(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 800 480**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **00 12360**

(51) Int Cl[7] : **G 06 F 12/14**

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 28.09.00.

(30) **Priorité :** 29.10.99 US 09429963.

(43) **Date de mise à la disposition du public de la demande :** 04.05.01 Bulletin 01/18.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Ce dernier n'a pas été établi à la date de publication de la demande.*

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

(71) **Demandeur(s) :** *IBM CORP INTERNATIONAL BUSINESS MACHINES CORPORATION Société de droit de l'état de New-York* — US.

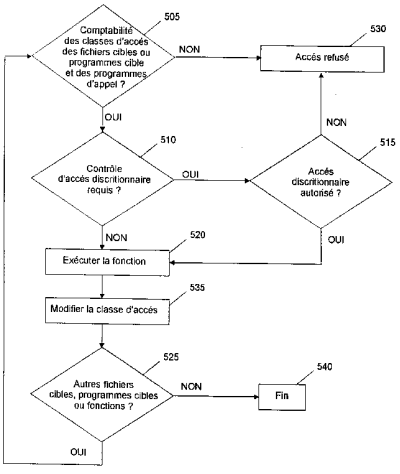
(72) **Inventeur(s) :** AUSTEL VERNON RALPH, KARGER PAUL ASHLEY et TOLL DAVID CLAUDE.

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

(74) **Mandataire(s) :** IBM FRANCE.

(54) METHODE ET SYSTEME DE SECURITE POUR LA PROTECTION DES FICHIERS.

(57) L'accès de programmes d'appel à des fichiers est contrôlé, les fichiers comprenant d'autres fichiers, des programmes et des données. Une classe d'accès initiale est assignée à chaque fichier et à chaque programme d'appel. Une classe d'accès comprend une classe d'intégrité et une classe de confidentialité. Une classe d'intégrité comprend des règles définissant la modification des données contenues dans les fichiers et une classe de confidentialité d'accès comprend des règles définissant la divulgation des données contenues dans les fichiers. Une classe d'intégrité comprend une série de règles pour permettre l'exécution d'une fonction de lecture et une autre série de règles pour permettre l'exécution d'une fonction d'écriture/ exécution. Une fonction d'exécution comprend des fonctions de transfert et de chaînage, le chaînage comprenant le lancement d'une autre opération dotée de classes de confidentialité et d'intégrité potentiellement différentes. Une classe de confidentialité comprend une série de règles pour permettre l'exécution d'une fonction d'écriture et une autre série de règles pour permettre l'exécution d'une fonction de lecture/ exécution. Les classes d'accès respectives du fichier cible ou du programme cible et du programme d'appel sont comparées. Si les résultats de la comparaison sont conformes aux exigences de sécurité, la fonction est exécutée.



FR 2 800 480 - A1

DOMAINE TECHNIQUE DE L'INVENTION

La présente invention concerne un dispositif de sécurité pour contrôler l'accès aux données, et plus particulièrement le contrôle de l'accès aux fichiers sur une unité de stockage telle qu'une carte à mémoire.

ETAT DE LA TECHNIQUE

Il est indispensable de disposer d'un modèle formel de sécurité lorsqu'on raisonne sur la sécurité d'un système. Les modèles de sécurité peuvent être répartis en trois grandes catégories : (1) les modèles qui protègent contre la divulgation non autorisée d'informations, (2) les modèles qui protègent contre l'altération non autorisée d'informations ou le sabotage et (3) les modèles qui protègent contre le déni de service. La protection contre la divulgation d'informations est celle qui est comprise depuis le plus longtemps et bénéficie des modèles les plus simples. La protection contre l'altération non autorisée d'informations ou le sabotage est beaucoup moins bien comprise et ce n'est que maintenant que des modèles appropriés sont en cours de développement. La protection contre le déni de service n'est pas encore bien comprise.

Une première exigence de nombreux systèmes de sécurité est d'empêcher la divulgation non autorisée d'informations. Les différents mécanismes comprennent des systèmes de contrôle d'accès discrétionnaires et des systèmes de contrôle d'accès obligatoires. Les systèmes de contrôle d'accès discrétionnaires sont des systèmes de sécurité courants basés sur la matrice d'accès générale de Lampson (Lampson, B.W., Protection. Operating Systems Review, janvier 1974. 8(1) : p. 18-24, initialement publié dans les actes de la Cinquième

conférence de Princeton sur l'informatique et les systèmes d'information, mars 1971). Ils sont appelés discrétionnaires parce que les droits d'accès à un objet peuvent être déterminés à la discrétion du propriétaire ou du contrôleur de l'objet. Les systèmes à liste et à fonction de contrôle d'accès sont des exemples de systèmes de contrôle d'accès discrétionnaires. La présence d'un cheval de Troie dans le système peut causer de grandes difficultés avec les systèmes de contrôle discrétionnaires. Le cheval de Troie peut modifier subrepticement les droits d'accès à un objet ou réaliser une copie d'une information protégée et transmettre cette copie à un utilisateur non autorisé. Tous les types de systèmes de contrôle discrétionnaires sont vulnérables à ce genre d'attaques par les chevaux de Troie. Un cheval de Troie dans un système à fonction de protection peut faire une copie d'une fonction concernant un objet protégé, puis stocker cette fonction dans un autre objet auquel un tiers peut avoir un accès en lecture. Dans les deux cas, l'information est transmise à un destinataire non autorisé.

Lampson (Lampson, B.W., A note on the confinement problem. Communications de l'ACM, octobre 1973. 16(10) : p. 613-615) a défini le problème du confinement comme la détection de la présence d'une série d'opérations dans un système de sécurité qui permettent finalement la divulgation d'informations à une personne non autorisée. Harrison, Ruzzo et Ullman (Harrison, M.A., W.L. Ruzzo et J.D. Ullman, Protection in Operating Systems. Communications de l'ACM, août 1976. 19(8) : p. 461-471) ont démontré qu'il n'existait pas de solution au problème du confinement pour les systèmes de contrôle d'accès discrétionnaires généraux tels que les système généraux à fonction ou à liste de contrôle. Leur argumentation est basée sur la modélisation des transitions d'état de la matrice d'accès prises comme les transitions d'état de la machine de Turing. Ils démontrent que résoudre le problème du confinement

reviendrait à résoudre le problème de l'arrêt de la machine de Turing.

Les canaux par lesquels un cheval de Troie divulgue des informations sont appelés passages secrets. Les passages secrets peuvent être divisés en deux grandes catégories : les canaux de mémoire et les canaux de synchronisation. Des informations peuvent être divulguées à travers un canal de mémoire en changeant les valeurs d'une des variables d'état du système. Ainsi, les canaux de stockage peuvent agir, par exemple, sur le contenu des fichiers, leur nom ou l'espace qu'ils utilisent sur le disque. Un cheval de Troie peut divulguer des informations à travers un canal de mémoire de manière purement asynchrone. Il n'y a aucun rapport avec la synchronisation.

A l'inverse, des informations peuvent être divulguées à travers un canal de synchronisation en modifiant le laps de temps nécessaire à l'exécution des fonctions du système. Par exemple, un cheval de Troie peut coder des informations pour modifier délibérément la cadence de pagination par défaut du système. Les canaux de synchronisation utilisent tous des communications synchrones et nécessitent une forme quelconque de synchronisation externe.

Des systèmes de contrôle d'accès obligatoires ont été développés pour remédier aux problèmes des chevaux de Troie des systèmes de contrôle d'accès discrétionnaires. La caractéristique des systèmes de contrôle d'accès obligatoires est que le gestionnaire du système ou le responsable de la sécurité peuvent contraindre le propriétaire d'un objet à déterminer qui peut avoir le droit d'accéder à cet objet. Tous les systèmes de contrôle obligatoires actuels sont basés sur des modèles de sécurité en treillis.

Il existe actuellement divers modèles pour décrire les propriétés de sécurité des systèmes informatiques et des utilisateurs. L'accès étant au coeur des exigences de sécurité des systèmes informatiques, nombre de ces modèles sont basés sur le contrôle d'accès. Les modèles de sécurité en treillis sont particulièrement intéressants. Un modèle de sécurité en treillis est constitué d'un ensemble de classes d'accès qui forment une classification partielle. Les classes d'accès qui ne font pas parties de cet ensemble sont dites non consécutives. Deux classes d'accès peuvent être inférieures, supérieures, égales ou non ordonnancées l'une par rapport à l'autre. De plus, il existe une classe d'accès inférieure, dite minimum système, qui est inférieure à toutes les autres classes d'accès. Il existe également une classe d'accès supérieure, dite maximum système, qui est supérieure à toutes les autres classes d'accès.

Un treillis très simple peut être constitué de deux classes d'accès : MINIMUM et MAXIMUM. MINIMUM est inférieur à MAXIMUM. MINIMUM est le minimum système et MAXIMUM le maximum système. Un modèle légèrement plus complexe peut comprendre une série de niveaux de confidentialité tels que NON SECRET, CONFIDENTIEL, SECRET et TOP SECRET. Dans ce cas, NON SECRET est le minimum système et TOP SECRET le maximum système. Chaque niveau de la série représente des données de confidentialité croissante.

Il n'est pas nécessaire qu'il existe une relation hiérarchique stricte entre les classes d'accès. Les services militaires américains utilisent une série de classes d'accès en deux parties : un niveau de confidentialité et une série de catégories. Les catégories représentent des ensembles d'informations pour lesquels un individu doit être expressément autorisé. Pour accéder aux informations d'une catégorie donnée, l'individu doit être autorisé non seulement pour le niveau de confidentialité de ces informations, mais

aussi pour la catégorie correspondante. Par exemple, s'il y a une catégorie NUCLEAIRE et que des informations sont classées SECRET - NUCLEAIRE, un individu bénéficiant d'une autorisation TOP SECRET ne peut pas accéder à ces informations s'il n'est pas expressément autorisé pour la catégorie NUCLEAIRE.

Des informations peuvent appartenir à plusieurs catégories; elles sont alors comparées à l'aide de sous-séries. Ainsi, dans le modèle en treillis militaire, pour que la classe d'accès A soit inférieure ou égale à la classe d'accès B, il faut que le niveau de confidentialité A soit inférieur ou égal au niveau de confidentialité B, et que la série de catégories de A soit un sous-ensemble fractionnaire de la série de catégories de B. Deux séries de catégories pouvant être non consécutives, la série complète des classes d'accès n'a qu'un ordonnancement partiel. Il y a une classe d'accès minimum, {NON SECRET - pas de catégories}, et une classe d'accès maximum {TOP SECRET - toutes catégories}. Les classes d'accès constituées de niveaux et de séries de catégories forment un treillis.

Les premiers modèles en treillis ont été développés chez MITRE Corporation, par Bell et LaPadula (Bell, D.E. et L.J. LaPadula, Secure Computer Systems: A Mathematical Model, ESD-TR-73-278, vol. II, novembre 1973, The MITRE Corporation, Bedford, MA. : Division des systèmes électroniques HQ, Hanscom AFB, MA.) et à la Case Western Reserve University, par Walter (Walter, K.G., W.F. Ogden, W.C. Rounds, F.T. Bradshaw, S.R. Ames et D.G. Shumway, Primitive Models for Computer Security, ESD-TR-74-117, 23 janvier 1974, Case Western Reserve University, Cleveland, OH. : Division des systèmes électroniques HQ, Hanscom AFB, MA.) pour formaliser le modèle de sécurité militaire et développer des techniques permettant de lutter contre les chevaux de Troie qui tentent de divulguer des informations. A cette époque, la lutte contre les chevaux de Troie était difficile; deux règles très simples ont

néanmoins été trouvées pour empêcher un cheval de Troie de divulguer des informations sensibles.

Premièrement, une règle de sécurité simple dit que si un sujet désire accéder en lecture à un objet, la classe d'accès de l'objet doit être inférieure ou égale à la classe d'accès du sujet. Il s'agit simplement d'une formalisation des procédures d'autorisation de sécurité militaire selon lesquelles on ne peut pas lire un document si l'on ne possède pas l'autorisation adéquate. Deuxièmement, selon la règle du confinement, si un sujet désire accéder en écriture à un objet, la classe d'accès du sujet doit être inférieure à la classe d'accès de l'objet. L'application de la règle du confinement aboutit à ce qu'un cheval de Troie qui tente de voler des informations dans une classe d'accès donnée ne peut pas stocker ces informations ailleurs que dans des objets qui possèdent une classe d'accès au moins égale à celle de la source des informations. Ainsi, le cheval de Troie peut modifier les informations, mais il ne peut pas les transmettre à un individu non autorisé.

Biba (Biba, K.J., Integrity Considerations for Secure Computer Systems, ESD-TR-76-732, avril 1977, The MITRE Corporation, Bedford, MA. : Division des systèmes électroniques HQ, Hanscom AFB, MA.) a ensuite développé un modèle d'intégrité obligatoire qui est un dual mathématique du modèle de sécurité obligatoire de Bell et LaPadula. Biba définit une série de classes d'intégrité d'accès qui sont semblables aux classes de sécurité d'accès, ainsi que des règles d'intégrité simple et de confinement d'intégrité qui sont semblables aux règles de sécurité simple et de confinement. La différence entre intégrité et sécurité réside dans le fait que le sens des signes "inférieur à" est inversé, de sorte qu'un programme à haute intégrité ne peut pas lire ou exécuter des objets à faible intégrité susceptibles d'être la source de trafics ou de sabotages. La principale difficulté du modèle d'intégrité

de Biba est qu'il ne correspond à aucun système existant. Contrairement aux modèles de sécurité qui ont été développés à partir de systèmes de sécurité militaires existants, le modèle d'intégrité de Biba a été développé à partir d'une analyse mathématique des modèles de sécurité.

Lipner a développé un modèle d'intégrité commercial (Lipner, S.B., Non-Discretionary Controls for Commercial Applications. Actes du symposium 1982 sur la sécurité et la confidentialité, 26 au 28 avril 1982. Oakland, CA : IEEE Computer Society, pp. 2-10) qui utilise à la fois les modèles de sécurité obligatoire et d'intégrité obligatoire pour constituer un environnement de développement de logiciel dans une banque. Son modèle d'intégrité était beaucoup plus proche de la réalité que celui de Biba, mais il restait relativement complexe. Les inventeurs n'ont pas connaissance de tentatives d'application concrète du modèle d'intégrité commercial de Lipner.

Une réalisation plus récente visant à empêcher l'altération d'informations et le sabotage est le modèle d'intégrité commercial de Clark et Wilson (Clark, D.D. et D.R. Wilson. A Comparison of Commercial and Military Computer Security Policies. Symposium IEEE 1987 sur la sécurité et la confidentialité, 27 au 29 avril 1987. Oakland, CA : IEEE Computer Society, pp. 184-194). Ils ont proposé un modèle d'intégrité des données dont ils affirment qu'il décrit plus précisément les besoins d'une application commercial que le modèle de sécurité en treillis de Bell et LaPadula (Bell, D.E. et L.J. LaPadula, Computer Security Model: Unified Exposition and Multics Interpretation, ESD-TR-73-306, juin 1975, The MITRE Corporation, Bedford, MA. : Division des systèmes électroniques HQ, Hanscom AFB, MA.). Le modèle de Clark et Wilson s'attache à deux notions : la bonne définition des transactions et la séparation des tâches. La séparation des tâches est couramment utilisée dans les entreprises

commerciales pour se protéger contre la fraude. Clark et Wilson se sont distingués dans leur travaux de l'interprétation de la sécurité commercial que Lipner donnait dans ses modèles de sécurité et d'intégrité en treillis, et ont conclu que le modèle commercial de Lipner ne traitait pas correctement la limitation de la manipulation des données à des programmes spécifiques pour implanter des transactions bien définies.

Karger (Karger, P.A., Implementing Commercial Data Integrity with Secure Capabilities. Symposium IEEE 1988 sur la sécurité et la confidentialité, 18 au 21 avril 1988. Oakland, CA : IEEE Computer Society, pp. 130-139) a proposé une application du modèle de sécurité commercial de Clark et Wilson et a montré comment un modèle à capacité restreinte combiné avec un modèle de sécurité en treillis pouvait faciliter cette application. Son article explique aussi pourquoi le modèle de sécurité de Clark et Wilson peut poser des problèmes beaucoup plus difficiles à résoudre que les modèles de sécurité en treillis qui sont relativement simples. Dans l'application, les analyses rétrospectives jouent un rôle beaucoup plus actif quant à la sécurité que sur les systèmes antérieurs. En particulier, des décisions de contrôle d'accès sont basées sur des informations historiques retrouvées par analyse rétrospective, ainsi que sur des règles descriptives de qui peut avoir accès à quoi. Cependant, le besoin d'analyses rétrospectives historiques peut rendre un tel modèle impossible à implanter sur une carte à mémoire du fait de l'extrême faiblesse de la mémoire disponible pour contenir les données d'analyse.

Un problème difficile à résoudre dans le domaine de la sécurité est celui de la lutte contre le déni de service. En effet, il n'existe pas de bonne définition de ce qu'est un déni de service. De plus, on peut estimer que détecter et empêcher un déni illégal de service revient à résoudre le

problème de l'arrêt de la machine de Turing. Divers systèmes ont été conçus pour allouer des quotas et limiter les dépenses de ressources dans les systèmes informatiques, mais aucun ne s'est attaqué au déni illégal de service qui peuvent se présenter sous forme de chevaux de Troie ou de pièges. Si les modèles d'intégrité peuvent être d'une certaine utilité, le déni de service reste un important problème non résolu pour la sécurité des ordinateurs.

OBJET DE L'INVENTION

L'objet de la présente invention est de fournir une méthode et un appareil pour contrôler l'accès aux fichiers par des programmes d'appel, les fichiers contenant d'autres fichiers, des programmes et des données. La méthode comprend les étapes d'assignation d'une classe d'accès initiale à chaque fichier et à chaque programme d'appel. Une classe d'accès comprend une classe d'intégrité et une classe de confidentialité. Une classe d'intégrité comprend des règles définissant la modification des données contenues dans les fichiers, et une classe de confidentialité comprend des règles définissant la divulgation des données contenues dans les fichiers. Une classe d'intégrité comprend une série de règles pour définir l'exécution d'une fonction de lecture et une autre série de règles pour définir l'exécution d'une fonction d'écriture/exécution. Une fonction d'exécution comprend des fonctions de transfert et de chaînage, la fonction de chaînage comprenant la capacité de lancer une autre opération dotée de classes de confidentialité et d'intégrité potentiellement différentes. Une classe de confidentialité comprend une série de règles pour définir l'exécution d'une fonction d'écriture et une autre série de règles pour définir l'exécution d'une fonction de lecture/exécution. Les classes d'accès respectives du fichier cible, du programme cible et du programme d'appel sont comparées. Si les résultats de la comparaison

correspondent aux exigences de sécurité de la fonction concernée, cette fonction est exécutée.

BREVE DESCRIPTION DES FIGURES

On comprendra mieux l'invention à la lecture de la description détaillée suivante à l'aide des figures d'accompagnement.

La figure 1 est un schéma illustrant un exemple d'application du dispositif de sécurité utilisant deux systèmes informatiques 105 et 110 comprenant chacun une unité de lecture/écriture 115 et une unité de stockage (telle qu'une carte à mémoire) 120.

La figure 2 est un schéma illustrant un exemple de version comprenant une compagnie aérienne 205, deux hôtels 210 et 215 et deux agences de location de voitures 220 et 225, un système de carte de fidélité appliquant le dispositif de sécurité avec une carte à mémoire 230.

La figure 3 est un schéma fonctionnel illustrant un exemple de version des éléments du dispositif de sécurité.

La figure 4 est un schéma fonctionnel illustrant un exemple de version de l'opération d'initialisation.

La figure 5 est un schéma fonctionnel illustrant un exemple de version du dispositif de sécurité.

Les figures 6 à 9 sont des schémas fonctionnels illustrant le fonctionnement d'un exemple de version de la présente invention.

DESCRIPTION DE L'INVENTION

Un exemple de version de la présente invention est constitué par un dispositif de sécurité assurant la protection contre les utilisateurs non autorisés et l'altération des informations. Ce résultat est obtenu en partie en contrôlant l'accès des utilisateurs et des programmes aux fichiers et aux autres programmes. On notera que, dans cette description, les fichiers comprennent des données et des programmes.

Deux classes d'accès sont décrites ici : une classe de confidentialité et une classe d'intégrité. Par confidentialité, on entend la protection des informations contenues dans des fichiers ou des programmes contre l'accès non autorisé par des utilisateurs ou d'autres programmes. Les règles utilisées pour déterminer les fichiers et programmes auxquels un utilisateur ou un programme peut avoir accès sont contenues dans la classe de confidentialité implantée. Par ailleurs, les règles définissant les différents types d'accès peuvent varier. Par exemple, un utilisateur doté de la classe de confidentialité "top secret" peut être autorisé à lire (avoir accès en lecture à) des fichiers et des programmes de classes de confidentialité "top secret", "secret" et "confidentiel", mais ne pouvoir accéder en écriture qu'à des fichiers et des programmes de classe de confidentialité "top secret". Cela permet d'éviter l'écriture accidentelle ou malintentionnée d'information top secret dans un fichier de classe de confidentialité "confidentiel" ou "secret".

L'autre classe de protection est la classe de protection d'intégrité. Par intégrité, on entend la protection d'informations contre l'altération, la modification non autorisée ou le sabotage au moyen par exemple d'un virus informatique ou d'un cheval de Troie. Les règles utilisées pour définir la protection d'intégrité sont contenues dans la classe d'intégrité implantée.

Les types d'accès décrits ici comprennent l'écriture, la lecture, le transfert de contrôle et le chaînage. Un transfert de contrôle a lieu lorsqu'un programme exécute une instruction de branchement, de transfert ou d'appel de sous-routine. Un programme de chaînage lance une autre opération dotée de classes de confidentialité et d'intégrité potentiellement différentes et interrompt l'opération en cours. Dans cette description, une fonction d'exécution comprend les fonctions de transfert et de chaînage.

Comme indiqué dans la description ci-dessus, l'accès est défini par la comparaison de classes d'accès et l'application de règles de sécurité aux résultats de la comparaison des classes d'accès. Dans une version de ce dispositif de sécurité, différentes séries de règles sont assignées à la classe de confidentialité en écriture, à la classe de confidentialité en lecture/exécution, à la classe d'intégrité en lecture et à la classe d'intégrité en écriture/exécution.

La figure 1 est un schéma illustrant un exemple d'application du dispositif de sécurité utilisant deux systèmes informatiques 105 et 110 comprenant chacun une unité de lecture/écriture (telle qu'une unité de lecture/écriture de carte à mémoire) 115 et une unité de stockage (telle qu'une carte à mémoire) 120. Des protocoles de communication pour les cartes à mémoire sont décrits, par exemple, dans Rankl, W. et al., Smart Card Handbook, John Wiley & Sons, 1997. A chaque système informatique de la figure 1 sont associées une classe de confidentialité et une classe d'intégrité. Chaque système informatique possède une unité de lecture/écriture de carte à mémoire 115 dans laquelle la carte à mémoire est insérée. Cette unité de lecture/écriture de carte à mémoire 115 constitue une exemple de mécanisme de communication entre la carte à mémoire 120 et les systèmes informatiques 105 et 110. La connexion entre l'unité de lecture/écriture de carte à mémoire 115 et les systèmes informatiques 105 et 110 peut

prendre différentes formes telles qu'une connexion câblée, une ligne téléphonique, Internet ou une liaison en radiofréquence (RF). De même, la connexion entre la carte à mémoire 120 et l'unité de lecture/écriture 115 peut prendre différentes formes telles que l'insertion de la carte dans l'unité ou une liaison en radiofréquence (RF). Si la connexion entre la carte à mémoire 120 et l'unité de lecture/écriture 115 se fait par une liaison en radiofréquence, la carte à mémoire est dite sans contact. Des fichiers et des programmes contenant des informations peuvent être écrits et lus sur la carte à mémoire 120 par les systèmes informatiques 105 et 110. A chaque fichier ou programme sont associées une classe de confidentialité et une classe d'intégrité. Les systèmes informatiques échangent entre eux les informations contenues dans ces fichiers et ces programmes via la carte à mémoire si leurs classes d'accès respectives correspondent aux règles de sécurité de l'opération en question. Les informations sont transmises aux utilisateurs en fonction des règles de chaque classe de confidentialité. La protection contre l'altération, la modification non autorisée d'informations et le sabotage dépend des règles de chaque classe d'intégrité.

Par exemple, supposons que le système informatique A 105 fait office d'administrateur de système. En tant qu'administrateur de système, le système informatique A 105 a l'autorité de modifier les classes de confidentialité et d'intégrité. Supposons que le système informatique B 110 n'a pas l'autorité de modifier une classe de confidentialité ou d'intégrité. Supposons aussi la hiérarchie suivante entre les classes de confidentialité : (1) un système informatique doté de la classe de confidentialité "top secret" a accès aux fichiers et programmes de trois classes de confidentialité (confidentiel, secret et top secret), (2) un système informatique doté de la classe de confidentialité "secret" a accès aux fichiers et programmes de classes de confidentialité "confidentiel" et "secret" et (3) un système informatique doté de la classe de

confidentialité "confidentiel" a accès aux fichiers et programmes de classes de confidentialité "confidentiel". Pour illustrer notre propos, aucune distinction n'est faite entre écriture, lecture et exécution, bien qu'une version de la présente invention contienne cette distinction et que celle-ci soit considérée comme une caractéristique importante. De même, pour illustrer notre propos, aucune catégorie n'est mentionnée, bien qu'une version de la présente invention contienne des catégories et que celles-ci soient également considérées comme une caractéristique souhaitable. Supposons également qu'au système informatique A 105 est assignée la classe de confidentialité "top secret" et au système informatique B 110 la classe de confidentialité "secret". Supposons qu'au départ, tous les fichiers et programmes résidant sur la carte à mémoire ont la même classe de confidentialité "confidentiel". Supposons aussi qu'aux fichiers et programmes résidant initialement sur le système informatique A 105 est assignée la classe de confidentialité "top secret" et aux fichiers et programmes résidant initialement sur le système informatique B 110 la classe de confidentialité "secret". Supposons enfin que la classe d'intégrité assignés à tous les systèmes informatiques, fichiers et programmes est telle qu'elle les protège contre tous les virus et chevaux de Troie connus. Si aucun virus ou cheval de Troie n'est présent et si les hypothèses ci-dessus sont respectées, les systèmes informatiques 105 et 110 ont accès aux informations de la carte à mémoire 120. Le système informatique A 105 a accès à tous les fichiers et programmes de la carte à mémoire 120 écrits par le système informatique B 110 car le système informatique A 105 a une classe de confidentialité "top secret" et les fichiers écrits par le système informatique B 110 ont une classe de confidentialité "secret". Le système informatique B 110 n'a pas accès aux fichiers de la carte à mémoire 120 écrits par le système informatique A 105. En tant qu'administrateur du système, le système informatique A 105 peut modifier (élever ou abaisser)

la classe de confidentialité d'un fichier ou programme résidant sur la carte à mémoire 120. Cela peut permettre au système informatique B 110 d'accéder à un fichier de la carte à mémoire 120 généré par le système informatique A 105. Comme cet exemple le montre, l'accès aux fichiers et programmes est défini par la comparaison des classes d'accès et l'application des règles de sécurité.

La figure 2 est un schéma illustrant un exemple de version comprenant une compagnie aérienne 205, deux hôtels 210 et 215 et deux agences de location de voitures 220 et 225, un système de carte de fidélité appliquant le dispositif de sécurité avec une carte à mémoire 230. Supposons que la compagnie aérienne A 205 a des liens avec les hôtels H 210 et M 215 et les agences de location de voitures B 220 et D 225. Un voyageur séjournant dans un des hôtels gagne des points de fidélité de la compagnie aérienne. L'hôtel H 210 donne des points de fidélité d'hôtel en plus de ceux de la compagnie aérienne, tandis que l'hôtel M 215 donne des points de fidélité d'hôtel ou de compagnie aérienne, mais pas les deux. Les points de fidélité de l'hôtel H 210 et ceux de l'hôtel M 215 constituent des systèmes complètement séparés. En outre, les hôtels considèrent les informations sur les dates et les lieux de séjour des clients comme des informations importantes car ils peuvent les utiliser à des fins de marketing. Cependant, les clients et la compagnie aérienne aimeraient que les trois systèmes de points de fidélité soient gérés avec une seule carte à mémoire 230, de manière à ce que les clients n'aient besoin de porter qu'une seule carte. Les hôtels H 210 et M 215 ne se font pas confiance mais aimeraient tous deux collaborer avec la compagnie aérienne A 205.

A partir de ces hypothèses, on comprendra que le logiciel gérant les points de fidélité de l'hôtel H 210 doit fonctionner différemment du logiciel gérant les points de fidélité de l'hôtel M 215. En effet, les hôtels ont des

politiques différentes en ce qui concerne l'attribution des points d'hôtel et de compagnie aérienne. En outre, les logiciels des hôtels et de la compagnie aérienne doivent être réactualisés périodiquement pour tenir compte des offres spéciales limitées dans le temps, des nouveaux partenaires ou des modifications importantes. La transmission de ces informations peut se faire à l'aide de la carte à mémoire 230 elle-même, plutôt que de demander aux serveurs centraux de la compagnie aérienne 205, des hôtels 210 et 215 et des sociétés de location de voitures 220 et 225 de tous communiquer les uns avec les autres. Ainsi, les fichiers de données concernant les systèmes de fidélité de chaque entreprise peuvent résider sur la carte à mémoire 230.

Au départ, le client se rend à l'aéroport pour prendre un vol de la compagnie aérienne A 205 et gagne des points de fidélité. Ils sont enregistrés dans un fichier auquel seule la compagnie aérienne a accès. La compagnie désire indiquer que le client a pris un vol ce jour-là, mais elle peut ne pas souhaiter donner de détails complets sur ce vol, pour préserver la confidentialité de ses activités et la vie privée du client. C'est pourquoi l'application de la compagnie aérienne écrit sur un fichier différent, accessible en lecture à ses partenaires, pour indiquer que le client a pris un vol ce jour-là, mais sans autre détail. L'application de n'importe quel partenaire peut lire cette information (en supposant qu'il y ait une seule compagnie aérienne). Ensuite, le client loue une voiture à la société B 220. Le code d'application de la compagnie B calcule le nombre de points de fidélité de compagnie aérienne qu'il doit accorder; cette société désire communiquer cette information à la compagnie aérienne, mais sans la révéler à la société de location de voitures D 225. Pour cela, la société B 220 écrit le nombre de points gagnés dans un fichier de communication. Elle modifie ensuite la classification de ce fichier pour qu'il soit accessible à la compagnie aérienne A 205 et à elle-même, mais pas à la société

D 225. Cependant, la modification de la classification de ce fichier est une opération dangereuse. Si le code d'application de la société B contient des défauts, ou pire, un cheval de Troie ou un virus, une information erronée peut être transmise à la compagnie aérienne A 205 ou la classification peut être modifiée de manière incorrecte et permettre son accès à la société D 225. Le programme qui exécute la modification de la classification doit avoir un haut niveau d'intégrité pour garantir que de telles erreurs accidentelles ou délibérées ne se produisent pas. Cet exemple illustre simplement une application pratique de l'accès aux fichiers et de la définition des programmes par la classe d'accès.

La figure 3 est un schéma fonctionnel illustrant un exemple de version des éléments du dispositif de sécurité. Cet exemple de version du dispositif de sécurité 305 comprend une classe de confidentialité 310 et une classe d'intégrité 315. Chacune de ces classes comprend des règles spécifiques. Par confidentialité, on entend la protection contre la divulgation des informations à des utilisateurs non autorisés. Par intégrité, on entend la protection des informations contre l'altération, les modifications non autorisées ou le sabotage (par exemple, cheval de Troie ou virus).

Dans la version donnée en exemple, une classe de confidentialité 310 comprend deux ensembles de règles. Un ensemble de règles 325 gouverne la fonction d'écriture; on l'appelle la classe d'accès en écriture. L'autre ensemble de règles 330 gouverne les fonctions de lecture et d'exécution, l'exécution comprenant le transfert et le chaînage. Dans la présente description, dans la situation où un programme tente d'accéder à un fichier ou un autre programme, le fichier visé sera appelé fichier cible et le programme visé programme cible. Une classe d'intégrité 315 comprend deux ensembles de règles. Un ensemble de règles 345 gouverne la fonction de lecture; on l'appelle la classe d'accès en lecture. L'autre

ensemble de règles 350 gouverne les fonctions d'écriture et d'exécution, la fonction d'exécution comprenant le transfert et le chaînage.

La version donnée en exemple du dispositif de sécurité comprend les règles suivantes :

Pour la permission de lecture des classes d'intégrité : la classe d'intégrité en lecture du programme d'appel doit être inférieure ou égale à la classe d'intégrité du fichier cible ou du programme cible.

Pour la permission d'écriture/exécution des classes d'intégrité : la classe d'intégrité en écriture/exécution du programme d'appel doit être supérieure ou égale à la classe d'intégrité du fichier cible ou du programme cible.

Pour la permission de lecture/exécution des classes de confidentialité : la classe de confidentialité en lecture/exécution du programme d'appel doit être supérieure ou égale à la classe de confidentialité du fichier cible ou du programme cible.

Pour la permission d'écriture des classes de confidentialité : la classe de confidentialité en écriture du programme d'appel doit être inférieure ou égale à la classe de confidentialité du fichier cible ou du programme cible.

Pour la permission d'exécution de transfert : la classe de confidentialité en lecture/exécution du programme d'appel doit être supérieure ou égale à la classe de confidentialité du fichier cible ou du programme cible, et la classe d'intégrité en écriture/exécution du programme d'appel doit être inférieure ou égale à la classe d'intégrité du fichier cible ou du programme cible.

Pour la permission d'exécution de chaînage : la classe de confidentialité en lecture/exécution du programme d'appel doit être supérieure ou égale à la classe de confidentialité du fichier cible ou du programme cible, la classe de confidentialité en écriture du programme d'appel doit être inférieure ou égale à la classe de confidentialité en lecture/écriture de la nouvelle opération, et la classe d'intégrité en écriture/exécution du programme d'appel doit être supérieure ou égale à la classe d'intégrité en lecture de la nouvelle opération.

Pour la permission de transfert, le programme cible tourne au niveau de l'intégrité du programme d'appel. Un programme à haut niveau d'intégrité ne peut pas appeler ou effectuer un transfert sur un fichier doté d'un niveau d'intégrité plus faible. Pour la permission de chaînage, la première règle garantit que le chaînage n'est possible qu'avec des fichiers pour lesquels le programme d'appel a la permission de confidentialité en lecture. La permission d'intégrité en lecture n'est pas nécessaire. La deuxième règle garantit que l'opération cible doit avoir la permission de confidentialité en lecture pour tous les arguments communiqués, et la troisième règle garantit que l'opération cible n'est pas contaminée par un argument de faible intégrité.

La figure 4 est un schéma fonctionnel illustrant un exemple de version de l'opération d'initialisation. Au départ, une classe d'accès est attribuée à chaque fichier par un administrateur de système (405). Dans l'exemple précédent, la compagnie aérienne 205 assignerait une classe d'accès initiale à chacun de ses fichiers et à chaque fichier qui doit être utilisé par les hôtels 210 et 215 et les agences de location de voitures 220 et 225. Puisqu'une classe d'accès comprend une classe de confidentialité et une classe d'intégrité, comme indiqué sur la figure 3, l'assignation initiale d'une classe d'accès à

chaque fichier comprend l'assignation de classes de confidentialité et d'intégrité initiales à chaque fichier. Ces fichiers peuvent contenir des données et des programmes. Une classe d'accès initiale est également assignée à chaque programme d'appel par un administrateur de système (410). Dans l'exemple précédent, la compagnie aérienne assignerait une classe d'accès initiale à chacun de ses programmes d'appel et à chaque programme d'appel qui doit être utilisé par les hôtels 210 et 215 et les agences de location de voitures 220 et 225. Les programmes d'appel comprennent des programmes qui appellent des fichiers et des programmes qui appellent d'autres programmes. Puisqu'une classe d'accès comprend une classe de confidentialité et une classe d'intégrité, comme indiqué sur la figure 3, l'assignation initiale d'une classe d'accès à chaque programme d'appel comprend l'assignation de classes de confidentialité et d'intégrité initiales à chaque programme d'appel. L'assignation initiale des classes d'accès peut être faite dans n'importe quel ordre, ou simultanément.

Décrivons maintenant la suite du fonctionnement de la version représentée sur la figure 2. Comme nous l'avons vu, la société B 220 désire modifier la classification d'un fichier de communication pour permettre son accès à la compagnie aérienne A 205 mais pas à la société D 225. Elle désire aussi s'assurer que les informations écrites dans ce fichier sont UNIQUEMENT les informations destinées à la compagnie aérienne A 205, et non d'autres informations que la compagnie A 205 n'a pas à connaître. Les figure 6, 7, 8 et 9 montrent les étapes de la procédure d'une approche permettant d'utiliser le modèle de sécurité pour atteindre ces objectifs. Les schémas fonctionnels de ces figures montrent les programmes et les fichiers de données utilisés par la société B 220 pour reclasser les informations et les transmettre à la compagnie aérienne A 205.

Les figures 6 à 9 représentent toutes les mêmes objets (programmes et fichiers) numérotés x10, x20, x30, x40 et x50, x étant une valeur de 6 à 9. Ces figures ont pour objet de montrer l'évolution dans le temps de ces objets.

La figure 6 montre le programme d'application 610 de la société B qui calcule le nombre de points de fidélité et écrit cette information dans le fichier de communication 620. Le programme d'application 610 est autorisé à accéder en lecture et en écriture aux informations dans la classe de confidentialité B et a un niveau d'intégrité minimum.

La figure 7 montre le programme d'application 710 de la société B qui élève la classe d'accès du fichier de communication 720 aux classes A et B, puis exécute une opération de CHAINAGE au programme de garde 730. Le programme de garde 730 est autorisé à accéder en lecture et en écriture aux informations des classes de confidentialité A et B et a un niveau d'intégrité de E4; il est autorisé à limiter les informations à la seule classe de confidentialité A.

La figure 8 montre le programme de garde 830 qui lit les informations du fichier de communication 820 dans les classes de confidentialité A et B et les limite à la seule classe de confidentialité A en les écrivant dans le fichier de communication 840. Le programme de garde 830 a un niveau d'intégrité élevé de E4; il est donc autorisé à d'abord inspecter les données du fichier 820 pour vérifier qu'il contient uniquement des informations sur les points de fidélité et que la valeur des points de fidélité est raisonnable. Ce n'est qu'après avoir effectué ces contrôles que le programme de garde écrit les données dans le fichier 840. Le programme de garde 830 effectue ensuite une opération de CHAINAGE avec le programme d'application 850 de la compagnie aérienne A.

La figure 9 montre le programme d'application 950 de la compagnie aérienne A qui est autorisé à lire et écrire des informations uniquement dans la classe de confidentialité A. Le programme d'application 950 lit les informations de fidélité dans le fichier de communication 940 et attribue le nombre approprié de points de fidélité.

Le processus représenté sur les figure 6 à 9 est donné uniquement à titre d'exemple, et les spécialistes comprendront que ce modèle de sécurité peut être utilisé de nombreuses autres manières pour contrôler la transmission d'informations d'une classe d'accès à une autre.

Dans une autre version du dispositif de sécurité, le système peut appliquer des dispositifs de sécurité discrétionnaires qui contiennent des règles qui doivent également être respectées avant que l'accès soit autorisé (415). Cela donne à l'utilisateur un contrôle discrétionnaire sur l'accès, dans la mesure où aussi bien les règles de contrôle d'accès discrétionnaire que les règles de classes de confidentialité et d'intégrité obligatoires s'appliquent. Ainsi, l'utilisateur a le pouvoir discrétionnaire de refuser l'accès même si les règles de confidentialité et d'intégrité sont respectées, mais il ne peut pas autoriser l'accès si les règles de confidentialité ou d'intégrité ne sont pas respectées. Parmi les dispositifs discrétionnaires, citons les listes d'accès de contrôle, les bits de permission, les listes de capacité, etc..

Dans une autre version du dispositif de sécurité, l'administrateur du système assigne une classe d'intégrité basée sur les résultats d'une procédure d'évaluation externe indépendante (420). Voici quelques exemples de procédures d'évaluation externe indépendante : les ITSEC (Critères d'évaluation de sécurité informatique (Information Technology Security Evaluation Criteria - ITSEC), juin 1991, Commission

des Communautés Européennes : Bruxelles, Belgiques) de niveaux E et les critères communs (Informatique - Techniques de sécurité - Critères d'évaluation de sécurité informatique, ISO/CEI 15408-1, 15408-2 et 15408-3, 1999) de niveaux EAL. L'évaluation externe indépendante permet à un administrateur de système d'assigner à un fichier ou un programme une classe d'intégrité plus élevée qu'il ne serait possible sans cette évaluation car le fichier ou le programme est considéré comme plus fiable.

La figure 5 est un schéma fonctionnel illustrant une version du dispositif de sécurité. Avant que l'accès soit autorisé, les classes d'accès du fichier cible ou du programme cible et du programme d'appel sont comparées (505). L'accès à un fichier cible ou un programme cible est déterminé en fonction des règles associées aux classes d'accès respectives. Si les classes d'accès respectives ne sont pas conformes aux règles de sécurité, l'accès est refusé (530). Si les classes d'accès respectives sont conformes aux règles de sécurité, il faut déterminer si un contrôle d'accès discrétionnaire est nécessaire (510). Si aucun contrôle d'accès discrétionnaire n'est nécessaire, la fonction est exécutée (520). Si un contrôle d'accès discrétionnaire est nécessaire, l'utilisateur doit déterminer si l'accès est autorisé (515). Si l'accès discrétionnaire est autorisé, la fonction est exécutée (520). Si l'accès discrétionnaire n'est pas autorisé, l'accès est refusé (530). Le contrôle d'accès discrétionnaire peut comprendre différents types de dispositifs tels que des listes de contrôle d'accès, des bits de permission, des listes de capacité et des dispositifs à base de rôles. Les fonctions (520) comprennent la lecture, l'écriture, le transfert et le chaînage. Les autorisations d'accès obligatoires et discrétionnaires peuvent être déterminées dans n'importe quel ordre ou simultanément.

Ensuite, la classe d'accès du fichier cible, du programme cible ou du programme d'appel est modifiée comme nécessaire (535). La modification comprend l'élévation ou l'abaissement de la classe d'accès. La modification est réalisée conformément aux règles de la classe d'accès correspondante. Dans une version du dispositif de sécurité, la modification de la classe de confidentialité d'accès d'un fichier cible ou d'un programme cible est basée sur la classe d'intégrité du programme d'appel. Par exemple, l'abaissement d'une classe de confidentialité peut être basé sur l'approche utilisée dans ce qu'on appelle les Yellow Books de la National Security Agency (Computer Security Requirements - Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments, CSC-STD-003-85, 25 juin 1985, DoD Computer Security Center : Ft. George G. Meade, MD. et Technical Rationale Behind CSC-STD-003-85 : Computer Security Requirements - Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments, CSC-STD-004-85, 25 juin 1985, DoD Computer Security Center : Ft. George G. Meade, MD). Les Yellow Books établissent la politique du Ministère de la Défense des Etats Unis en ce qui concerne les niveaux d'évaluation de sécurité nécessaires en fonction du risque. Cette approche examine le niveau de confidentialité maximum des données et le niveau de fiabilité minimum des utilisateurs pour déterminer une plage de risque sur laquelle le logiciel doit préserver la sécurité. Plus la plage de risque est élevée, plus le niveau d'évaluation doit être élevé. Si plusieurs fichiers ou programmes doivent être appelés, l'opération est répétée; sinon, l'opération s'arrête (540).

Bien qu'elle soit illustrée et décrite ci-dessus par rapport à certaines versions spécifiques, la présente invention ne doit pas être interprétée de manière limitative du fait des détails spécifiés. Au contraire, différentes modifications peuvent être apportées aux détails tout en respectant l'objet et la

portée des revendications ci-après ou de revendications équivalentes, et sans trahir l'esprit de l'invention.

REVENDICATIONS

1. Une méthode pour contrôler l'accès de fichiers par des programmes d'appel, ladite méthode comprenant les étapes suivantes :

a) assignation d'une classe d'accès initiale aux dits fichiers, étant entendu que

ladite classe d'accès comprend une classe d'intégrité et une classe de confidentialité,

ladite classe de confidentialité comprend des règles définissant la divulgation des données contenues dans les fichiers,

ladite classe d'intégrité comprend des règles définissant la modification des données contenues dans les fichiers,

ladite classe d'intégrité comprend deux éléments, à savoir un premier élément qui comprend des règles pour définir l'exécution d'une fonction de lecture et un second élément qui comprend des règles pour définir l'exécution d'écriture et d'exécution,

ladite classe de confidentialité comprend deux éléments, à savoir un premier élément qui comprend des règles pour définir l'exécution d'une fonction d'écriture et un second élément qui comprend des règles pour définir l'exécution de lecture et d'exécution,

b) assignation d'une classe d'accès initiale à chacun desdits programmes d'appel, chaque programme d'appel exécutant au moins une des fonctions d'écriture, de lecture et d'exécution,

c) comparaison de la classe d'accès de l'un desdits fichiers, ce dit fichier étant un fichier cible, avec la classe d'accès de l'un desdits programmes d'appel, ce dit programme d'appel étant un programme d'appel d'exécution, pour traiter ledit fichier cible conformément aux règles respectives de chaque classe d'accès,

d) permission au dit programme d'appel d'exécution de traiter ledit fichier cible si ladite comparaison aboutit à ce que ledit fichier cible et ledit programme d'appel d'exécution soient autorisés par les règles de sécurité,

e) modification de la classe d'accès dudit fichier cible conformément à au moins une des règles de ladite classe d'accès du programme d'appel d'exécution et des règles de ladite classe d'accès du fichier cible, ladite modification comprenant au moins l'abaissement ou l'élévation de la classe d'accès, et

f) modification de la classe d'accès dudit programme d'appel d'exécution conformément à au moins une des règles de ladite classe d'accès du programme d'appel d'exécution et des règles de ladite classe d'accès du fichier cible, ladite modification comprenant au moins l'abaissement ou l'élévation de la classe d'accès.

2. La méthode décrite dans la revendication 1, les étapes c à f étant répétées jusqu'à ce qu'il ne reste plus de fichiers cibles à traiter par les programmes d'appel d'exécution.

3. La méthode décrite dans la revendication 1, l'assignation de ladite classe d'intégrité comprenant une opération d'évaluation externe indépendante.

4. La méthode décrite dans la revendication 3, l'opération d'évaluation externe indépendante étant sélectionnée dans le

groupe constitué des ITSEC de niveau E et des critères communs de niveaux EAL.

5. La méthode décrite dans la revendication 1, l'étape de modification de ladite classe de confidentialité dudit fichier cible comprenant en outre l'évaluation de la classe d'intégrité du programme d'appel d'exécution et l'utilisation des résultats de ladite évaluation pour déterminer l'ampleur de ladite modification.

6. La méthode décrite dans la revendication 5, l'exécution comprenant le transfert et le chaînage, et le chaînage comprenant le lancement d'une autre opération.

7. La méthode décrite dans la revendication 1, ladite autorisation dudit programme d'appel d'exécution à traiter ledit fichier cible étant contrôlée indépendamment par un dispositif de contrôle d'accès discrétionnaire.

8. La méthode décrite dans la revendication 1, lesdits fichiers résidant sur une carte à mémoire.

9. Un système pour contrôler l'accès de fichiers, le système comprenant des moyens pour mettre en oeuvre la méthode selon l'une quelconque des revendications 1 à 8.

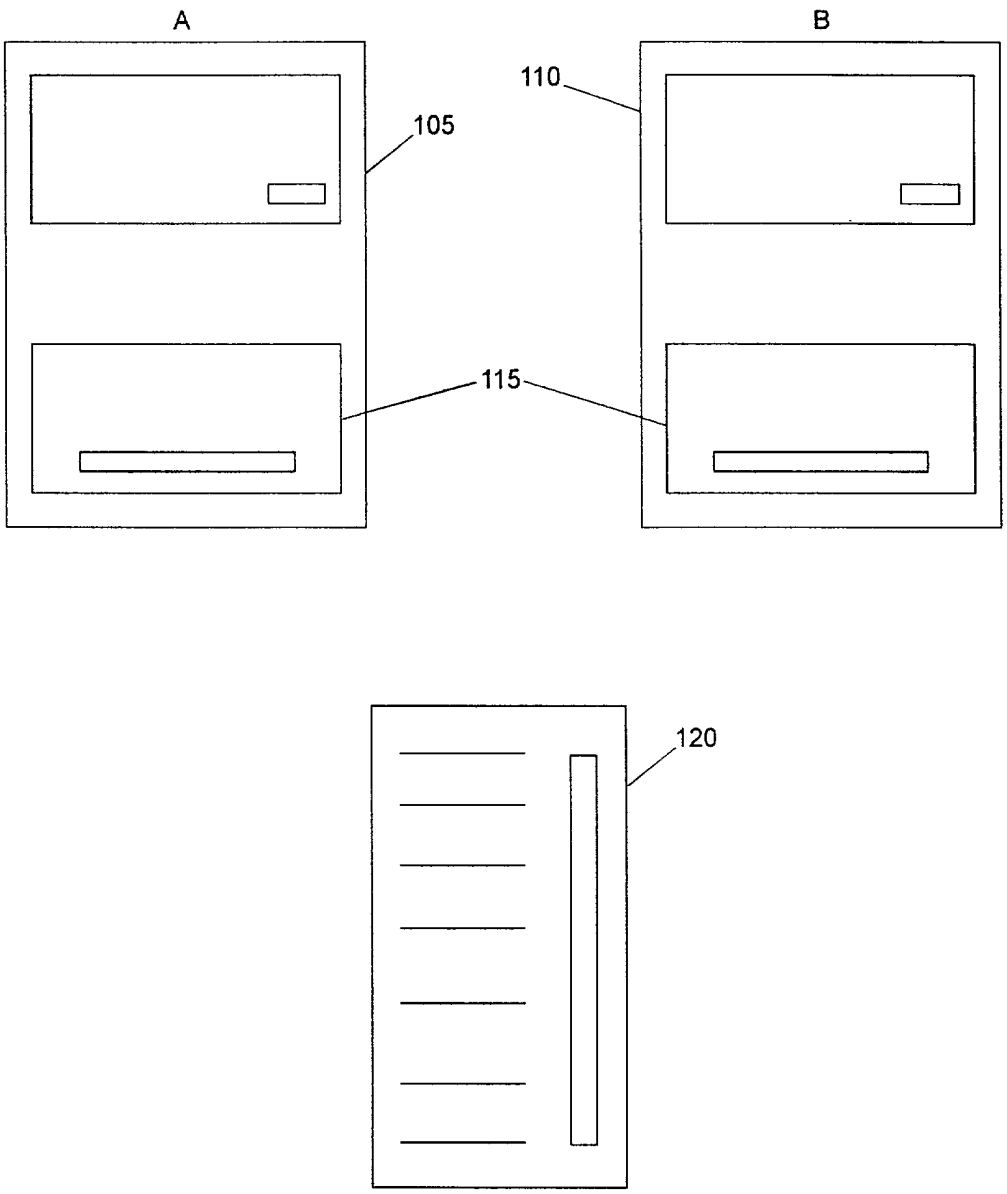


FIG. 1

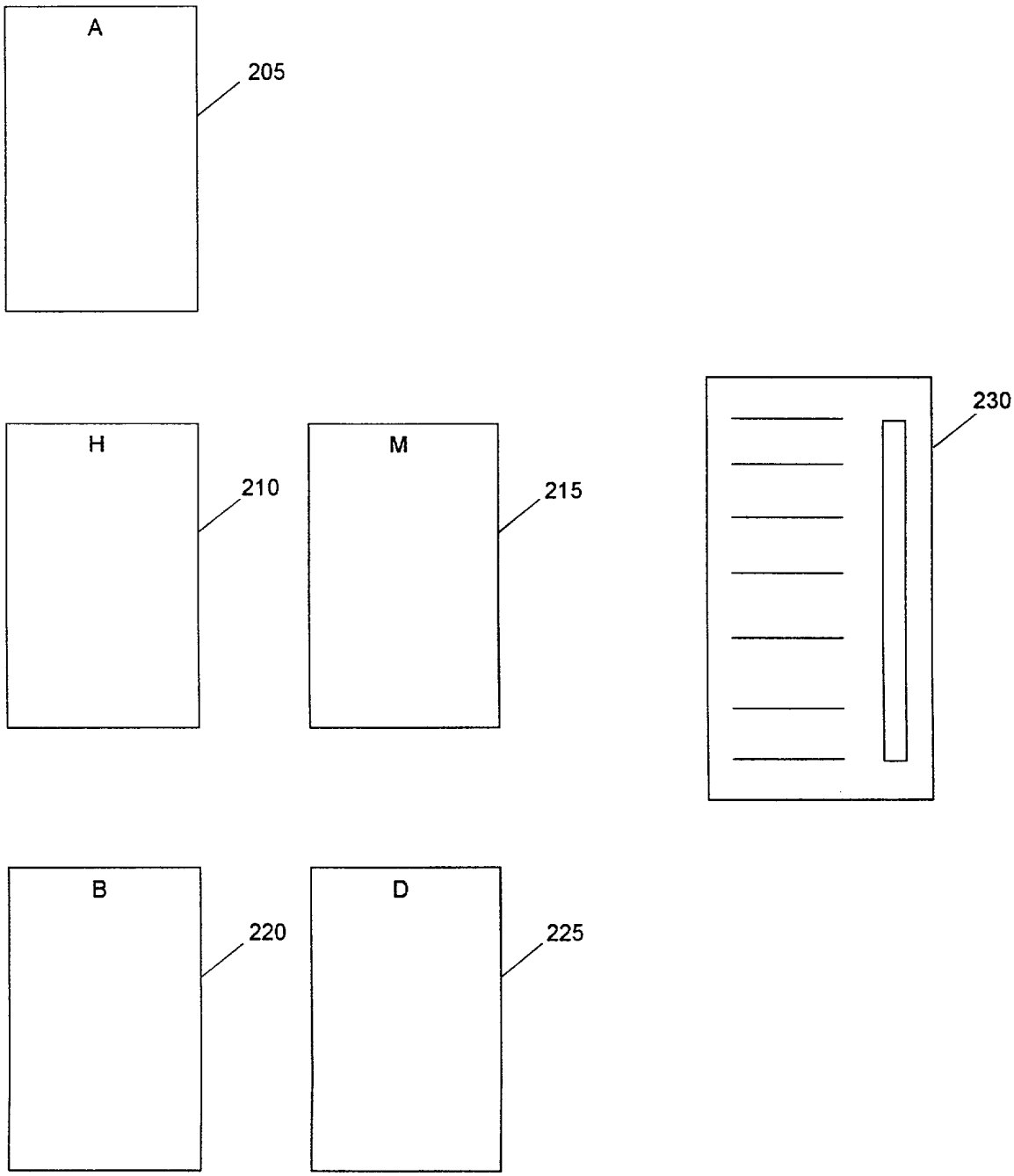


FIG. 2

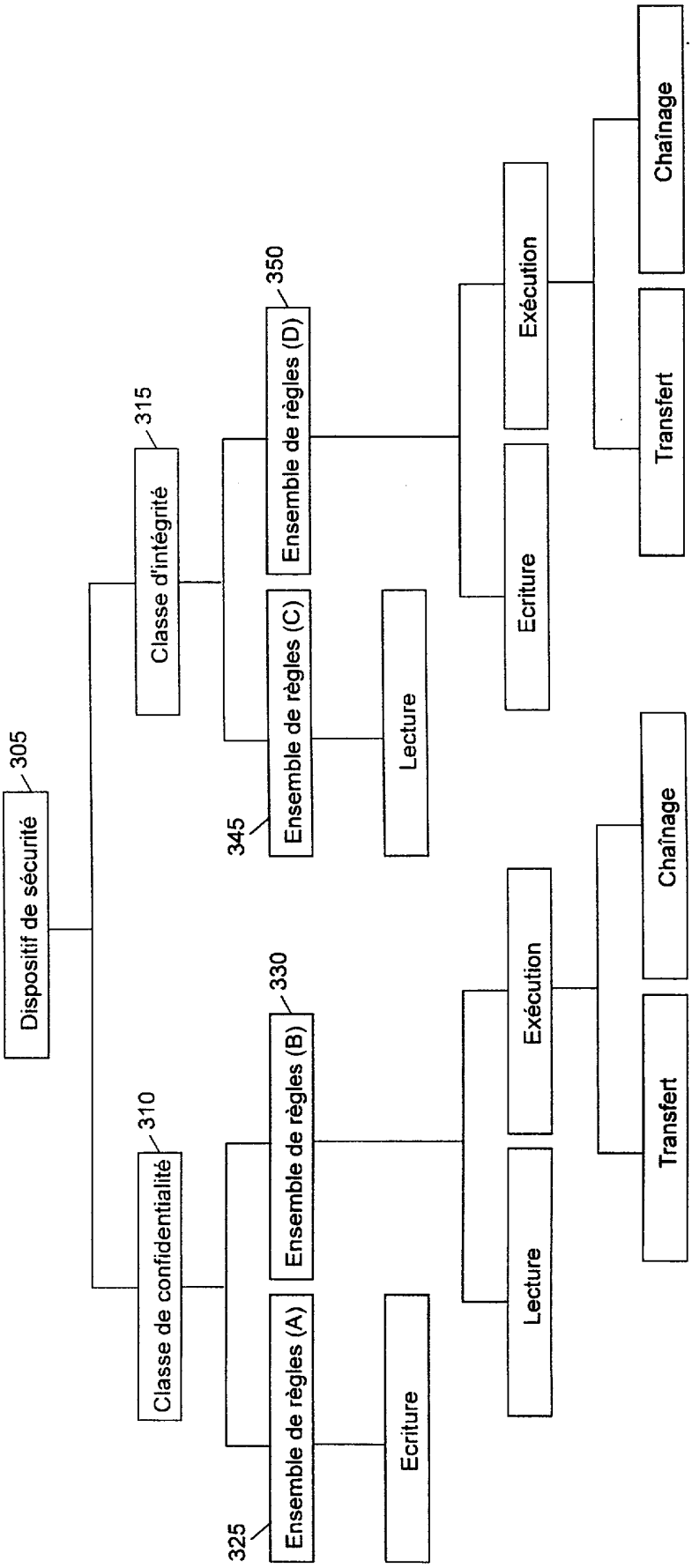

Dispositif de sécurité
305
Classe de confidentialité
310
Classe d'intégrité
315
Ensemble de règles (A)
325
Ensemble de règles (B)
330
Ensemble de règles (C)
345
Ensemble de règles (D)
350
Ecriture
Lecture
Exécution
Chaînage
Transfert
Ecriture
Lecture
Exécution
Chaînage
Transfert

FIG. 3

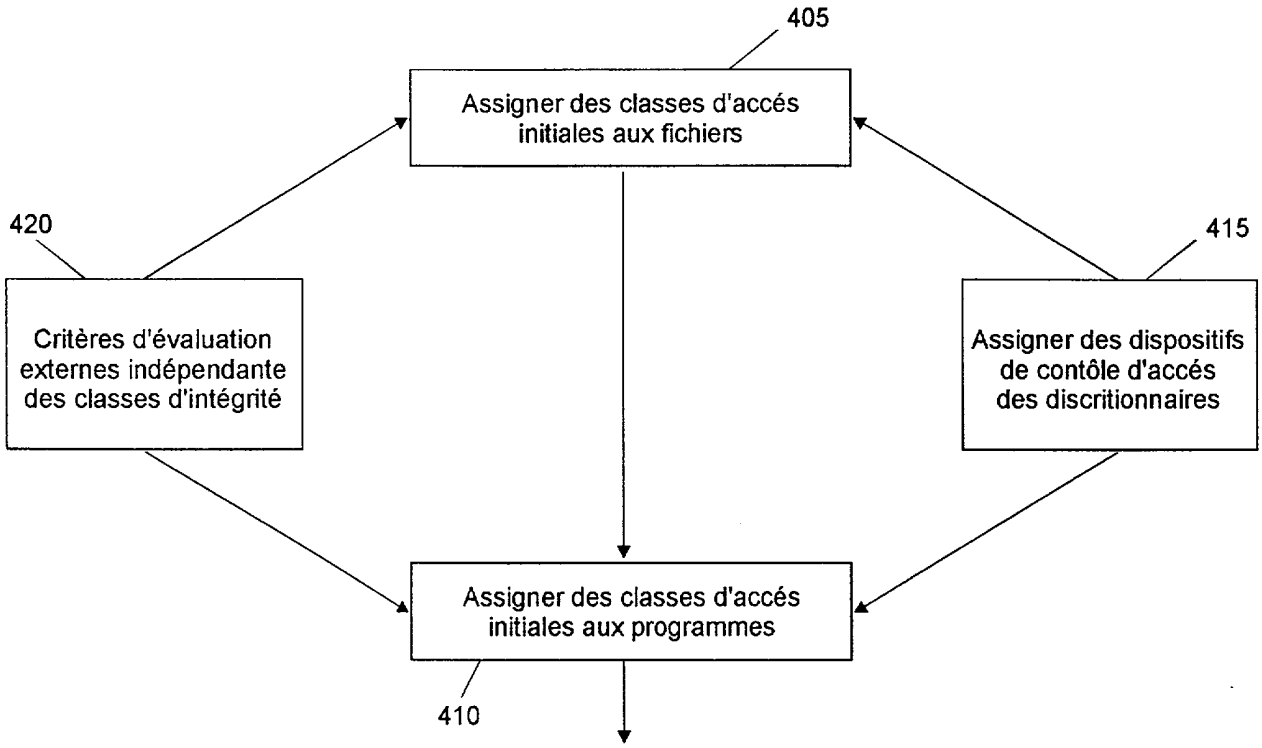

405
Assigner des classes d'accés initiales aux fichiers
420
Critères d'évaluation externes indépendante des classes d'intégrité
415
Assigner des dispositifs de contôle d'accés des discritionnaires
Assigner des classes d'accés initiales aux programmes
410

FIG. 4

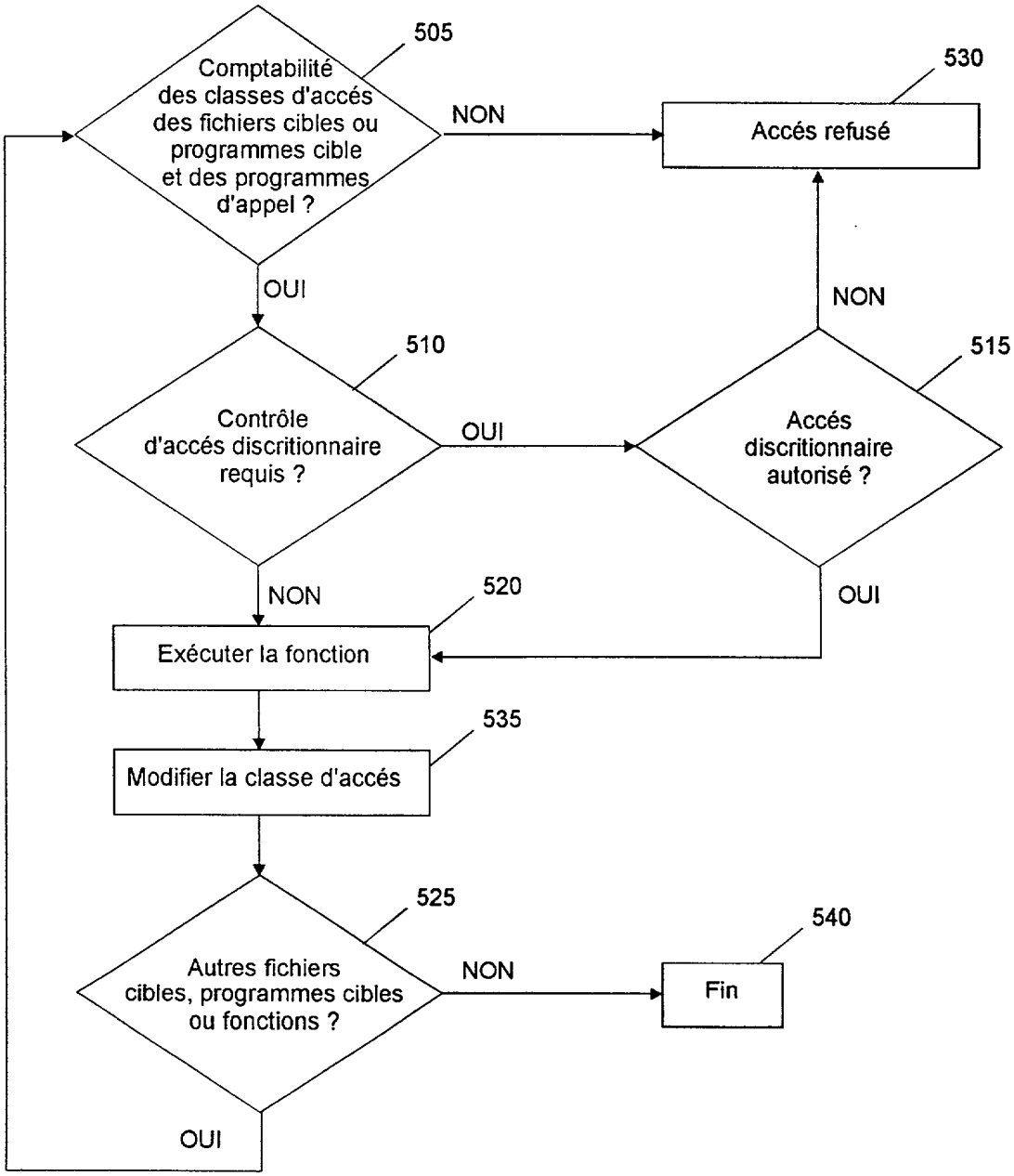

505
Comptabilité des classes d'accès des fichiers cibles ou programmes cible et des programmes d'appel ?
NON
530
Accès refusé
OUI
NON
510
Contrôle d'accès discritionnaire requis ?
OUI
515
Accès discritionnaire autorisé ?
NON
OUI
520
Exécuter la fonction
535
Modifier la classe d'accès
525
Autres fichiers cibles, programmes cibles ou fonctions ?
NON
540
Fin
OUI

FIG. 5

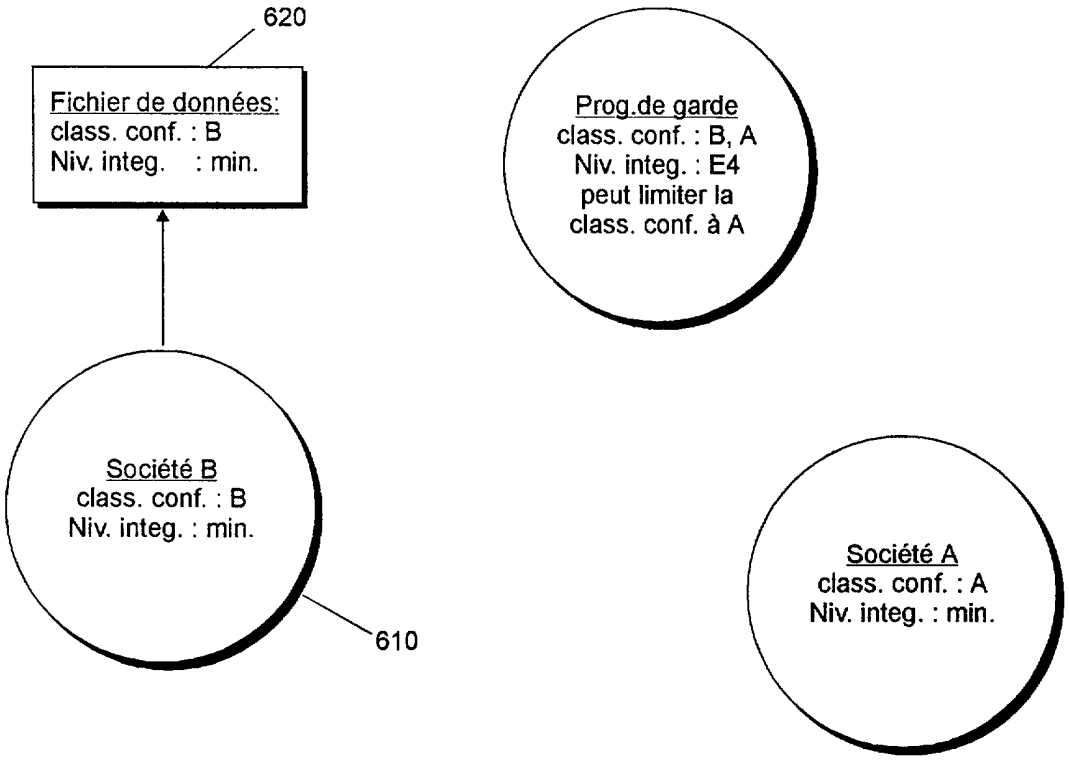

620
Fichier de données:
class. conf. : B
Niv. integ. : min.
Prog.de garde
class. conf. : B, A
Niv. integ. : E4
peut limiter la
class. conf. à A
Société B
class. conf. : B
Niv. integ. : min.
610
Société A
class. conf. : A
Niv. integ. : min.

FIG. 6

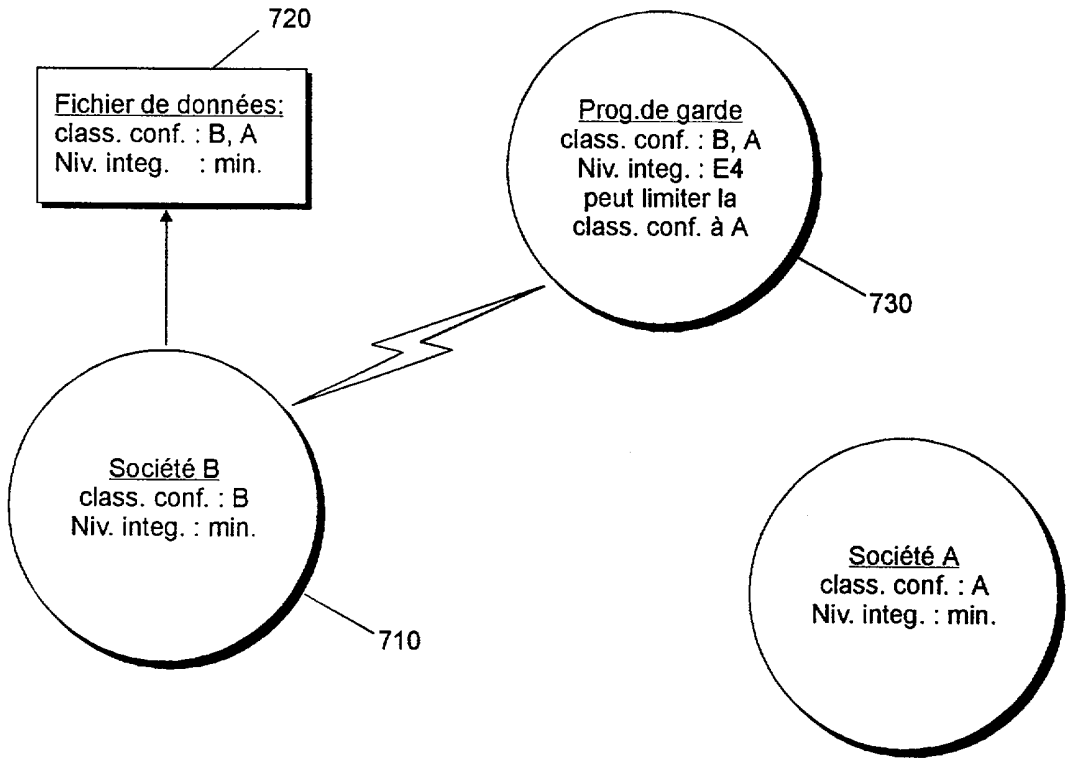

720
Fichier de données:
class. conf. : B, A
Niv. integ. : min.
Prog.de garde
class. conf. : B, A
Niv. integ. : E4
peut limiter la
class. conf. à A
730
Société B
class. conf. : B
Niv. integ. : min.
710
Société A
class. conf. : A
Niv. integ. : min.

FIG. 7

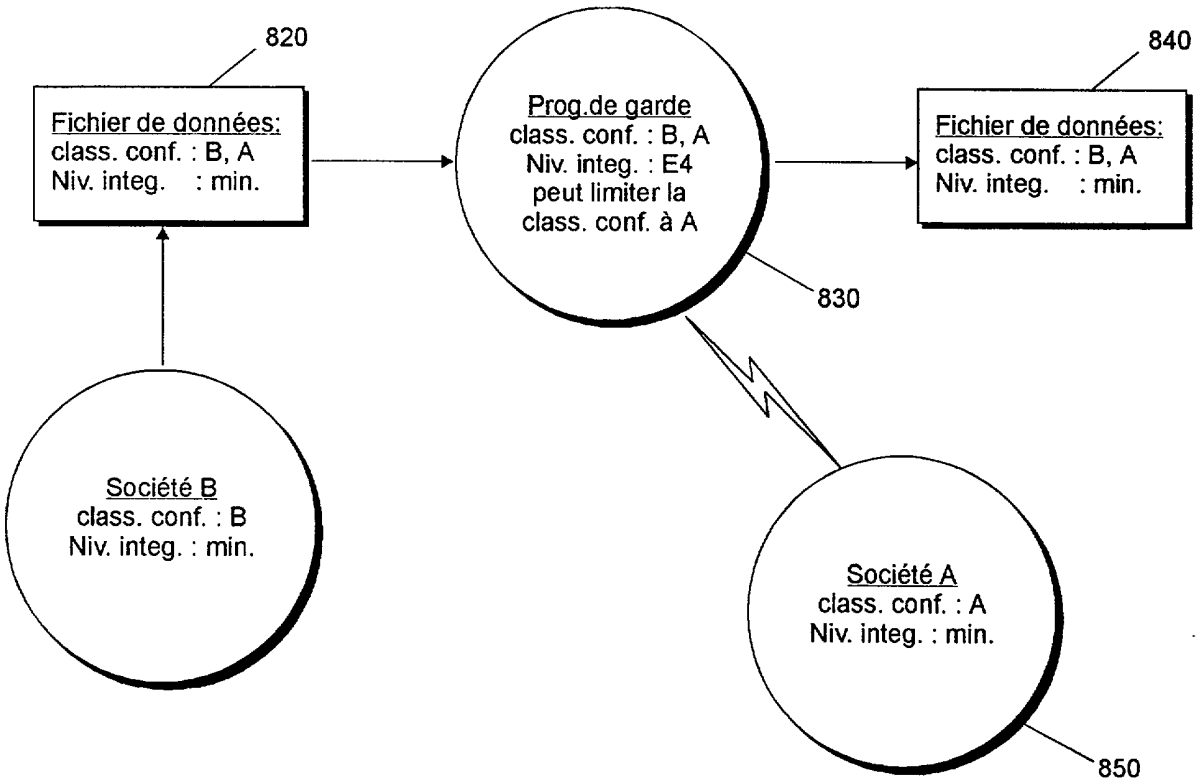

820
Fichier de données:
class. conf. : B, A
Niv. integ. : min.
Prog.de garde
class. conf. : B, A
Niv. integ. : E4
peut limiter la
class. conf. à A
830
840
Fichier de données:
class. conf. : B, A
Niv. integ. : min.
Société B
class. conf. : B
Niv. integ. : min.
Société A
class. conf. : A
Niv. integ. : min.
850

FIG. 8

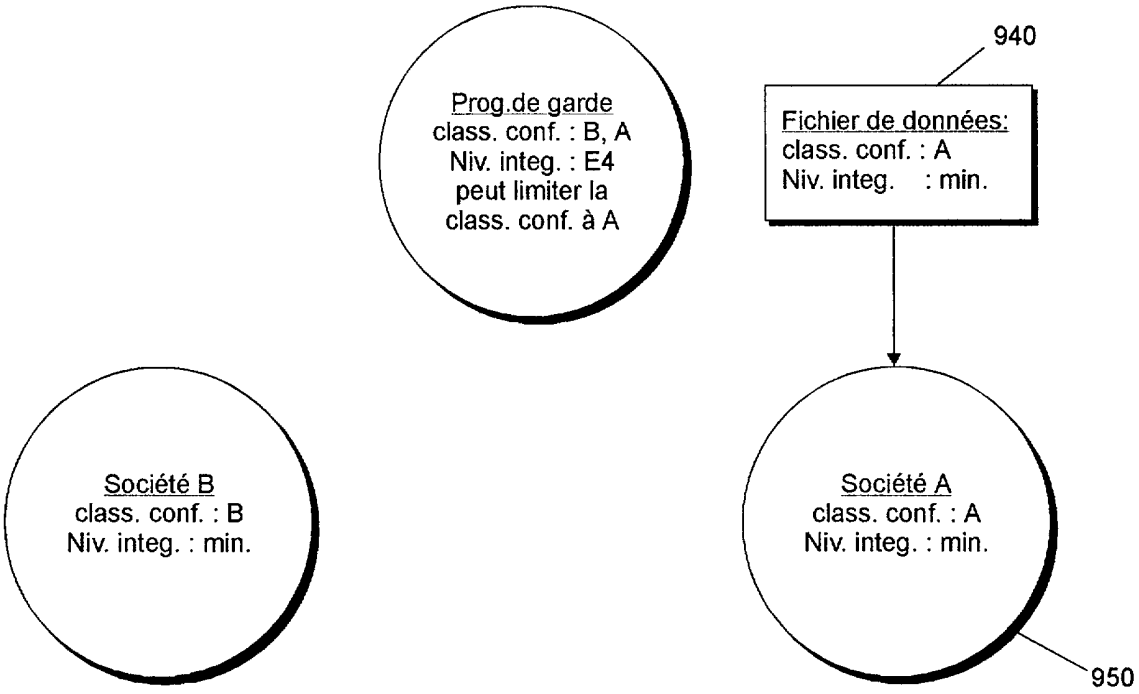

Prog.de garde
class. conf. : B, A
Niv. integ. : E4
peut limiter la
class. conf. à A
940
Fichier de données:
class. conf. : A
Niv. integ. : min.
Société B
class. conf. : B
Niv. integ. : min.
Société A
class. conf. : A
Niv. integ. : min.
950

FIG. 9