

# Certification et Développement Sécurisé

Damien Sauveron  
[damien.sauveron@unilim.fr](mailto:damien.sauveron@unilim.fr)  
<http://damien.sauveron.fr/>

(et tous les auteurs cités en bibliographie)

1

## Intervenants dans le module

### Damien Sauveron (Cours, TD et TP) :

- Membre de l'équipe « Sécurité de l'information »
- Thème de recherche : sécurité des cartes à puce et des réseaux ad hoc

### Pour me contacter :



Damien Sauveron  
XLIM UMR 6172 CNRS -- Université de Limoges  
123 avenue Albert Thomas  
87060 Limoges Cedex, FRANCE

Email: [Damien.Sauveron@unilim.fr](mailto:Damien.Sauveron@unilim.fr)  
Web: <http://damien.sauveron.fr/>  
Phone: +33 (0) 5 87 50 67 93

### 2004-2006 :

Visiteur/Postdoctorant dans le Smart Card Centre de l'Information Security Group du Royal Holloway, University of London (6 mois)

ATER à l'université de Limoges

### 2001-2004 :

Thèse au LaBRI – Université Bordeaux 1 sur la *Sécurité de la Technologie Java Card*

Ingénieur R&D dans le CESTI de SERMA Technologies (Pessac) spécialisé sur *les évaluations sécuritaires puces*

2

## Plan

### La sécurité ?

- Quelques définitions
- Quoi protéger ?
- Comment ?
- Vérifier la sécurité ?
- UML
- Méthodes formelles & Test
- Évaluation sécuritaire et certification
- Open source vs propriétaire
- Outils d'analyse logiciel
- Analyse de sécurité
- Procédures d'attaques

### Les sécurités

- Au niveau physique
- Au niveau logicielle
- Au niveau de l'environnement de production

3

**Qu'est ce que la sécurité ?**

4

# Qu'est ce que la sécurité ?

- Comme vous venez de le voir définir la sécurité n'est pas aussi simple qu'on le pense
- L'**approche la plus courante** concernant la sécurité informatique se résume sous le terme **CIA**  
**Confidentialité** (Confidentiality)  
**Intégrité** (Integrity)  
**Disponibilité** (Availability)
- Pour d'autres types de sécurité (des personnes par exemple), nous devons la définir en relation avec les risques

## Confidentialité

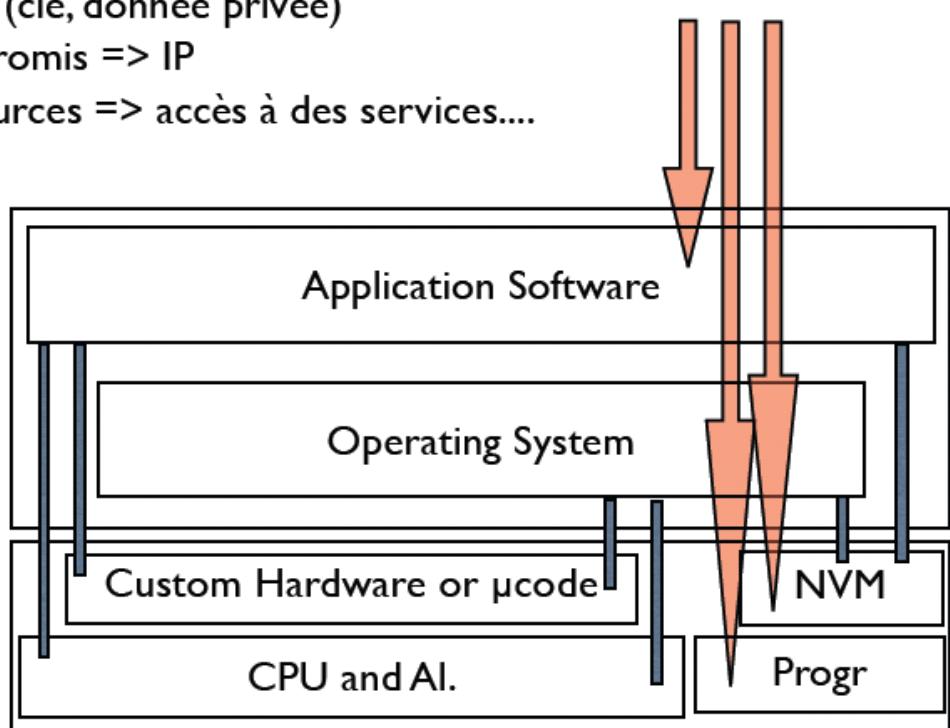
- La confidentialité consiste à empêcher **des utilisateurs non autorisés de lire** des informations pour lesquelles ils n'ont pas le droit.
- C'est en quelque sorte la propriété de « secret » attachée aux informations
- Traditionnellement, pour la plupart des gens la sécurité est assimilé à la confidentialité

# Confidentialité

- Comment peut on assurer cette propriété ?  
par chiffrement (mécanisme cryptographique) pour assurer la confidentialité lors de transmission d'information, i.e. pour les communications, mais aussi pour le stockage  
par l'utilisation de contrôle d'accès
  - Pour un SEM, on peut utiliser des contrôles d'accès aux informations ou aux ressources (problème de granularité des droits d'accès)
    - Contrôle d'accès au SEM
    - Contrôle d'accès aux ressources internes au SEM
    - Contrôle d'accès logique à l'intérieur du SEM
- Attention : Il existe une propriété proche de la confidentialité qui est l'anonymat. Elle consiste à assurer la confidentialité d'une donnée bien particulière : l'identité (on peut la voir comme un sous ensemble)

## La confidentialité

- Data compromise (clé, donnée privée)
- Programme compromis => IP
- Accès à des ressources => accès à des services....



# Intégrité

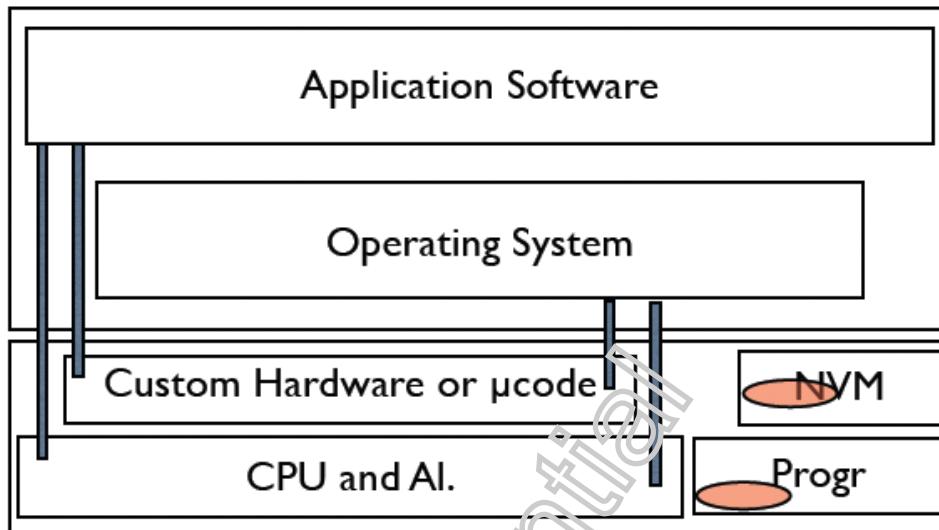
- L'intégrité consiste à s'assurer que les choses seront faites comme elles devraient l'être
- Dans le *contexte de l'informatique*, l'intégrité consiste à empêcher des utilisateurs **non autorisés d'écrire** des informations pour lesquelles ils n'ont pas le droit.
- Dans un *contexte général*, l'intégrité consiste à s'assurer que l'état d'un système n'a pas été modifié par des personnes qui n'ont pas le droit de le faire.

# Intégrité

- Dans le contexte *des communications de données* l'intégrité se restreint souvent à détecter les modifications de ces informations.  
Une propriété proche consiste à identifier de qui l'information provient.  
C'est l'**authentification** des données (peut être vu comme une généralisation de l'authentification d'entité qui porte sur l'authentification d'une donnée bien particulière : l'identité )
- Comment peut on assurer cette propriété ?  
Par du contrôle d'accès  
Par des mécanismes de MAC ou de signature numérique (cryptographie asymétrique)  
Par des mécanismes de CRC ou Checksum (somme de contrôle) ou de redondance
  - Mécanismes assurés par du matériel ou du logiciel

# L'intégrité

- Données et programmes changés



11

## Intégrité

- Informations concernées par l'intégrité :
  - Les données
  - Les programmes (i.e. le code)
    - Peut avoir un impact sur les données des applications, sur les données du système lui-même (autorisation de nouveau service, modification de table d'accès, etc.)
    - L'altération peut être volontaire (e.g. une attaque) mais également le fruit d'une erreur de programmation (i.e. bug)
- L'exécution d'un programme ? **Comment garantir cela ?**

# Disponibilité

- La disponibilité est la propriété d'un système d'**être accessible à la demande** par un utilisateur autorisé à y accéder.
- Cette notion va bien au delà de la sécurité et concerne aussi la tolérance aux fautes.
- D'un point de vue sécurité, on s'intéresse principalement à éviter **les attaques par déni** de service d'utilisateurs non autorisés.
  - Par exemple, dans le cas d'un serveur web, la disponibilité sera le fait que les requêtes sont satisfaites et que l'attaque par déni de service sera sans effet.
- On peut associer à la notion de disponibilité, celle de **QoS** (Quality of Service – Qualité de Service). Cette propriété **fonctionnelle** englobe plusieurs paramètres comme par exemple la capacité à répondre dans un délai prévu.

**Quelles autres propriétés ?**

# D'autres propriétés

- Attention, il y a d'autres propriétés de sécurité en plus de CIA.
- **Responsabilité** (Accountability)

En pratique, on ne peut pas empêcher des actions malveillantes de se produire.

=> Les utilisateurs doivent être rendus **responsables** de leurs actions, y compris les mauvais usages du système

- Cela se traduit souvent par la mise en place d'un mécanisme d'authentification puis un traçage et un enregistrement des actions ayant trait à la sécurité dans un log d'audit.

Cette notion permet de faire de l'**imputation** et de la **non répudiation**

# D'autres propriétés

- En lien avec la sécurité :

**Fiabilité** (Reliability)

- Pour les systèmes qui doivent fonctionner correctement dans des conditions adverses, la sécurité est clairement proche de la fiabilité (reliability) et de la sûreté (safety)

**Dépendabilité** (Dependability)

- Terme parfois utilisé pour englober la sûreté et la fiabilité.

- Si les objectifs de la sécurité et de la dépendabilité sont proches, les méthodes utilisées pour évaluer les systèmes sont souvent similaires

# Les Systèmes Embarqués et Mobiles

- Particularités des SEM

Le dispositif est entre toutes les mains

- Porte ouverte aux intrusions, ...
- En opposition à un data center qui est une unité de lieu bien délimité  
Toutefois les grilles de calculs distribués ouvertes connaissent un peu la même problématique
- Dispose souvent d'une connectivité  
Porte d'entrée d'une possible attaque
- Plate-forme parfois ouverte (type Java)  
Facilite le time-to-market et l'écosystème mais aussi les attaques de l'intérieur

- Les menaces sont celles des systèmes informatiques classiques

Mais :

- Le **contexte d'utilisation** de l'objet peut être **spécifique**  
PayTV
- Les **aspects système** peuvent être **spécifiques**  
Attaques par déclenchement des chauffages en forçant la température sur les capteurs de chauffage
- Les **enjeux**  
Carte bancaire = argent

Qu'est ce  
qu'on veut  
sécuriser ?

## RAPPEL

- Sécurité
  - Dépendabilité : Fiabilité & Sureté
  - Confidentialité et Anonymat
  - Intégrité et « authentification »
  - Disponibilité
  - Responsabilité
- => Imputation/Non Répudiation

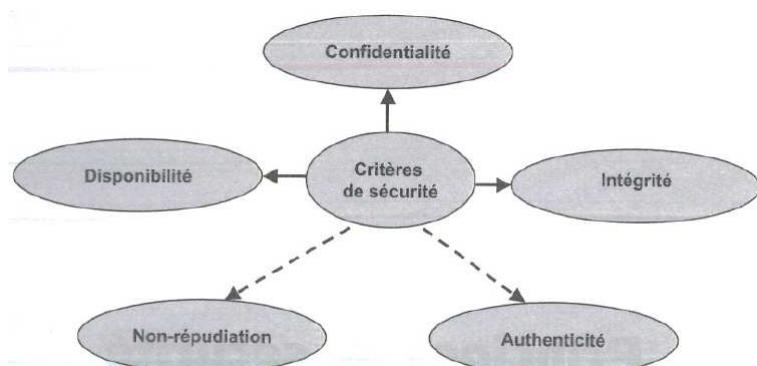
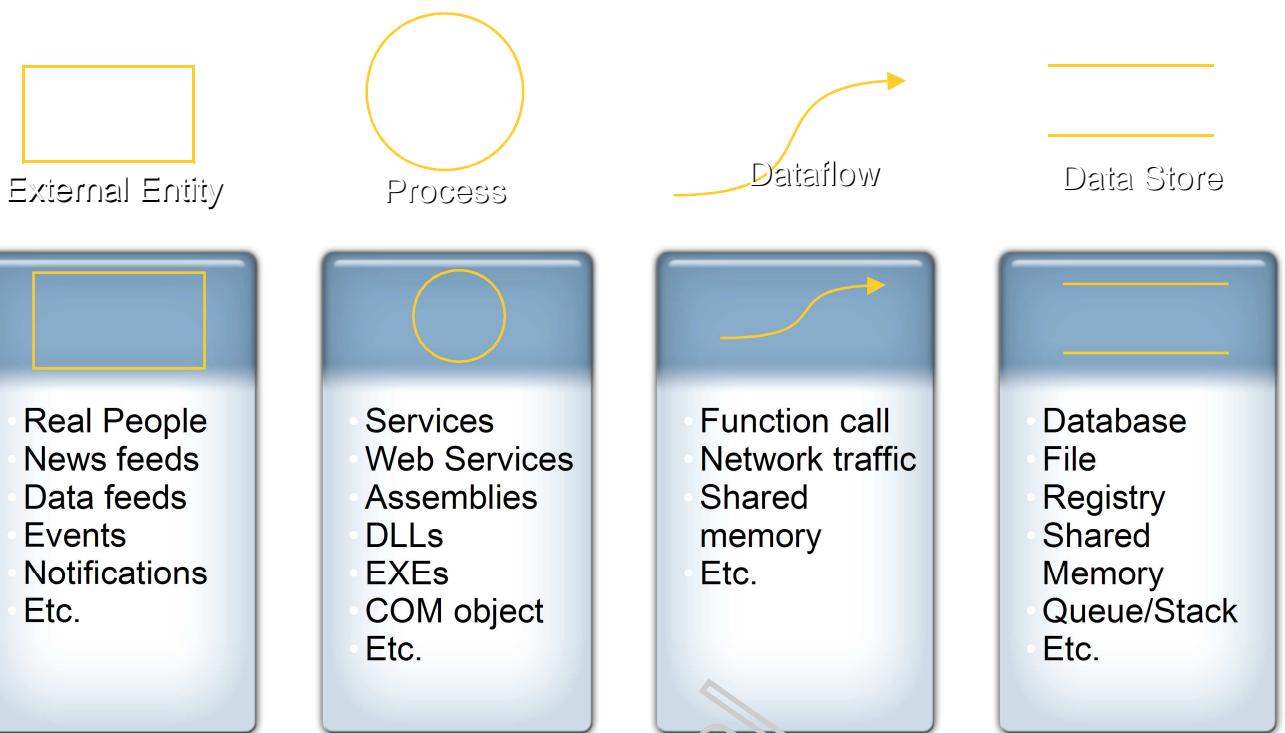


Figure 1.1 – Critères de sécurité.

## Asset (les biens !)



19

## Une autre définition de la sécurité

- Une autre façon de définir la sécurité, en particulier dans le cas de système non informatique, consiste à faire tout d'abord une **analyse des menaces**. La sécurité est alors définie comme la lutte contre les menaces perçues.
- Toutes les menaces ne sont pas toujours toutes intéressantes à contrer.  
Il faut donc faire une analyse coût/bénéfice.
- L'**analyse de risques** consiste à évaluer l'importance de chaque menace pour décider si elle doit ou non être combattue.  
**En effet, il peut être moins coûteux de vivre avec certaines menaces que de les empêcher.**

# Intégration de la sécurité

- De façon générale, on peut se demander comment la sécurité peut être intégrée dans un système ?
- Quels sont les grands types de contrôles de sécurité et où sont-ils intégrés ?
- La sécurité intégrée dans une application peut être contournée par une attaque sur le système d'exploitation !
- La sécurité intégrée dans le système d'exploitation peut quand à elle être contournée par une attaque au niveau du matériel !
- Aussi penser l'intégration de la sécurité dans un système **depuis sa conception** est bien meilleure que l'ajouter après coup.

## Quels grands types de contrôles ?

- Lorsque l'on protège des données dans un système informatique, on utilise souvent des règles sur le comportement du système :  
Ces **règles** peuvent être :
  - Limiter les façons pour manipuler une donnée
  - Limiter les opérations qui peuvent être réalisées sur des données
  - Limiter les utilisateurs qui peuvent réaliser certains types d'actionsLes contrôles de sécurité peuvent cibler les données, les opérations ou les utilisateurs (ou des combinaisons).
- Les règles ne sont pas les seuls types de contrôles de sécurité  
Il ne faut pas oublier la **sécurité physique**.

# Localisation des contrôles

- Un système IT classique peut se modéliser comme 5 couches :  
Applications/Programmes  
Services : e.g. fourni par un SGBD ou un système de fichier distribué  
Le système d'exploitation : gestion des fichiers, des impressions, etc.  
Le noyau de l'OS (Kernel) : intermédiaire pour l'accès au processeur et à la mémoire  
Le matériel : le processeur et la mémoire
- **Les contrôles de sécurité peuvent être localisés dans n'importe laquelle de ces couches.**
- Les mécanismes bas niveaux (près du matériel) seront plus génériques et orienté vers la sécurité du système alors que plus les mécanismes haut niveaux (près des applications) seront plus orientés utilisateur.

## Remarque : Identification/Authentification

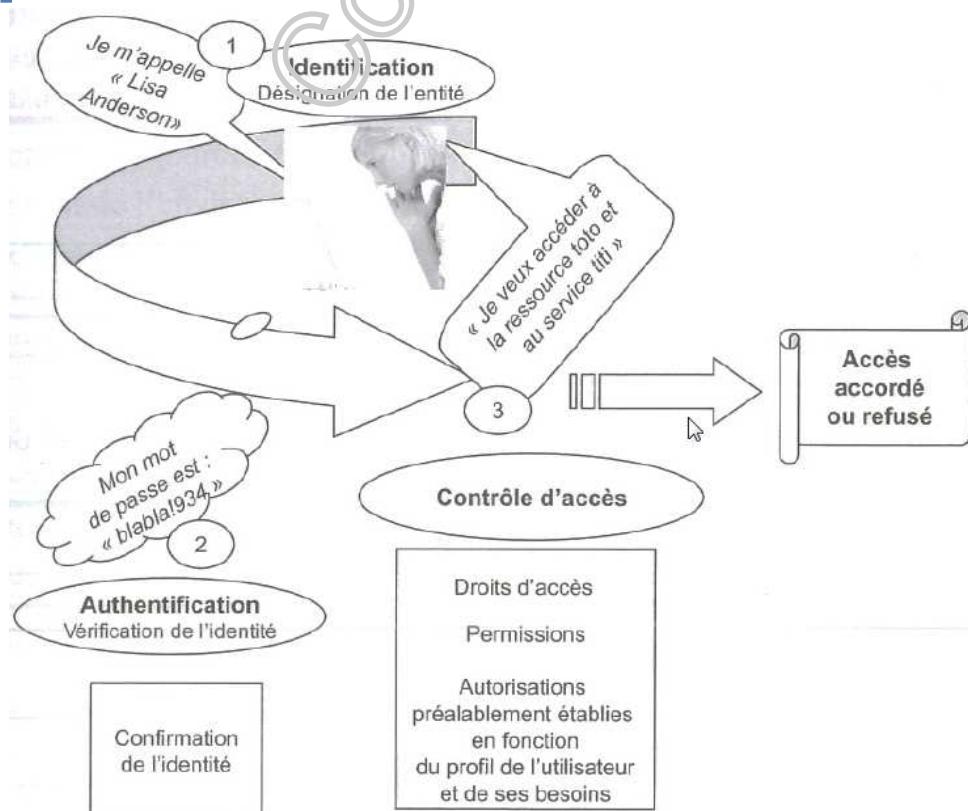
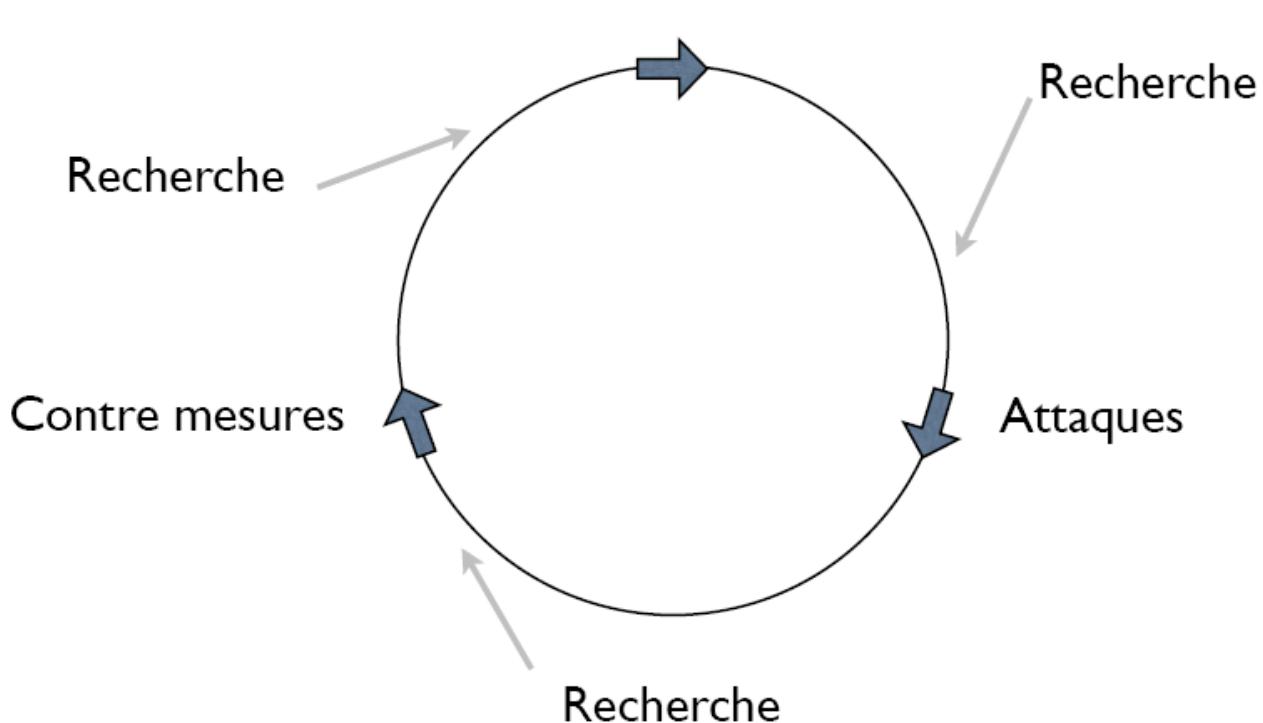


Figure 1.2 – Identification et authentification.

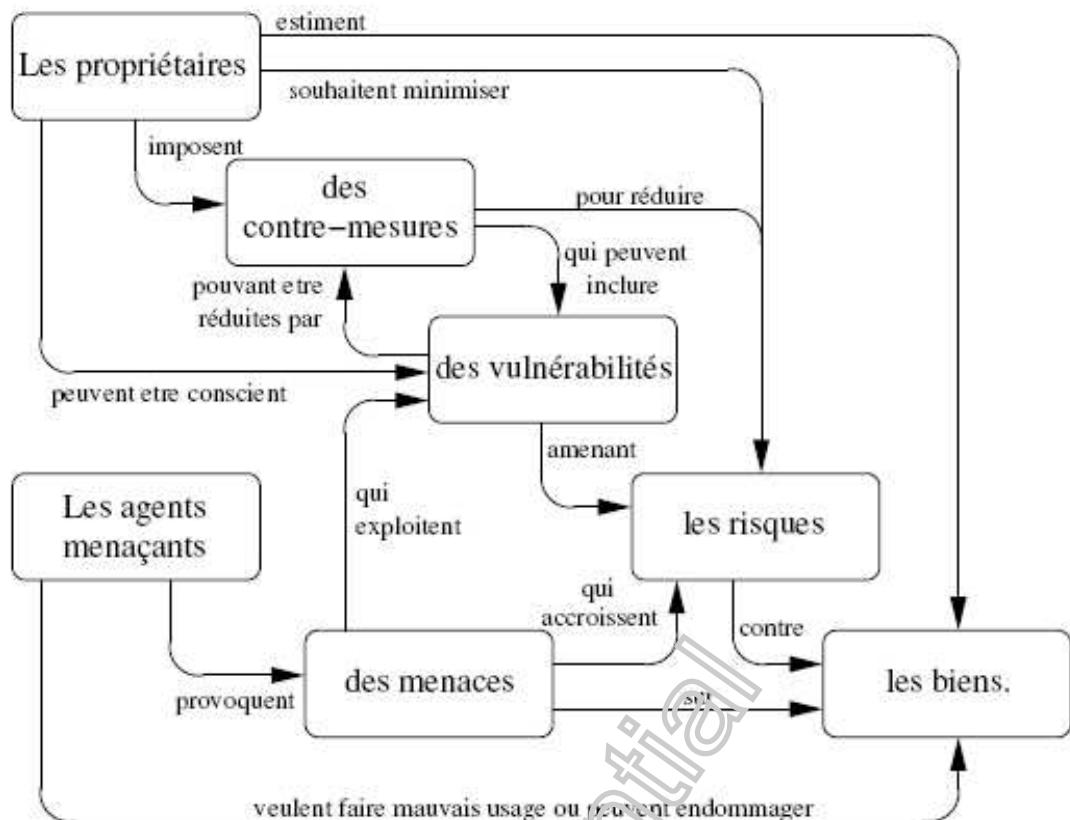
# Quelques remarques

- Pour la sécurité des communications, le principal outil est la cryptographie
- Cette différence technologique est à la base du développement séparé de :
  - Comsec : Communication Security
  - Compusec : Computer Security

Un cycle infernal!

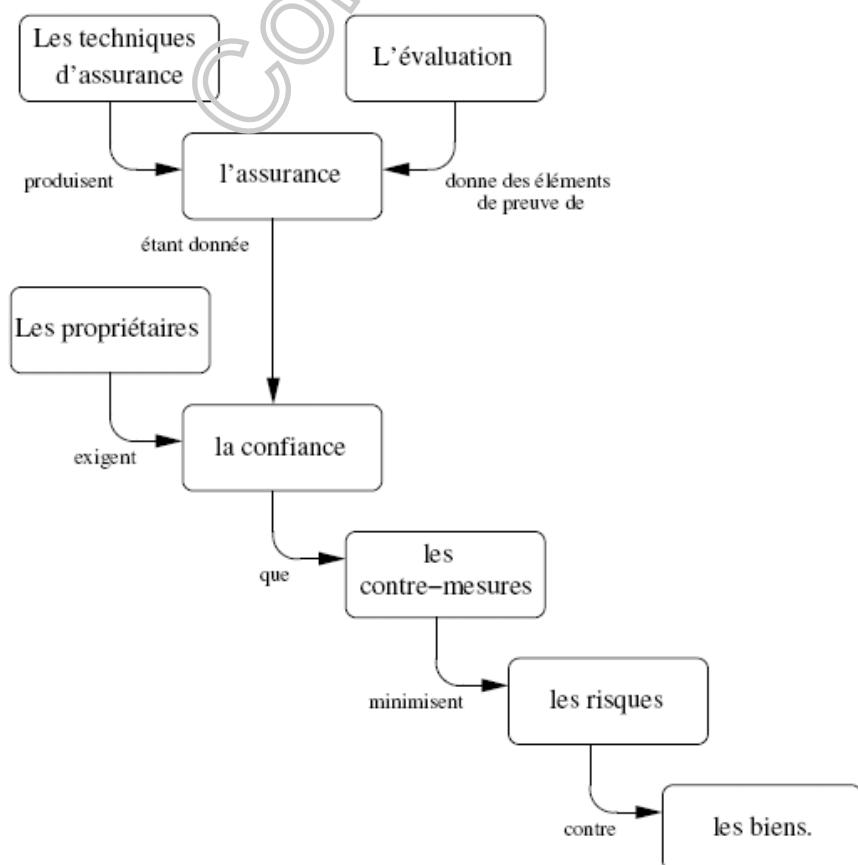


## Concepts de sécurité et relations



27

## Concepts de l'évaluation et relations



28

# Functionality versus Assurance

In assessing secure systems, two different aspects need considering

- ▶ **Functionality**, i.e. what security facilities are provided.

- ▶ **Assurance**, i.e. what guarantees are offered that the security functionality performs as claimed.

These two aspects are often reflected in **Security Evaluation Criteria**, from the **Orange Book** onwards.

## Assurance versus Complexity

The work involved in providing a high level of assurance in security features is proportional to the complexity of those features.

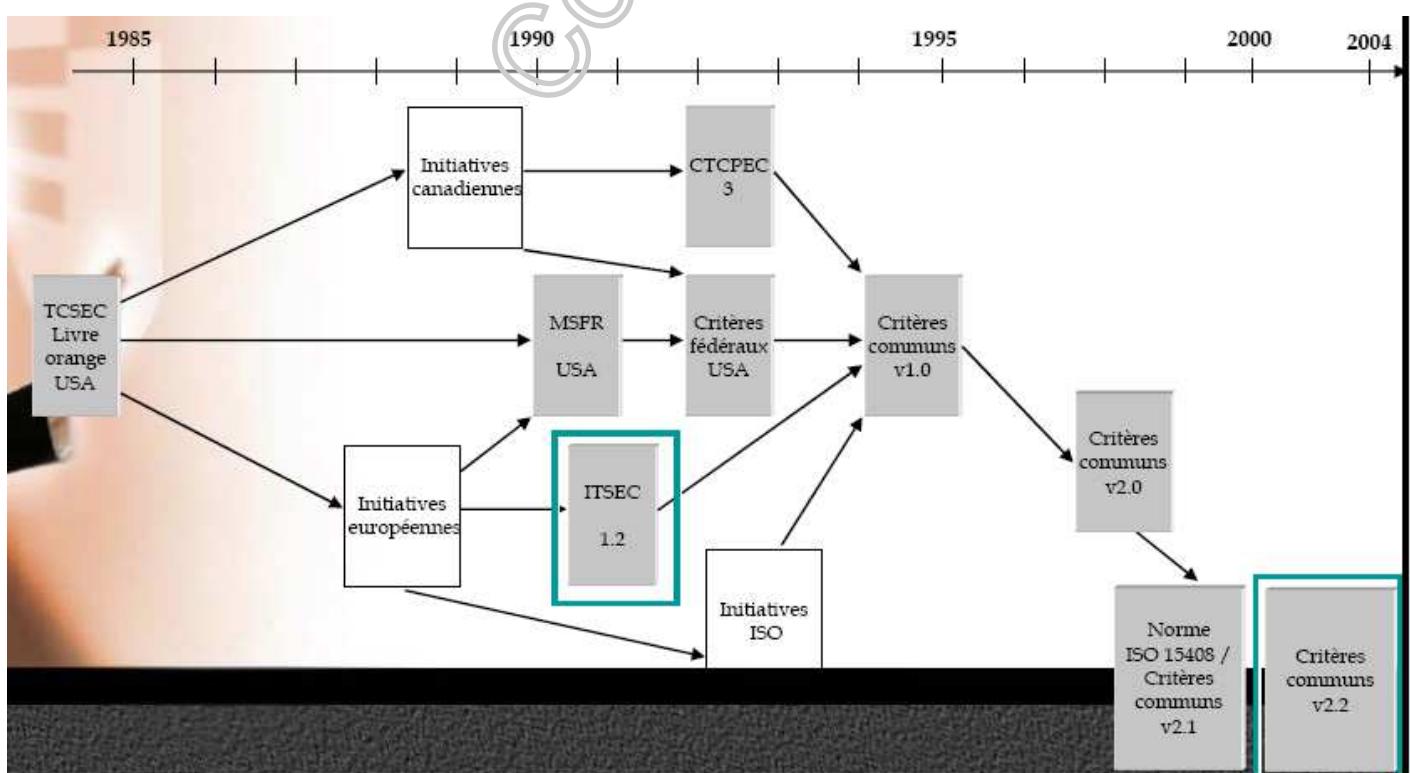
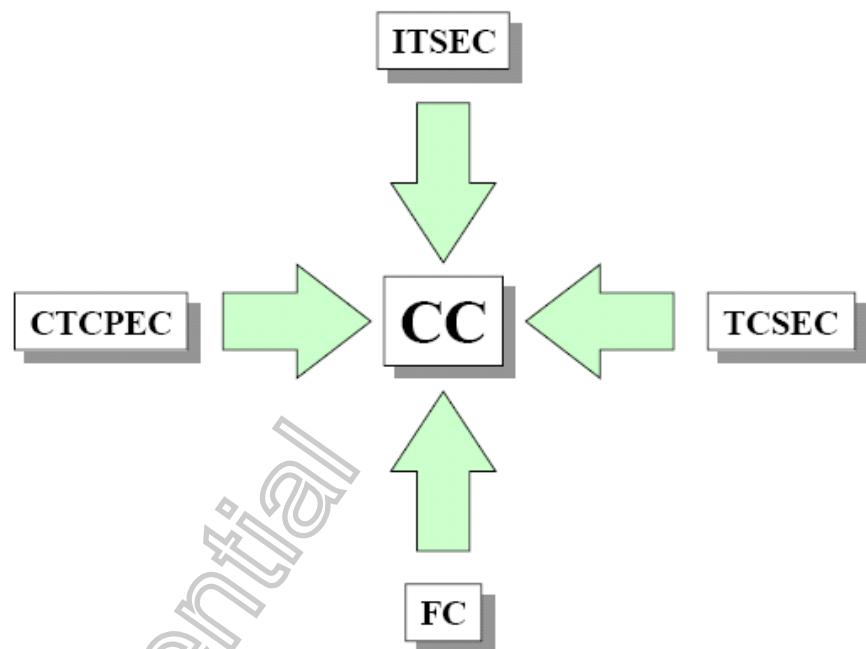
Hence, if a high level of assurance is required it makes sense to minimise complexity.

## Normes pour l'évaluation

Avant l'apparition des CC, les pays utilisaient des critères d'évaluation différents :

- l'Orange Book pour les États Unis ;
- les ITSEC pour la France, l'Allemagne, le Royaume-Uni et les Pays-Bas ;
- le CTCPEC pour le Canada.

## Harmonisation des critères existants



## Critères Communs

Les Critères Communs (CC) sont une norme qui a pour vocation d'être utilisée comme base pour l'évaluation des propriétés de sécurité des produits et systèmes des Technologies de l'Information (TI). Ils sont définis par le Common Criteria Interpretations Management Board (CCIMB) et sont depuis quelques années une norme ISO définie comme ISO/IEC 15408

- **CC : Common Criteria v2.1, August 1999**  
(<http://www.sssi.gouv.fr/documents/docs/CC/cc21.html>)
  - Part 1 : Introduction and general model  
August 1999 Version 2.1
  - Part 2 : Security functional requirements  
August 1999 Version 2.1
  - Part 3 : Security assurance requirements  
August 1999 Version 2.1
- **CEM : Common Methodology v1.0, August 1999**  
(<http://www.sssi.gouv.fr/documents/docs/CEM/cem.html>)
  - Evaluation methodology for PP and ST  
Version 1.0
  - Evaluation Methodology for /evals EAL1 to EAL4  
Version 1.0

33

## Une norme internationale

ISO/IEC JTC 1/SC27  
Technologies de l'information - Techniques de sécurité

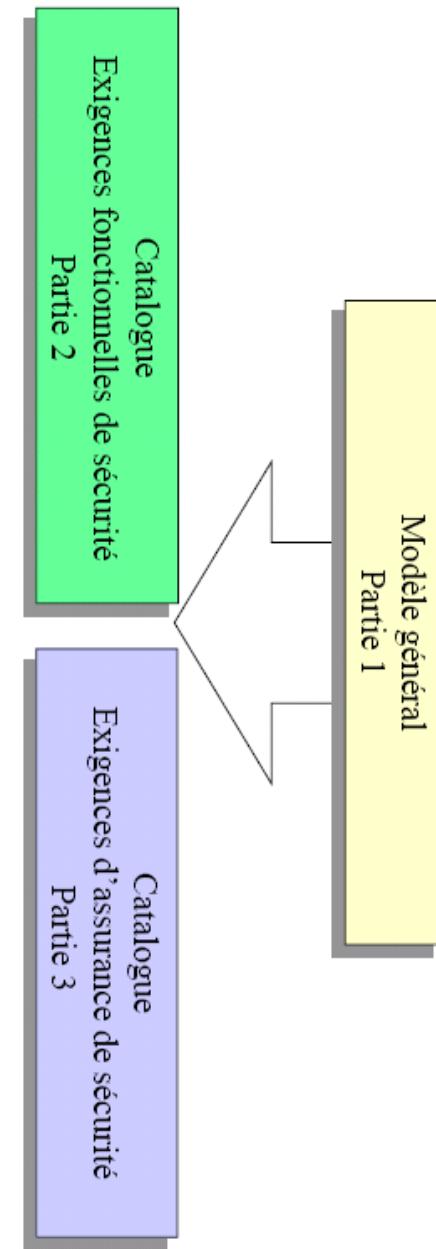


Juin 1999 Normalisation des Critères Communs

ISO/IEC IS 15408-1	Partie 1 Introduction et modèle général
ISO/IEC IS 15408-2	Partie 2 Exigences de sécurité fonctionnelles
ISO/IEC IS 15408-3	Partie 3 Exigences de sécurité d'assurance

34

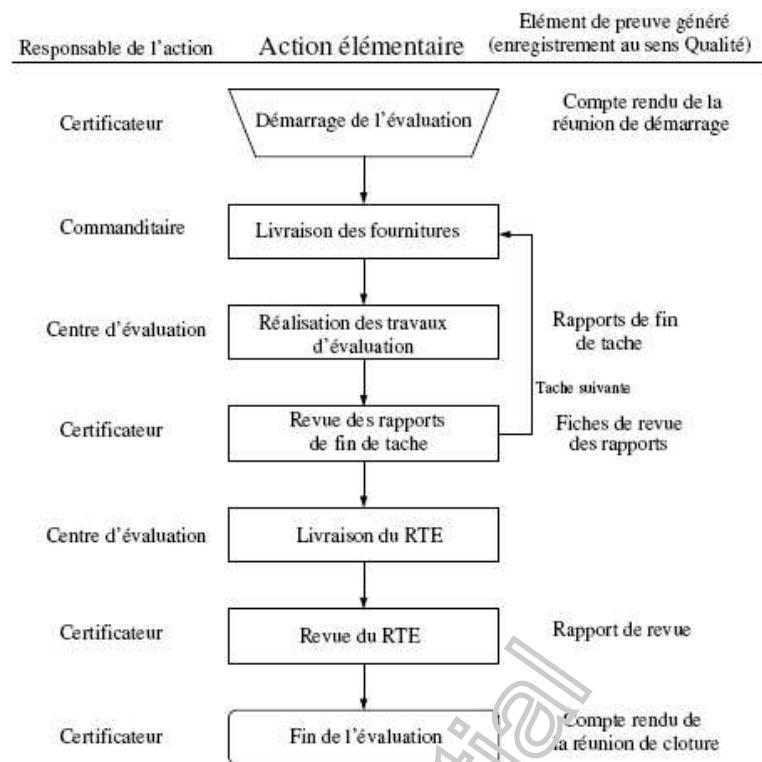
## Expression des exigences



Public

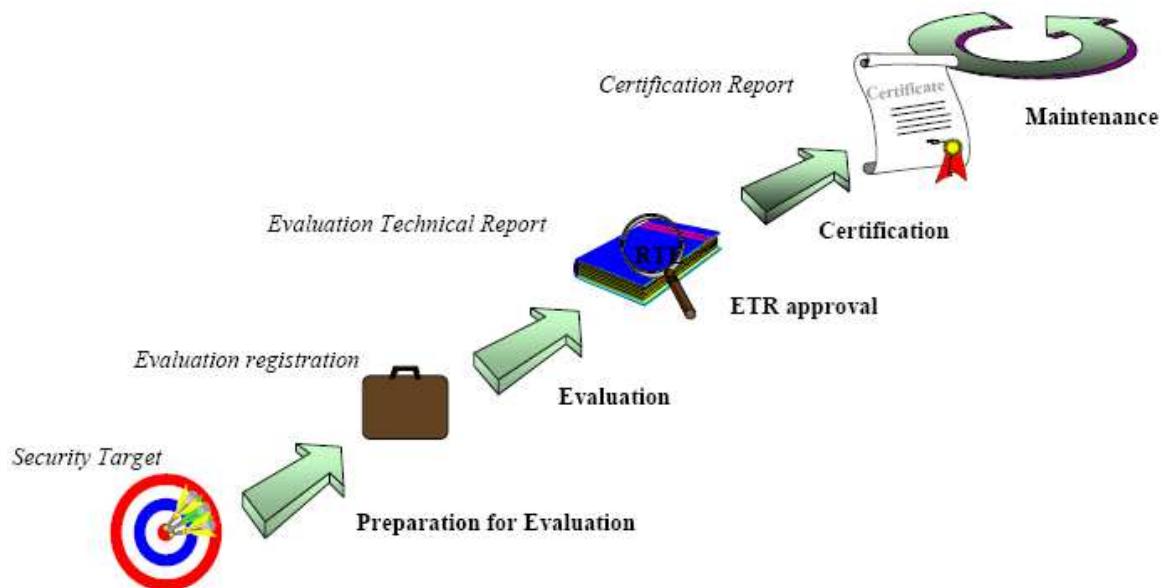
	Consumers	Developers	Evaluators
<b>Part 1</b>	Use for background information and reference purposes. Guidance structure for pPs.	Use for background information and reference for the development of requirements and formulating security specifications for TOEs.	Use for background information and reference purposes. Guidance structure for pPs and STs
<b>Part 2</b>	Use for guidance and reference when formulating statements of requirements for security functions	Use for reference when interpreting statements of functional requirements and formulating functional specifications for TOEs.	Use as mandatory statement of evaluation criteria when determining whether a TOE effectively meets claimed security functions.
<b>Part 3</b>	Use for guidance when determining required levels of assurance.	Use for reference when interpreting statements of assurance requirements and determining assurance approaches of TOEs.	Use as mandatory statement of evaluation criteria when determining the assurance of TOEs and when evaluating pPs and STs.

## Déroulement de l'évaluation



37

## Evaluation Process : main steps



38

## Quelques définitions

TOE (Target Of Evaluation) pour définir la cible d'évaluation, i.e. le produit ou le système TI à évaluer ;

ST (Security Target) pour désigner une cible de sécurité, i.e. un document rassemblant l'ensemble des exigences de sécurité et des spécifications à utiliser comme base pour l'évaluation d'une TOE identifiée ;

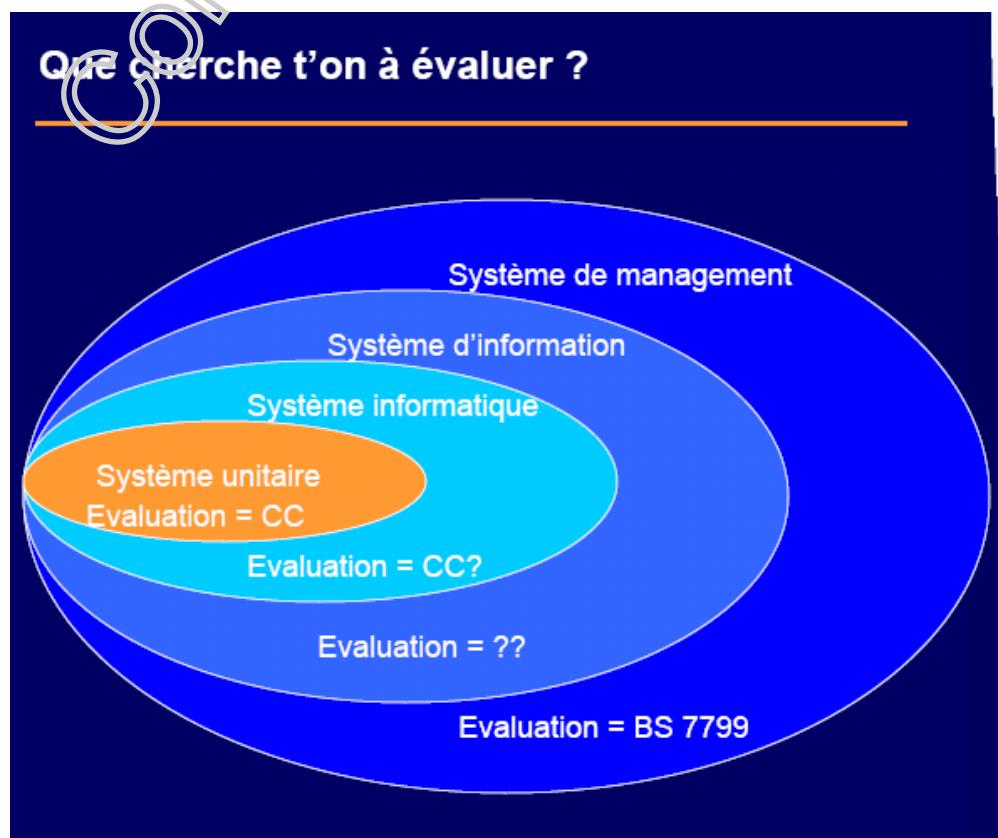
PP (Protection Profil) pour désigner un profil de protection, i.e. un document rassemblant l'ensemble des exigences de sécurité valables pour une catégorie de TOE qui satisfait des besoins spécifiques d'utilisateurs, indépendamment de son implantation (e.g. il existe des PPs pour l'évaluation de puce nue – i.e. sans applicatif –, pour le domaine des applications bancaires)

- Target of evaluation (TOE) : an IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation. It defines assets to protect.
- TOE Security Functions (TSF) : A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
- TOE Security Policy (TSP) : A set of rules that regulate how assets are managed, protected, and distributed within a TOE
- Security Target (ST) : Defines the target of evaluation, the environment, the threats, assets to protect, security objectives, assumptions.

39

## Périmètre

Et même un sous ensemble



40

## Etablissement de la confiance !

### Pourquoi une évaluation des systèmes d'information?

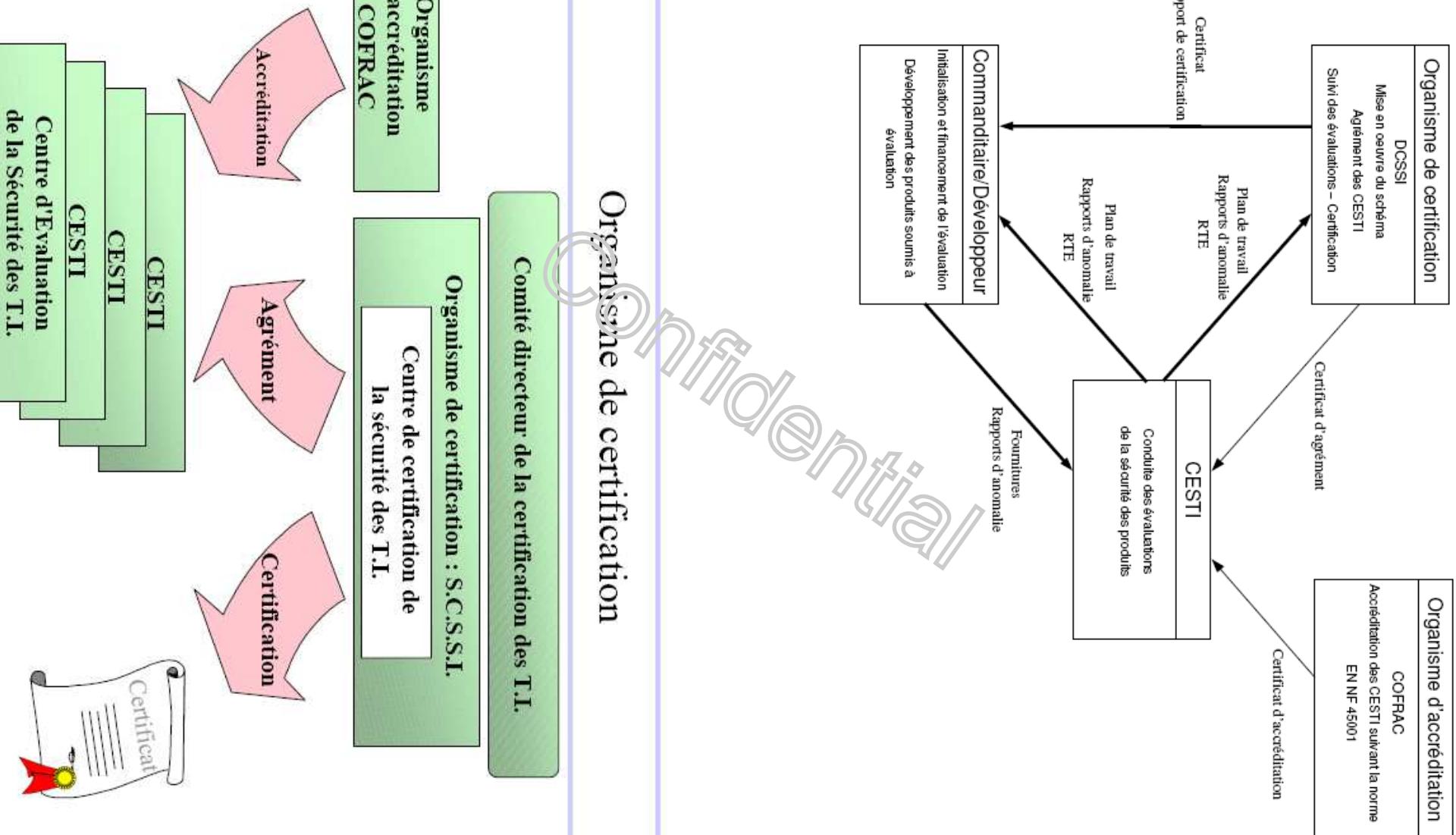


## Etablissement de la confiance !

### Usages actuels des Critères Communs

- Dans les projets, pour qualifier une technologie
  - Chronotachygraphe
  - GIE Cartes Bancaires
  - Messagerie OTAN
- Pour la fourniture de matériels de sécurité
  - Cartes bancaires
  - Firewalls
  - Modules cryptographiques
- Dans le domaine de la Défense
  - Évaluation produits de haute sécurité
  - Évaluation de systèmes complexes (ex SINAPSE)

## Le schéma Français



## Centres d'évaluation

CESTI	Opérationnel	En formation
Algoriel AQL CEACI (CNES-SOREP) CEA/LETI SERMA Technologies	ES <sup>2</sup>	

Agréés pour le domaine de la carte à puce :  
CEACI,  
CEA/LETI,  
SERMA Technologies



Un environnement international...

45

Organismes de certification:

BSI	Allemagne
CESG	Royaume Uni
DSD	Australie
CSE	Canada
NIST et NSA	Etats-Unis
SCSSI	France

## Reconnaissance mutuelle



Mars 1998

Accord SOGIS

Senior Officials Group for Information Security of the European Commission

- Accord entre 12 pays européens :

Allemagne, Espagne, Finlande, France, Grèce, Italie, Pays-Bas, Portugal, Norvège, Royaume Uni, Suède et Suisse

- Organismes de certification qualifiés

Allemagne, France et Royaume Uni

- Certificat ITSEC E1 à E6 et CC EAL1 à EAL7



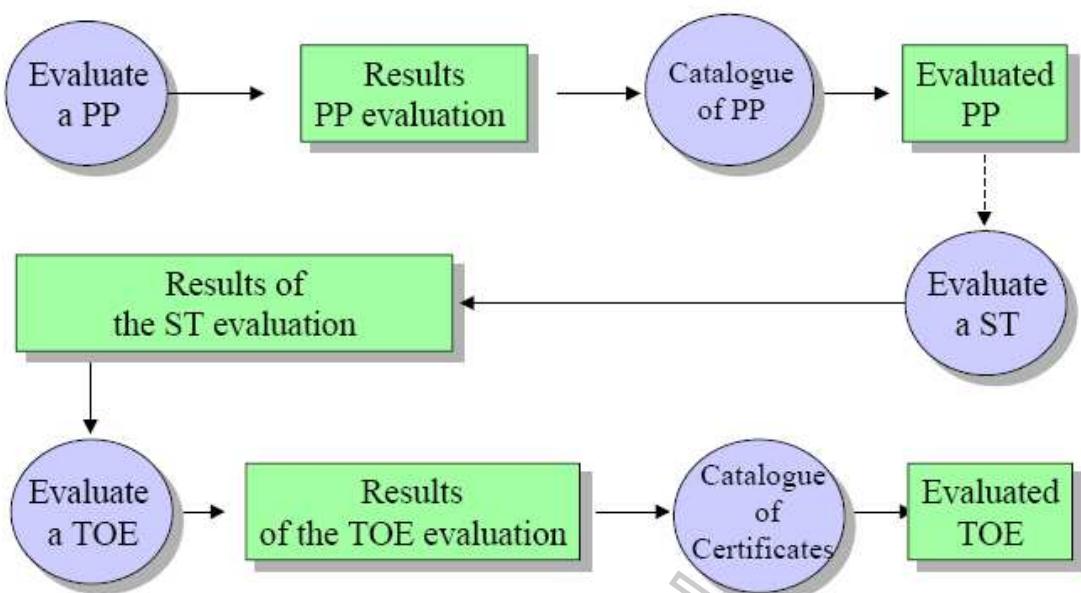
## Reconnaissance mutuelle

Mai 2000      Arrangement harmonisé



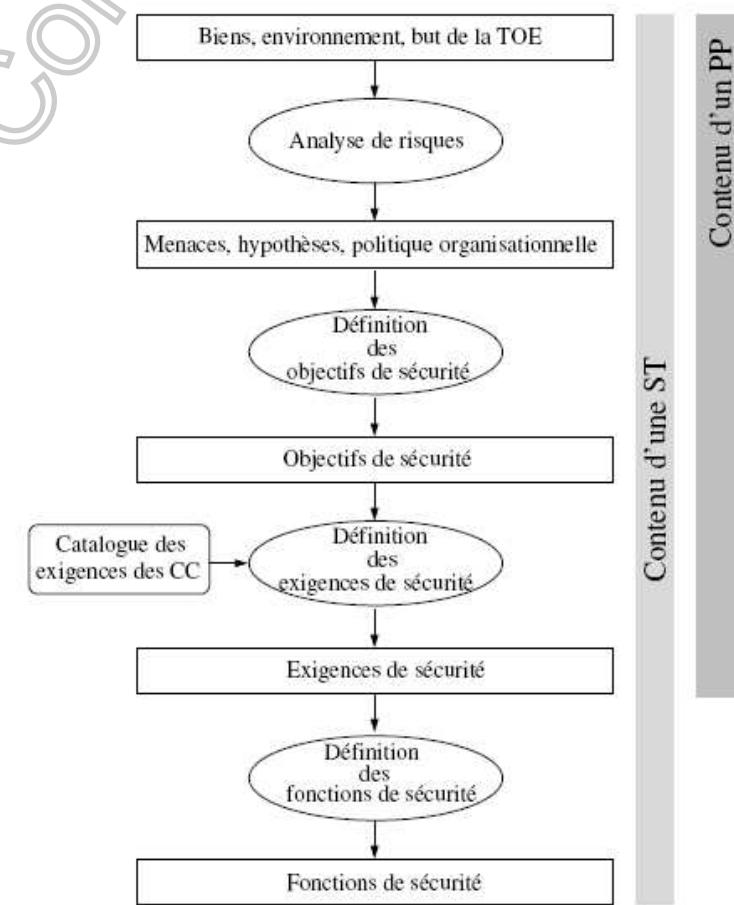
- Accord entre 13 pays :  
Allemagne, Australie&Nouvelle Zélande, Canada, Espagne, États-Unis, Finlande, France, Grèce, Italie, Pays-Bas, Norvège, Royaume Uni et Suède
- Organismes de certification qualifiés  
BSI (Allemagne), DSD (Australie), CSE (Canada), NIST/NSA (États-Unis), SCSSI (France) et CESG (Royaume Uni)
- Organismes de certification commerciaux
- Certificat CC EAL1 à EAL4

## CC evaluations and their results

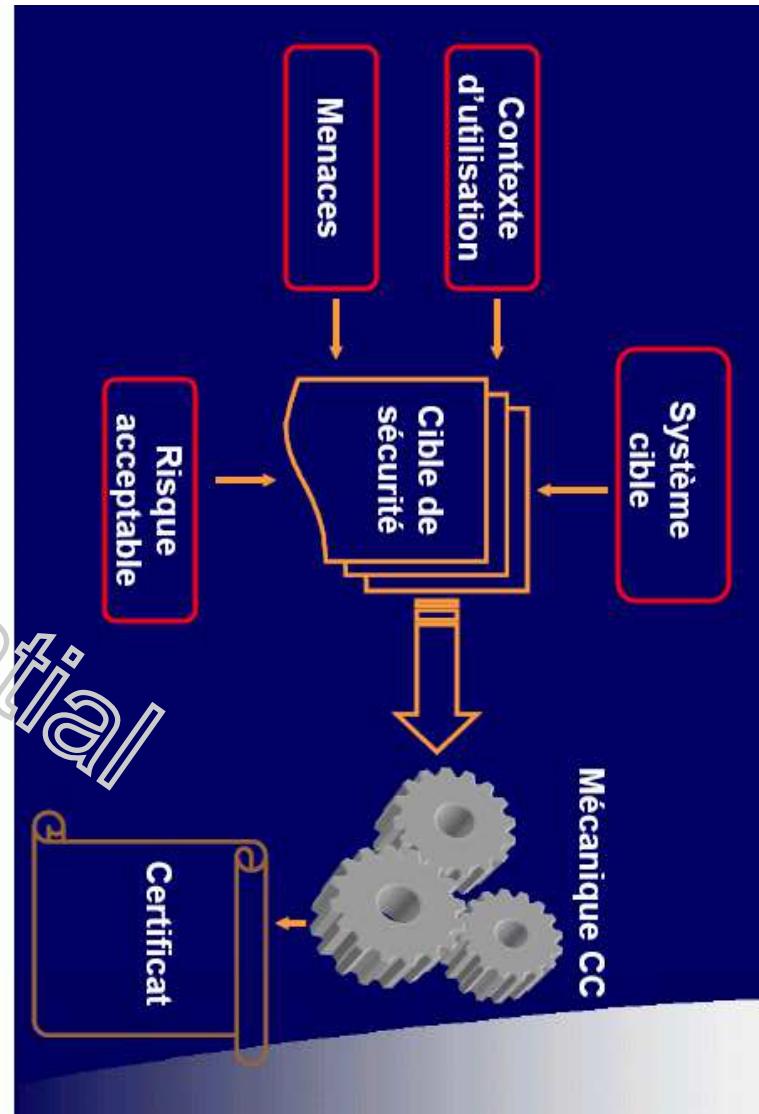


49

## Contenu d'une ST et d'un PP



50



## Un certificat

### Retour d'expérience sur l'évaluation d'un VPN

EADS

Schéma français d'évaluation et de certification de la sécurité des technologies de l'information

CERTIFICAT 2002/26

M>Tunnel 2.5  
(référence MT25-BA3-08)

Développeur : EADS Telecom

**Critères Communs**

**EAL2 Augmenté**

(ADV, HLD, ATE, UCL, J\*, ANA, VLA, 2\*)

\*enfert à la partie finale de l'évaluation au risque d'immunité des anomalies

Commanditaire : EADS Telecom  
Centre d'évaluation : AQI - groupe Silicom

Le 7 février 2003.

Le Directeur Général de la Sécurité des Systèmes d'Information  
Henri Serre

*HS*



Le présent document est une évaluation et une certification de la sécurité d'un système de communication et de transmission de données à distance (VPN) développé par EADS Telecom dans le cadre du programme de recherche et développement national de sécurité de l'information (RASI) et en accord avec les normes et critères de sécurité définis par la Direction Générale de la Sécurité des Systèmes d'Information (DGSI). Il vise à démontrer que ce système répond aux exigences de sécurité fixées par ces normes et critères. L'évaluation a été effectuée par un organisme indépendant et impartial, conformément aux méthodes et procédures recommandées par la DGSI. Le résultat de cette évaluation est considéré comme étant satisfaisant pour l'application de la certification. Il est recommandé aux utilisateurs de prendre leurs propres mesures pour assurer la sécurité de leur système de communication et de prendre en compte les résultats de cette évaluation dans leur décision d'achat et d'utilisation.

Document officiel de la défense nationale. Direction centrale de la sécurité des systèmes d'information. 51 boulevard de la Paix. 92100 Paris. FR 45

## Quelques chiffres !

### Retour d'expérience sur l'évaluation d'un VPN

#### ⊕ Contexte :

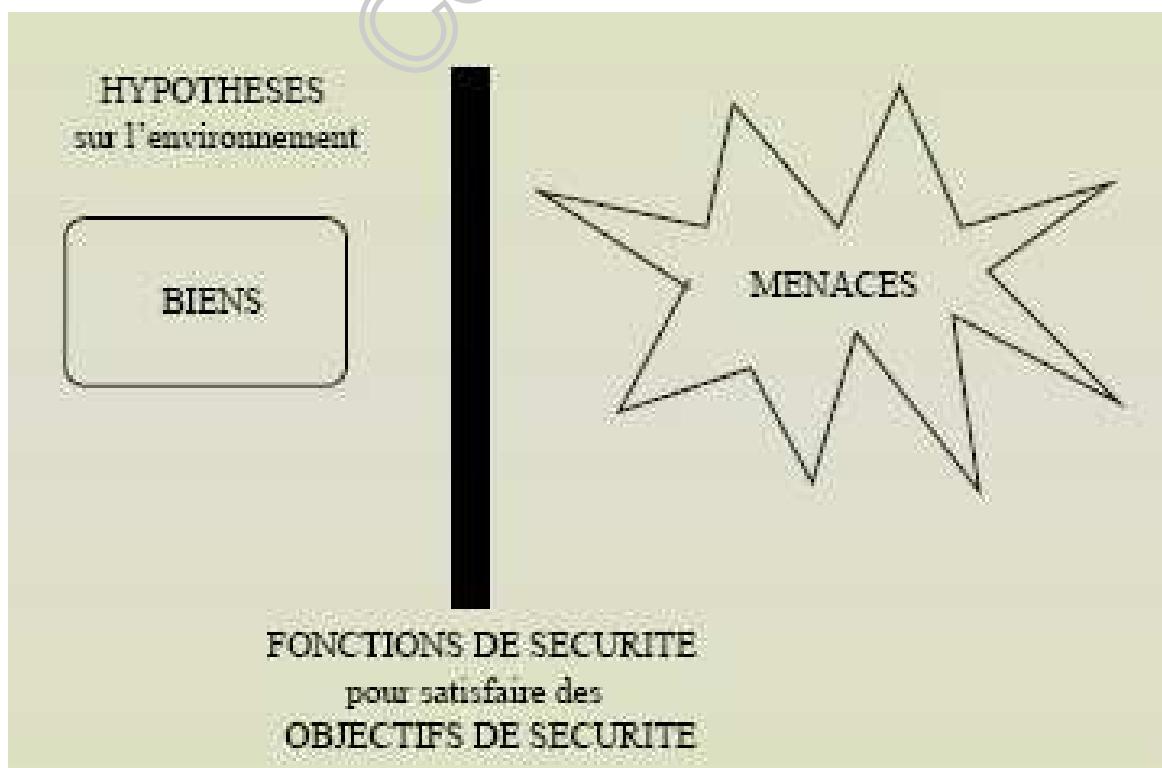
- Sollicitation pour les besoins gouvernementaux
- Évaluation CC EAL2+ (Qualification standard)
- Périmètre le plus englobant possible
- Certification body : DCSSI
- Centre d'évaluation : AQL
- Obtention du certificat en décembre 2002

#### ⊕ Caractéristiques du projet

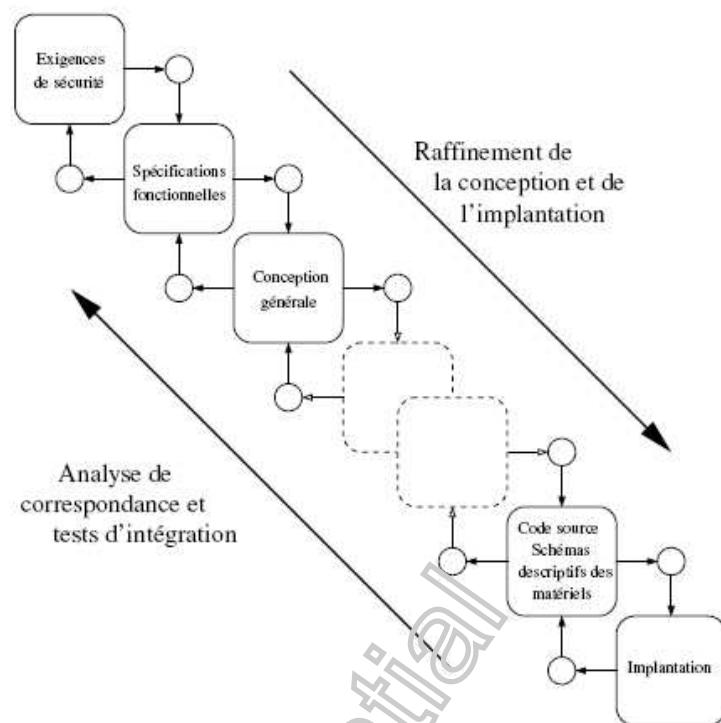
- Durée : 7 mois
- Coût : 200K€

## Les bases des critères communs

### Critères Communs (ISO15408)

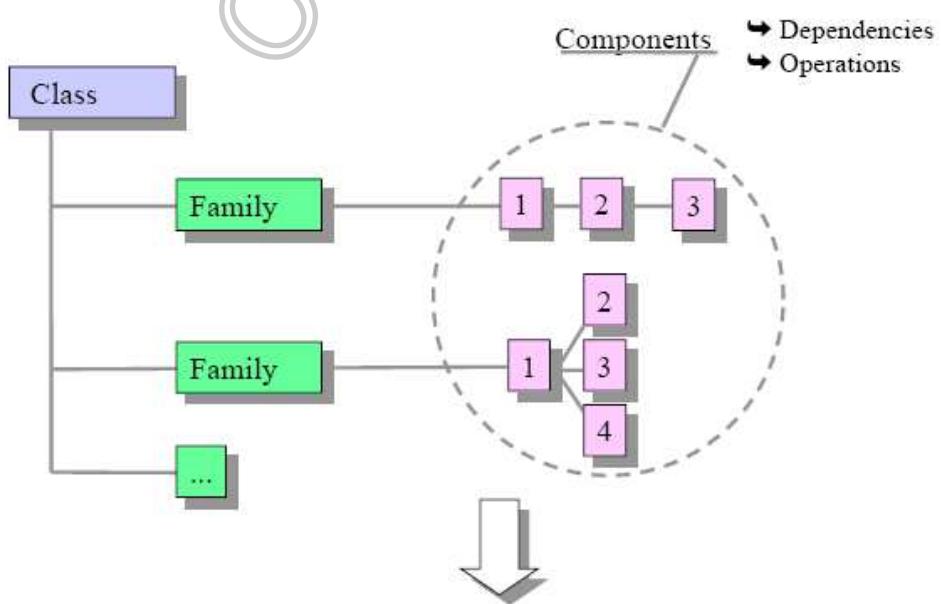


## Modèle de développement d'une TOE



55

## Requirements : hierarchical structure

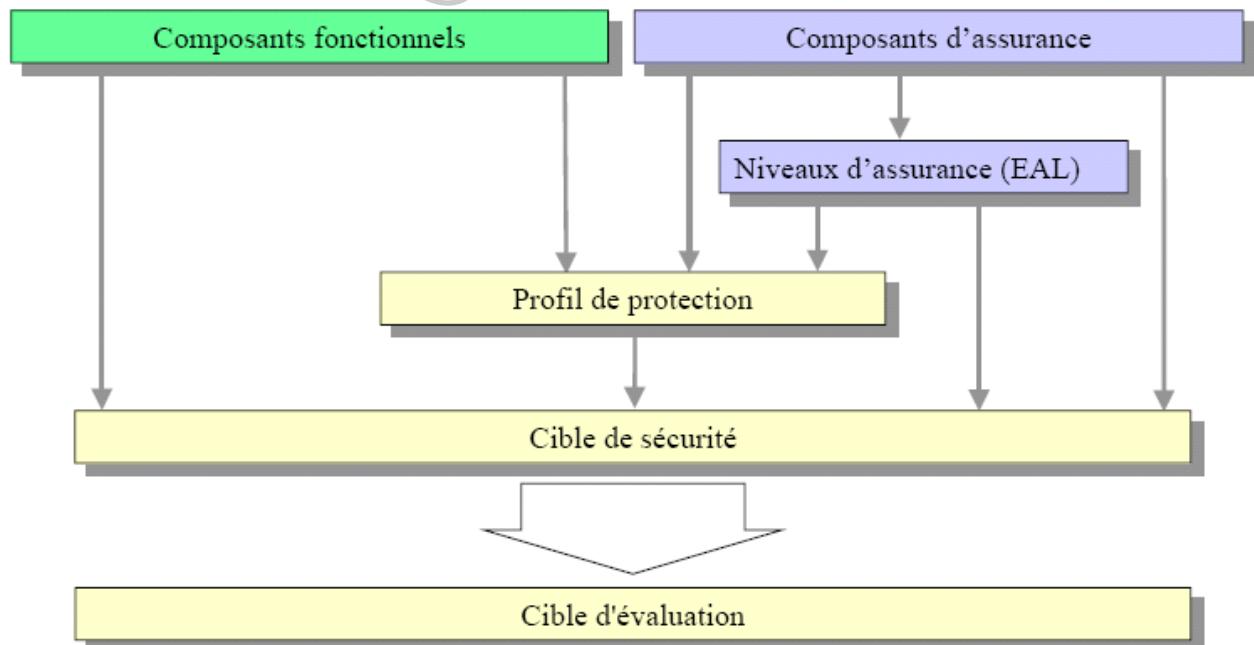


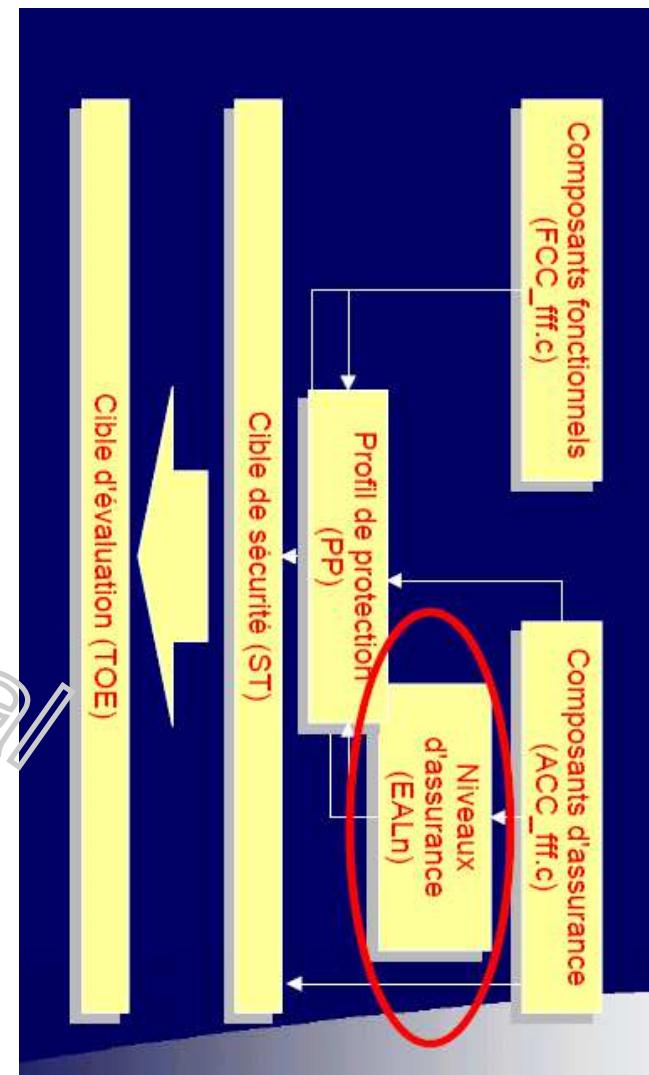
Applies to functional and assurance requirements

56

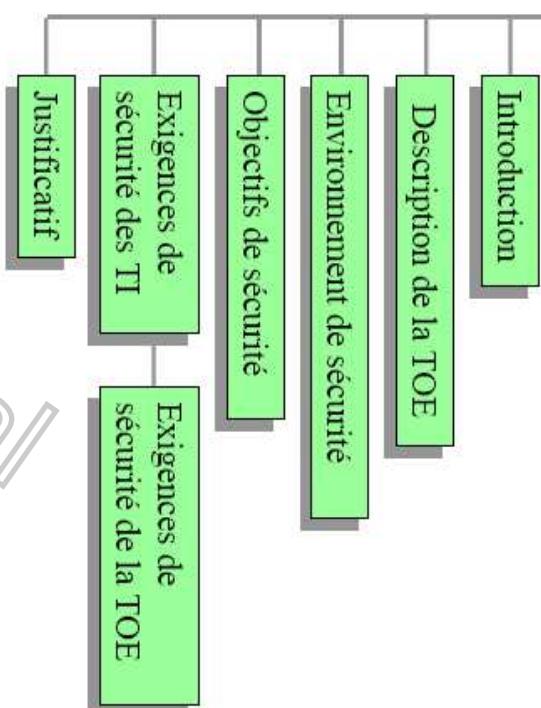
- **Class** : A grouping of families that share a common focus.
- **Family** : A grouping of components that share security objectives but may differ in emphasis or rigour.
- **Component** : The smallest selectable set of elements that may be included in a PP, an ST, or a package.
- **Package** : A reusable set of either functional or assurance components (eg. an EAL), combined together to satisfy a set of identified security objectives.

## Utilisation des catalogues

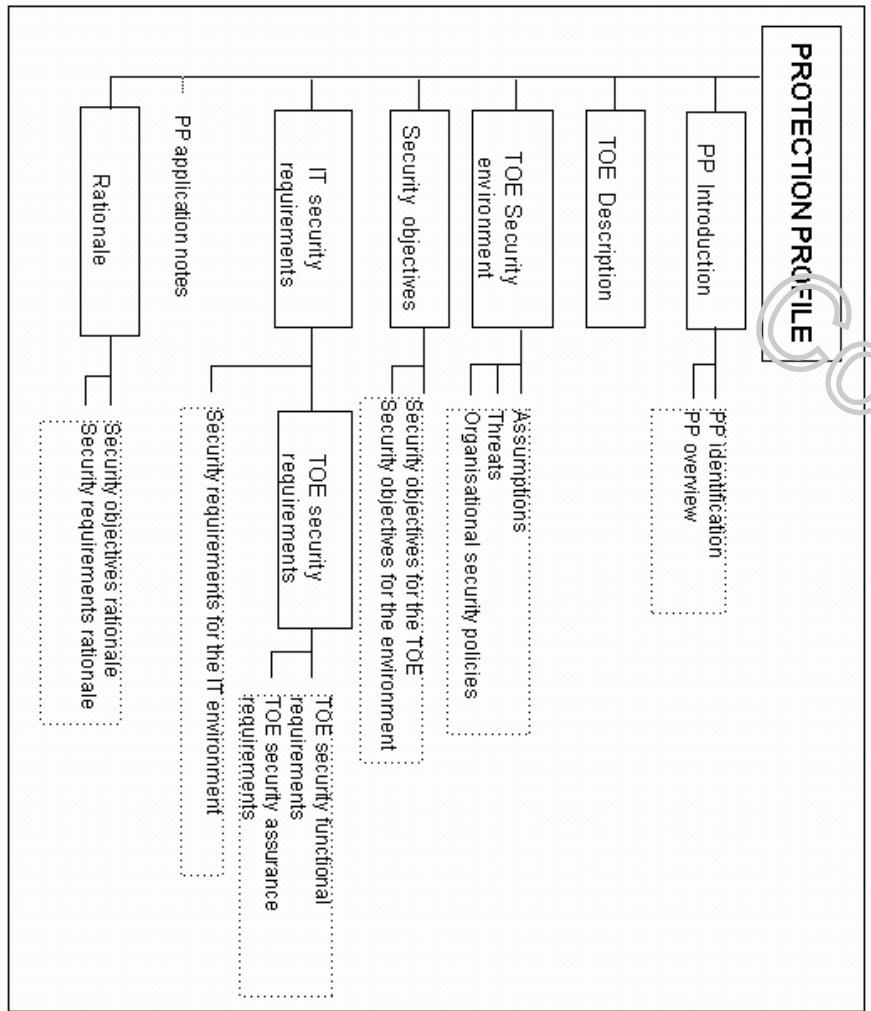




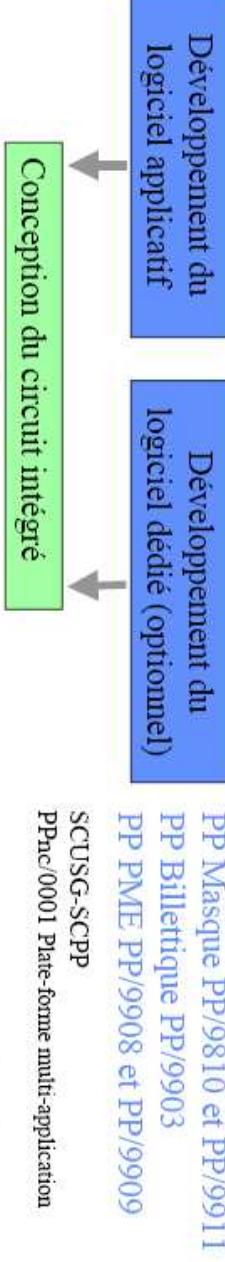
## Profil de protection



## Protection Profile



## PP Cartes à puce

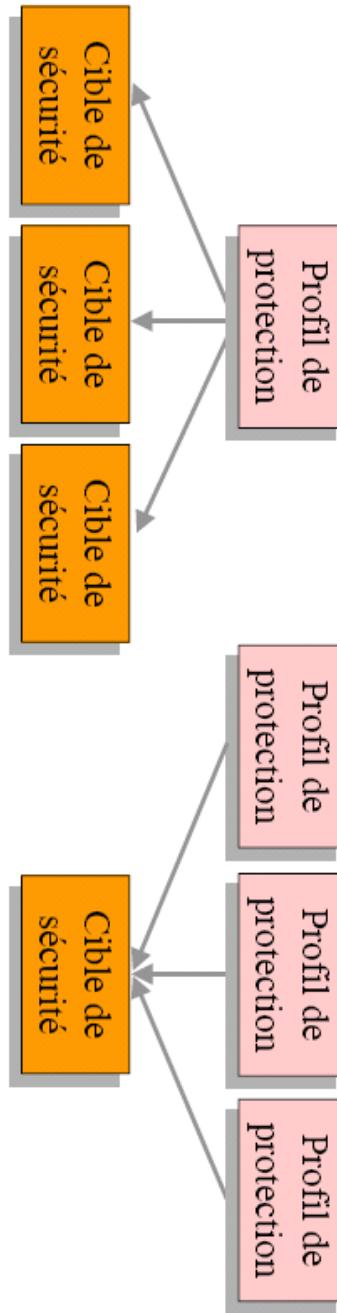


PP Circuit intégré PP/9806

PP Encartage PPnc/9910

PP Personnalisation PPnc/9912

## Utilisation des profils de protection



## PP Masque

PP/9810: Smartcard integrated circuit with embedded software

EAL4+ (ADV\_IMP.2, ALC\_DVS.2, AVA\_VLA.4)

Schlumberger



PP/9911: Smartcard integrated circuit with embedded software

EAL4+ (ADV\_IMP.2, ALC\_DVS.2, AVA\_VLA.4)

EuroSmart



Confidential

PP Billetique

PP/9903: Carte à puce billetique avec et sans contact

EAL4+ (ADV\_IMP.2, AVA\_VLA.4)

SNCF, RATP



RATP

PP PME

PP/9908: Intersector Electronic Purse and Purchase Device

Version for Pilot Schemes only

EAL1+ (AVA\_VLA.2)

Banque de France, EuroSmart, GIE Cartes Bancaires CB



PP/9909: Intersector Electronic Purse and Purchase Device

EAL4+ (ADV\_IMP.2, ALC\_DVS.2, AVA\_VLA.4)

Banque de France, EuroSmart, GIE Cartes Bancaires CB



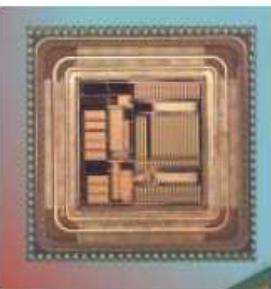
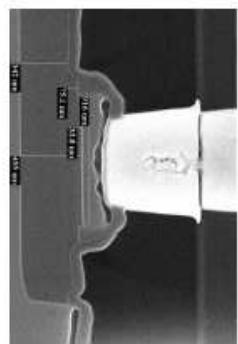
67

PP Composant pour carte à puce

PP/9806: Smartcard integrated circuit

EAL4+ (ADV\_IMP.2, ALC\_DVS.2, AVA\_VLA.4)

EuroSmart,  
Motorola, Philips, Siemens, STMicroelectronics,  
Texas-Instruments



68

## PP Encartage et personnalisation

PPnc/9910: SmartCard embossing Sites

Paquet d'assurance spécifique

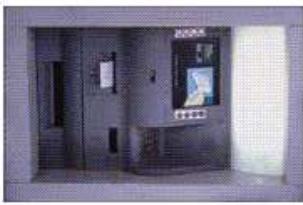
AFCP (Association des Fabricants et des Personnaliseurs de Carte)



PPnc/9912: SmartCard personalization Sites  
Paquet d'assurance spécifique  
GIP CPS



## PP Terminaux



PP/9907: Automates bancaires

EAL4+ (AVA\_VLA.3)  
Bull  
Dassault A.T.  
IBM

NCR  
Siemens Nixdorf  
Wang Global

PP/0002: Lecteur transactionnel de cartes à puce

EAL4+ (ADV\_IMP.2, AVA\_VLA.3)  
Cyber-COMM

## PP Echanges de données informatisées

PP/9802: Transactions portant sur des données non confidentielles

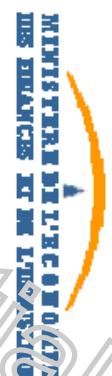
EAL3+ (ADV\_IMP.1, ADV\_LLD.1, ALC.TAT.1)

Ministère de l'Economie, des Finances et de l'Industrie

PP/9803: Transactions portant sur des données confidentielles

EAL3+ (ADV\_IMP.1, ADV\_LLD.1, ALC.TAT.1)

Ministère de l'Economie, des Finances et de l'Industrie



71

## Confidential PP Firewalls

PP/9904: Firewall à exigences réduites

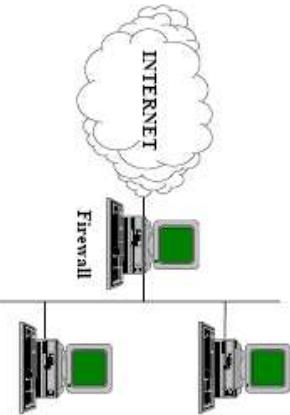
EAL4+ (ADV\_IMP.2, AVA\_CCA.1, AVA\_VLA.3)  
DGA

PP/9905: Firewall à exigences élevées

EAL5+ (ALC\_FLR.2, AVA\_VLA.4)  
DGA

PP/9906: Passerelle filtrante de sécurité

configurable  
EAL5  
DGA



## PP Infrastructure de Gestion de Clés

Ressource cryptographique  
PPnc/0003: Ressource cryptographique pour une infrastructure de gestion des clés  
EAL 5+ (AVA\_VLA4, AVA\_CCA3)

Infrastructure de gestion de clés  
PPnc/0004: Infrastructure de gestion de clés  
EAL 3+ (ADV\_IMP1, ADV\_LLD1, ALC\_TAT1, AVA\_VLA3)  
SCSSI

Autorité d'enregistrement

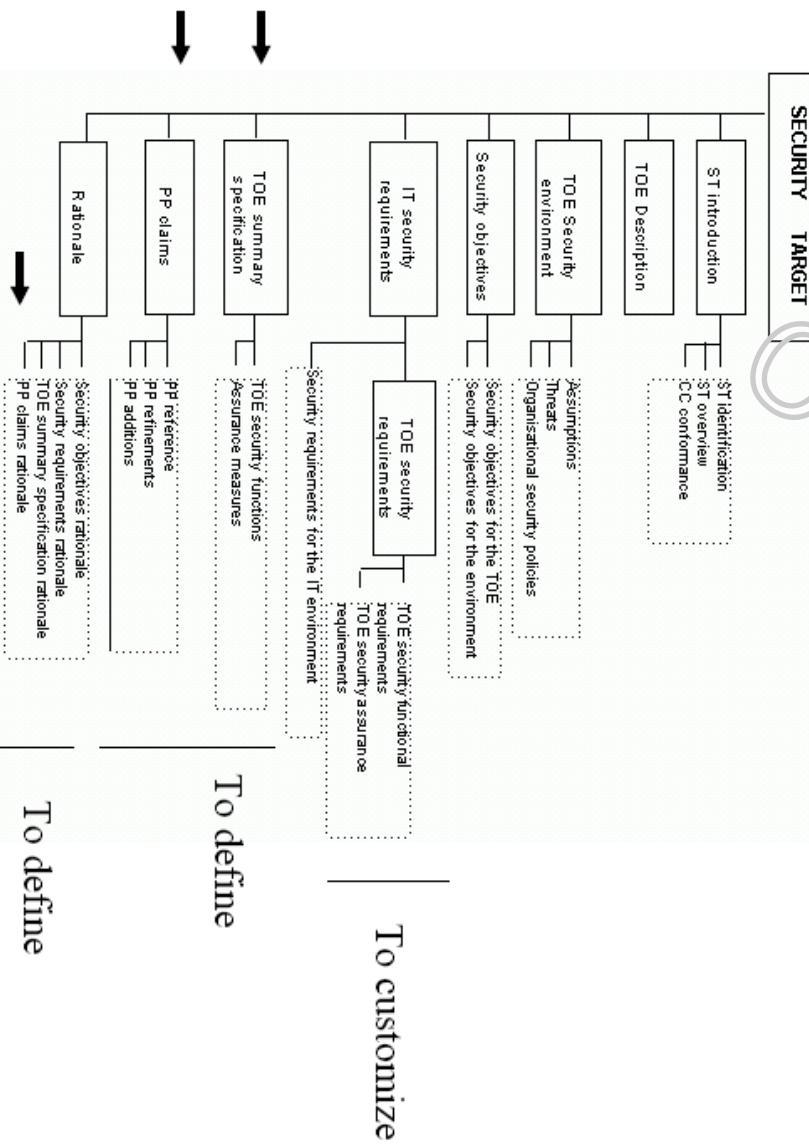
PPnc/0005: Autorité d'enregistrement  
EAL 3+ (AVA\_VLA3, ADV\_IMP1, ADV\_LLD1, ADV\_SPM1, ALC\_TAT1)

SCSSI

Autorité de certification

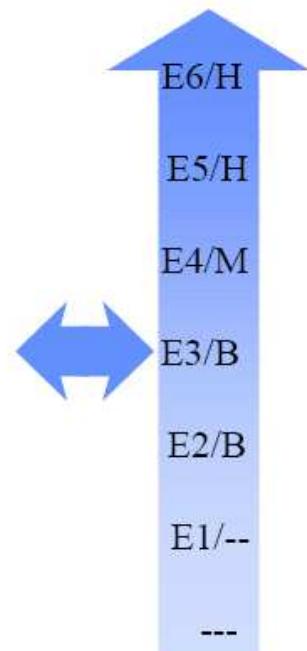
PPnc/0006: Autorité de certification  
EAL 3+ (AVA\_VLA3, ADV\_IMP1, ADV\_LLD1, ADV\_SPM1, ALC\_TAT1)  
SCSSI

## Security Target



EAL 7	formally verified design and tested
EAL 6	semi-formally verified design and tested
EAL 5	semi-formally designed and tested
EAL 4	methodically designed, tested, and reviewed
EAL 3	methodically tested and checked
EAL 2	structurally tested
EAL 1	functional tested

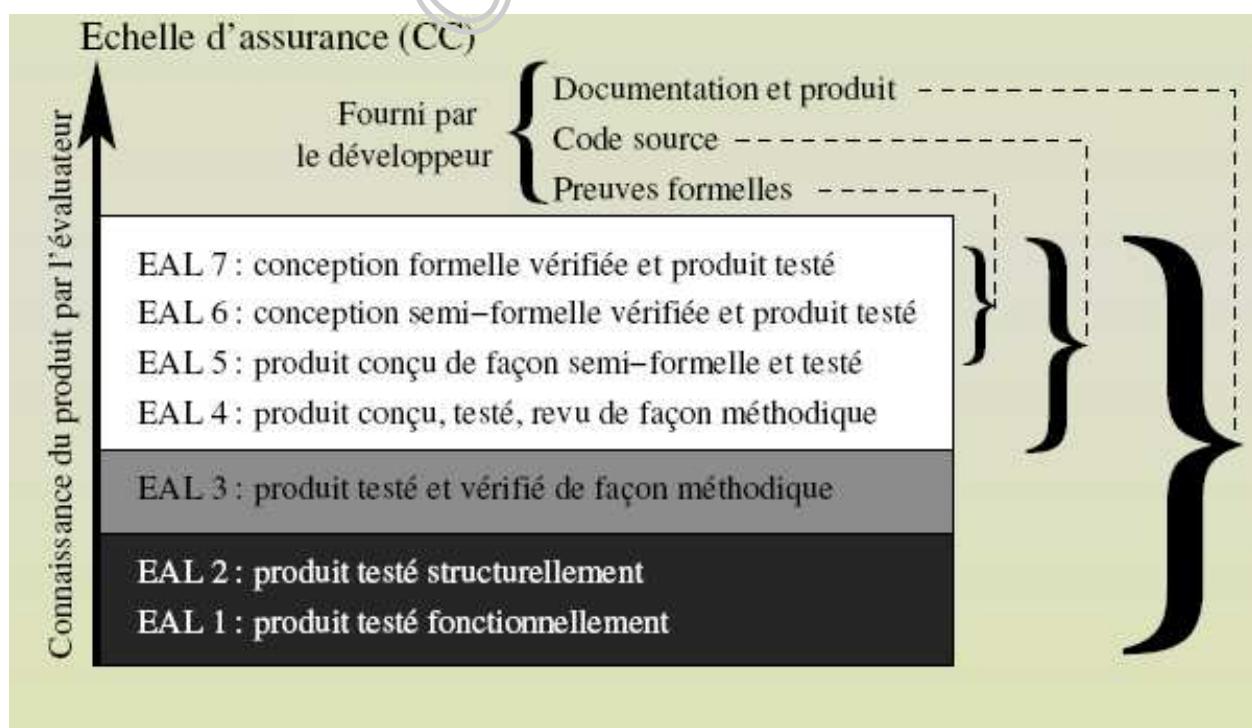
CC



ITSEC

75

## L'échelle d'assurance



76

## L'assurance des Critères Communs

**ST**  
Biens  
Menaces -- Hypothèses  
Politique  
Objectifs de sécurité  
Exigences de sécurité  
Fonctions de sécurité

Assurance que le processus de développement est adéquatement organisé

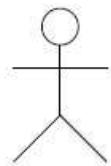
Assurance que la ST est complètement et correctement implantée



Assurance que le client déploie correctement la TOE

Assurance que la TOE fait exactement ce qu'elle est supposée faire et ni plus, ni moins

Utilisateur final



77

## Les exigences d'assurance des Critères Communs

**ST**  
Biens  
Menaces -- Hypothèses  
Politique  
Objectifs de sécurité  
Exigences de sécurité  
Fonctions de sécurité

Modélisation du cycle de vie

Développement sécurisé

Outils et techniques

Gestion de configuration

Spécification fonctionnelle

Design de haut niveau

Design de bas niveau

Implantation



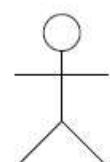
Livraison

Installation

Utilisation sécurisée

Correction des défauts

Utilisateur final



Tests

Vulnérabilité / canaux cachés

Guide d'administration

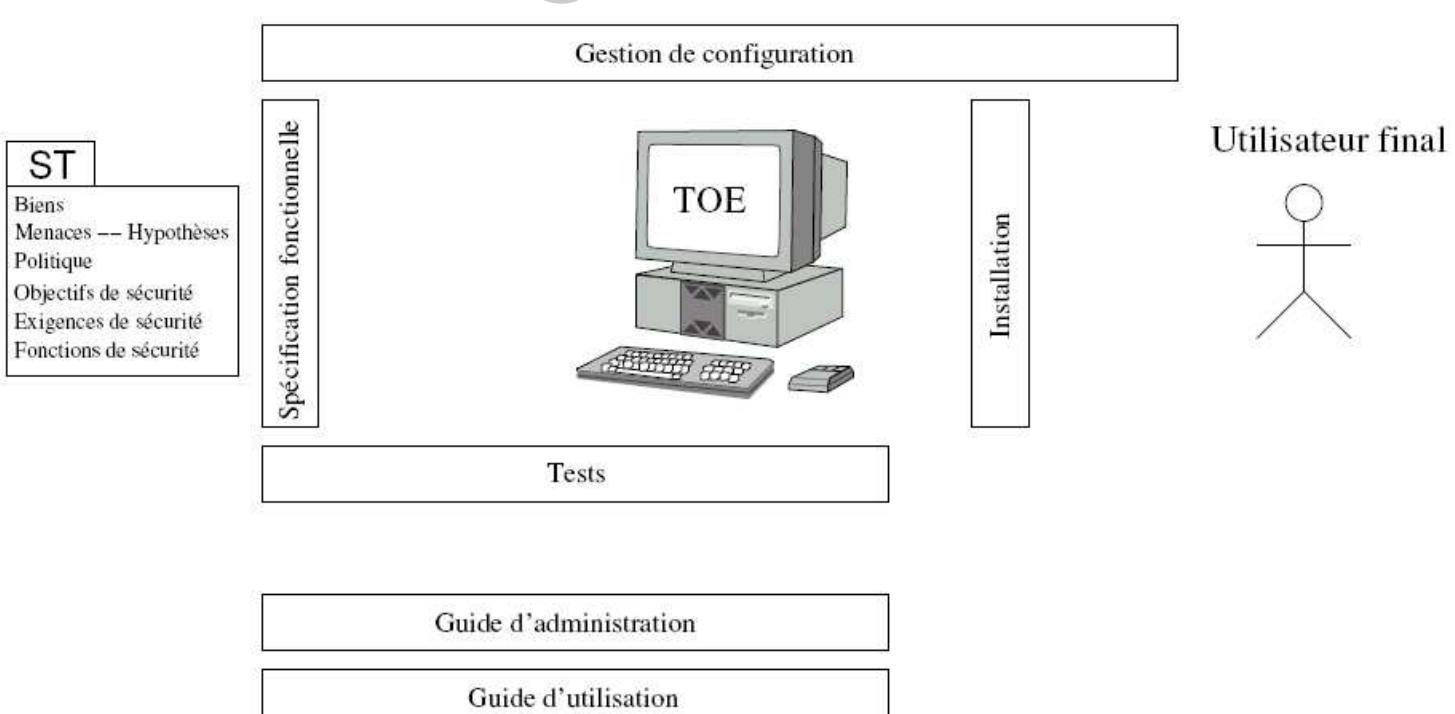
Guide d'utilisation

# Evaluation Assurance level summary

Assurance Class§	Assurance Family§	Assurance Components by Evaluation Assurance Levels						
		EAL1§	EAL2§	EAL3§	EAL4§	EAL5§	EAL6§	EAL7§
Configuration management§	ACM_AUT§	1§	1§	1§	1§	1§	2§	2§
	ACM_CAPS§	1§	2§	3§	4§	4§	5§	5§
Delivery and operations§	ACM_SCPS§	1§	1§	1§	2§	3§	3§	3§
	ADO_DELS§	1§	1§	1§	2§	2§	2§	3§
Development§	ADO_IGSS§	1§	1§	1§	1§	1§	1§	1§
	ADV_FSPS§	1§	1§	1§	2§	3§	3§	4§
Life cycle support§	ADV_HLDS§	1§	1§	2§	2§	3§	4§	5§
	ADV_IMPS§	1§	1§	1§	1§	2§	3§	3§
Tests§	ADV_INTS§	1§	1§	1§	1§	1§	2§	3§
	ADV_LLD§	1§	1§	1§	1§	1§	2§	2§
Vulnerability assessment§	ADV_RCRS§	1§	1§	1§	1§	2§	2§	3§
	ADV_SPM§	1§	1§	1§	1§	3§	3§	3§
Guidance documents§	AGD_ADM§	1§	1§	1§	1§	1§	1§	1§
	AGD_USRS§	1§	1§	1§	1§	1§	1§	1§
TOE	ALC_DVSS§	1§	1§	1§	1§	1§	2§	2§
	ALC_FLRS§	1§	1§	1§	1§	1§	2§	2§
TOE	ALC_LCD§	1§	1§	1§	1§	2§	2§	3§
	ALC_TATS§	1§	1§	1§	1§	2§	3§	3§
TOE	ATE_COVS§	1§	1§	2§	2§	2§	3§	3§
	ATE_DPT§	1§	1§	1§	1§	2§	2§	3§
TOE	ATE_FUN§	1§	1§	1§	1§	1§	2§	2§
	ATE_IND§	1§	2§	2§	2§	2§	2§	3§
TOE	AVA_CCAS§	1§	1§	1§	1§	1§	2§	2§
	AVA_MSUS§	1§	1§	1§	2§	2§	3§	3§
TOE	AVA_SOFS§	1§	1§	1§	1§	1§	1§	1§
	AVA_VLAS§	1§	1§	1§	2§	3§	4§	4§

79

## Les exigences d'assurance des Critères Communs pour le niveau EAL1



80

## EAL 1

- Could be used for an evaluation without the developer

- TOE security functions analysis
- TOE functional specifications and interfaces
- Security functions independent testing

## EAL 2

- Low-level independent evaluation

- TOE security functions analysis
- TOE functional specification and interfaces
- TOE sub-systems high-level design
- Review of security functions black-box tests done by the developer
- Obvious vulnerability assessment

## EAL 3

- Moderate level evaluation

- Grey-box testing
- Independent confirmation of a selected sample of developer tests results
- Search for vulnerabilities justified by the developer
- Development environment control
- TOE configuration management

- **Complete white-box evaluation**

- TOE modules low-level design
- Subset of implementation representation
- Independent search for vulnerabilities
- Conformance of Development process against a life-cycle model
  - Tools identification
  - Automated configuration management

- **High level of assurance obtained through a rigorous method of development**

- All implementation analyses
- Formal model and semi-formal presentation of functional specification and high-level design
- Semi-formal Demonstration of correspondence
- Modular design
- Search for vulnerabilities and resistance to moderate potential attacks
- Covert-channel analysis

- *High-level of risks product evaluation*

- Modular design and design by successive refinements
- Structured implementation of the TSF
- High-controlled development environment and advanced configuration management
- Systematic search for vulnerabilities and resistance to high potential attacks

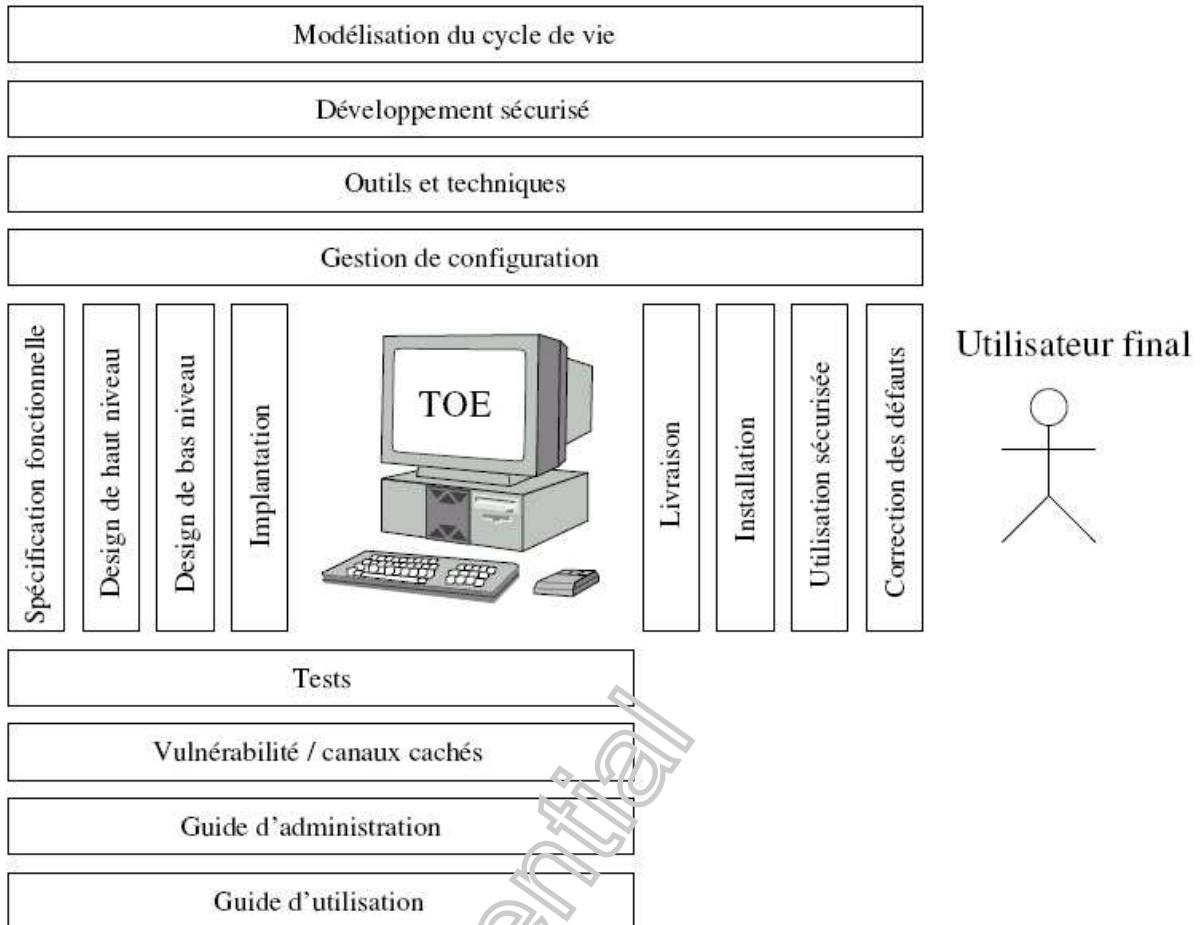
EAL 7

- *Very-high level of risks product evaluation*

- Formal presentation of functional specification and high level design
- White-box developer Comprehensive testing
- Complete and independent confirmation of developer tests results
- Minimisation of design complexity

## Les exigences d'assurance des Critères Communs pour le niveau EAL7

**ST**  
 Biens  
 Menaces -- Hypothèses  
 Politique  
 Objectifs de sécurité  
 Exigences de sécurité  
 Fonctions de sécurité



## Les niveaux de confiance

Classe d'assurance	Famille d'assurance	Composants d'assurance par niveau d'assurance de l'évaluation							niveau retenu pour le paquet EAL2+ de qualification au niveau
		EAL 1	EAL 2	EAL 3	EAL 4	EAL 5	EAL 6	EAL 7	
	ACM AUT				1	1	2	2	
Gestion de configuration	ACM CAP	1	2	3	4	4	5	5	2
	ACM SCP			1	2	3	3	3	
Livraison et exploitation	ADO DEL		1	1	2	2	2	3	1
	ADO_IGS	1	1	1	1	1	1	1	1
Développement	ADV_FSP	1	1	1	2	3	3	4	1
	ADV_HLD		1	2	2	3	4	5	2
Guides	ADV_IMP				1	2	3	3	1
	ADV_INT					1	2	3	
	ADV_LLD				1	1	2	2	1
	ADV_RCR	1	1	1	1	2	2	3	1
	ADV_SPM				1	3	3	3	
	AGD ADM	1	1	1	1	1	1	1	1
Support au Cycle de vie	AGD_USR	1	1	1	1	1	1	1	1
	ALC_DVS			1	1	1	2	2	1
	ALC_FLR								3
	ALC_LCD				1	2	2	3	
	ALC_TAT				1	2	3	3	1
	ATE_COV		1	2	2	2	3	3	1
Tests	ATE_DPT			1	1	2	2	3	
	ATE_FUN		1	1	1	1	2	2	1
	ATE_IND	1	2	2	2	2	3	3	2
	AVA_CCA				1	2	2		
Estimation des vulnérabilités	AVA_MSU			1	2	2	3	3	1
	AVA_SOF		1	1	1	1	1	1	1
	AVA_VLA		1	1	2	3	4	4	2

## • Exemple « Qualification niveau standard »

**EAL2 augmenté des composants d'assurance suivants : ADV\_HLD.2, ADV\_LLD.1, ADV\_IMP.1, ALC\_TAT.1, ALC\_DVS.1, ALC\_FLR.3, AVA\_MSU.1, AVA\_VLA.2**

1: évaluation de la cible  
de sécurité  
ASE\_\*\*\*



2: évaluation de la documentation de développement haut niveau  
ADV\_FSP.1  
ADV\_RCR.1/4  
ADV\_HLD.2  
ADV\_RCR.1/24

3: évaluation de l'environnement de développement  
ACM\_CAP.2  
ADO\_DEL.1  
ALC\_DVS.1  
ALC\_FLR.3

4: évaluation de la documentation de développement de bas niveau des fonctions cryptographiques  
ADV\_LLD.1  
ADV\_RCR.1/34  
ADV\_IMP.1  
ADV\_RCR.1/44  
ALC\_TAT.1

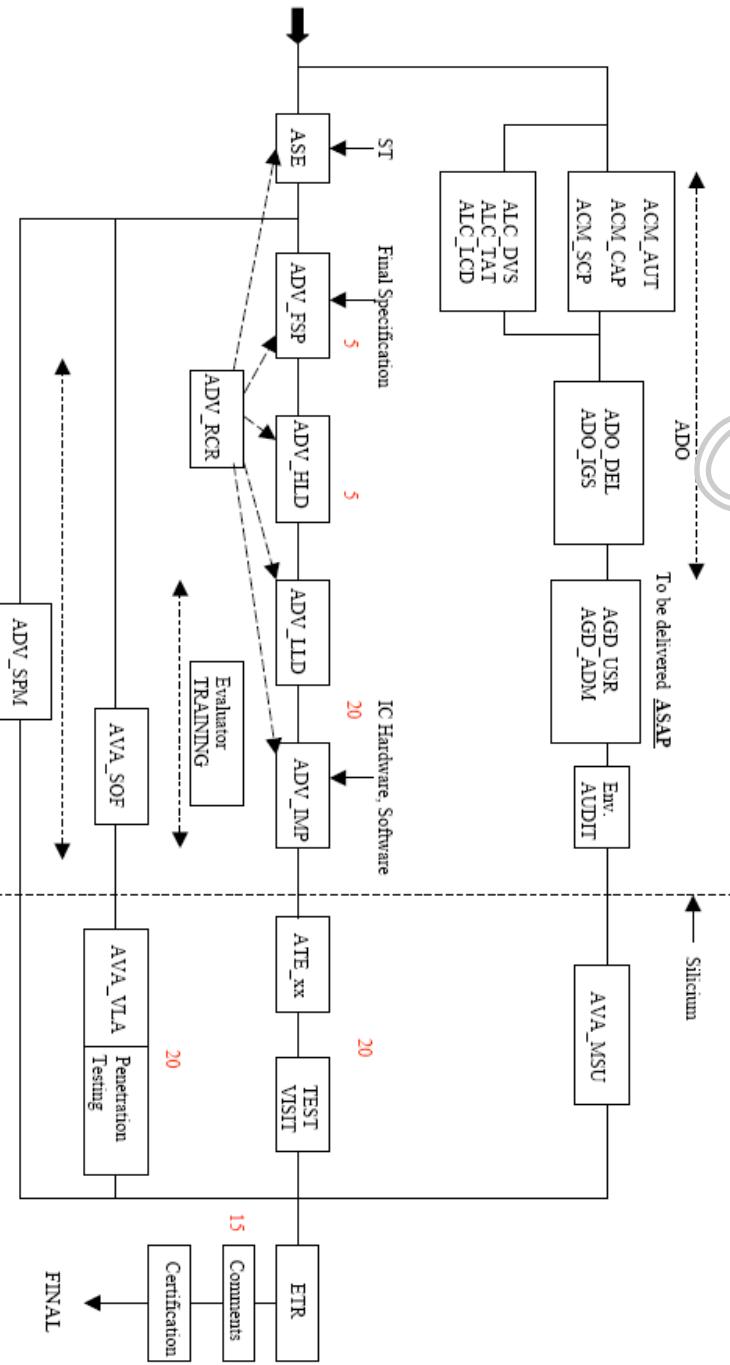
5: évaluation de la documentation de test  
ATE\_COV.1  
ATE\_FUN.1

6: manipulation de la cible d'évaluation  
ADO\_IGS.1  
ATE\_IND.2

7: évaluation des documents d'exploitation  
AGD\_ADMIN.1  
AGD\_USR.1

8: estimation de la vulnérabilité  
AVA\_SOFTWARE.1  
AVA\_VLA.2

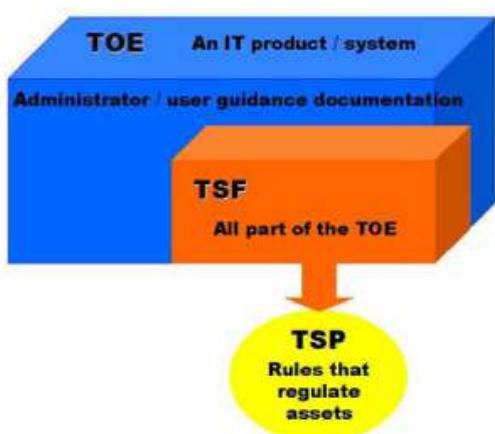
9: finalisation du RTE



## ON SWAPPE SUR MERLE.PDF p24

91

### Résumé



Confidential



92

## Offre de produits certifiés

Systèmes d'exploitation (19) : *Windows NT 4 SP3, Trusted Solaris, AIX, HP-UX, SCO, ...*

Cartes à puces (18) : *Cartes à mémoire, cartes multiapplicatives (Javacard, Multos), ...*

Lecteurs de cartes (28) : *Certification obligatoire en Allemagne*

Firewall (16) : *Netwall, Firewall-1, PIX, Gauntlet, Lucent, ...*

Bases de données (7) : *Oracle, Informix, Ingres, ...*

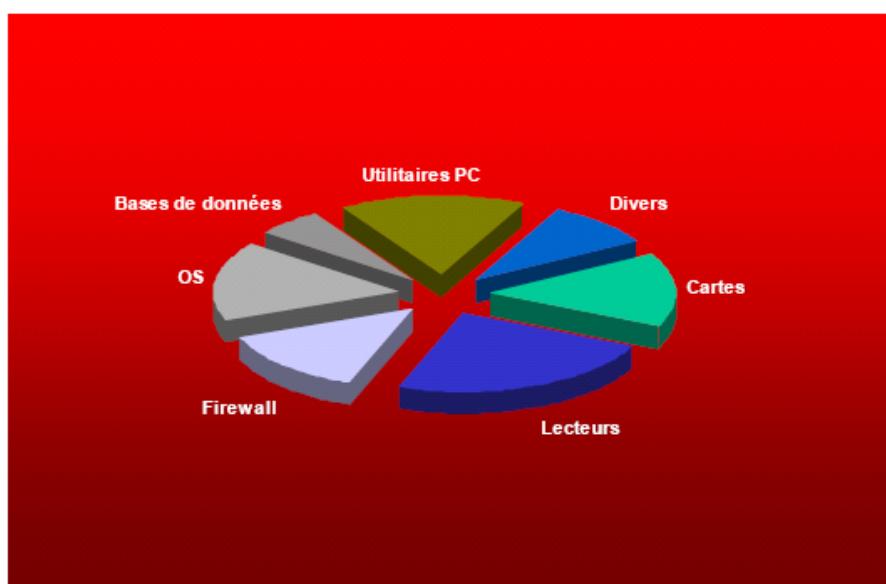
Utilitaires PC (19) : *Contrôle d'accès, utilitaires disques, ...*

Divers (11) : *Chiffrement, PKI, ...*

→ [www.scssi.gouv.fr](http://www.scssi.gouv.fr)

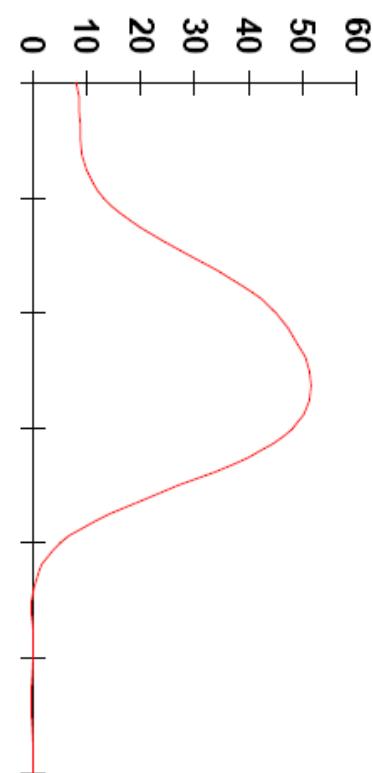
Source : Certificats reconnus par le schéma français au 1er Jan. 2000

## Certificats / Domaines couverts



Source : Certificats reconnus par le schéma français au 1er Jan. 2000

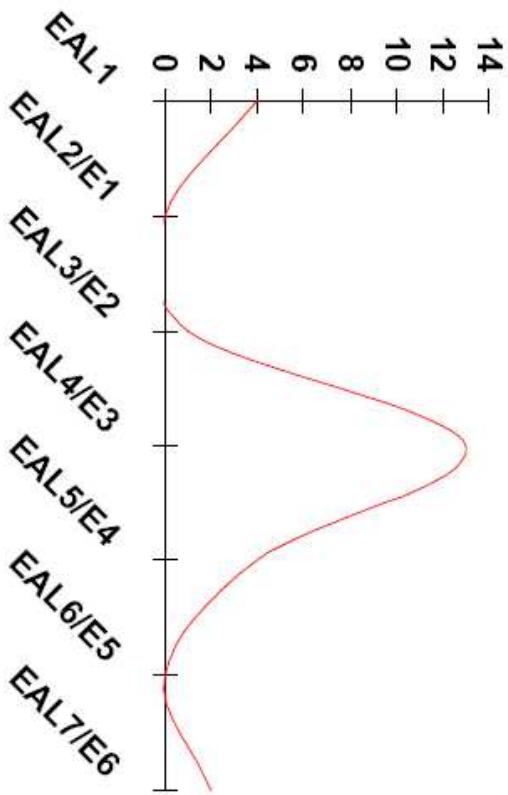
## Certificats : Répartition par niveau d'assurance



Source : Certificats reconnus par le schéma français au 1er Jan. 2000

95

Cartes à puce certifiées



Source : Certificats reconnus par le schéma français au 1er Jan. 2000

96

## Offre de Profils de Protection

Cartes à puces : *Composants, applications, sites de production, ...*

Firewalls : *DGA, NIST/NSA*

Messagerie électronique : *SCSSI, MEFI*

Bases de données : *Oracle*

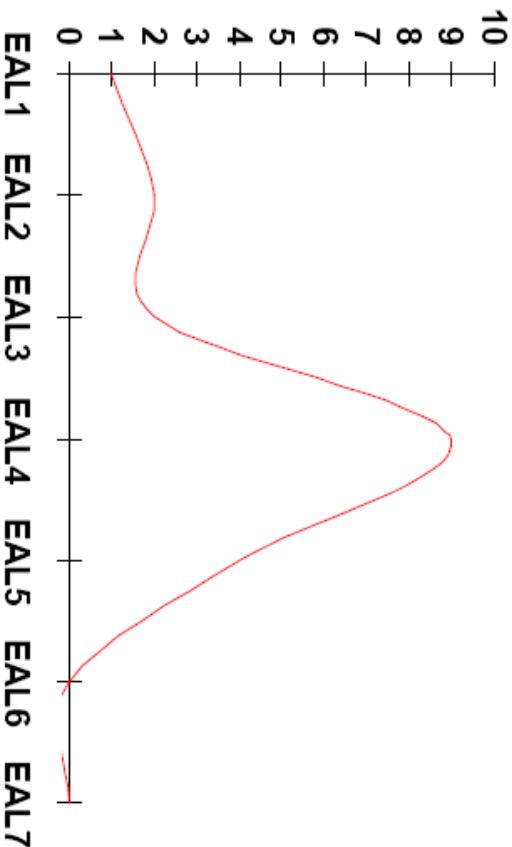
Divers : *Contrôle d'accès, DAB, lecteurs de cartes, ...*

→ [www.scssi.gouv.fr](http://www.scssi.gouv.fr)

Source : Certificats reconnus par le schéma français au 1er Jan. 2000

97

### PP : Répartition par niveau d'assurance



Source : Certificats reconnus par le schéma français au 1er Jan. 2000

98

## Un exercice de style : la rédaction de la Cible de sécurité



⇒ Choix du périmètre : commercial, réaliste ou rigoriste?

– Frontière physique de la TOE

– Hypothèses d'utilisation

- « Les administrateurs sont de confiance et ont été formés »

– Menaces contrées

- « Un individu peut effectuer une attaque en confidentialité sans connaissance des secrets »

⇒ Niveau d'assurance :

– Garder en tête les fondamentaux : EAL2 = Qualité, EAL4 = Sécurité

– Chercher ce qui se cache derrière le « + » (EALx+)

⇒ L'avenir : la conformité aux Profils de Protection (PP)

## Les limites du processus Critères Communs



⇒ La complexité du processus : trop long et trop coûteux

⇒ La difficile maintenance des certificats

- Les procédures de surveillance sont inadaptées aux produits logiciels

- Import de la notion américaine « assurance continuity »?

⇒ La complexité de l'interprétation des résultats d'une évaluation

⇒ L'impossibilité de composer les évaluations CC

⇒ L'impossibilité d'évaluer les systèmes complexes à un niveau décent

## Usage particulier en France des CC

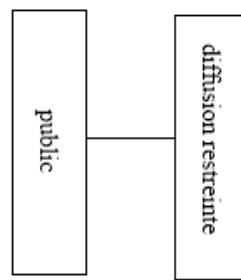
très secret défense

secret défense

confidentiel défense

Objectif : fournir à l'administration et au secteur privé un cadre opérationnel cohérent pour le traitements des informations sensibles et classifiées

Méthode : bâtir un catalogue de solutions qualifiées, sur la base de leur évaluation suivant les CC



Sensibilité de l'information	Niveau de qualification requis	Niveau CC requis
Sensible non classifié	Niveau Standard	EAL2+
Classifié Confidentiel Défense	Niveau Basique	EAL4+

## Critères Communs : les bonnes questions



- Qu : Quelles parties du produit ont été analysées?  
R : La TOE est décrite dans la Cible de Sécurité
- Qu : Quelle confiance accorder aux vérifications effectuées?  
R : Une totale confiance en terme de qualité (agrément COFRAC)
- Qu : Quels sont les documents publics associés aux certifications?  
R : Le rapport de certification et la cible de sécurité
- Qu : Où trouve t-on ces documents?  
R : Sur [www.commoncriteriaportal.org](http://www.commoncriteriaportal.org) ou [www.ssi.gouv.fr](http://www.ssi.gouv.fr)

# Classification des attaques

Les attaques se font par rapport aux propriétés non fonctionnelles :

- Confidentialité :

- L'objectif est d'obtenir des informations.

- Intégrité :

- L'objectif est de changer, supprimer ou ajouter des informations.

- Disponibilité :

- L'objectif est de rendre le dispositif ~~inutilisable~~ par suite de très nombreuses requêtes.

103

## Les types d'attaques

Les attaques physiques :

- Microprobing

- Electromagnétisme,...

Les canaux cachés :

- Observations précises du comportement en mode en fonction du dispositif,

- Les observations peuvent porter sur le temps, la consommation,...

Les attaques par injection de fautes,

Les attaques logicielles :

- Virus, Chevaux de Troie,...

104

## Différentes catégories

### Non invasives

Le produit reste fonctionnel

### Invasives

Le produit est partiellement endommagé

105

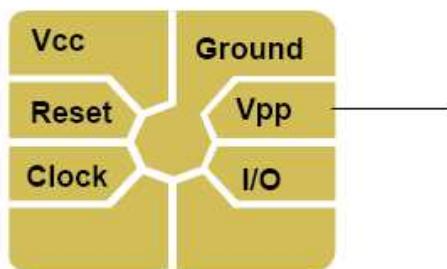
### Non invasives

modification des conditions opérationnelles (Vcc, F)

modification de la température

modification des rayonnements lumineux (UV, rayon X, lumière blanche, IR, ...)

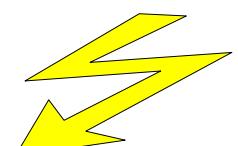
injection de fautes (glitches, rayonnements lumineux)



- Attack on VPP
- Using nail polish
- Card not debited...

(injection de fautes sur la JVM : <http://www.cs.princeton.edu/~sudhakar/papers/>)

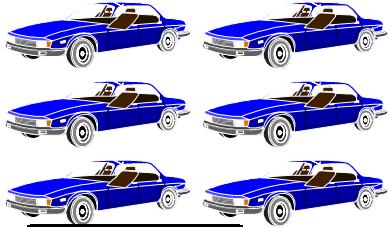
attaques sur les canaux cachés



106

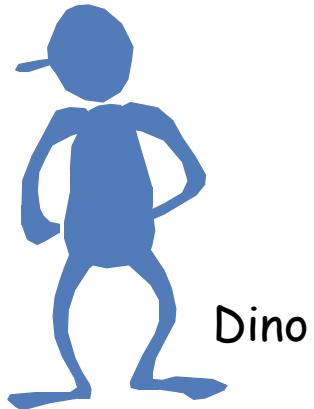
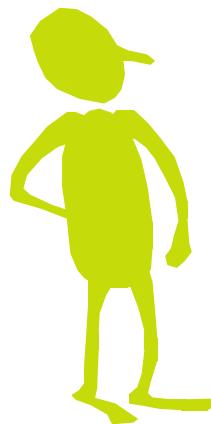
# *Êtes vous prêt ?*

Politique : Les jouets cassés ne sont pas payés

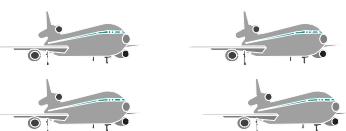


voiture = \$3

Jack



Dino

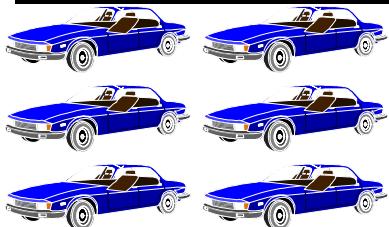


avion = \$5

*Dino achète des jouets à Jack*

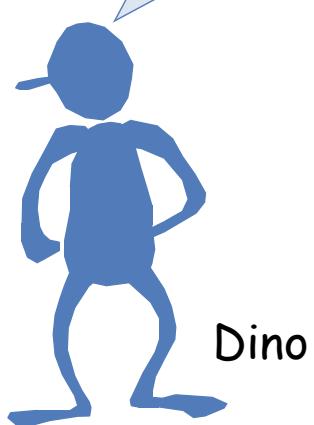
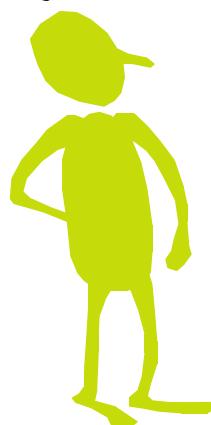
# *Êtes vous prêt ?*

Politique : Les jouets cassés ne sont pas payés

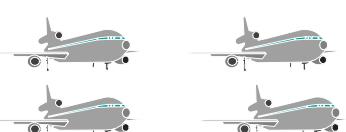


voiture = \$3

Jack



Je voudrais acheter  
3 avions

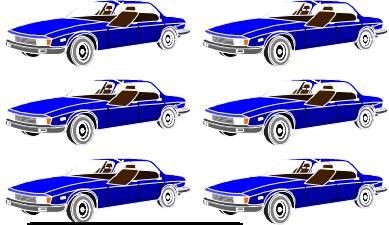


avion = \$5

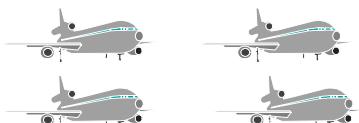
*Dino achète des jouets à Jack*

# Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés



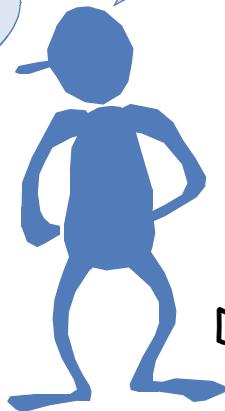
voiture = \$3



avion = \$5



Ça fera \$15

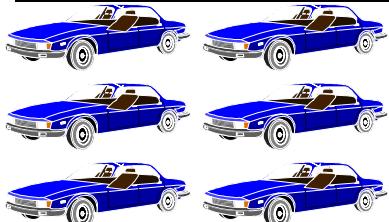


Je voudrais acheter  
3 avions

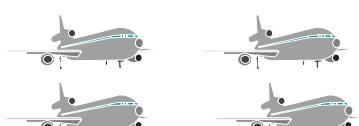
Dino achète des jouets à Jack

# Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés



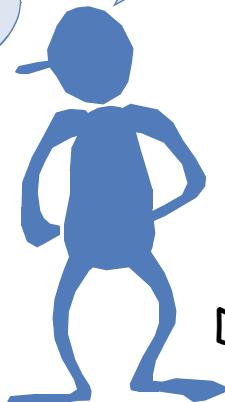
voiture = \$3



avion = \$5



Ça fera \$15

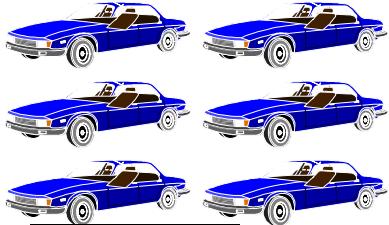


OK, s'il vous plaît  
envoyez le par DHL

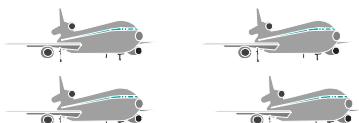
Dino achète des jouets à Jack

# Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés



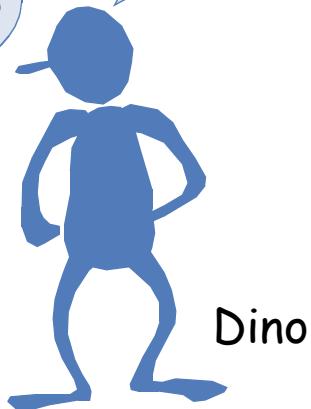
voiture = \$3



avion = \$5



Comment payez vous ?



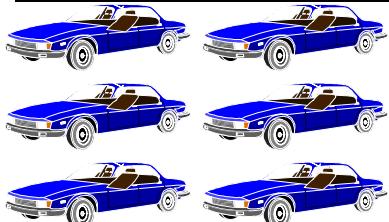
Dino

OK, s'il vous plaît  
envoyez le par DHL

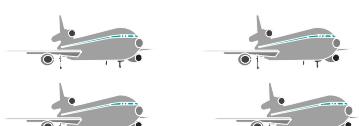
Dino achète des jouets à Jack

# Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés



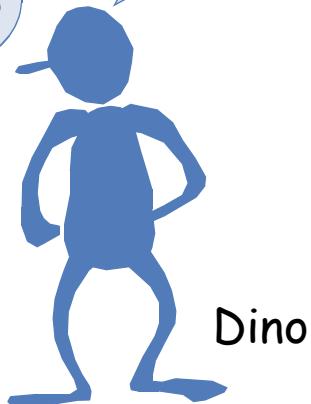
voiture = \$3



avion = \$5



Comment payez vous ?



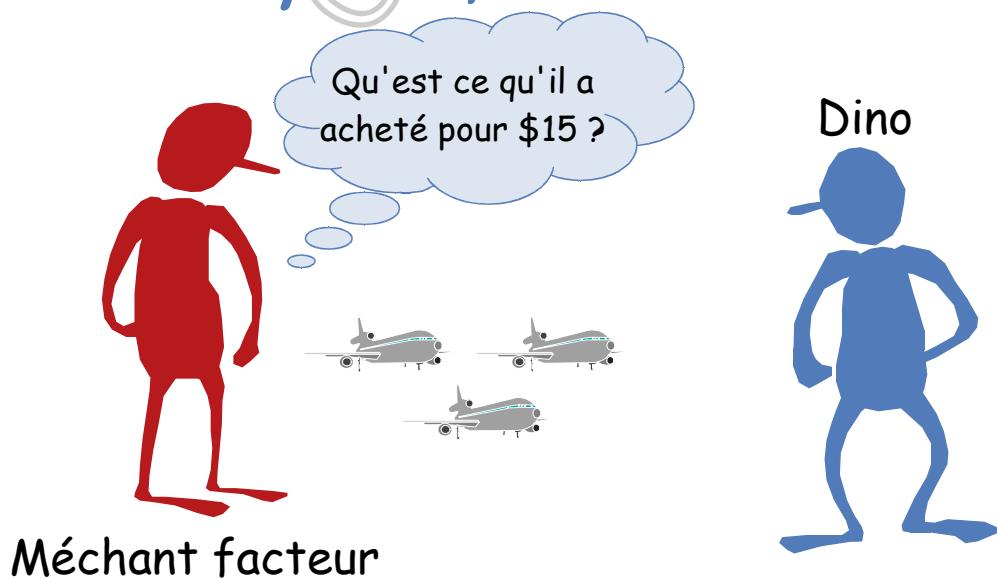
J'enverrai \$15  
en mandat postal

Dino achète des jouets à Jack

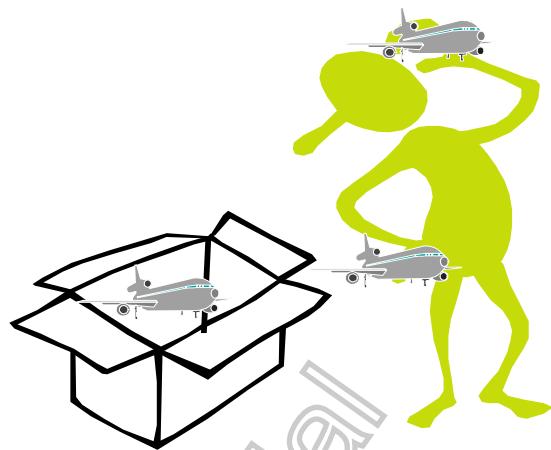
*Le facteur veux savoir ce que Dino a acheté pour \$15*



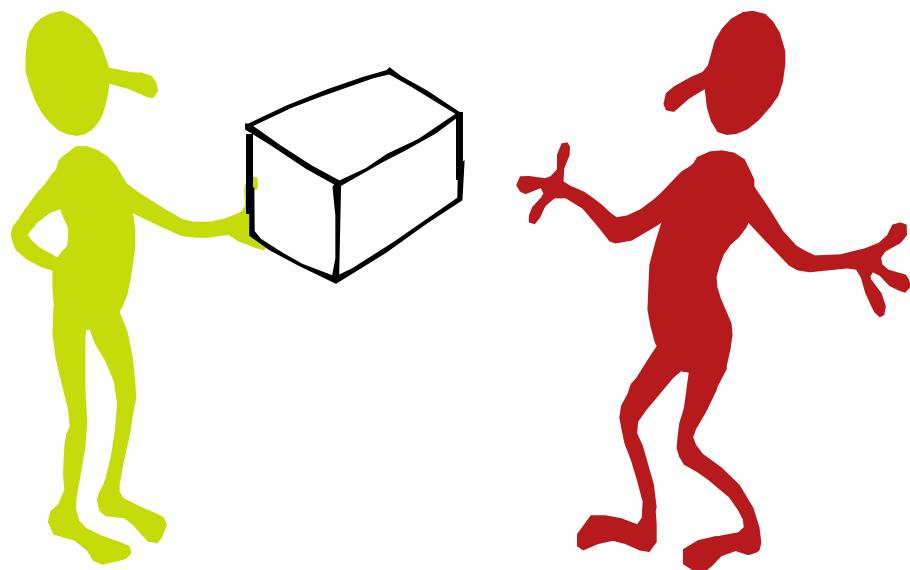
*Le facteur veux savoir ce que Dino a acheté pour \$15*



*Pendant ce temps, Jack prépare le colis DHL*



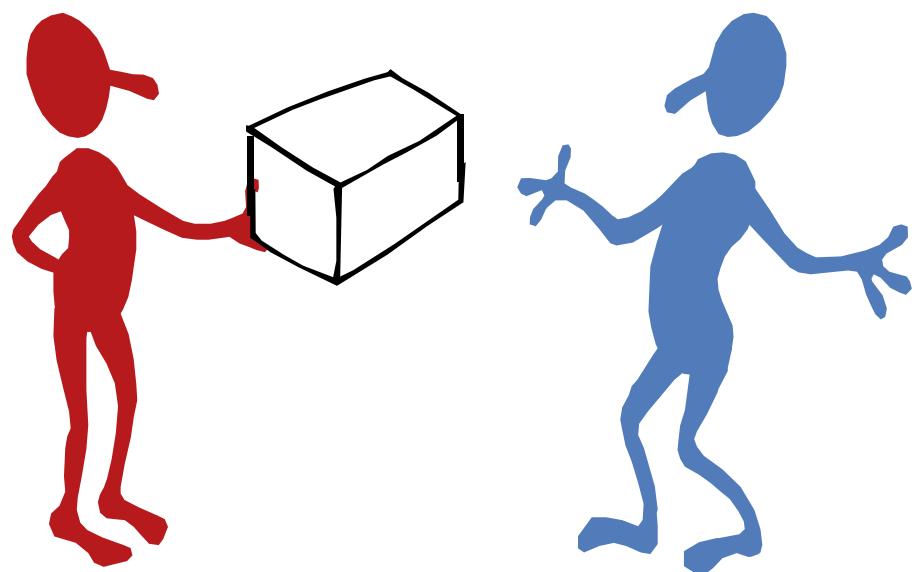
*Et le donne au facteur*



*Qui tape dedans assez fort pour  
casser un jouet*



*Et le donne à Dino*



Confidential

*Une semaine plus tard, il surveille  
le mandat postal de Dino...*



La leçon apprise : **Les attaques en faute** peuvent aussi permettre d'extraire des secrets de tokens!

Les fautes matérielles peuvent être venir de diverses sources :  
Glitches de tension, faisceau lumineux, faisceau laser ...

Utilisable sur la **signature RSA**, le chiffrement DES, ...  
Il y a des détecteurs lasers embarqués sur les puces récentes !  
(voir thèse d'Alexandre SARAFIANOS)

#### Attaques software/hardware

Pour forger des pointeurs et ensuite ...



## Les canaux cachés

### le temps d'exécution

=> nombre de cycle d'une instruction ou d'un algorithme.

### la consommation de courant

=> Les modifications rapides de la tension et de l'intensité du courant au sein du même composant sont à la base des émissions du circuit car ils conduisent des courants RF à l'intérieur et à l'extérieur du chip.

### les émissions électromagnétiques

=> Les courants RF entraînent un rayonnement électromagnétique.

121

## Le matériel

un oscilloscope numérique,

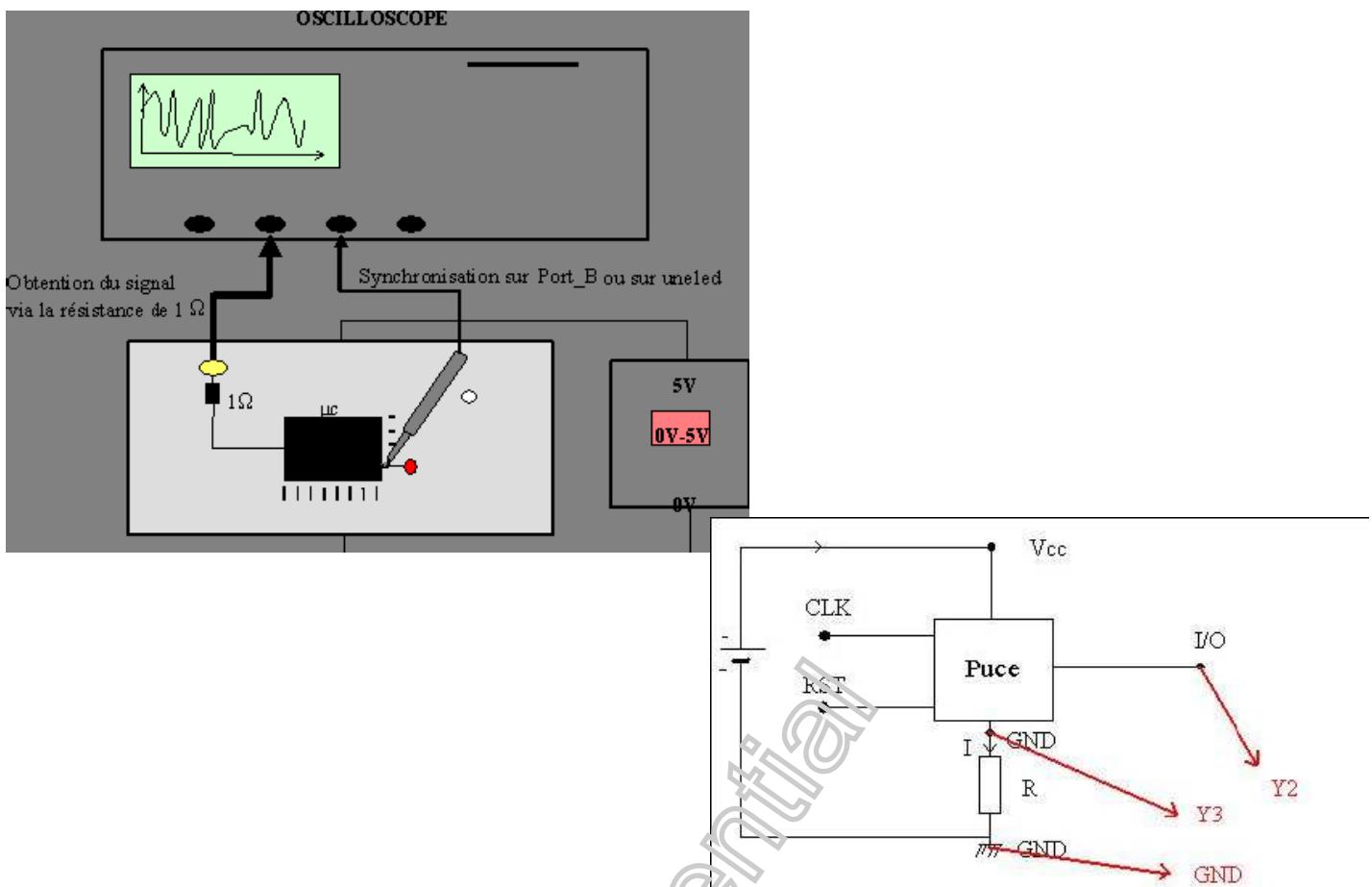
un lecteur de carte à puce,

un pc équipé de cartes d'acquisition et de logiciels mathématique pour le traitement des données,

une sonde CEM si on veut étudier les émissions électromagnétiques.



## Le montage pour suivre la consommation en courant



## La “timing attack”

Cette attaque consiste à mesurer le temps d'exécution d'un algorithme.

=> Révèle des informations sur les opérations et/ou les opérandes.

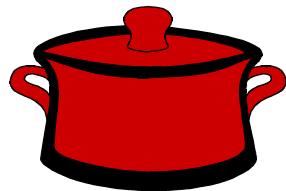
Nécessite souvent :

un grand nombre d'exécution à messages choisis,  
un traitement statistique des résultats obtenus.

Exemple:

## *Un jeu*

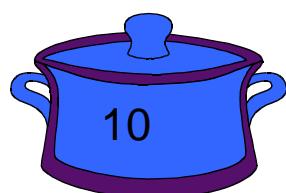
- Mettez 28 dans un des pots et 10 dans l'autre :



## *Un jeu*

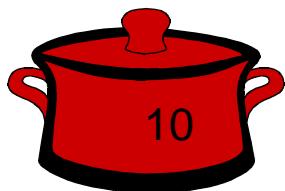
Confidential

- Mettez 28 dans un des pots et 10 dans l'autre :



## *Un jeu*

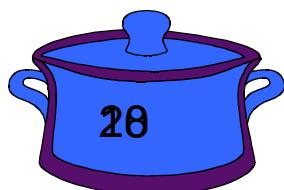
- Mettez 28 dans un des pots et 10 dans l'autre :



## *Un jeu*

Confidential

- Mettez 28 dans un des pots et 10 dans l'autre :



- Je vous demande de multiplier le contenu du pot bleu par 10 et le contenu du pot rouge par 7, d'additionner les deux résultats et de me dire si la somme est paire ou impaire.
- Est-ce que votre réponse est suffisante pour révéler le contenu de chaque pot ?

# *Est-ce que ce jeu à un sens ?*

## *Est-ce que ce jeu à un sens ?*

- Et bien, normalement non :

$$28 \times 7 + 10 \times 10 = 296 \quad \text{est un nombre pair}$$

et

$$10 \times 7 + 28 \times 10 = 350 \quad \text{est aussi un nombre pair...}$$

- Pourtant, juste en observant le temps pris pour donner la réponse (le calcul mental conduisant à 296 est plus compliqué que celui conduisant à 350), on peut dire quelle valeur était dans quel pot !

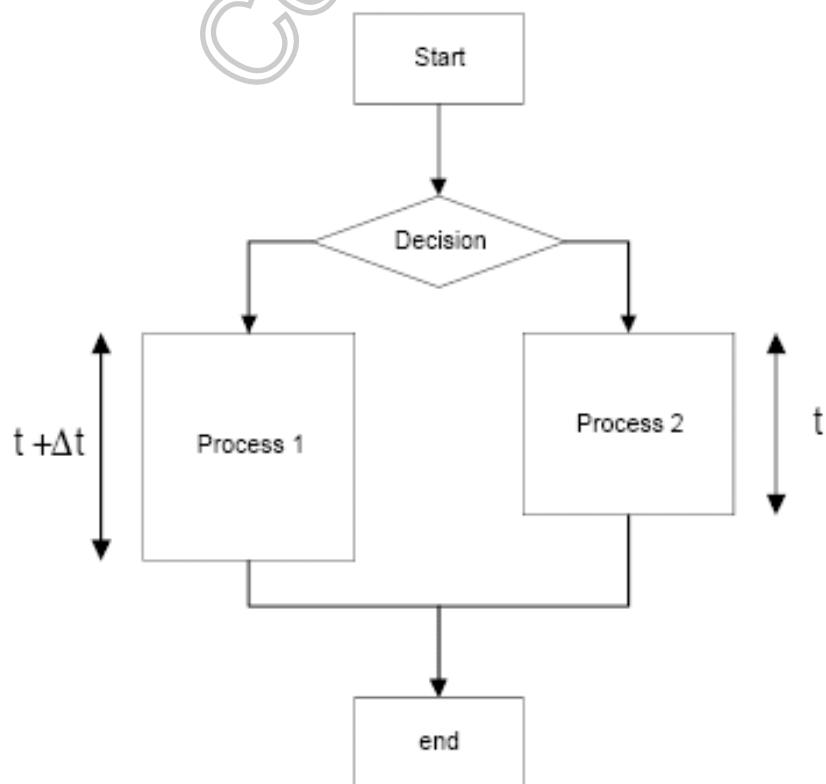
## Conclusion

Une observation externe du temps de calcul d'une carte peut conduire à la fuite d'informations secrètes vers l'extérieur (par exemple des clés, des PINs, etc.).

**Les timing attacks** sont apparues au début des années 1990.

La leçon apprise : **Les logiciels actuels pour cartes s'exécute en temps constant.**

La “timing attack”

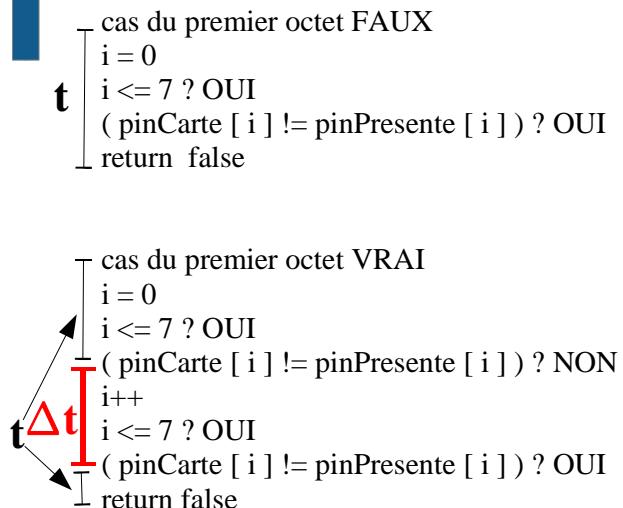


## La “timing attack” sur le PIN (1/2)

```
for ( i = 0 ; i <= 7; i++)
    if ( pinCarte [ i ] != pinPresente [ i ] )
        return false ;
return true ;
```



Présentons un PIN quelconque et déroulons le programme :



pinPresente  
0,0,0,0,0,0,0,0

pinPresente  
1,0,0,0,0,0,0,0

## La “timing attack” sur le PIN (2/2)

```
for ( i = 0 ; i <= 7; i++)
    if ( pinCarte [ i ] != pinPresente [ i ] )
        return false ;
return true
```



Présenter les n valeurs possibles de pinPresente[0] (256 valeurs) (n,0,0,0,0,0,0).

Mesurer la durée d'exécution de la commande T pour les n valeurs.

Calculer T[n0] le maximum des T

- T[n0] = max(T[n]); n = 0; ... ; 255

n0 est la solution pour pinCarte[0]

Itérer sur tous les pinPresente[i]

Nombre d'essais :  $8 * 256 = 2048$  (contre  $256^8$  en force brute)

pinPresente  
0,0,0,0,0,0,0,0  
1,0,0,0,0,0,0,0

...  
255,0,0,0,0,0,0,0

On peut même s'arrêter avant n=255 !

## Exemples de protection (1/2)

```
alea = random ( 0 , 7 ) ; // alea situe dans l'intervalle [0 ; 7]
for ( i = 0 ; i <= 7; i++) {
    octet = ( alea + i ) mod 8 ;
    if ( pinCarte [ octet ] != pinPresente [ octet ] )
        return false ;
}
return true ;
```

Nombre d'essais en moyenne :  $8 * 2048$  (contre  $256^8$  en force brute)  
Possibilité d'attaques de type MasterMind.

135

## Exemples de protection (2/2)

```
boolean test = true ;
for ( i = 0 ; i <= 7; i++) {
    if ( pinCarte [ i ] != pinPresente [ i ] )
        test = test && false ;
    else
        test = test && true ;
}
return test ;
```

équivaut à :

```
boolean test = true ;
for ( i = 0 ; i <= 7; i++)
    test = ( pinCarte [ i ] == pinPresente [ i ] ) && test ;
return test ;
```

**Et pas** test = test && ( pinCarte [ i ] == pinPresente [ i ] );

**Des attaques en temps existent sur d'autres supports (e.g. PC).**  
Voir par exemple : *Cache-timing attacks on AES* ou *CACHE MISSING FOR FUN AND PROFIT*

136

Et sinon vous savez codé un vérification de PIN sécurisé maintenant ?

137

## And what about security ?

- Write **securely** a function that checks that an array is equal to another with less than 3 trials.

```
boolean verify (byte[] buffer, short ofs, byte  
len)  
{...}
```

- Need a constant `maxTries` initialized at 3,
- A field which memorize the trial number, says `triesLeft`

```
boolean verify (byte[] buffer, short ofs, byte len)
{
    // No comparison if PIN is blocked
    if (triesLeft <= 0)
        return false;

    // Main comparison
    for(short i=0; i < len; i++)
        if (buffer[ofs+i] != pin[i])
    {
        triesLeft-- ;
        authenticated[0] = false ;
        return false ;
    }

    // Comparison is successful
    triesLeft = maxTries ;
    authenticated[0] = true ;
    return true ;
}
```

Check len  
before ;-)

```
boolean verify (byte[] buffer, short ofs, byte len)
{
    // No comparison if PIN is blocked
    if (triesLeft <= 0)
        return false ;

    // First decrements the number of remaining tries
    triesLeft-- ;

    // Main comparison
    boolean equal = true ;
    for(short i=0; i < len; i++)
        equal = (equal && (buffer[ofs+i] != pin[i])) ;//  
cst time

    if (!equal) {
        // Comparison failed
        authenticated[0] = false ;
        return false ;
    }
    else {
        // Comparison is successful
        triesLeft = maxTries ;
    }
}
```

```
boolean verify (byte[] buffer, short ofs, byte len)
{
    // No comparison if PIN is blocked
    if (triesLeft <= 0)
        return false ;

    // First decrements the number of remaining tries
    triesLeft-- ;

    // Main comparison
    boolean equal = true ;
    for(short i=0; i < len; i++)
        equal = (equal && (buffer[ofs+i] != pin[i])) ;//  
cst time
```

No!

```
if (!equal) {
    // Comparison failed
    authenticated[0] = false ;
    return false ;
}
else {
    // Comparison is successful
    triesLeft = maxTries ;
```

boolean verify (byte[] buffer, short ofs, byte len)
{
 // No comparison if PIN is blocked
 if (triesLeft <= 0)
 return false ;

 // First decrements the number of remaining tries
 triesLeft-- ;

```
    // Main comparison
    boolean equal = true ;
    for(short i=0; i < len; i++)
        equal = ((buffer[ofs+i] != pin[i]) && equal) ;//  
cst time
```

Yes!

```
if (!equal) {
    // Comparison failed
    authenticated[0] = false ;
    return false ;
}
else {
    // Comparison is successful
    triesLeft = maxTries ;
```

```
public final static short BOOL_TRUE = (short)0x5a5a ;
public final static short BOOL_FALSE = (short)0xa5a5 ;
short equal = BOOL_TRUE ;

...
// Main comparison
for(short i=0; i < len; i++)
    equal = (short)(equal &
                    ((buffer[ofs+i] != pin[i])? BOOL_FALSE: BOOL_TRUE));

if (equal == BOOL_TRUE) {
    // Comparison is successful
    triesLeft = maxTries ;
    authenticated[0] = true ;
    return true ;
}
else {
    ...
}
```

// First checks the integrity of the variable  
if (triesLeft != triesLeftBackup)  
 takeCountermeasure() ;  
  
// No comparison if PIN is blocked  
if (triesLeft < 0)
 return false ;

•Correct ?

```

// First checks the integrity of the variable
if (triesLeft != triesLeftBackup)
    takeCountermeasure() ;

// No comparison if PIN is blocked
if (triesLeft < 0)
    return false ;

```

- It protects only against a writing between the last and the current one. If the attack is during the evaluation...

- RAM is safer than EEPROM

```

// Transfer in a local and check the integrity of the
variable
byte tl = triesLeft ;

```

```

if (tl != triesLeftBackup)
    takeCountermeasure() ;

```

```

// No comparison if PIN is blocked
if (tl < 0)
    return false ;

```

```

boolean verify (byte[] buffer, short ofs, byte len) {
    byte tl = triesLeft ;
    if (tl != (short)(~triesLeftBackup)) takeCountermeasure() ;
    if (tl < 0) return false;
    JCSYSTEM.beginTransaction() ;
    triesLeft = --tl ;
    triesLeftBackup++ ;
    JCSYSTEM.commitTransaction() ;
    if (triesLeft != (short)(~triesLeftBackup)) takeCountermeasure() ;
    short equal = BOOL_TRUE ;
    for(short i=0; i < len; i++)
        equal = (short)(equal &
                        ((buffer[ofs+i]!=pin[i])?BOOL_FALSE:BOOL_TRUE)) ;

    if (equal == BOOL_TRUE) {
        JCSYSTEM.beginTransaction() ;
        triesLeft = maxTries ;
        triesLeftBackup = (byte)(~maxTries) ;
        JCSYSTEM.commitTransaction() ;
        // Verifies the new value
        if (triesLeft!=(short)(~triesLeftBackup))
            takeCountermeasure() ;
        authenticated[0] = true ;
        return true ;
    } else {
        authenticated[0] = false ;
        return false ;
    }
}

```

```
boolean verify(byte[] buffer, short ofs, byte len)
{
    // Initializes the step counter
    short stepCounter = INITIAL_COUNTER ;

    // First checks the integrity of the variable
    byte tl = triesLeft ;
    stepCounter++ ;
    if (tl != (short)(~triesLeftBackup)) takeCountermeasure() ;
    stepCounter++ ;
    // No comparison if PIN is blocked
    if (tl < 0) return false ;
    stepCounter++ ;
    // First decrements the number of remaining tries
    JCSSystem.beginTransaction() ;
    triesLeft = --tl ;
    stepCounter++ ;
    triesLeftBackup++ ;
    JCSSystem.commitTransaction() ;
    stepCounter++ ;
    // Verifies the new value
    if (triesLeft != (short)(~triesLeftBackup)) takeCountermeasure() ;

    stepCounter++ ;
    short equal = BOOL_TRUE ; // Main comparison
    stepCounter++ ;
    for(short i=0;i<len;i++)
        equal = (short)(equal &
                        ((buffer[ofs+i]!=pin[i])?BOOL_FALSE:BOOL_TRUE));
    stepCounter++ ;
    if (equal == BOOL_TRUE) { // Comparison is successful
        //Reset the tries to the max
        stepCounter++ ;
        JCSSystem.beginTransaction() ;
        triesLeft = maxTries ;
        triesLeftBackup = (byte)(~maxTries);
        JCSSystem.commitTransaction() ;
        stepCounter++ ;
        if (triesLeft!=(short)(~triesLeftBackup))
            takeCountermeasure() ;
        stepCounter++ ;
        authenticated[0] = true ;
        if (stepCounter == (short)(INITIAL_VALUE+11) )
            return true ;
    } else { // Comparison failed
        stepCounter++ ;
        authenticated[0] = false ;
        if (stepCounter == (short)(INITIAL_VALUE+9) )
            return false ;
    }
    takeCountermeasure() ; // Should have returned at this point
}
```

## Other well known rules ...

```
if (pin.isValidated() == true)
{
    // sensitive operations
}
else
{
    ISOException.throwIt((short) (0x6300+pin.getTriesRemaining()));
}
```

## Other well known rules ...

```
if (pin.isValidated() == true)
{
    if (pin.isValidated() == false)
    {
        ISOException.throwIt((short) (0x6300+pin.getTriesRemaining()));
    }
    else
    {
        // sensitive operations
    }
}
else
{
    ISOException.throwIt((short) (0x6300+pin.getTriesRemaining()));
}
```

- Références :

<http://javacard.vetilles.com/2008/05/15/jc101-12c-defending-against-attacks/>

Ronald De Keulenaer, Jonas Maebe, Koen De Bosschere, Bjorn De Sutter. Link-time smart card code hardening. International Journal of Information Security, 2015

## La consommation de courant

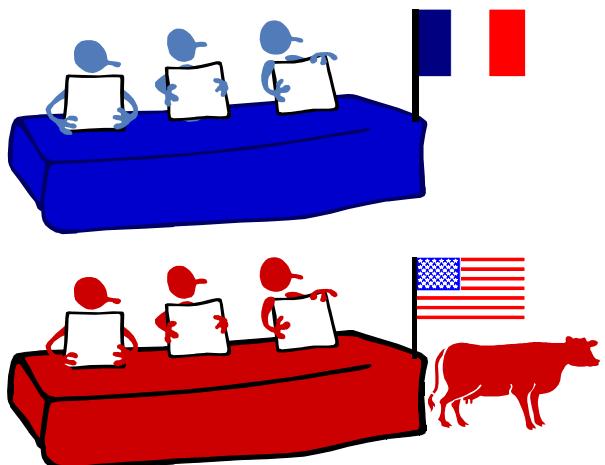
Elle est surtout utilisée dans le domaine de la cryptographie.

### Il existe différentes attaques :

- la SPA (Simple Power Analysis)
- la DPA (Differential Power Analysis)
- la HODPA (High Order Differential Power Analysis)

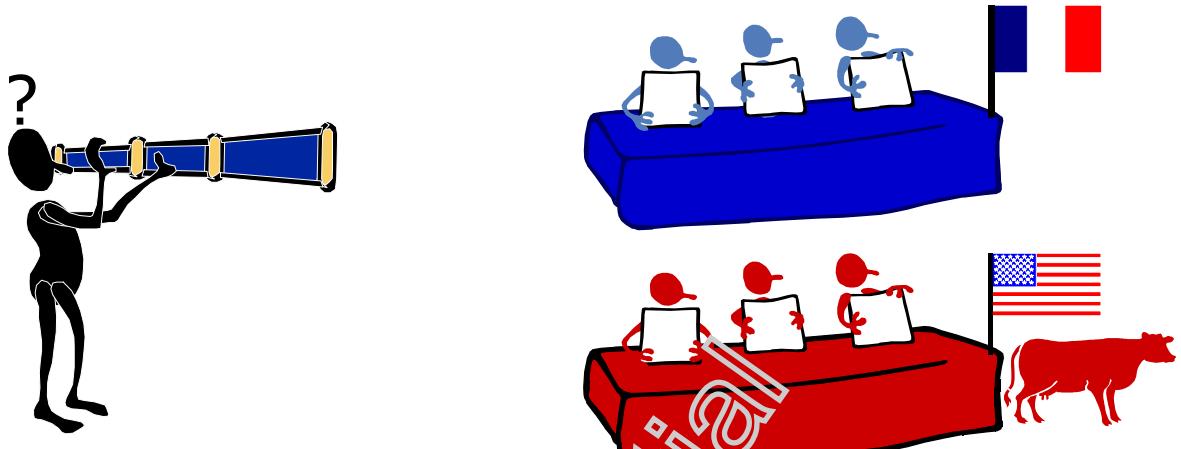
## IMPORTATION DU BOEUF ?

- Seattle, 1999.
- Les représentants Français et Américains négocient sous quelles conditions le boeuf pourrait être importé en France.



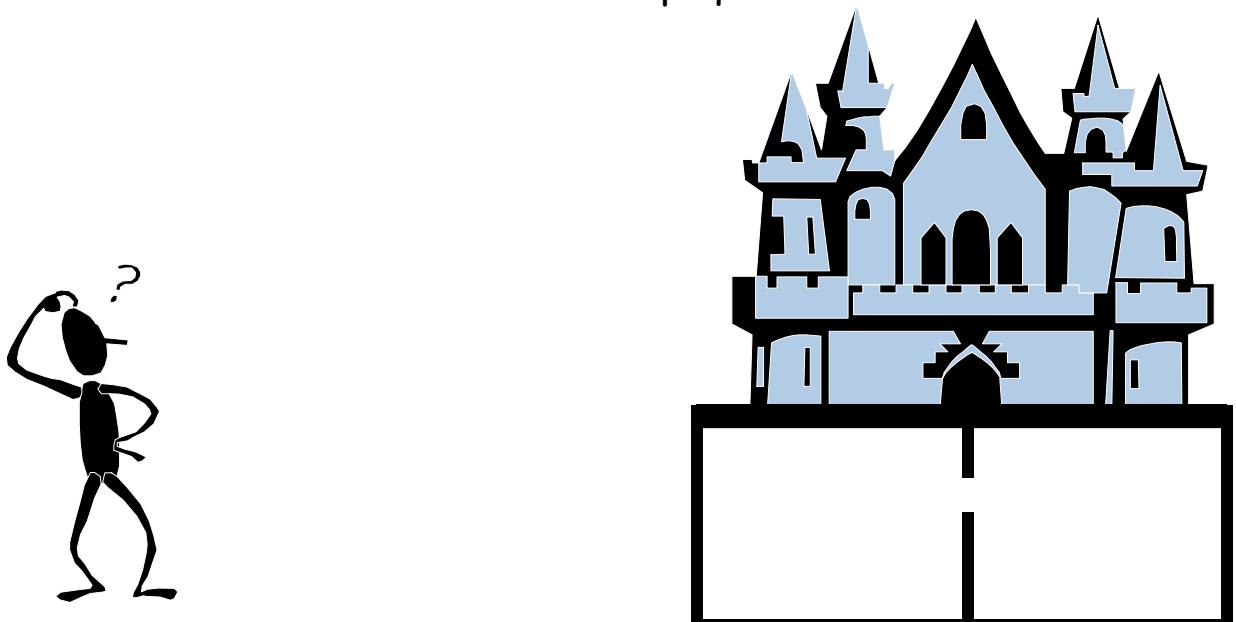
## *IMPORTATION DU BOEUF ?*

- Seattle, 1999.
- Les représentants Français et Américains négocie sous quelles conditions le boeuf pourrait être importé en France.
- «The Sun» envoie un journaliste pour enquêter :



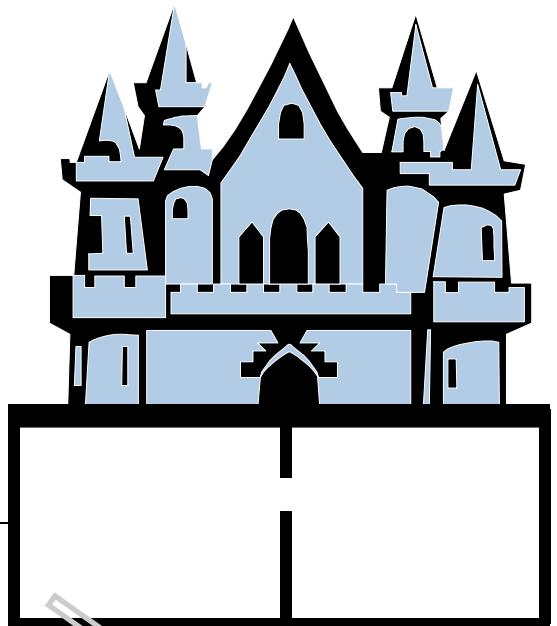
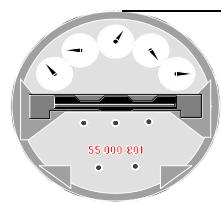
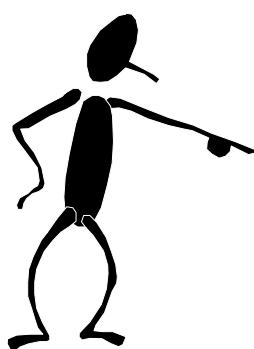
## *IMPORTATION DU BOEUF ?*

- Mais il y a un problème technique : les négociations se déroule dans un hotel aux vitres opaques



## POWER ATTACKS

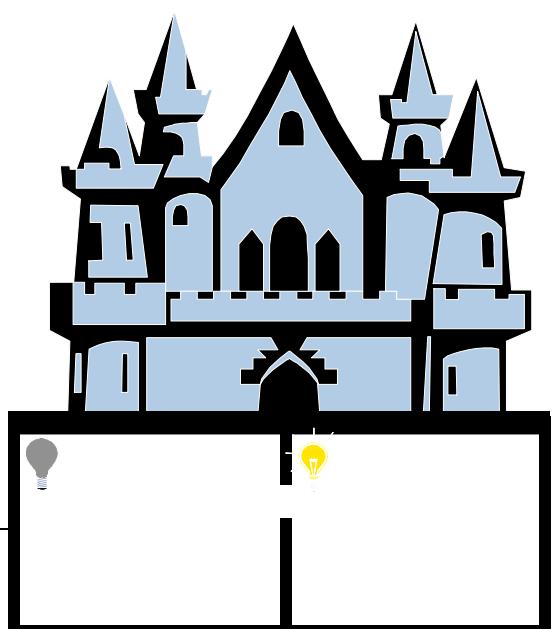
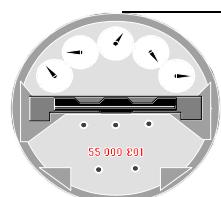
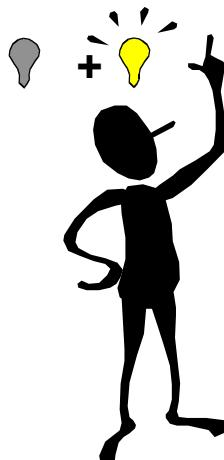
- Idée : Regarder le compteur électrique de l'hôtel !



## POWER ATTACKS

- Le disque tourne lentement:

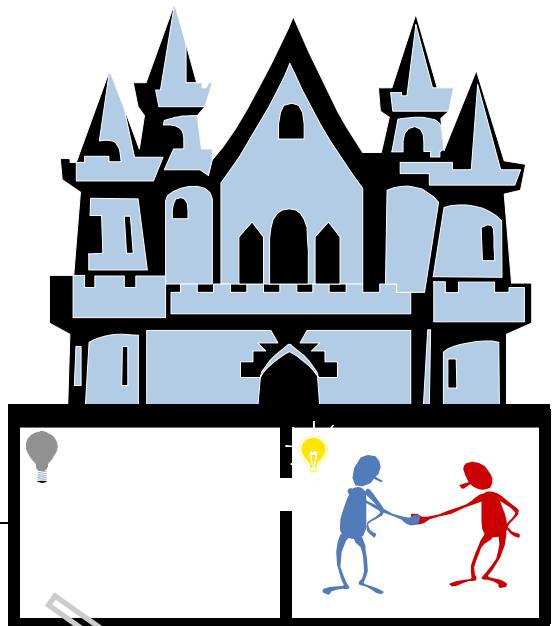
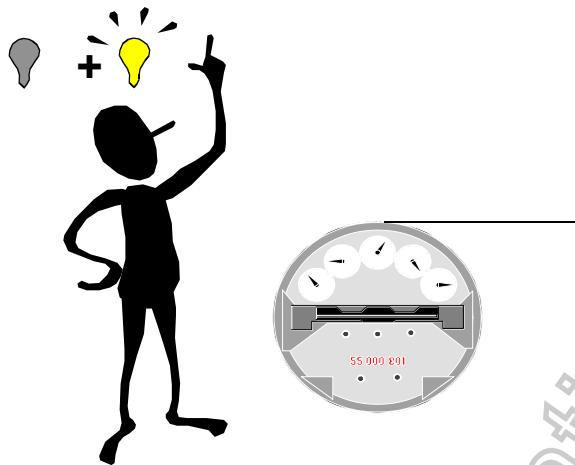
*DEAL CONCLUDED*



## POWER ATTACKS

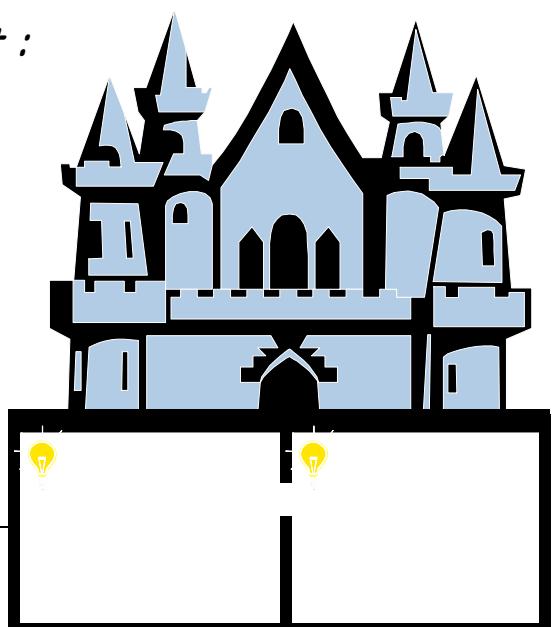
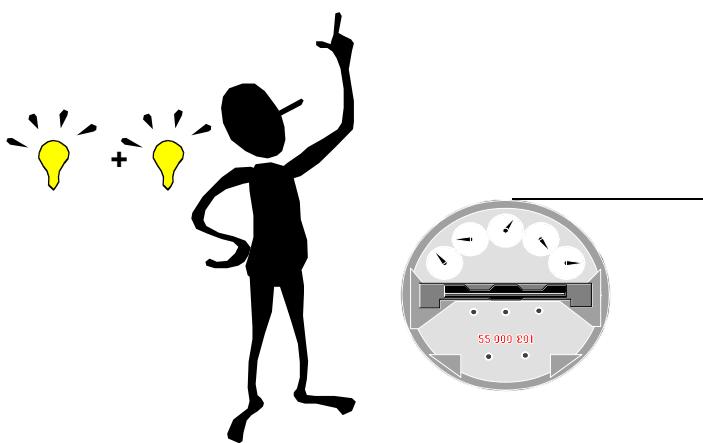
- Le disque tourne lentement:

*DEAL CONCLUDED*



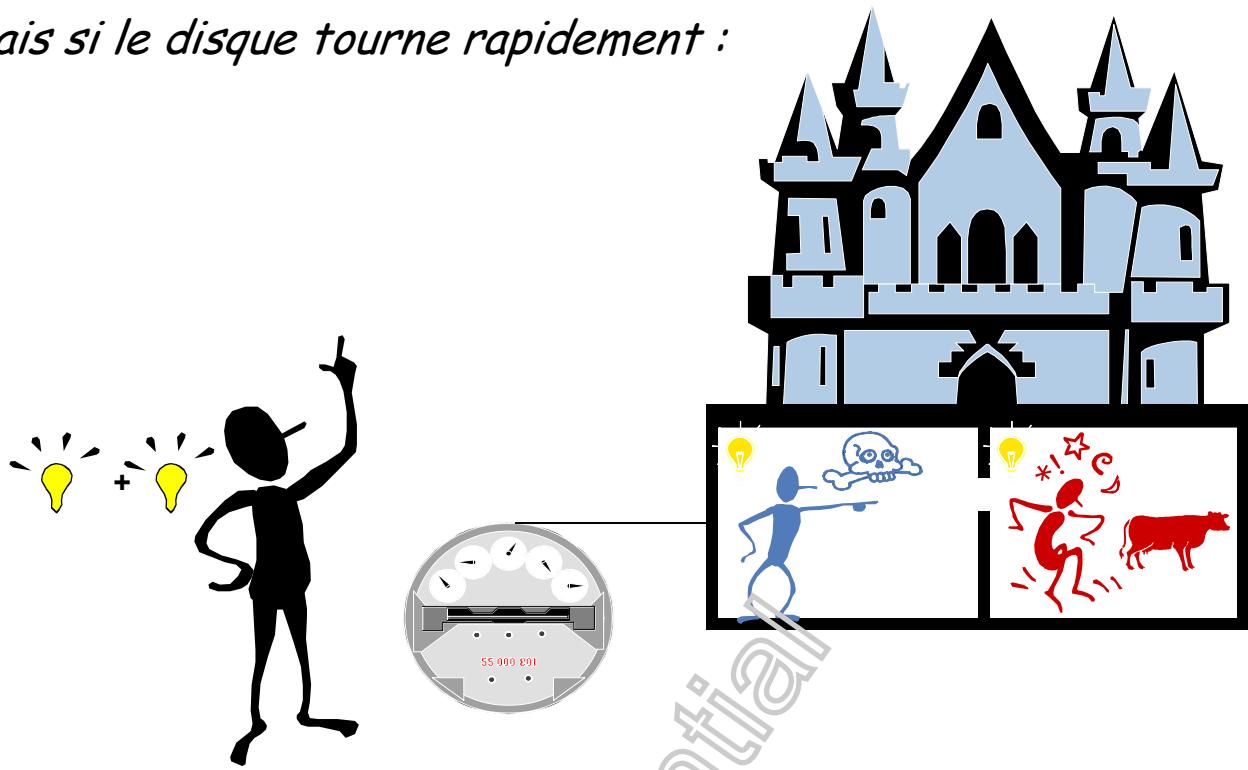
## POWER ATTACKS

- Mais si le disque tourne rapidement :



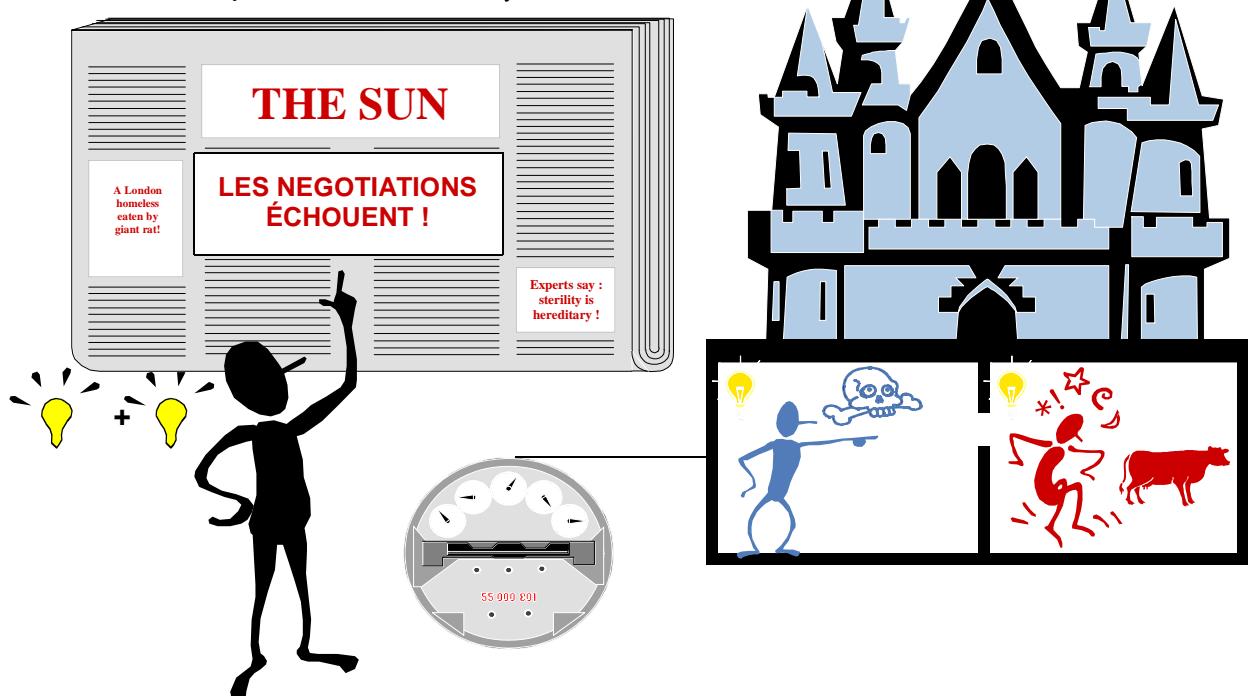
## POWER ATTACKS

- Mais si le disque tourne rapidement :



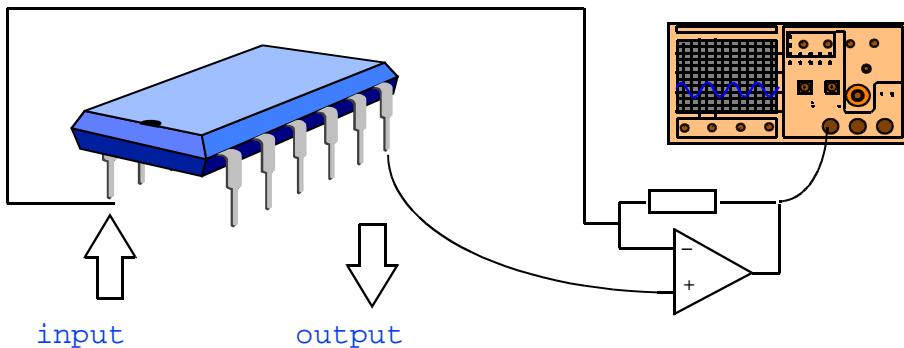
## POWER ATTACKS

- Mais si le disque tourne rapidement :



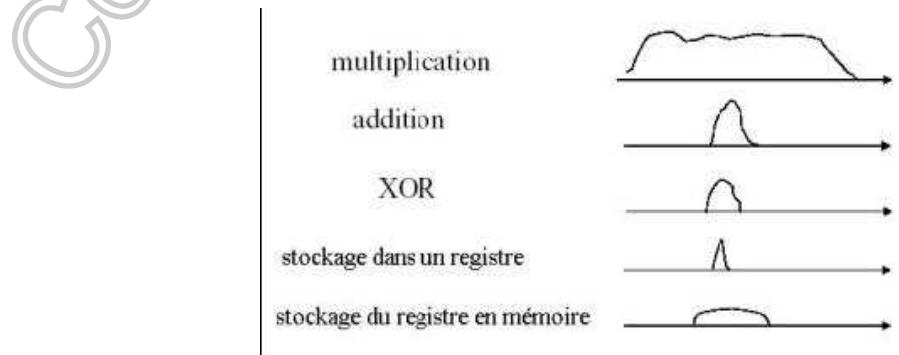
# CONCLUSION

The card's current consumption may reveal secret information.

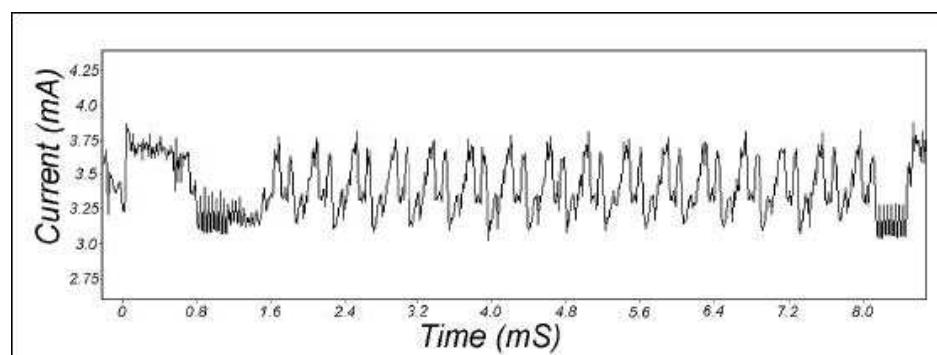


## La SPA (Simple Power Analysis)

Principe : Des instructions différentes ont une trace différente.



Consommation en courant d'un DES. On peut voir la permutation initiale, suivie des 16 tours.

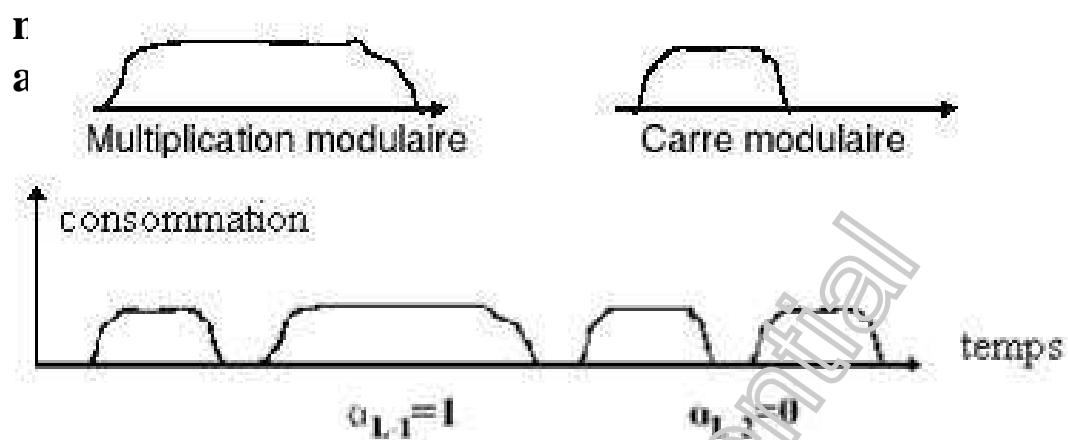


## La SPA sur la signature RSA

```
s = 1 ;  
for ( i = L - 1 ; i >= 0; i--) {  
    s = s*s mod n ;  
    if ( a [ i ] == 1)  
        s = s*y mod n ;  
}
```

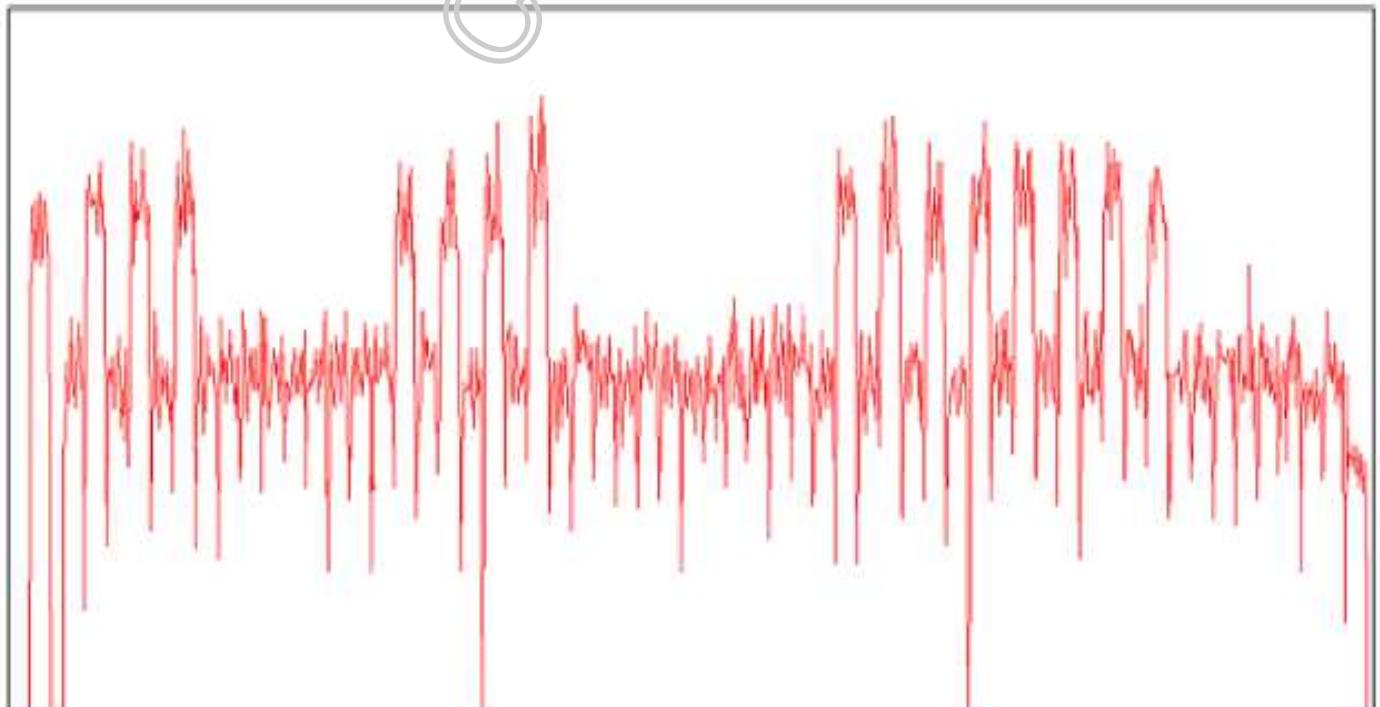
## Signature RSA : $y^a \text{ mod } n$

y est le message à signer



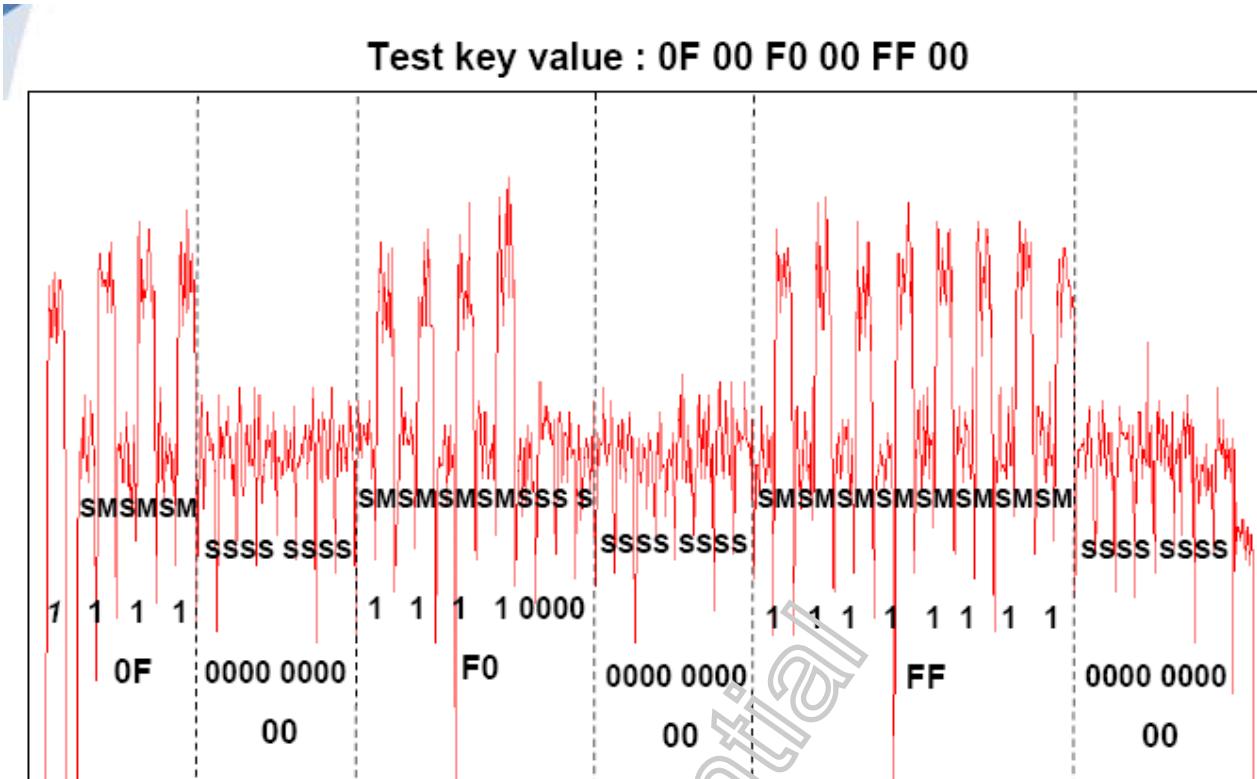
163

## Exemple « réel »



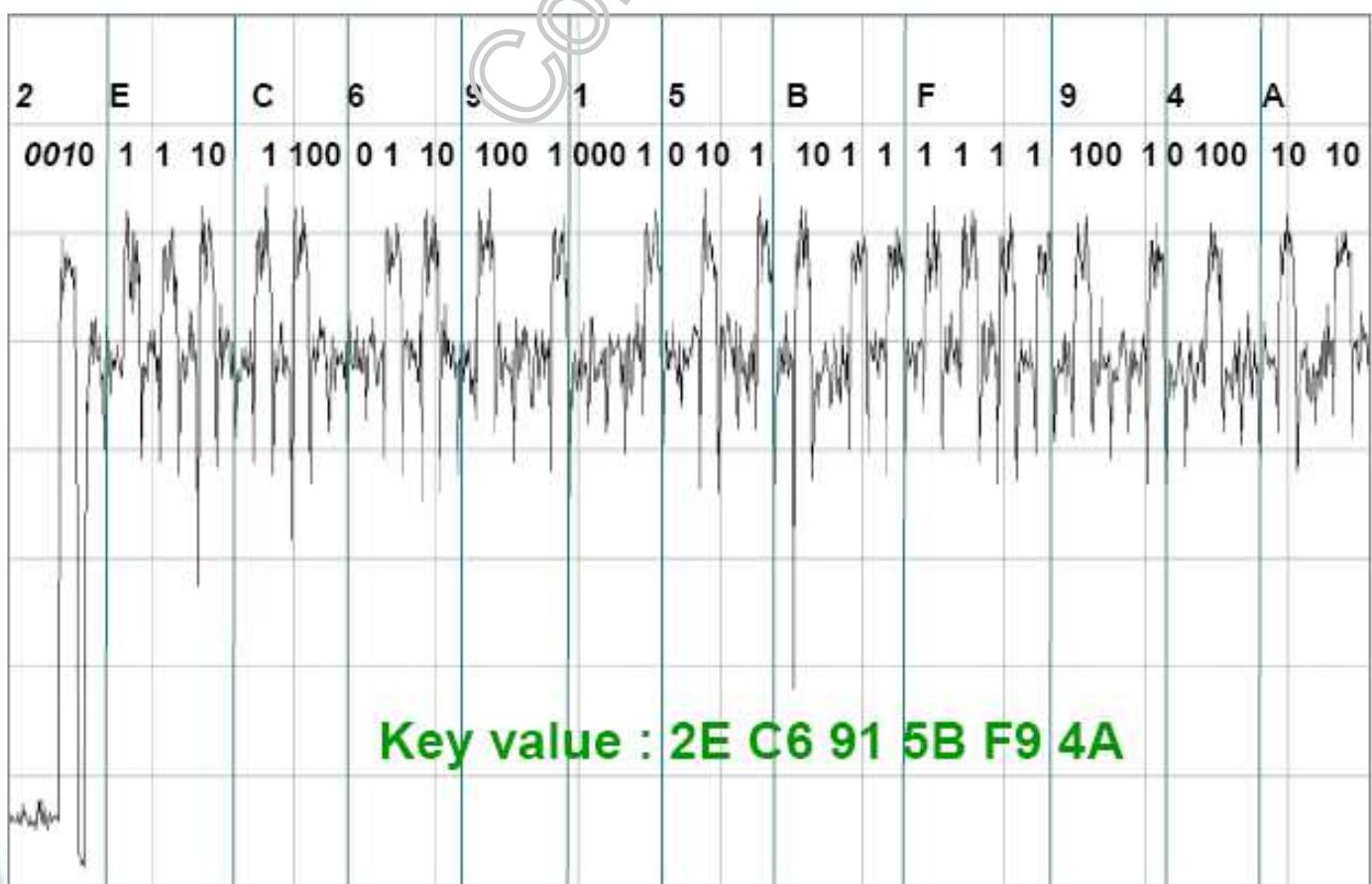
164

## Exemple « réel »



165

## Exemple « réel »



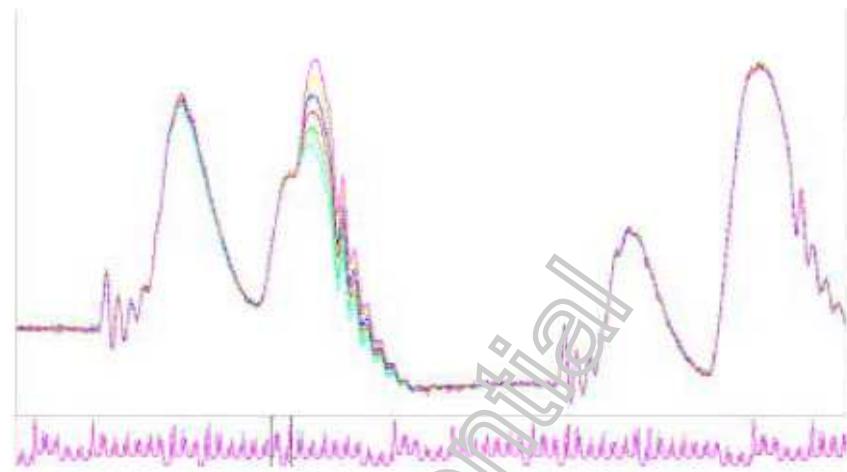
## La DPA (Differential Power Analysis)

Principe : La consommation dépend des opérations effectuées (les instructions) mais aussi des opérandes.

Elle utilise des fonctions statistiques adaptées à l'algorithme visé qui font ressortir des corrélations entre un bit intermédiaire  $a$  (ne dépendant que d'un fragment  $Kr$  de  $r$  bits de la clé et du message d'entrée  $M$ ) et la consommation de courant.

- Based on SPA

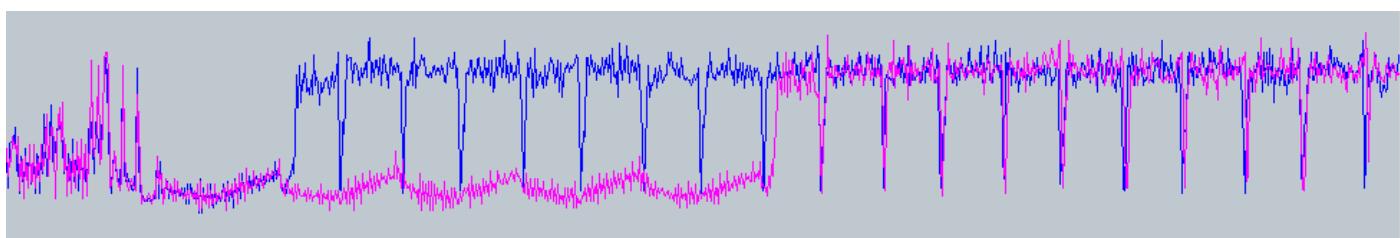
Adding the power of statistics to separate signal from noise



167

## CONCRETE ATTACK ON CONCRETE KEYS

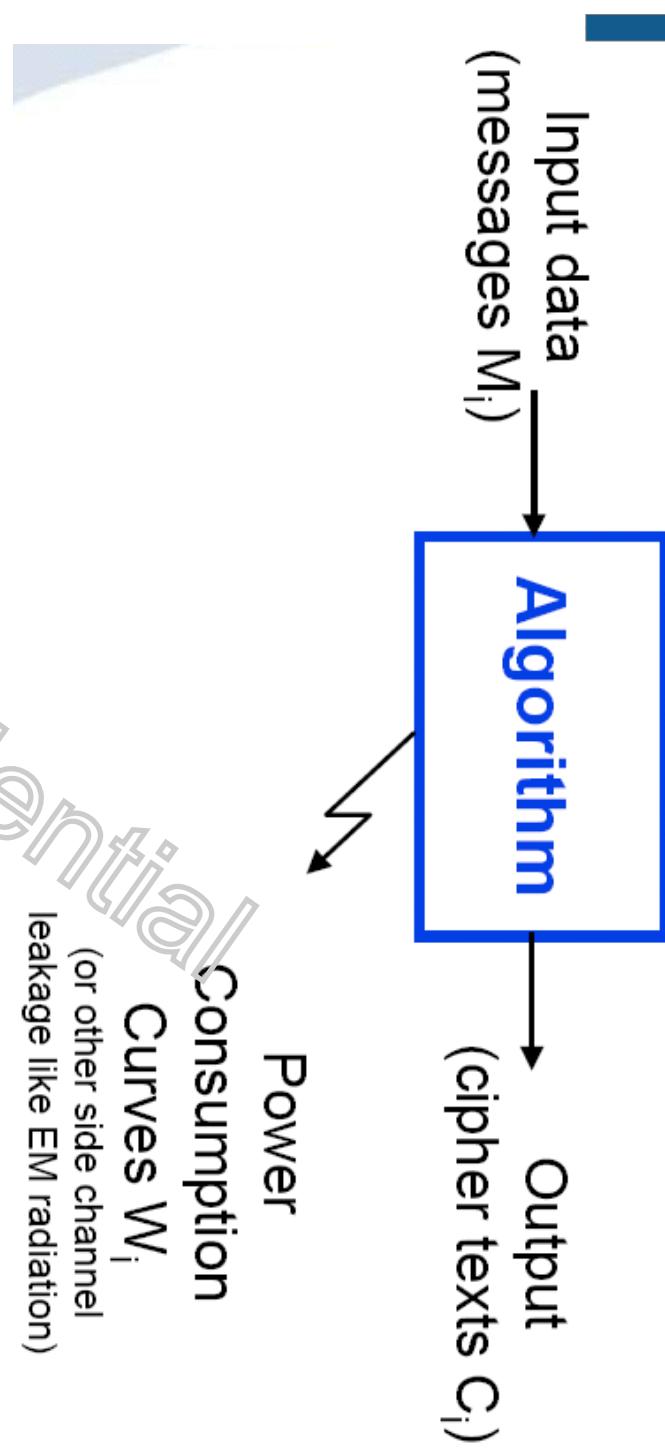
- Comparing:
  - decryption with key = 0000 1111 ...
  - decryption with key = 1111 1111 ...



token designed and manufactured in 1998...

# DPA Hypothesis

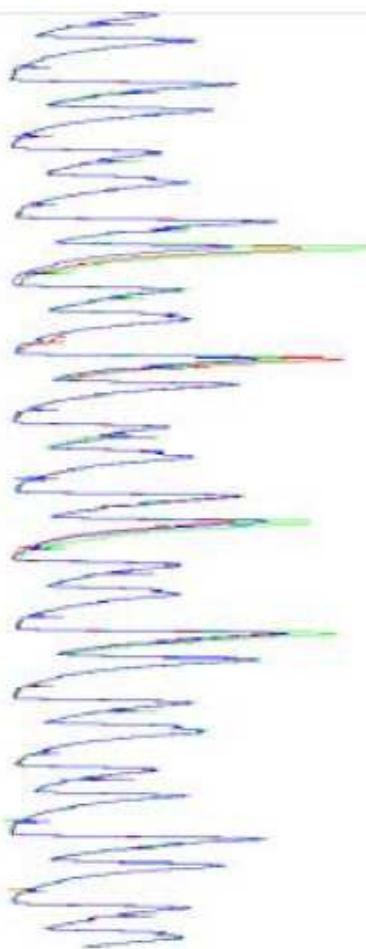
Play the algorithm N times  
 $(100 < N < 100000)$



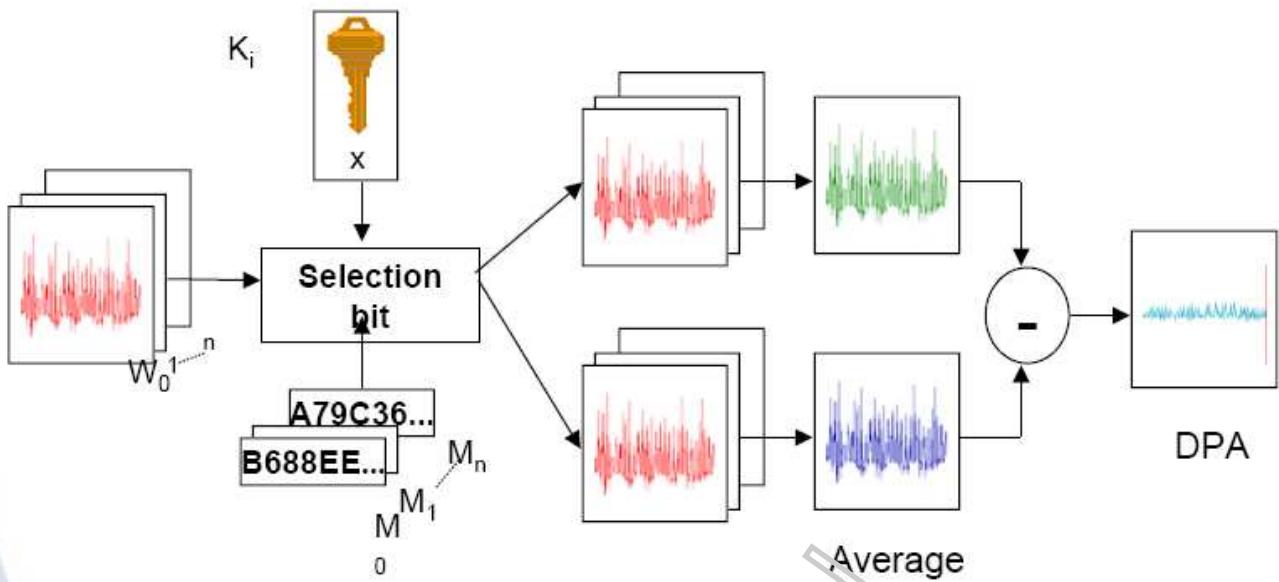
## Acquisition Procedure

- After data collection, what is available ?

- N plain and/or cipher random texts
  - 00**
  - 01**
  - 02**
  - B688EE57BB63E03E**
  - 185D04D77509F36F**
  - C031A0392DC881E6** ...
- N corresponding power consumption waveforms



## Hypothèse et test



171

## La HODPA (High Order DPA)

La HODPA ou DPA d'ordre n est aussi basée sur une étude statistique de la consommation de courant de la carte.

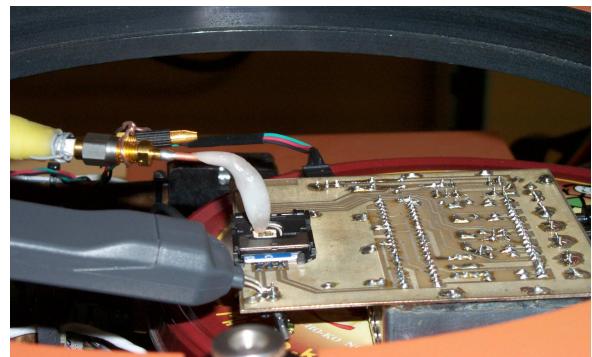
Différence : elle utilise des corrélations entre la consommation de courant et n variables intermédiaires ne dépendant que d'un fragment de la clé et du message d'entrée.

Elle est beaucoup plus difficile à réaliser, mais beaucoup plus puissante.

172

# COUNTER-COUNTER MEASURE

- Against *adding noise to the power consumption signal.*
- Capture electromagnetic radiation at various chip locations!
- Equipment:



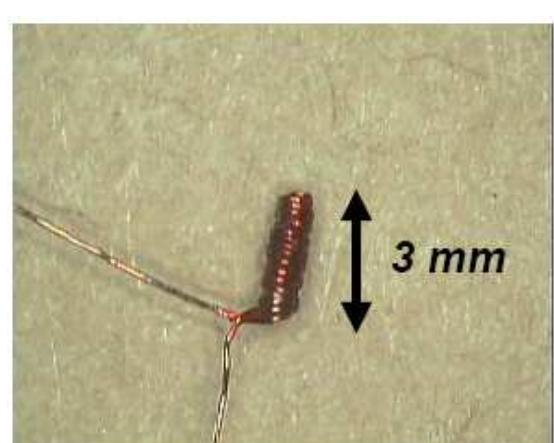
## Les émissions EM

**Principe :** Les courants qui circulent dans la puce induisent des champs électromagnétiques qui sont susceptibles de donner le même type d'information que le courant.

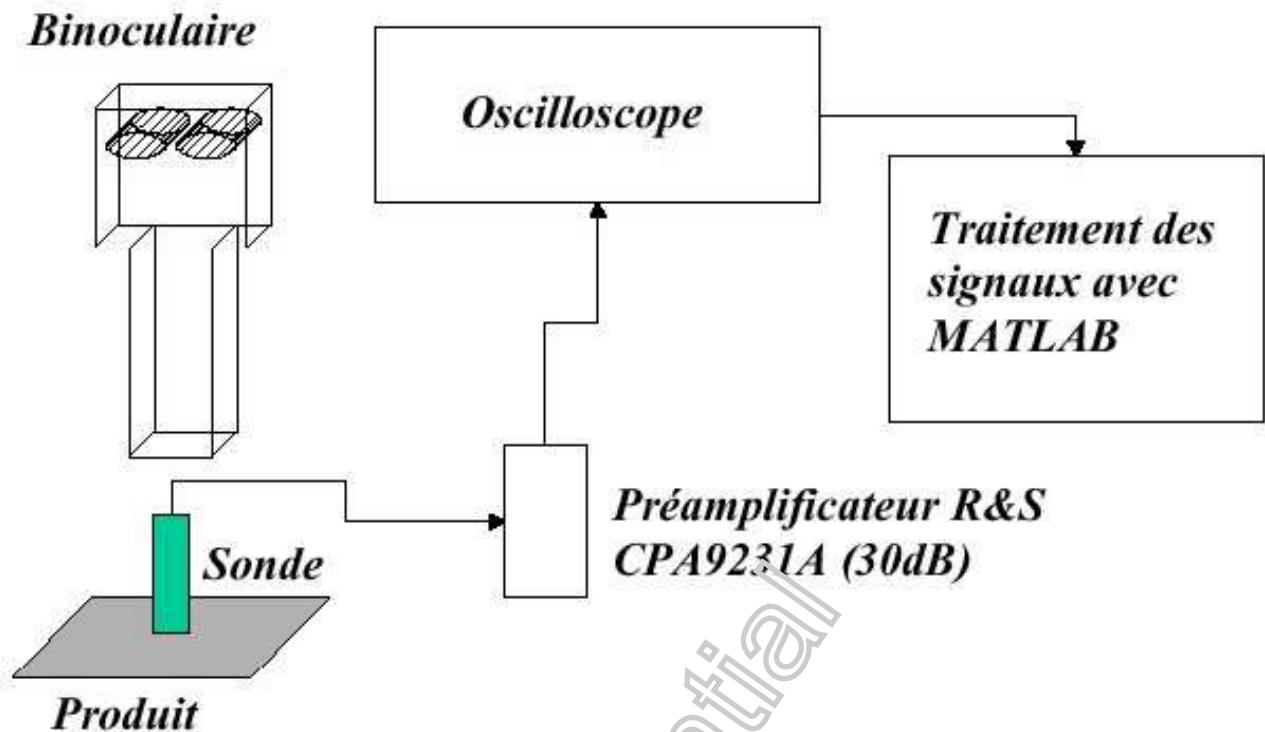
**Différence :** L'information est plus locale. On peut déplacer la micro-sonde électromagnétique au dessus de la zone qui nous donnera le plus d'informations (exemple : co-processeur cryptographique).

**Avantage :** Insensible aux contre-mesures physiques tels que l'ajout de bruit en sortie ou le lissage de la consommation globale de courant.

**Inconvénient :** La reproductibilité des mesures est difficile.



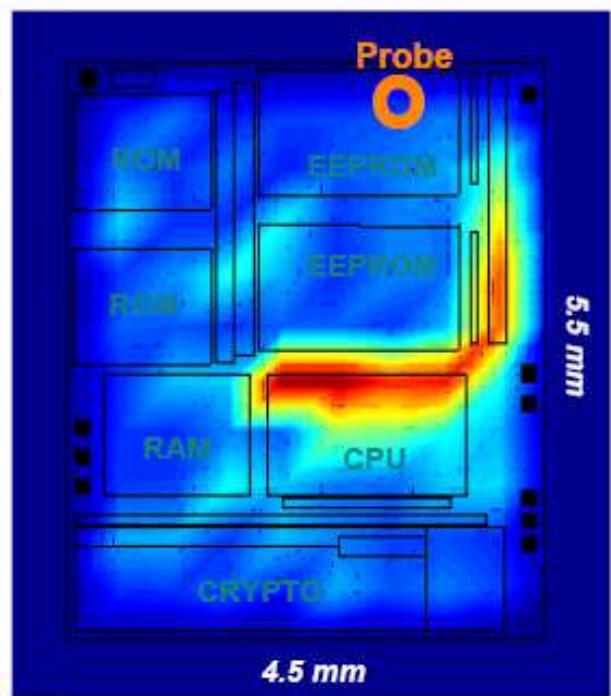
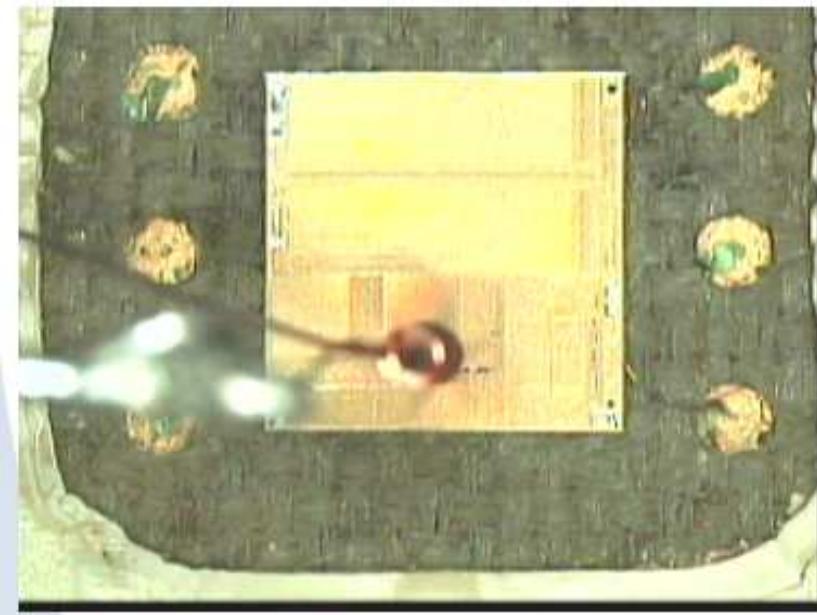
## Le dispositif de cartographie EM



175

## Le dispositif de cartographie EM

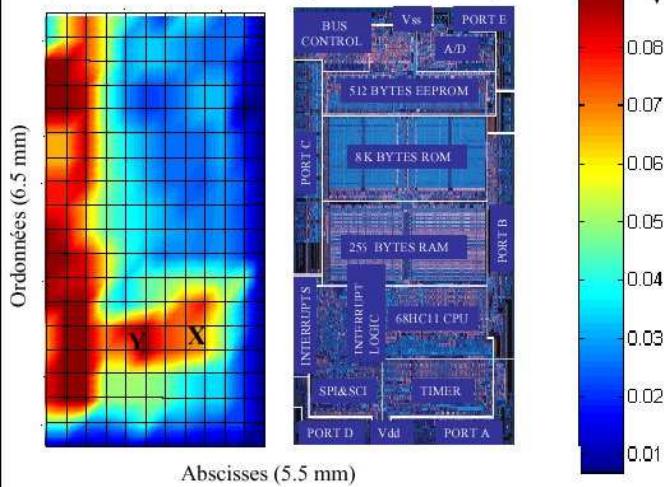
- Horizontal cartography (XY plane)
  - to pinpoint instruction related areas
  - better if automated



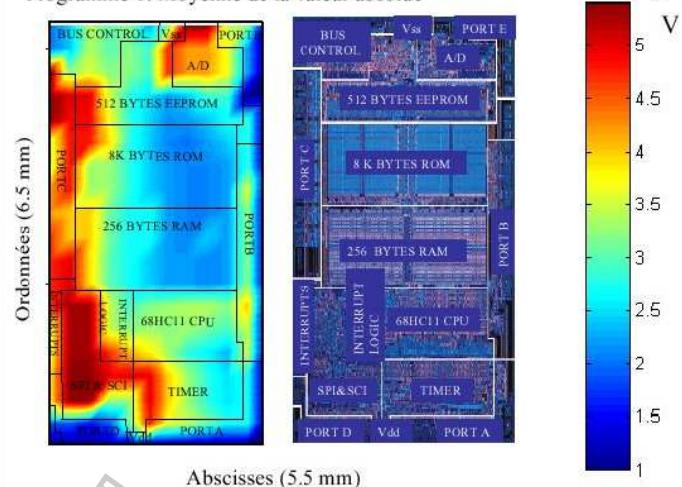
176

## Cartographie pour l'algo 1

Programme 1: Maximum en amplitude de chaque acquisition

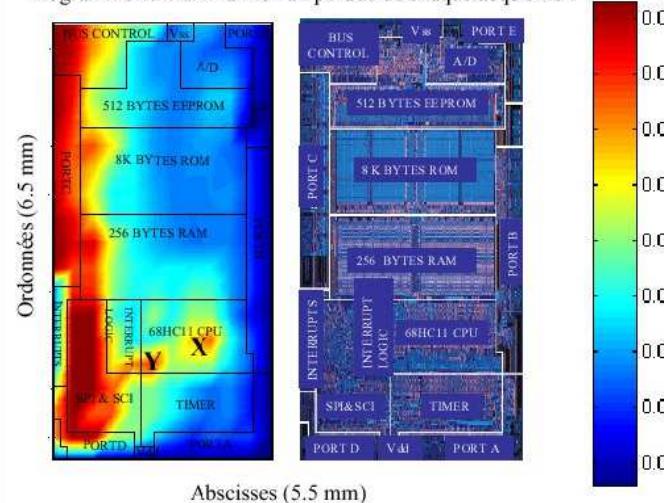


Programme 1: Moyenne de la valeur absolue

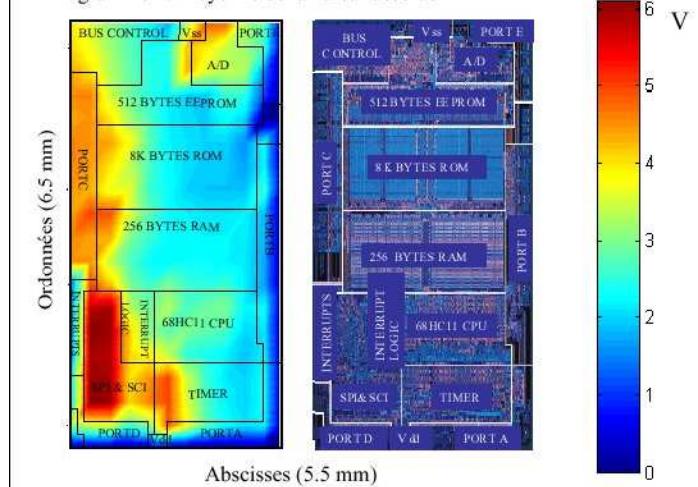


## Cartographie pour l'algo 2

Programme 2: Maximum en amplitude de chaque acquisition



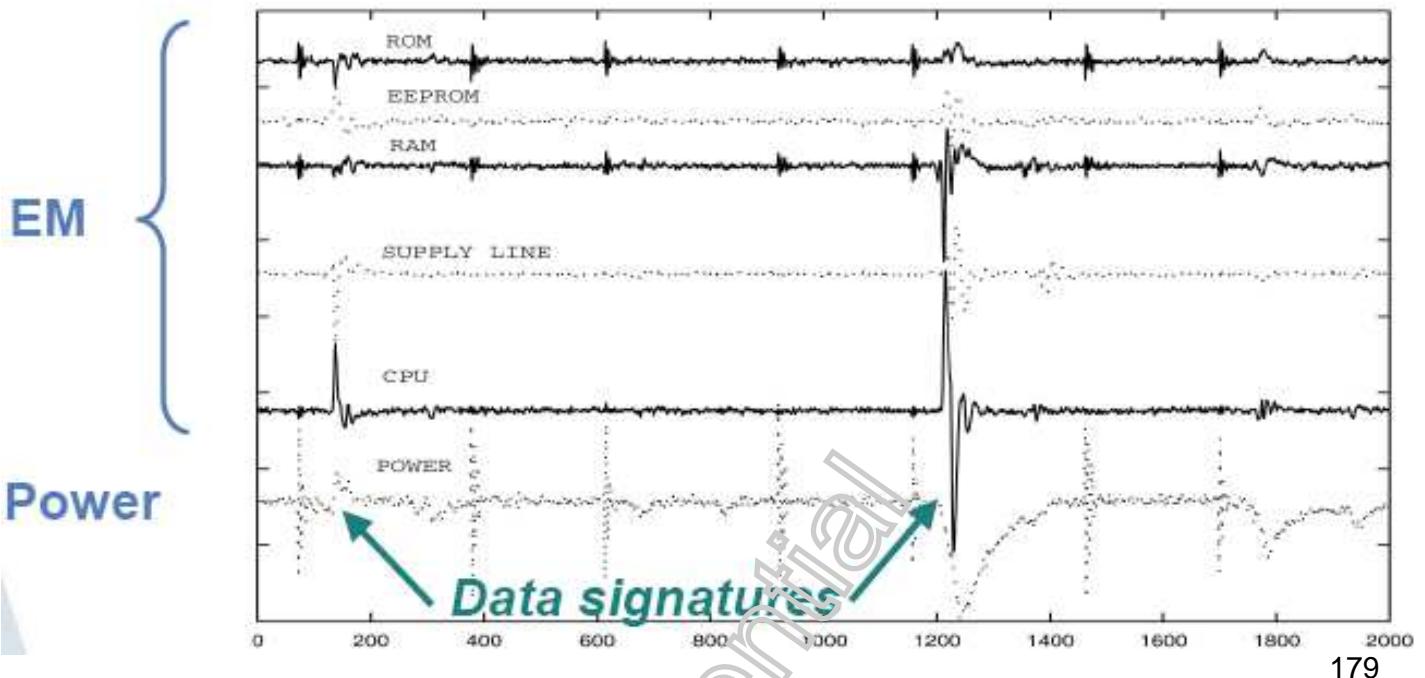
Programme 2: Moyenne de la valeur absolue



## Différence EM/Power

- EM signals versus XY probe position

Differential traces between (00h ⊕ 00h) and (FFh ⊕ 00h) picked up at different locations



179

## Attaques EM

la SEMA (Simple EM Analysis)

la DEMA (Differential EM Analysis)

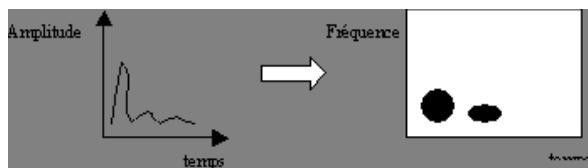
=> nécessite moins d'acquisitions

180

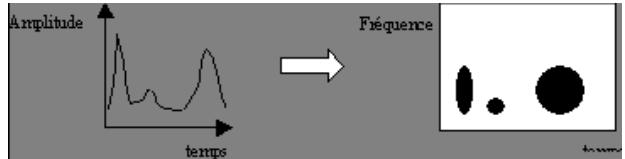
## La rétro-conception logicielle

méthode du « dictionnaire » :

Une instruction :



Une autre instruction :



- La séquence des deux :



Utilisation possible de la consommation en courant ou des émissions EM.

Statut : recherches en cours.

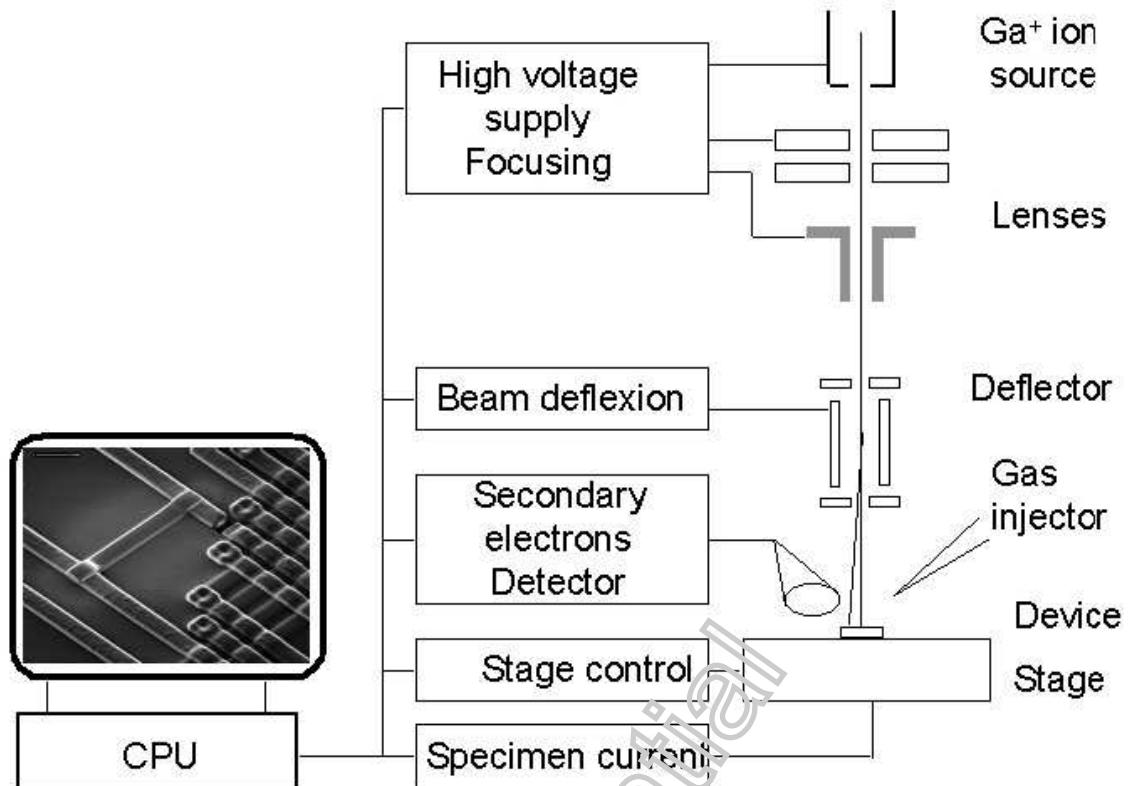
Invasives

micro-probing

modification de circuit (FIB)

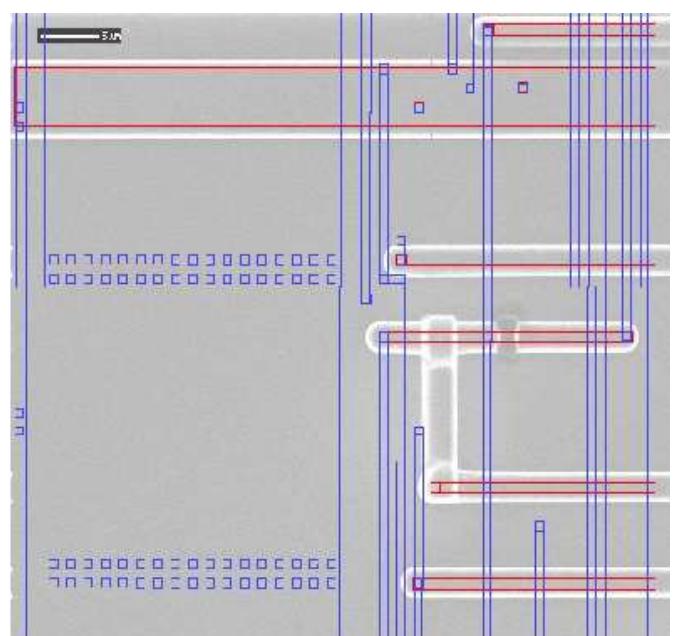
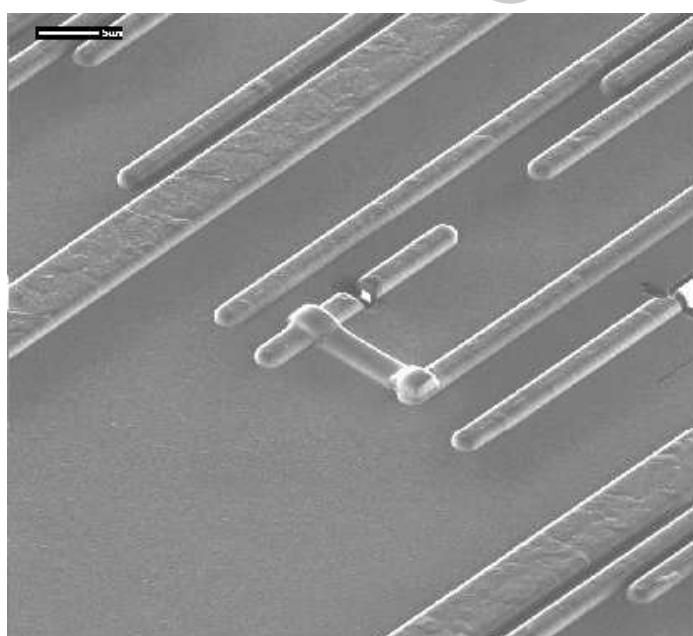
réetro-conception du circuit

## Le FIB (Focused Ion Beam)



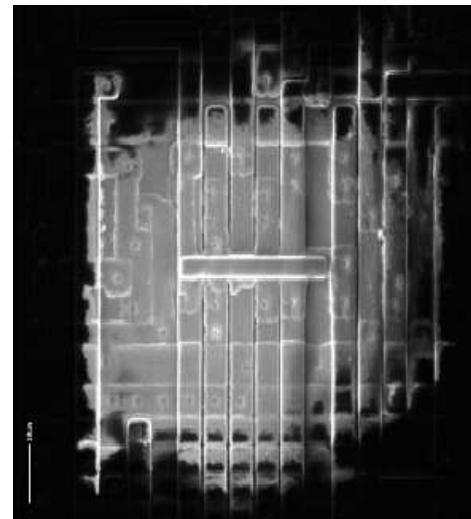
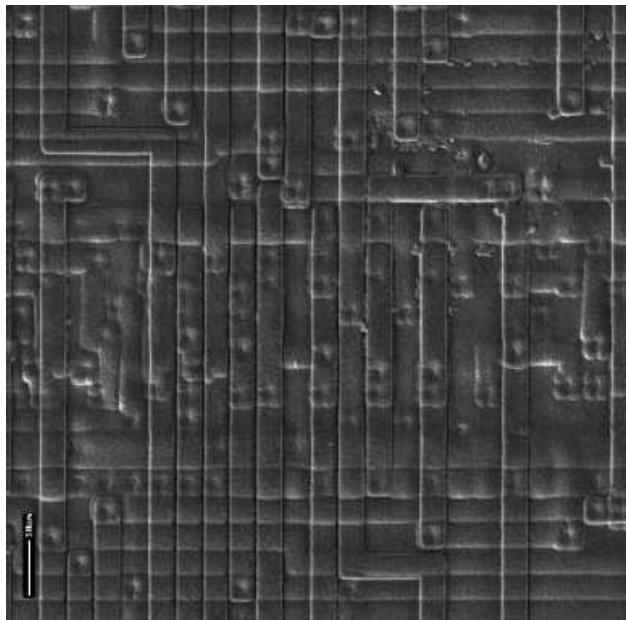
183

## Modifications de circuits (1/2)

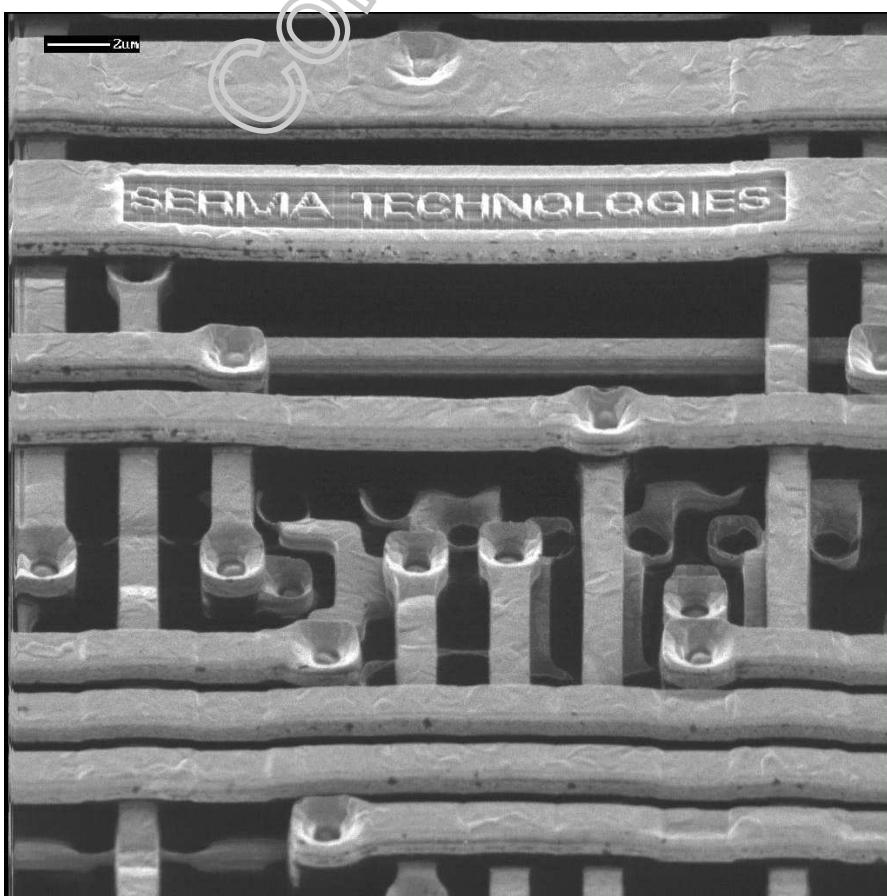


184

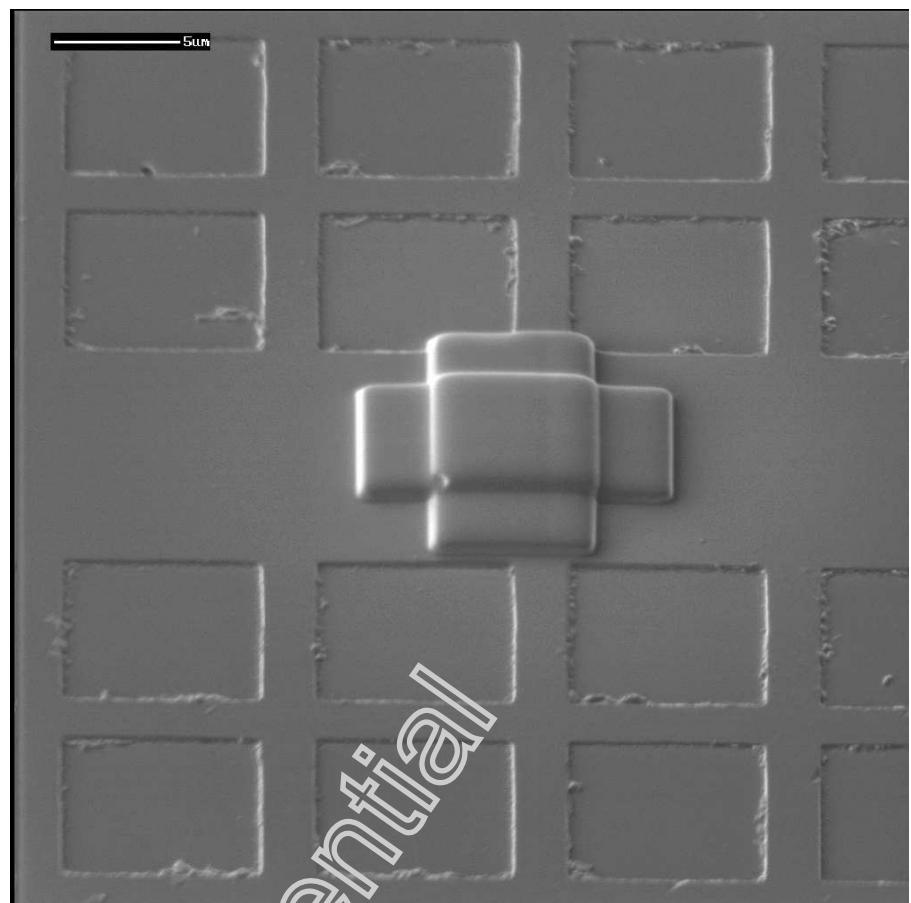
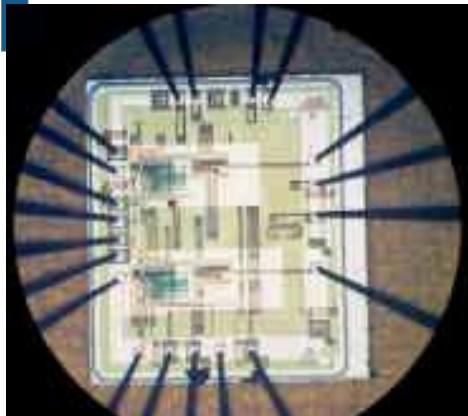
## Chip Re-Wiring / Addition of a Track



## Modifications de circuits (2/2)



## Micro-probing

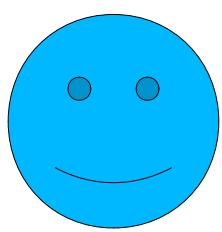
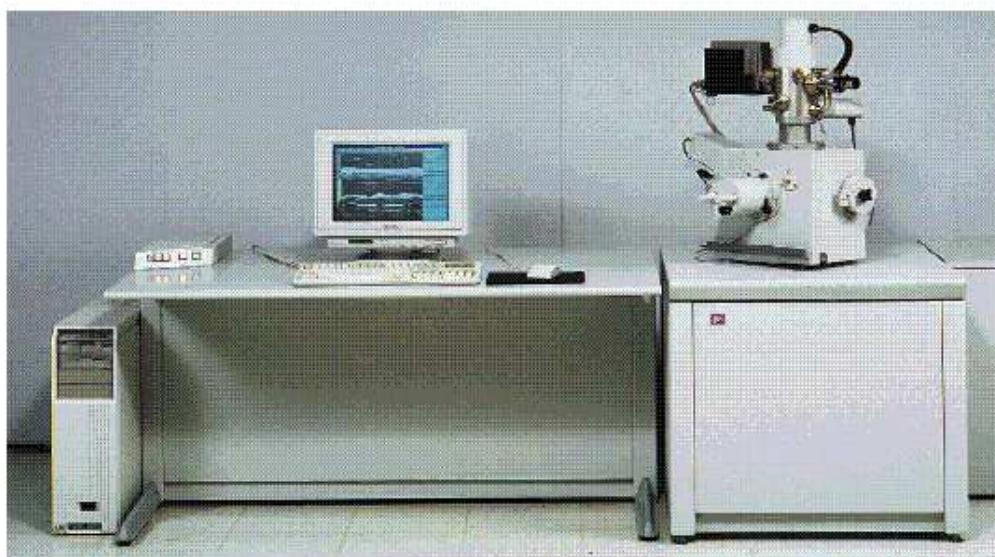


187

## Coût

**Très cher !**

- If you have more money or if you are a student.



188

## Cotation d'une attaque

### Niveau d'expertise requise

Expert, ..., homme de la rue

### Connaissance du produit

Information publique, diffusion restreinte, sensible, critique

### Accès au produit

Nombre d'échantillon nécessaire : <10 , <100, <1000, non pratiquable

### Temps d'identification

<1 heure, 1 jour, 1 semaine, 1 mois, ...

### Temps d'exploitation

<1 heure, 1 jour, 1 semaine, 1 mois, ...

### Equipement

Aucun, standard, spécialisé, ...

...

=> La sécurité et le business s'opposent. Il faut donc trouver un juste compromis !

189

## La norme FIPS 140



FIPS 140-2, June 2001, Security requirements for Cryptographic Modules :

• <http://www.csrc.nist.gov/publications/fips/>



Recommandations pour :

• Physique,

• OS,

• Key Management

• EM

• Self Tests

• Design Assurance

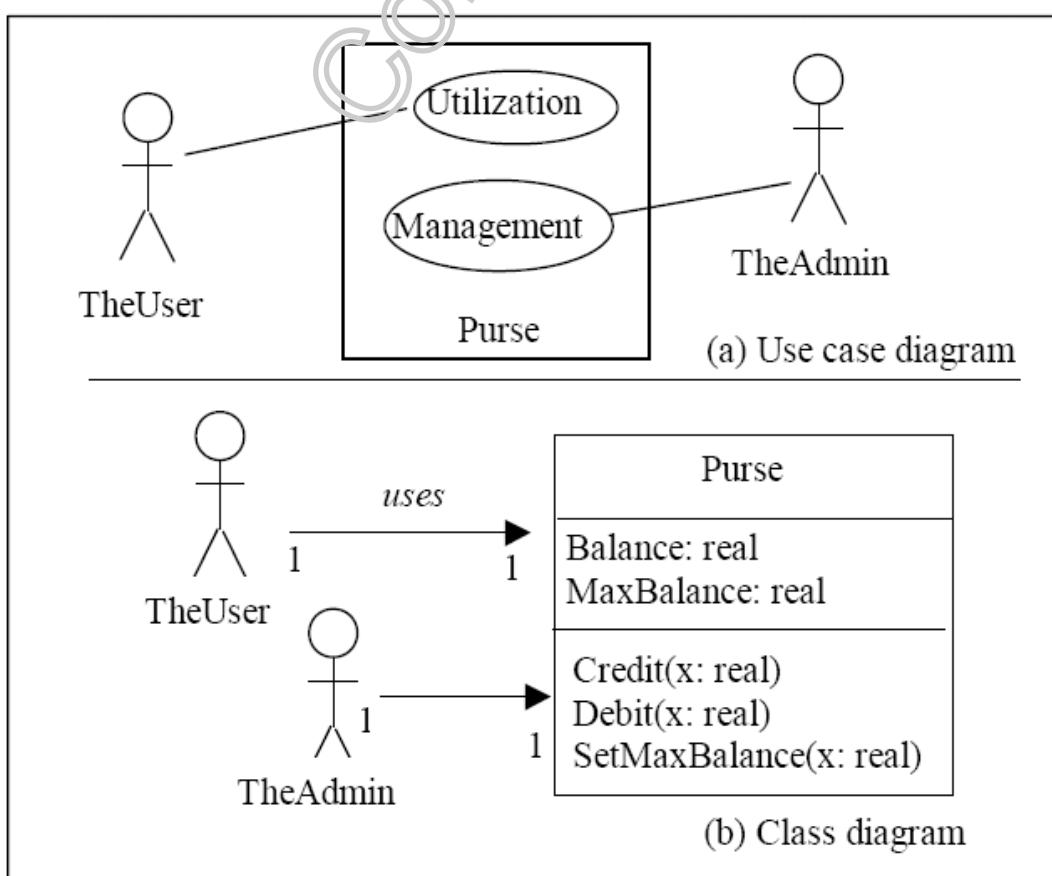
190

# Sécurité physique dans FIPS

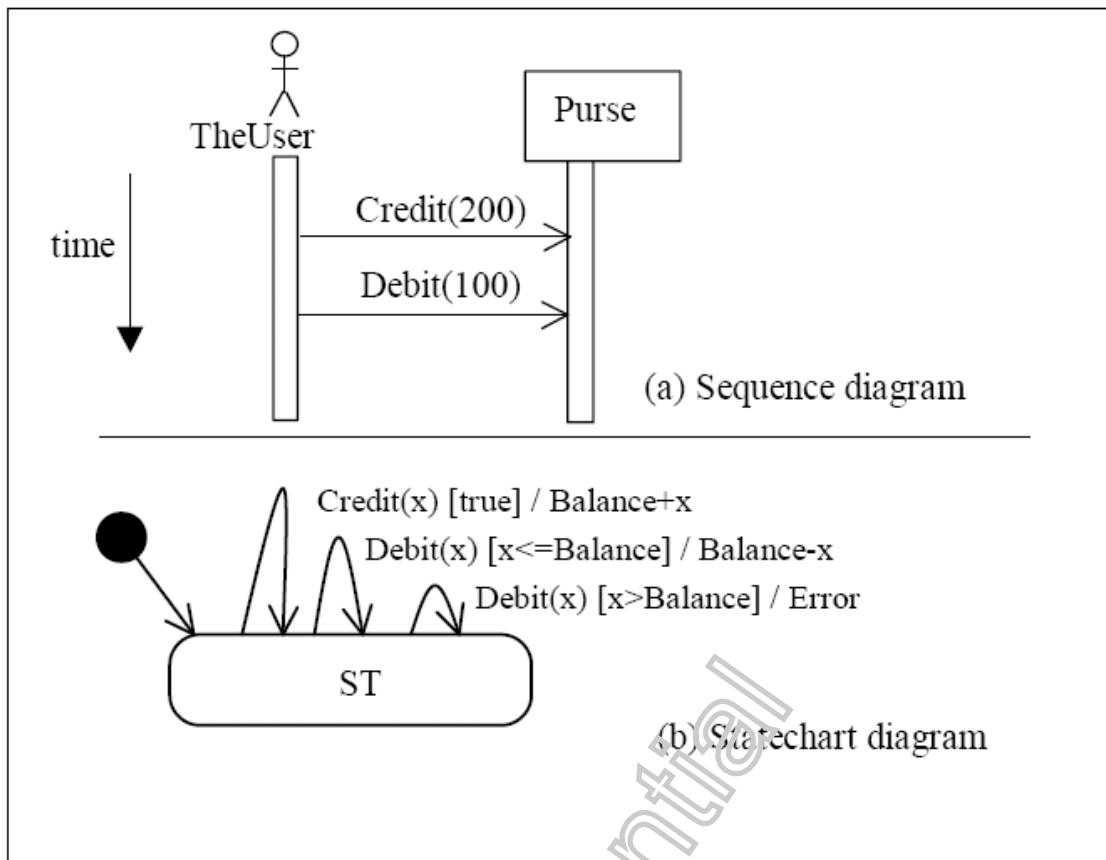
	General Requirements for all Embodiments	Single-Chip Cryptographic Modules	Multiple-Chip Embedded Cryptographic Modules	Multiple-Chip Standalone Cryptographic Modules
Security Level 1	Production-grade components (with standard passivation).	No additional requirements.	If applicable, production-grade enclosure or removable cover.	Production-grade enclosure.
Security Level 2	Evidence of tampering (e.g., cover, enclosure, or seal).	Opaque tamper-evident coating on chip or enclosure.	Opaque tamper-evident encapsulating material or enclosure with tamper-evident seals or pick-resistant locks for doors or removable covers.	Opaque enclosure with tamper- evident seals or pick-resistant locks for doors or removable covers.
Security Level 3	Automatic zeroization when accessing the maintenance access interface. Tamper response and zeroization circuitry. Protected vents.	Hard opaque tamper-evident coating on chip or strong removal-resistant and penetration resistant enclosure.	Hard opaque potting material encapsulation of multiple chip circuitry embodiment or applicable Multiple-Chip Standalone Security Level 3 requirements.	Hard opaque potting material encapsulation of multiple chip circuitry embodiment or strong enclosure with removal/penetration attempts causing serious damage.
Security Level 4	EFP or EFT for temperature and voltage.	Hard opaque removal-resistant coating on chip.	Tamper detection envelope with tamper response and zeroization circuitry.	Tamper detection/ response envelope with tamper response and zeroization circuitry.

Table 2: Summary of physical security requirements

## Simple UML use case and class diagrams



## Simple UML sequence diagram and statechart for the Purse object



193

## Behaviors

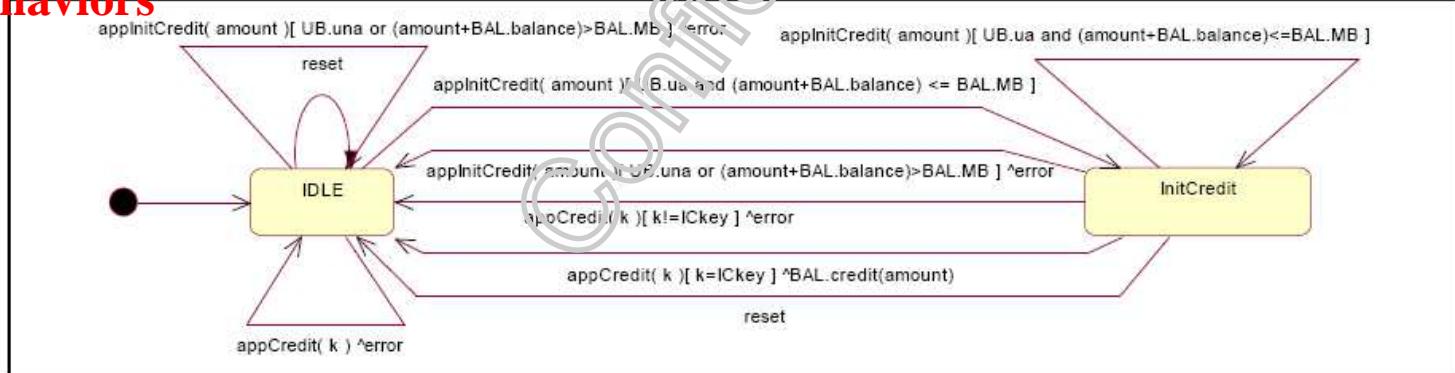


Figure 4. Limited Purse behavior

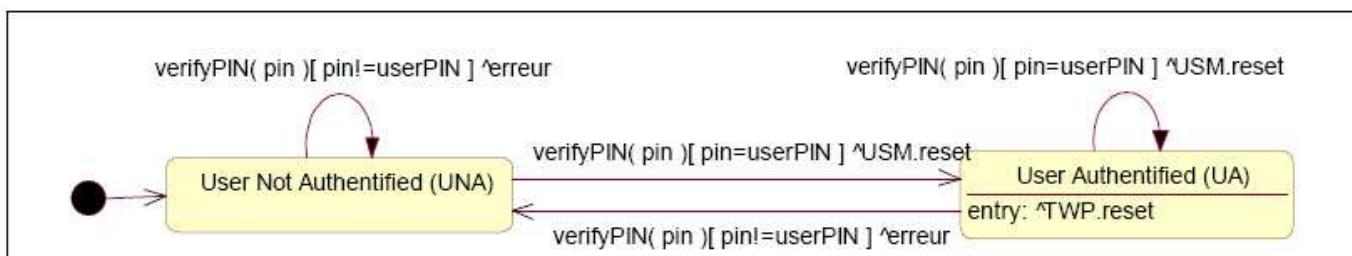
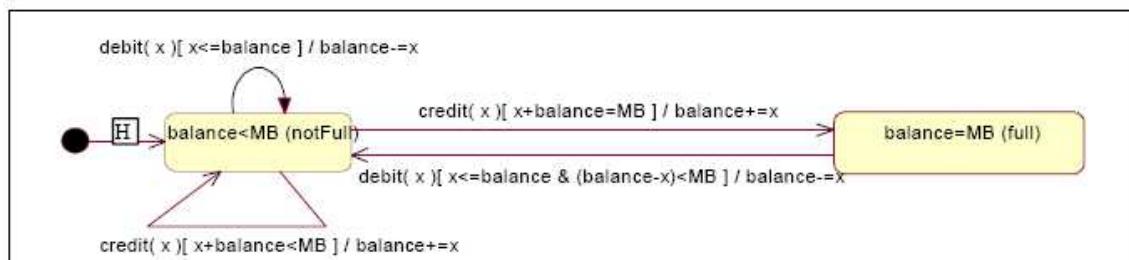


Figure 5. Purse authentication behavior



194

Figure 6. Balance behavior

## Les méthodes formelles et le test

Conception formelle de certains partie de l'OS (voire de l'OS complet)

Preuve formelle sur des aspects sécuritaire

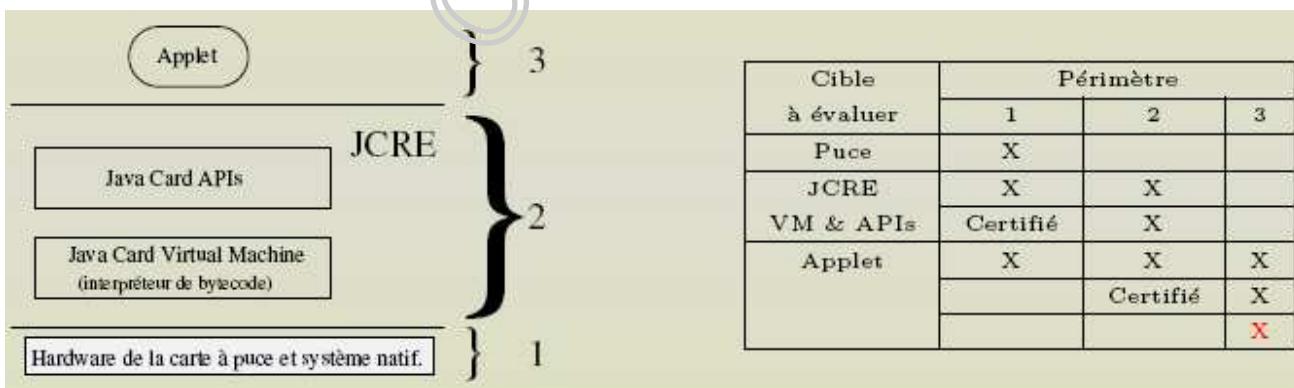
Génération de jeux de tests associés aux spécifications

...

195

## Le périmètre d'évaluation

Défini dans la cible de sécurité dans la description de la cible d'évaluation.



Nouveaux problèmes apportés par les applets Java Card :

- pas de définition précise du périmètre d'évaluation ;
- pas de méthodologie d'évaluation d'une applet ;
- la multi-application.

196

The end (temporaire !)

Confidential