GENIE LOGICIEL
et
SECURITE

S. Natkin

12/01

PLAN DE L ’EXPOSE

- Introduction
- Politique de sécurité
- Terminologie:
  Propriétés de sécurité, Menaces et attaques
- Introduction à la cryptographie et aux protocoles de sécurité
- Les critères communs
- Méthodes formelles et sécurité

1- Introduction

DOIT ON AVOIR PEUR DES ORDINATEURS?
LE COMPLEXE DE FRANKENSTEIN

Importance croissante du rôle de la diffusion de l ’information via des systèmes techniques de plus en plus complexes, dans des domaines de plus en plus variés.

Par exemple:

- Le contrôle de processus critiques (transport, énergie…)
- Les relations commerciales (commerce et les électronique et monétique)
- La gestion de données confidentielles (secrets militaires ou industriels, données personnelles…)

La peur d ’une technologie nouvelle, incomplètement maîtrisée, l’immaturité sociale dans la pratique de ces nouveaux outils induit une peur => Complexe de Frankenstein

DOIT ON AVOIR PEUR DES ORDINATEURS? LES RISQUES LIES A LA COMPLEXITE

L ’informatique induit la possibilité de construire rapidement et à faible coût des systèmes de traitement de l ’information d ’une incroyable complexité:

Peu de limites aux demandes des utilisateurs qui ne maîtrisent pas la viabilité et la fragilité intrinsèque de cette expression de besoins.

Les limites des possibilités de validation de ces systèmes par rapport à ce qu’ils doivent faire et surtout ce qu ’ils ne doivent pas faire.

L ’incapacité d ’évaluer correctement les conséquences des éventuelles défaillances et donc, a fortiori, des agressions.

L ’incapacité de vérifier à posteriori ce que fait un système automatisé de traitement de l ’information.

SECURITE DES SYSTEMES INFORMATIQUES

Couvre en français deux domaines:

Les méthodes et moyens mis en oeuvre pour éviter les défaillances "naturelles" dont les effets ont un caractère catastrophique (safety)

Les méthodes et moyens mis en oeuvre pour se protéger contre les défaillances résultant d'une action intentionnelle (security)

Nous traitons ici du second domaine

LE CADRE JURIDIQUE (1)

Validité juridique d'opérations informatiques

Certaines transactions informatiques entraînent des obligations légales de responsabilité => Elles sont considérées comme valides juridiquement par la loi ou la jurisprudence.

Exemples

Ordres de virement informatique (par exemple deux fois le même ordre de paiement doit-être honoré) ou ordre de commande dans le cas d ’un contrat de droit privé

Factures électroniques et comptabilité reconnues par l ’administration fiscale

Principe et conditions d ’utilisation de la signature électronique comme élément de preuve (position commune arrêtée par le conseil de l ’union européenne le 28 juin 1999)

CADRE JURIDIQUE (2)

Loi informatique et liberté

La Loi 78_17 du 6/1/1978 Définit la constitution et le rôle de la CNIL (Commission Nationale Informatique et Liberté)

Une entreprise ou une administration qui traite des fichiers administratifs nominatifs est responsable relativement à la non divulgation des informations qu'elle gère.

- Nécessité de formalités préalables à la mise en oeuvre des traitements automatisés pour la collecte, l'enregistrement et la conservation des informations nominatives
- Exercice du droit d'accès
- Dispositions pénales de non respect

CADRE JURIDIQUE (3)

Loi no 85-660 du 3/7/1985

Décrit les règles relatives aux contrefaçons et au droit d'auteur

Par exemple la copie (autre que pour sauvegarde) est punissable de trois mois à deux ans de prison , d'une amende de 6000 à 12000 Francs.

Loi no 88-19 du 5/1/1988

Loi relative à la fraude informatique

Sont passibles de sanctions pénales pouvant atteindre 5 ans de prison, une amende de 2 millions les faits suivants:

. accès frauduleux aux données.

. l'introduction de données

. l'entrave au fonctionnement du système.

CADRE JURIDIQUE (4)

Loi relatives à l'usage de la cryptographie (loi du 19/03/99)

En France l 'usage de moyens de chiffrement est limité:

Utilisation libre concernant l'authentification et l'intégrité et des moyens de chiffrement à clefs de moins de 128 bits (ceux ayant des clefs de plus de 40 bits doivent être déclarés)

Déclaration de commercialisation et d'importation pour les produits de chiffrement ayant des clefs comprises entre 40 et 128 bits

Demande d'autorisation de distribution et d 'utilisation pour les produits de chiffrement ayant des clefs de longueur supérieure à 128 bits

Demande d 'autorisation pour l 'exportation de produit de chiffrement

Auprès du Service Central de Sécurité des systèmes informatiques (SCSSI)

L ’ORIGINE DES RISQUES EST SOUVENT HUMAINE

- Défaillances des systèmes techniques (usure des équipements, panne catalectique, catastrophes naturelles)
- Erreur de conception
- Erreur de réalisation
- Erreur d ’exploitation
  - La négligence , l ’inattention…
  - Les fautes réelles (violation d ’une procédure formalisée)
- Malveillance à caractère ludique
- Fraude, Vol
- Sabotage

QUELQUES STATISTIQUES : COUT DES SINISTRES EN MFF EN 1996 (Clusif)

Accidents		
Physiques (incendie, explosion, dégât des eaux…)	1630	12,81
Pannes	1110	8,73
Force majeure	35	0,28
Perte services essentiels (Télécoms, électricité…)	280	2,20
Total	**3055**	**24,02**
Erreurs humaines		
Utilisation	800	6,29
Conception, Réalisation	1020	8,02
Total	**1820**	**14,31**
Malveillances		
Vol, vandalisme physique	240	1,89
Fraude non physique	2300	18,08
Sabotage	5	0,04
Attaque logique	1090	8,57
Divulgation	1100	8,65
Autres (copies de logiciels)	3110	24,45
Total	**7845**	**61,67**
TOTAL	**12720**	**100**

2-Politique de sécurité

NOTION DE POLITIQUE DE SÉCURITÉ D ’UN SYSTÈME D ’INFORMATION

Assurer la sécurité ne peut être défini et mis en œuvre
que relativement à des objectifs clairement définis:

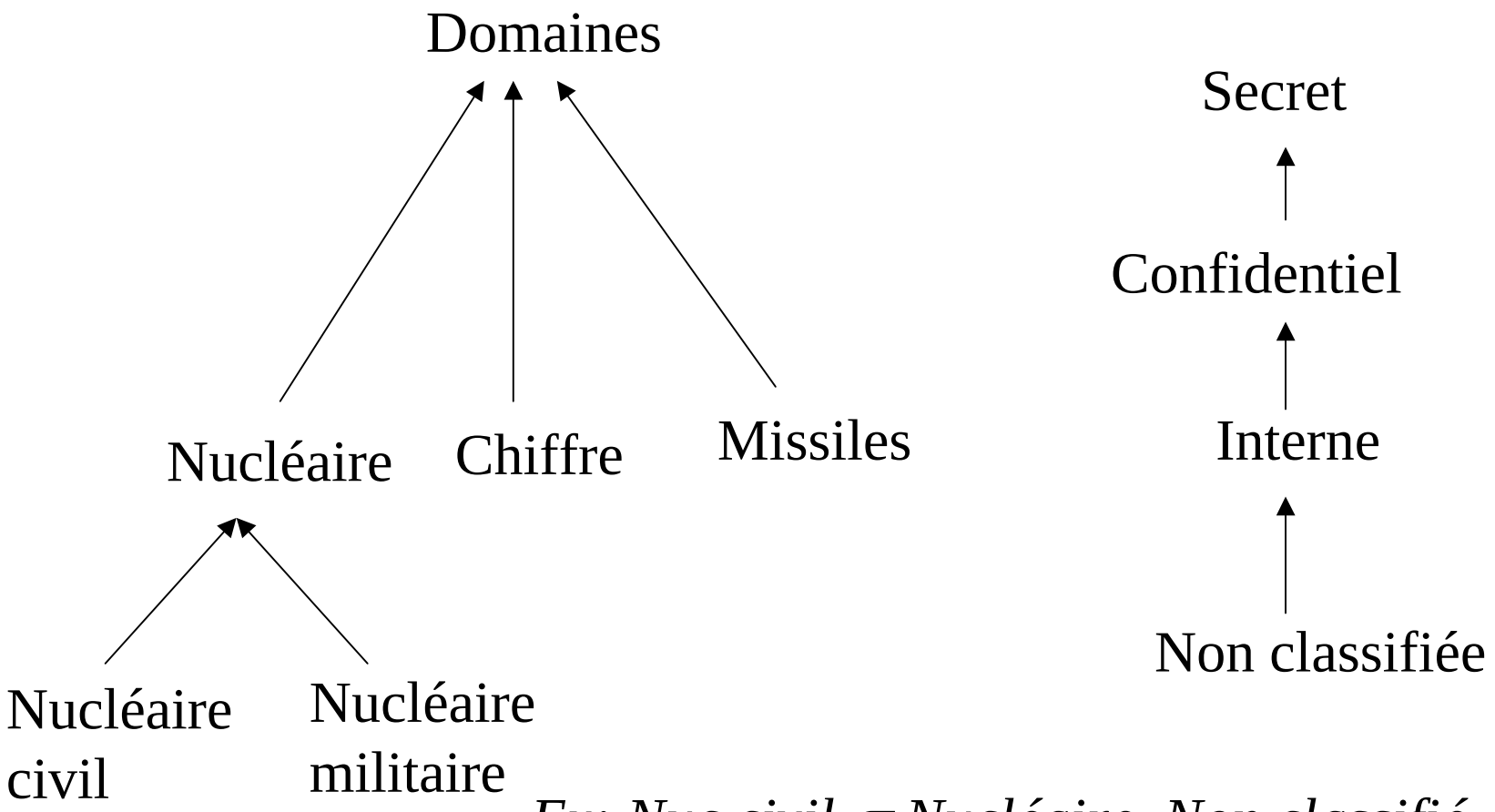
1) Un périmètre d’application
(qui est concerné ou et quand, avec quels moyens…)
qui détermine le système d'information sur lequel porte la politique.
2) Des règles définissant les actions autorisées (**les droits**)
ou interdites réalisées par des hommes sur des hommes ou
des biens matériels ou immatériels.
3) La nature et la force des attaquants éventuels
4) La nature des défaillances auquel doit être
capable de résister une politique

POLITIQUES DE ROLES ET NOMINATIVES DISCRETIONNAIRES ET OBLIGATOIRES

Une politique telle que tous les droits d'une politique sont attribués aux personnes uniquement en fonction du rôle qu'elles jouent dans le système d'information (administrateur système, responsable de sécurité, chef comptable…) est appelée **politique de rôle**. Une telle politique doit préciser les procédures appliquées pour attribuer un rôle à une personne.
Une politique telle qu'au moins un droit est attribué à une personne intutae personnae est dite **politique nominative**.

Une politique de sécurité est **discrétionnaire** si l'entité qui possède un objet à tous les droits pour propager les droits sur cet objet.
Si ce processus de propagation est limité par des règles générales, alors la politique est dite **obligatoire**

EXEMPLE: PROTECTION DE L ’ACCÈS
AU DOCUMENTS (1): HIÉRARCHIES



Ex: Nuc.civil ⊂ Nucléaire, Non classifié < Interne

EXEMPLE: PROTECTION DE L ’ACCÈS AU DOCUMENTS (2): RÈGLES

Toute personne est habilitée à certains niveaux dans certains domaines:
Général X:((secret, nucléaire), (confidentiel, chiffre))

Tout document est classé par un couple:
Doc A (confidentiel, nucléaire civil)
Doc B (interne,missile)

Pour avoir accès à un document D (a,b) il faut avoir une habilitation (x, y) avec $a \subseteq x$ et $b \leq y$
Le Général X peut lire A car nucléaire civil $\subseteq$ nucléaire et secret<confidentiel
Il ne peut lire B car il n ’a aucune habilitation dans un domaine inclus dans les missiles

EXEMPLE: PROTECTION DE L ’ACCÈS AU DOCUMENTS (3): NIVEAU D ’ATTAQUE

Le niveau d ’attaque considéré est maximal:
Agresseurs spécialistes en espionnage militaire, disposant de moyens matériels et financiers illimités

La politique doit rester opérationnelle quelle que soit la nature des défaillances et erreurs pouvant affecter les systèmes physiques considérés.

Identification de tous les acteurs (humain, physique)
pouvant agir sur le système.

Le personnel d'un hôpital classés par unités de soin (US),
les médecins en relation avec l'hôpital (M),
l'administrateur du système (A),
les patients qui ont ou sont soignés à l'hôpital (P)
le reste de l'humanité.

Identification de toutes les ressources sur lequel une action peut porter: les pièces des dossiers médicaux

- Des dossiers D,
- Une table d'accréditation des médecins TM
- Une table des patients TP
- Une table patient/médecin TPM
- Des courriers électroniques ME

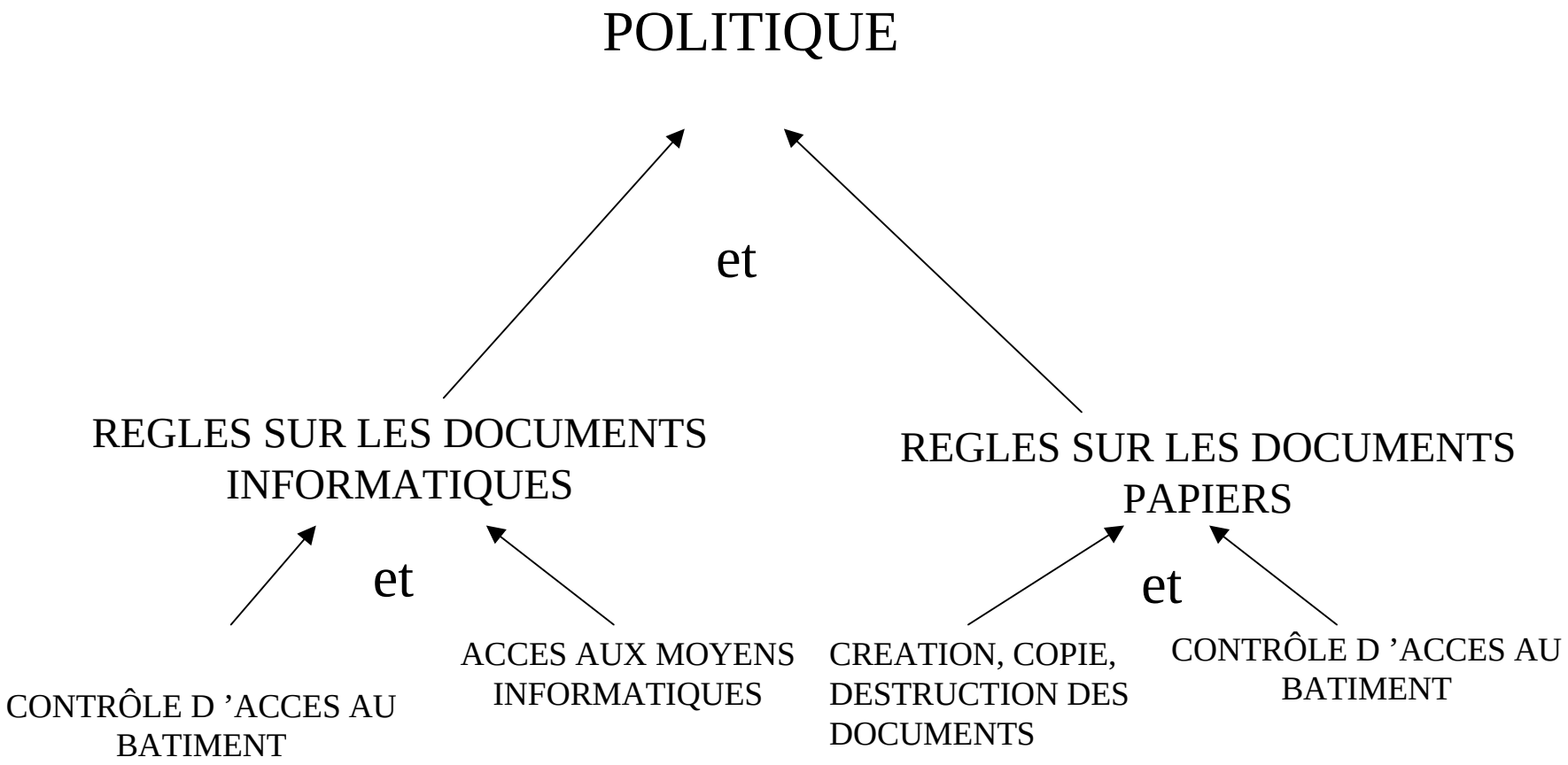
les actions possibles sont créer, détruire, lire, modifier un document, accréditer un médecin externe, autoriser l'accès à un dossier à un médecin externe. Les droits donnés sont, par exemple:

•Un droit illimité d'accès des dossiers par les membres du CHU
•Un droit d'accréditation d'un médecin ayant signé la convention accordée par A (procédure papier)
•Un droit de lecture de M à D, dossier d'un patient P,
accordé par par P et uniquement si M est accrédité (procédure papier)
•Un droit de modification sur le serveur de la table l'accréditation d'un médecin
TM accordé à un administrateur A ou des membres désignés d'une unités de soins US, tous membres du CHU….

Le niveau d ’attaque considéré est intermédiaire:
Agresseurs utilisant des techniques espionnage civil, disposant
de moyens matériels et financiers importants mais limité

La politique doit rester opérationnelle en présence de pannes catalectiques
(interruption de services) des systèmes physiques impliqués

Construction déductive des moyens
mis en oeuvre



3- Terminologie:
Propriétés de sécurité, Menaces et attaques

C'est la propriété qui assure la reconnaissance sûre de l ’identité d ’une entité

L'authentification protège de l'usurpation d'identité.

Signature (au sens classique) = Authentification:
La première idée contenue dans la notion habituelle de signature est que
le signataire est le seul à pouvoir réaliser le graphisme (caractérisation psychomotrice)

Entités à authentifier:
- une personne
- un programme qui s ’exécute (processus)
- une machine dans un réseau

C'est la propriété qui assure que l'auteur d'un acte ne peut ensuite dénier l'avoir effectué.

Signature (au sens habituel) = Authentification+Non répudiation :

La seconde idée contenue dans la notion habituelle de signature
est que le signataire s'engage à honorer sa signature:
engagement contractuel, juridique, il ne peut plus revenir en arrière.

Deux aspects spécifiques de la non répudiation dans les transactions électroniques:

a) La preuve d'origine

Un message (une transaction) ne peut être nié par son émetteur.

b) La preuve de réception

Un récepteur ne peut ultérieurement nier avoir reçu un ordre
s'il ne lui a pas plu de l'exécuter alors qu'il le devait juridiquement.

Exécution d'ordre boursier, de commande, ..

C'est la propriété qui assure qu'une information n'est modifiée que par des entités habilitées (selon des contraintes précises)

Une modification intempestive (même très temporaire)
est à interdite sur une écriture comptable validée

Le code binaires des programmes ne doit pas pouvoir être altéré

Les messages de l ’ingénieur système doivent pouvoir être lus et non modifiés

C'est la propriété qui assure qu'une information ne peut être lue que par des entités habilitées (selon des contraintes précises)

Un mot de passe ne doit jamais pouvoir être lu par un autre que son possesseur

Un dossier médical ne doit pouvoir être consulté que par les malades et le personnel médical habilité

C'est la propriété qui assure la capacité à détecter et à enregistrer de façon infalsifiable les tentatives de violation de la politique de sécurité.

Disponibilité :capacité de rendre un service
correct à un instant donné,
Fiabilité :capacité à rendre continûment un service correct

Relèvent de la terminologie de la **sûreté de fonctionnement**
On retiendra toutefois que les actions de sabotage
d'un système visent justement à diminuer sa disponibilité ou sa fiabilité

Détournement des données

Exemples: espionnage industriel , espionnage commercial, violations déontologiques

Détournement des logiciels

Exemple: copies illicites

LES MENACES AYANT POUR OBJECTIF LA FRAUDE OU LE SABOTAGE

Par modification des informations ou des dispositifs techniques et humains

Exemple : La fraude financière informatique, la destruction des informations (logique), le sabotage destiné à rendre inefficaces certaines fonctions (déni de service)

Déguisement (Mascarade)

Pour rentrer dans un système on essaye de piéger des usagers et de se faire passer pour quelqu'un d'autre:

Exemple: simulation d'interface système sur écran,
simulation de terminal à carte bancaire

Modification de messages, de données

Une personne non autorisée, un usager ou même un agent autorisé s'attribuent des avantages illicites en modifiant un fichier, un message (le plus souvent cette modification est réalisée par programme et entre dans la catégorie suivante)
Ex modification des données sur un serveur Web

Répétition ("replay")

Espionnage d'une interface, d'une voie de communication (téléphonique, réseau local) pour capter des opérations (même cryptées elles peuvent être utilisables)

Répétition de l'opération pour obtenir une fraude.

Exemple: Plusieurs fois la même opération de crédit d'un compte bancaire.

Modification des programmes

Les modifications à caractère frauduleux

Pour s'attribuer par programme des avantages.
Exemple: virement des centimes sur un compte

Les modifications à caractère de sabotage

Pour détruire avec plus ou moins de motivations des systèmes ou des données

Deux types de modifications

a) Infections informatiques à caractère unique

Bombe logique ou cheval de Troie

- Dans un programme normal on introduit un comportement illicite mis en action par une condition de déclenchement ou trappe
(la condition, le moment ou l'on bascule d'un comportement normal à anormal)

Exemples:licenciement de l'auteur du programme

b) Infections auto reproductrices

Il s'agit d'une infection informatique simple (du type précédent)
qui contient de plus une partie de recopie d'elle même afin d'en assurer la propagation

***Virus* :** à action brutale

Ver : à action lente (détruisant progressivement les ressources d'un systèmes)

QUELQUES CLASSES DE VIRUS
(implantation)

- Les virus à secteur d ’amorçage
- Les virus à infection de fichiers
- Les macro virus
- Les virus furtifs
- Les virus polymorphes (mutants)
- Les virus réseaux

Les attaques ayant pour but le vol d'information via un réseau par **espionnage des transmissions de données** (espion de ligne, accès aux données dans des routeurs et des serveurs Internet)

Canaux cachés

Analyse de trafic

On observe le trafic de messages échangés pour en déduire des informations sur les décisions de quelqu'un.

Exemples:
Bourse : augmentation des transactions sur une place financière.
Militaire : le début de concentration entraîne un accroissement de trafic important.

Inférence

On obtient des informations confidentielles à partir d'un faisceau de questions autorisées
(et d'un raisonnement visant à faire ressortir l'information).

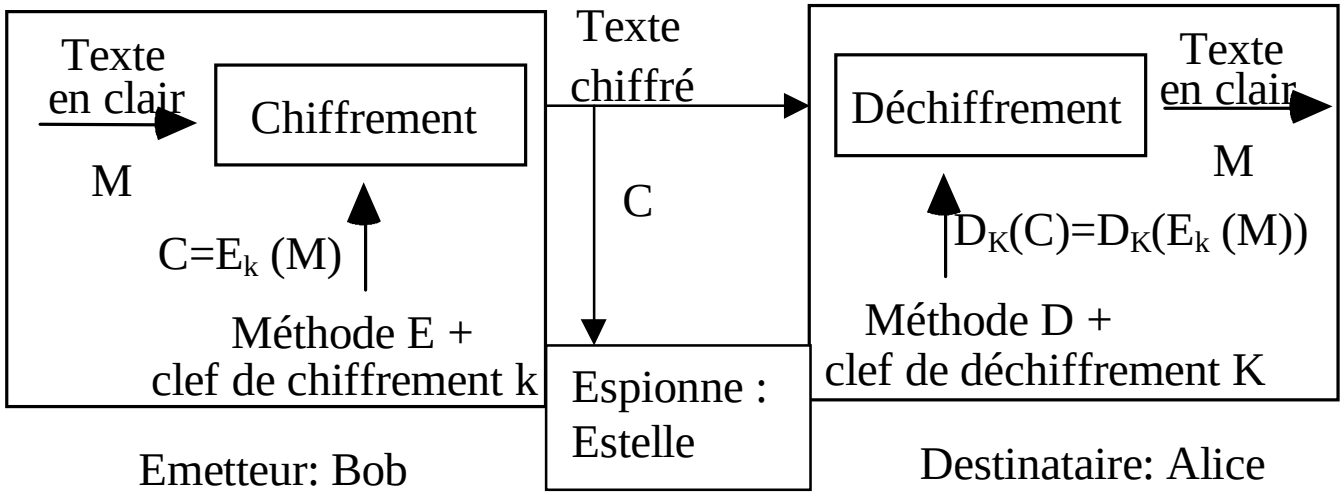
Attaque par violation de protocole

Erreur très rare en fonctionnement normal et non supportées par le protocole

Attaque par saturation

Envoi de messages trop nombreux provoquant un écroulement des systèmes et réseaux

4- Introduction à la cryptographie et aux protocoles de sécurité


Texte
en clair
M
Chiffrement
C=Ek (M)
Méthode E +
clef de chiffrement k
Emetteur: Bob
Texte
chiffré
C
Espionne :
Estelle
Déchiffrement
Texte
en clair
M
DK(C)=DK(Ek (M))
Méthode D +
clef de déchiffrement K
Destinataire: Alice

Bob, doit transmettre à Alice, un message M ∈ MESSAGES_A_ENVOYER.
M est dit “en clair“.

Estelle, une espionne, d’écouter la voie de communication pour connaître M.

Bob, construit un texte chiffré C∈ MESSAGES_CHIFFRES.

$$C = E_k(M). \quad \text{ou} \quad C = \{M\}_k^E$$

La fonction E_k dépend d’un paramètre k appelé clef de chiffrement.

Le **chiffrement** est donc une transformation d'un texte pour en cacher le sens

La possibilité de chiffrer repose donc sur la connaissance de l’algorithme de chiffrement E et de la clef k de chiffrement.

Le **déchiffrement** est l'opération inverse permettant de récupérer le texte en clair à partir du texte C chiffré.
Il repose sur la fonction D_K de MESSAGES_CHIFFRES dans MESSAGES_A_ENVOYER telle que

$$M= D_{K'}(C) \text{ ou } C = \{M\}_K^D$$

On doit avoir

$$D_K (E_k(M))=M$$

D_K est donc une fonction inverse à gauche de E_k.

Pour un couple cr = (E,D) donné de famille de fonction de chiffrement et de déchiffrement, l'ensemble des couples (k, K) vérifiant cette propriété est noté CLE(cr).

Pour que ces opérations assurent la confidentialité du transfert entre Alice et Bob, il est nécessaire qu'au moins une partie des informations E, D, k, K soit ignorée du reste du monde.

Décrypter ou casser un code c'est parvenir au texte en clair sans posséder au départ ces informations secrètes. C'est l'opération que doit réaliser Estelle pour retrouver M.

L'art de définir des codes est la cryptographie. Un spécialiste en cryptographie est appelé cryptographe.

L'art de casser des codes est appelé cryptanalyse ou cryptologie. Un spécialiste en cryptanalyse est appelé cryptanalyste.

Un crypto-système est l'ensemble des deux méthodes de chiffrement et de déchiffrement utilisable en sécurité.

Tels que soit k=K, soit la connaissance d'une des deux clefs permet d'en déduire facilement l'autre.

Conséquences :
Dichotomie du monde : les bons et les mauvais

Multiplication des clefs (un secret n'est partagé que par 2 interlocuteurs), donc pour N interlocuteurs N.(N-1)/2 couples

La qualité d'un crypto système symétrique s'analyse par rapport à des propriétés statistiques des textes chiffrés et la résistance aux classes d'attaques connues.

En pratique tant qu'un crypto système symétrique n'a pas été cassé, il est bon, après il est mauvais.

Tels que la connaissance de k (la clef de chiffrement) ne permet pas d'en déduire celle de K(la clef de déchiffrement).

La clef k est appelée la **clef publique**, la clef K est appelée la **clef privée**.

Fondement théorique : montrer que la recherche de K à partir de k revient à résoudre un problème mathématique notoirement très compliqué, c'est à dire demandant un grand nombre d'opérations et beaucoup de mémoire pour effectuer les calculs.

RSA (l'algorithme le plus utilisé à l'heure actuel) la déduction de K à partir de k revient à résoudre le problème de factorisation d'un grand nombre un problème sur lequel travaille les mathématiciens depuis plus de 2000 ans,

On estime que le plus rapide ordinateur que l'on puisse construire utilisant la meilleure méthode connue met plus de 1000 ans pour retrouver la clef privée d'un système RSA utilisant un modulo de 1024 bits (ordre de grandeur de la taille des clefs).

Une fonction de hachage h est une fonction qui à un message M de longueur quelconque fait correspondre un message H(M) (notée aussi $\{M\}^H$) de longueur constante.
L'intérêt d'une fonction de hachage est que M peut être arbitrairement grand alors que H(M) a une longueur donnée.
Terminologie: Résumé, fonction de contraction, digest, empreinte digitale, …

Exemple: Hasch codes des systèmes de fichiers, codes détecteurs d 'erreurs

f(M) telle que f est une fonction de hachage par rapport à M

f est à collision faible difficile: il est calculatoirement difficile de trouver M significatif tel que f(M)=K

f est à collision forte difficile:il est calculatoirement difficile de trouver M et M’ tel que f(M)=f(M ’)

Elle est avec clef si son calcul dépend d ’une information secrète la clef K

Une signature manuscrite idéale est réputée posséder les propriétés suivantes:

- La signature **ne peut-être imitée.**
Elle prouve que le signataire a délibérément signé le document.

- La signature **authentifie** le signataire.
Seul le signataire peut avoir signé.

- La signature appartient à un seul document (elle **n'est pas réutilisable**).

- Le document signé ne peut être partiellement ou totalement **modifié**.

- La signature ne peut-être **reniée**.

-La signature peut être **contrôlée**

Base de la signature numérique:une fonctions de hachage
H sécuritaire et d ’une fonction à sens unique f avec brèche.

La signature est composée de $f^{-1}(H(M))$

Seul le signataire sait calculer f^{-1}
Tout le monde peut calculer H et f et donc pour M donné vérifier la signature

Si H est a collision faible, on ne pourra pas coller une signature sur un document à créer

Si h est à collision forte difficile Estelle ne pourra pas fabriquer 2 documents,
un signable par Bob ,l ’autre pas ayant le même hache donc la même signature

A: clef d ’Alice (chiffrement symétrique)
a : clef d ’Alice (déchiffrement symétrique)
A: clef privée de Alice (déchiffrement asymétrique)
a: clef publique de Alice (chiffrement asymétrique)

$\{X\}_{Clef}^{CRY}$ Chiffrement /Déchiffrement selon le crypto sytème CRY avec la clef Clef

Crypto systèmes symétriques

$\{X\}_{a}^{SYM}$ Chiffrement $\{X\}_{A}^{SYM}$ Déchiffrement

Crypto systèmes asymétriques

$\{X\}_{a}^{ASY}$ Chiffrement (clef publique) $\{X\}_{A}^{ASY}$ Déchiffrement (clef privée)

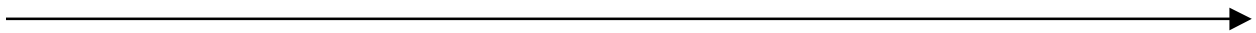
$\{X\}^{H}$ Résumé de sécurité $\{X\}_{A}^{SIG} = \{\{X\}^{H}\}_{A}^{ASY}$ Signature de X par Alice

Format des messages :Type, Emetteur, Destinataire, Contenu

Alice

Bob

M



Dans le protocole Alice envoie M à Bob

ANNUAIRE DES CERTIFICATS

NOM	Clef	Validité	Extensions	Signature
Alice	a	Valida	Para	$\{Alice, a, Valida, Para\}^{SIG}_{AC}$
Bob	b	Validb	Parb	$\{Bob, b, Validb, Parb\}^{SIG}_{AC}$
Charles	c	Validc	Parc	$\{Charles, c, Validc, Parc\}^{SIG}_{AC}$

AC: autorité de certification

Norme de représentation des certificats X509

Norme de protocole d'accès: LDAP

Toutes entités impliquées dans un schéma à clef publique doit détenir la clef publique de l ’autorité de certification.

Tout accès à un certificat doit être contrôlé:
Vérifier que la signature est valide
Vérifier que la date courante est dans la période de validité

Pour éviter les rejeux de certificats invalidés le serveur d ’annuaire doit :
Soit s’authentifier
Soit dater et signer sa réponse
Soit transmettre périodiquement des listes de révocation datée et signées

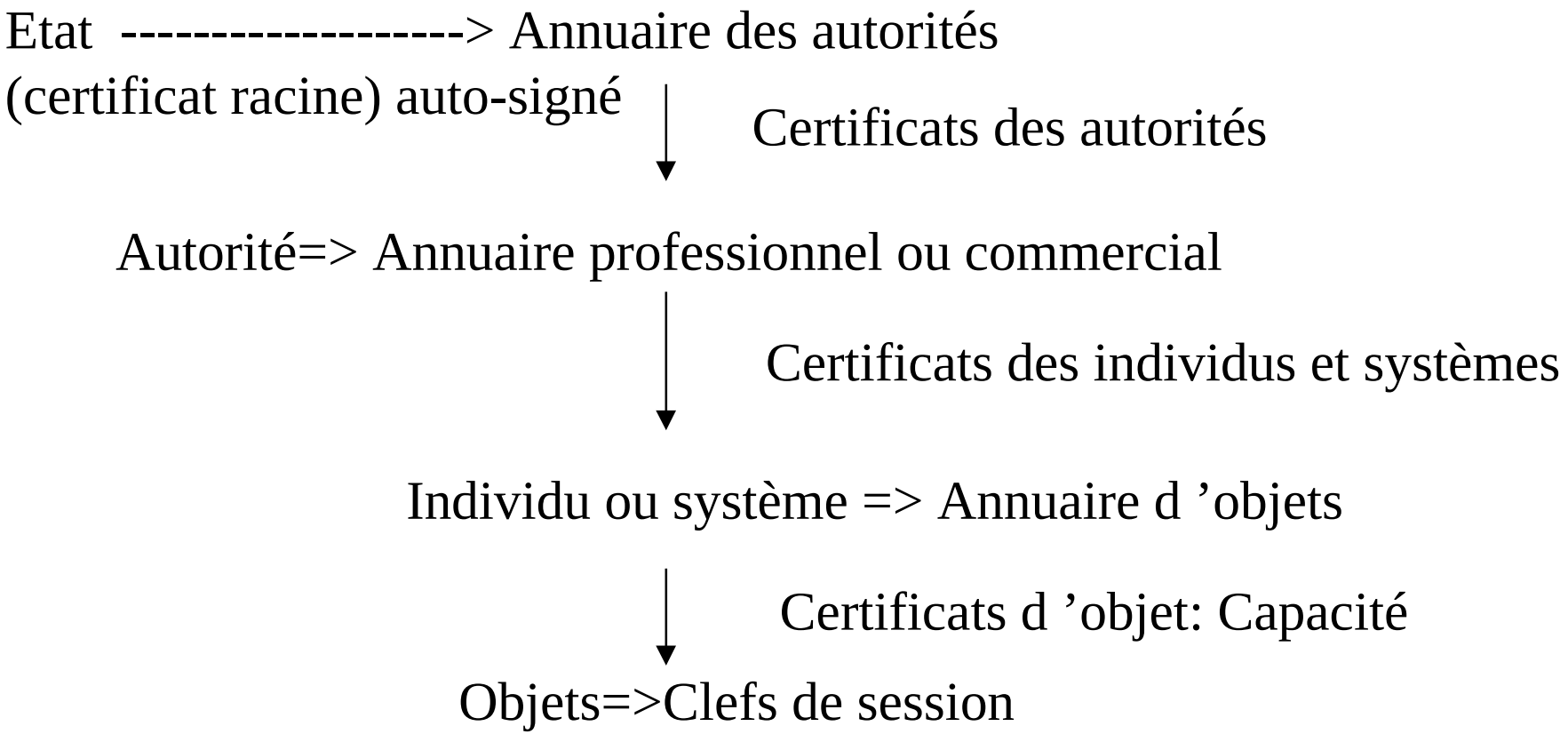
Clef publique de l’autorité, ne doit pas pouvoir être modifiée: Dans le code en dur , sur un support fiable (carte à puce)

Clef privée de l ’utilisateur, ne doit pas pouvoir être lue: sur un support confidentiel (carte à puce) ou un fichier chiffré avec un mot de passe (local au poste ou sur disquette)

Certificat de l ’utilisateur: Annuaire+support local ou carte ou disquette

Annuaire: Annuaire central+version locales (cache, annuaire privé

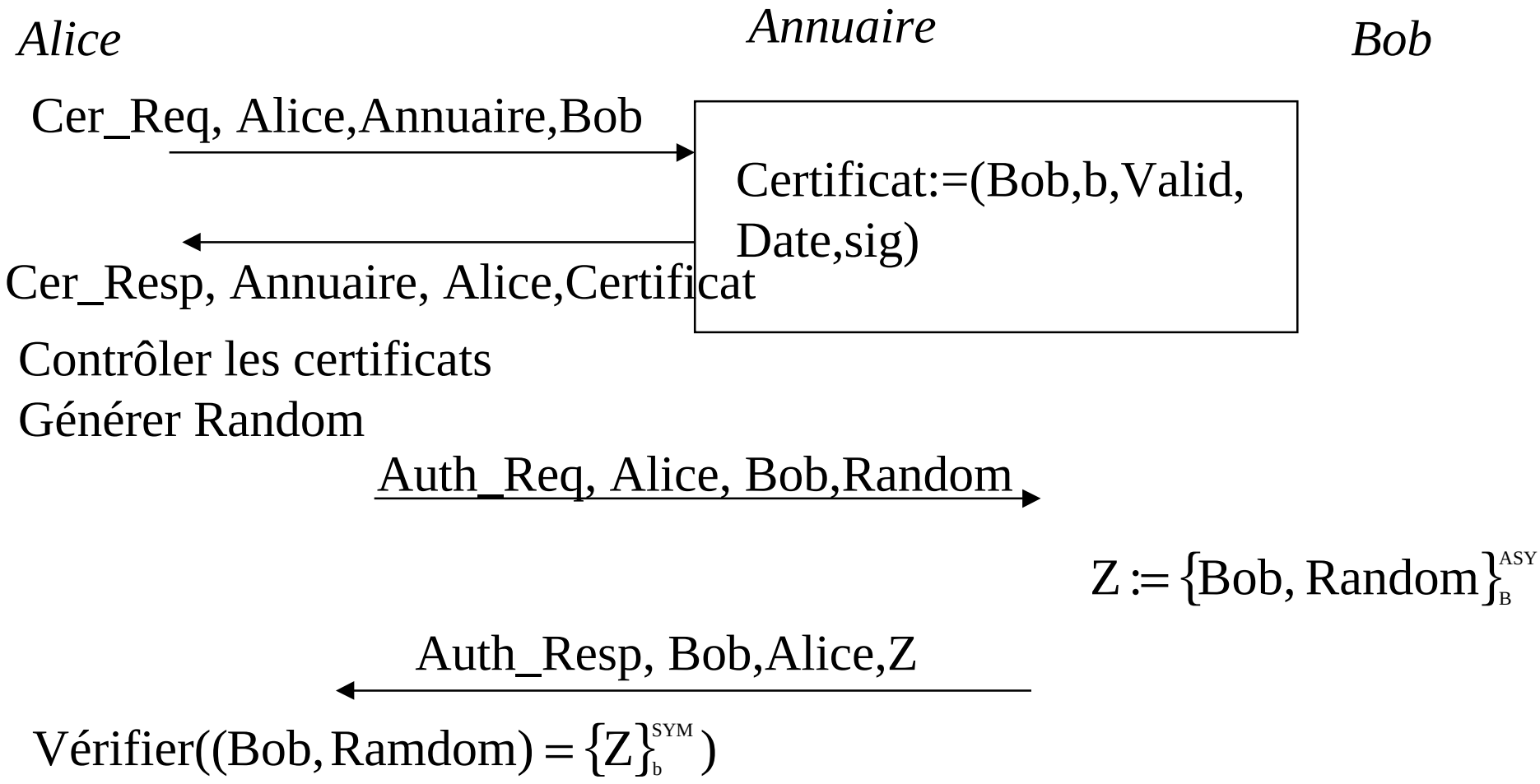
SYSTÈME ASYMÉTRIQUE:
HIÉRARCHIE DES AUTORITÉS DE CERTIFICATION
(CHAÎNE DE CERTIFICATION)



Protocole permettant à Bob de prouver à Alice qu'il est Bob

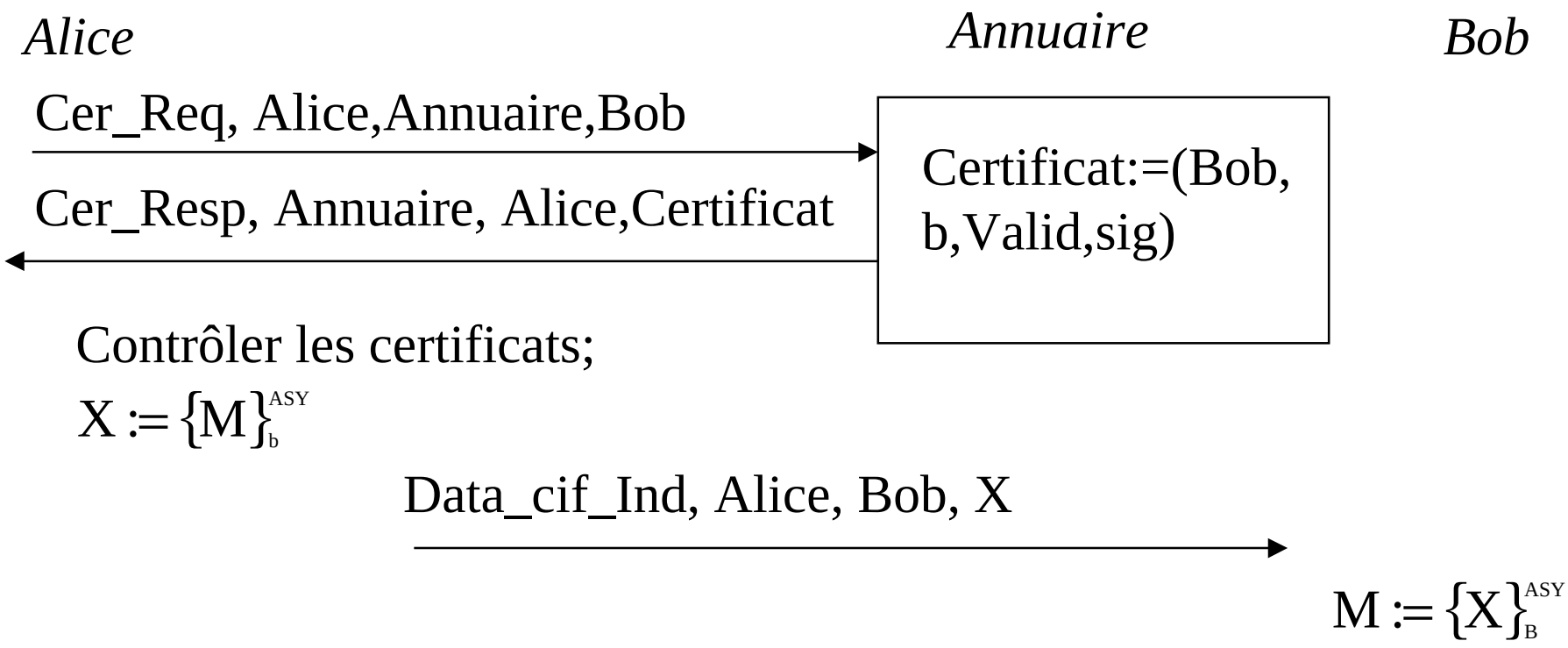
Bob détient un secret sur lequel repose l'authentification
Bob ne doit pas révéler le secret à Alice

Il existe un tiers fiable qui a authentifié Bob
(gardien des clefs ou annuaire de certificats)


Alice
Annuaire
Bob
Cer_Req, Alice,Annuaire,Bob
Certificat:=(Bob,b,Valid, Date,sig)
Cer_Resp, Annuaire, Alice,Certificat
Contrôler les certificats
Générer Random
Auth_Req, Alice, Bob,Random
$Z := \{Bob, Random\}_B^{ASY}$
Auth_Resp, Bob,Alice,Z
$Vérifier((Bob, Ramdom) = \{Z\}_b^{SYM})$

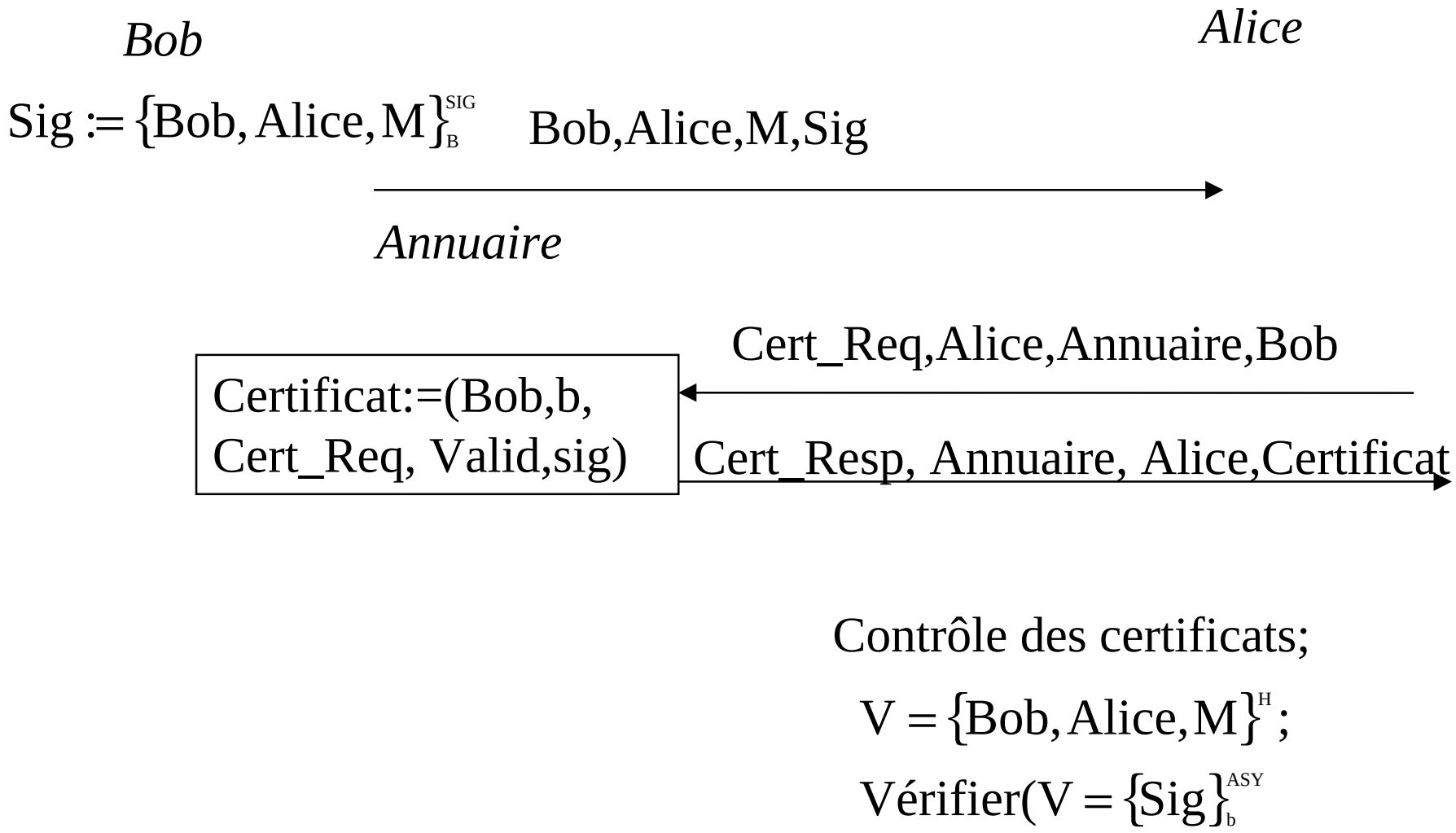
Alice doit transmettre à Bob un message que eux seuls doivent connaître

CONFIDENTIALITÉ AVEC CHIFFRE ASYMÉTRIQUE



Très peu utilisé car très lent, sert à échanger des clefs d ’algorithmes symétriques beaucoup plus rapides
On échange ainsi une clef de session pour chiffre symétrique

Alice doit envoyer à Bob un message, tel que Bob puisse contrôler que le message n’a pas été modifié et a bien été créé par Alice


Bob
Alice
$Sig := \{Bob, Alice, M\}_B^{SIG}$
Bob,Alice,M,Sig
Annuaire
Cert_Req,Alice,Annuaire,Bob
Certificat:=(Bob,b,
Cert_Req, Valid,sig)
Cert_Resp, Annuaire, Alice,Certificat
Contrôle des certificats;
$V = \{Bob, Alice, M\}^H$;
Vérifier($V = \{Sig\}_b^{ASY}$

Intégrité d ’un message: problème voisin de la signature
Utilisation de fonction de Hachage sécuritaire ou de MAC basé sur un chiffre symétrique en mode chaîné

Intégrité du flot de message: Possibilité de rejeu
Utilisation d ’un **Nonce** (Used Only Once), qui distingue chaque message:
Numéro de séquence sur un modulo grand
Heure
Nombre aléatoire

5- les critères communs

L ’APPROCHE ITSEC, CRITÈRES COMMUNS

1) L'objectif de sécurité (Politique de sécurité pour un système ou argumentaire du produit) qui précise le besoin qui tend à être satisfait, les menaces qui doivent être couvertes et l'environnement d'exploitation visé,

2) La spécification fonctions dédiées à la sécurité,

3) La définition des mécanismes de sécurité utilisés, c'est à dire les principes de la solution retenue

4) La cotation annoncée de la résistance minimum des mécanismes. Cette cotation se fait selon trois niveaux: faible (attaque accidentelle et aléatoire), moyenne (attaque volontaire reposant sur des moyens limités), forte (attaque par des experts disposant de moyens importants).

5) Le niveau d'évaluation en conformité visé

MODELE DE DEVELOPPEMENT

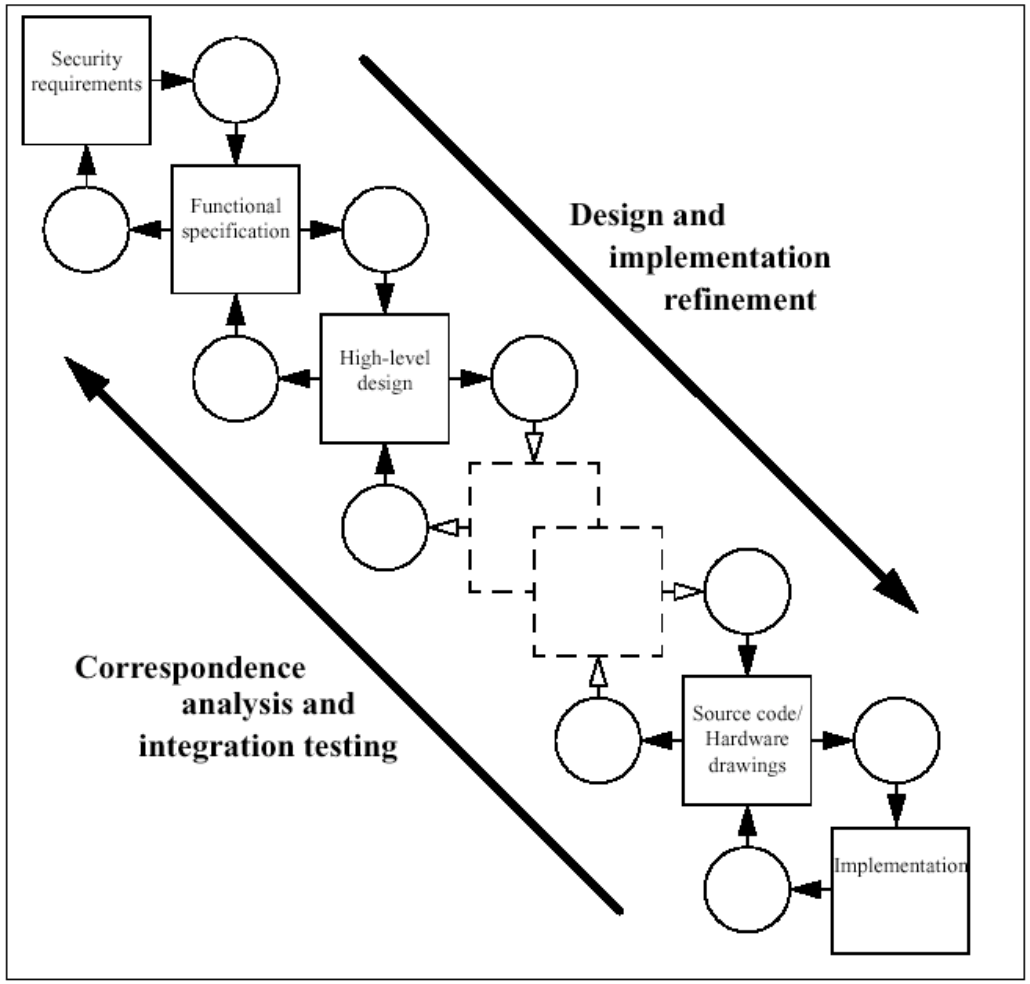


Figure 4.3 - TOE development model

PROCESSUS D ’EVALUATION

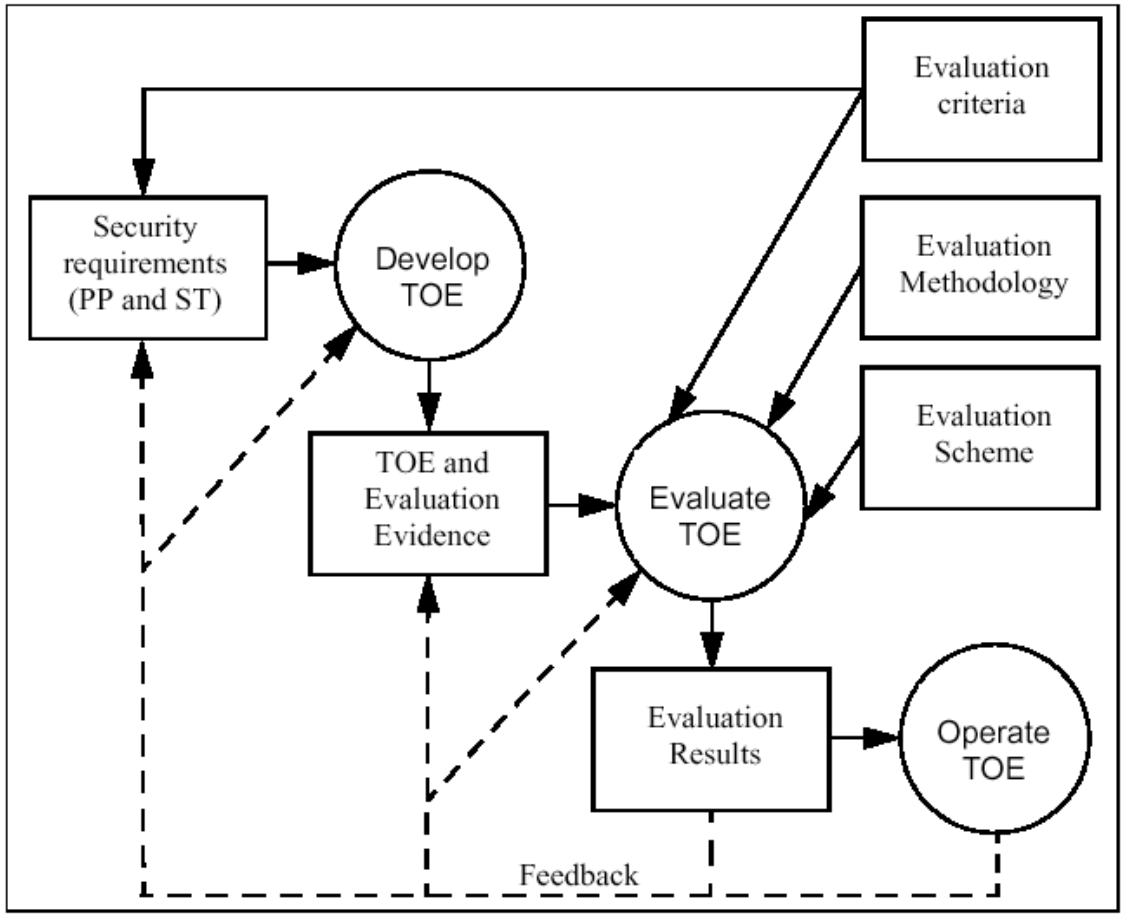


Figure 4.4 - TOE evaluation process

CONSTRUCTION DES OBJECTIFS

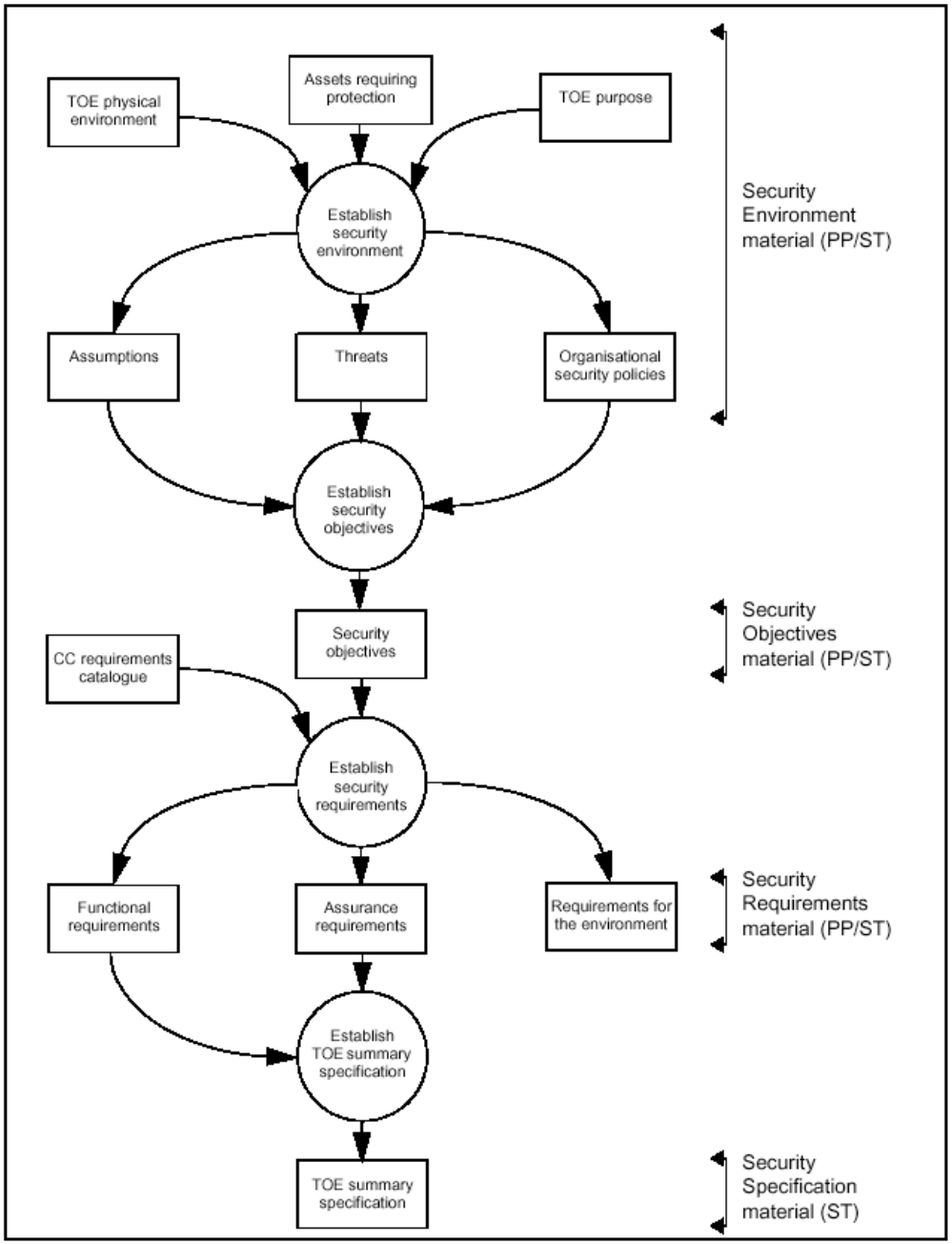


Figure 4.5 - Derivation of requirements and specifications

PROCESSUS DE CERTIFICATION

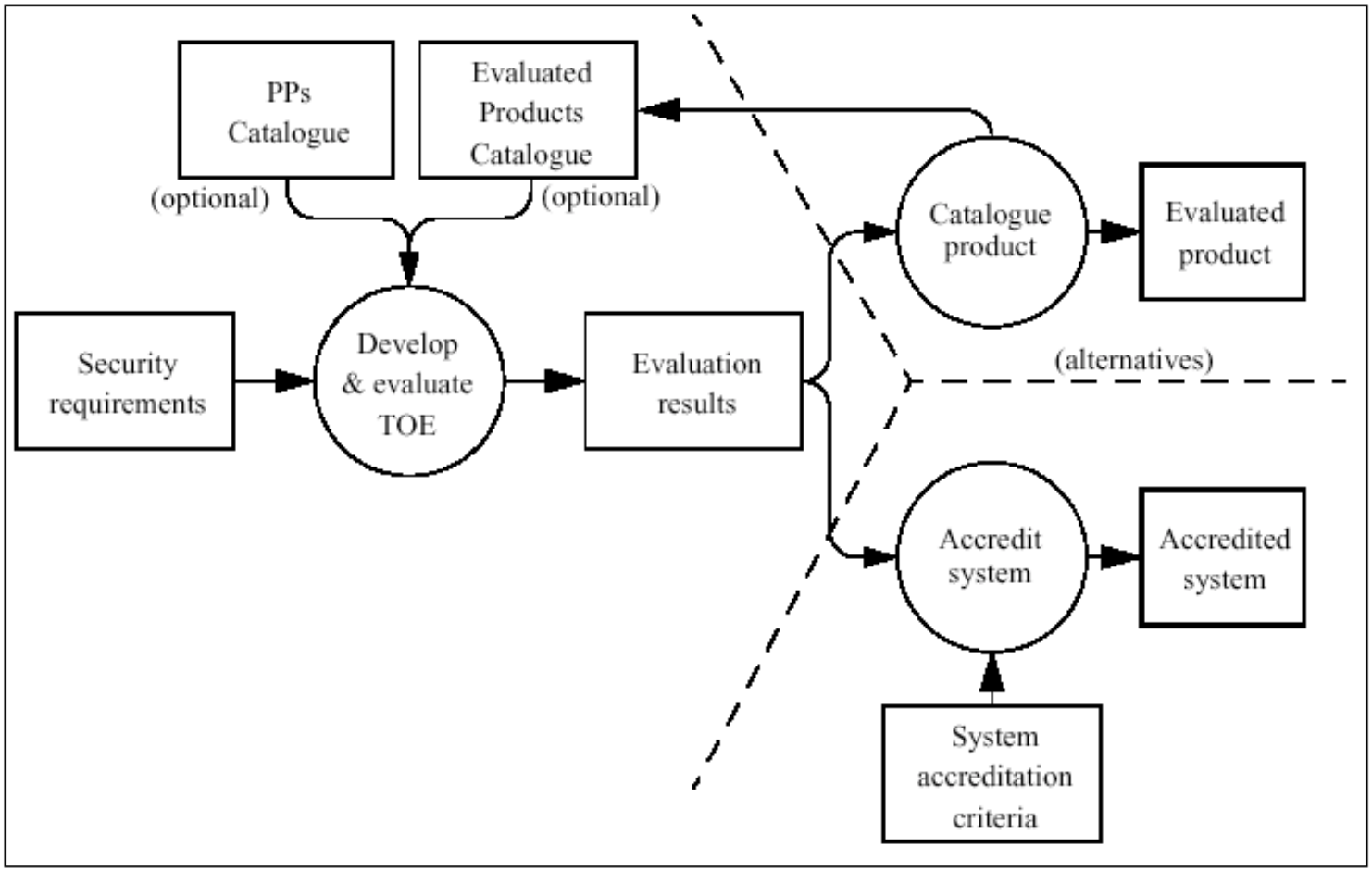


Figure 5.2 - Use of TOE evaluation results

ORGANISATION DES OBJECTIFS

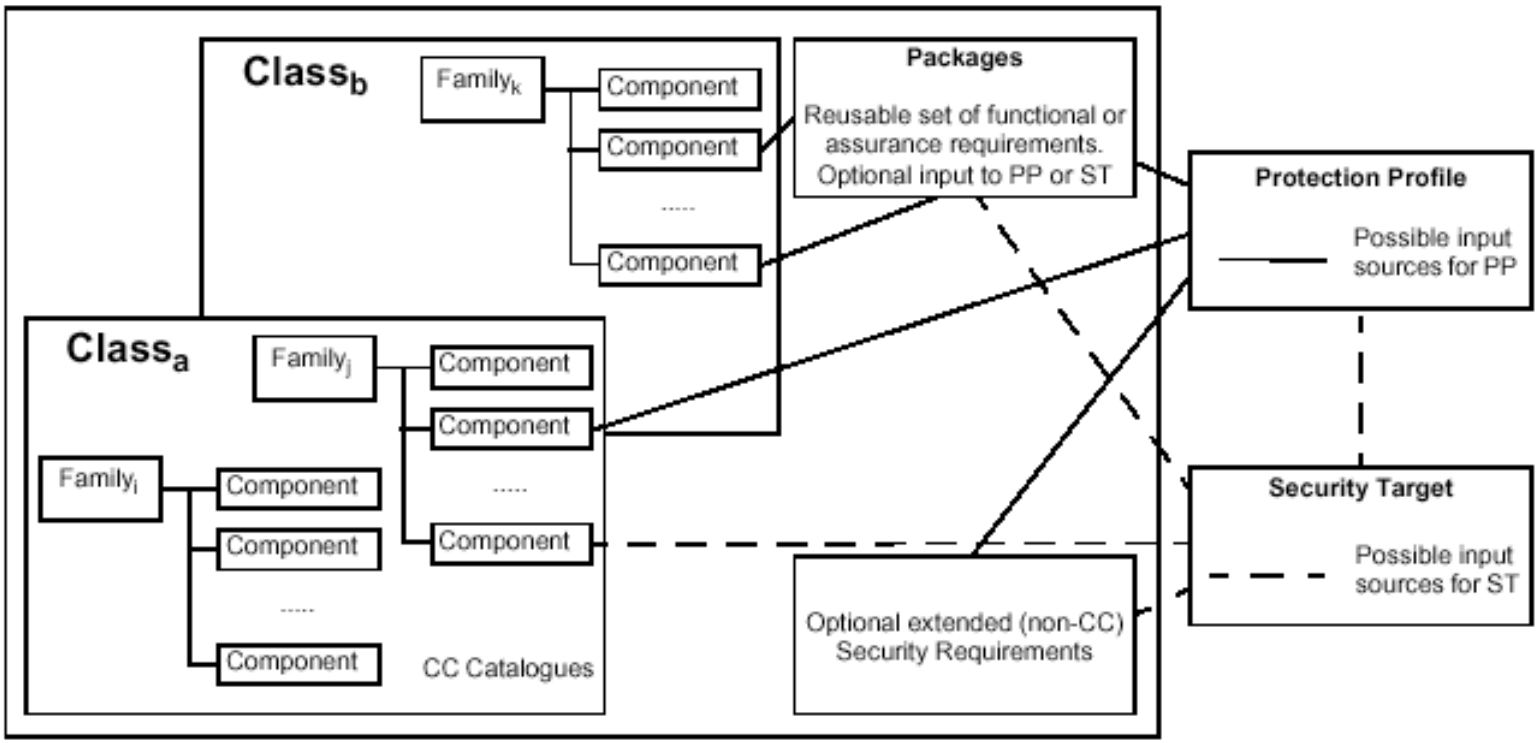


Figure 4.6 - Organisation and construction of requirements

PROFIL DE PROTECTION

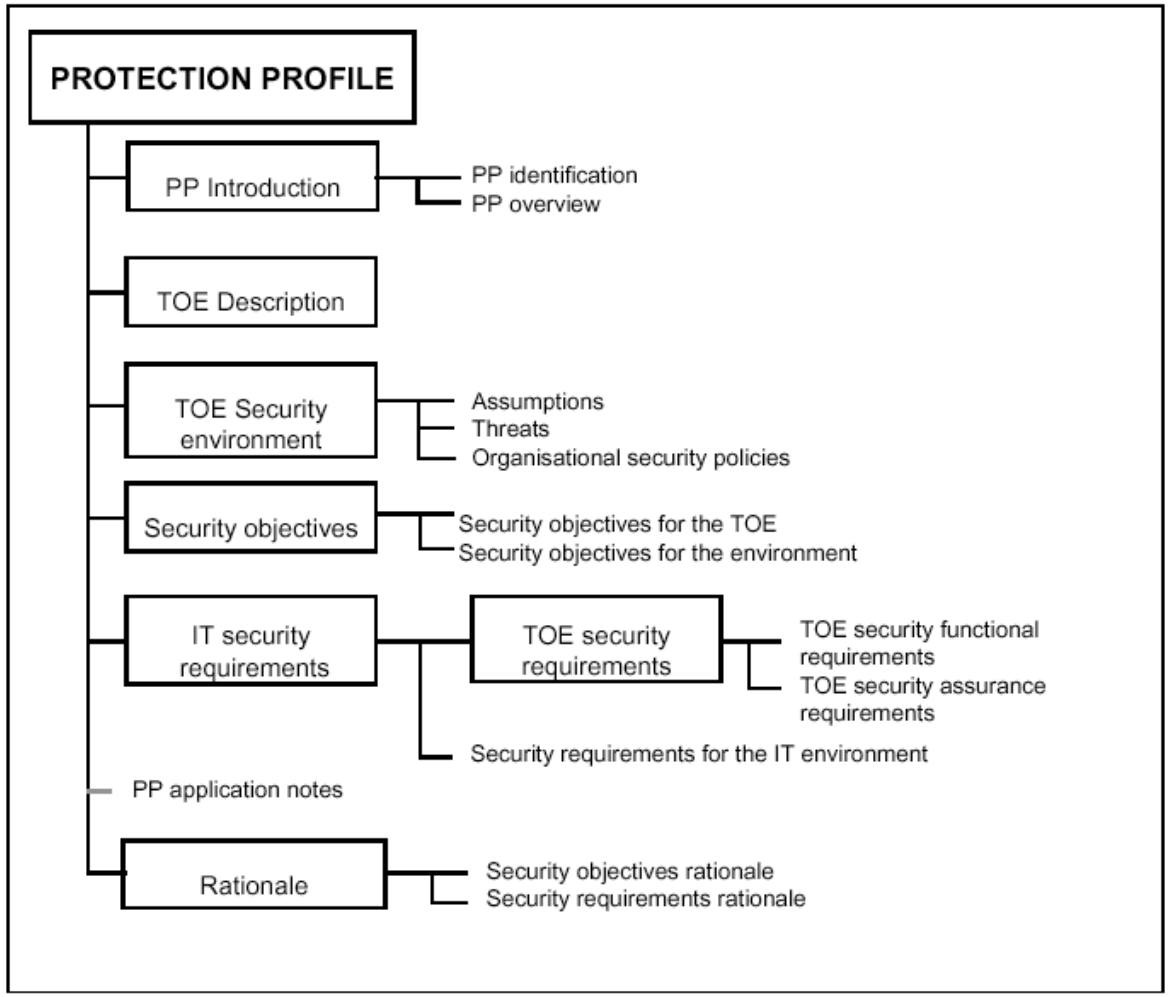


Figure B.1 - Protection Profile content

CIBLE DE SECURITE

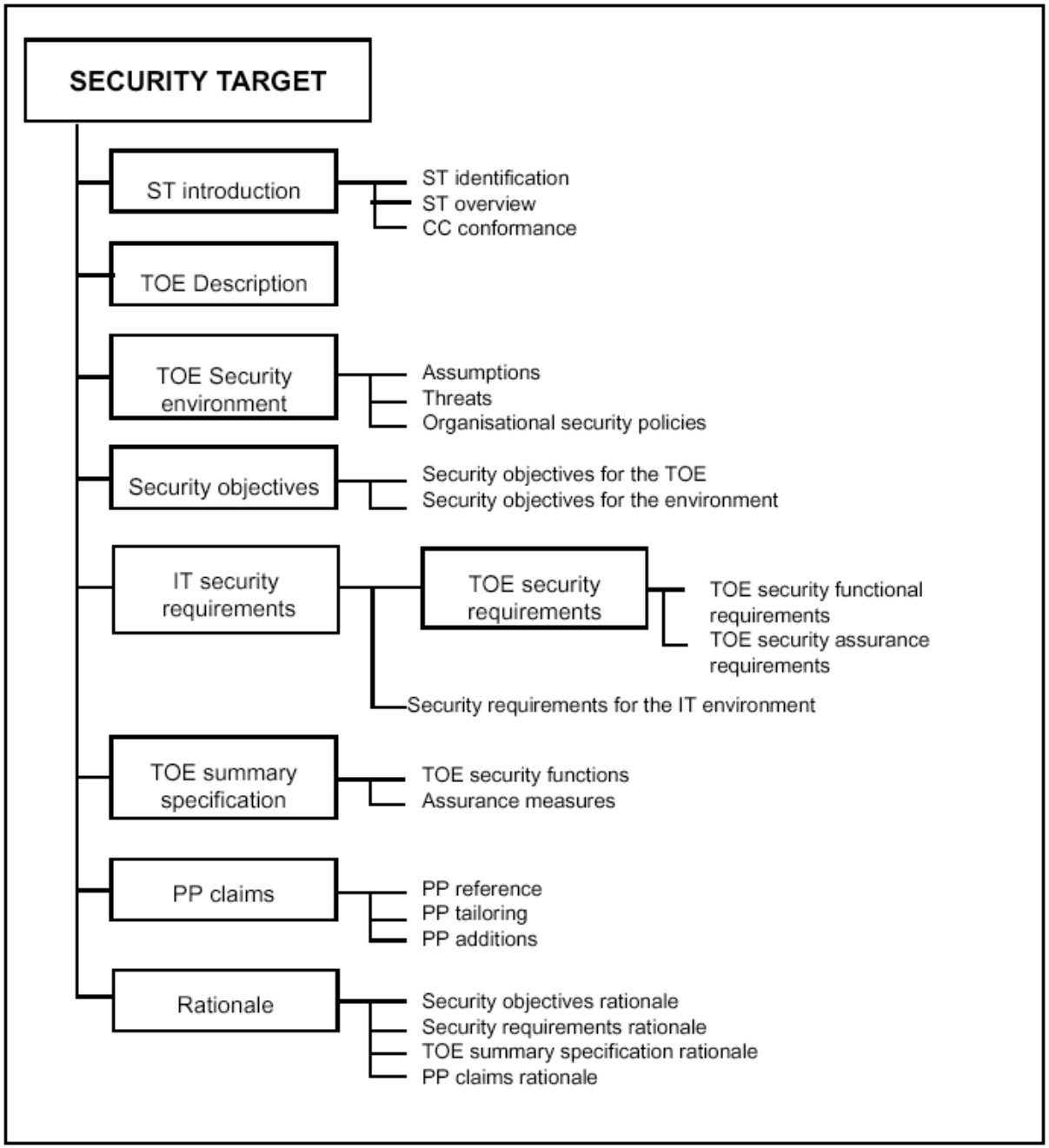
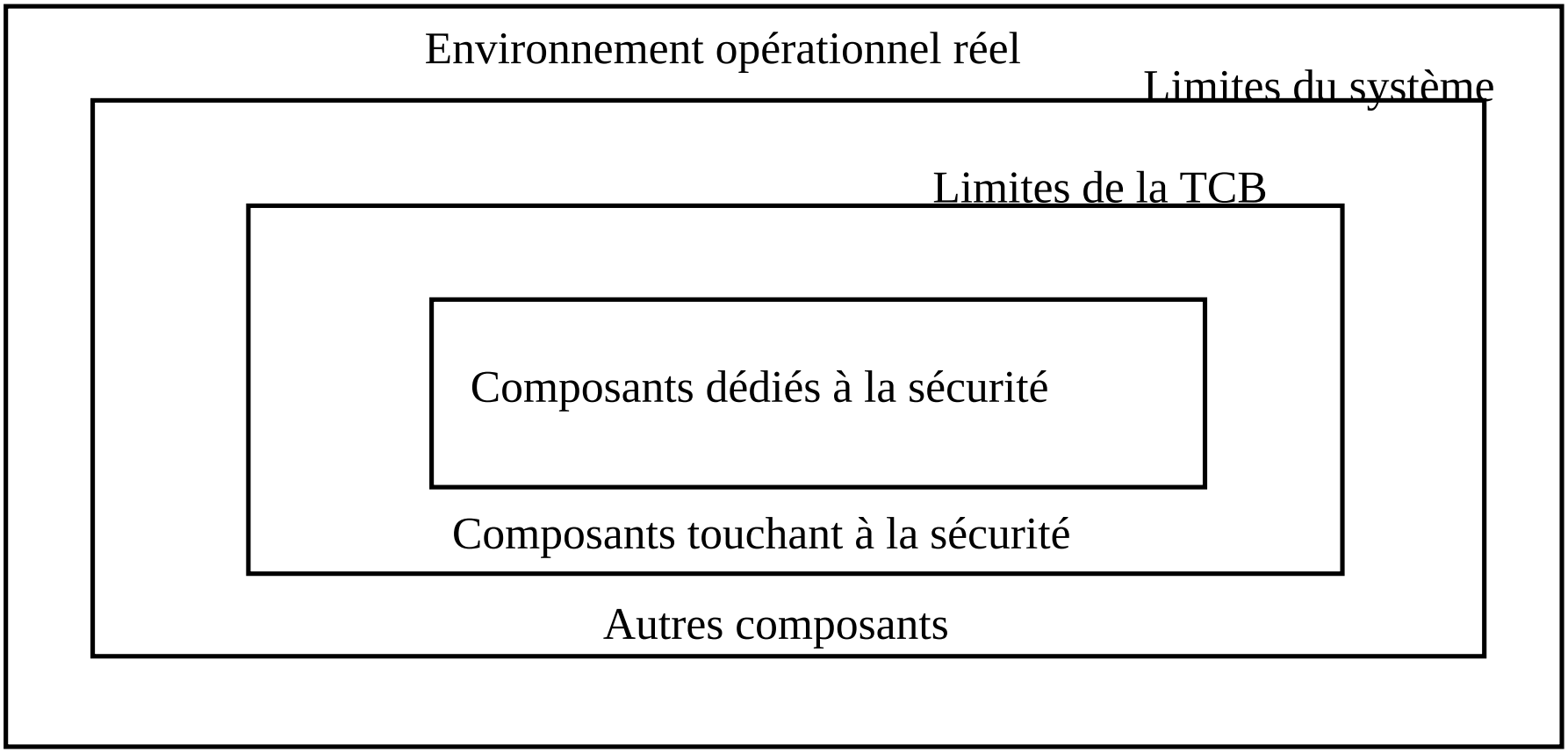


Figure C.1 - Security Target content

CIBLE DE SÉCURITÉ



LES COMPOSANST DE LA CIBLE

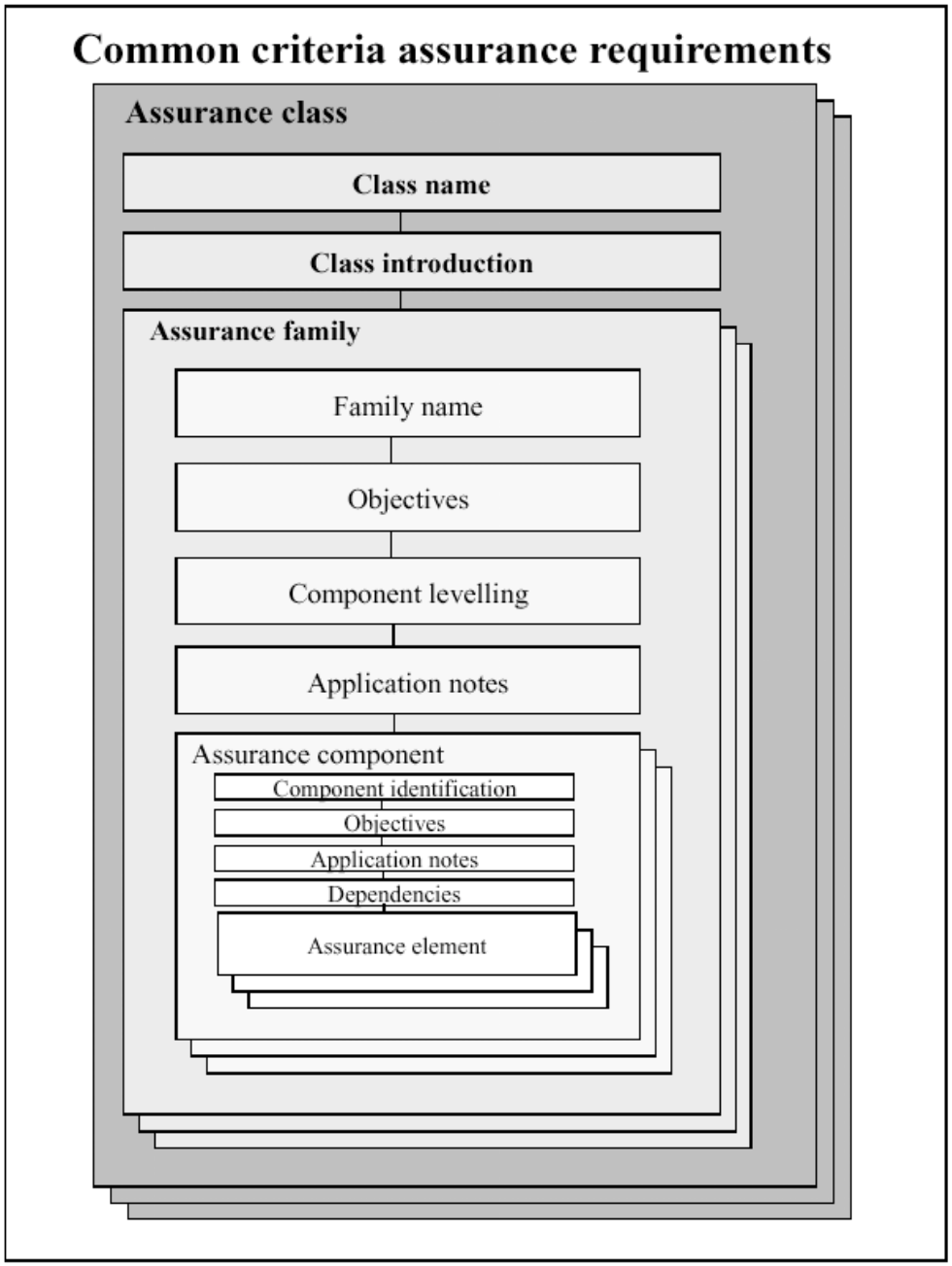


Figure 2.1 - Assurance class/family/component/element hierarchy

LES COMPOSANTS D ’ASSURANCE

Assurance Class	Assurance Family	Abbreviated Name
Class ACM: Configuration management	CM automation	ACM_AUT
	CM capabilities	ACM_CAP
	CM scope	ACM_SCP
Class ADO: Delivery and operation	Delivery	ADO_DEL
	Installation, generation and start-up	ADO_IGS
Class ADV: Development	Functional specification	ADV_FSP
	High-level design	ADV_HLD
	Implementation representation	ADV_IMP
	TSF internals	ADV_INT
	Low-level design	ADV_LLD
	Representation correspondence	ADV_RCR
	Security policy modeling	ADV_SPM
Class AGD: Guidance documents	Administrator guidance	AGD_ADM
	User guidance	AGD_USR
Class ALC: Life cycle support	Development security	ALC_DVS
	Flaw remediation	ALC_FLR
	Life cycle definition	ALC_LCD
	Tools and techniques	ALC_TAT
Class ATE: Tests	Coverage	ATE_COV
	Depth	ATE_DPT
	Functional tests	ATE_FUN
	Independent testing	ATE_IND
Class AVA: Vulnerability assessment	Covert channel analysis	AVA_CCA
	Misuse	AVA_MSU
	Strength of TOE security functions	AVA_SOF
	Vulnerability analysis	AVA_VLA

LES NIVEAUX D'ASSURANCE

Evaluation assurance levels .....

Evaluation assurance level (EAL) overview .....

Evaluation assurance level details .....

EAL1 - functionally tested .....

EAL2 - structurally tested .....

EAL3 - methodically tested and checked .....

EAL4 - methodically designed, tested, and reviewed

EAL5 - semiformally designed and tested .....

EAL6 - semiformally verified design and tested ....

EAL7 - formally verified design and tested .....

PAR NIVEAUX

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Class ACM: Configuration management	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Class ADO: Delivery and operation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Class ADV: Development	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
Class AGD: Guidance documents	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Class ALC: Life cycle support	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Class ATE: 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**
Class AVA: Vulnerability assessment	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

EAL 1

Assurance class	Assurance components
Class ACM: Configuration management	ACM_CAP.1 Version numbers
Class ADO: Delivery and operation	ADO_IGS.1 Installation, generation, and start-up procedures
Class ADV: Development	ADV_FSP.1 Informal functional specification
	ADV_RCR.1 Informal correspondence demonstration
Class AGD: Guidance documents	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Class ATE: Tests	ATE_IND.1 Independent testing - conformance

EAL 3

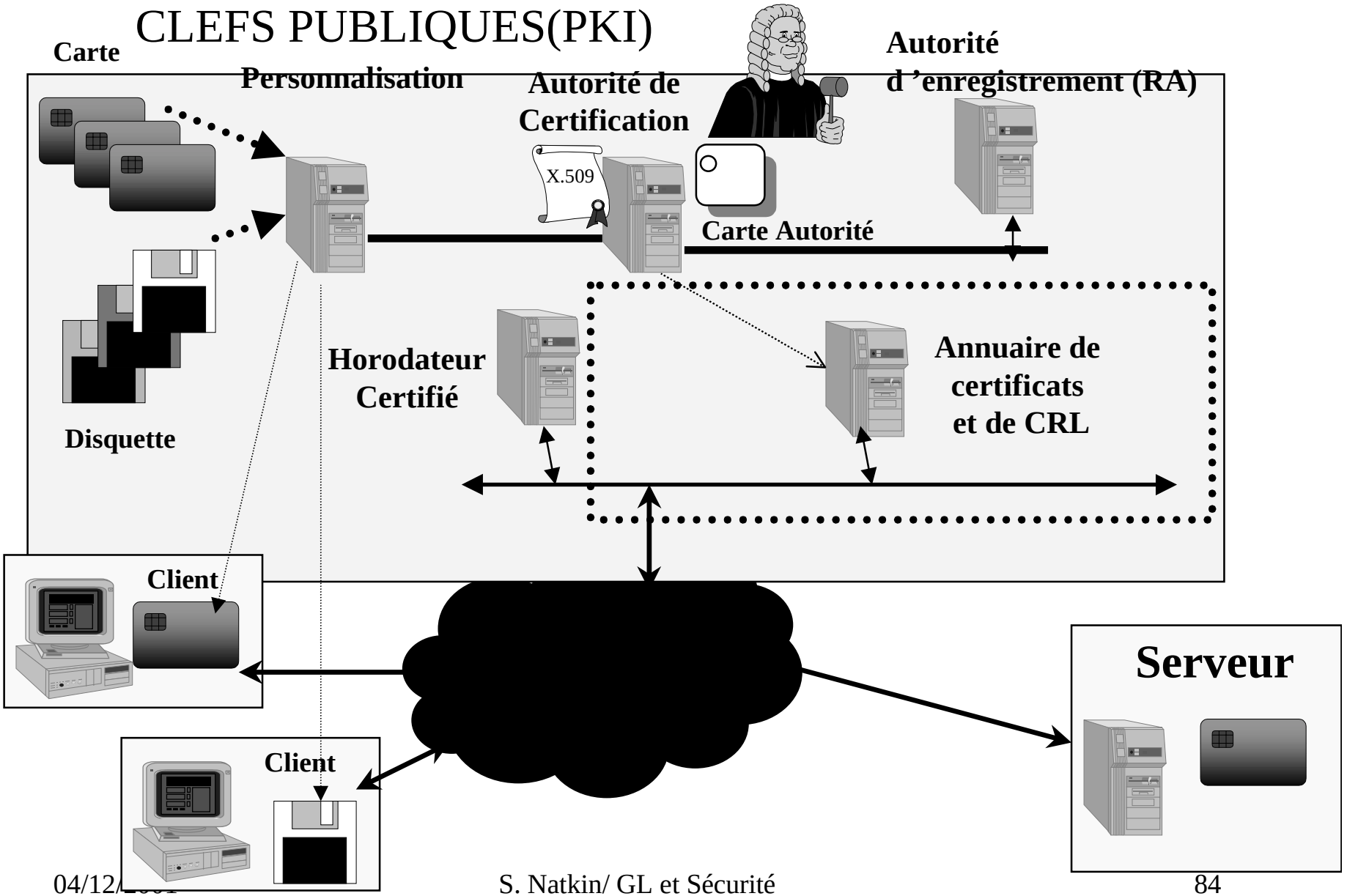
Assurance class	Assurance components
Class ACM: Configuration management	**ACM_CAP.3 Authorisation controls**
	ACM_SCP.1 TOE CM coverage
Class ADO: Delivery and operation	ADO_DEL.1 Delivery procedures
	ADO_IGS.1 Installation, generation, and start-up procedures
Class ADV: Development	ADV_FSP.1 Informal functional specification
	ADV_HLD.2 Security enforcing high-level design
	ADV_RCR.1 Informal correspondence demonstration
Class AGD: Guidance documents	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Class ALC: Life cycle support	**ALC_DVS.1 Identification of security measures**
Class ATE: Tests	**ATE_COV.2 Analysis of coverage**
	ATE_DPT.1 Testing: high-level design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
Class AVA: Vulnerability assessment	**AVA_MSU.1 Examination of guidance**
	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.1 Developer vulnerability analysis

E
A
L
7

Assurance class	Assurance components
Class ACM: Configuration management	ACM_AUT.2 Complete CM automation
	ACM_CAP.5 Advanced support
	ACM_SCP.3 Development tools CM coverage
Class ADO: Delivery and operation	**ADO_DEL.3 Prevention of modification**
	ADO_IGS.1 Installation, generation, and start-up procedures
Class ADV: Development	**ADV_FSP.4 Formal functional specification**
	ADV_HLD.5 Formal high-level design
	ADV_IMP.3 Structured implementation of the TSF
	ADV_INT.3 Minimisation of complexity
	ADV_LLD.2 Semiformal low-level design
	ADV_RCR.3 Formal correspondence demonstration
	ADV_SPM.3 Formal TOE security policy model
Class AGD: Guidance documents	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Class ALC: Life cycle support	ALC_DVS.2 Sufficiency of security measures
	ALC_LCD.3 Measurable life-cycle model
	ALC_TAT.3 Compliance with implementation standards - all parts
Class ATE: Tests	ATE_COV.3 Rigorous analysis of coverage
	ATE_DPT.3 Testing: implementation representation
	ATE_FUN.2 Ordered functional testing
	ATE_IND.3 Independent testing - complete
Class AVA: Vulnerability assessment	AVA_CCA.2 Systematic covert channel analysis
	AVA_MSU.3 Analysis and testing for insecure states
	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.4 Highly resistant

EXEMPLE:
PROFIL DE PROTECTION
D ’UNE ICP (Trustycom)

FONCTIONNEMENT DES INFRASTRUCTURES À CLEFS PUBLIQUES(PKI)



COMPOSANTS (1)

- **Les autorités d'enregistrement** (notées AE) collectent les informations nécessaires à l'identification des utilisateurs et vérifient leurs droits pour les communiquer aux AC.
- **Les autorités de certification** (notées AC) ont pour principale fonction de signer les certificats et de communiquer au centre de publication les certificats de clés publiques. Les AC assurent la révocation des certificats qui ne sont plus de confiance, ainsi que la mise en publication de la liste des certificats révoqués (CRL).
- **Le Centre de Publication** (noté CP) assure, sur demande des AC, la publication des certificats et des listes de révocation au sein de l'IGC et pour les applications utilisatrices.

COMPOSANTS (2)

- **Le Centre de Génération de Clés** (noté CGC) assure, sur demande de l'AC, la génération de bi-clés. Il transmet alors la clé publique à l'AC pour certification
- **Une Base de Données** (noté BD) qui contient tout les biens sensibles de la TOE : les clés privées, les évènements d'audit, les codes PIN et les codes de déblocage des exploitants, les attributs de sécurité des exploitants et des rôles. Tous ces biens sont chiffrés à l'aide de clés symétriques, de plus, certains d'entre eux sont protégés en intégrité.

REFERENCE A DES PROFILS DE PROTECTION

- PP_AC : Profil de protection pour une autorité de certification dans le cadre d’une infrastructure de gestion de clés.
- PP_AE : Profil de protection pour une autorité d'enregistrement pour une infrastructure de gestion de clés.
- PP_IGC : Profil de protection pour une infrastructure de gestion de clés.
- PP_RCIGC Profil de protection pour les ressources cryptographiques d'une infrastructure de gestion de clés.

HYPOTHÈSES

Dans cette cible d'évaluation, il est considéré :

- - qu'aucune hypothèse n'est définie sur la nature des relations entre AC (graphe et hiérarchie);
- - qu'aucune hypothèse n'est définie sur le caractère générique des AC (AC dédiée à une ou plusieurs applications);
- - qu'aucune hypothèse n'est définie sur les supports de communication entre l'AC et les composantes de l'IGC (réseaux ouverts ou fermés).
- Par définition, un grand nombre de mesures de protection mises en œuvre au sein d'une IGC s'appuie sur des mécanismes cryptographiques.

Le fonctionnement de l'IGC est régi par une politique de sécurité.

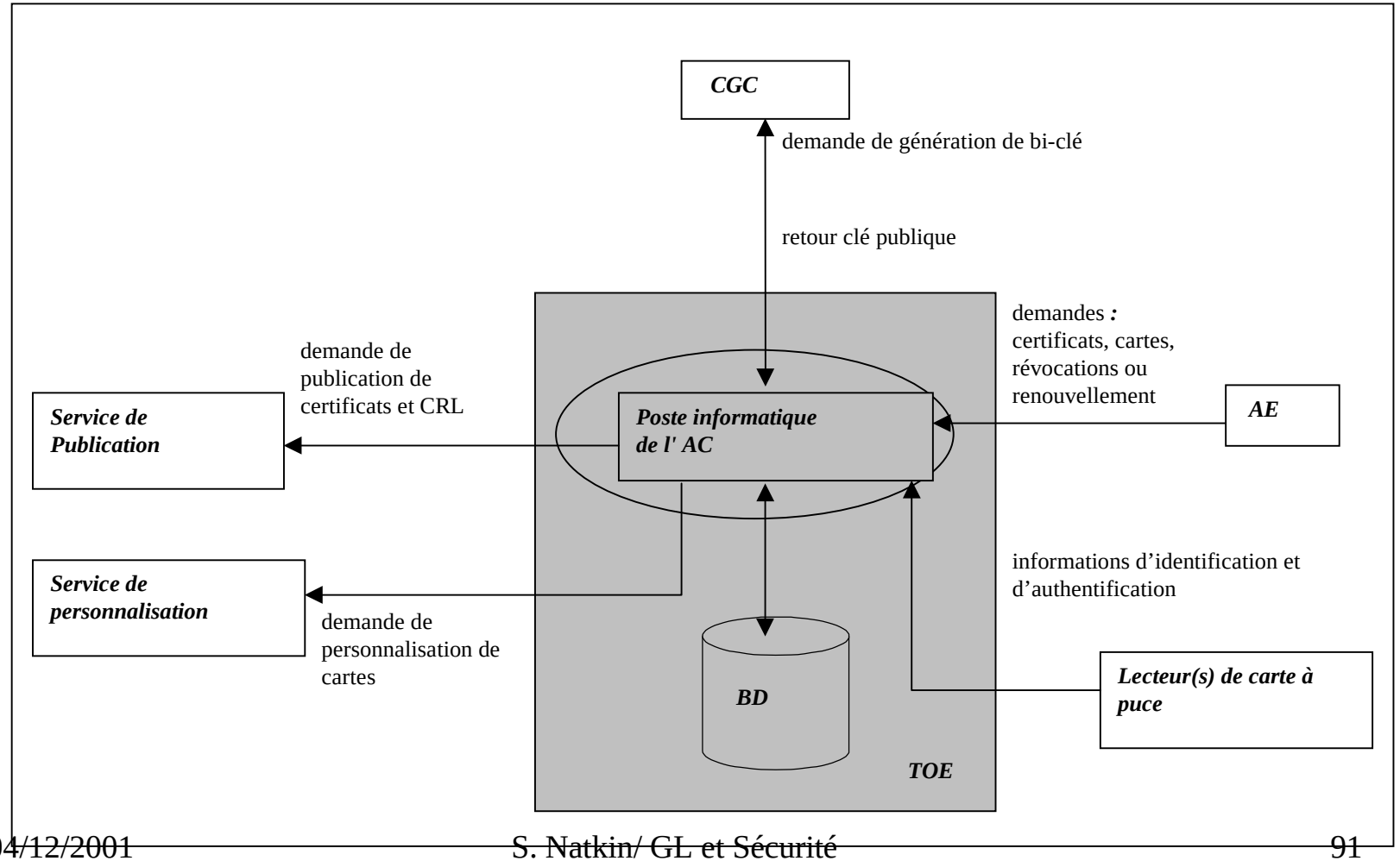
SERVICES VITAUX

- S_DATAREVOC : recevoir et authentifier les demandes de révocation transmises par l'AE,
- S_REVOC : générer les listes de certificats révoqués (notés CRL),
- S_TRANSREVOC : transmettre les demandes de publication de CRL au Centre de Publication,
- S_AUDIT : enregistrer des traces sur toutes les opérations effectuées en s’assurant de leur imputabilité.

SERVICES ÉLÉMENTAIRES

- S_ADMIN : gérer les profils et les droits des exploitants, configurer le poste AC et superviser les actions réalisées (S_AUDIT),
- S_AUTHEN : identifier et authentifier des exploitants,
- S_DATACERT : recevoir et authentifier les demandes de certificats et de renouvellement transmises par l'AE ; ces demandes peuvent être accompagnées d'une demande de génération de bi-clé, à transmettre au CGC,
- S_CERTIF : générer les certificats et renouveler des certificats demandés par l'AE,
- S_TRANSCERT : transmettre les demandes de publication de certificats au Centre de Publication,

ENVIRONNEMENT



ENVIRONNEMENT SUITE

- Les systèmes d'exploitation sont Windows NT 4.0 et Windows 2000.
- Le logiciel TrustyKey nécessite la présence d'une base de données. Les bases de données supportées sont Oracle et SQL-Serveur.
- TrustyKey fonctionne avec un contrôle direct par carte à puce.
- La TOE est hébergée sur une plate-forme informatique (matériel , logiciel, réseaux) s'appuyant sur les ressources cryptographiques du CGC.
- La TOE couvre le développement, l'initialisation et l'exploitation de l'AC ainsi que les états transitoires (livraison, installation).
- Les ordinateurs sont reliés entre eux par un réseau Ethernet 100 Base T, sous TCP/ IP.
- Les réseaux locaux dans une même pièce, sans connexion vers l'extérieur sont considérés comme sécurisés.
- Le personnel autorisé à utiliser le système est " habilité " par l'organisme utilisant TrustyKey. Tous les locaux entreposant le système ou une partie du système sont sous haute surveillance (contrôle d'accès, gardiennage …).

RÔLES

Utilisateurs

Rôles d ’administration

- Ingénieur système
- Master-administrateur

Rôles de service

- Officier de certification
- Opérateur de certification.
- Officier de gestion
- Superviseur

TYPOLOGIE DES ATTAQUANTS

- un exploitant autorisé mais pouvant commettre une erreur ou un acte de malveillance,
- les personnes ou machines qui disposent de moyens d'investigation et d'action sur les réseaux supports de la TOE,
- les personnes ou machines qui disposent de moyens d'investigation et d'action sur la TOE (personnel de passage sur le site de la TOE, personnel de nettoyage, gardien, etc.).

BIENS A PROTEGER

Les demandes émises par l ’AC

B.DEM_CERTIF, B.DEM_REVOC, B.DEM_BICLE,…

Les demandes reçues par l ’AC

B.DEM_PUB_CERTIF, B.DEM_PUB_REVOC, B.DEM_GEN_BICLE

les données internes à l'AC

B.TRACES_AUDIT , B.CONFIG

les services assurés par l'AC

B.SERVICES

MENACES

Issues des EBIOS

- M.CONFIDENTIALITE
- M.INTERGRITE
- M.SINISTRE
- M.MASCARADE
- M.VOL
- ….
- **M.REPUDIATION :** un utilisateur ou un exploitant nie être l'auteur d'une action.

Reniement d’actions (EBIOS 41)

Cette menace concerne *les actions déclenchées par l'AC, donc les biens suivants :*

B.DEM_PUB_CERTIF, *B.DEM_PUB_REVOC, B.DEM_GEN_BICLE*.

AUTRES HYPOTHESES

- Politiques d ’organisation (références)
- Hypothèses d ’utilisation

OBJECTIFS DE LA TOE (EXEMPLES)

- **OT.IDENT** : la TOE doit être capable d'identifier de façon unique et d'authentifier les utilisateurs avant d'autoriser tout accès aux biens protégés par la TOE.
- **OT.NOBUG** : la TOE ne doit pas présenter de défauts majeurs de développement qui remettraient en cause sa capacité à remplir sa mission : cela concerne en particulier la qualité de développement des composants logiciels ou matériels de la TOE.
- **OT.TRACE** : toute opération réalisée par un exploitant de la TOE doit être tracée, et imputable à son auteur. L'intégrité des traces doit être garantie par la TOE.
- **OT.PROTECT_DATA** : la TOE doit limiter l'accès aux biens confidentiels qu'elle manipule aux personnes ayant le besoin d'en connaître, conformément aux rôles identifiés. En particulier, la TOE doit contrôler l'accès aux biens suivants : B.TRACES_AUDIT, B.CONFIG.

EXIGENCES FONCTIONNELLES (EXEMPLE)

- **FAU_GEN.1** Génération de données d'audit
- **FAU_GEN.2** Lien avec l'identité de l'utilisateur
- **FAU_SAR.1** Revue d'audit
- **FAU_SAR.2** Revue d'audit restreinte
- **FAU_SAR.3** Revue d'audit sélective
- **FAU_STG.2** Garanties de disponibilité des données d'audit
- **FAU_STG.4** Prévention des pertes de données d'audit

EXEMPLE DE SPECIFICATION

FCO_NRO.2.1 La TSF doit mettre en œuvre la génération de la preuve de l'origine à tout moment pour [**" les demandes émises par l'AC "**] transmises.

Raffinement : "*les* demandes *émises par l'AC*" pour lesquelles une preuve d'origine sera générée sont les demandes de publications B.DEM_PUB_CERTIF, B.DEM_PUB_REVOC et les demandes de génération de bi-clés B.DEM_GEN_BICLE.

Raffinement *: Dans POLITIQUE_SECU, ces demandes correspondent aux demandes de mise en publication de certificats, de CRL, ou d'ARL envoyées au Centre de Publication (DEM_PUB_CERT, DEM_PUB_AUT, DEM_PUB_CRL, DEM_PUB_ARL),et les demandes de génération de bi-clé à transmettre au Centre de Génération de Clés (DEM_GEN)*

NIVEAU D ’ASSURANCE

EAL 3 avec modèle informel de la politique de sécurité

EXEMPLE: GESTION DE CONFIGURATION

ACM_SCP.3.1C The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, **and development tools and related information.**

ACM_SCP.3.2C The CM documentation shall describe how configuration items are tracked by the CM system.

Evaluator action elements:

ACM_SCP.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

CONCEPTION FORMELLE

ADV_HLD.5 Formal high-level design

Dependencies:

ADV_FSP.4 Formal functional specification

ADV_RCR.3 Formal correspondence demonstration

Developer action elements:

ADV_HLD.5.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.5.1C The presentation of the high-level design shall be **formal.**

ADV_HLD.5.2C The high-level design shall be internally consistent.

ADV_HLD.5.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.5.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

CONCEPTION DESCRIPTIVE

ADV_HLD.1 Descriptive high-level design

Dependencies:

ADV_FSP.1 Informal functional specification

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_HLD.1.1D **The developer shall provide the high-level design of the TSF.**

Content and presentation of evidence elements:

ADV_HLD.1.1C **The presentation of the high-level design shall be informal.**

ADV_HLD.1.2C **The high-level design shall be internally consistent.**

ADV_HLD.1.3C **The high-level design shall describe the structure of the TSF in terms of subsystems.**

ADV_HLD.1.4C **The high-level design shall describe the security functionality provided by each subsystem of the TSF.**

TRACABILITE

ADV_RCR.3 Formal correspondence demonstration

Application notes

364 The developer must either demonstrate or prove correspondence, as described in the requirements below, commensurate with the level of rigour of presentation style. For example, correspondence must be proven when corresponding representations are formally specified.

Dependencies:

No dependencies.

Developer action elements:

ADV_RCR.3.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.3.2D **For those corresponding portions of representations that are formally specified, the developer shall prove that correspondence.**

Content and presentation of evidence elements:

ADV_RCR.3.1C For each adjacent pair of provided TSF representations, the analysis shall **prove or** demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

"PREUVE INFORMELLE »:
COUVERVURE DES MENACES PAR LES OBJECTIFS

M.MASCARADE	**OT.IDENT** **OT.TRACE** **OT.ADMINIS** **OE.SENSIB_EXPL** **OE.ACCES**	- L'identification et l'authentification des exploitants rendent difficile cet abus, le contrôle d'accès physique à la TOE venant renforcer ces objectifs. - La sensibilisation des exploitants, à la nécessité de protéger leurs moyens d'authentification notamment, rend moins probable la possibilité de réussir une telle attaque. - La séparation des tâches et la nécessaire collaboration entre exploitants, ainsi que l'administration correcte de ces rôles renforce le contrôle de légitimité des exploitants. - La traçabilité de toutes les opérations, et l'exploitation de ces traces aide à détecter une éventuelle mascarade.

6. Méthodes formelles et sécurité

OBJECTIFS

- Spécification des politiques
- Démonstration en conformité
- Spécification de la conception
- Pas d ’utilisation en efficacité (génération de codes)

METHODES ET OUTILS

- Méthodes de validation par inférences : Logique modale d ’apprentissage des connaissances (BAN)
- Méthodes de preuves par construction: (Utilisation de Coq par D. Bolignano)
- Méthodes d ’états transitions pour la modélisation des attaques (Murphi)

CARACTERISTIQUES

- Propriétés négatives a prouver
- Modélisation des attaques

FORMALISATION
DES POLITIQUES DE SÉCURITÉ

MATRICE DES DROITS

Définit à chaque instant les droits de chaque utilisateur sur chaque objet

	Dossier P 1	**Dossier P 2**	**TM**	**TP**	**TPM**	**ME**
US	**cr, lec, mod,dt**	**cr, lec, mod,dt**				**cr, em, lec**
MED 1	**lec**					**cr, em, lec**
MED 2	**lec**					**cr, em, lec**
A			**cr, mod**	**cr, mod**	**cr**	
PAT 1					**cr,dt**	
PAT 2					**cr,dt**	

cr: créer, lec : lire, mod: modifier, dt: détruire, em: émettre

EVOLUTION DE LA MATRICE DES DROITS

La matrice des droits évolue en fonction des évènements suivants:

- évolution de la population des utilisateurs
- création et destruction des objets
- création et destruction des droits
- propagation des droits

MODELE DE BELL LAPDULA (1)

Le niveau d'habilitation
H={non classifié,privé, confidentiel, secret} totalement ordonné
non classifié>privé>confidentiel> secret
Le domaine DOM={domaine, nucléaire, nucl. civil, nucl. mil , cryptographie, missile ....}
qui est un thésaurus ordonné partiellement par des relations d'inclusions
Par exemple nucl. civil est plus restreint que nucléaire
Un couple (h,d) est dominé par une couple (h',d')
si h<= h' et d est inclus au sens large par d ’.
On dit que (h,d) est plus confidentiel
que (h’,d ’) .Ceci définit une structure de treillis sur HxDOM, que nous notons >=

A tout sujet est associé un ensemble de couples (h,d)
qui pour chaque domaine d définit le niveau d'habilitation du sujet

A tout droit sur un objet (exécution d'une opération) est associé un couple (h',d')

Les politiques de sécurité reposent sur des préconditions
pour réaliser une opération jouant sur les couples (h,d) et (h',d')

MODELE DE BELL LAPDULA (2)

Cette politique a pour objectif contrôler la confidentialité des données et d'éviter la propagation d'information d'un domaine de sécurité supérieur vers un domaine inférieur.

Notons D(o,j) le couple (h,d) associé au droit j (Lecture, Lecture+Ecriture) sur l'objet o et A(s) l'ensemble des couples associés au sujet s

Un système de sécurité vérifie la politique de Bell Lapadula si à tout instant la matrice de contrôle des droits M est telle que

- M(s,o)=Lecture implique
il existe (h,d) dans A(s) avec D(o,Lecture)>=(h,d)
Un sujet ne peut lire un document que s'il est habilité dans un domaine plus restrictif et à un niveau plus restreint que celui du document

- M(s,o)=Lecture et M(s,o')=Lecture+Ecriture implique
D(o',lecture)>=D(o,lecture)

Pour tout couple de document o et o' tel qu'un sujet a juste le droit de lire o et le droit de lire et écrire o', alors o' est plus confidentiel que o (dans un domaine donné)

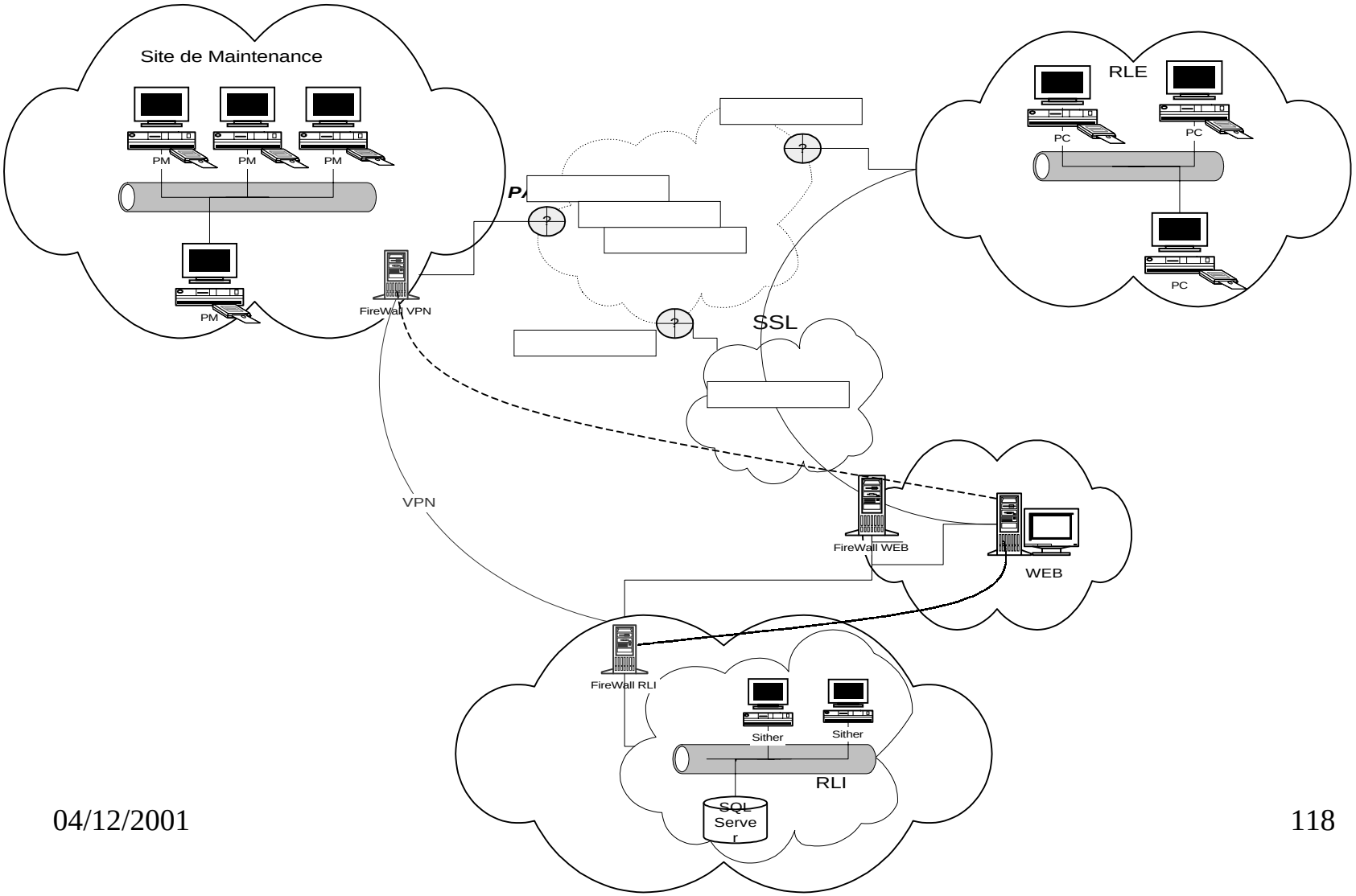
MODELE DE BELL LAPDULA (3)

Cette politique est incomplètement spécifiée: Elle ne détermine pas le mode d'évolution des droits intiaux sur les objets introduit dans le système (sans recopie)

Elle amène, si la granularité des objets est insuffisante, à surclassifier tous les objets

EXEMPLE DE DEMONSTRATION SEMI FORMELLE

ARCHITECTURE



MATRICE DES DROITS

	RÔLES												
ACTIONS	ADM_SYS_LOC	OPE	ADM_STH	OPE_CHI	OPE_CDT	ADM_BD_LOC	MNT_STH	ADM_SEC	ADM_BD	CONS_WEB	ADM_SYS_MNT	WEB	
Créer un compte et donner les droits ADM_SYS_LOC	O												
Créer un compte et donner les droits ADM_STH	O												
Créer un compte et donner les droits ADM_BD_LOC	O												
Créer un compte et donner des droits associés à un rôle TRUC			O										
Donner un droit CONF_FW	O												
Donner un droit AUD_SEC	O												
Donner un droit AUD_WEB	O												
Utiliser TRUC en mode opérateur (O)	O	O	O	O	O		O						
Utiliser TRUC en mode opérateur chimie (CHM)				O									
Utiliser TRUC en mode commande (T)					O								
Administrer la configuration TRUC (A)			O										
Administrer la BD local RLI (B)						O							
Valider une nouvelle version de TRUC			O										
Configuration des dispositifs de sécurité (CONF_FW)								O					

PROPRIÉTÉS RACINES

CI_RLI : Intégrité du RLI

Les seules opérations portant sur une ressource du RLI sont celles autorisées par la politique de sécurité.

Démonstration

Cette propriété est couverte par les trois propriétés suivantes:

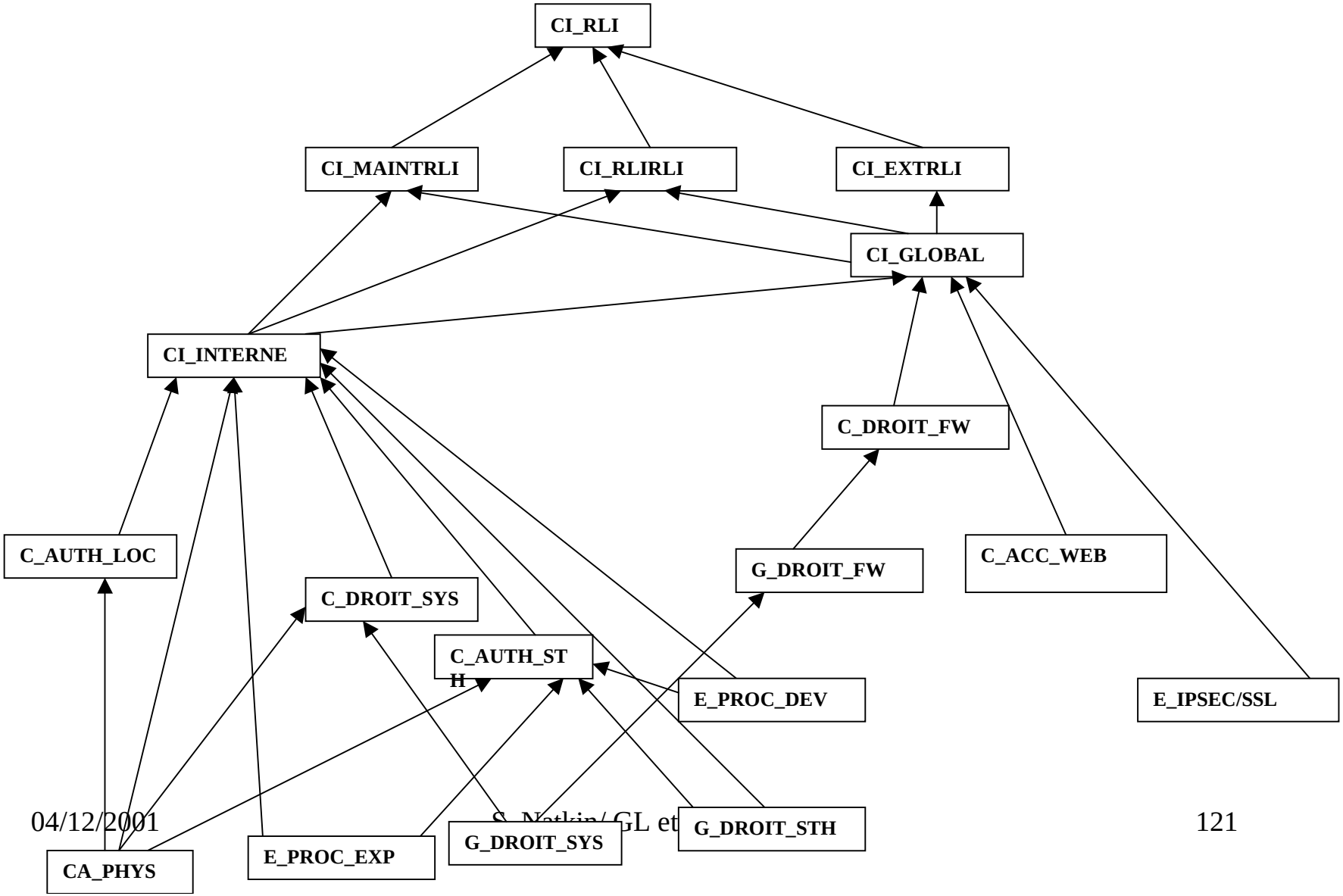
: *Les seules opérations portant sur une ressource du RLI réalisée à partir du site de maintenance sont celles autorisées par la politique de sécurité.*

CI_RLIRLI : *Les seules opérations portant sur une ressource du RLI réalisée à partir du RLI sont celles autorisées par la politique de sécurité.*

CI_EXTRLI : *Les seules opérations portant sur une ressource du RLI sont réalisées à partir du RLI ou du site de maintenance.*

Fin démonstration

ARBRE DE DÉMONSTRATION



PROPRIÉTÉS FEUILLES

E_PROC_DEV Efficacité des procédures de développement

Les procédures de développement des logiciels en site de maintenance interdisent toute action sur le procédé. Ces procédures sont telles qu'il est pratiquement impossible de violer cette propriété accidentellement.

Démonstration

Application du PAQL [], du principe de gestion de configuration [] et des procédures d'audit []

Fin démonstration

1.1.1 E_PROC_WEB Efficacité des procédures définissant l'accès au Web

les personnes ayant un rôle CONS_WEB ne réalisent pas volontairement d'opération non conforme à la politique de sécurité.

Démonstration

Il s'agit d'un principe de base qui ne peut être obtenu que par une formation des personnels à la politique de sécurité. Cette formation doit montrer les risques résultant d'un traitement laxiste de la politique de droits.

Vérification du contenu de formation des personnels, vérification périodique de l'existence et du suivi de formations (procédures à spécifier)

Fin démonstration

VALIDATION DE SSL 3. 0
AVEC MURPHI

Murphi

- Logiciel de validation état transition développé à Stanford basé sur CSP
- Validation matérielle
- Protocole
- Protocoles de sécurité

ETAT

- Ensemble de variable typées
- Un processus a un état local
- Cet état change quand il reçoit, capte, construit des messages par assemblage de variables typées

EXEMPLE

Cl->Se Cl, VerCl, Suite Cl

Se->Cl Se, VerSe, Suite Ce, s

Cl->Se $\{scs\}_s^{ASY}$

SSL/TLS

Service

- SSLV3.0 et TLS offrent des connexions asymétrique et possédant toutes les fonctionnalités des connexions TCP. Elles assurent en outre :

- Une authentification forte d'un ou des deux parties en utilisant un système de certification basé sur le RSA, Diffie Hellman ou le DSA. Le protocole ne précise rien sur la gestion des certificats proprement dite.

- Une protection contre les attaques d'interception: une fois la connexion établie l'échange est garanti se dérouler entre les parties authentifiées.

- Une protection en intégrité des messages et du flux de messages, par utilisation conjointe d'un numérotation des messages et une fonction de hachage sécurisée (basée sur le MD5 ou SHA1).

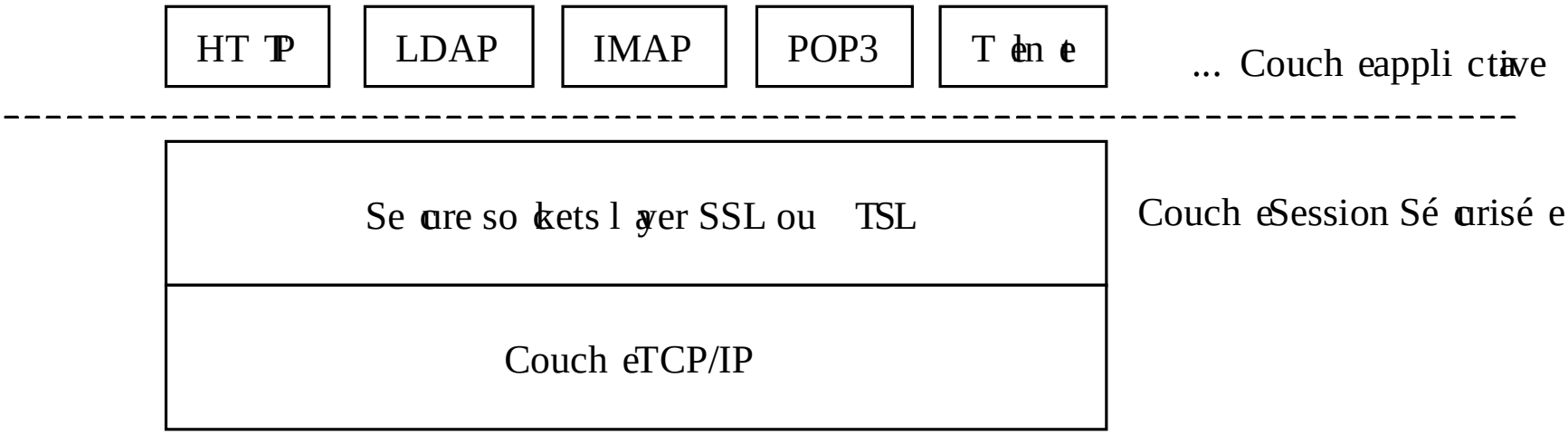
Service (2)

- Optionnellement la compression des données en utilisant tout algorithme de compression implanté de part et d'autre.

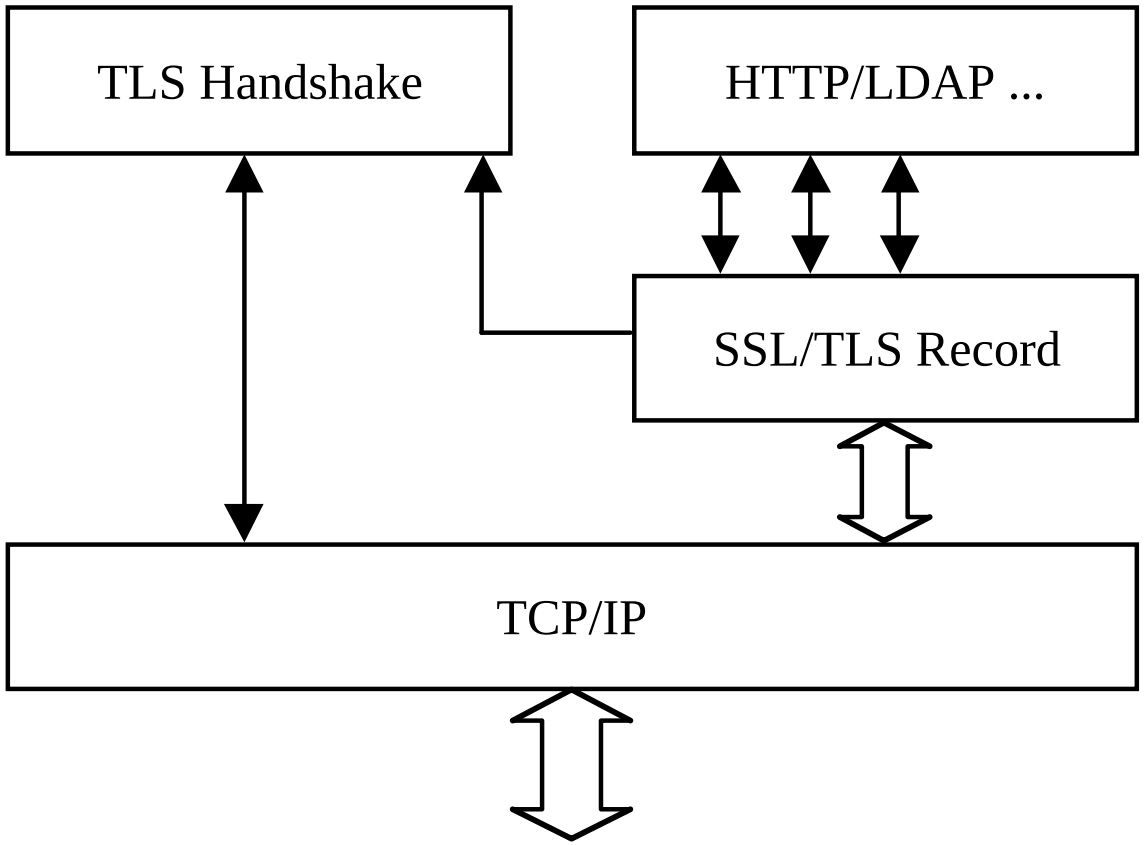
- Optionnellement une protection en confidentialité des données en utilisant tout algorithme cryptographique symétrique implanté de part et d'autre et une clef de session unique par connexion. La plus part des implantations supportent le DES, le 3DES (à clefs de 112 et 168 bits), le RC5.

- La suite cryptographique utilisée est négociée à l'ouverture de connexion.

Positionnement architectural



Architecture



TLS Handshake

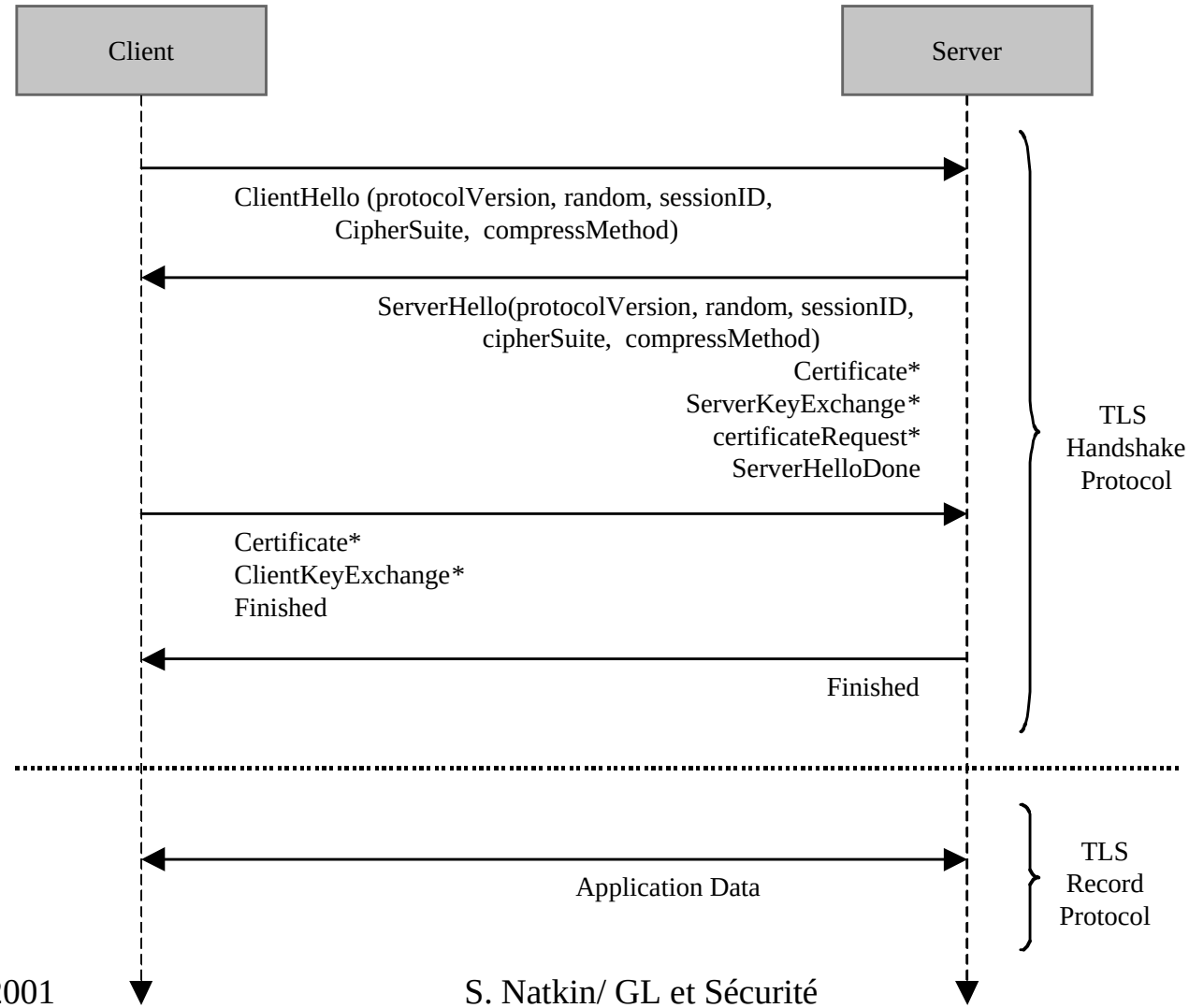
Le protocole TLS Handshake consiste en une suite de trois sous protocoles qui sont utilisées par les entités communicantes pour s'authentifier entre elles, créer un contexte de sécurité négocié utilisé ensuite par TLS Record et gérer et signaler des conditions d'erreur:

- Négociation
- Changement des paramètres de chiffrement
- Alerte

Contexte de session

- Un identifiant de session.
- Un certificat de l’entité distante. éventuellement nul.
- Une méthode de compression.
- Les spécifications du chiffrement. algorithme de chiffrement symétrique (nul, DES, etc.) algorithme de hachage (comme MD5 et SHA-1).
- La clé maître. Un secret de 48 octets partagé entre le client et le serveur.
- Un indicateur indiquant si la session peut couvrir plusieurs connexions TCP.

Exemple de MSC



MODELE MURPHI

- Construit incrémentalement
- 2 Clients, un attaquant, un serveur

PROTOCOLE Z

Cl->Se	Cl, VerCl, Suite Cl,NoCl
Se->Cl	Se, VerSe, Suite Ce, NoCe,Cert(Se)
Cl->Se	Cert(Cl),$\{scs\}_s^{ASY}$ $\{Message\}_c^{SIG}$
Se->Cl	Message, $\{\{Message\}^H\}_{scs}^{SYM}$

MODELE DE L ’ATTAQUANT

- Pas d ’attaque cryptographique
- Peut et détruire lire tous les messages
- Peut insérer des messages construit à partir de son état local et cosntruire ses propres variables selon les types du protocole
- Peut insérer des messages

PROPRIETES A PROUVER

- L ’attaquant n ’acquière pas de données secrètes
- Le savoir partagé entre partie de confiance est cohérent en particulier l ’authentification est correcte
- Le protocole se termine par une fermeture négociée
- La suite cryptographique utilisée est la meilleure possible
- Les versions du protocole utilisées sont les meilleures possibles

RESULTAT

- Montre la possibilité de trouver des faiblesses (processus incrémental)
- Modèle final 200000 états correct
- Correct
- Sous les hypothèses de la modélisation

CONCLUSION

Pas une démonstration au sens mathématique du terme:

Une méthode pour (se) convaincre:

- Que tout ce qui était nécessaire a été fait
- Que tout ce qui a été fait était nécessaire

Pour aboutir aux objectifs spécifiés

Bibliographie

- A Logic of Authentication by Burrows, Abadi and Needham, Timo Kyntaja, http://www.tml.hut.fi/Opinnot/Tik-110.501/1995/ban.html
- Can Finite-State System Verification Methods Help Cryptographic Protocol Analysis? Keijo Heljanko, http://www.tml.hut.fi/Opinnot/Tik110.501/1998/papers/13finitestate/finitestate.htm
- Finite-state analysis of SSL 3.0 (Usenix '98) et autre papier sur Murphi http://theory.Stanford.EDU/people/jcm/
- Formals Methods in Practice the missing links, D Bolignano, De Le Métayer, C Loiseau, www.trustedlogic.com/
- Common Criteria For Information Technoly Security EvaluationV2.0, http://csrc.nist.gov/cc/ccv20/ccv2list.htm
- Security Protocols over Open Networks: Formal Methods for their Analysis, Design and Verification, S. Gritzalies, D. Spinellis, P. Georgiadis, Computer Communication, Vol 22, N°8, Mai 1999.
- Profil de Protection, Politique de sécurité pour Trustykey. Documentation Trustycom, Novembre 2000
- Les protocoles de sécurité de l ’Internet, S. Natkin, Dunod, à paraître 2002