Politiques et modèles d'autorisation discrétionnaires (DAC) et Obligatoire (MAC)

Politiques et modèles d'autorisation discrétionnaires (DAC)

Présentation des DAC

- Les politiques de contrôle d'accès discrétionnaires s'appuient sur les notions de propriété (tout sujet est propriétaire d'un ensemble d'objets) et de droit d'accès (lecture, écriture, etc).
- Le contrôle d'accès est dit discrétionnaire lorsque la technique de restriction d'accès aux objets est basée sur l'identité des sujets et/ou des groupes auxquelles ils appartiennent.
- Le contrôle est discrétionnaire dans le sens où un sujet possédant un certain droit d'accès est capable de conférer ce droit à tout autre utilisateur.

Présentation des DAC

- La gestion des droits d'accès est généralement donnée par la règle suivante : Un sujet s peut donner un droit d'accès sur un objet o à un autre sujet s‘ si et seulement si s est le propriétaire de l'objet o.
- De même, les politiques de contrôle d'accès discrétionnaire intègrent généralement le droit de propriété, permettant ainsi à un sujet de céder la propriété d'un objet à un autre sujet.
- Le problème du contrôle d'accès discrétionnaire est qu'il est nécessaire d'avoir confiance en les utilisateurs du système.
- De plus, cette approche ne permet pas de garantir que l'accès à un objet est interdit à un sujet : il suffit que le sujet obtienne, d'une manière ou d'une autre, le droit d'accès à l'objet pour être autorisé à y accéder.
- Les politiques discrétionnaires ne peuvent rien non plus contre l'utilisation de canaux cachés de stockage utilisés notamment par les chevaux de Troie.

Modèle de Lampson

- La notion de matrice de contrôle d'accès, dédiée à la représentation des droits d'accès (autorisations), a été introduite par Lampson dès 1971.
- La structure de ce modèle est celle d'une machine à états où chaque état est un triplet (S,O,M) avec : S désignant un ensemble de sujets, O un ensemble d'objets et M une matrice de contrôle d'accès.
- Chaque cellule M(s,o) de cette matrice contient les droits d'accès que le sujet s possède sur l'objet o.

Modèle de Lampson

- Les droits correspondent généralement à des actions comme lire ou écrire.
- La matrice des droits d'accès n'est pas figée. Si de nouveaux objets, de nouveaux sujets ou de nouvelles actions sont ajoutées dans le système, il est nécessaire d'enregistrer toutes les permissions accordées pour ces nouvelles entités.
- La mise à jour d'une politique de sécurité exprimée par ce modèle est quelque peu fastidieuse.
- Ce modèle ne permet pas d'exprimer directement des interdictions ou des obligations.
- Le modèle de Lampson a été progressivement amélioré pour donner naissance à d'autres modèles tels que HRU et Take-Grant.

Modèle HRU

- Le modèle HRU s'est intéressé à la complexité de la tâche de vérification des propriétés assurées par une politique d'autorisation. Comme dans le modèle de Lampson, HRU utilise une matrice d'accès classique, la différence réside en ce que HRU précise les commandes qui peuvent lui être appliquées. Les seules opérations possibles sont données dans le tableau 1.
- Le format des commandes est présenté dans le tableau 2, où x_i est un paramètre de la commande, $a^{(i)} \in A$ est un droit, et op_i est une des six opérations élémentaires possibles.

Enter a ***into*** $M(s, o)$	***delete*** a ***from*** $M(s, o)$
Create subject s	***destroy subject*** s
Create object o	***destroy object*** o

Format d'une commande HRU.

Modèle HRU

Command $\alpha(x_1, x_2, .., x_k)$
If $a' \in M(s', o')$ *and* $a'' \in M(s'', o'')$ *and* $\ldots$ *and* $a^{(m)} \in M(s^{(m)}, o^{(m)})$
Then $op_1 ; op_2 ; .. ; op_n$
End

Opérations élémentaires de HRU.

- Etant donné un système, une configuration initiale Q_0, et un droit α, on dit que Q_0 est sûr pour α, s'il n'existe aucune séquence d'opérations qui, exécutée à partir de l'état Q_0, peut amener le droit α dans une cellule particulière (i.e. pas dans n'importe laquelle) de la matrice de contrôle d'accès dans laquelle α ne se trouve pas déjà. La démonstration de cette propriété constitue le problème de protection (safety problem).
- En fait, ce problème revient à vérifier qu'un schéma d'autorisation est correct vis-à-vis d'un ensemble d'objectifs de sécurité.
- Harrison, Ruzzo et Ullman ont démontré deux théorèmes fondamentaux concernant la complexité du problème de protection :

Modèle HRU

- le problème de protection est indécidable dans le cas général, c-à-d, étant donné une matrice d'accès initiale et un ensemble de commandes, il est impossible de savoir si aucune séquence d'applications de ces commandes n'aura pour conséquence de mettre un droit particulier dans un endroit de la matrice où il ne se trouvait pas initialement ;
- le problème de protection est décidable pour les systèmes à mono-opération, c-à-d dont les commandes ne contiennent qu'une seule opération élémentaire.

- Le modèle HRU à mono-opération est très pratique à manipuler mais, il reste trop simple pour couvrir des politiques de sécurité intéressantes.
- Dans la mesure où il n'y a pas d'opération élémentaire qui permette simultanément de créer un objet et d'y associer des droits, il ne permet pas d'exprimer des politiques dans lesquelles les sujets qui créent des objets se voient attribuer des droits spécifiques sur ces objets.

Modèle Take-Grant

- Diverses évolutions issues du modèle HRU ont tenté de déterminer un modèle suffisamment expressif pour représenter des politiques d'autorisation sophistiquées, mais pour lequel le problème de protection reste décidable .
- Le modèle Take-Grant est constitué d'un graphe dont les nœuds sont des sujets ou des objets, et des règles de modification de ce graphe.
- Il peut être vu comme une variante d'HRU à ceci près qu'il restreint les différentes commandes en les répartissant en quatre catégories (figure 1 ; où P et R sont des sujets, Q est un sujet ou un objet) :

Modèle Take-Grant

- la commande create qui permet de créer un objet et d'attribuer initialement un droit d'accès à un sujet sur cet objet ;

- la commande remove qui permet de retirer un droit d'accès d'un sujet sur un objet ;

- la commande take, représentée par un arc étiqueté t entre un sujet P et un sujet (ou objet) R, indique que P peut prendre tous les droits que R possède ;

- la commande grant qui permet à un sujet P possédant un droit d'accès sur Q ainsi que le droit g sur un autre sujet R, de céder à R le droit sur Q (que P possède sur Q) ;

Modèle Take-Grant

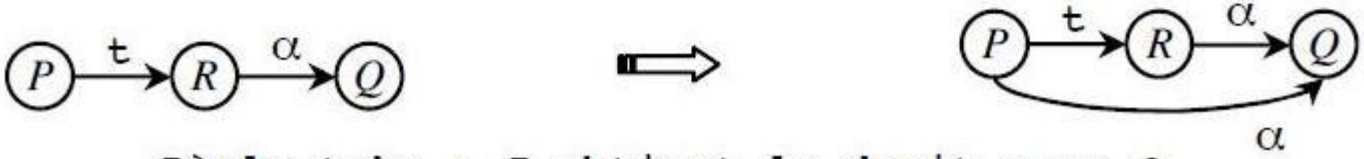
- Dans ce modèle, le graphe représentant l'état de protection du système, peut être assimilé à la matrice d'accès, et les quatre règles ci-dessus (dites de réécriture), correspondent au schéma d'autorisation, c'est-à-dire aux commandes.
- Même si ces règles peuvent paraître simples, leurs combinaisons peuvent mener le système dans des états d'insécurité.
- En effet, l'application successive de plusieurs règles (bien choisies) peut donner d'autres droits à des sujets, ce qui risque de compromettre certains objectifs de sécurité.
- L'exemple de la figure 2, montre un graphe de protection qui contient deux sujets, P et R et un objet O.
- Dans l'état de protection initial, R possède le droit α sur O et le droit t sur P.
- Considérons un objectif de sécurité stipulant que le système est déclaré non-sûr si P parvient à acquérir le droit sur O.

Modèle Take-Grant

Règle create : *P* crée l'objet *Q*

Règle remove : retire le droit γ de *P* sur *Q*

Règle grant : *P* cède le droit α sur *Q* à *R*



Règle take : *P* obtient le droit α sur *Q*

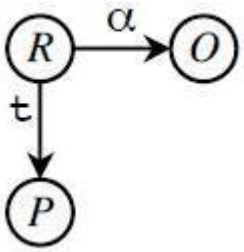
Modèle Take-Grant

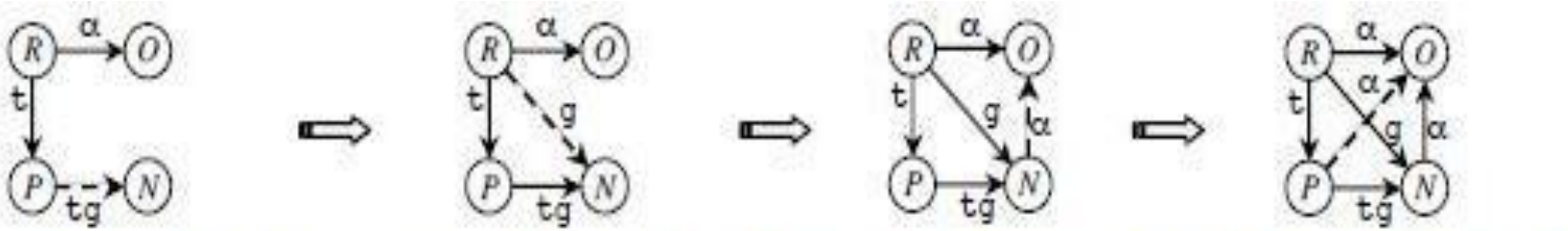
- La séquence d'application des règles décrites dans la figure 3 indique que le système n'est pas sûr, alors que ce constat n'est pas directement explicite dans la figure 2.
- Pour pallier ce type de problème, Jones et al. ont étudié le problème de protection dans le cadre du modèle Take-Grant.
- Ils définissent le prédicat can de la façon suivante : « P can α Q » est vrai si et seulement s'il existe une séquence de graphes « G_1, .., G_n » telle que P ait le droit α sur Q dans le graphe G_n.
- Jones et al. définissent les conditions nécessaires et suffisantes pour que le prédicat soit satisfait.
- Ils établissent également l'existence d'une solution algorithmique de complexité linéaire permettant d'établir si le prédicat est vérifié.

Modèle Take-Grant

- Toutefois, les hypothèses sous-jacentes à ce modèle sont assez peu réalistes.
- En effet, et comme on peut le constater avec l'exemple présenté, s'il est vrai que P peut parvenir à acquérir le droit α sur O, il ne peut le faire que si R collabore avec lui.
- En réalité, il est difficile d'imaginer que tous les sujets vont collaborer afin de mettre la sécurité en péril.
- Une telle hypothèse est donc de pire cas sur le comportement des utilisateurs du système. Plusieurs raffinements des propriétés démontrables grâce au modèle Take-Grant ont été proposés, notamment afin de lever cette hypothèse et de se concentrer sur les cas où un utilisateur ou un ensemble d'utilisateurs tentent de mettre en défaut les objectifs de sécurité.

Modèle Take-Grant





1/ *P* crée un arc étiqueté *tg* vers un nouveau sujet *N*

2/ *R* reprend le droit (*g* sur *N*) de *P*

3/ *R* accorde le droit (α vers *O*) à *N*

4/ *P* prend le droit (α vers *O*) de *N*

L'étude des propriétés de ce modèle dans des cadres spécifiques a fait l'objet de nombreux travaux.

Modèle TAM

•S'inspirant du modèle HRU, Sandhu a présenté un modèle appelé TAM (Typed Access Matrix).

•Dans TAM, chaque objet appartient à un certain type qui ne peut changer. Les commandes utilisent cette notion de type.

•Un modèle de sécurité utilisant TAM est composé d'un ensemble fini A de droits, d'un ensemble de types d'objets T et d'un ensemble fini de types de sujets t_s ($t_s \subseteq T$).

•Ces éléments sont utilisés pour définir l'état de protection à l'aide d'une matrice de contrôle d'accès typée.

•Le schéma d'autorisation est constitué de A, T, et d'une collection finie de commandes TAM. Les opérations primitives de TAM sont données dans le tableau 3 ($t_s \in T_s$; $t \in T$).

•Les commandes TAM sont présentées dans le tableau 4, où, $x_i : t_i$ exprime que le paramètre x_i est de type t_i.

Modèle TAM

•Sandhu montre qu'il est possible de résoudre le problème de protection dans bon nombre de cas pratiques, sans perdre de puissance d'expression.

•Il décrit un algorithme qui permet d'obtenir un état maximal de protection.

•Cet état se caractérise par une matrice d'accès sur laquelle on ne peut plus exécuter de règles du schéma d'autorisation.

•Une version dite augmentée de TAM, appelée ATAM, a été proposée afin de fournir un moyen simple de détecter l'absence de droits dans une matrice d'accès. Pour cela, le modèle ATAM offre la possibilité d'utiliser des tests du type « $a_i \notin M(s,o)$ » dans la partie conditionnelle de la commande.

•La façon de gérer ce type de commande et de résoudre le problème de protection a également été définie.

•L'intérêt de cette démarche consiste à modéliser facilement la séparation des pouvoirs.

Modèle TAM

Enter a ***into*** $M(s, o)$	***delete*** a ***from*** $M(s, o)$
Create subject s ***of type*** t_s	***destroy subject*** s ***of type*** t_s
Create object o ***of type*** t	***destroy object*** o ***of type*** t

Opérations élémentaire de TAM.

Command $\alpha(x_1 : t_1, x_2 : t_2, .., x_k : t_k)$
If $a' \in M(s', o')$ *and* $a'' \in M(s'', o'')$ *and* ... *and* $a^{(m)} \in M(s^{(m)}, o^{(m)})$
Then op_1 ; op_2 ; .. ; op_n
End

Format d'une commande TAM.

Politiques et modèles d'autorisation obligatoires (*MAC)*

Introduction

- Les politiques obligatoires décrètent des règles incontournables destinées à forcer le respect des exigences de sécurité.
- Ainsi, les politiques multi-niveaux affectent aux objets et aux sujets des niveaux non-modifiables par les usagers, et donc qui limitent leur pouvoir de gérer les accès aux données qu'ils possèdent.
- Les politiques de Bell et LaPadula visant à assurer la confidentialité, et celle de Biba s'intéressant à l'intégrité, en sont les exemples les plus anciens.
- D'autres politiques ont été développées pour les systèmes commerciaux et pour les institutions financières.

Politique du DoD et modèle de Bell-LaPadula

- La politique *multi-niveaux de Bell-LaPadula est une politique obligatoire, développée pour le DoD (Department of Defense) des Etats-Unis [Bell-LaPadula 1976].*
- *Elle a été mise au point au* moment où les efforts se concentraient pour concevoir un système d'exploitation multiutilisateurs.
- Les permissions d'accès sont définies à travers une matrice de contrôle d'accès et un ensemble de niveaux de sécurité.
- Cette politique considère seulement les flux d'informations qui se produisent quand un sujet observe ou modifie un objet. Destinée au domaine militaire, la propriété à assurer est la confidentialité.

Politique du DoD et modèle de Bell-LaPadula

- Le modèle associé à la politique de Bell-LaPadula est fondé sur la notion de treillis. Il s'appuie sur l'association de différents niveaux aux sujets (niveaux d'habilitation) et aux objets (niveaux de classification).
- Chaque niveau *n = (cl, C) est caractérisé par ses deux attributs :*

  - *C : un compartiment défini par un ensemble de catégories, par exemple {nucléaire,* défense}.

  - *cl : une classification prise dans un ensemble totalement ordonné, par exemple {non classifié,* confidentiel, secret}.

Politique du DoD et modèle de Bell-LaPadula

- Pour un objet *o, la classification c(o) est un moyen de représenter le danger que peut* constituer la divulgation de l'information contenue dans cet objet ; pour un sujet *s, c'est une* habilitation *h(s) qui désigne la confiance qui lui est accordée.*
- *Les niveaux constituent un treillis partiellement ordonné par une relation de dominance notée "≤" et définie par :*
- si $n = (cl, C)$ et $n' = (cl', C')$; $n \leq n'$ (n' domine n) si et seulement si $cl \leq cl'$ et $C \subseteq C'$

Politique du DoD et modèle de Bell-LaPadula

- Les objectifs de sécurité de cette politique sont les suivants :

- interdire toute fuite d'information d'un objet possédant une certaine classification vers un objet possédant un niveau de classification inférieur ;
- interdire à tout sujet possédant une certaine habilitation d'obtenir des informations d'un objet d'un niveau de classification supérieur à cette habilitation.

- Le schéma d'autorisation associé à ces objectifs de sécurité en découle directement.

Politique du DoD et modèle de Bell-LaPadula

- On considère que l'on peut distinguer vis-à-vis de la confidentialité, parmi les différentes opérations qu'un sujet peut effectuer sur un objet, les opérations de lecture et d'écriture, et on introduit les règles suivantes, incontournables même par les propriétaires des informations:
- *Propriété simple : un sujet s_i ne peut lire un objet o_j que si son habilitation $h(s_i)$ domine la* classification *$c(o_j)$ de l'objet : $(s_i, o_j, lire) \Rightarrow h(s_i) \geq c(o_j)$*
- *Propriété étoile : un sujet ne peut lire un objet o_j et en écrire un autre o_k que si la* classification d'*o_k domine celle d'o_j : $(s_i, o_j, lire) \wedge (s_i, o_k, écrire) \Rightarrow \quad c(o_k) \geq c(oj)$*

Politique du DoD et modèle de Bell-LaPadula

- La propriété simple interdit de lire des informations d'une classification supérieure à l'habilitation, et la propriété étoile empêche les flux d'information d'une classification donnée vers une classification inférieure, ce qui constituerait une fuite d'information.
- Le modèle permet de mettre en évidence la cohérence de la politique, c-à-d que le schéma d'autorisation ne peut conduire à un état où la propriété « *une information classifiée ne doit pas être transmise à un utilisateur non habilité à la connaître » ne soit pas respectée.*

Politique du DoD et modèle de Bell-LaPadula

- Pour plus d'expressivité, d'autres variantes de ce modèle ont été introduites, notamment en traduisant les notions d'*observation et de modification de l'information, non* seulement par les opérations élémentaires : *lire, et écrire, mais aussi par exécuter, ajouter :*

- *exécuter : accès sans modification ni observation ;*

- *lire : observer sans modifier ;*

- *ajouter (append) : modification sans observation, par exemple dans le cas des écritures* dans les fichiers d'audits (*audit logs)*;

- *écrire : observation et modification.*

Politique du DoD et modèle de Bell-LaPadula

Le modèle peut être représenté par une machine à états où chaque état est défini par un triplet (*S,O,M), où :*

- *S : un ensemble de sujets ;*

- *O : un ensemble d'objets ;*

- *A : un ensemble d'opérations d'accès ; A = {exécuter, lire, ajouter, écrire} ;*

- un ensemble de niveaux de sécurité muni d'une relation d'ordre partiel ;

- *B : l'ensemble de tous les états possibles ;*

- *M : l'ensemble des matrices* $(M_{so})s \in S$; $o \in O$,

- $F \subset L_s \times L_c \times L_o$ *: l'ensemble des niveaux de sécurité ; dans un état donné,* f_s *est le plus haut* niveau de sécurité qu'un sujet peut avoir ; f_c *est le niveau courant de chaque sujet ;* f_o *est la* classification de l'objet.

Politique du DoD et modèle de Bell-LaPadula

Les deux règles de contrôle d'accès sont :

- *La propriété simple : un état est sûr si et seulement si, pour chaque sujet s qui observe un* objet *o, le niveau de sécurité maximal du sujet domine le niveau de sécurité de l'objet.*

- *La propriété étoile : un état est sûr si et seulement si, pour chaque sujet s qui modifie un* objet *o, le niveau de sécurité de o domine le niveau courant de sécurité de s.*

Politique du DoD et modèle de Bell-LaPadula

D’une manière plus précise :

- Si l’opération est un ajout, elle ne sera possible que si le niveau de sécurité de l’objet domine le niveau courant de sécurité du sujet ;
- si l’opération consiste à créer un objet et d’écrire dedans, le niveau de sécurité de l’objet est égal au niveau courant de sécurité du sujet ;
- si l’opération est une lecture, le niveau de sécurité de l’objet est dominé par le niveau courant du sujet.

Politique du DoD et modèle de Bell-LaPadula

- La politique de Bell-Lapabula présente plusieurs inconvénients.
- Le plus important est la dégradation du service provoquée par la *surclassification des informations.*
- *En effet, au cours* de sa vie, le niveau d'une information ne peut que croître : si une information non classifiée est utilisée par un sujet habilité au secret, tout objet modifié par ce sujet avec cette information sera classifié secret.
- Petit à petit, les niveaux de classification des informations croissent de façon automatique, et il faut les "déclassifier", manuellement par un officier de sécurité ou par un processus dit "de confiance" (*trusted process) n'obéissant pas aux règles du modèle.*

Politique du DoD et modèle de Bell-LaPadula

- De plus, il est possible de construire un système, appelé Système *Z [McLean 1985], qui* vérifie bien les deux propriétés, et qui n'est pourtant pas sûr.
- Le système *Z est un système où* un utilisateur de niveau minimal met les niveaux de tous les sujets et de tous les objets au niveau minimal, et autorise l'accès de tous les utilisateurs à tous les objets.
- Ceci est possible car le niveau d'un objet peut, lui-même, être mémorisé dans un objet ; la valeur de ce dernier peut donc être modifiée par un utilisateur de niveau minimal (puisque l'écriture dans un niveau dominant est autorisée).

Politique d'integrité de Biba

- La politique de *Biba [Biba 1977] applique à l'intégrité un modèle* analogue à celui de *Bell-LaPadula pour la confidentialité.*
- *À chaque sujet s on affecte un* niveau d'intégrité *is(s), correspondant à la confiance qu'on a dans ce sujet et chaque objet o* possède un niveau d'intégrité *io(o), correspondant au niveau d'intégrité du sujet qui l'a créé.*

Politique d'integrité de Biba

- Les objectifs de sécurité de cette politique visent à :

  - interdire toute propagation d'information d'un objet situé à un certain niveau d'intégrité vers un objet situé à un niveau d'intégrité supérieur ;

  - et interdire à tout sujet situé à un certain niveau d'intégrité de modifier un objet possédant un niveau d'intégrité supérieur.

- Le schéma d'autorisation découle de la recherche de ces propriétés, en considérant des labels d'intégrité et le fait que les opérations peuvent être regroupées en trois classes : observation, modification et invocation.

Politique d'integrité de Biba

- Un sujet ne peut modifier un objet que si le label d'intégrité du sujet domine le label d'intégrité de l'objet : $(s_i, o_j, \textit{modifier}) \Rightarrow io(o_j) \leq is(s_i)$.

- Un sujet ne peut observer un objet que si le label d'intégrité de l'objet domine le label d'intégrité du sujet : $(s_i, o_j, \textit{observer}) \Rightarrow \quad is(si) \leq io(oj)$.

- Un sujet s_i *ne peut invoquer un sujet* s_j *que si le label d'intégrité de* s_i *domine le label* d'intégrité de s_j *:* $(s_i, s_j, \textit{invoquer}) \Rightarrow is(s_j) \leq is(s_i)$.

Politique d'integrité de Biba

- Ces règles garantissent qu'une information ne pourra pas "polluer" celle d'un niveau d'intégrité supérieur.
- Le modèle de Biba présente un inconvénient analogue à celui de Bell-LaPadula : la dégradation des niveaux d'intégrité.
- Si une information d'un niveau d'intégrité donné est utilisée par un sujet d'un niveau inférieur, tous les objets modifiés ou créés par ce sujet à partir de cette information seront d'un niveau d'intégrité inférieur.
- Il faut alors remonter artificiellement le niveau d'intégrité de certains objets par des sujets "de confiance", autorisés à violer la politique de sécurité.
- Cette politique a été étendue par Totel pour permettre la collaboration de logiciels de différents niveaux de criticité dans un même.

Politique *de Clark et Wilson*

- Dans l'environnement commercial, la prévention contre les modifications non autorisées de l'information (essentiellement contre les fraudes et les erreurs) est un atout primordial.
- Et même si un utilisateur est autorisé à manipuler certaines données, cela ne doit pas pouvoir entraîner une perte ou une falsification des actifs (capitaux, informations).
- Afin de répondre à ce type d'objectifs, Clark et Wilson proposent un autre type de politique obligatoire pour l'intégrité.

Politique *de Clark et Wilson*

- Celle-ci vise à assurer les besoins suivants :

  - Une *cohérence (uniformité interne) des données faisant partie de l'état interne du système.* Ce type de cohérence peut être imposé par le système de calcul.

  - Une *cohérence entre l'état interne du système informatique et le monde réel qu'il* représente.

- La politique de Clark et Wilson repose sur deux anciens principes bien connus : les *transactions bien formées et la séparation des pouvoirs.*

Politique *de Clark et Wilson*

- Le concept de *transactions bien formées stipule que les utilisateurs n'ont pas le droit de* manipuler les données d'une manière arbitraire, mais seulement au travers des procédures de transformation spécifiques, préservant l'intégrité des données.
- Une procédure de transformation peut être par exemple un programme qui applique le principe de l'équilibre de la balance en comptabilité (toute modification dans les grands livres de la comptabilité doit figurer dans les deux parties de la balance).

Politique *de Clark et Wilson*

- Le concept de *séparation des pouvoirs (separation of duty) est l'un des mécanismes* classiques de contrôle de fraudes et des erreurs.
- Il est fondé sur la répartition des opérations entre plusieurs parties, et l'attribution des droits différents, mais complémentaires, à différentes catégories de personnes.
- Dans certaines entreprises par exemple, l'achat d'une marchandise nécessite l'intervention du service commercial qui fait la commande, du service de contrôle qui vérifie l'acquisition de la marchandise et du service financier qui paye les fournisseurs.

Politique *de Clark et Wilson*

- Notons que toutes ces opérations sont accompagnées, à la fois de traces papiers (bon de commande, bon de livraison, facture), et de traces informatiques.
- Une exécution convenable de l'opération globale implique une cohérence entre les représentations interne et externe des données, c'est-à-dire la correspondance entre les procédures informatiques et celles du monde réel.
- Les procédures (transactions bien formées) manipulant les données doivent être examinées et bien établies ; la capacité à les installer et les modifier doit être contrôlée ; la validité des données doit toujours faire l'objet d'une vérification ; l'affectation des droits aux utilisateurs ainsi que la séparation des pouvoirs doivent êtres menées à bien, etc.

Politique *de Clark et Wilson*

- Pour cela, la politique de Clark et Wilson commence par séparer les données manipulées en deux groupes : les données contraintes (notées CDI pour *Constrained Data Items) et les* données non contraintes (notées UDI pour *Unconstrained Data Items).*
- *Le premier groupe* désigne les données soumises à des règles de manipulation strictes visant à conserver leur intégrité.
- En effet, les différentes opérations de transformation doivent permettre de garantir que l'intégrité des données contraintes persiste.

Politique *de Clark et Wilson*

- Le deuxième groupe concerne les données dont l'intégrité n'est pas garantie et qui peuvent être manipulées arbitrairement.
- Ces données sont importantes car elles représentent la manière selon laquelle une nouvelle information est introduite dans le système, comme les entrées tapées par l'utilisateur sur le clavier.
- Afin de représenter les règles appliquées dans leur politique de sécurité, Clark et Wilson définissent leur modèle de la manière suivante :

Politique *de Clark et Wilson*

- l'ensemble des utilisateurs, U = {*u1, u2, …} ;*

- l'ensemble des données contraintes, CDI = {*CDI1, CDI2, …} ;*

- l'ensemble des données non contraintes, UDI = {*UDI1, UDI2, …} ;*

- un ensemble d'opérations de vérification de l'intégrité des données,

IVP = {*IVP1, IVP2, …}. Ainsi, les CDI sont certifiées par des IVP ;*

- l'ensemble des opérations de transformation des données (*Transformation Procedures, en* anglais), TP = {*TP1, TP 2, ...} ;*

les CDI ne peuvent être manipulées que par des TP ;

Politique *de Clark et Wilson*

- une relation R*c liant chaque opération de transformation TPi, à un sous ensemble c de* CDI ; R*c précise les données que l'opération peut manipuler ;*

- Une relation R*u liant chaque utilisateur u et chaque opération de transformation TPi, à un* sous-ensemble de CDI ; R*u précise les données contraintes qu'un utilisateur est autorisé à* manipuler (par le biais de *TPi).*

Politique *de Clark et Wilson*

- L'objectif de cette politique de sécurité est de garantir l'intégrité des données contraintes et de satisfaire le principe de séparation de pouvoirs. Pour mettre en œuvre la politique d'intégrité, les règles obligatoires du modèle de *Clark et Wilson imposent que :*

  - les CDI soient dans un état valide, vérifié par les IVP ;

  - toutes les opérations de TP doivent être certifiées ; si une donnée contrainte est valide avant l'exécution d'une opération *TPi, elle reste toujours valide après l'exécution de TPi ;* de plus, les opérations de TP ne peuvent être effectuées que sur des données contraintes spécifiées par R*c ;*

Politique *de Clark et Wilson*

- chaque utilisateur n'effectue des opérations de transformation sur des données contraintes que si celles-ci lui sont associées par R*u ;*

- la relation R*u doit être certifiée afin de refléter les besoins de séparation de pouvoirs ;*

- le système doit authentifier l'identité de chaque utilisateur souhaitant exécuter une opération de TP ;

- avant d'être exécutées, les opérations de TP doivent enregistrer leurs identifiants dans les journaux d'audit ; ceux-ci sont considérés comme données contraintes ;

- chaque opération TP_i *qui accepte une donnée* UDI_j *en entrée doit garantir que, quelle que* soit la valeur de UDI_j, *soit* TP_i *n'effectue que des transformations conduisant à une donnée* contrainte valide CDI_k, *soit* UDI_j *est rejetée.*

Politique *de Clark et Wilson*

- En dépit de sa présentation relativement informelle, le modèle de Clark et Wilson met en avant clairement un certain nombre de préoccupations qui peuvent apparaître dans une organisation commerciale.
- Tout d'abord, ce modèle s'intéresse à l'assurance de la propriété d'intégrité par le biais de l'utilisation de procédures de transformation certifiées.
- Il sensibilise ainsi à la *certification, activité importante lors de la conception d'un système.*
- *En outre, ce* modèle met en évidence deux préoccupations importantes dans bon nombre de structures : la *traçabilité, c'est-à-dire la possibilité de reconstituer les actions importantes du point de vue des* objectifs de sécurité, et la *séparation des pouvoirs.*

Politique *de Clark et Wilson*

- L'existence d'un historique de l'exécution du système permet d'identifier les comportements erronés qui ont pu être à l'origine d'une violation des objectifs de sécurité (par exemple, une faute dans l'implémentation d'une opération, non détectée par la certification).
- Ce souci de fournir, outre des mécanismes garantissant la sécurité du système, des mécanismes capables de fonctionner en l'absence de propriétés de sécurité attendues pour le système, participe à la volonté de définir des politiques de sécurité applicables dans un environnement moins rigide que ceux dans lesquels des politiques de sécurité comme les politiques multi-niveaux sont utilisées.