N a saber x, y, fd_3 , fd_{N-1} , z son renombradas a manera de obtener respectivamente $R(x)$, $R(y)$, $R(fd_3)$..., $R(fd_{N-1})$, $R(z)$.

La figura 6c ilustra la aplicación completa de la invención. En este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3, de la primera parte de ejecución 2pes del software protegido 2p, y en presencia de la unidad 6, aparece:

en los momentos t_1 , t_2 , la ejecución de comandos iniciadores en órdenes renombradas $CDCR_1$, $CDCR_2$ que transfieren hacia la unidad 6, las órdenes renombradas $R(x)$, $R(y)$ así como los datos X, Y inician en la unidad 6 el restablecimiento por medio de los medios de restablecimiento 20, de órdenes renombradas para reestablecer las órdenes, a saber, la identidad de zonas de memoria x, y, desde la ejecución por medio de la segunda parte de ejecución 2peu, de funciones dependientes fd_1 , fd_2 correspondientes que aseguran la transferencia de datos X, Y desde el sistema de procesamiento de datos 3 hacia las zonas de memoria respectivamente x, y situadas en los medios de memoria 15 de la unidad 6, estos comandos iniciadores en órdenes renombradas $CDCR_1$, $CDCR_2$ están representados respectivamente por OUT ($R(x)$, X), OUT ($R(y)$, Y),

en los momentos t_3 a t_{N-1} , la ejecución de comandos iniciadores en órdenes renombradas $CDCR_3$ a $CDCR_{N-1}$, que transfieren hacia la unidad 6, las órdenes renombradas $R(fd_3)$ a $R(fd_{N-1})$, iniciando en la unidad 6 el

restablecimiento por medio de medios de restablecimiento 20, de órdenes, a saber, fd_3 a fd_{N-1} , desde la ejecución por medio de la segunda parte de ejecución 2peu, de funciones dependientes fd_3 a fd_{N-1} , estos comandos iniciadores en órdenes renombradas $CDCR_3$ a $CDCR_{N-1}$ son representados
5 respectivamente por $TRIG(R(fd_3))$ a $TRIG(R(fd_{N-1}))$.

 y en el momento t_N , la ejecución del comando iniciador en orden renombrada $CDCR_N$ que transfiere hacia la unidad 6, la orden renombrada $R(z)$ iniciando en la unidad 6 el restablecimiento por medio de medios de restablecimiento 20, de la orden, a saber, la identidad de la zona de memoria
10 z , desde la ejecución por medio de la segunda parte de ejecución 2peu, de la función dependiente fd_N que asegura la transferencia del resultado del procesamiento algorítmico contenido en la zona de memoria z de la unidad 6 hacia el sistema de procesamiento de datos 3 con el fin de asignarlo a la variable Z , este comando iniciador en orden renombrada $CDCR_N$ es
15 representado por $IN(R(z))$.

 En el ejemplo ilustrado, los comandos iniciadores en órdenes renombradas 1 a N son ejecutados sucesivamente. Cabe señalar que se pueden aportar dos mejoras:

 La primera mejora se refiere al caso en donde varios
20 procesamientos algorítmicos son deportados a la unidad 6 y por lo menos el resultado de un procesamiento algorítmico es utilizado por otro procesamiento algorítmico. En este caso, ciertos comandos iniciadores en órdenes renombradas que sirven para la transferencia, pueden ser eventualmente suprimidos.

25 La segunda mejora contempla en optar por un ordenamiento pertinente de los comandos iniciadores en órdenes renombradas entre el conjunto de ordenamientos que permiten la ejecución del software protegido 2p. Al respecto, es preferible seleccionar un ordenamiento de comandos

iniciadores en órdenes renombradas que disocie temporalmente la ejecución de funciones dependientes, intercalando entre ellas porciones de código ejecutado en el sistema de procesamiento de datos 3 y que comprende o no comandos iniciadores en órdenes renombradas que sirven para la determinación de otros datos. Las figuras 6d y 6e ilustran el principio de dicha modalidad.

La figura 6d muestra un ejemplo de ejecución de un software vulnerable 2v. En este ejemplo, aparece, en el transcurso de la ejecución del software vulnerable 2v, en el sistema de procesamiento de datos 3, la ejecución de dos procesamientos algorítmicos que conducen a la determinación de Z y Z', tales como $Z \leftarrow F(X, Y)$ y $Z' \leftarrow F'(X', Y')$.

La figura 6e ilustra un ejemplo de aplicación del método de acuerdo con la invención para el cual los dos procesamientos algorítmicos seleccionados en la figura 6d son deportados a la unidad 6. De acuerdo con dicho ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes del software protegido 2p y en presencia de la unidad 6, aparece, como se explicó anteriormente, la ejecución de comandos iniciadores en órdenes renombradas CDCR, a CDCR_N que corresponden a la determinación de Z y la ejecución de comandos iniciadores en órdenes renombradas CDCR', a CDCR'_M que corresponden a la determinación de Z'. Como se ilustra, los comandos iniciadores en órdenes renombradas CDCR₁ a CDCR_N no son ejecutados consecutivamente, en la medición en que los comandos iniciadores en órdenes renombradas CDCR', a CDCR'_M así como otras porciones de códigos son intercalados. En el ejemplo, también se modaliza el ordenamiento siguiente: CDCR₁, porción de código intercalado, CDCR', a CDCR₂, porción de código intercalado, CDCR'₂, CDCR'₃, porción de código intercalado, CDCR'₄, CDCR₃, CDCR₄, ..., CDCR_N, CDCR'_M.

Cabe señalar que durante la ejecución de una porción de la primera parte de ejecución 2pes del software protegido 2p, los comandos iniciadores en órdenes renombradas ejecutadas en el sistema de procesamiento de datos 3, inician en la unidad 6 el restablecimiento de la identidad de funciones dependientes correspondientes desde la ejecución de las mismas. Así, resulta que en presencia de la unidad 6, esta porción es ejecutada correctamente y que en consecuencia, el software protegido 2p es completamente funcional.

La figura 6f ilustra un ejemplo de intento de ejecución del software protegido 2p, mientras que la unidad 6 está ausente. En este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes del software protegido 2p, en todo momento, la ejecución de un comando iniciador en orden renombrada no puede iniciar el restablecimiento de la orden ni la ejecución de la función dependiente correspondiente, en razón de la ausencia de la unidad 6. El valor a asignar a la variable Z no puede ser entonces determinado correctamente.

Resulta entonces, que en ausencia de la unidad 6, por lo menos una petición de una porción de la primera parte de ejecución 2pes del software protegido 2p, de iniciar el restablecimiento de una orden y la ejecución de una función dependiente en la unidad 6 no puede ser satisfecha correctamente, de modo que por lo menos esta porción no es ejecutada correctamente y que en consecuencia, el software protegido 2p no es completamente funcional.

Gracias a este principio de protección por renombramiento, el examen en el software protegido 2p de los comandos iniciadores en órdenes renombradas, no permite determinar la identidad de funciones dependientes antes de ser ejecutadas en la unidad 6. Cabe señalar que el renombramiento de orden se realiza durante la modificación del software vulnerable 2v en un

software protegido 2p.

De acuerdo con una variante del principio de protección por renombramiento, se define por lo menos una función dependiente, una familia de funciones dependientes algorítmicamente equivalentes pero iniciadas por

5 comandos iniciadores en órdenes renombradas diferentes. De acuerdo con esta variante, para al menos un procesamiento algorítmico que utiliza funciones dependientes, este procesamiento algorítmico es fragmentado en funciones dependientes que para al menos una de ellas es reemplazada por una función dependiente de la misma familia en lugar de conservar varias

10 ocurrencias de la misma función dependiente. Para este efecto, los comandos iniciadores en órdenes renombradas son modificados para tener en cuenta el reemplazo de funciones dependientes por funciones dependientes de la misma familia. En otros términos, dos funciones dependientes de la misma familia tienen órdenes diferentes y en consecuencia, comandos iniciadores en

15 órdenes renombradas diferentes y, no es posible, en el examen del software protegido 2p, revelar que las funciones dependientes llamadas son algorítmicamente equivalentes.

De acuerdo con una modalidad preferida de la variante del principio de protección por renombramiento, se define para al menos una

20 función dependiente, una familia de funciones dependientes algorítmicamente equivalente, concatenando un campo de ruido a la información que define la parte funcional de la función dependiente para ejecutar en la unidad 6.

De acuerdo con una segunda modalidad preferida de la variante del principio de protección por renombramiento, se define para al menos una

25 función dependiente, una familia de funciones dependientes algorítmicamente equivalentes utilizado campos de identificación.

De acuerdo con una variante preferida de modalidad del principio de protección por renombramiento, se define como método de

renombramiento de órdenes, un método de codificación que permite codificar las órdenes para transformarlas en órdenes renombradas. Cabe recordar que el renombramiento de orden se realiza en la fase de protección P. Para esta variante preferida, los medios de restablecimiento 20 son medios que aplican
5 un método de decodificación que permite decodificar las órdenes renombradas y de reestablecer así la identidad de funciones dependientes para ejecutar en la unidad 6. Estos medios de restablecimientos son aplicados en la unidad 6 y pueden ser de naturaleza de software o material. Estos medios de restablecimiento 20 son solicitados en la fase de uso U cada vez
10 que un comando iniciador en orden renombrada es ejecutado en el sistema de procesamiento de datos 3 con el objetivo de iniciar en la unidad 6 la ejecución de una función dependiente.

De acuerdo con otra característica ventajosa de la invención, el método de protección contempla aplicar un principio de protección llamado
15 por "derivación condicional" cuya descripción se realiza con relación a las figuras 7a a 7c.

Para la aplicación del principio de protección por derivación condicional, se selecciona en la fuente del software vulnerable 2vs, por lo menos una derivación condicional BC. También se selecciona por lo menos
20 una porción de la fuente del software vulnerable 2vs que contiene al menos una derivación condicional BC seleccionada.

Por lo menos una porción seleccionada de la fuente del software vulnerable 2vs es entonces modificada, a manera de obtener la fuente del software protegido 2ps. Esta modificación es tal que especialmente durante la
25 ejecución del software protegido 2p:

por lo menos una porción de la primera parte de ejecución 2pes, que es ejecutada en el sistema de procesamiento de datos 3, toma en cuenta que la funcionalidad de por lo menos una derivación condicional BC

seleccionada es ejecutada en la unidad 6,

y la segunda parte de ejecución 2peu, que es ejecutada en la unidad 6, ejecuta al menos la funcionalidad de por lo menos una derivación condicional BC seleccionada y pone a disposición del sistema de procesamiento de datos 3, una información que permite a la primera parte de ejecución 2pes, continuar su ejecución en el entorno seleccionado.

La primera parte de ejecución 2pes del software protegido 2p, ejecutada en el sistema de procesamiento de datos 3, ejecuta comandos de derivaciones condicionales, que inician en la unidad 6, la ejecución por medio de la segunda parte de ejecución 2peu, de derivaciones condicionales deportadas bc cuya funcionalidad equivale a la funcionalidad de derivaciones condicionales BC elegidas.

La figura 7a ilustra un ejemplo de ejecución de un software vulnerable 2v. En este ejemplo, aparece, en el transcurso de la ejecución del software vulnerable 2v en el sistema de procesamiento de datos 3 en un momento determinado, una derivación condicional BC que indica al software vulnerable 2v el entorno en dónde continuar su desarrollo, a saber, uno de tres entornos posibles B_1 , B_2 o B_3 . Se debe entender que la derivación condicional BC toma la decisión de continuar la ejecución del software en el entorno B_1 , B_2 o B_3 .

La figura 7b ilustra un ejemplo de aplicación de la invención para el cual la derivación condicional seleccionada para ser deportada a la unidad 6, corresponde a la derivación condicional BC. En este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes del software protegido 2p y en presencia de la unidad 6, aparece:

en el momento t_1 , la ejecución del comando de derivación condicional CBC, que inicia en la unidad 6, la ejecución por medio de la

segunda parte de ejecución 2peu, de la derivación condicional deportada bc algrítmicamente equivalente a la derivación condicional BC, este comando de derivación condicional CBC, es representado por TRIG(bc),

5 en el momento t_2 , la transferencia de la unidad 6 hacia el sistema de procesamiento de datos 3, de la información que permite a la primera parte de ejecución 2pes, continuar su ejecución en el entorno seleccionado, a saber el entorno B_1 , B_2 o B_3 .

Cabe señalar que durante la ejecución de una porción de la primera parte de ejecución 2pes del software protegido 2p, los comandos de derivaciones condicionales ejecutados en el sistema de procesamiento de datos 3 inician la ejecución de derivaciones condicionales deportadas correspondientes en la unidad 6. Así, resulta que en presencia de la unidad 6, esta porción es ejecutada correctamente y que en consecuencia, el software protegido 2p es completamente funcional.

15 La figura 7c ilustra un intento de ejecución del software protegido 2p, mientras la unidad 6 está ausente. En este ejemplo, durante la ejecución en el sistema de procesamiento de datos 3 de la primera parte de ejecución 2pes del software protegido 2p:

20 en el momento t_1 , la ejecución del comando de derivación condicional CBC₁, no puede iniciar la ejecución de la derivación condicional deportada bc, teniendo en cuenta la ausencia de la unidad 6,

y en el momento t_2 , la transferencia de información que permite a la primera parte de ejecución 2pes, continuar en el entorno seleccionado fracasa teniendo en cuenta la ausencia de la unidad 6.

25 Resulta entonces que en ausencia de la unidad 6, por lo menos una petición de una porción de la primera parte de ejecución 2pes de iniciar la ejecución de una derivación condicional deportada a la unidad 6, no puede ser satisfecha correctamente, de modo que por lo menos esta porción no es

ejecutada correctamente y en consecuencia, el software protegido 2p no es completamente funcional.

En la descripción anterior con relación a las figuras 7a a 7c, el objeto de la invención contempla deportar a la unidad 6, una derivación
5 condicional. Desde luego, una modalidad preferida de la invención puede consistir en deportar a la unidad 6, una serie de derivaciones condicionales cuya funcionalidad global es equivalente al conjunto de funcionalidades de derivaciones condicionales que han sido deportadas. La ejecución de la
10 conduce a la puesta a disposición, para el sistema de procesamiento de datos 3, de una información que permite a la primera parte de ejecución 2pes del software protegido 2p, continuar su ejecución en el entorno seleccionado.

En la descripción precedente en relación a las figuras 4a a 7c, se explican cuatro principios diferentes de protección de un software de
15 manera generalmente independiente unos de otros. El procedimiento de protección de conformidad con la invención se pone en marcha al utilizar el principio de protección por detección y coerción, relacionado eventualmente a uno o varios principios de protección. En el caso en el que el principio de protección por detección y coerción se complete mediante la puesta en
20 marcha de al menos otro principio de protección, se completa convenientemente el principio de protección por detección y coerción por el principio de protección por variable y/o el principio de protección por renombramiento y/o el principio de protección por derivación condicional.

Y cuando se pone en marcha le principio de protección por
25 renombramiento, éste puede ser completado a su vuelta por el principio de protección por derivación condicional.

En la modalidad preferida, se completa el principio de protección por detección y coerción mediante el principio de protección por variable y por

el principio de protección por renombramiento, y se completa con el principio de protección por derivación condicional.

En el caso de que se aplique un principio de protección, como complemento del principio de protección por detección y coerción, tal descripción efectuada anteriormente debe comprender, para tener conciencia de dicha puesta en marcha combinada, las siguientes modificaciones:

la noción de software vulnerable debe ser comprendida como software vulnerable frente al principio de protección durante la descripción. Así, en el caso en donde un principio de protección ya ha sido aplicado al software vulnerable, la expresión "software vulnerable" debe ser interpretada por el lector como la expresión "software protegido por el o los principios de protección ya aplicados";

la noción de software protegido debe ser comprendida como software protegido frente al principio de protección durante la descripción. Así, en el caso en donde un principio de protección ya ha sido aplicado, la expresión "software protegido" debe ser interpretada por el lector como la expresión "nueva versión del software protegido";

y la o las selecciones efectuadas para la aplicación del principio de protección durante la descripción debe tener en cuenta la o las selecciones efectuadas para la aplicación del o de los principios de protección ya aplicados.

El desarrollo de la descripción permite comprender mejor la aplicación del método de protección de acuerdo con la invención. Este método de protección de acuerdo con la invención incluye, como aparece de manera más precisa en la figura 8:

primero, una fase de protección P en el transcurso de la cual un software vulnerable 2v es modificado en un software protegido 2p,

posteriormente, una fase de uso U en el transcurso de la cual se

aplica el software protegido 2p. En esta fase de uso U:

5 en presencia de la unidad 6 y cada vez que lo imponga una porción de la primera parte de ejecución 2pes ejecutada en el sistema de procesamiento de datos 3, se ejecuta una funcionalidad impuesta en la unidad 6, de modo que esta porción es ejecutada correctamente y que en consecuencia, el software protegido 2p es completamente funcional,

10 en ausencia de la unidad 6 y a pesar de la petición de una porción de la primera parte de ejecución 2pes de ejecutar una funcionalidad en la unidad 6, esta petición no puede ser correctamente satisfecha, de modo que por lo menos esta porción no es ejecutada correctamente y en consecuencia, el software protegido 2p no es completamente funcional,

15 y eventualmente una fase de recarga R en el transcurso de la cual se acredita al menos un uso complementario de una funcionalidad protegida por la aplicación de la segunda variante preferida de modalidad del principio de protección por detección y coerción que utilizan como característica, una variable de medición.

20 La fase de protección P puede ser fragmentada en dos subfases de protección P_1 y P_2 . La primera, llamada subfase de protección corriente arriba P_1 , es aplicada independientemente del software vulnerable 2b a proteger. La segunda, llamada subfase de protección corriente abajo P_2 es dependiente del software vulnerable 2v a proteger. Cabe señalar que las subfases de protección corriente arriba P_1 y corriente abajo P_2 pueden ser modalizadas ventajosamente por dos personas diferentes o dos equipos diferentes. Por ejemplo, la subfase de protección corriente arriba P_1 puede ser
25 modalizada por una persona o una sociedad que asegure el desarrollo de sistemas de protección de software, mientras que la subfase de protección corriente abajo P_2 puede ser modalizada por una persona o una sociedad que asegure el desarrollo de softwares antes de ser protegidos. Desde luego, es

claro que las subfases de protección corriente arriba P_1 y corriente abajo P_2 pueden también ser modalizadas por la misma persona o el mismo equipo.

La subfase de protección corriente arriba P_1 incluye varios periodos $S_{11}, \dots, S_{1n}$ para cada uno de los cuales se van a efectuar diferentes tareas o trabajos.

En primer período de esta subfase de protección corriente arriba P_1 es llamada "período de definiciones S_{11} ". Durante este período de definiciones S_{11} :

se selecciona:

– el tipo de la unidad 6. Como ejemplo, se puede elegir en cuanto a la unidad 6, un lector 8 de tarjetas y la tarjeta virgen 7 relacionada con el lector,

- y los medios para transferir 12,13 destinados a ser puestos en marcha respectivamente en el sistema de procesamiento de datos 3 y en la unidad 6, en el transcurso de la fase de utilización U, y que están capacitados para asegurar la transferencia de datos entre el sistema de procesamiento de datos y la unidad 6,

• está definida:

- al menos una característica de ejecución del software susceptible de ser monitoreada al menos en parte en una unidad 6;

- al menos un criterio a acatar por al menos una característica de ejecución de software;

- medios de detección 17 a ser ejecutados en una unidad 6 y que permiten detectar que al menos una característica de ejecución del software no acata al menos un criterio relacionado; y

- medios de coerción 18 a ser ejecutados por una unidad y que permiten informar al sistema de procesamiento de datos 3 y/o modificar la ejecución de un software cuando no se acata por lo menos un criterio;

y en el caso en donde el método de protección de acuerdo con la invención aplique el principio de protección por detección y coerción utilizando como característica una variable de medición de la ejecución del software, también se define:

5 como característica de ejecución de software susceptible de ser vigilada, una variable de medición del uso de una funcionalidad de un software,

 como criterio a obedecer, por lo menos un umbral asociado a cada variable de medición,

10 y medios de actualización que permiten actualizar al menos una variable de medición,

 y en el caso en donde el método de protección de acuerdo con la invención aplique una primera variante preferida de modalidad del principio de protección por detección y coerción utilizando como característica una
15 variable de medición de la ejecución del software, también se define:

 para al menos una variable de medición, varios umbrales asociados,

 y medios de coerción diferentes que corresponden a cada uno de estos umbrales,

20 y en el caso en donde el método de protección de acuerdo con la invención aplica una segunda variante preferida de modalidad del principio de protección por detección y coerción utilizando como característica una variable de medición de la ejecución del software, también se definen medios de recarga que permiten acreditar por lo menos un uso complementario para
25 al menos una funcionalidad de software vigilada por una variable de medición,

 y en el caso en donde el método de protección de acuerdo con la invención aplique el principio de protección por detección y coerción utilizando como característica un perfil de uso de software, también se define:

como característica de ejecución de software susceptible de ser vigilada, un perfil de uso de software,

y como criterio a obedecer, por lo menos una característica de ejecución de software,

5 y en el caso en donde el método de protección de acuerdo con la invención aplique el principio de protección por detección y coerción utilizando como característica de ejecución a obedecer, la vigilancia del encadenamiento de ejecución, también se define:

un juego de instrucciones en donde las instrucciones son
10 susceptibles de ser ejecutadas en la unidad 6,

un juego de comandos de instrucciones para este juego de instrucciones, estos comandos de instrucción son susceptibles de ser ejecutados en el sistema de procesamiento de datos 3 y de iniciar en la unidad 6 la ejecución de las instrucciones,

15 como perfil de uso, el encadenamiento de las instrucciones, como característica de ejecución, un encadenamiento deseado para la ejecución de instrucciones,

como medios de detección 17, medios que permitan detectar que el encadenamiento de instrucciones no corresponde a aquel deseado,

20 y como medios de coerción 18, medios que permitan informar al sistema de procesamiento de datos 3 y/o de modificar el funcionamiento de la porción de software protegido 2p cuando el encadenamiento de las instrucciones no corresponde a aquel deseado,

y en el caso en donde el método de protección de acuerdo con
25 la invención aplique una variante preferida de modalidad del principio de protección por detección y coerción utilizando como característica de ejecución a obedecer, la vigilancia del encadenamiento de ejecución, también se define:

como juego de instrucciones, un juego de instrucciones en donde por lo menos ciertas instrucciones trabajan sobre registros y utilizan por lo menos un operando con el fin de arrojar un resultado,

por lo menos una parte de instrucciones que trabajan sobre
5 registros:

una parte PF que define la funcionalidad de la instrucción,
y una parte que define el encadenamiento deseado para la ejecución de instrucciones y que comprende campos de bits que corresponden a:

10 un campo de identificación de la instrucción CII,
y para cada operando de la instrucción:
un campo bandera CD_k ,
y un campo de identificación previsto CIP_k del operando,
para cada registro que pertenece a los medios de
15 aprovechamiento y utilizado por el juego de instrucciones, un campo de identificación generada CIG_k , en el cual se memoriza automáticamente la identificación de la última instrucción que haya arrojado su resultado en este registro,

como medios de detección 17, medios que permiten, durante la
20 ejecución de una instrucción, para cada operando, cuando lo imponga el campo bandera CD_k , controlar la igualdad entre el campo de identificación generada CIG_k , que corresponde al registro utilizado por este operando, y el campo de identificación prevista CIP_k del origen de este operando,

y como medios de coerción 18, medios que permiten modificar el
25 resultado de las instrucciones, si por lo menos una de las igualdades controladas es falsa.

y en el caso en donde el método de protección de acuerdo con la invención aplique el principio de protección por renombramiento, también

se define:

- en cuanto a un comando iniciador, un comando de instrucción,
- en cuanto a una función dependiente, una instrucción,

como una orden, por lo menos un argumento para un comando
 5 iniciador, que corresponde por lo menos en parte a la información transmitida
 por el sistema de procesamiento de datos 3 a la unidad 6, con el fin de iniciar
 la ejecución de la función dependiente correspondiente,

un método de renombramiento de órdenes que permite
 renombrar las órdenes con el fin de obtener comandos iniciadores en órdenes
 10 renombradas,

y medios de restablecimiento 20 destinados a ser aplicados en
 la unidad 6 en el transcurso de la fase de uso U, y que permiten encontrar la
 función dependiente a ejecutar, a partir de la orden renombrada,

y en el caso en donde el método de protección de acuerdo con
 15 la invención aplica una variante del principio de protección por
 renombramiento, también se define para al menos una función dependiente,
 una familia de funciones dependientes algorítmicamente equivalentes, pero
 iniciadas por comandos iniciadores cuyas órdenes renombradas son
 diferentes,

20 y en el caso en donde el método de protección de acuerdo con
 la invención aplica una u otra de las modalidades preferidas de la variante del
 principio de protección por renombramiento, también se define para al menos
 una función dependiente, una familia de funciones dependientes
 algorítmicamente equivalentes:

25 concatenando un campo de ruido con la información que define
 la parte funcional de la función dependiente a ejecutar en la unidad 6,

o utilizando el campo de identificación de la instrucción CII y los
 campos de identificación prevista CIP_x de los operandos,

y en el caso en donde el método de protección de acuerdo con la invención aplica una variante preferida del principio de protección por renombramiento, también se define:

como método de renombramiento de órdenes, un método de
5 codificación para codificar las órdenes,

y como medios de restablecimiento 20, medios que aplican un método de decodificación para decodificar las órdenes renombradas y reestablecer así la identidad de funciones dependientes para ejecutar en la unidad 6.

10 Después de la subfase de protección corriente arriba, a la fase de definición S_{11} le sigue una fase llamada "fase de construcción S_{12} ". En dicho estado S_{12} se construyen los medios de transferencia 12,13 y los medios de aprovechamiento que corresponden a las definiciones de la fase de definición S_{11} .

15 Después de esta fase de construcción S_{12} , entonces se procede:

- A la construcción de medios de transferencia 12,13 que permitan, en el transcurso de la fase de utilización U, la transferencia de datos entre el sistema de procesamiento de datos 3 y la unidad 6,
- 20 • A la construcción:
 - medios de aprovechamiento que permitan a la unidad 6, en el transcurso de la fase de utilización U, de poner en marcha los medios de detección 17 y los medios de coerción 18,
 - y eventualmente los medios de aprovechamiento permiten a la
 - 25 unidad 6, en el transcurso de la fase de utilización U, poner también en marcha los medios de actualización,

- y eventualmente los medios de aprovechamiento permiten a la unidad 6, en el transcurso de la fase de nueva carga, poner también en marcha los medios de nueva carga,

- y eventualmente los medios de aprovechamiento permiten también a la unidad 6, en el transcurso de la fase de utilización U, ejecutar las instrucciones de la serie de instrucciones,

• Y cuando se pone en marcha el principio de protección por renombramiento, a la construcción de medios de aprovechamiento que permiten a la unidad 6, en el transcurso de la fase reutilización U, de poner en marcha también los medios de restablecimiento.

La construcción de medios de aprovechamiento se realiza de manera habitual, a través de una unidad de desarrollo de programa que toma en cuenta las definiciones incluidas en el período de definiciones S_{11} . Dicha unidad es descrita en el desarrollo de la descripción de la figura 9.

Durante la subfase de protección corriente arriba P_1 , el período de construcción S_{12} puede ser seguido por un período llamado "período de prepersonalización S_{13} ". Durante este período de prepersonalización S_{13} , por lo menos una parte de los medios de transferencia 13 y/o los medios de aprovechamiento son cargados en por lo menos una unidad virgen 60 con el fin de obtener por lo menos una unidad prepersonalizada 66. Cabe señalar que una parte de los medios de aprovechamiento, una vez transferida en una unidad prepersonalizada 66, ya no es directamente accesible desde el exterior de esta unidad prepersonalizada 66. La transferencia de medios de aprovechamiento en una unidad virgen 60 puede ser realizada a través de una unidad de prepersonalización adaptada, que se describe en el desarrollo de la descripción de la figura 10. En el caso de una unidad prepersonalizada 66, constituida por una tarjeta de chip 7 y de su lector 8, la prepersonalización sólo se refiere a la tarjeta de chip 7.

Durante la subfase de protección corriente arriba P_1 , se puede aplicar, después del período de definición S_{11} y, eventualmente después del período de construcción S_{12} , un período llamado "período de confección de herramientas S_{14} ". Durante este período de confección de herramientas S_{14} ,
 5 se confeccionan herramientas que permiten ayudar a la generación de softwares protegidos o de automatizar la protección de softwares. Dichas herramientas permiten:

ayudar a seleccionar o a seleccionar automáticamente en el software vulnerable $2v$ a proteger:

- 10 - las características de ejecución a ser monitoreadas y, eventualmente, los procesamientos algorítmicos susceptibles de ser desglosados en instrucciones deportables en la unidad 6,
- las porciones susceptibles de ser modificadas,
- y cuando también se pone en marcha el principio de
 15 protección por variable, las variables susceptibles de ser deportadas en la unidad 6,
- y cuando también se pone en marcha el principio de protección por renombramiento, los procesamientos algorítmicos susceptibles de ser desglosados en funciones dependientes deportables en la unidad 6 y
 20 por las cuales pueden ser renombradas las órdenes de comandos iniciadores,
- y cuando también se pone en marcha el principio de protección por derivación condicional, las derivaciones condicionales por las cuales la funcionalidad es susceptibles de ser deportada en la unidad 6,
- y eventualmente, ayudar a generar los software
 25 protegidos o de automatizar la protección de los software.

Estas herramientas diversas pueden ser realizadas independientemente o en combinación y cada herramienta puede tomar

diversas formas, como por ejemplo preprocesador, compilador, ensamblador, etc.

Le sigue a la subfase de protección corriente arriba P_1 una subfase de protección corriente abajo P_2 que depende del software vulnerable $2v$ a proteger. Estas subfase de protección corriente abajo hace intervenir igualmente varias fases. La primer fase que corresponde a la apuesta en marcha del principio de protección por detección y coerción se llama "fase de creación S_{21} ". En esta fase de creación S_{21} , se utiliza las opciones que intervienen en la fase de definición S_{11} . Con la ayuda de estas opciones y eventualmente con las herramientas construidas en la fase de fabricación de herramientas S_{14} , se crea el software protegido $2p$:

- al elegir al menos una característica de ejecución del software a ser monitoreada entre las características de ejecución susceptibles de ser monitoreadas
- elegir al menos un criterio a acatar por al menos una característica de ejecución de software elegido,
- elegir al menos un procesamiento algorítmico, el cual, cuando se ejecuta un software vulnerable $2v$, utiliza al menos un operando y permite obtener al menos un resultado, y por el cual se monitorea al menos una característica de ejecución de software elegido,
- elegir por lo menos una porción de una fuente de software vulnerable $2vs$ que contiene al menos un procesamiento algorítmico elegido,
- producir la fuente de software protegido $2ps$ a partir de la fuente de software vulnerable $2vs$, que modifica al menos una porción elegida de la fuente de software vulnerable $2vs$ para obtener al menos una porción modificada de una fuente de software protegido $2ps$; dicha modificación es tal que:

- cuando se ejecuta el software protegido 2p se ejecuta una primera parte de ejecución 2pes en el sistema de procesamiento de datos 3 y una segunda parte de ejecución 2peu se ejecuta en una unidad 6, que se obtiene a partir de la unidad virgen 60 después de cargar informaciones,

5 - la segunda parte de ejecución 2peu ejecuta al menos la funcionalidad de por lo menos un procesamiento algorítmico elegido,

 - y cuando se ejecuta el software protegido 2p por lo menos una característica de ejecución elegida es monitoreada por la segunda parte de ejecución 2peu, y no se acata un criterio que termina en una modificación de
10 la ejecución del software protegido (2p),

 • y producir: una primera parte objeto 2pos del software protegido 2p, a partir de la fuente del software protegido 2ps, dicha primera parte objeto 2pos es tal que cuando se ejecuta el software protegido 2p aparece una primera parte de ejecución 2pes que se ejecuta en el sistema de
15 procesamiento de datos 3 y en el cual al menos una porción detecta que al menos se monitorea una característica de ejecución de software,

 • y una segunda parte objeto 2pou del software protegido, que contiene los medios de aprovechamiento que ponen en marcha los medios de detección 17 y los medios de coerción 18, esta segunda parte
20 objeto es tal que, después de ser cargada en la unidad virgen 60 y cuando se ejecuta el software protegido 2p aparece la segunda parte de ejecución 2peu en medio de la cual al menos se monitorea una característica de ejecución de software, y en medio de la cual el no acatamiento de un criterio termina en una modificación de la ejecución del software protegido 2p.

25 En breve, el principio de protección por detección y coerción de la invención puede aplicarse directamente para el desarrollo de un software novedoso sin que requiere la realización previa de un software vulnerable 2v. De esta manera, se obtiene directamente un software protegido 2p.

Para la aplicación del principio de protección por detección y coerción que utiliza como característica una variable de medición de la ejecución del software, el software protegido 2p es modificado:

5 seleccionando como característica de ejecución de software a vigilar, por lo menos una variable de medición del uso de una funcionalidad de un software,

seleccionando:

por lo menos una funcionalidad del software protegido 2p cuyo uso es susceptible de ser vigilado gracias a una variable de medición,

10 por lo menos una variable de medición que sirve para cuantificar el uso de dicha funcionalidad,

por lo menos un umbral asociado con una variable de medición seleccionada que corresponde a un límite de uso de dicha funcionalidad,

15 y por lo menos un método de actualización de una variable de medición seleccionada en función del uso de dicha funcionalidad,

y modificando por lo menos una porción seleccionada de la fuente del software protegido 2ps, esta modificación es tal que durante la ejecución del software protegido 2p, la variable de medición es actualizada por medio de la segunda parte de ejecución 2peu, en función del uso de dicha
20 funcionalidad y se toma en cuenta por lo menos un exceso de umbral.

Para la aplicación de una primera variante preferida de modalidad del principio de protección por detección y coerción que utiliza como característica, una variable de medición, el software protegido 2p es modificado:

25 seleccionando en la fuente del software protegido 2ps, por lo menos una variable de medición seleccionada a la cual deben estar asociados varios umbrales correspondientes a los límites diferentes de uso de la funcionalidad,

seleccionando al menos dos umbrales asociados con la variable de medición seleccionada,

y modificando por lo menos una porción seleccionada de la fuente del software protegido 2ps, esta modificación es tal que durante la ejecución del software protegido 2p, se toman en cuenta los excesos de diversos umbrales, por medio de la segunda parte de ejecución 2peu, de manera diferente.

Para la aplicación de una segunda variante preferida de modalidad del principio de protección por detección y coerción que utiliza como característica, una variable de medición, el software protegido 2p es modificado:

seleccionando en la fuente del software protegido 2ps, por lo menos una variable de medición seleccionada que permite limitar el uso de una funcionalidad a la cual se debe poder acreditar por lo menos un uso complementario,

y modificando por lo menos una porción seleccionada, esta modificación es tal que en una fase llamada de recarga, por lo menos un uso complementario de por lo menos una funcionalidad que corresponde a una variable de medición seleccionada puede ser acreditada.

Para la aplicación del principio de protección por detección y coerción que utiliza como característica, un perfil de uso de software, el software protegido 2p es modificado:

seleccionando como característica de ejecución de software a vigilar por lo menos un perfil de uso de software,

seleccionando por lo menos una característica de ejecución que debe obedecer por lo menos un perfil de uso seleccionado,

y modificando al menos una porción seleccionada de la fuente del software protegido 2ps, esta modificación es tal que durante la ejecución

del software protegido 2p, la segunda parte de ejecución 2peu obedece todas las características de ejecución seleccionadas.

Para la aplicación del principio de protección por detección y coerción que utilizan como característica de ejecución a obedecer, la
5 vigilancia del encadenamiento de ejecución, el software protegido 2p es modificado:

al modificar al menos una porción elegida de la fuente del software protegido 2ps, siendo tal modificación tal que:

- al menos se fragmenta un proceso algorítmico de manera que
10 cuando se ejecuta el software protegido 2p, se ejecuta dicho proceso algorítmico en el medio de la segunda parte de ejecución 2peu utilizando instrucciones;

- para al menos un proceso algorítmico elegido, se integran los comandos de instrucciones en la fuente de software protegido 2ps de manera
15 que cuando se ejecuta el software protegido 2p se ejecuta cada comando de instrucciones en la primera parte de ejecución 2pes e inicia en la unidad 6 la ejecución de al menos la segunda parte de ejecución 2peu de una instrucción,

- se elige una orden de comandos de instrucciones entre el conjunto de órdenes que permiten la ejecución del software protegido 2p,

20 - y el enlace de que deben acatar por lo menos ciertas instrucciones cuando se especifica su ejecución en la unidad 6.

En la subfase de protección corriente abajo P_2 , y cuando se aplique al menos un principio de protección en más de un principio de protección por detección y coerción, se pone en marcha una "fase de
25 modificación S_{22} ". En esta fase de modificación S_{22} , se utilizan las definiciones que se utilizaron en la fase de definiciones S_{11} . Con la ayuda de estas definiciones y eventualmente con las herramientas generadas en la fase de fabricación de herramientas S_{14} , se modifica el software protegido 2p para

permitir la puesta en marcha de principios de protección en una de las disposiciones definidas anteriormente.

Cuando se pone en marcha el principio de protección por variable, se modifica el software protegido 2p:

- 5 • al menos una variable que se utilice en al menos un proceso algorítmico elegido, que cuando ejecuta el software protegido 2p define parcialmente el estado de software protegido 2p,
- al modificar al menos una porción elegida de la fuente del software protegido 2ps siendo esta modificación tal que cuando se ejecuta el
- 10 software protegido 2p, al menos una variable elegida o al menos una copia de la variable elegida reside dentro de la unidad 6,
- y que produce:
 - la primera parte objeto 2pos del software protegido 2p, esta primera parte objeto 2pos es tal que cuando se ejecuta el software protegido
 - 15 2p, al menos una porción de la primera parte de ejecución 2pes se hace cargo también al menos de una variable o al menos de una copia de variable que reside en la unidad 6,
 - y la segunda parte objeto 2pou del software protegido 2p, siendo segunda parte objeto 2pou tal que, después de cargar lo que la unidad
 - 20 6 y cuando se ejecuta el software protegido 2p, aparece la segunda parte de ejecución 2peu, en medio del cual al menos una variable elegida o al menos una copia de variable elegida reside también dentro de la unidad 6.

Cuando el principio de protección por renombramiento es aplicado, el software 2p es modificado:

- 25 seleccionando la fuente del software protegido 2ps, comandos iniciadores,
- modificando por lo menos una porción seleccionada de la fuente del software protegido 2ps renombrando las órdenes de comandos iniciadores

seleccionados, con el fin de disimular la identidad de funciones dependientes correspondientes,

y produciendo:

la primera parte objeto 2pos del software protegido 2p, esta
5 primera parte objeto 2pos es tal que durante la ejecución del software protegido 2p, se ejecutan los comandos iniciadores en órdenes renombradas,
y la segunda parte objeto 2pou del software protegido 2p que contiene los medios de aprovechamiento que aplican también los medios de reestablecimiento 20, esta segunda parte objeto 2pou es tal que después de
10 carga en la unidad 6 y durante la ejecución del software protegido 2p, la identidad de funciones dependientes cuya ejecución es iniciada por la primera parte de ejecución 2pes es restablecida por medio de la segunda parte de ejecución 2peu, y las funciones dependientes son ejecutadas por medio de la segunda parte de ejecución 2peu.

15 Para la aplicación de una variante del principio de protección por renombramiento, el software protegido 2p es modificado:

seleccionando en la fuente del software protegido 2ps por lo menos un comando iniciador en orden renombrada,

y modificando al menos una porción seleccionada de la fuente
20 de software protegido 2ps reemplazando al menos la orden renombrada de un comando iniciador en orden renombrada seleccionada, por otra orden renombrada, que inicia una función dependiente de la misma familia.

Cuando se aplica el principio de protección por derivación condicional, el software protegido 2p es modificado:

25 seleccionado en la fuente del software protegido 2ps, por lo menos una derivación condicional efectuada por lo menos en un procesamiento algorítmico seleccionado,

modificando la menos una porción seleccionada de la fuente del

software protegido 2ps; esta modificación es tal que durante la ejecución del software protegido 2p, se ejecuta la funcionalidad de por lo menos una derivación condicional seleccionada, por medio de la segunda parte de ejecución 2peu, en la unidad 6,

5 y produciendo:

la primera parte objeto 2pos del software protegido 2p, esta primera parte objeto 2pos es tal que durante la ejecución del software protegido 2p, se ejecuta la funcionalidad de por lo menos una derivación condicional seleccionada en la unidad 6,

10 y la segunda parte objeto 2pou del software protegido 2p, esta segunda parte objeto 2pou es tal que después de carga en la unidad 6 y durante la ejecución del software protegido 2p, aparece la segunda parte de ejecución 2peu por medio de la cual se ejecuta la funcionalidad de al menos una derivación condicional seleccionada.

15 Para la aplicación de una modalidad preferida del principio de protección por derivación condicional, el software protegido 2p es modificado:

seleccionando, en la fuente del software protegido 2ps por lo menos una serie de derivaciones condicionales seleccionadas,

modificando al menos una porción seleccionada de la fuente del
20 software protegido 2ps, esta modificación es tal que durante la ejecución del software protegido 2p, la funcionalidad global de al menos una serie seleccionada de derivaciones condicionales es ejecutada por medio de la segunda parte de ejecución 2peu, en la unidad 6,

y produciendo:

25 la primera parte objeto 2pos del software protegido 2p, esta primera parte objeto 2pos es tal que durante la ejecución del software protegido 2p, la funcionalidad de al menos una serie seleccionada de derivaciones condicionales es ejecutada en la unidad 6,

y la segunda parte objeto 2pou del software protegido 2p, esta segunda parte objeto 2pou es tal que después de carga en la unidad 6 y durante la ejecución del software protegido 2p, aparece la segunda parte de ejecución 2peu por medio de la cual se ejecuta la funcionalidad global de al menos una serie seleccionada de derivaciones condicionales.

Desde luego, los principios de protección de acuerdo con la invención pueden ser aplicados directamente durante el desarrollo de un nuevo software sin necesitar la realización previa de softwares protegidos intermediarios. De esta forma, los periodos de creación S_{21} y de modificación S_{22} pueden ser efectuados de manera concomitante con el fin de obtener directamente el software protegido 2p.

En la subfase de protección corriente abajo P_2 , se pone en marcha después de la fase de creación S_{21} del software protegido 2p, y eventualmente después de la fase de modificación S_{22} , una fase llamada "fase de personalización S_{23} ". En esta fase de personalización S_{23} , la segunda parte objeto 2pou que contiene los medios de aprovechamiento se carga en al menos una unidad virgen 60, para obtener al menos una unidad 6, o una parte de la segunda parte objeto 2pou que contiene los medios de aprovechamiento se carga en al menos una unidad pre-personalizada 66, para obtener al menos una unidad 6. La carga de estas informaciones de personalización permiten hacer operativo al menos una unidad 6. Debe observarse que una parte de estas informaciones, una vez que son transferidas a una unidad 6, ya no puede ser acezada directamente desde el exterior de esta unidad 6. La transferencia de informaciones de personalización en una unidad virgen 60 o en una unidad pre-personalizada 66 puede ser realizada mediante la intervención de una unidad de personalización adaptada que se describe después de la descripción a la

figura 13. En el caso de una unidad 6, que está constituida de una tarjeta de chip 7 y de su lector 8, la personalización solo concierne a la tarjeta de chip 7.

Para la aplicación de la fase de protección P, se describen diferentes medios técnicos de manera más precisa con relación a las figuras 9, 10, 11, 12 y 13.

La figura 9 ilustra un ejemplo de modalidad de un sistema 25 que permite aplicar el período de construcción S_{12} que toma en cuenta las definiciones contenidas en el período de definiciones S_{11} y en el transcurso del cual se construyen los medios de transferencia 12, 13 y eventualmente, los medios de aprovechamiento destinados a la unidad 6. Dicho sistema 25 comprende una unidad de desarrollo de programa o estación de trabajo y se presenta por lo regular bajo la forma de una computadora que comprende una unidad central, una pantalla, periféricos del tipo teclado-ratón, y que comprende, especialmente, los siguientes programas: editores de ficheros, ensambladores, preprocesadores, compiladores, interpretadores, depuradores y editores de enlaces.

La figura 10 ilustra un ejemplo de modalidad de una unidad de prepersonalización 30 que permite cargar al menos en parte los medios de transferencia 13 y/o los medios de aprovechamiento en por lo menos una unidad virgen 60 con el fin de obtener por lo menos una unidad prepersonalizada 66. Esta unidad de prepersonalización 30 comprende un medio de lectura y de escritura 31 que permite prepersonalizar de manera eléctrica, una unidad virgen 60, a manera de obtener una unidad prepersonalizada 66 en la cual los medios de transferencia 13 y/o de aprovechamiento han sido cargados. La unidad de prepersonalización 30 puede también comprender medios de personalización física 32 de la unidad virgen 60 que se pueden presentar por ejemplo, en forma de una impresora. En el caso en donde la unidad 6 esté constituida por una tarjeta de chip 7 y su

lector 8, la prepersonalización se refiere de manera general únicamente a la tarjeta de chip 7.

La figura 11 ilustra un ejemplo de modalidad de un sistema 35 que permite efectuar la confección de herramientas que permiten ayudar a
5 generar softwares protegidos o de automatizar la protección de softwares. Dicho sistema 35 comprende una unidad de desarrollo de programa o estación de trabajo y se presenta por lo regular en forma de una computadora que comprende una unidad central, una pantalla, periféricos del tipo teclado-ratón, y que comprenden, especialmente, los siguientes programas: editores
10 de ficheros, ensambladores, preprocesadores, compiladores, interpretadores, depuradores y editores de enlaces.

La figura 12 ilustra un ejemplo de modalidad de un sistema 40 que permite crear directamente un software protegido 2p o de modificar un software vulnerable 2v con el fin de obtener un software protegido 2p. Dicho
15 sistema 40 comprende una unidad de desarrollo de programa o estación de trabajo y se presenta por lo regular en forma de una computadora que comprende una unidad central, una pantalla, periféricos del tipo teclado-ratón, y que comprende, especialmente, los siguientes programas: editores de ficheros, ensambladores, preprocesadores, compiladores, interpretadores,
20 depuradores y editores de enlaces, así como herramientas que permiten ayudar a generar softwares protegidos o de automatizar la protección de softwares.

La figura 13 ilustra un ejemplo de modalidad de una unidad de personalización 45 que permite cargar la segunda parte objeto 2pou en por lo
25 menos una unidad virgen 60 con el fin de obtener por lo menos una unidad 6 o una parte de la segunda parte objeto 2pou en por lo menos una unidad prepersonalizada 66 con el fin de obtener por lo menos una unidad 6. Esta unidad de personalización 45 comprende un medio de lectura y de escritura

46 que permite personalizar de manera eléctrica, por lo menos una unidad
virgen 60 o al menos una unidad prepersonalizada 66, a manera de obtener al
menos una unidad 6. Al final de esta personalización, una unidad 6
comprende las informaciones necesarias para la ejecución del software
5 protegido 2p. La unidad de personalización 45 puede también comprender
medios de personalización física 47 para al menos una unidad 6 que se
pueden presentar por ejemplo, en forma de una impresora. En el caso en
donde una unidad 6 esté constituida por una tarjeta de chip 7 y su lector 8, la
personalización se refiere de manera general únicamente a la tarjeta de chip
10 7.

El método de protección de la invención puede ser aplicado con
las siguientes mejoras:

Se puede prever utilizar conjuntamente varias unidades de
procesamiento y de memoria en las cuales se reparte la segunda parte objeto
15 2pou del software protegido 2p de manera que su uso conjunto permite
ejecutar el software protegido 2p, la ausencia de por lo menos una de estas
unidades de procesamiento y de memoria impiden el uso del software
protegido 2p.

Igualmente, después del período de prepersonalización S_{13} y
20 durante el período de personalización S_{23} , la parte de la segunda parte objeto
2pou necesaria para transformar la unidad prepersonalizada 66 en una
unidad 6 puede estar contenida en una unidad de procesamiento y de
memoria utilizada por la unidad de personalización 45 con el fin de limitar el
acceso a esta parte de la segunda parte objeto 2pou. Desde luego, esta parte
25 de la segunda parte objeto 2pou puede ser repartida en varias unidades de
procesamiento y de memoria de manera que esta parte de la segunda parte
objeto 2pou, sea accesible únicamente durante el uso conjunto de estas
unidades de procesamiento y de memoria.

NUEVEDAD DE LA INVENCION

REIVINDICACIONES

- 5 1.- Un procedimiento para proteger, a partir de por lo menos una
unidad virgen (60) que comprende al menos un medio de memoria (15) y un
medio de procesamiento (16), un software vulnerable (2v) contra su uso no
autorizado; dicho software vulnerable (2v) funciona en un sistema de
procesamiento de datos (3), caracterizado porque consiste, en una fase de
10 protección (P): definir al menos una característica de ejecución del software
susceptible de ser monitoreada al menos en parte en una unidad (6); al
menos un criterio a obedecer por al menos una característica de ejecución de
software; medios de detección (17) a ser ejecutados en una unidad (6) y que
permiten detectar que al menos una característica de ejecución del software
15 no obedece al menos un criterio relacionado; y medios de coerción (18) a ser
ejecutados por una unidad (6) y que permiten informar al sistema de
procesamiento de datos (3) y/o modificar la ejecución de un software cuando
no se obedece por lo menos un criterio; construir medios de aprovechamiento
que permitan transformar la unidad virgen (60) en una unidad (6) capaz de
20 poner en marcha los medios de detección (17) y los medios de coerción (18);
crear un software protegido (2p): al seleccionar por lo menos una
característica de ejecución de software para ser monitoreado, entre las
características de ejecución susceptibles de ser monitoreadas, seleccionar
por lo menos un criterio a obedecer por al menos una característica de
25 ejecución de software que se haya seleccionado, seleccionar al menos un
procesamiento algorítmico, el cual, cuando se ejecuta un software vulnerable
(2v), utiliza al menos un operando y permite obtener al menos un resultado, y
por el cual se monitorea al menos una característica de ejecución de software

seleccionado, seleccionar por lo menos una porción de una fuente de software vulnerable (2vs) que contiene al menos un procesamiento algorítmico seleccionado, producir la fuente de software protegido (2ps) a partir de la fuente de software vulnerable (2vs), que modifica al menos una

5 porción seleccionada de la fuente de software vulnerable (2vs) para obtener al menos una porción modificada de una fuente de software protegido (2ps); dicha modificación es tal que: cuando se ejecuta el software protegido (2p) se ejecuta una primera parte de ejecución (2pes) en el sistema de procesamiento de datos (3) y una segunda parte de ejecución (2peu) se ejecuta en una

10 unidad (6), que se obtiene a partir de la unidad virgen (60) después de cargar informaciones, la segunda parte de ejecución (2peu) ejecuta al menos la funcionalidad de por lo menos un procesamiento algorítmico seleccionado, y cuando se ejecuta el software protegido (2p), por lo menos una característica de ejecución seleccionada es monitoreada por la segunda parte de ejecución

15 (2peu), y no se obedece un criterio que termina en una modificación de la ejecución del software protegido (2p); y producir: una primera parte objeto (2pos) del software protegido (2p), a partir de la fuente del software protegido (2ps), dicha primera parte objeto (2pos) es tal que cuando se ejecuta el software protegido (2p) aparece una primera parte de ejecución (2pes) que se

20 ejecuta en el sistema de procesamiento de datos (3) y en el cual al menos una porción detecta que al menos se monitorea una característica de ejecución de software, y una segunda parte objeto (2pou) del software protegido, que contiene los medios de aprovechamiento que ponen en marcha los medios de detección (17) y los medios de coerción (18), esta

25 segunda parte objeto es tal que, después de ser cargada en la unidad virgen (60) y cuando se ejecuta el software protegido (2p) aparece la segunda parte de ejecución (2peu) en medio de la cual al menos se monitorea una característica de ejecución de software, y en medio de la cual el no

98
97

obedecimiento de un criterio termina en una modificación de la ejecución del software protegido (2p), y cargar la segunda parte objeto (2pou) en la unidad virgen (60) en virtud de obtener la unidad (6); y en una fase de utilización (U) en el curso de la misma se ejecuta el software protegido (2p): en presencia de

5 la unidad (6): siempre que se obedezcan todos los criterios que corresponden a todas las características de ejecución monitoreadas de todas las porciones modificadas del software protegido (2p), se permite el funcionamiento nominal de estas porciones del software protegido (2p) y en consecuencia se permite el funcionamiento nominal del software protegido (2p), y si al menos no se

10 obedece uno de los criterios que corresponde a una característica de ejecución monitoreada de una porción del software protegido (2p), informar al sistema de procesamiento de datos (3) y/o modificar el funcionamiento de la porción de software protegido (2p), de manera que se modifica el funcionamiento del software protegido (2p); y en ausencia de la unidad (6) a

15 pesar de la petición de una porción de la primera parte de ejecución (2pes) de iniciar la ejecución en la unidad (6) de la funcionalidad de un procesamiento algorítmico seleccionado, no poder responder correctamente a esa petición, de manera que al menos esta porción no se ejecuta correctamente y que, en consecuencia, el software protegido (2p) no es completamente funcional.

20 2.- El procedimiento para limitar la utilización de un software protegido (2p) de conformidad con la reivindicación 1, caracterizado además porque consiste: en la fase de protección (P): en definir: como característica de ejecución de software susceptible de ser vigilada, una variable de medida del uso de una funcionalidad de un software, como criterio a obedecer, por lo

25 menos un umbral asociado con cada variable de medida, y medios de actualización que permiten actualizar por lo menos una variable de medida, en construir los medios de aprovechamiento que permiten a la unidad (6) aplicar también los medios de actualización, y en modificar el software

protegido (2p): seleccionando como característica de ejecución de software a vigilar, por lo menos una variable de medida del uso de una funcionalidad de un software, seleccionando: por lo menos una funcionalidad del software protegido (2p) en donde el uso es susceptible de ser vigilado gracias a una variable de medida, por lo menos una variable de medida que sirve para cuantificar el uso de dicha funcionalidad, por lo menos un umbral asociado con una variable de medida seleccionado que corresponde a un límite de uso de dicha funcionalidad, y por lo menos un método para actualizar una variable de medida seleccionada en función del uso de dicha funcionalidad, y modificando por lo menos una porción seleccionada de la fuente del software protegido (2ps), esta modificación siendo tal que, durante la ejecución del software protegido (2p), la variable de medida es actualizada por medio de la segunda parte de ejecución (2peu), en función del uso de dicha funcionalidad y por lo menos se toma en cuenta un exceso de umbral, y en la fase de uso (U), en presencia de la unidad (6), y en el caso en donde se detecte por lo menos un exceso de umbral correspondiente a por lo menos un límite de uso, en informar al sistema de procesamiento de datos (3) y/o modificar el funcionamiento de la porción del software protegido (2p), de modo que el funcionamiento de software protegido (2p) sea modificado.

3.- El procedimiento de conformidad con la reivindicación 2, caracterizado además porque consiste: en la fase de protección (P): en definir: para al menos una variable de medida, varios umbrales asociados, y medios de coerción diferentes que correspondan a cada uno de estos umbrales, y en modificar el software protegido (2p): seleccionando en la fuente del software protegido (2ps), por lo menos una variable de medida seleccionada a la cual deben estar asociados varios umbrales correspondientes a los límites diferentes de uso de la funcionalidad, seleccionando por lo menos dos umbrales asociados con la variable de

medida seleccionada, y modificando por lo menos una porción seleccionada de la fuente de software protegido (2ps), esta modificación siendo tal que, durante la ejecución del software protegido (2p), se tomen en cuenta los excesos de diversos umbrales, por medio de la segunda parte de ejecución (2peu), de manera diferente, y en la fase de uso (U): en presencia de la unidad (6): en el caso en donde se detecta el exceso de un primer umbral, en ordenar al software protegido (2p) de no utilizar más la funcionalidad correspondiente, y en el caso en donde se detecte el exceso de un segundo umbral, en hacer inoperante la funcionalidad correspondiente y/o por lo menos una porción del software protegido (2p).

4.- El procedimiento de conformidad con la reivindicación 2 ó 3, caracterizado además porque consiste: en la fase de protección (P): en definir medios de recarga que permitan acreditar por lo menos un uso complementario para al menos una funcionalidad de software vigilada por una variable de medida, en construir los medios de aprovechamiento que permitan también a la unidad (6) aplicar los medios de recarga, y en modificar el software protegido (2p): seleccionando la fuente de software protegido (2ps), por lo menos una variable de medida seleccionada que permita limitar al uso de una funcionalidad a la cual por lo menos un uso complementario debe poder ser acreditado, y modificando al menos una porción seleccionada, esta modificación siendo tal que en una fase llamada de recarga, por lo menos se puede acreditar un uso complementario de por lo menos una funcionalidad que corresponde a una variable de medida seleccionada, y en la fase de recarga: en reactualizar por lo menos una variable de medida seleccionada y/o al menos un umbral asociado, a manera de permitir por lo menos un uso complementario de la funcionalidad.

5.- El procedimiento de conformidad con la reivindicación 1, caracterizado además porque consiste: en la fase de protección (P): en

definir: como característica de ejecución de software susceptible de ser vigilada, un perfil de uso de software, y como criterio a obedecer, por lo menos una característica de ejecución de software, y en modificar el software protegido (2p): seleccionando como característica de ejecución de software a

5 vigilar por lo menos un perfil de uso de software, seleccionando por lo menos una característica de ejecución que debe obedecer por lo menos un perfil de uso seleccionado, y modificando al menos una porción seleccionada de la fuente de software protegido (2ps), esta modificación siendo tal que durante la ejecución del software protegido (2p), la segunda parte de ejecución (2peu)

10 obedece todas las características de ejecución seleccionadas, y en la fase de uso (U) en presencia de la unidad (6), y en el caso en donde se detecta que por lo menos una característica de ejecución no es obedecida, en informar al sistema de procesamiento de datos (3) y/o en modificar el funcionamiento de la porción del software protegido (2p), de modo que el funcionamiento del

15 software protegido (2p) sea modificado.

6.- El procedimiento de conformidad con la reivindicación 5, caracterizado además porque consiste: en la fase de protección (P): en definir: un juego de instrucciones en donde las instrucciones son susceptibles de ser ejecutadas en la unidad (6), un juego de comandos de instrucciones

20 para este juego de instrucciones, estos comandos de instrucciones son susceptibles de ser ejecutados en el sistema de procesamiento de datos (3) y de iniciar en la unidad (6) la ejecución de las instrucciones, como perfil de uso, el encadenamiento de las instrucciones, como característica de ejecución, un encadenamiento deseado para la ejecución de las

25 instrucciones, como medios de detección (17), medios que permitan detectar que el encadenamiento de las instrucciones no corresponde a aquel deseado, y como medios de coerción (18), medios que permitan informar al sistema de procesamiento de datos (3) y/o de modificar el funcionamiento de la porción

de software protegido (2p) cuando el encadenamiento de instrucciones no corresponda a aquel deseado, en construir los medios de aprovechamiento que permitan también a la unidad (6) ejecutar las instrucciones del juego de instrucciones, la ejecución de estas instrucciones son iniciadas por la
5 ejecución en el sistema de procesamiento de datos (3), de comandos de instrucciones, y en modificar el software protegido (2p): modificando por lo menos una porción seleccionada de la fuente del software protegido (2ps), esta modificación siendo tal que: al menos se fragmenta un proceso algorítmico de manera que cuando se ejecuta el software protegido (2p), se
10 ejecuta dicho proceso algorítmico en el medio de la segunda parte de ejecución (2peu) utilizando instrucciones; para al menos un proceso algorítmico seleccionado, se integran los comandos de instrucciones en la fuente de software protegido (2ps) de manera que cuando se ejecuta el software protegido (2p) se ejecuta cada comando de instrucciones en la
15 primera parte de ejecución (2pes) e inicia en la unidad (6) la ejecución de al menos la segunda parte de ejecución (2peu) de una instrucción, se elige una orden de comandos de instrucciones entre el conjunto de órdenes que permiten la ejecución del software protegido (2p), y el enlace de que deben obedecer por lo menos ciertas instrucciones cuando se especifica su
20 ejecución en la unidad (6), y en la fase de utilización (U), en presencia de la unidad (6), en el caso que se detecte que el enlace de las instrucciones ejecutadas en la unidad (6) no corresponden a aquel que se desea, informar al sistema de procesamiento de datos (3) y/o modificar el funcionamiento de la porción de software protegido (2p), de manera que se modifique el
25 funcionamiento del software protegido (2p).

7.- El procedimiento de conformidad con la reivindicación 6, caracterizado además porque consiste: en la fase de protección (P): en definir: como juego de instrucciones, un juego de instrucciones en donde por

lo menos ciertas instrucciones trabajan sobre registros y utilizan al menos un operando con el fin de arrojar un resultado, por lo menos una parte de las instrucciones que trabajan sobre los registros: una parte (PF) que define a la funcionalidad de la instrucción, y una parte que define el encadenamiento deseado para la ejecución de instrucciones y que comprende campos de bits que corresponden a: un campo de identificación de la instrucción (CII), y para cada operando de la instrucción: un campo bandera (CD_n), y un campo de identificación prevista (CIP_n) del operando, para cada registro que pertenece a los medios de aprovechamiento y utilizado por el juego de instrucciones, un campo de identificación generada (CIG_n) en la cual se memoriza automáticamente la identificación de la última instrucción que haya arrojado su resultado en este registro, como medios de detección (17), medios que permiten, durante la ejecución de una instrucción, para cada operando, cuando lo impone el campo bandera (CD_n), controlar la igualdad entre el campo de identificación generada (CIG_n) que corresponde al registro utilizado por este operando, y el campo de identificación prevista (CIP_n) del origen de este operando, y como medios de coerción (18), medios que permiten modificar el resultado de las instrucciones, si por lo menos una de las igualdades controladas es falsa.

8.- El procedimiento de conformidad con una de las reivindicaciones 1 a 7, caracterizado además porque consiste: en la fase de protección (P): en modificar el software protegido (2p): al seleccionar al menos una variable que se utilice en al menos un proceso algorítmico seleccionado, que cuando ejecuta el software protegido (2p) define parcialmente el estado de software protegido (2p), al modificar al menos una porción seleccionada de la fuente del software protegido (2ps) siendo esta modificación tal que cuando se ejecuta el software protegido (2p), al menos una variable seleccionada o al menos una copia de la variable seleccionada

reside dentro de la unidad (6), y que produce la primera parte objeto (2pos) del software protegido (2p), esta primera parte objeto (2pos) es tal que cuando se ejecuta el software protegido (2p), al menos una porción de la primera parte de ejecución (2pes) se hace cargo también al menos de una variable o al menos de una copia de variable que reside en la unidad (6), y la segunda parte objeto (2pou) del software protegido (2p), siendo segunda parte objeto (2pou) tal que, después de cargar lo que la unidad (6) y cuando se ejecuta el software protegido (2p), aparece la segunda parte de ejecución (2peu), en medio del cual al menos una variable seleccionada o al menos una copia de variable seleccionada reside también dentro de la unidad (6), y en la fase de utilización (U), en presencia de la unidad (6) cada vez que se le imponga una porción de la primera parte de ejecución (6), utilizar una variable o una copia de variable que resida en la unidad (6), de suerte que esa porción se ejecuta correctamente y que, en consecuencia, sea completamente funcional el software protegido (2p), y en ausencia de la unidad (6), a pesar de la petición de una porción de la primera parte de ejecución (2pes) de utilizar una variable o una copia de variable que reside la unidad (6), no poder responder correctamente a esa petición, de suerte que al menos no se ejecute esta porción correctamente y que, en consecuencia, no sea completamente funcional el software protegido (2p).

9.- El procedimiento de conformidad con la reivindicación 6, caracterizado además porque también consiste: en la fase de protección (P): en definir: como un comando iniciador, un comando de instrucción, como una función dependiente, una instrucción, como una orden, por lo menos un argumento para un comando iniciador, que corresponde por lo menos en parte a la información transmitida por el sistema de procesamiento de datos (3) a la unidad (6), con el fin de iniciar la ejecución de la función dependiente correspondiente, un método de renombramiento de órdenes que permita

renombrar las órdenes con el fin de obtener comandos iniciadores en órdenes renombradas, y medios de restablecimiento (20) destinados a ser aplicados en la unidad (6) en el curso de la fase de uso (U), y que permitan encontrar la función dependiente a ejecutar, a partir de la orden renombrada, en construir

5 medios de aprovechamiento que permitan a la unidad (6) aplicar también los medios de restablecimiento, y en modificar el software protegido (2p): seleccionando en la fuente del software protegido (2ps), comandos iniciadores, modificando por lo menos una porción seleccionada de la fuente del software protegido (2ps) renombrando las órdenes de comandos

10 iniciadores seleccionados, a fin de disimular la identidad de las funciones dependientes correspondientes, y produciendo: la primera parte objeto (2pos) del software protegido (2p), esta primera parte objeto (2pos) siendo tal que durante la ejecución del software protegido (2p), se ejecutan los comandos iniciadores en órdenes renombradas, y la segunda parte objeto (2pou) del

15 software protegido (2p) que contiene los medios de aprovechamiento que aplican también los medios de restablecimiento (20), esta segunda parte objeto (2pou) siendo tal que, después de carga en la unidad (6) y durante la ejecución del software protegido (2p), se reestablece la identidad de las funciones dependientes en donde la ejecución es iniciada por la primera parte

20 de ejecución (2pes), por medio de la segunda parte de ejecución (2peu), y las funciones dependientes son ejecutadas por medio de la segunda parte de ejecución (2peu), y en la fase de uso (U): en presencia de la unidad (6) y cada vez que un comando iniciador en orden renombrada, contenido en una porción de la primera parte de ejecución (2pes) lo imponga, reestablecer en la

25 unidad (6), la identidad de la función dependiente correspondiente y en ejecutar la misma, de modo que esta porción se ejecuta correctamente y que en consecuencia, el software protegido (2p) es completamente funcional, y en ausencia de la unidad (6), a pesar de la petición de una porción de la primera

parte de ejecución (2pes), iniciar la ejecución de una función dependiente en la unidad (6), en no poder responder correctamente a esta petición, de modo que por lo menos esta porción no se ejecuta correctamente y que en consecuencia, el software protegido (2p) no es completamente funcional.

5 10.- El procedimiento de conformidad con la reivindicación 9, caracterizado además porque consiste: en la fase de protección (P): en definir para al menos una función dependiente, una familia de funciones dependientes algorítmicamente equivalentes, pero iniciadas por comandos iniciadores en donde las órdenes renombradas son diferentes, y en modificar
10 el software protegido (2p): seleccionando en la fuente del software protegido (2ps) por lo menos una comando iniciador en orden renombrada, y modificando al menos una porción seleccionada de la fuente del software protegido (2ps) reemplazando por lo menos la orden renombrada de un comando de iniciador en orden renombrada seleccionada, por otra orden
15 renombrada, que inicie una función dependiente de la misma familia.

 11.- El procedimiento de conformidad con la reivindicación 10, caracterizado además porque consiste: en la fase de protección (P), en definir, para al menos una función dependiente, una familia de funciones dependientes algorítmicamente equivalentes: concatenando un campo de
20 ruido con la información que define la parte funcional de la función dependiente a ejecutar en la unidad (6), o utilizando el campo de identificación de la instrucción (CII) y los campos de identificación prevista (CIP_i) de los operandos.

 12.- El procedimiento de conformidad con la reivindicación 9, 10
25 u 11, caracterizado además porque consiste: en la fase de protección (P): en definir: como método de renombramiento de órdenes, un método de codificación para codificar las órdenes, y como medios de restablecimiento (20), medios que aplican un método de decodificación para decodificar las

órdenes renombradas y reestablecer así la identidad de las funciones dependientes que se ejecutarán en la unidad (6).

13.- El procedimiento de conformidad con una de las reivindicaciones 1 a 12, caracterizado además porque consiste: en la fase de protección (P): en modificar el software protegido (2p): seleccionando en la fuente del software protegido (2ps), por lo menos una derivación condicional efectuada en por lo menos un procesamiento algorítmico seleccionado, modificando por lo menos una porción seleccionada de la fuente del software protegido (2ps), esta modificación es tal que durante la ejecución del software protegido (2p), se ejecuta la funcionalidad de por lo menos una derivación condicional seleccionada, por medio de la segunda parte de ejecución (2peu), en la unidad (6), y produciendo: la primera parte objeto (2pos) del software protegido (2p), esta primera parte objeto (2pos) siendo tal que durante la ejecución del software protegido (2p), se ejecuta la funcionalidad de por lo menos una derivación condicional seleccionada en la unidad (6), y la segunda parte objeto (2pou) del software protegido (2p), esta segunda parte objeto (2pou) siendo tal que después de carga en la unidad (6) y durante la ejecución del software protegido (2p), aparece la segunda parte de ejecución (2peu) por medio de la cual se ejecuta la funcionalidad de por lo menos una derivación condicional seleccionada, y en la fase de uso (U): en presencia de la unidad (6) y cada vez que lo impone una porción de la primera parte de ejecución (2pes), ejecutar la funcionalidad de por lo menos una derivación condicional en la unidad (6), de modo que esta porción se ejecuta correctamente y que en consecuencia, el software protegido (2p) es complementemente funcional, y en ausencia de la unidad (6) y a pesar de la petición de una porción de la primera parte de ejecución (2pes), ejecutar la funcionalidad de una derivación condicional en la unidad (6), en no poder responder correctamente a esta petición, de modo que por lo menos esta

porción no se ejecuta correctamente y que en consecuencia, el software protegido (2p) no es completamente funcional.

14.-El procedimiento de conformidad con la reivindicación 13, caracterizado más porque consiste, en la fase de protección (P), en modificar
5 el software protegido (2p): seleccionando, en la fuente del software protegido (2ps) por lo menos una serie de derivaciones condicionales seleccionadas, modificando la menos una porción seleccionada de la fuente del software protegido (2ps), esta modificación es tal que durante la ejecución del software protegido (2p), se ejecuta la funcionalidad global de por lo menos una serie
10 seleccionada de derivaciones condicionales por medio de la segunda parte de ejecución (2peu), en la unidad (6), y produciendo: la primera parte objeto (2pos) del software protegido (2p), esta primera parte objeto (2pos) es tal que durante la ejecución del software protegido (2p), se ejecuta la funcionalidad de por lo menos una serie seleccionada de derivaciones condicionales en la
15 unidad (6), y la segunda parte objeto (2pou) del software protegido (2p), esta segunda parte objeto (2peu) siendo tal que después de carga en la unidad (6) y durante la ejecución del software protegido (2p), aparece la segunda parte de ejecución (2peu) por medio de la cual se ejecuta la funcionalidad global de por lo menos una serie seleccionada de derivaciones condicionales.

20 15.- El procedimiento de conformidad con una de las reivindicaciones 1 a 14, caracterizado además porque consiste fragmentar la fase de protección (P) en una subfase de protección corriente arriba (P1), independiente del software a proteger y una subfase de protección corriente abajo (P2), dependiente del software a proteger.

25 16.- El procedimiento de conformidad con la reivindicación 15, caracterizado además porque consiste, durante la subfase de protección corriente arriba (P1), en contener un período de definiciones (S11) en el cual se efectúan todas las definiciones.

17.- El procedimiento de conformidad con la reivindicación 16, caracterizado además porque consiste, después del período de definiciones (S11), en incluir un período de construcción (S12) en el cual se construyen los medios de aprovechamiento.

5 18.- El procedimiento de conformidad con la reivindicación 17, caracterizado además porque consiste, después del período de construcción (S12), en contener un período de prepersonalización (S12), que consiste en cargar en una unidad virgen (60), por lo menos una parte de los medios de aprovechamiento con el fin de obtener una unidad prepersonalizada (66).

10 19.- El procedimiento de conformidad con la reivindicación 16 ó 17, caracterizado más porque consiste, durante la subfase de protección corriente arriba (P1), en contener un período de confección de herramientas (S14) en el cual se confeccionan herramientas que permiten ayudar a generar softwares protegidos o automatizar la protección de softwares.

15 20.- El procedimiento de conformidad con las reivindicaciones 15 y 18, caracterizado además porque consiste en fragmentar la subfase de protección corriente abajo (P2) en: un período de creación (S21) en el cual se crea el software protegido (2p), a partir del software vulnerable (2v), eventualmente, un período de modificación (S22) en el cual se modifica el
20 software protegido (2p), y un período de personalización (S23) en el cual: se carga la segunda parte objeto (2pou) del software protegido (2p) que contiene los medios de aprovechamiento, en por lo menos una unidad virgen (60) para obtener al menos una unidad (6), o cargar una parte de la segunda parte objeto (2pou) del software protegido (2p) que contienen eventualmente los
25 medios de aprovechamiento en por lo menos una unidad prepersonalizada (66) para obtener al menos una unidad (6).

21.- El procedimiento de conformidad con las reivindicaciones 19 y 20, caracterizado además porque consiste, durante el período de creación

109/110

(S21) y eventualmente del período de modificación (S22), en utilizar por lo menos una de las herramientas de ayuda para la generación de softwares protegidos o automatización de la protección de softwares.

22.- Sistema para la aplicación del método de la reivindicación

5 17, caracterizado porque comprende una unidad de desarrollo de programas, que sirve, durante el estado de construcción (S12), para efectuar la construcción de medios de aprovechamiento destinados a la unidad (6), que toman en cuenta la definiciones contenidas en el período de definiciones (S11).

no permite uso

10 23.- Sistema para la aplicación del método de la reivindicación 18, caracterizado porque comprende una unidad de prepersonalización (30) que permite cargar al menos una parte de los medios de aprovechamiento en por lo menos una unidad virgen (60), con el fin de obtener por lo menos una unidad prepersonalizada (66).

15 24.- Sistema para la aplicación del método de la reivindicación 19, caracterizado porque comprende una unidad de desarrollo de programas, que sirven para efectuar durante el período de confección de herramientas (S14), la confección de herramientas de ayuda para la generación de programas protegidos o automatización de la protección de softwares.

20 25.- Sistema para la aplicación del método de la reivindicación 20 ó 21, caracterizado porque comprende una unidad de desarrollo de programas que sirven para crear o modificar un software protegido (2p).

25 26.- Sistema para la aplicación del método de la reivindicación 20, caracterizado porque comprende una unidad de personalización (45) que permite cargar: la segunda parte objeto (2pou) en por lo menos una unidad virgen (60), con el fin de obtener al menos una unidad (6), o una parte de la segunda parte objeto (2pou) en por lo menos una unidad prepersonalizada (66), con el fin de obtener al menos una unidad (6).

110
no
de
la
acción?
no

27.- Unidad prepersonalizada (66), caracterizada porque se obtiene con el sistema de la reivindicación 23.

28.- Unidad (6) que permite ejecutar un software protegido (2p) y de impedir su utilización no autorizada, caracterizada porque contiene la
5 segunda parte objeto (2pou) del software protegido (2p) cargado con la ayuda de una unidad de personalización (45) de la reivindicación 26.

29.- Conjunto de unidades (6), caracterizado porque la segunda parte objeto (2pou) del software protegido (2p), que se carga con ayuda de una unidad de personalización (45) de la reivindicación 26, se reparte en
10 varias unidades de procesamiento y de memoria, de manera que su utilización conjunta permite ejecutar el software protegido (2p).

30.- Conjunto de distribución (2pd) de un software protegido (2p), caracterizado porque comprende: una primera parte de distribución (2pds) que contiene la primera parte objeto (2pos) y que está destinada a
15 funcionar en un sistema de procesamiento de datos (3), y una segunda parte de distribución (2pdu) que se presenta bajo la forma: de una unidad virgen (60), o de una unidad prepersonalizada (66) de la reivindicación 27, que es capaz de, después de cargar informaciones de personalización, de transformarse en una unidad (6), o de una unidad (6) de la reivindicación 28.

20 31.- El conjunto de distribución (2pd) de un software protegido (2p) de conformidad con la reivindicación 30, caracterizado además porque la primera parte de distribución (2pds) se presenta en la forma de un medio de distribución física, por ejemplo CDROM, o en la forma de registros distribuidos a través de una red.

25 32.- El conjunto de distribución (2pd) de un software protegido (2p) de conformidad con la reivindicación 30, caracterizado además porque la segunda parte de distribución (2pdu) que se presenta en la forma de unidades

vírgenes (60), de unidades prepersonalizadas (66) o de unidades (6),
comprende al menos una tarjeta de chip (7).

33.- Unidad de procesamiento y de memoria, caracterizada
porque contiene la parte de la segunda parte objeto (2pou) necesaria para
5 transformar una unidad prepersonalizada (66) de la reivindicación 27, en una
unidad (6) de la reivindicación 28.

34.- Conjunto de unidades de procesamiento y de memoria,
caracterizado porque las unidades de procesamiento y de memoria al
utilizarse conjuntamente, contienen la parte de la segunda parte objeto (2pou)
10 necesaria para transformar una unidad prepersonalizada (66) de la
reivindicación 27 en una unidad (6) de la reivindicación 28.

RESUMEN DE LA INVENCION

La invención involucra un método para proteger, a partir de una unidad, un software vulnerable contra su uso no autorizado, dicho software vulnerable opera sobre un sistema de procesamiento de datos; el método
5 consiste en definir: al menos una característica de ejecución del software, capaz de ser monitoreado por lo menos parcialmente en una unidad; por lo menos un criterio a ser monitoreado por parte de por lo menos una característica de ejecución del software; un medio de detección a ser
10 implementado en una unidad y que permite detectar si por lo menos una característica de ejecución de software no obedece por lo menos un criterio relacionado; y un medio de coerción a implementarse en una unidad para informar al sistema de procesamiento de datos y/o modificar la ejecución de software cuando por lo menos no se obedece un criterio.

4A

.. P03/1743F

(WPC14117 / ID.7)

113 H4

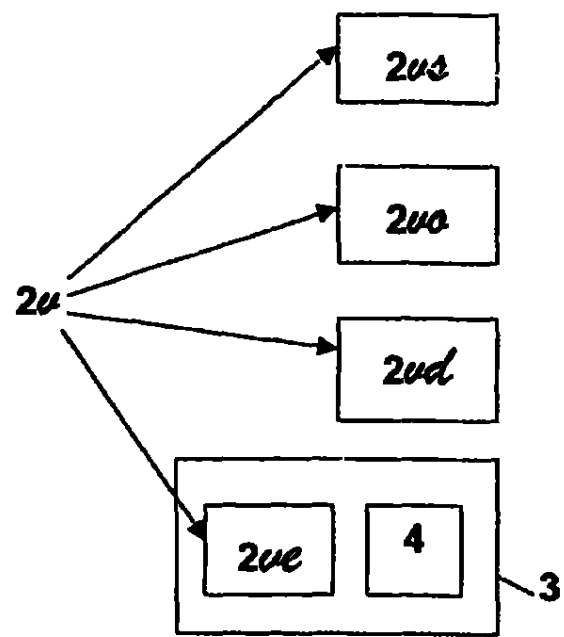


FIG. 1a

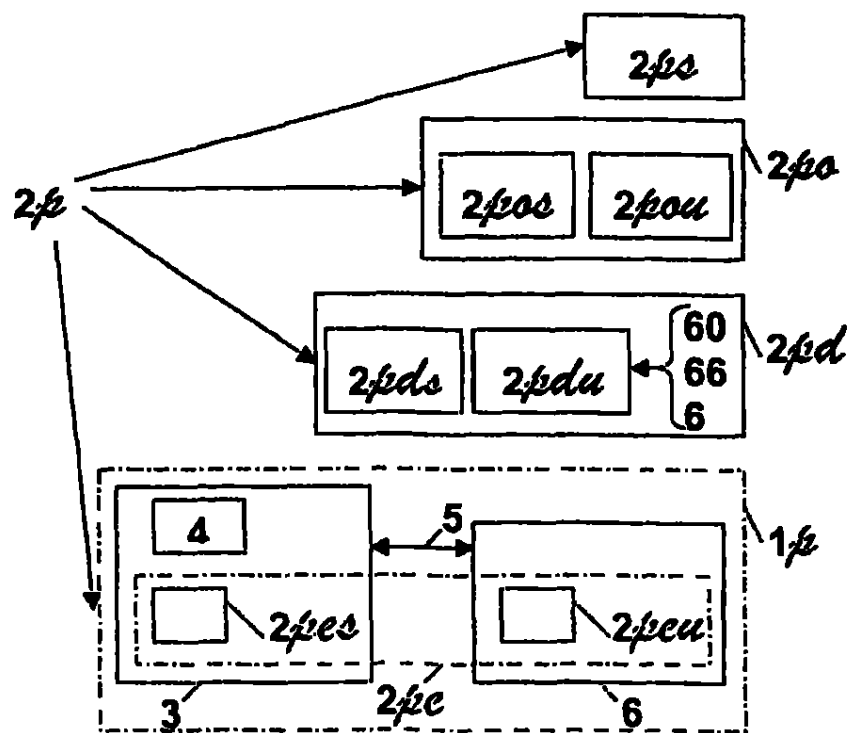


FIG. 1b

2/11

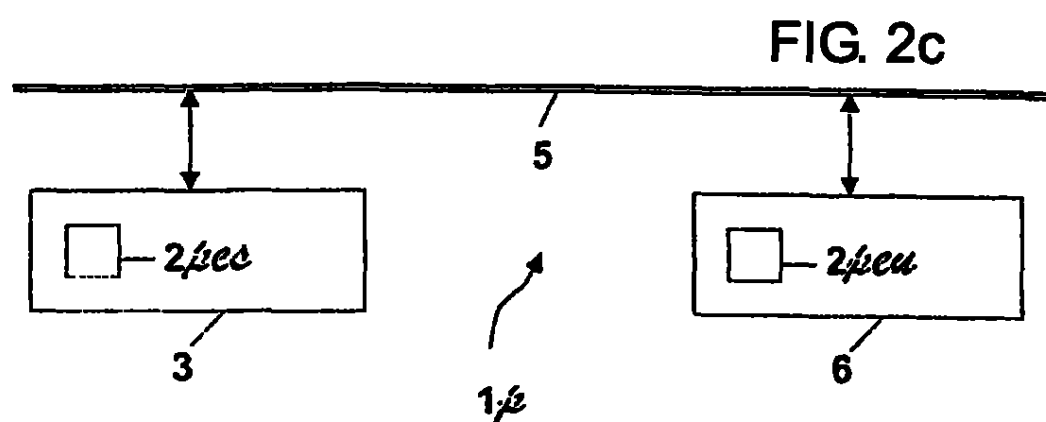
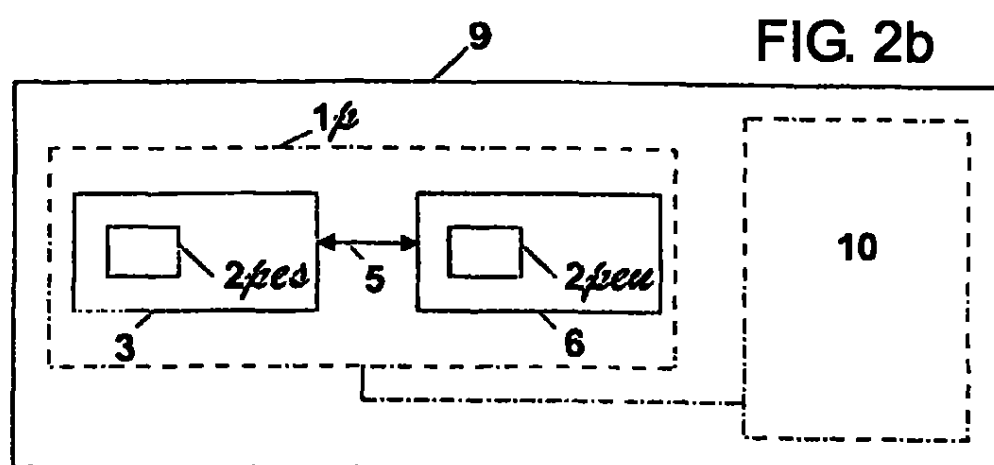
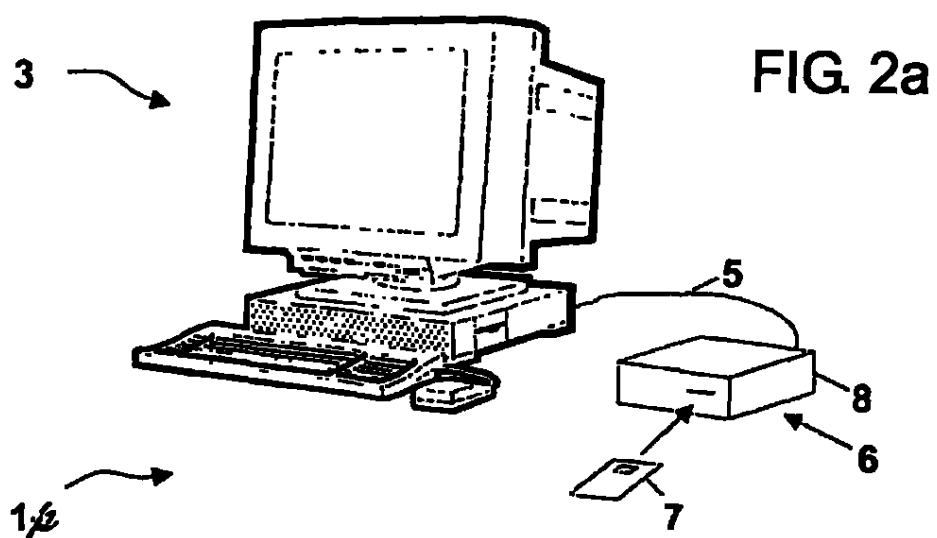


FIG. 3a

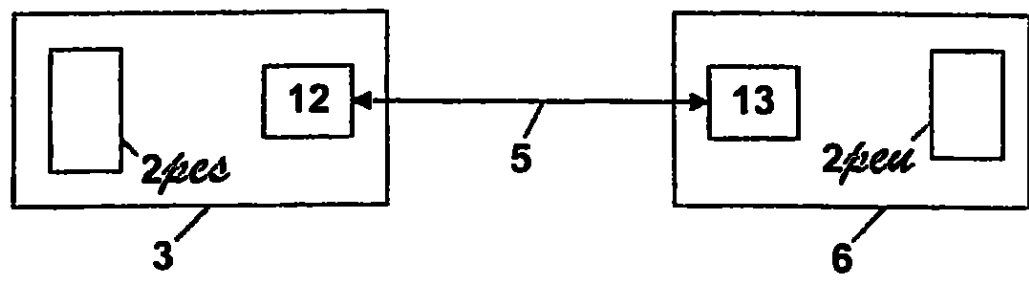


FIG. 3b

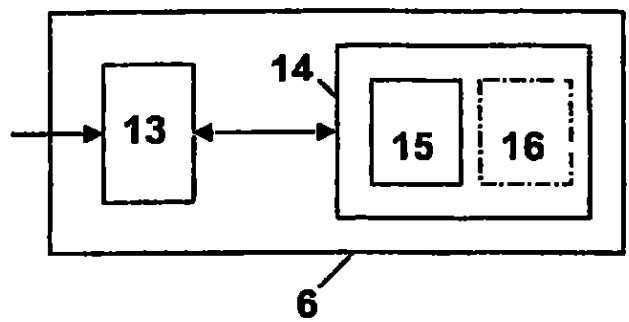


FIG. 4a

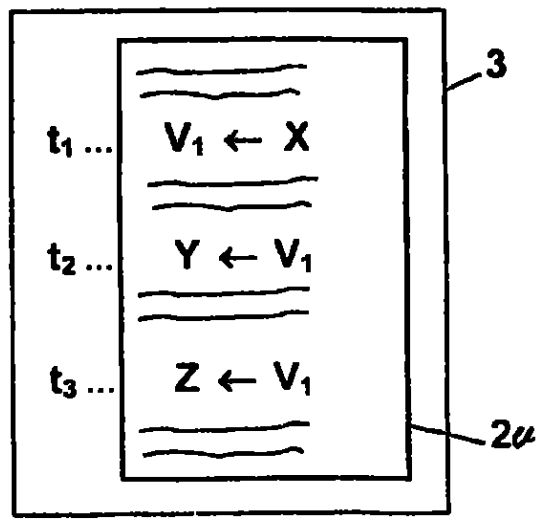


FIG. 4b

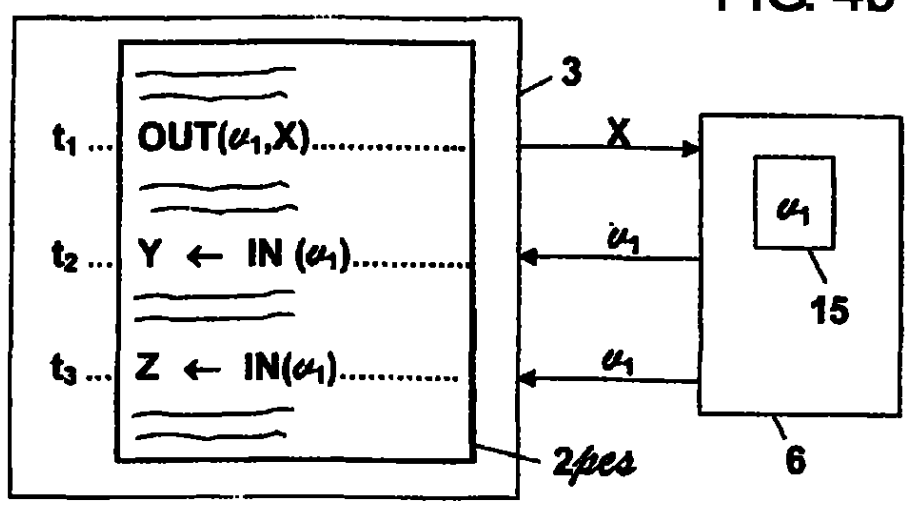


FIG. 4c

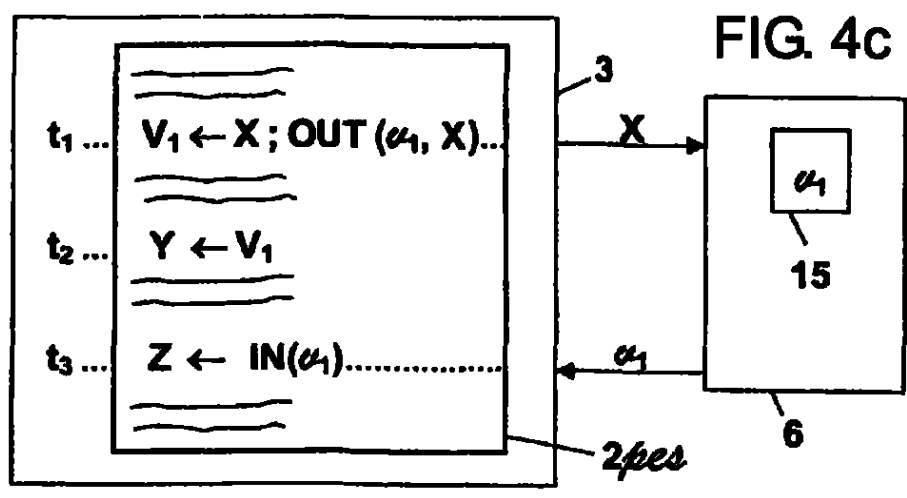
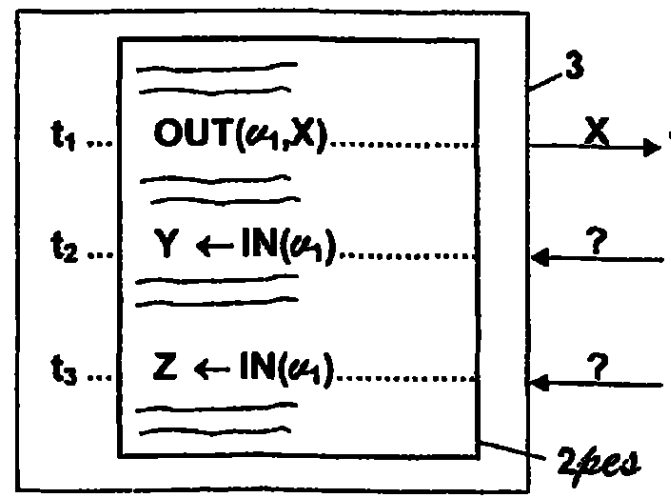


FIG. 4d



117 ~~118~~

FIG. 5a

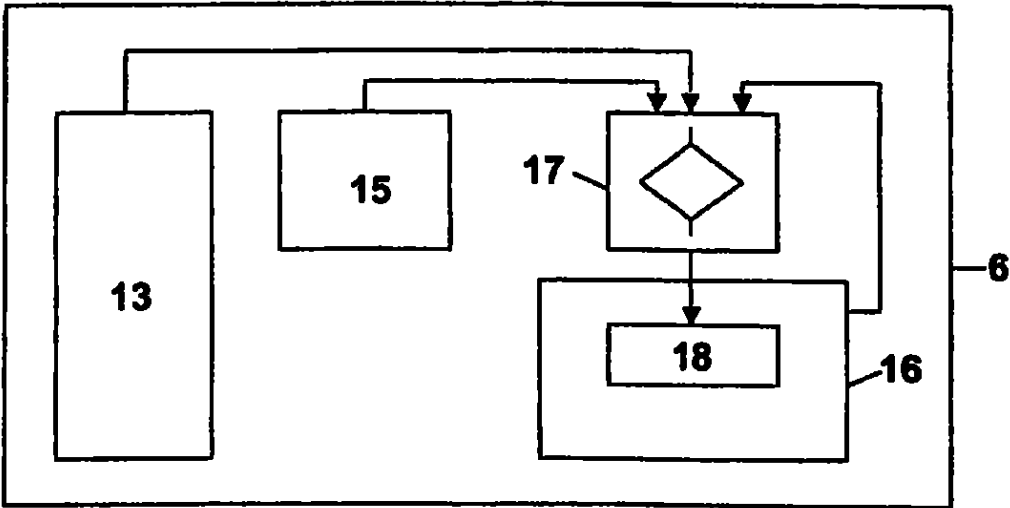
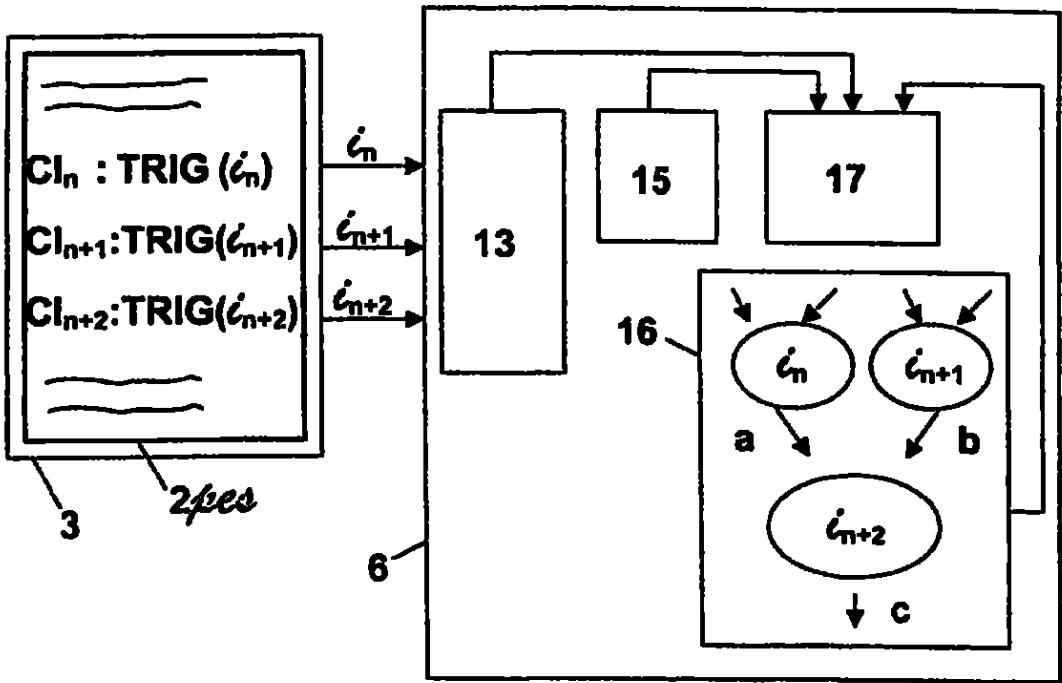


FIG. 5b



110/H9

FIG. 5c

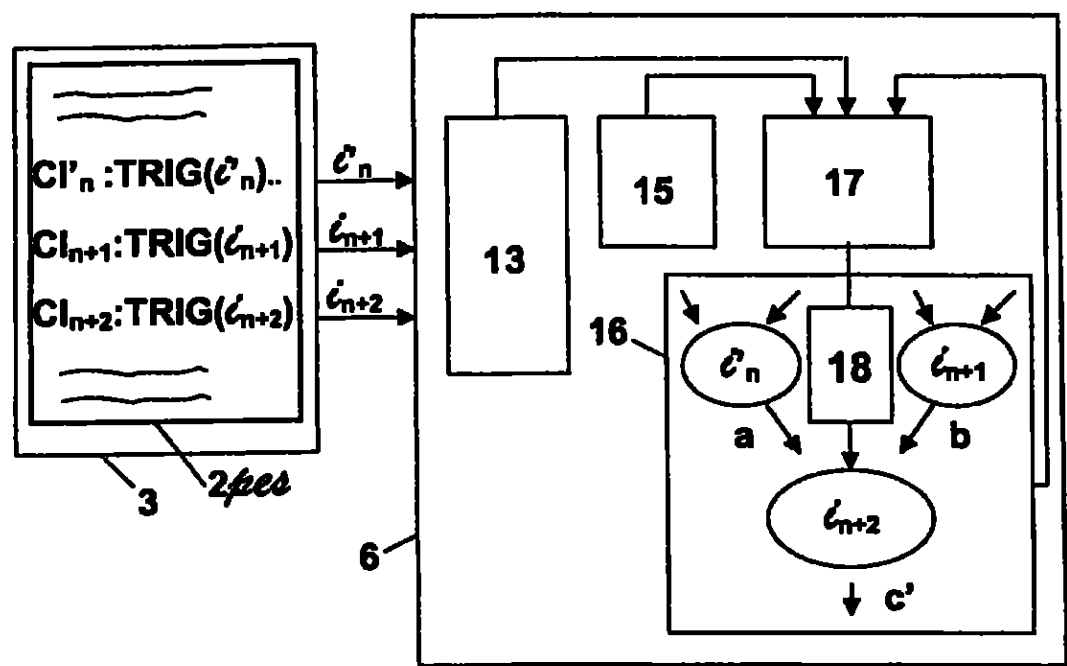


FIG. 5d

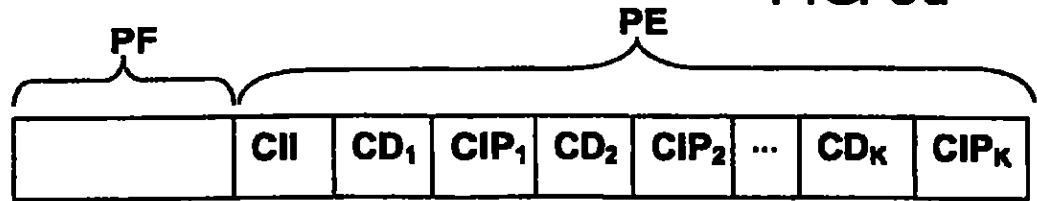


FIG. 5e



FIG. 6a

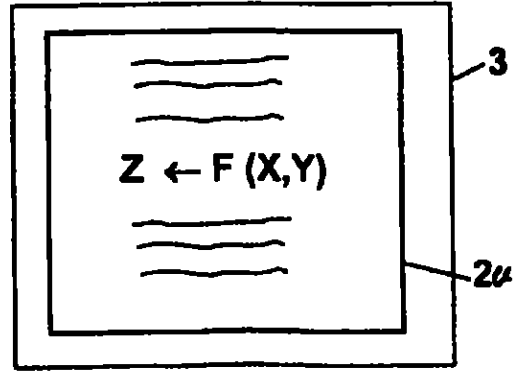


FIG. 6b

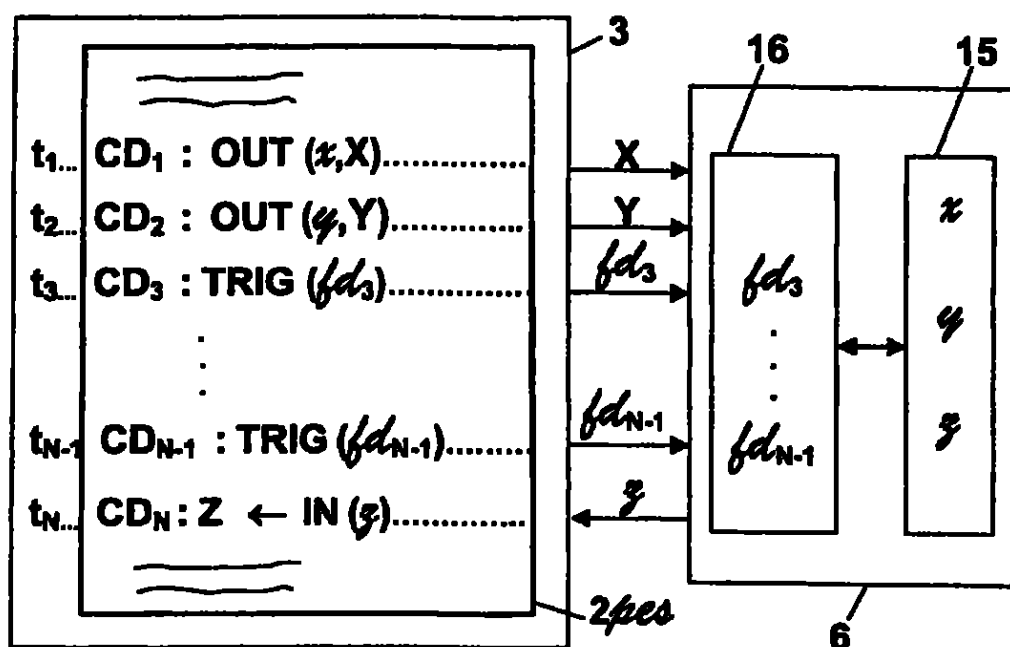
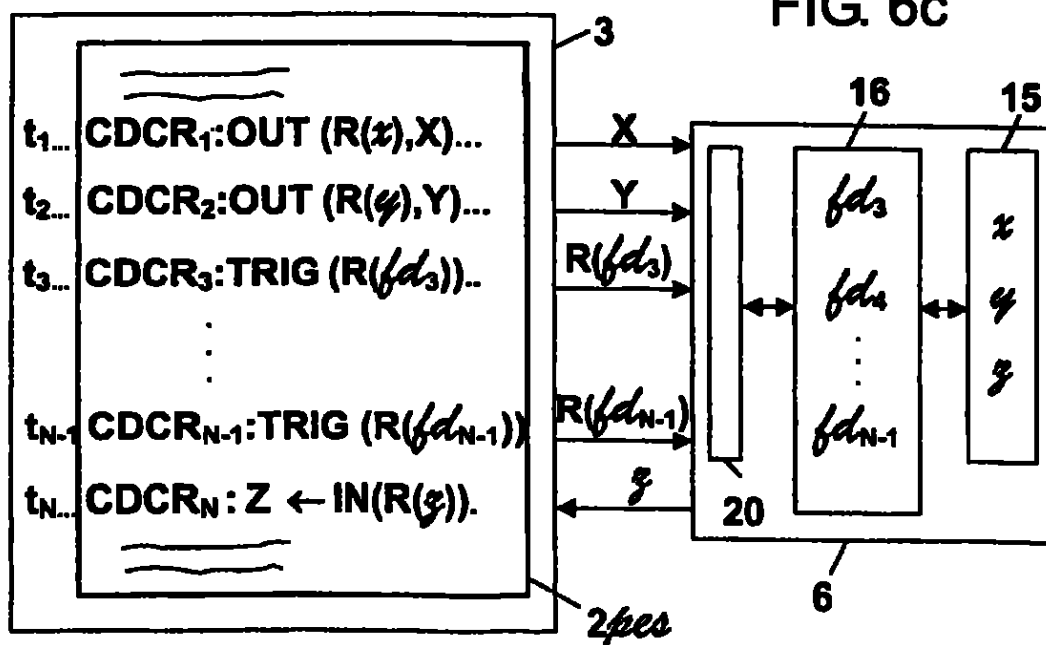


FIG. 6c



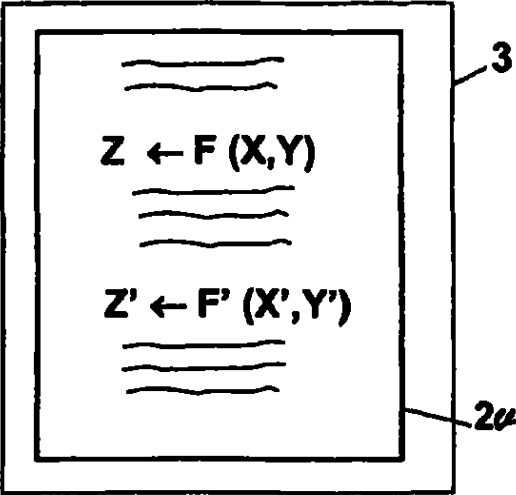


FIG. 6d

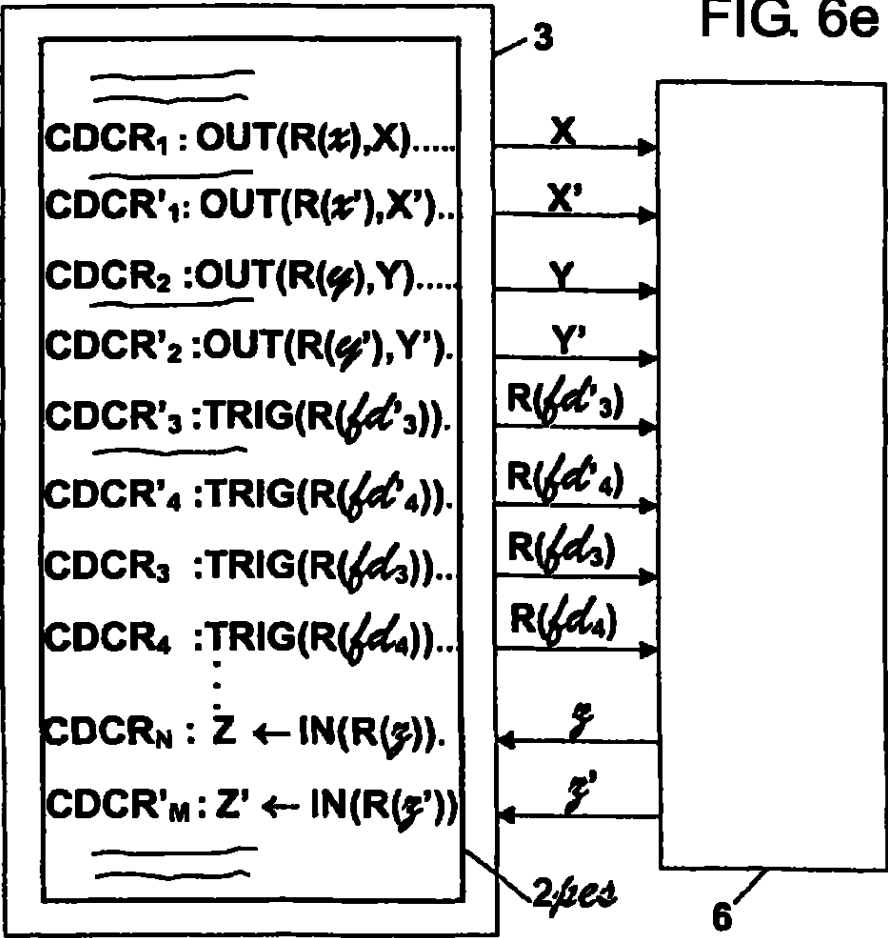
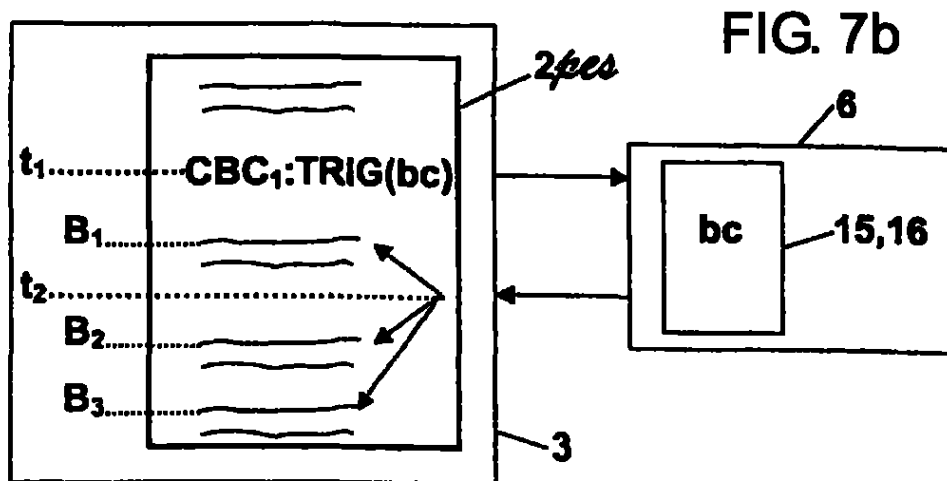
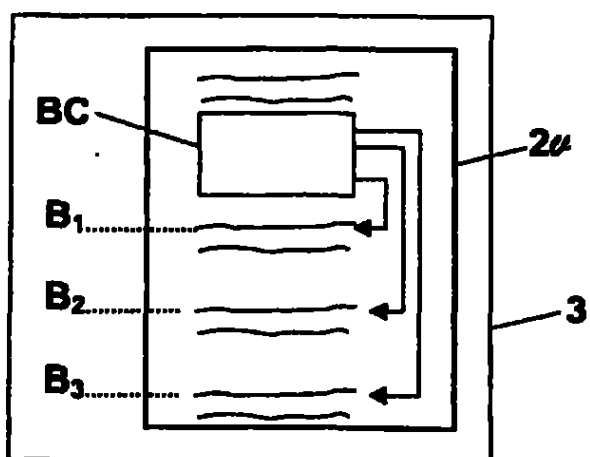
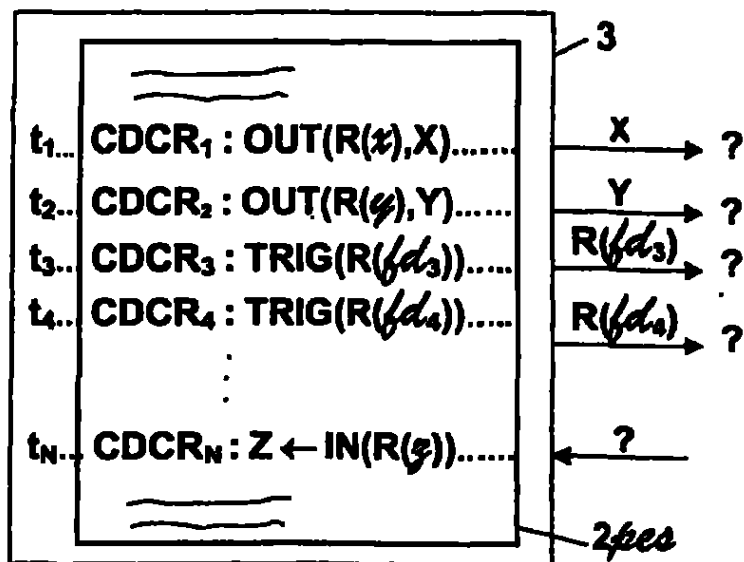


FIG. 6e

12/22



122 123

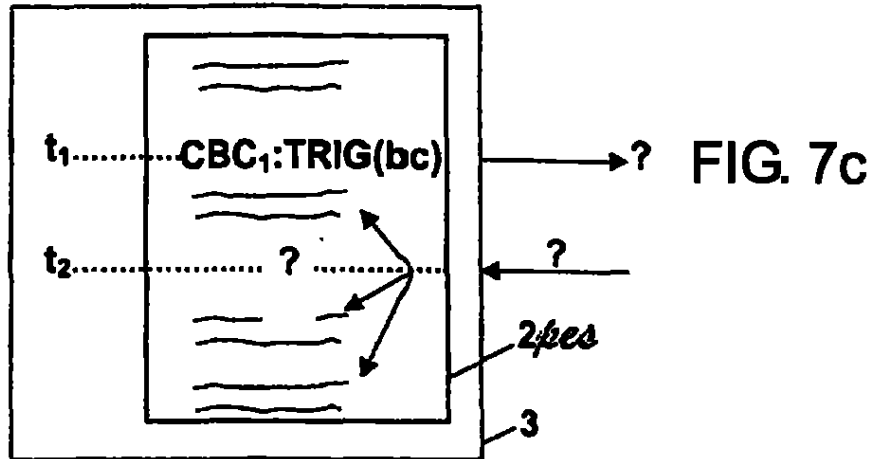
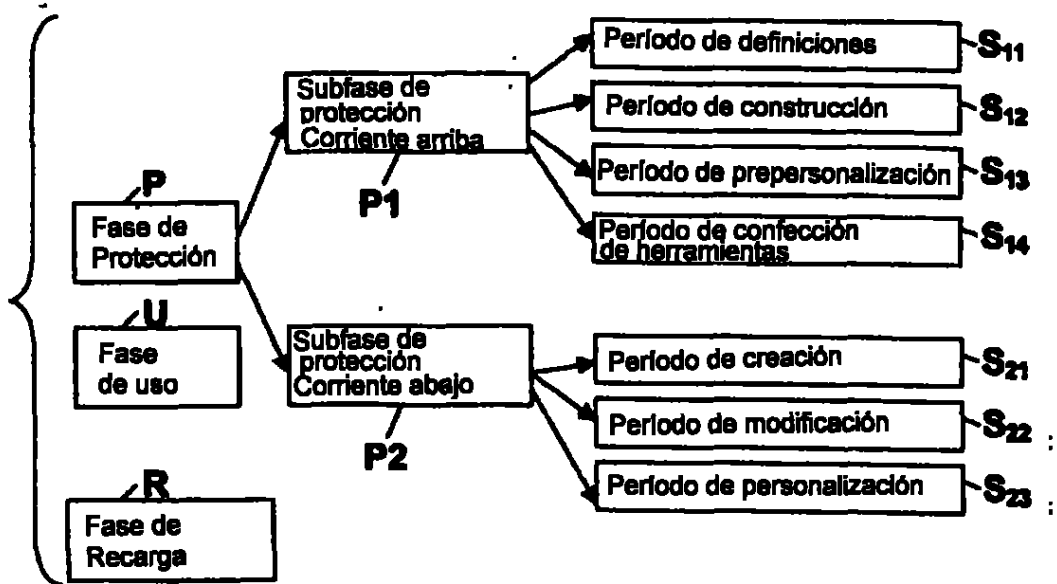


FIG. 8



25

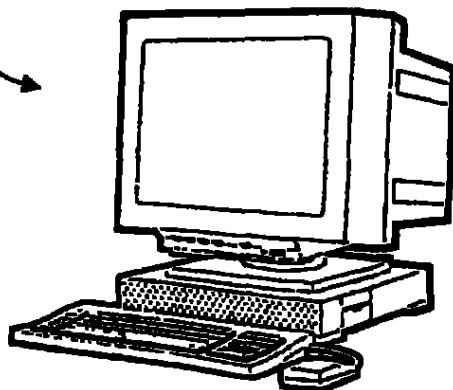
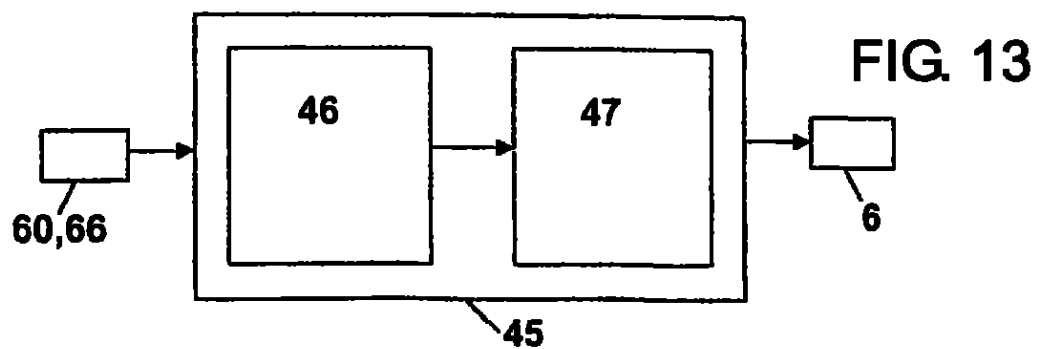
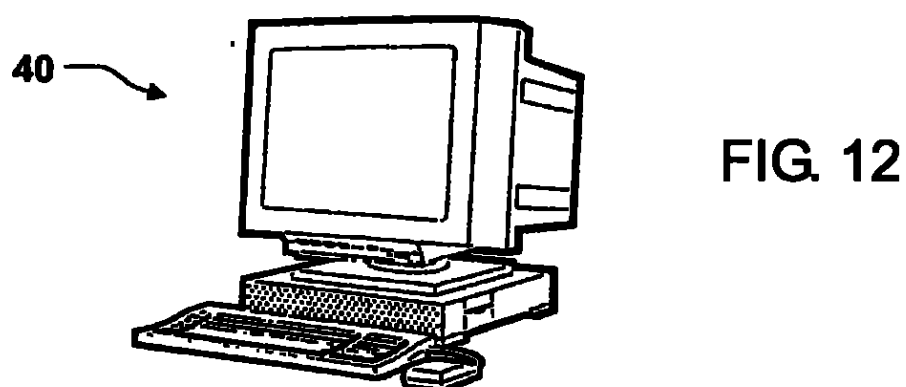
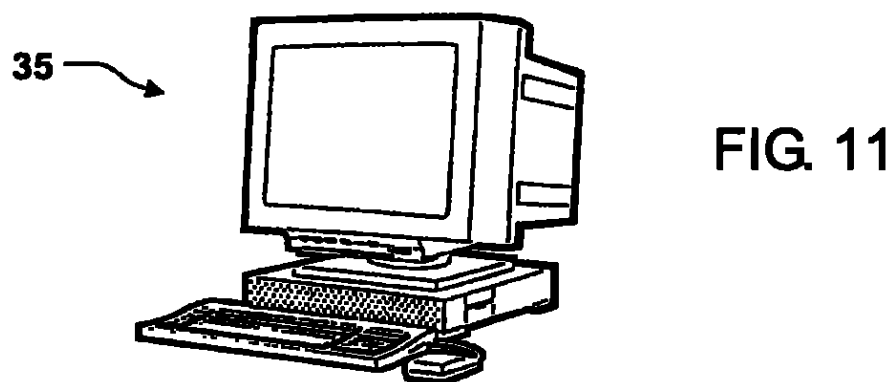
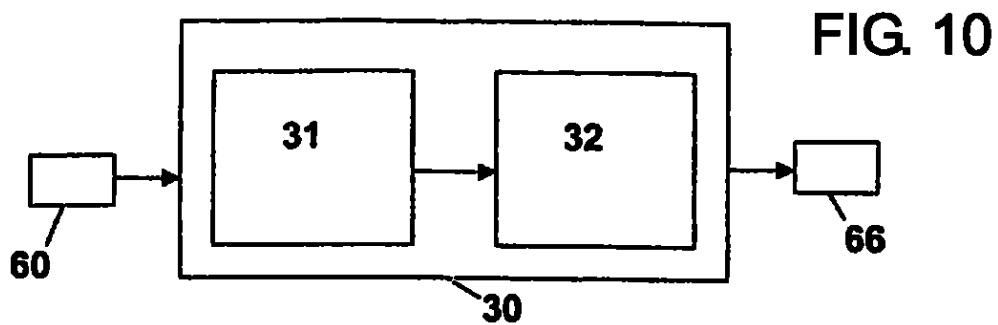


FIG. 9

123-124

11/11



125
12A

SUPERINDUSTRIA Y COMERCIO

Radicación : 04683757 C0606360

Folios: 125

Fecha (MCD): 2004-01-20 17:02:57

NIT : 800176089-2 Trámite : 011 FCT-PATENT 379 FASE NACI 411 PRESENTA
Dependencia: 8920 D:VISION DE NUEVAS CREACIONES

RECIBO OFICIAL DE CAJA : 4 - 3,590
FECHA : ENERO 19 DE 2004

***** CONSIGNACION *****

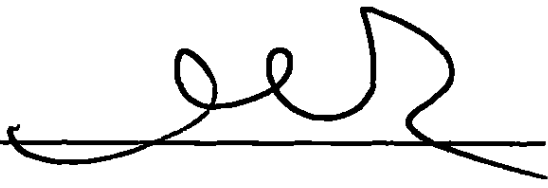
DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CAVELIER ABOGADOS	CONSIGNACION	BANCO POPULAR	050-00110-6	10497413	19/01/2004	38,989,800.00

***** C O N C E P T O *****

CANT.	RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	50005-01-01 SOLICITUDES	1 TRAMITES DE SOL. DE PATENTE DE IN	422,000.00
		TOTAL :	\$ 422,000.00

SOM: CUATROCIENTOS VEINTIDOS MIL PESOS

RESPONSABLE :



RECIBO DE CAJA APLICADO AL ASUNTO No.

Ob 14117-841-002-5

REQUISITOS NECESARIOS PARA PRESENTACION Y AGILIZACION DEL TRAMITE

126
125

		USUARIO	RADICADOR
PATENTE DE INVENCION	MODELO DE UTILIDAD	()	()
- Indicación de que se solicita la concesión de una patente.		()	()
- Datos de identificación del solicitante o de la persona que se presenta la solicitud.		()	()
- Descripción de la invención.		()	()
- Dibujos de ser estos pertinentes.		()	()
- Comprobante de Pago de las tasas establecidas.		()	()
COMPLETA ()	INCOMPLETA ()		

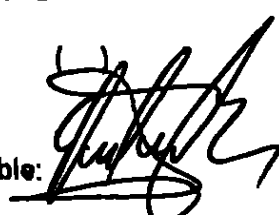
SEÑO INDUSTRIAL ()

- Indicación de que se solicita el registro de Diseño Industrial	()	()
- Datos de identificación del solicitante o de la persona que presenta la solicitud.	()	()
- Representación gráfica y fotográfica del Diseño Industrial o muestra del Material que incorpora el diseño.	()	()
- Comprobante de pago de las tasas establecidas	()	()
COMPLETA ()	INCOMPLETA ()	

ESQUEMA DE TRAZADO ()

- Indicación de que solicita el registro de un Esquema de trazado.	()	()
- Datos de identificación del solicitante o de la persona que presenta La solicitud.	()	()
- Representación gráfica del esquema de trazado	()	()
- Comprobante de pago de las tasas establecidas.	()	()
COMPLETA ()	INCOMPLETA ()	

Firma Responsable:



SUPERINDUSTRIA Y COMERCIO

Fecha: Radicación: 0403757 0000000 Folios: 125

Fecha (UTC): 2004-01-28 17:02:57

Trámite: 011 PCT-PATENT 379 FASE INCI 411 PRESENTAC

República: 2020 DIVISION DE NUEVAS CREACIONES



Industria y Comercio
SUPERINTENDENCIA

126
127

2020
Bogotá, D.C.,

Doctor:
Emilio Ferrero

Oficio N° 03243

ASUNTO:	Radicación N°	04-3757
	Solicitud Internacional	PCT/FR2002/02343
	Fecha Inicio fase nacional	02/03/2004
	Trámite	011
	Evento	379
	Actuación	430
	Folios	001

Como resultado del examen de forma, realizado con fundamento en el artículo 38 de la Decisión 486 de la Comisión de la Comunidad Andina, artículo 27 del Tratado de Cooperación en Materia de Patentes (PCT), regla 51 bis del Reglamento y Circular Única de la Superintendencia de Industria y Comercio, se le comunica que:

- ❑ La solicitud no contiene la copia del documento en que consta la cesión del derecho a la patente del inventor al solicitante o a su causante. (Art. 26-k) Decisión 486).
- ❑ La solicitud no contiene el poder debidamente otorgado, con el lleno de los requisitos exigidos por los artículos 63 y subsecuentes del Código de Procedimiento Civil. Adicionalmente, deberá presentar el documento que acredita la existencia y representación legal de la sociedad solicitante, cuando ésta fuere persona jurídica.

En cumplimiento de lo dispuesto por el artículo 39 de la Decisión 486, se le requiere para que dentro del término de dos meses siguientes a la fecha de notificación del presente oficio, complete los requisitos anteriormente enunciados. En caso de no dar cumplimiento al presente requerimiento, la solicitud se considerará abandonada y perderá su prelación.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 5.2, literal e), del capítulo quinto del título primero de la Circular Única N° 10 de 2001.


ALIX CARMENZA CESPEDES DE VERGEL
Jefe de la División de Nuevas Creaciones.

El anterior requerimiento fue notificado por fijación en lista de fecha 29 JUL 2020.

202063