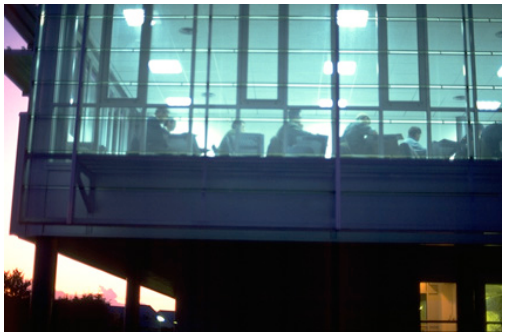


Sécurité et sûreté des systèmes embarqués et mobiles

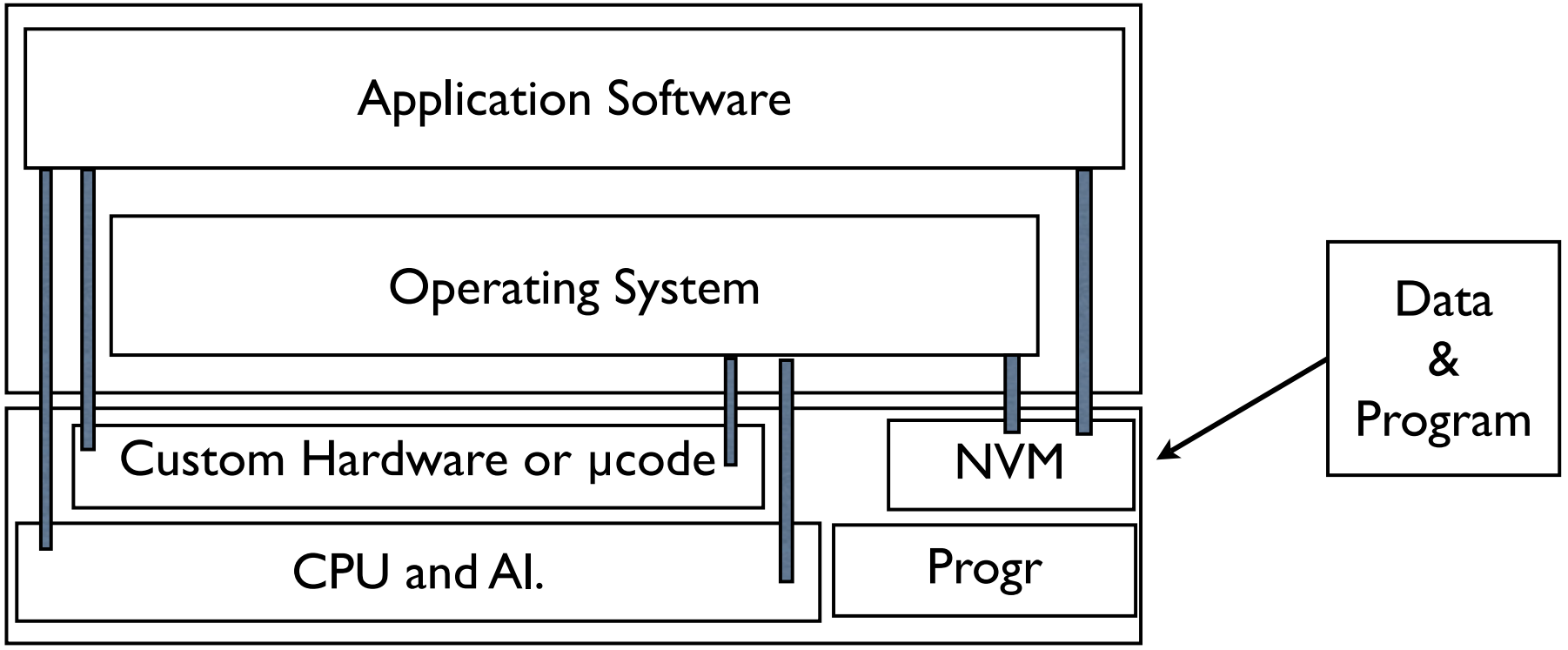
Pierre.Paradinas / @ / cnam.fr

Cnam/Cedric
Systèmes Enfouis et Embarqués (SEE)

Plan du cours

- Sécurité des SEMs
  - La plate-forme et exemple (GameBoy, Smart Card)
  - Les propriétés de confidentialité, intégrité et disponibilité
  - Typologie des attaques
  - Normes, exemple et implémentation
  - Biométrie
- Sécurité et code mobile
  - 2 ème cours

La plate-forme



La plate-forme

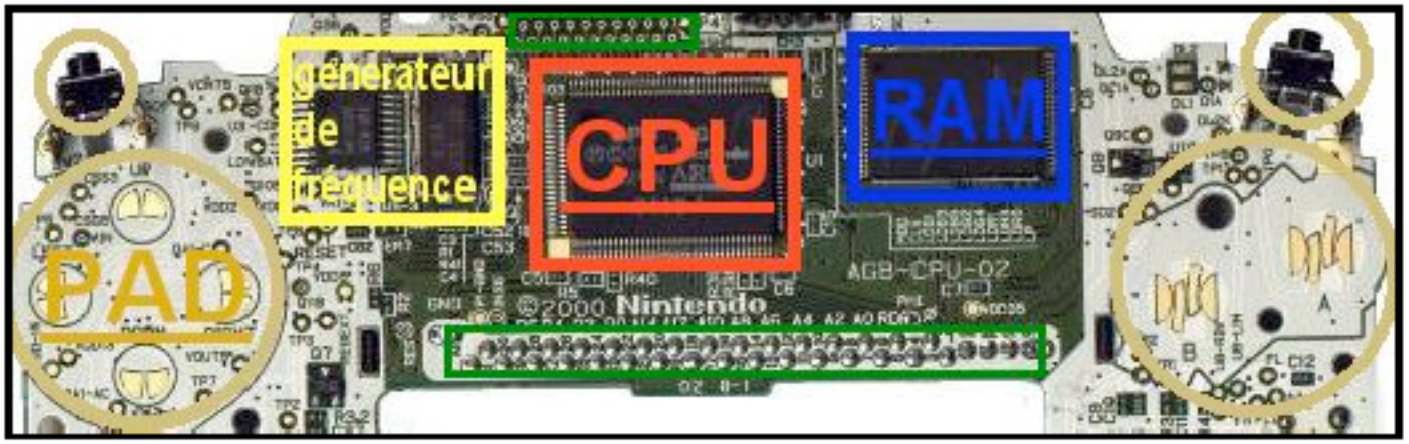
- Nombreux exemples :
  - De la machine à laver,
  - ...
  - StB, carte à microprocesseur, GBA,...

La plate-forme (exemple 1)

- La carte à microprcesseur (smart card)

La plate-forme (exemple 2)

- Game Boy Advanced



Définitions

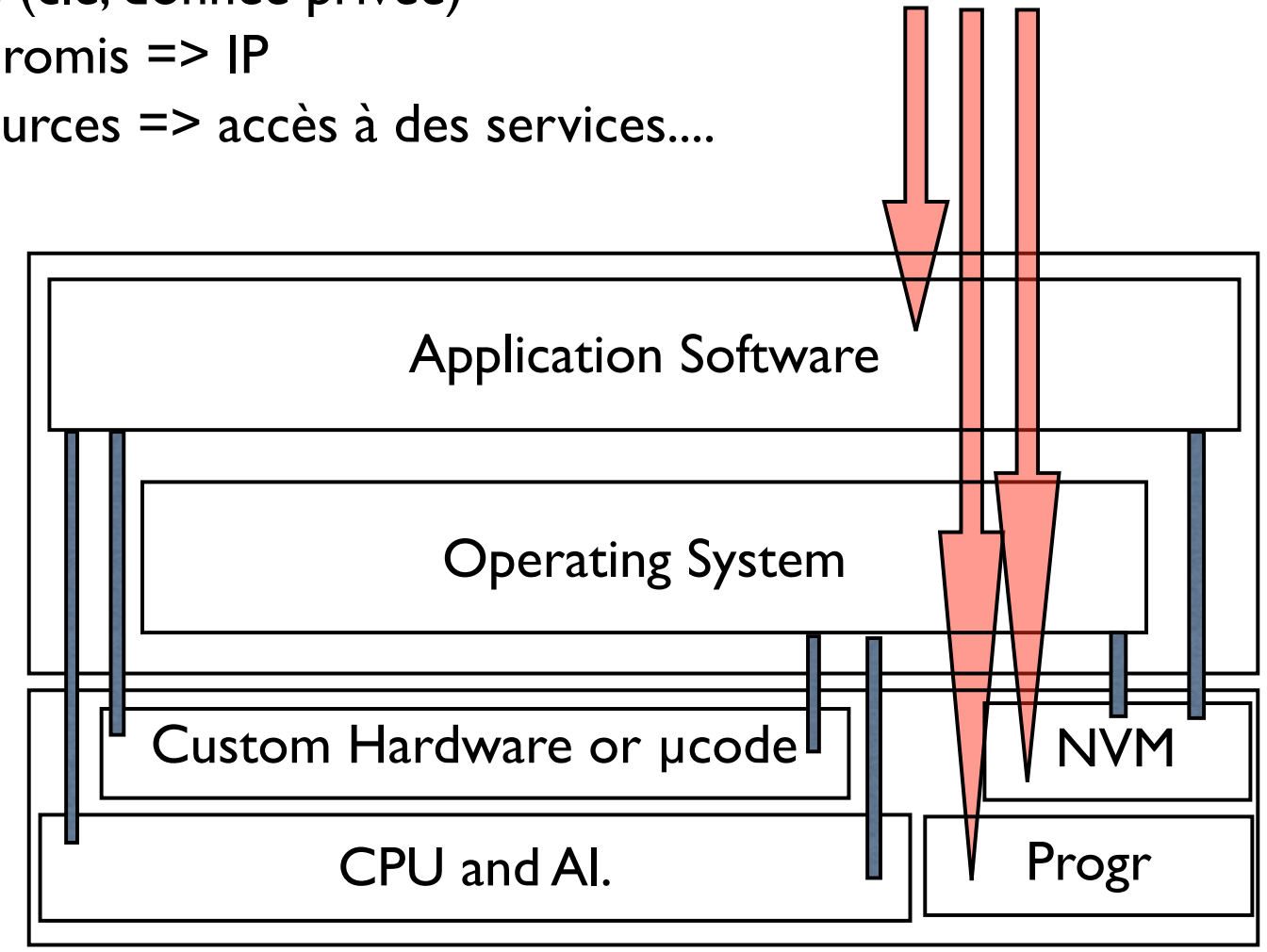
- Des définitions et des traductions :
  - Sécurité : Security
  - Fialailité/Sûreté : Dependability
  - Confidentialité : Privacy
  - Intégrité : Integrity
  - Disponibilité : Availability

La confidentialité

- La confidentialité est la propriété de “secret” attaché aux informations. Seul les entités autorisées à accéder aux ressources le sont.
- Elle est assurée par le contrôle d’accès aux informations et ressources du SEM. De même le SEM ne peut communiquer de l’information qu’aux entités autorisées à recevoir cette information.
  - Mise en oeuvre de techniques pour assurer la confidentialité
    - Contrôle d’accès au SEM
    - Contrôle d’accès aux ressources internes dans le SEM

La confidentialité

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

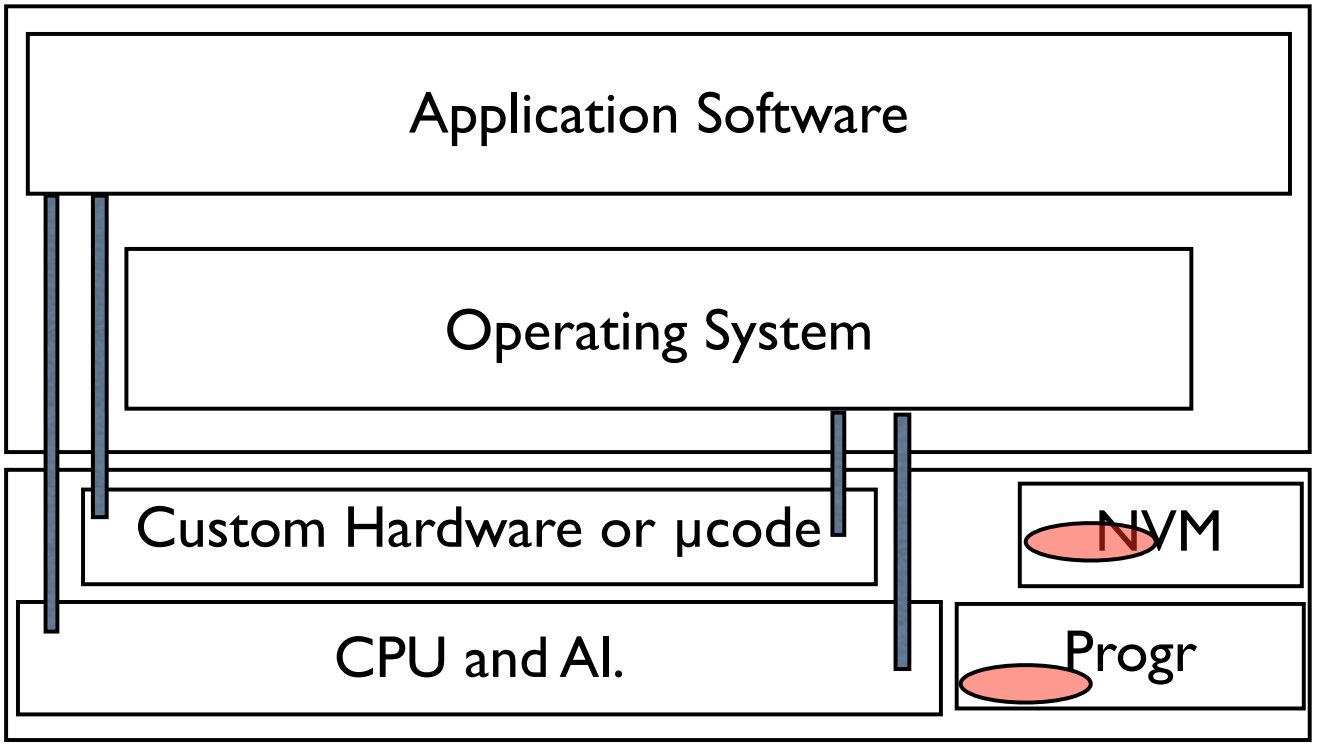


L'intégrité

- L'intégrité est la propriété qui assure qu'une information n'a pas été altérée.
  - Exemple : "*L'intégrité d'une somme*". Sur un chèque la valeur est en chiffre et en toute lettre.
- L'altération d'une donnée concerne :
  - Modification, suppression ou ajout sur les données sans en avoir le droit ou l'autorisation de réaliser ces opérations.
  - Modification, suppression ou ajout sur les programmes sans en avoir le droit ou l'autorisation de réaliser ces opérations.

L'intégrité

- Données et programmes changés



L'intégrité sur les données

- L'intégrité sur les données est assurée par le contrôle sur les actions d'écriture.
- De manière passive on peut vérifier que des données ne sont pas altérées en effectuant des contrôle sur les valeurs :
  - Par redondance, par CRC, par signature,...
    - Ceci peut être fait par matériel ou logiciel

L'intégrité sur les programmes

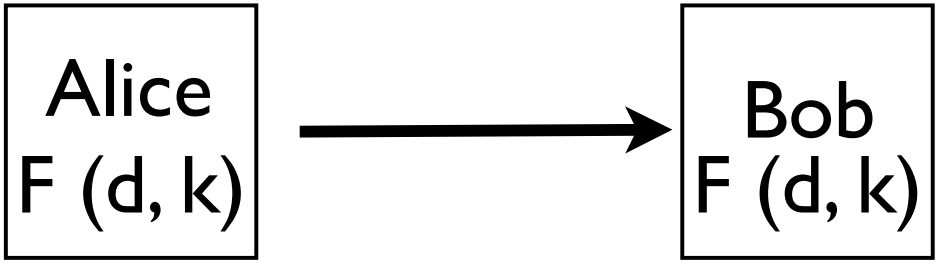
- Un programme ayant subit une altération peut avoir des conséquences très importantes :
  - Modification sur ses données des applications,
  - Modifications sur des données du système d'exploitation :
    - table d'autorisations, programmes,...
    - ouverture de nouveau services
    - ...
  - L'altération initiale peut être volontaire mais peut aussi être le résultat d'une erreur de programmation!

La disponibilité

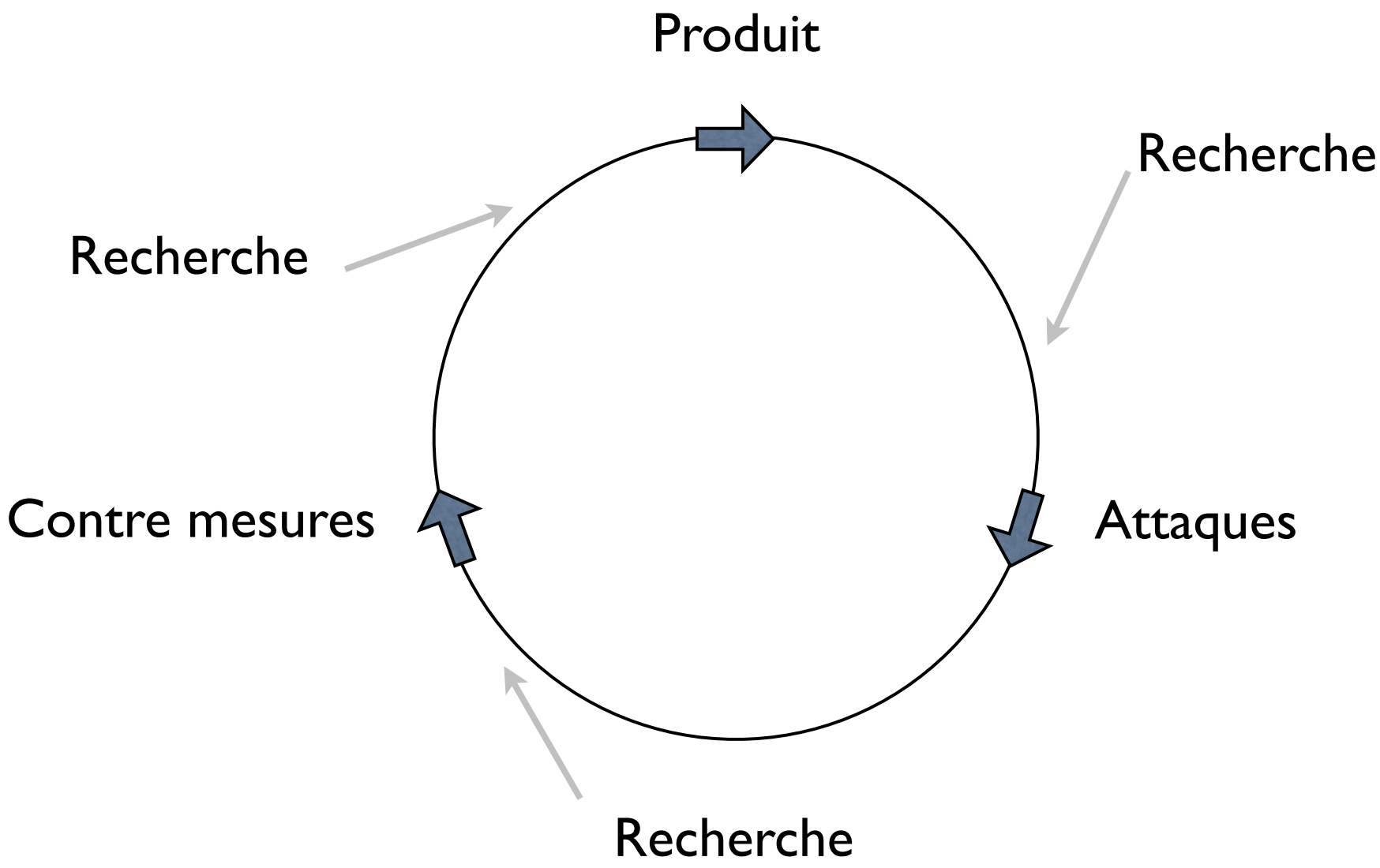
- La disponibilité est la propriété qui assure que les fonctions du du SE&M sont utilisables pour les utilisateurs dans les délais prévus.
  - Dans le cas d'un serveur Web la disponibilité sera le fait que les requêtes sont satisfaites et que l'attaque par déni de services sera sans effet.

Mécanismes et implémentations

- Par exemple dans le cas d'un échange d'information entre Alice et Bob,
- La confidentialité est apportée par un protocole cryptographique (voir cours SN),
- La propriété est présente si :
  - La clé n'est pas révélée,
  - La sécurité sera "évaluée" par la longueur de la clé et les propriétés mathématiques,



Un cycle infernal!



Les menaces sur les plate-formes

- Les menaces sur les SEMs (systèmes embarqués et mobiles) sont celles classiques des systèmes informatique,
- Mais :
  - Le contexte d'utilisation de l'objet peut être spécifique
    - PayTv
  - Les aspects systèmes peuvent être spécifiques
    - Attaques par déclenchement des chauffages en forçant la température sur les capteurs chauffage
  - Les enjeux
    - Carte bancaire = argent

Particularité des SE&Ms

- Le dispositif est entre toutes les mains
  - Porte ouverte aux intrusions...
  - A la différence d'un centre de traitement qui est une unité de lieu, néanmoins les grilles de calculs "ouvertes" connaissent un peu la même problématique
- Via le réseau des attaques sont possibles (porte d'entrée),
- La plate-forme peut être ouverte (Java Architecture)

Classification des attaques

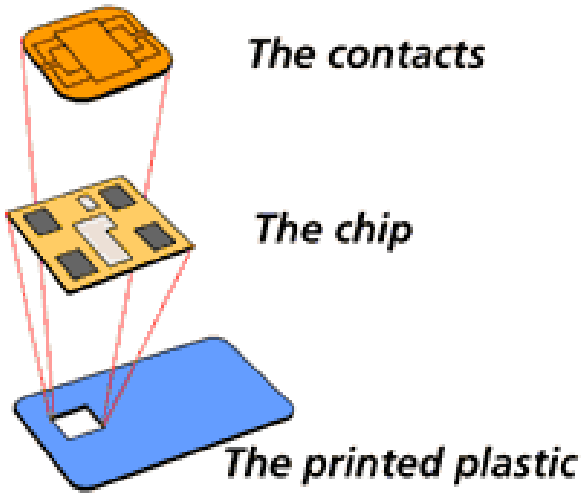
- Les attaques ce font par rapport aux propriétés non fonctionnelles :
  - Confidentialité :
    - L'objectif est d'obtenir des informations.
  - Intégrité :
    - L'objectif est de changer, supprimer ou ajouter des informations.
  - Disponibilité :
    - L'objectif est de rendre le dispositif inutilisable par suite de très nombreuses requêtes.

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,...

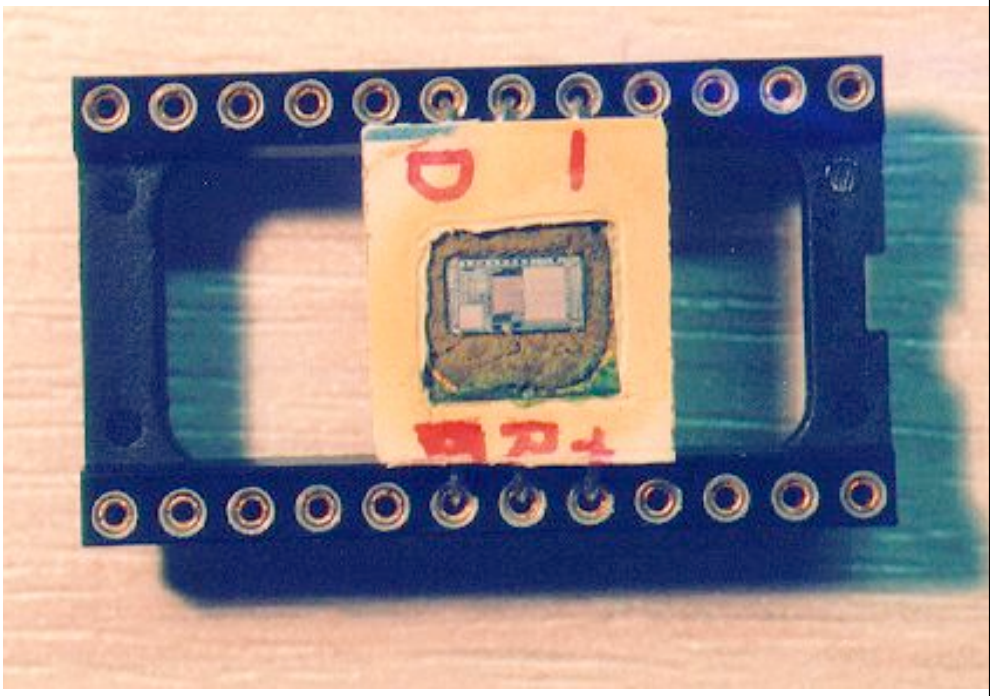
La plate-forme carte à microprocesseur

- Architecture
  - CPU : 8, 16 & 32 bits
  - Mémoires
    - RAM : 512 ko...4/16ko
    - EEPROM/Flash : 128/256 ko...1/4mo
    - ROM : 256/512
  - Cellule cryptographique et générateur de nombre (aléatoire)
  - Détecteur de sécurité,...



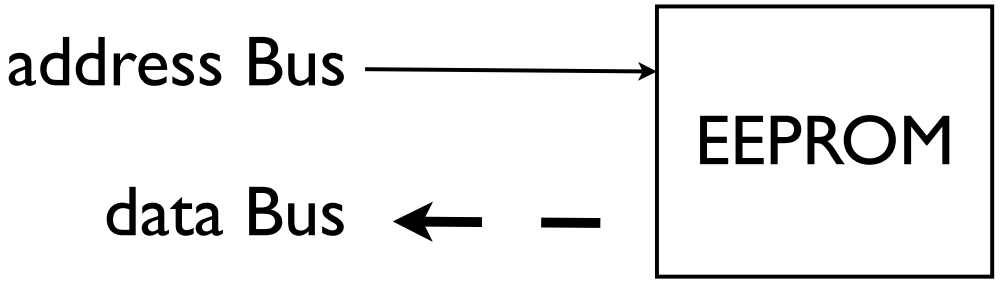
L'attaque physique

- Dé-packaging
- Objectifs : accéder aux éléments à différents niveaux du composants,
  - Supprimer la plaque des contacts, retirer la résine du module avec des solvants,
  - En fonction du degré de sophistication de la fabrication cet aspect est plus ou moins difficile,



L'attaque physique (suite)

- Reconstituer l'architecture du composant,
  - Mémoires, BUS, adresses,...
- Accéder aux informations,
  - Dans les mémoires et/ ou sur le BUS.
- Probing : descente de sonde et action sur le composant



Des attaques physiques :

- Plus ou moins complexes,
  - Nécessitent de l'outillages (!).
- Première étapes à un autre type d'attaque,
- Ce sont des attaques destructives,
- Des contre-mesures
  - Packaging,
  - Détecteur de passivation,
  - Silicium,
  - Architecture.

Des protections silicium et architecture

- Silicium sur plusieurs niveaux (bus plié, mélangés,...),
- Structures des mémoires aléatoires,
- Bus chiffré,
- ...

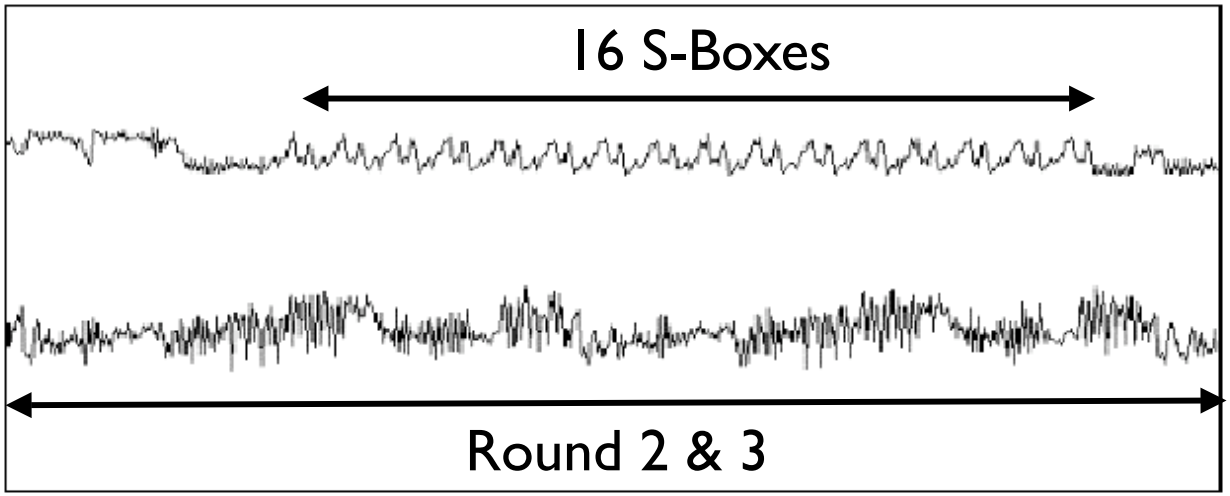
Les attaques par canaux cachés

- Analyse des consommation
  - Power Analysis Attacks
    - SPA et DPA
- Les attaques en temps,
- Les attaques par injections de fautes,
- Les attaques par champs électromagnétiques

Les attaques en courant

- Premières publications autour de 1995,
- L'exécution d'un programme dans un composant requiert que celui-ci soit alimenté,
- La consommation de courant dans le composant est fonction des opérations réalisées,
- Par conséquence, dans le cas ou le programme exécute un calcul cryptographique (ou autre), une observation fine et précise des consommations donnera de l'information,

SPA (Simple Power Analysis)



- D'après Cryptography Research :
  - Algorithme DES,
  - Permutations (initiales et finales) et 16-Round (S-Boxes)

Un autre exemple

- D'après NEC & Princeton

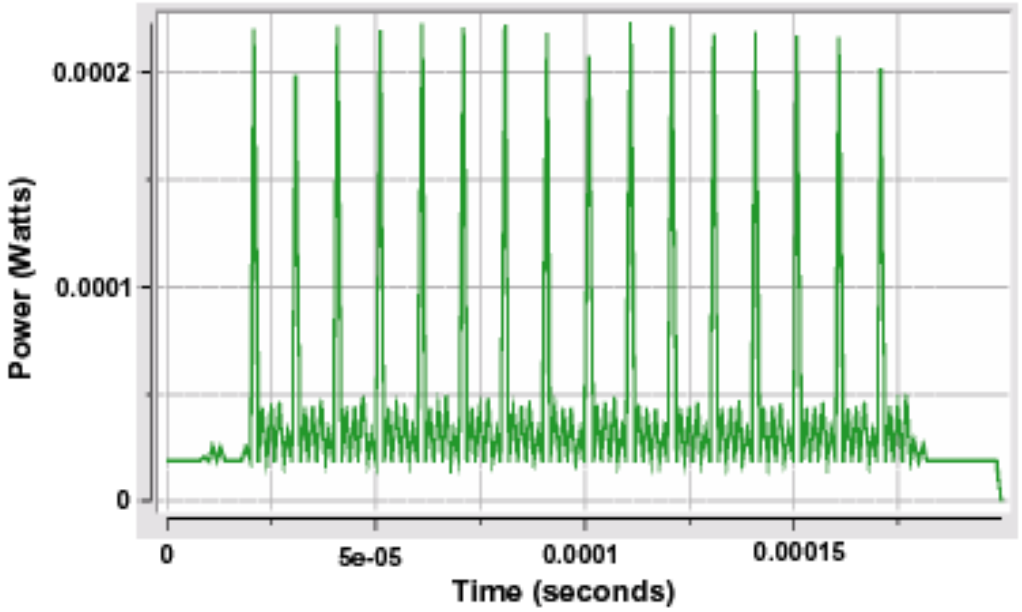


Figure 2: The power consumption profile of a custom hardware implementation of the DES algorithm

La DPA (Differential Power Analysis)

- Plus complexe que la SPA,
- Mise en oeuvre de donnée statistiques,

DPA : l'idée

- n chiffrement sont effectués sur une entrée,
- Pour chaque entrée une collecte complète des données est effectué
- Le cryptanalyste introduit une hypothèse sur la clé qui est testée
  - Par exemple, on peut choisir une fonction de sélection qui sur le DES doit calculer le bit 2 comme le résultat de la 5 ème S-Box lors du dernier tour avec une valeur donnée de clé pour la S-Box de valeur 010110.
  - => Si la fonction de sélection est correcte alors la valeur du bit 2 est correcte sinon dans un cas sur deux elle est fausse.
  - Le processus est reproduit sur l'ensemble des bits de la clés

DPA : exemple

- Exemple de DPA avec 4 courbes (d'après Cryptography Research) :
  - Référence,
  - Hypothèse vraie,
  - Hypothèse fausse,
  - Hypothèse fausse.

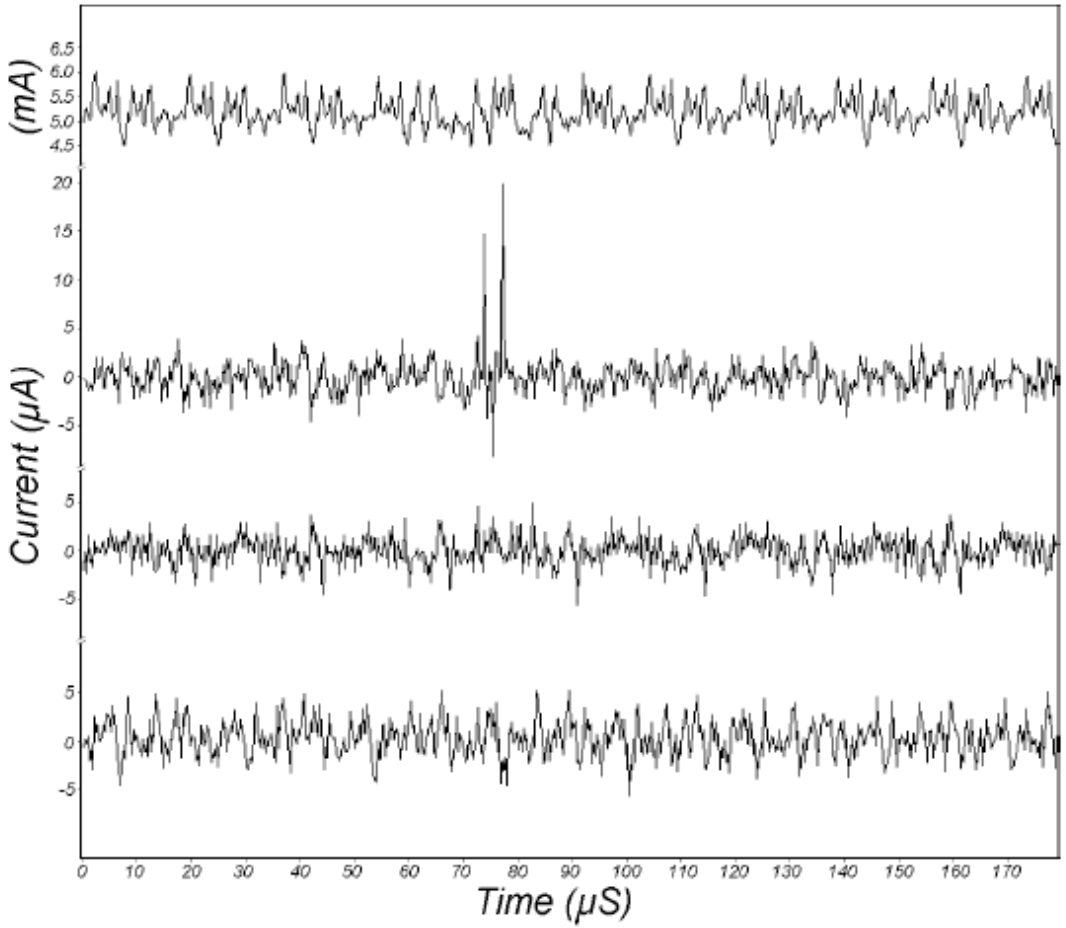


Figure 4: DPA traces, one correct and two incorrect, with power reference.

SPA et DPA Contre-mesures

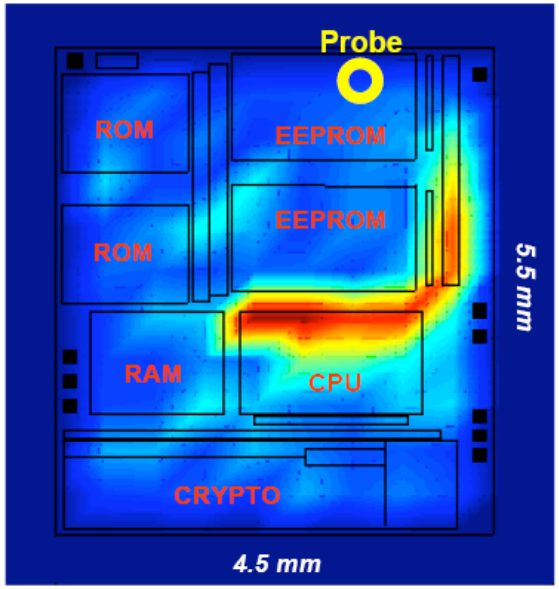
- Diminuer les signaux, flot d'exécution constant, flot d'exécution choisit (peu de fuite),...
  - mais difficile de réduire à 0 !
- Introduction de bruit dans la consommation et/ou le timing,
- "Cacher" l'information sensible;
  - => Néanmoins, en augmentant le nombre d'échantillons une partie de ces approches tombent,
  - => Les fonctions peuvent aussi avoir des plages d'utilisation qui limitent une trop grande collection de données.

Les attaques en temps

- Il s'agit d'observer finement le temps d'exécution des programmes,
- Les variations sont fonctions des données, mais aussi du matériel et de l'implémentation,
  - Dans le cadre de l'exponentiation, les différents calculs peuvent conduire à disposer d'information sur la clé !
  - De même les optimisations peuvent conduire dans le cas du RSA avec CRT à déduire p et q (n = p * q).
- Des précautions doivent donc être prise dans l'implémentation des algorithmes.

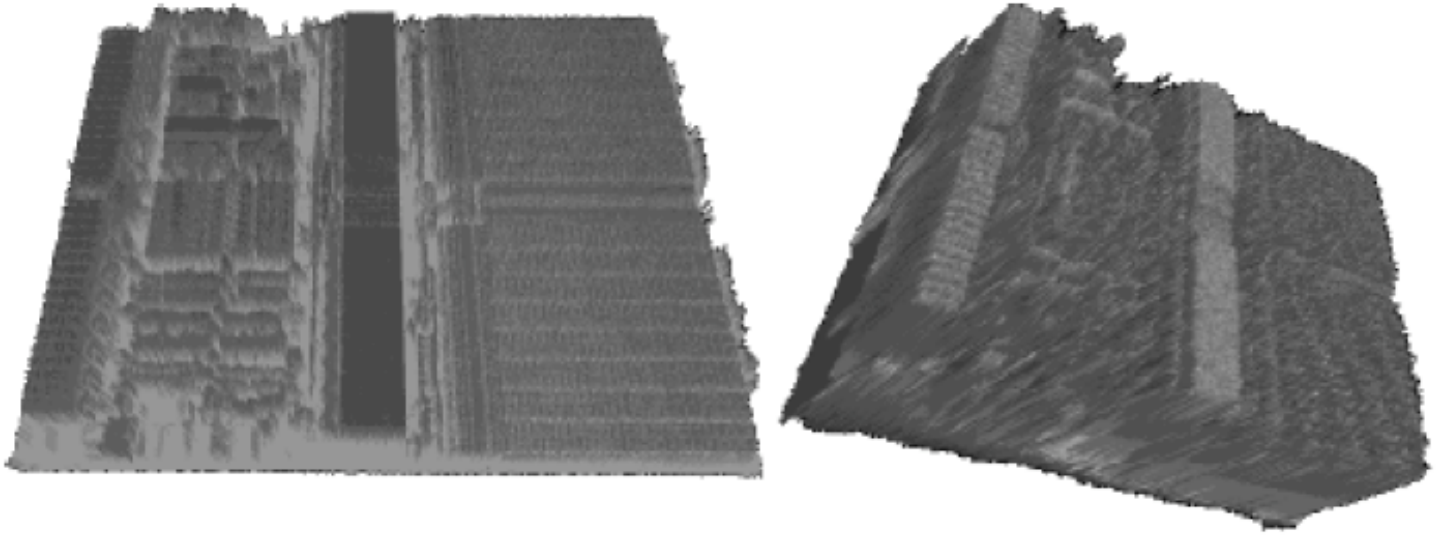
Les attaques électromagnétiques

- A l'origine utilisé pour les radiations des écrans...
- Comme pour la SPA et la DPA on a SEMA (Simple Electromagnetic Analysis) et DEMA (Differential Electromagnetic Analysis)



Signature 3D du composant...

- Thanks to UCL



Les attaques par injection de fautes

- Perturber le fonctionnement :
  - Modifier, forcer des valeurs via le probing dans les zones de données et/ou de programme.
  - Modifier et/ou forcer des valeurs spécifiques PC ou Virtual_Machine register.
  - Modifier l'horloge, la tension,...
    - exemple lecture compteur de code faux sur EPROM avec des valeurs hors spécifications.

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

Sécurité physique dans FIPS

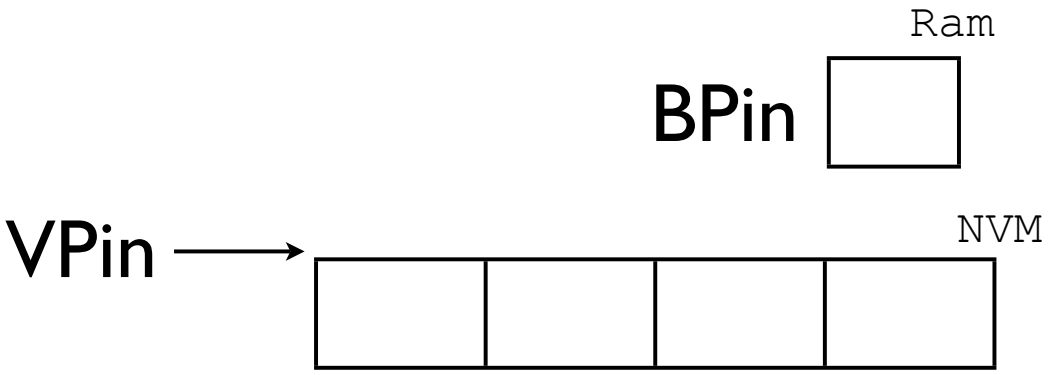
	General Requirements for all Embodiments	Single-Chip Cryptographic Modules	Multiple-Chip Embedded Cryptographic Modules	Multiple-Chip Standalone Cryptographic Modules
Security Level 1	Production-grade components (with standard passivation).	No additional requirements.	If applicable, production-grade 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*

Exemple pratique :Verify_Pin

```
fonction Verifier_Pin (string [ ]) : boolean;

% VPin variable globale contenant le Pin %
   Debut
      i = 1;
      BPin = vrai;
      TQ i<=4 et BPin faire
         si string [i ]<> VPin [i ] alors
            i++;
         sinon
            BPin = faux
         fsi
      FTQ
      Verifier_Pin = BPin;
   Fin
```

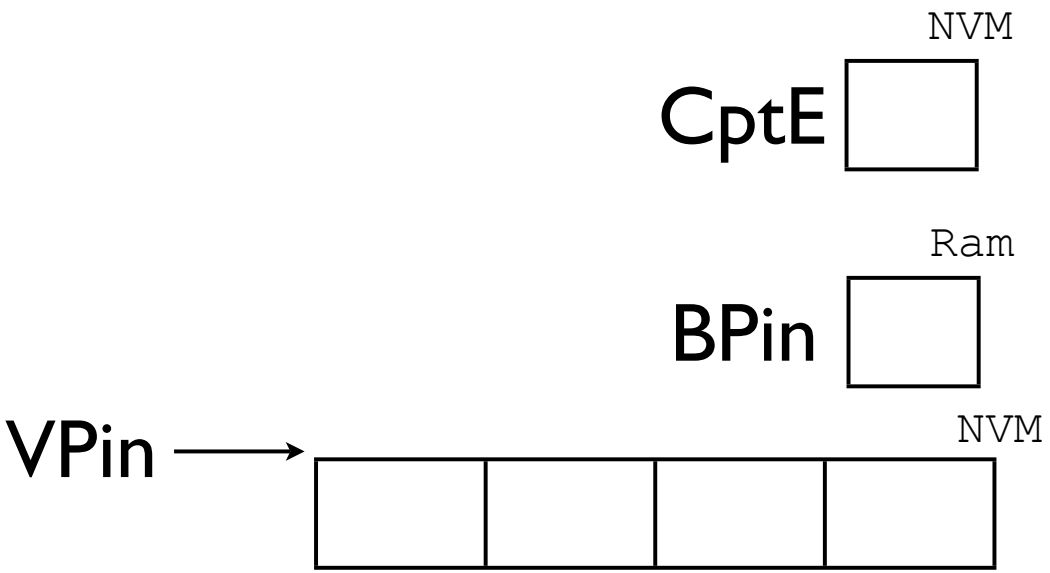
REMARQUES : ???



Verify_Pin (suite)

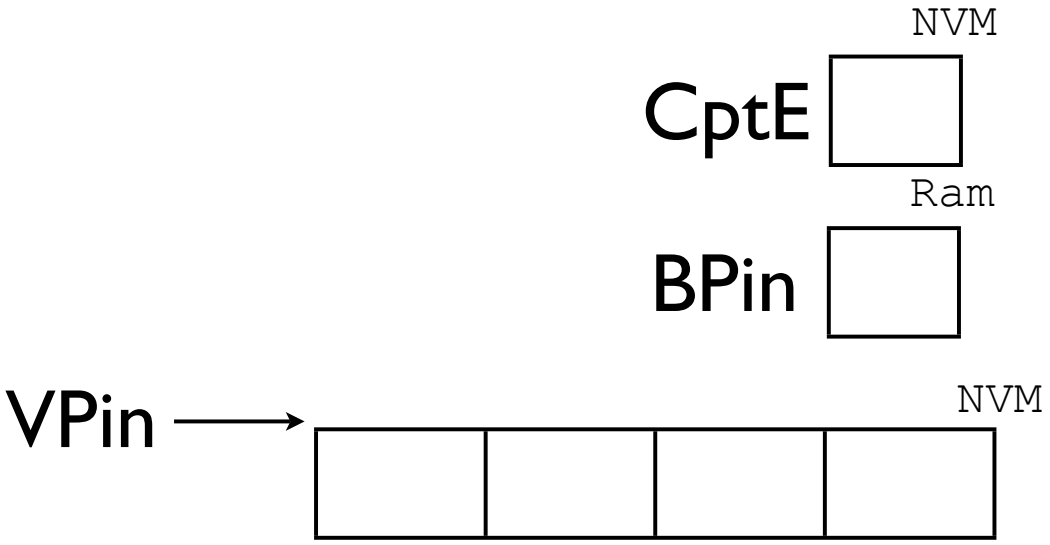
```
fonction Verifier_Pin (string [ ]) : boolean;

% VPin variable globale contenant le Pin %
   Debut
      i = 1;
      BPin = vrai;
      TQ i<=4 et BPin faire
         si string [i ]<> VPin [i ] alors
            i++;
         sinon
            BPin = faux
         fsi
      FTQ
      Verifier_Pin = BPin;
   Fin
```

REMARQUES :
ne prévoit pas l'attaque par dictionnaire



Verify_Pin (suite)

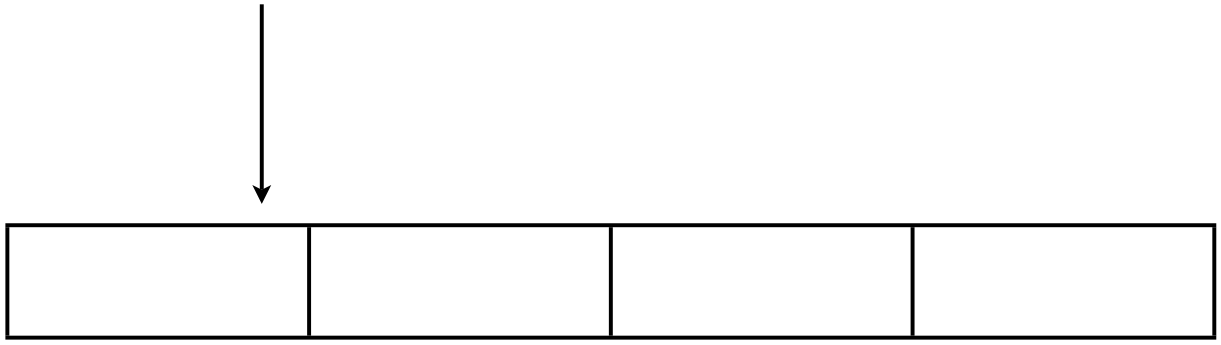
```
fonction Verifier_Pin (string [ ]) : boolean;

% VPin variable globale contenant le Pin + CptE [0..3 ]
% CptE = 0 pas erreur, 1 erreur,..
   Debut
      i = 1;
      BPin = vrai;
      TQ CptE < 3 et i<=4 et BPin faire
         si string [i ]<> VPin [i ] alors
            i++;
         sinon
            BPin = faux;
            CptE ++;
         fsi
         Verifier_Pin = BPin;
      FTQ
```



Verify_Pin (suite)

- Attaques sur :
  - CptE, consommation inc et/ou affectation
  - Temps sur les itérations et écriture CptE

```
si string [i ]<> VPin [i ] alors
  i++;
sinon
  BPin = faux;
  CptE ++;
```

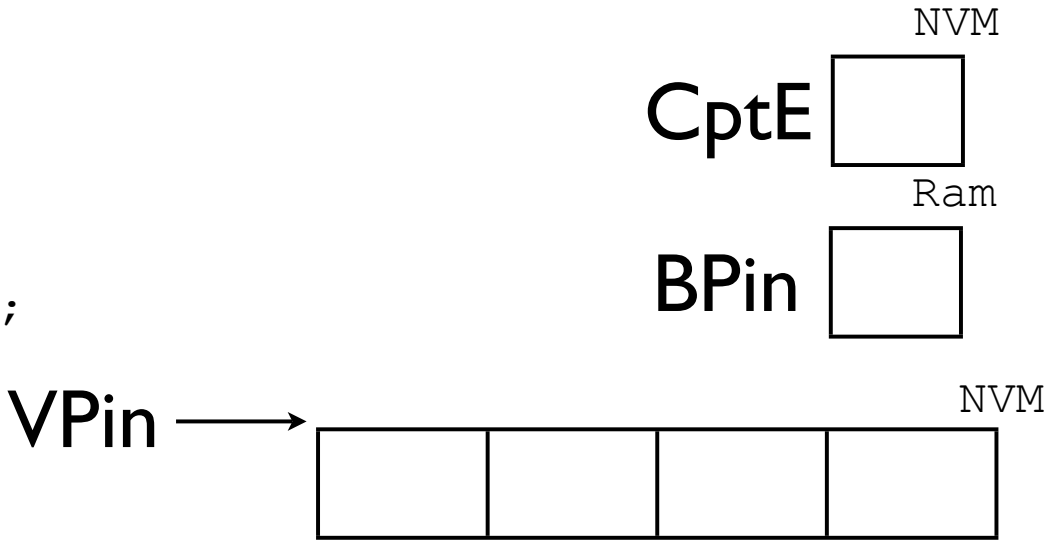


Verify_Pin (suite)

```
fonction Verifier_Pin (string [ ]) : boolean;

% VPin variable globale contenant le Pin + CptE [0..3 ]
% CptE = 0 pas erreur, 1 erreur,..
   Debut
      i = 1;
      BPin = vrai;
      TQ CptE < 3 et i<=4 et BPin faire
         si string [i ]<> VPin [i ] alors
            i++;
         sinon
            BPin = faux;
            CptE ++;
         fsi
         si BPin alors
            CptE = 0;
            Verifier_Pin = BPin;
      FTQ
```

Remarques :

temps est constant



Verify_Pin (fin)

- Un algorithme simple peut être victime d'attaques :
  - Simple (temps de réponse),
  - ou plus complexe temps, consommation,...
  - mais aussi probing sur CptE lecture du Pin !
- Voir le cercle...

Remarques

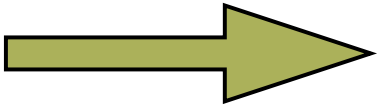
- Des attaques :
  - Intrusives non destructives,
  - Destructives mais aide au reverse pour d'autres attaques.
- Un élément compromis implique-t-il que tout le système est compromis ?

Biométrie

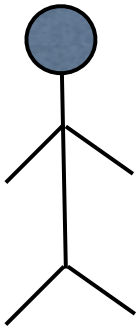
- L'identification est en générale basée sur le partage d'une information (le mot de passe-ce que je sais),

login : pparadin
pwd : *******

- La biométrie est basée sur une caractéristique de la personne (ce que je suis),



L'enregistrement

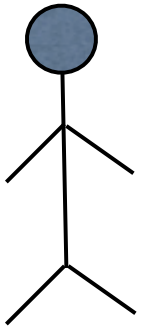


Caractéristique physique :
- empreinte digitale,
- forme de la main
- visage,
- iris de l'oeil,
- ...

. Valeur de référence :
Personne <=> ValRefPers

. Rangement dans SGBD

Le contrôle

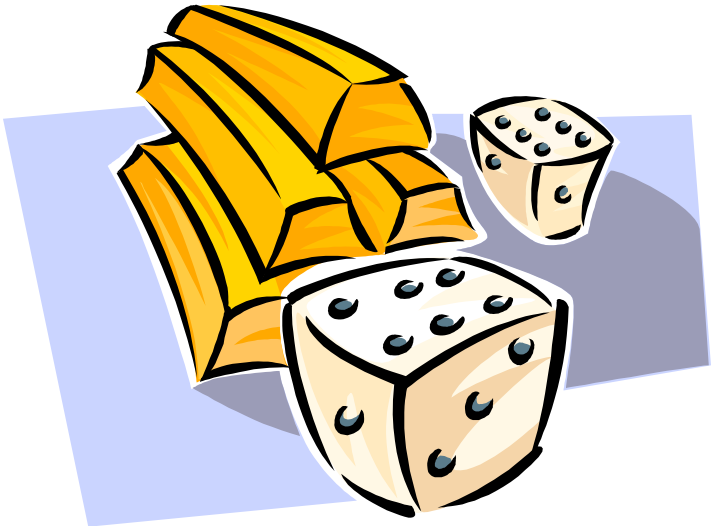


CAPTURE → *ValRefPers* →

Si COMPARE (*ValRefPers*) alors
    ACCEPTATION
sinon
    REJET
fsi;
% COMPARE (ValRefPers);
% retourne vrai si ValRefPers est dans la base

Le niveau de risque

- Deux taux sont significatifs de la qualité d'un système biométrique :
  - Le taux de rejet vrai
  - Le taux d'acceptation faux



Bibliographie

- Security Engineering: A Guide to Building Dependable Distributed Systems, Ross Anderson
- Tamper Resistance - a Cautionary Note

  Ross Anderson, Markus Kuhn, http://www.cl.cam.ac.uk/users/rja14/tamper.html
- http://www.cl.cam.ac.uk/users/rja14/tamper.html
- Paul Kocher
- Tamper Resistance Mechanisms for Secure Embedded Systems
  Srivaths Ravi, Anand Raghunathan and Srimat Chakradhar
  NEC Laboratories America, Princeton, NJ 08540
- http://perso.wanadoo.fr/fingerchip/index.htm (J-F Mainguet)
- Photos crédits : UCL, Ross Anderson,...
- Copyright : Les marques citées sont la propriétés des sociétés Nokia, ...