Introduction aux critères communs

Nicolas Stouls (Nicolas.Stouls@imag.fr)
Étudiant en thèse à l'INP Grenoble dans l'équipe VASCO du LSR-IMAG
681, rue de la Passerelle, BP 53 38041 Grenoble Cedex 9,
sous la tutelle de : Mme Marie-Laure Potet et M. Sylvain Boulmé

Version 1.0

1 Historique des versions

Version 1.0 (2004) : Adaptation du document *Présentation des critères communs et application au projet EDEN*

2 Introduction

Certains logiciels peuvent être soumis à différentes contraintes de sécurité. Cela peut être pour plusieurs raisons comme, par exemple, s'ils sont sensés gérer des données confidentielles. Comment, dans ces cas là, peut-on avoir une garantie que le logiciel a été développé avec sérieux ? Comment se convaincre qu'il n'existe pas de failles de sécurité dues à l'implémentation ou même à l'installation ?

Pour permettre aux producteurs de logiciels de fournir des garanties sur les parties sensibles de leurs produits, certains gouvernements ont mis en place dans leur pays des critères d'évaluation. Ces critères permettent d'attribuer une note à chaque **S**ystème d'**I**nformation évalué. En fonction de cette note, on connaît les soins apportés à son développement, et on peut en déduire un degré de confiance.

Ces critères diffèrent selon les pays. En Europe, par exemple, ce sont les **ITSEC** (**I**nformation **T**echnology **S**ecurity **E**valuation **C**riteria) qui définissent ces critères depuis 1993, alors que ceux du département de la défense américaine sont les TCSEC (**T**rusted **C**omputer **S**ecurity **E**valuation **C**riteria). Ces différents critères ne sont pas toujours comparables. Ainsi, un produit certifié en Europe peut ne pas être reconnu aux états Unis.

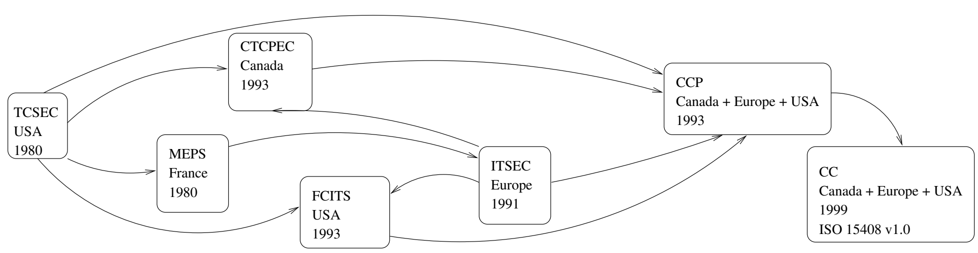


Figure 1: Historique des Critères Communs - déduit de [Bou97, Tro99, DCS, Crib]

C'est pour cela qu'ont été mis en place les **CC** (**C**ritères **C**ommuns de certification) [Com99a, Com99b, Com99c]. Ils sont issus d'une coopération entre trois nations : le Canada, les états Unis et l'Europe. La première version de la norme ISO 15408, définissant les **C**ritères **C**ommuns, est apparue en 1999 comme le montre la figure 1 page 1. Ces critères sont maintenant reconnus de manière internationale et permettent aux producteurs de logiciels de faire reconnaître la qualité de leurs produits dans le monde. En effet, dans la version actuelle des **C**ritères **C**ommuns (Version 2.1), on considère qu'il existe 15 nations reconnaissant ces certificats (Allemagne, Australie, Canada, Espagne, États-Unis, Finlande, Grèce, Hollande, Israël, Italie, Nouvelle-Zélande, Royaume Uni et Suisse) dont 6 seulement étant habilités à en délivrer (Allemagne, Canada, États-Unis, France, Pays-Bas et Royaume Uni)

Les premiers critères d'évaluation étaient principalement basés sur la vérification de la correction du logiciel fourni, ainsi que du matériel utilisé. On ne tenait pas compte du lien entre les deux. Les nouveaux critères permettent maintenant d'évaluer des **S**ystème d'**I**nformation.

On appelle **SI** (**S**ystème d'**I**nformation) *"l'utilisation pratique d'une information au travers de son traitement par un ordinateur de quelque nature qu'il soit"* [Bou97], comme dans l'exemple ci-après.

Prenons l'exemple d'une carte à puce de paiement. Sur cette carte se trouve un logiciel d'identification de l'utilisateur (par code secret par exemple) et l'utilisation de celle-ci nécessite un terminal (terminal bancaire de retrait automatique d'argent par exemple). Cet ensemble (logiciel + carte + terminal) est un **S**ystème d'**I**nformation, et pour l'évaluer il ne faudra pas juste vérifier que le logiciel est correctement pensé et programmé, la carte bien sécurisée et le terminal sans failles, mais il faudra aussi vérifier que l'utilisation de ce logiciel sur cette carte avec ce terminal ne produit pas de nouvelles failles de sécurité.

Ainsi, l'évaluation d'un logiciel de mail destiné à tourner sur un PC sous Windows© chez un particulier ne sera pas la même, à niveau d'assurance équivalent, que l'évaluation du même logiciel de mail destiné à être utilisé sur un serveur sous UNIX derrière un pare-feu dans un bureau de la défense nationale.

Lors de son évaluation selon les **CC**, le **SI** n'est pas vérifié dans son intégralité. Seules les parties dites "sensibles" sont évaluées. C'est à dire que si, par exemple, on évaluait un logiciel de gestion des E-mails qui fournirait la possibilité de chiffrer ou déchiffrer des messages, alors seul le module de chiffrement/déchiffrement du logiciel, ainsi que les modules qui communiquent avec lui seraient évalués et notés (toujours en fonction de leur utilisation). On appelle **TOE** (Target Of Evaluation) la partie évaluée du **SI**.

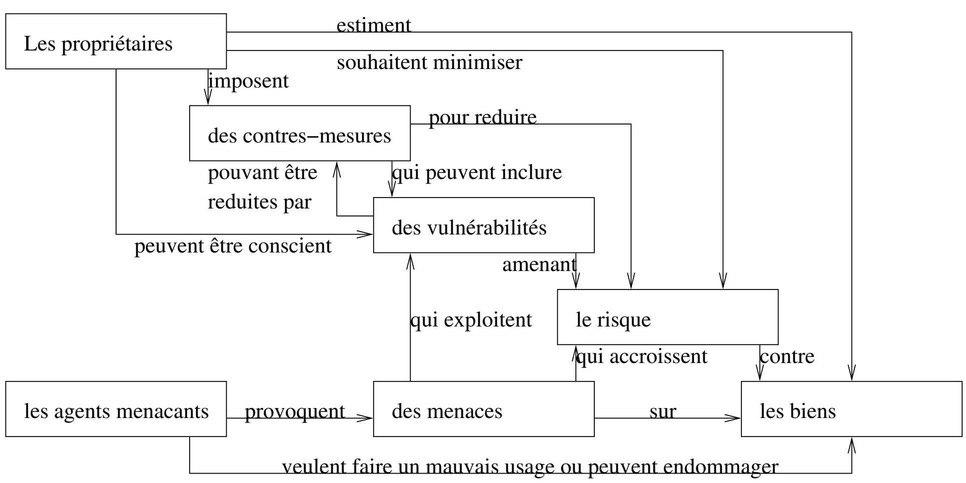


Fig. 2 – *études de sécurité d'un* **SI** *avant les Critères Communs [Com99a]*

En informatique, les études de sécurité n'ont pas toujours, voire même rarement, été faites dès la conception mais, le plus souvent, rajoutées à l'existant. Le schéma de développement d'un **SI** ressemble alors au schéma de la figure 2.

Parmi les objectifs des **C**ritères **C**ommuns, on peut trouver la volonté de rationaliser cette étude préalable pour, si possible, se rapprocher d'un schéma de développement ressemblant un peu à cela :

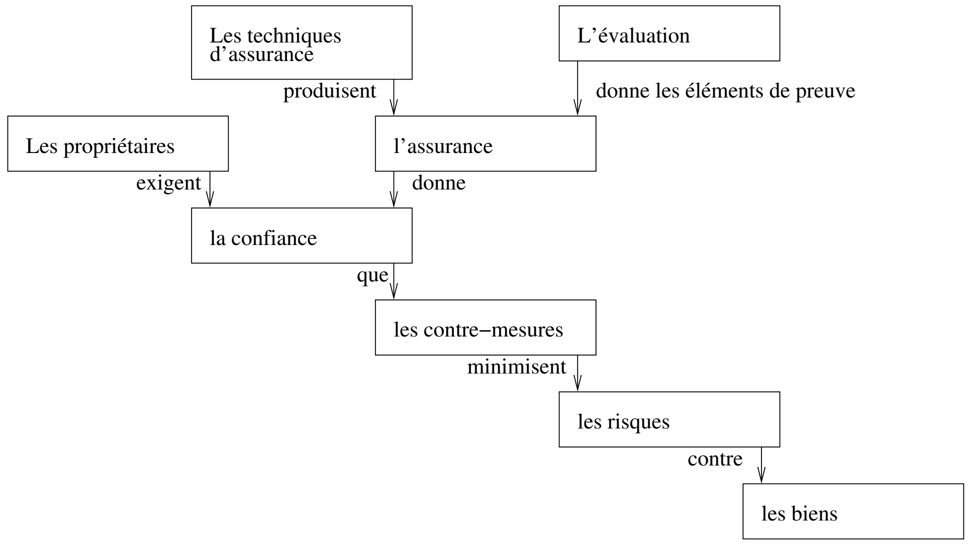


Fig. 3 – *études de sécurité d'un* **SI** *avec les Critères Communs [Com99a]*

Les études de sécurité sans les **C**ritères **C**ommuns étaient faites de manière empirique et instinctive. C'est à dire que l'on commençait par penser tout le système et ensuite on le confrontait aux différents types d'attaques connues. On y rajoutait alors des défenses et l'on vérifiait qu'aucune faille n'ait été rajoutée. Cette étude était donc circulaire et pouvait durer assez longtemps avant que le client ne s'estime satisfait du niveau de sécurité. De plus, cela alourdissait beaucoup le logiciel et chaque itération de cette répétition devenait de plus en plus complexe.

Ensuite, avec l'arrivée des **C**ritères **C**ommuns, la méthode de conception a changé. En effet, avant de concevoir toute la spécification du logiciel, on peut maintenant se référer aux listes de composants de sécurité fonctionnelle des **CC** pour avoir la quasi totalité des cas de faille à prendre en compte, et pour avoir les spécifications déjà faites et sécurisées des différents modules de sécurité.

On peut considérer que les documents des **C**ritères **C**ommuns sont perçus différemment par les différentes populations d'utilisateurs (les développeurs, les évaluateurs et les utilisateurs). Leurs différents points de vue peuvent être résumés comme suit :

– Pour les développeurs de **SI**, les **C**ritères **C**ommuns sont avant tout une liste d'exigences fonctionnelles qui permet de choisir le niveau de sécurité du produit réalisé.
– Pour les évaluateurs, les **C**ritères **C**ommuns sont principalement une liste d'exigences d'assurance qui indique le niveau de confiance que l'on peut avoir en l'implémentation qui a été faite des exigences fonctionnelles, ainsi que le niveau de risque d'erreurs induit par les manuels.
– Enfin, pour les utilisateurs, les **C**ritères **C**ommuns sont avant tout une fiche technique d'un produit écrite en termes connus et non ambiguës et un rapport d'évaluation permettant de connaître le niveau de fiabilité du dit produit.

Une plus grande compréhension des principaux mécanismes de fonctionnement des **C**ritères **C**ommuns peut être tirée de la lecture des documents [Com99a, Com99b, Com99c, Tro99, Bou97, Elu01, DCS, Crib, Syn02, CL02, Mer00] décrits très rapidement dans la section 5.1 page 14.

3 Principe des Critères Communs

Trois documents décrivent les **C**ritères **C**ommuns :

– Le premier document [Com99a] présente les motivations générales et décrit le mode d'utilisation conseillé des deux autres documents.
– Le deuxième [Com99b] est une liste exhaustive des *exigences fonctionnelles* reconnues par les **C**ritères

Communs. C'est à dire que c'est une liste de spécifications de modules de sécurité, triés en fonction de leur rôle.

- Le troisième [Com99c] est une liste exhaustive des *exigences d'assurance* reconnues par les **CC**. C'est à dire une liste de critères de qualité. La note donnée lors d'une évaluation selon les **C**ritères **C**ommuns n'est que le reflet de la qualité du **SI** en fonction des éléments de cette liste.

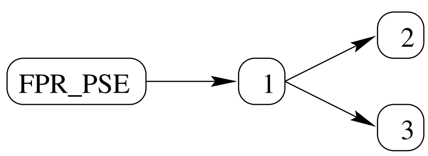
Dans un souci de rapidité de recherche des informations, ces listes sont organisées de manière hiérarchique. Tout d'abord il y a des *classes*, qui sont subdivisées en *familles*. Celles-ci sont des ensembles de *composants "qui ont en commun les mêmes objectifs de sécurité mais qui peuvent différer dans l'accentuation ou dans la rigueur"* [Com99a, §4.4.1.2]. Le composant est la plus petite unité de mesure de ces listes.

La subdivision en classes et familles est effectuée en fonction de la nature des composants, afin de les regrouper par objectif. Cela n'est fait que pour faciliter le travail de recherche dans la documentation.

Un composant peut être le fils ou le frère d'un autre composant de la même famille. Ce lien correspond à de l'héritage. C'est à dire qu'un composant fils contient au moins les spécificités de son composant père.

Prenons l'exemple de trois composants C1, C2 et C3, tels que C2 et C3 soient les fils du composant C1 (comme montré sur la figure 4). Alors C2 regroupe toutes les fonctionnalités de C1 plus quelques unes supplémentaires et, de même, C3 contient toutes les fonctionnalités de C1 plus d'autres qui lui sont propres.

Leur représentation hiérarchique peut être schématisée comme suit :



On appellera *paquet* le regroupement d'un ensemble de composants d'une même liste. Un paquet peut regrouper des composants venant de familles différentes ou de la même famille. Cependant, ce dernier cas n'est possible, que si les différents composants n'héritent pas les uns des autres. Par exemple, sur la figure 4, les composants C1 et C2 ne peuvent pas être mis dans un même paquet, alors que les composants C2 et C3 le peuvent.

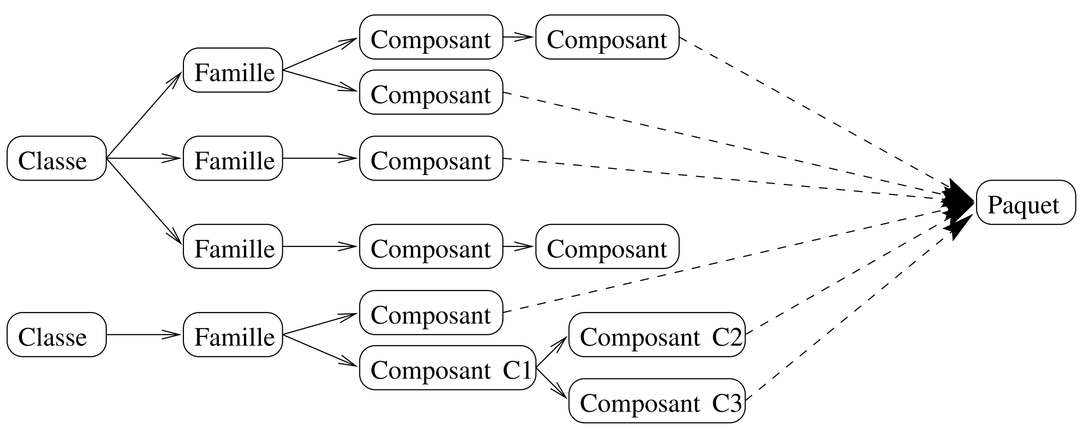
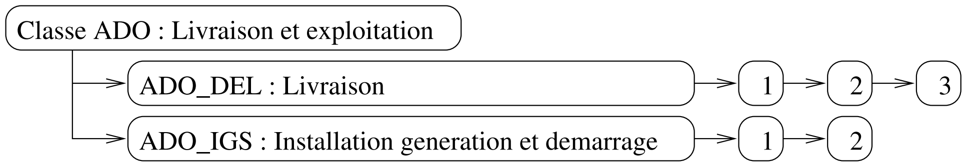


Fig. 4 – *Exemple de hiérarchie*

Un développeur de **SI** choisira les fonctionnalités de son produit dans la liste des exigences fonctionnelles, pour former un paquet, et en fera l'implantation qu'il voudra. Ensuite, il formera un paquet des différentes assurances qu'il a en son **SI** en les tirant de la liste des exigences d'assurance. En fonction de ces choix, il pourra postuler à une note.

La note obtenue au terme de l'évaluation d'un produit ne porte bien évidemment pas que sur l'aspect programmation, mais aussi sur la qualité et les possibilités de maintenance du système, sur la documentation utilisateur, sur la documentation administrateur, sur la clarté des messages, etc.

Prenons l'exemple de la classe ADO (livraison et exploitation) décrite comme suit :



Tout d'abord, il faut remarquer que différents composants portent le même nom (1 et 2). Alors, pour les distinguer, on préfixe leur nom avec le nom de leur famille, qui, lui même, est préfixé par le nom de la classe. Ainsi, on parlera des composants ADO_DEL.1 et ADO_IGS.1 qui sont deux composants distincts.

Cette classe est définie plus précisément en section 3.2 page 6. Avec l'exemple de cette classe, nous pouvons voir qu'il existe certains critères qui ne portent ni sur la programmation du **SI**, ni sur son côté purement matériel. En effet, cette classe est uniquement dédiée à l'évaluation du **SI** dans son contexte d'utilisation.

3.1 Exigences fonctionnelles

La liste des exigences fonctionnelles définissant les **C**ritères **C**ommuns ne contient que des spécifications de fonctions de sécurité. Onze classes de spécifications relatives à la sécurité sont proposées :

- FAU (Audit de sécurité) : "Auditer la sécurité implique la reconnaissance, l'enregistrement, le stockage et l'analyse d'informations associées à des activités touchant à la sécurité" [Com99b, page 21].
- FCO (Communication) : Cette classe ne contient que deux familles, dédiées à la non répudiation des émissions et des réceptions.
- FCS (Support cryptographique) : Ensemble de composants permettant la gestion de crypto-systèmes.
- FDP (Protection des données de l'utilisateur) : "La présente classe contient des familles qui spécifient des exigences pour les fonctions de sécurité de la **TOE** et pour les politiques des fonctions de sécurité portant sur la protection des données de l'utilisateur." [Com99b, page 49]
- FIA (Identification et Authentification) : "Les familles de la présente classe traitent des exigences pour que des fonctions établissent et contrôlent l'identité annoncée d'un utilisateur." [Com99b, page 87]
- FMT (Administration de la sécurité) : "La présente classe est destinée à définir l'administration de plusieurs aspects des fonctions de sécurité de la **TOE** : attributs de sécurité, données et fonctions de sécurité. Les différents rôles d'administration et leurs interactions, comme par exemple la séparation des privilèges, peuvent être spécifiés." [Com99b, page 103]
- FPR (Protection de la vie privée) : "La présente classe contient les exigences relatives à la protection de la vie privée. Ces exigences fournissent à un utilisateur une protection contre la découverte et le mauvais usage de son identité par d'autres utilisateurs." [Com99b, page 119]
- FPT (Protection de l'ensemble des fonctions de sécurité) : "La présente classe contient des familles qui se rapportent à l'intégrité et à l'administration des mécanismes qu'offrent les fonctions de sécurité." [Com99b, page 129]
- FRU (Utilisation des ressources) : "Cette classe inclut trois familles qui concernent la disponibilité des ressources nécessaires telles que la capacité de calcul ou la capacité de stockage." [Com99b, page 165]
- FTA (Accès à la **TOE**) : "La présente classe spécifie des exigences fonctionnelles pour contrôler l'établissement d'une session utilisateur." [Com99b, page 173]
- FTP (Chemins et canaux de confiance) : "Les familles de la présente classe fournissent des exigences pour l'établissement d'un chemin de communication de confiance entre les utilisateurs des fonctions de sécurité, et d'un canal de communication de confiance entre ces dernières et d'autres produits **SI**

de confiance." [Com99b, page 183]

L'intérêt de cette liste de choix est double. Tout d'abord, le programmeur peut se baser sur ce document pour établir plus rapidement les différentes fonctionnalités de son programme, comme l'on choisirait l'agencement de sa cuisine dans un catalogue. Ensuite, le deuxième intérêt est que ce paquet des fonctionnalités réunies dans le programme peut être utilisé en argument marketing auprès des clients. Ceux-ci peuvent alors comparer différents programmes à partir de données précises et de termes techniques qui ne changent pas de signification d'un concepteur à un autre.

La liste des exigences fonctionnelles n'est pas exhaustive vis-à-vis des possibilités de fonctionnalité possiblement intégrables dans un **SI**. Un programmeur peut décider de définir de nouvelles classes, de nouvelles familles ou de nouveaux composants. Il lui faudra alors expliciter chacune des spécifications et les liens qui les relient. Cependant, cette liste est déjà très complète et ce cas n'arrive que rarement.

3.2 Exigences d'assurance

La seconde liste définissant les **C**ritères **C**ommuns est la liste des exigences d'assurance. Dans un **S**ystème d'**I**nformation, l'assurance que "tout ira bien" est importante. Mais, pour arriver à garantir cela, il y a un certain nombre de points à vérifier. De plus, chaque vérification peut-être faite de manière plus ou moins poussée. C'est pourquoi les vérifications les plus courantes, dans leurs différents niveaux de force, ont été réunies dans une grande liste : la liste des exigences d'assurance des **C**ritères **C**ommuns.

Toujours pour plus de facilités, cette liste a été triée par classes, puis par familles. Les différents composants d'une même famille correspondent à un niveau plus ou moins poussé de vérification d'un même point. Dans cette liste, chaque composant ne peut avoir qu'un fils au plus et ne peut pas avoir de frère. Les différentes classes d'exigences d'assurance sont les suivantes :

- ACM (Gestion de configuration) : Cette classe aide à garantir que l'intégrité de la **TOE** est bien préservée, en exigeant une discipline et des contrôles dans le processus de raffinement de la **TOE**.
- ADO (Livraison et exploitation) : La présente classe "définit les exigences pour les mesures, procédures et normes qui parlent de livraison, d'installation et d'utilisation opérationnelle sûr de la **TOE**, garantissant que la protection en terme de sécurité offerte par la **TOE** n'est pas compromise pendant son transfert, son installation, son démarrage et son exploitation." [Com99c, page 18]
- ADV (Développement) : Cette classe définit des exigences pour le raffinement pas-à-pas des fonctions de sécurité depuis les spécifications globales de la **TOE** jusqu'à l'implémentation effective.
- AGD (Guides) : Cette classe "définit des exigences destinées à permettre la compréhension, la couverture et la complétude de la documentation d'exploitation fournie par le développeur." [Com99c, page 20]
- ALC (Support au cycle de vie) : Cette classe offre des exigences pour obtenir une assurance au moyen de l'adoption d'un modèle de cycle de vie bien défini qui couvre toutes les étapes du développement de la **TOE**.
- ATE (Tests) : Cette classe formule des exigences de tests qui démontrent que les fonctions de sécurité satisfont aux exigences fonctionnelles de sécurité de la **TOE**.
- AVA (Estimation des vulnérabilités) : Cette classe "définit des exigences destinées à l'identification des vulnérabilités exploitables. Elle concerne, de façon spécifique, les vulnérabilités introduites pendant la construction, l'exploitation, l'utilisation impropre ou la configuration incorrecte de la **TOE**." [Com99c, page 22]
- APE (Evaluation d'un profil de protection) : "Cette classe permet de montrer que le profil de protection (**PP**) est complet, cohérent et techniquement correct. Un **PP** évalué convient pour servir de base au développement de cibles de sécurité ." [Com99c, page 31]
- ASE (Evaluation d'une cible de sécurité) : "Le but de l'évaluation d'une cible de sécurité est de montrer que cette dernière est complète, cohérente, techniquement correcte et, par conséquent, convient pour servir de base à l'évaluation de la **TOE** correspondante."[Com99c, page 42]
- AMA (Maintenance de l'assurance) : Cette dernière est destinée à maintenir l'assurance que la **TOE** continuera à satisfaire à sa cible de sécurité quand des changements sont effectués sur la **TOE** ou son

environnement.

Comme pour les exigences fonctionnelles, cette liste n'est pas exhaustive. Tout développeur peut, à souhait, définir que son **SI** remplit d'autres exigences d'assurance. Ce point sera plus détaillé dans la section 3.4, page 10.

3.3 Les PP (Profils de Protection)

Lors du développement d'un **SI**, les premiers choix qui sont faits permettent de déterminer la politique de sécurité. Une fois que celle-ci est définie il faut choisir les fonctionnalités de sécurité qui doivent être implantées de telle sorte que la politique de sécurité choisie soit respectée. Cela représente un long travail de recherche et de vérification de cohérence.

Par exemple, si le **S**ystème d'**I**nformation est un serveur d'informations personnelles sur les utilisateurs d'un serveur Web, alors le **SI** sera soumis aux contraintes légales sur les serveurs de ce type. Il faudra alors établir quelles sont les fonctions qu'il faudra mettre en place dans la **TOE** pour respecter le minimum légal et éviter les failles de sécurité.

L'aboutissement d'une telle étude mène à la production d'un **P**rofil de **P**rotection (**PP**). Si ce dernier est développé de manière générique, alors il sera réutilisable. De même, cette étude peut être remplacée par le choix d'un **P**rofil de **P**rotection déjà existant. C'est là l'intérêt des **P**rofil de **P**rotection.

En effet, un **P**rofil de **P**rotection contient un certain nombre d'informations, dont une description de la **TOE** qui doit être implémenté pour lui correspondre. Voici un schéma résumant les différentes parties d'un **P**rofil de **P**rotection :

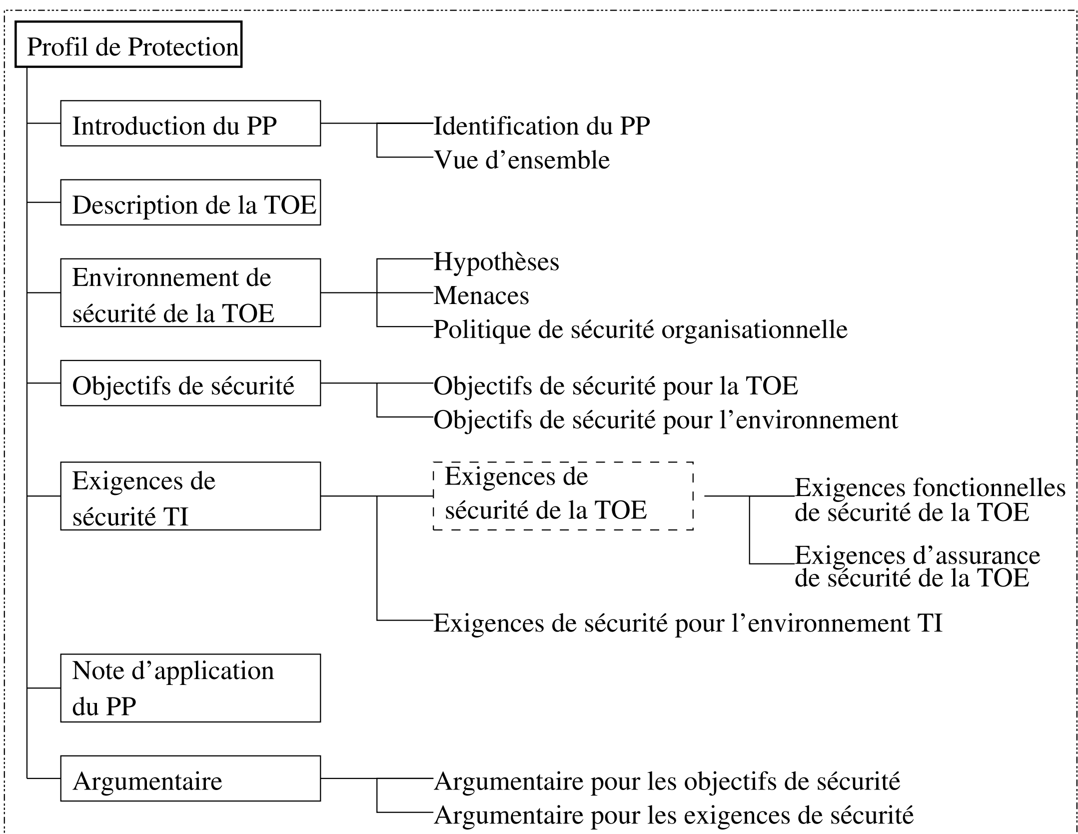


Fig. 5 – *Contenu d'un profil de protection*[1]

Ce qu'il est intéressant de remarquer dans cette généralisation des **PP**s, c'est que les fonctions mises en

[1]La description exacte de chacun des champs d'un **PP** est donnée dans *Common Criteria for Information Technology Security Evaluation Norme ISO 15408 - Part 1 : Introduction and general model - version 2.1* [Com99a, Annexe B, page 46]

œuvre dans la **TOE** sont toutes justifiées par la prise en compte de l'environnement de sécurité de la **TOE**. C'est à dire que c'est la présence d'une menace ou d'une autre politique de sécurité qui justifiera la présence d'une fonction. De plus, chacun des raisonnements aboutissant à l'ajout d'une fonction dans la **TOE** doit être détaillé dans l'argumentaire du **PP**.

Le résultat d'une telle méthode de mise en place d'une **TOE** est une cible de sécurité plus légère (car plus ciblée) et donc plus simple à évaluer. De plus, il est alors possible d'évaluer la cohérence des **P**rofils de **P**rotection afin d'en permettre une large diffusion (commerciale ou non) sur d'autres projets.

On peut noter la présence des documents [Cria, Gra] permettant une meilleure compréhension des **P**rofil de **P**rotection.

3.4 Les EALs (Evaluation Assurance Level)

Une évaluation selon les **C**ritères **C**ommuns aboutit à la remise d'un certificat d'évaluation qui certifie que le **SI** évalué respecte un certain niveau d'assurance. Ce niveau d'assurance est une note allant de EAL1 à EAL7. EAL1 étant le plus bas niveau de garantie et EAL7 le plus fort.

A chacune de ces notes on peut associer une équivalence aux autres grandes normes de certification (ITSEC - Europe, TCSEC -états Unis et CTCPEC - Canada), et une signification informelle grossière, comme montré dans le tableau suivant :

EAL	**Niveau de sécurité**	**TCSEC**	**ITSEC**	**CTCPEC**
-	Niveau minimum de sécurité	D	E0	T0
EAL1	Tests fonctionnels	-	-	T1
EAL2	Tests structurels	C1	E1	T2
EAL3	Tests et vérifications méthodiques	C2	E2	T3
EAL4	Conception, tests et vérifications méthodiques	C3	E3	T4
EAL5	Conception semi-formelle et tests	C4	E4	T5
EAL6	Vérification semi-formelle de la conception générale	C5	E5	T6
EAL7	Vérification formelle de la conception générale	C6	E6	T7

Remarques :
- Chacun des niveaux d'évaluation comprend l'intégralité des niveaux précédents.
- Les niveaux EAL5 à EAL7 nécessitent l'utilisation de méthodes semi-formelles et formelles pour le développement du **SI**, mais requièrent encore les niveaux de tests très élevés des EALs inférieurs.

Chacune de ces notes est associée à un paquet minimum d'exigences d'assurance. En effet, les niveaux d'assurance sont reliés directement, et exclusivement, aux différentes garanties de sécurité. Or, comme au sein de chaque famille, les composants sont triés par ordre croissant de sécurité, on peut alors définir une EAL en fonction de chaque famille de chaque classe. Ainsi, pour une EAL donnée, on peut avoir la liste minimum des composants de chaque famille qui sont nécessaires à son obtention. La figure 2 montre l'intégralité de ces liens.

Dans ce tableau, on voit l'ensemble des classes d'assurance et de leurs familles définies dans les **C**ritères **C**ommuns. Pour chacun des niveaux d'EAL, on représente, dans les colonnes 3 à 9 (étiquetées EAL1 à EAL7), le composant minimum de chaque famille devant être présent pour pouvoir postuler au dit EAL. Ce sont les chiffres ce trouvant dans les colonnes, en face de leur classe et famille.

En effet, chaque ligne correspond à une famille différente. Ainsi, par le croisement d'une ligne et d'une colonne, on identifie le composant minimum de la famille donnée qui doit être présent dans le paquet d'assurances d'un **SI** pour pouvoir postuler à l'EAL de la colonne choisie. Ce composant est matérialisé par son numéro dans la case désignée par l'intersection décrite.

En effet, comme dit plus avant, chaque composant est désigné par un numéro d'identification qui est choisi en fonction de son niveau d'assurance. C'est à dire qu'un composant d'une famille ayant un numéro plus grand qu'un autre composant de la même famille fournira un niveau d'assurance plus important.

De fait, une case vide signifie que, pour l'obtention de l'EAL donnée, la famille associée à cette ligne n'a pas à être représentée dans le paquet des exigences d'assurance.

Classe	**Famille**	**Composants par niveau d'assurance**						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Gestion de configuration	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Livraison et exploitation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Développement	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_INT					**1**	**2**	**3**
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guides	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Support au cycle de vie	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	**3**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	2	2	2	2	2	**3**
Estimation des vulnérabilités	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4
Evaluation d'un profil de protection	APE_DES							
	APE_ENV							
	APE_INT							
	APE_OBJ							
	APE_REC							
	APE_SRE							
Evaluation d'une cible de sécurité	ASE_DES							
	ASE_ENV							
	ASE_INT							
	ASE_OBJ							
	ASE_PPC							
	ASE_REQ							
	ASE_SRE							
	ASE_TSS							

Figure 2: Tableau récapitulatif des exigences des différentes EALs

Prenons l'exemple de la classe ADO (Livraison et exploitation) :

Classe	**Famille**	**Composants par niveau d'assurance**						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Livraison et exploitation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1

Si l'on voulait postuler à l'EAL4, alors il faudrait au moins avoir dans notre paquet de composants d'assurance les composants ADO_DEL.2 et ADO_IGS.1, alors que pour l'EAL1 on n'aurait besoin que du composant ADO_IGS.1.

On représente en gras les changement entre 2 EALs contiguës. Par exemple, le passage de l'EAL3 à l'EAL4 nécessite la présence du composant ADO_DEL.2 au moins et non plus du composant ADO_DEL.1. C'est pourquoi le 2 de la ligne ADO_DEL et de la colonne EAL4 est en gras.

Par la lecture de ce tableau, nous pouvons voir quels sont les différents paquets nécessaires aux différentes notes. Nous pouvons également constater que certaines des exigences d'assurance prévues par les **C**ritères **C**ommuns n'entrent en compte dans aucune des notes (comme par exemple les classes APE - Evaluation d'un profil de protection - et ASE - Evaluation d'une cible de sécurité -, ou la famille ALC_FLR - Support au cycle de vie / Correction d'anomalies).

En fait, il est tout à fait possible de demander à être évalué sur un paquet d'exigences d'assurance qui serait à cheval sur plusieurs EALs. La note finale sera la plus importante dont l'ensemble des composants nécessaires à son obtention sont présents.

Cela dit, si des composants supplémentaires, présents dans la liste, sont présents dans le paquet évalué, alors on dira que le niveau atteint est augmenté.

De même, si des composants supplémentaires, non présents dans la listes, sont présents dans le paquet évalué, alors on dira que le niveau atteint est étendu.

Par exemple, voici le paquet qui, si son évaluation est un succès, correspond à l'EAL1 :
ACM_CAP.1, ADO_IGS.1, ADV_ESP.1, ADV_RCR.1, AGD_ADM.1, AGD_USR.1 et ATE_IND.1.

Si maintenant, on ajoute à ce paquet la famille AVA_VLA.1, alors son évaluation donnera la note EAL1 augmentée.

Enfin, si l'on ajoute au paquet correspondant à l'EAL1, une famille ADV_PERSO définie par nos soins, alors son évaluation donnera la note EAL1 étendue.

Les documents [dlsdt02, AFRT02] montrent deux exemples concrets d'extension des **C**ritères **C**ommuns dans le cas d'applications spécialisées.

3.5 Les différentes étapes menant à la certification

3.5.1 Etape 1 : préparation à l'évaluation

Cette étape, bien que fortement conseillée, est facultative. Elle permet de consolider la description des cibles de l'évaluation.

Lors de cette phase, on détermine un calendrier, avec les dates de livraison des différents composants, et l'on fait une première estimation des coûts ainsi que du niveau de certification visé et de ses chances de succès.

3.5.2 Etape 2 : évaluation

Cette étape se déroule dans les locaux d'un organisme indépendant autorisé (CESTI en France). Elle a pour objectif l'acquisition de la confiance en les critères de sécurité dans les documents fournis.

En France, l'évaluation porte sur les **C**ritères **C**ommuns ainsi que sur les critères européens ITSEC.

Cette évaluation, qui peut éventuellement faire appel aux développeurs du produit, donne en résultat un rapport technique d'évaluation (RTE) remis au commanditaire ainsi qu'à l'organisme de certification (en France, le DCSSI - Direction Centrale de la Sécurité des Systèmes d'Information [DCS]).

3.5.3 Etape 3 : rapport de certification

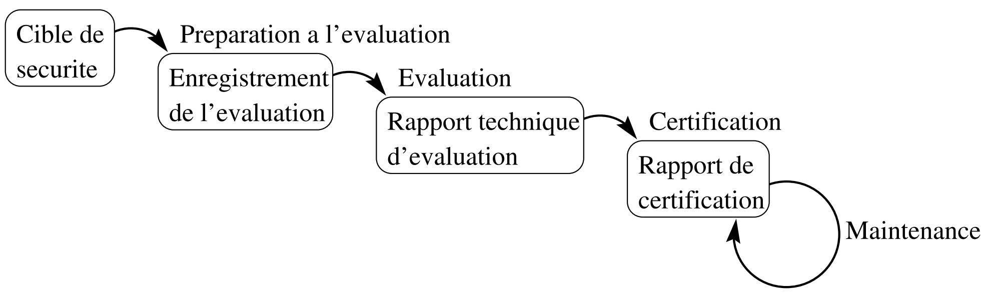
Enfin, Le rapport de certification, qui est basé sur le rapport technique d'évaluation, est remis par l'organisme de certification (DCSSI en France) au commanditaire. Il atteste d'une évaluation conforme aux normes et indique le niveau de certification atteint. Il y est également mentionné l'objet et les fonctions de sécurité soumises à l'évaluation.

De plus, ce rapport peut recommander la mise en oeuvre de mesures particulières pour augmenter le niveau d'assurance.

Des exemples de rapports de certifications sont disponibles en libre accès en [cdlsdsd01a, cdlsdsd01b, dlsdt00].

3.5.4 En résumé

Le processus de certification peut se résumer avec le schéma suivant :



Un développement plus détaillé des différentes étapes menant à la certification peut être trouvé dans [Mer00] d'Alain Merle.

4 Exemple approfondi : la classe d'assurance ADV (Développement)

4.1 Rappel de quelques informations de la classe ADV (Développement)

Cette classe permet de définir le soin apporté aux différentes étapes du développement du produit. Elle donne une assurance sur la qualité de ce dernier. Elle définit des exigences pour le raffinement des fonctions de sécurité depuis les spécifications globales de la **TOE** jusqu'à l'implantation effective.

En effet, suivant le niveau de sécurité recherché, les **C**ritères **C**ommuns exigent un développement des fonctions de sécurité par raffinement, sur un nombre de niveaux qui peut varier entre 1 (spécification et implémentation) et 5.

Voici la liste des différentes familles et autres composants de cette classe :

Classe	Famille	Liste des composants
Développement (ADV)	ADV_FSP : Spécifications fonctionnelles	1,2,3,4
	ADV_HLD : Conception de haut niveau	1,2,3,4,5
	ADV_IMP : Représentation de l'implémentation	1,2,3
	ADV_INT : Parties internes des fonctions de sécurité	1,2,3
	ADV_LLD : Conception de bas niveau	1,2,3
	ADV_RCR : Correspondance des représentations	1,2,3
	ADV_SPM : Modélisation de la politique de sécurité	1,2,3

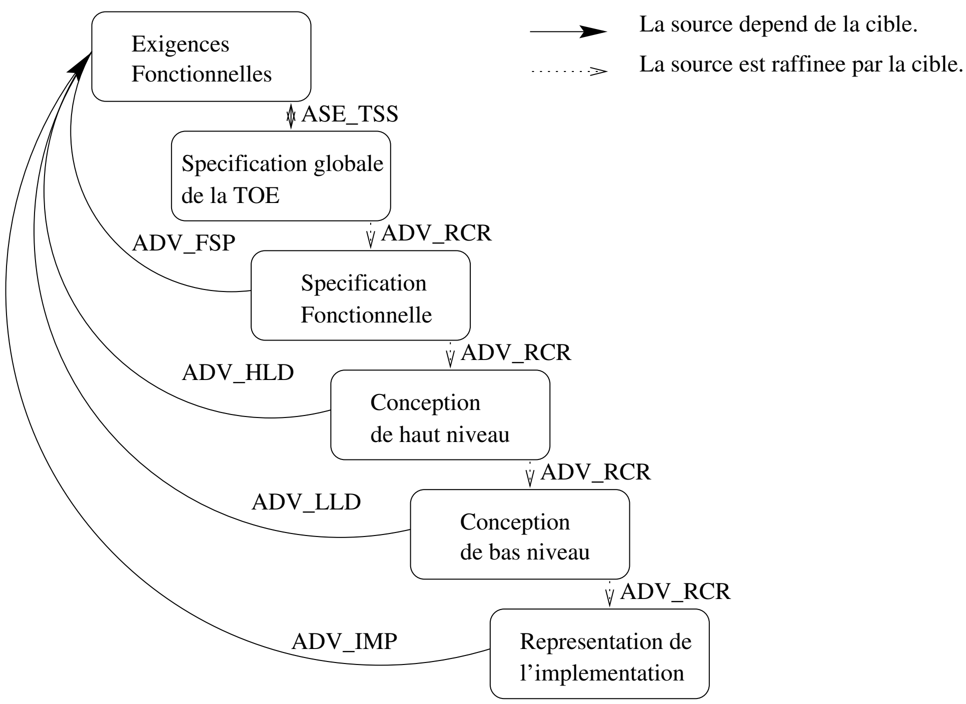
Rappelons aussi la liste des interactions entre la classe ADV et les différents EALs :

Classe	**Famille**	**Composants par niveau d'assurance**						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Développement	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_INT					**1**	**2**	**3**
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3

Nous pouvons déjà constater que le composant ADV_LLD.3 existe dans la liste des composants de la classe ADV, mais n'est pas présent dans les paquets des différents EALs. En effet, même le niveau EAL7 ne requiert pas la présence de ce composant. La différence entre les composants ADV_LLD.2 et ADV_LLD.3 est que dans le premier cas le raffinement de bas niveau doit être prouvé formellement, alors que, dans le second, il doit être prouvé de manière semi-formelle.

4.1.1 Interactions entre les différentes familles

Voici un schéma montrant les interactions entre ces différentes familles, dans le processus de développement d'un **S**ystème d'**I**nformation :



Relations entre les représentations et les exigences de la **TOE**

Ce schéma nous permet de bien illustrer, et donc mieux comprendre, l'utilité de chacune des familles qui nous importent :

- La famille ASE_TSS (Evaluation d'une cible de sécurité / spécifications globales de la **TOE**) est la spécification globale de la **TOE**. C'est à dire qu'elle décrit le paquet des fonctionnalités choisies pour ce produit. En fait, elle met en relation deux à deux les différents composants fonctionnels choisis avec les fonctions fournies.
  Cette famille est optionnelle. Elle n'est requise dans aucun des EALs. Sa présence est donc un plus appréciable et relativement simple à fournir.

- La famille ADV_RCR (Développement / Correspondance entre les représentations) définit la liaison entre les différents niveaux de raffinement des fonctions de sécurité. C'est l'équivalent de l'invariant de liaison en B. Cette famille aura donc pour but de montrer que les raffinements proposés, à travers leurs différents niveaux, sont corrects et complets par rapport aux fonctions de sécurité de la **TOE**.
- La famille ADV_FSP (Développement / spécification fonctionnelle) est le premier niveau de raffinement de la **TOE**. Son évaluation a pour but de définir si les développeurs ont fourni une description adéquate de toutes les fonctions de sécurité de la **TOE**, et si les fonctions de sécurité fournies par la **TOE** sont suffisantes pour satisfaire les exigences fonctionnelles de la cible de sécurité.
  Les spécifications fonctionnelles doivent également fournir l'ensemble du descriptif des interfaces des fonctions de sécurité (ensemble des états et des comportements des différents protocoles ainsi que l'action de ces fonctions, les messages d'erreurs et les exceptions possible).
- La famille ADV_HLD (Développement / Description de haut niveau) doit décrire l'ensemble des fonctions de sécurité en termes de sous-systèmes dont la description fonctionnelle (Action, effets en termes de sécurité) sera donnée. Cette description doit également identifier les exigences des fonctions de sécurité en termes de composants matériels, logiciels système et logiciels.
  L'évaluation de cette famille a pour but de définir si la description de haut niveau fournie est suffisante pour satisfaire les exigences fonctionnelles de la cible de sécurité et de garantir que la description des fonctions et interfaces des grandes structures décrites sont une correcte réalisation de la spécification fonctionnelle.
- La famille ADV_LLD (Développement / Description de bas niveau) doit fournir une description des fonctions de sécurité en terme de modules. Pour chacun d'entre eux, on donnera son but, ses interactions avec les autres modules et ses dépendances.
  L'évaluation de la description de bas niveau a pour but de définir si elle est suffisante pour satisfaire les exigences fonctionnelles de la cible de sécurité et de vérifier si cette description est effectivement un raffinement de la description de haut niveau.
- La famille ADV_IMP (Développement / Représentation de l'implémentation) se présente sous la forme de schémas, de masques ou autres descriptions VHDL[2] pour les parties matérielles, et sous la forme de code source pour la partie logicielle. Cette représentation doit être inambiguë pour permettre une génération automatique des logiciels ou matériels sans autres informations.
  L'évaluation de l'implémentation a pour but de définir si elle est suffisante pour satisfaire les exigences fonctionnelles de la cible de sécurité et de vérifier si cette description est effectivement un raffinement de la description de bas niveau.

Cette description succincte de chacune de ces familles n'est qu'un aperçu de leur rôle global dans le processus d'évaluation selon les **C**ritères **C**ommuns. En effet, suivant les composants choisis dans chacune des ces familles, les exigences lors de l'évaluation ne seront pas les mêmes. Le principal changement réside dans la présentation des descriptions qui peut être demandée sous forme informelle, semi-formelle ou même formelle. Cependant, la plupart des différences entre les composants d'une même famille sont des exigences supplémentaires d'informations dans les descriptions fournies.

[2]**V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage : Langage de description de composants électroniques. De plus amples informations pourront être trouvées sur le site : http ://www.vhdl-online.de/

5 Description des différentes références bibliographiques

5.1 Les références vers des documents d'introduction aux Critères Communs

- [Com99a] : *Common Criteria for Information Technology Security Evaluation - Part 1 : Introduction and general model*
  Ce document est la première partie de la description des **C**ritères **C**ommuns. Cette partie est dédiée à l'explication globale de leur fonctionnement. C'est la première documentation à lire si l'on veut comprendre les **CC**.
  Pour aider à la lecture de ce document, il y a deux choses à savoir :
  1. la liste de toutes les abréviations et des mots clefs avec leur signification sont dans la partie "scope" qui introduit le document.
  2. toutes les informations en gras symbolisent une nouveauté. C'est à dire que dans la description des composants, par exemple, le composant 1 sera écrit tout en gras, puis le composant 2 n'aura en gras que ce qui le différencie du composant 1, etc.
- [Com99b] : *Common Criteria for Information Technology Security Evaluation - Part 2 : Security fonctional requirements*
  Ce document est la deuxième partie de la description des **C**ritères **C**ommuns. Cette partie est dédiée au exigences fonctionnelles. Passée les deux chapitres d'introduction, chaque autre chapitre est dédié à une classe d'exigences fonctionnelle. Cette partie ne sera donc principalement utilisée que par les développeurs, les évaluateurs et les universitaires désespérés à la recherche de données sur les **C**ritères **C**ommuns.
  Pour aider à la lecture de ce document, il y a deux choses à savoir :
  1. la liste de toutes les abréviations et des mots clefs avec leur signification sont dans la partie "scope" qui introduit le document.
  2. toutes les informations en gras symbolisent une nouveauté. C'est à dire que dans la description des composants, par exemple, le composant 1 sera écrit tout en gras, puis le composant 2 n'aura en gras que ce qui le différencie du composant 1, etc.
- [Com99c] : *Common Criteria for Information Technology Security Evaluation - Part 3 : Security assurance requirements*
  Cette troisième et dernière partie des **C**ritères **C**ommuns est dédiée aux exigences d'assurance. Les 7 premiers chapitres sont consacrés à l'introduction des exigences d'assurance et à l'aide à la compréhension des notes, des profils de protections et autres données d'assurances. Ensuite, les chapitres suivants énumèrent, chapitre par chapitre, la liste des classes d'assurance disponibles et les décrivent en profondeur.
  Pour aider à la lecture de ce document, il y a deux choses à savoir :
  1. la liste de toutes les abréviations et des mots clefs avec leur signification sont dans la partie "scope" qui introduit le document.
  2. toutes les informations en gras symbolisent une nouveauté. C'est à dire que dans la description des composants, par exemple, le composant 1 sera écrit tout en gras, puis le composant 2 n'aura en gras que ce qui le différencie du composant 1, etc.
- [Tro99] : *Introduction to the Common Criteria for IT Security (ISO 15408)*
  de Gene Troy
  Ce document est un exposé présentant les **C**ritères **C**ommuns. Il n'apporte que peut de chose en plus par rapport à [Bou97] sur la compréhension globale des **C**ritères **C**ommuns.
- [Bou97] : *Les* ***C****ritères* ***C****ommuns pour l'évaluation de la sécurité des systèmes d'information*
  de Gerard Bouget
  Cette lettre ouverte visant à apporter une première connaissance globale des **C**ritères **C**ommuns est très bien faite, courte (deux feuilles) et explique très bien les points sombres des **CC**.
- [Elu01] : *Analyse de sécurité pour la certification d'applications JavaCard*
  de Marc Eluard

Le chapitre 2 de cette thèse est dédié à la description des **C**ritères **C**ommuns. Assez succincte, elle se veut avant tout énumérative, comme la section 3 du présent document.
- [DCS] : *Site du service gouvernemental français chargé de la sécurité des systèmes d'information*
Ce site est le portail français officiel de toutes les informations relatives aux **C**ritères **C**ommuns. On y trouve toutes les données locales à la France en plus des liens vers les sites internationaux.
- [Crib] : *Site officiel du projet des **C**ritères **C**ommuns*
Ce site est le portail officiel international des **C**ritères **C**ommuns. Il contient donc toutes les dernières informations et tous les liens vers les partenaires des **CC** de par le monde.
- [Syn02] : *Brochure introductive aux **C**ritères **C**ommuns du gouvernement canadien*
Petite brochure explicative des **C**ritères **C**ommuns, ce fascicule publicitaire dédié à la démocratisation des **C**ritères **C**ommuns est simple à lire, mais n'apporte que peu de renseignements.
- [CL02] : *An Interpretation of the Common Criteria EAL7 level*
de C. Loiseaux, E. Giménez, B.Chetali et O. Ly
Ce document propose un début d'interprétation de la classe ADV de l'EAL7. En fait, seul la partie 2 de ce document est énumérative des données, et aucune interprétation n'est mise en avant. Il faut comparer les documents des **C**ritères **C**ommuns [Com99c] pour trouver les subtils ajouts.
Cela dit, les parties 1, 3 et 4 mettent en avant un exemple basé sur la JavaCard et les cartes à puces.
- [Mer00] : *Evaluation des produits suivant les **C**ritères **C**ommuns*
de Alain Merle
Bien que cet exposé ne soit dédié principalement qu' à la découverte des **C**ritères **C**ommuns, la partie 2 de ce dernier (pages 24 à 84) donne une description très détaillée des objectifs, documents à fournir et méthodes d'évaluation pour l'EAL4. Cela peut grandement aider à la compréhension des différentes familles et des différents composants mis en cause.

5.2 Les références vers des exemples d'extensions aux Critères Communs

- [dlsdt02] : *Biometric Evaluation Methodology Supplement*
Ce document décrit le passage depuis les **C**ritères **C**ommuns vers les critères dévaluation des systèmes biométriques. Ce document se base sur les **C**ritères **C**ommuns et redéfinit certaines EAL pour les adapter au domaine particulier de la biométrie.
Pour aider à la lecture de ce document, il y a une chose à savoir :
    1. toutes les informations en gras symbolisent une nouveauté par rapport aux **C**ritères **C**ommuns.
- [AFRT02] : *Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems*
de Jim Alves-Foss, Bob Rinker and Carol Taylor
Ce document est dédié à la norme de sécurité DO-178B. Cette norme est dédiée à l'aéronautique et se base sur les **C**ritères **C**ommuns. Plus exactement sur l'EAL5. Ce document est en fait l'extension de l'EAL5 à la norme DO-178B par ajout de contraintes sur les différents composants.
C'est un excellent exemple montrant l'extension à de nouveaux composants.
Pour aider à la lecture de ce document, il y a une chose à savoir :
    1. les informations écritent en italique sont celles qui viennent tout droit des **CC**. Celles écrites normalement sont les ajouts effectués.

5.3 Les références vers des exemples de certificats d'évaluation

- [cdlsdsd01a] : *Rapport de certification 2001/05 - Carte mixe MONEO/CB*
Direction centrale de la sécurité des systèmes d'information
Rapport de certification de la Carte mixte MoneoCB. Cette carte de paiement à été évaluée à l'EAL4+. Ce rapport atteste des résultats obtenus lors de l'évaluation.
- [cdlsdsd01b] : *Rapport de certification 2001/10 - Carte mixe MONEO/CB*
Direction centrale de la sécurité des systèmes d'information
Rapport de certification de la Carte mixte MoneoCB. Cette carte de paiement à été évaluée à l'EAL4+. Ce rapport atteste des résultats obtenus lors de l'évaluation.

– [dlsdt00] : *Rapport de certification 1999-LGS-01 version 2.0 - Evaluation EAL1 du produit SecureDoc Disk Encryption*
Centre de la sécurité des télécommunications
Rapport de certification du logiciel SecureDoc Disk Encryption pour windows. Cette logiciel à été évaluée à l'EAL1. Ce rapport atteste des résultats obtenus lors de l'évaluation.

5.4 Profils de protection

– [Cria] : *Page des profiles de protection des CC*
Cette page du site des **C**ritères **C**ommuns permet d'avoir accès à la description de chacun des **PP** mis sur le marché à ce jour avec leur description et les coordonnées des ayant droit.
– [Gra] : *TCPA TPMPP Version 0.45*
de David Grawrock
Ce document est un exemple concret de description d'un **P**rofil de **P**rotection en cours de développement qui devrait permettre la mise en place d'un **SI** de niveau EAL3.

5.5 Divers

– [TSCL$^+$02] : *Description du projet* **EDEN**
Document d'appel de proposition du projet RNTL EDEN.
– [Métb] : *TL-FIT White paper*
de D. Le Métayer
Description de l'outils TL-FIT de Trusted Logic, qui permet de générer les documentations pour une évaluation selon les **C**ritères **C**ommuns, jusqu'à l'EAL5, d'un **S**ystème d'**I**nformation qui aurait été développé avec ce logiciel. C'est l'amélioration de cet outil pour qu'il traite les EAL6 et EAL7 qui est l'objectif du projet EDEN.
– [Méta] : *TL-CAT White paper*
de D. Le Métayer
Description de l'outil TL-CAT de Trusted Logic, qui permet de générer automatiquement des jeux de tests. Cet outil se joint à TL-FIT pour aider à faciliter et automatiser les évaluations selon les **C**ritères **C**ommuns.

References

[AFRT02] J. Alves-Foss, B. Rinker, and C. Taylor. Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems. Center for secure and dependable systems, University of Idaho, 2002.

[Bou97] G. Bouget. Les critères communs pour l'évaluation de la sécurité des systèmes d'information. Lettre numéro 23, Institut européen de cindynique, février 1997.

[cdlsdsd01a] Direction centrale de la sécurité des systèmes d'information. *Rapport de certification 2001/05 - Carte mixe MONEO/CB*, avril 2001. http ://www.moneo.net/atouts_techniques/2001_05.pdf.

[cdlsdsd01b] Direction centrale de la sécurité des systèmes d'information. *Carte mixe MONEO/CB, rapport de certification 2001/10*, avril 2001. http ://www.moneo.net/atouts_techniques/2001_10.pdf.

[CL02] B.Chetali et O. Ly C. Loiseaux, E. Giménez. An Interpretation of the Common Criteria EAL7 level. In *ICCC 3*, avril 2002.

[Com99a] Common Criteria. *Common Criteria for Information Technology Security Evaluation, Norme ISO 15408 - Part 1 : Introduction and general model - version 2.1*, Aout 1999. CCIMB-99-031.

[Com99b] Common Criteria. *Common Criteria for Information Technology Security Evaluation, Norme ISO 15408 - Part 2 : Security fonctional requirements - version 2.1*, Aout 1999. CCIMB-99-032.

[Com99c] Common Criteria. *Common Criteria for Information Technology Security Evaluation, Norme ISO 15408 - Part 3 : Security assurance requirements - version 2.1*, Aout 1999. CCIMB-99-033.

[Cria] Common Criteria. Page des profiles de protection des cc. http ://www.commoncriteria.org/index_protection_profile.htm.

[Crib] Common Criteria. Site officiel du projet des **CC**. http ://www.commoncriteria.org/.

[DCS] DCSSI. Site du service gouvernemental francais chargé de la sécurité des systèmes d'information. http ://www.ssi.gouv.fr/fr/.

[dlsdt00] Centre de la sécurité des télécommunications. *Rapport de certification 1999-LGS-01 version 2.0 - Evaluation EAL1 du produit SecureDoc Disk Encryption*, avril 2000.

[dlsdt02] Centre de la sécurité des télécommunications. *Biometric Evaluation Methodology Supplement - version 1.0*, avril 2002.

[Elu01] M. Eluard. *Analyse de sécurité pour la certification d'applications JavaCard.* PhD thesis, Lande - Irisa - Rennes 1, décembre 2001.

[Gra] David Grawrock. Tcpa tpmpp version 0.45. http ://www.commoncriteria.org/ccc/protection_profiles/ppdetail.jsp ?id=PP-002.

[Mer00] A. Merle. *Evaluation des produits suivant les critères communs*, 2000.

[Méta] D. Le Métayer. *TL-CAT White paper*. Trusted Logic.

[Métb] D. Le Métayer. *TL-FIT White paper*. Trusted Logic.

[Syn02] Syntegra. Brochure introductive aux critères communs du gouvernement canadien. http ://www.cse-cst.gc.ca/fr/documents/services/ccs/brochure.pdf, 2002.

[Tro99] G. Troy. *Introduction to the Common Criteria for IT Security (ISO 15408).* US NIST, mars 1999.

[TSCL+02] TrustedLogic, SchlumbergerSema, CEA-LIST, CEA-LETI, and LSR-IMAG. *Description du projet EDEN*, février 2002.

Contents