

# ***Certification de Sécurité de Premier Niveau***

***Une réponse pragmatique aux besoins du marché civil***

***SAR-SSI, La Rochelle – 20/05/2011***



## 1. Schéma de Certification de Sécurité de Premier Niveau (CSPN)

- Principes fondateurs
- Principes techniques
- Aspects réglementaires et Juridiques

## 2. Retours d'expérience

## Pourquoi un nouveau schéma ?

Des constats :

- ✓ Une offre très limitée de produits SSI dont la qualité a été attestée;
- ✓ Une absence de certification de logiciels libres ;
- ✓ Des coûts et délais d'évaluation/certification pas adaptés au marché des produits SSI civils;
- ✓ Les produits de sécurité ont des mécanismes de base de plus en plus complexe et hétérogènes.

### Les Critères communs (ISO 15408) :

- ✓ constituent un guide flexible et robuste de spécifications d'exigences et d'assurances de sécurité
- ✓ permettent de définir des profils de protection spécifiques à chaque domaine technologique :
  - *Cartes à puces*
  - *Firewall*
  - *OS*
  - *Base de données.../...*
- ✓ se focalisent sur la démarche méthodologique sans exigences spécifiques sur l'expertise technique
  - *ex: sans référence à un Profil de protection*

***Trop de libertés sur le choix de la cible (TOE)***

### 1. Exposé de la problématique

- Les Critères communs (ISO 15408) :
  - ✓ ne sont pas conçus pour évaluer des mécanismes cryptographiques

#### Evaluation of strength of TOE security functions (AVA\_SOF.1)

##### Objectives

The objectives of this sub-activity are to determine whether SOF claims are made in the ST for all probabilistic or permutational mechanisms and whether the developer's SOF claims made in the ST are supported by an analysis that is correct.

##### Application notes

SOF analysis is performed on mechanisms that are probabilistic or permutational in nature, such as password mechanisms or biometrics. Although cryptographic mechanisms are also probabilistic in nature and are often described in terms of *strength*, AVA\_SOF.1 is not applicable to cryptographic mechanisms. For such mechanisms, the evaluator should seek scheme guidance.

⇒ **application d'une méthodologie spécifique pour la cotation cryptographique**

## Un état de fait. Les produits de sécurité ont :

- ✓ des mécanismes de base :
  - de plus en plus complexes et hétérogènes
    - chiffrement/signature/effacement sécurisé/...
- ✓ souvent hors de portée du cadre normatif
  - stéganographie, watermarking, biométrie
- ✓ de plus intégré à l'OS et à l'environnement en général :
  - gestion du fichier swap, intégration dans un domaine, gestion des utilisateurs, ....
- ✓ de plus en plus de fonctions de haut niveau
  - Chiffrement logiciel+client VPN+intégration à une PKI

***L'évaluation CC peut devenir un vrai casse-tête***

***Sans délimitation préalable du périmètre fonctionnel de la TOE***



- **Un état de fait. les produits embarquant de la crypto comportent de nombreuses failles avant évaluation (Source NIST)**
  - Cryptographic Modules Surveyed (during testing)
    - 48.8% Security Flaws discovered
    - 96.3% Documentation Errors
  - Algorithm Validations (during testing)  
(DES, Triple-DES, DSA and SHA-1)
    - 26.5% Security Flaws
    - 65.1% Documentation Errors

### **Pour résumé, la mise en place du schéma CSPN :**

- ✓ répond à des besoins liés au marché de l'évaluation des produits de sécurité civils :
  - Temps contraint adapté au versionnage des produits
  - Coûts optimisés
- ✓ répond à des besoins purement techniques :
  - Restreindre le périmètre de la cible d'évaluation (Plus c'est complexe, moins c'est fiable et plus c'est cher) : L'évaluation d'un produit est réalisée par rapport à un ou plusieurs contextes d'emplois déterminés dans la cible
  - Privilégier l'expertise technique à la complétude méthodologique
  - Prendre en compte nativement l'évaluation des mécanismes crypto
- ✓ répond aux recommandations du député Lasbordes sur la SSI, notamment :
  - « Axe 3 : ...Développer la politique de certification et de qualification par une augmentation des produits certifiés et qualifiés et une réduction des délais et des coûts de certification. »
  - ✓ Remarque : la mise en place de la CSPN est un constat partagé par nos partenaires européens (anglais et allemand notamment)



## Le schéma CSPN repose sur

- ✓ Une catégorisation des produits
  - *pour limiter le périmètre fonctionnel*
- ✓ Une évaluation/certification en temps contraint (2 mois maximum)
  - *pour s'adapter au cycle de développement des produits du marché civil*
- ✓ Un budget borné pour le commanditaire
  - *pour augmenter le nombre de produits certifiés*
- ✓ L'application d'une méthodologie d'évaluation formalisée
  - *Privilégiant l'expertise technique (équilibre efficacité/conformité des fonctions)*
  - *Ne négligeant pas les aspects méthodologiques (inspirée des ITSEC)*
  - *adossée au décret 2002-535 relatif à la certification de la sécurité*

## Le schéma CSPN aboutit\* à :

- ✓ **L'émission in fine par l'ANSSI, d'un certificat à portée nationale\*\***

\*en cas de rapport d'évaluation positif

\*\* ce certificat sera imposé à court terme aux éditeurs pour les réponses aux appels d'offres publics




# Principes techniques (2)

- ✓ L'ANSSI délivre le certificat au vu d'un rapport d'évaluation fourni par un laboratoire qu'elle a agréé.
- ✓ L'évaluation consiste à vérifier que le produit est conforme à ses spécifications de sécurité, à coter les mécanismes de façon théorique, à recenser les vulnérabilités connues de produits de sa catégorie et à stresser le produit en temps contraint.
- ✓ Les CESTI en charge de ces évaluations sont agréés par l'ANSSI \* qui atteste à la fois leur compétence sur des briques technologiques spécifiques (rétro ingénierie, crypto, détection d'intrusion, sécurité des réseaux,...) et leur indépendance...
- ✓ L'ANSSI peut refuser une demande de certification si le produit ne s'y prête pas.

\* via un audit d'agrément tous les 18 mois

- L'évaluation doit notamment s'assurer de la conformité au RGS Référentiel Général de Sécurité de l'ANSSI disponible sur le site [http://www.ssi.gouv.fr/site\\_article38.html](http://www.ssi.gouv.fr/site_article38.html)
  - ✓ Exemple pour les mécanismes cryptos : 3 référentiels d'exigences

## DOCUMENTS CONCERNANT L'UTILISATION DE MÉCANISMES CRYPTOGRAPHIQUES DANS LES FONCTIONS DE SÉCURITÉ

Acronyme	Titre du document	Version
 RGS_B_1	Mécanismes cryptographiques : Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques	1.20
 RGS_B_2	Gestion des clés cryptographiques : Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques	1.10
 RGS_B_3	Authentification : Règles et recommandations concernant les mécanismes d'authentification	1.0

- Exemple de règles et de recommandations issues du RGS 1/3

RègleSGestSym-1	L'emploi d'une même clé pour plus d'un usage est exclu.
RègleSGestSym-2	Les éventuelles clés différenciées utilisées avec un mécanisme conforme au référentiel doivent être générées en utilisant un mécanisme de diversification conforme au référentiel.
RègleSGestSym-3	Les éventuelles clés dérivées doivent être générées en utilisant un mécanisme de diversification de niveau adapté au risque pesant sur le système global.
RecomSGestSym-1	L'emploi de clés communes est déconseillé.

➤ Exemple de règles et de recommandations issues du RGS 2/3

RègleInjectionCléLocale	L'injection d'une clé cryptographique générée localement doit bénéficier de moyens de protection conformes au référentiel. Ces moyens doivent garantir l'authenticité, l'intégrité et la confidentialité de la clé injectée.
RègleInjectionCléCentral	L'injection dans l'environnement de confiance de l'utilisateur final d'une clé cryptographique générée aléatoirement de façon centralisée doit bénéficier de moyens de protection conformes au référentiel. Ces moyens doivent garantir l'authenticité, l'intégrité et la confidentialité de la clé injectée.



## ➤ Exemple de règles et de recommandations issues du RGS 3/3

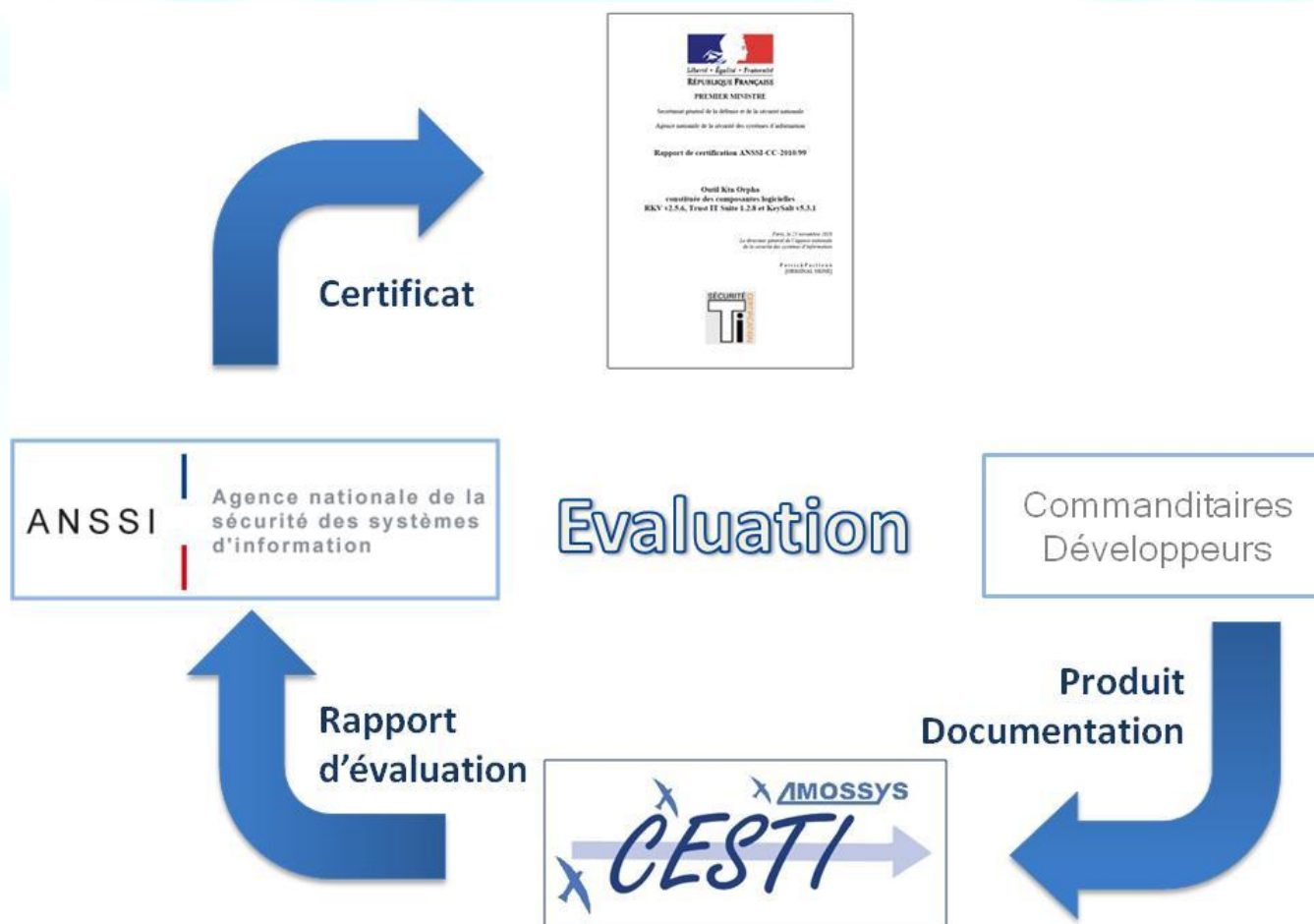
RègleEffacement	À la déconnexion d'une session authentifiée, si des éléments secrets ont été échangés lors de la phase d'authentification, ils doivent être effacés.
RecomMémoireVolatile	Il est recommandé que les éléments secrets échangés lors de la connexion d'une session authentifiée soient uniquement stockés en mémoire volatile et jamais sur un support magnétique.
RecomInactivité	Au cours d'une session authentifiée, il est recommandé d'incorporer un dispositif de déconnexion automatique en cas d'inactivité.
RègleAudit	Toute erreur survenant au cours d'une session authentifiée doit générer une trace d'alarme ne pouvant être modifiée ni effacée.
RecomAudit	Il est recommandé que toute transition d'état survenant au cours d'une session authentifiée génère une trace d'alarme ne pouvant être modifiée ni effacée.
RègleAuthentification	L'authentification d'un utilisateur auprès d'un SI distant doit faire intervenir un environnement de confiance local déverrouillé par l'utilisateur et réalisant, pour son compte, une authentification de machine à machine conforme au référentiel général de sécurité.



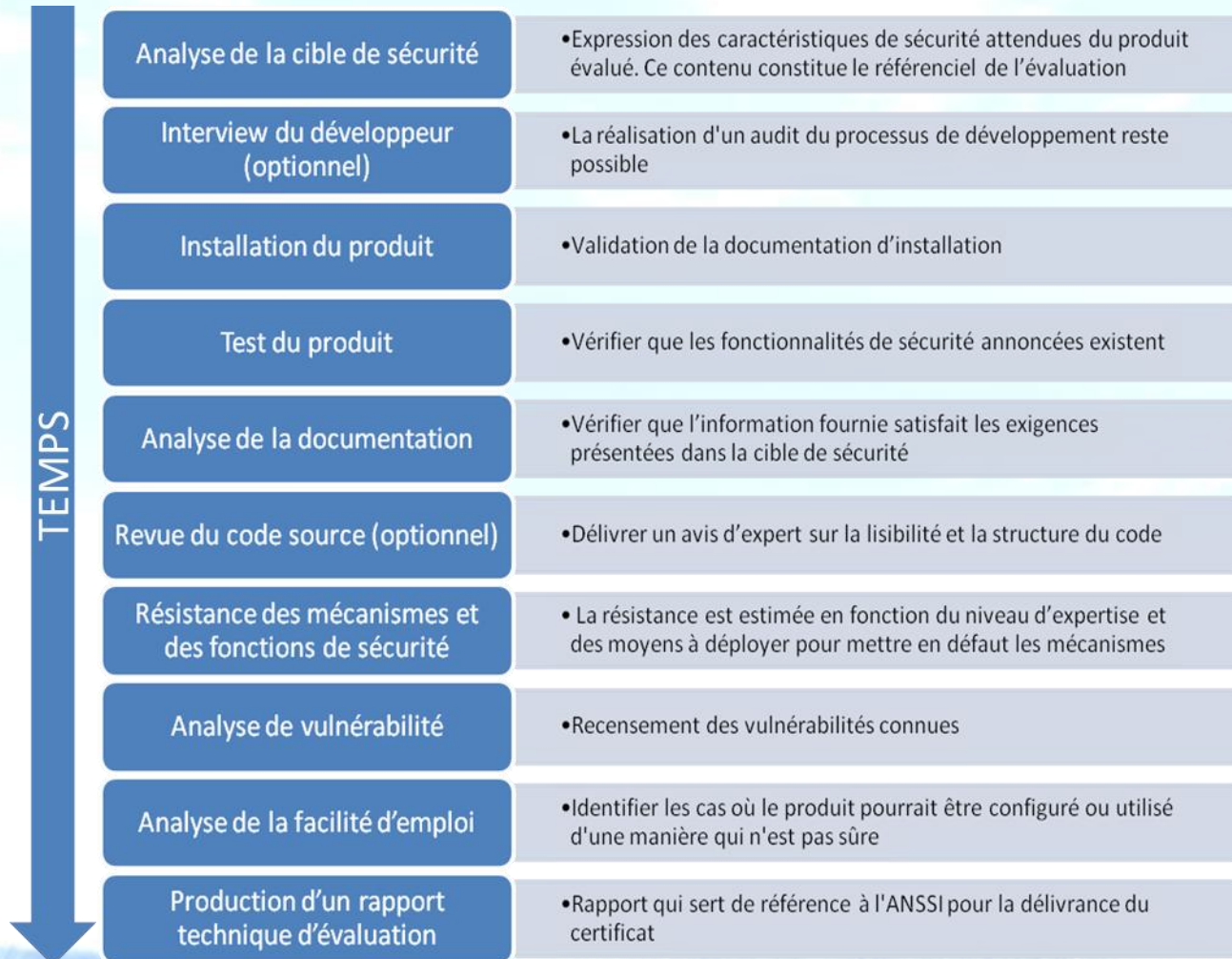
Catégories CSPN*
Détection d'intrusions
Anti-virus, protection contre les codes malicieux
Firewall
Effacement de données
Administration et supervision de la sécurité
Identification, authentification et contrôle d'accès
Communication sécurisée
Messagerie sécurisée
Stockage sécurisé
Matériel et logiciel embarqué

\* Une catégorie Set TOP Box (STB) est en cours de mise en place

# Processus de certification



# Démarche d'évaluation



# Positionnement du schéma CSPN

	<b>CERTIFICATION DE PREMIER NIVEAU</b>	<b>CERTIFICATION CC</b>	<b>QUALIFICATION STANDARD</b>
<b>Vérification des fonctionnalités de sécurité</b>	Évaluation réalisée par un laboratoire agréé	Évaluation reposant sur les Critères Communs réalisée par un centre agréé selon le décret 2002-535	Évaluation reposant sur les Critères Communs réalisée par un centre agréé selon le décret 2002-535
<b>Signification du label</b>	« Le produit a subi avec succès des tests fonctionnels et résiste aux vulnérabilités connues pour le type de service de sécurité qu'il fournit »	Conformité aux classes des CC mentionnées dans la cible	« le produit est recommandé pour les systèmes d'information sensible »
<b>Exigences sur les spécifications de sécurité</b>	Aucune	Aucune	Spécifications de sécurité validées par la l'ANSSI

# Positionnement du schéma CSPN

	<b>CERTIFICATION DE PREMIER NIVEAU</b>	<b>CERTIFICATION CC</b>	<b>QUALIFICATION STANDARD</b>
<b>Exigences sur les développements</b>	Aucune	Contrôle effectué lors de l'évaluation dans le respect des Critères Communs	Contrôle effectué lors de l'évaluation dans le respect des Critères Communs et vérifications contextuelles par la DCSSI
<b>Adossement à une réglementation</b>	Décret 2002-535	Décret 2002-535	Adossé au décret 2002-535
<b>Durée moyenne de l'évaluation</b>	2 mois	1 an	1 an



### Premières statistiques sur le schéma CSPN

- ✓ Schéma mis en place en 2008
- ✓ 13 produits certifiés à ce jour
- ✓ Nombres d'évaluation réalisées : au moins 25

**Ratio d'environ  $\frac{1}{2}$**



### **Le schéma ASD :**

- ✓ Schéma mis en place en 2010 pour les besoins de la défense
- ✓ Constitue un sur-ensemble de la CSPN
- ✓ Peut aboutir à la certification
- ✓ 30 évaluations réalisées en 18 mois

### Premières statistiques.

Résultat de l'évaluation	Pourcentage
Produit utilisable dans le contexte d'emploi (Certifiable en l'état)	20%
Produit utilisable avec recommandations dans le contexte d'emploi	40%
Produit ne pouvant être utilisé dans le contexte d'emploi	7%
Produit à proscrire (Vulnérabilités critiques dans le contexte d'emploi)	33%

### Typologie des évaluations techniques réalisées

Type	Nombre
Détection d'intrusions	5
Anti-virus	4
Effacement	1
Administration	3
Authentification	4
Communications	5
Messagerie	2
Stockage	1
Embarqué	2

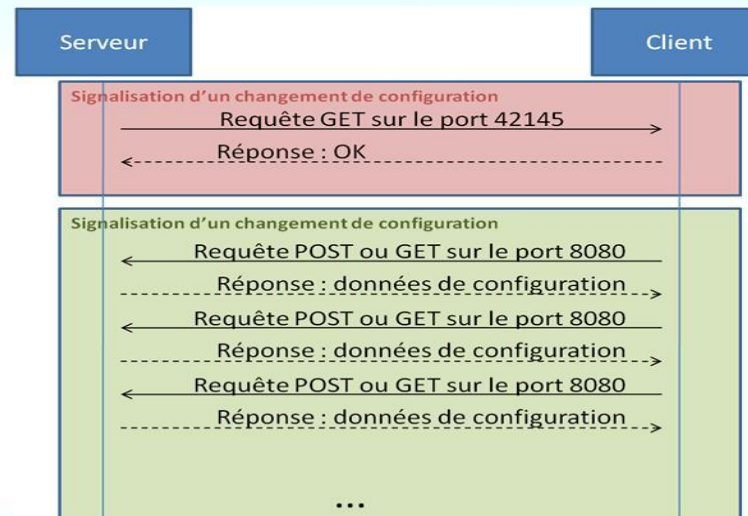
### Evaluation hors catégories CSPN

Type	Nombre
Virtualisation	2
Infra. d'accès	1
Editeur PDF	2
Serveur d'appli. Web	1
Test sécurité réseau	1
SAS Réseau	1

- ✓ Fait. **Le certificat CSPN fournit des recommandations d'emploi utiles pour les intégrateurs/admin système afin de palier les limitations éventuelles du produit.**
- ✓ Fait . **Les éditeurs de produits de sécurité se concentrent généralement sur leur cœur de métier**
  - Manque de prise en compte globale de la sécurité dans la conception
  - Nombreuses vulnérabilités :
    - ✓ de conception
    - ✓ de construction
    - ✓ en exploitation

### ✓ Exemple : nombreuses vulnérabilités critiques sur les antivirus

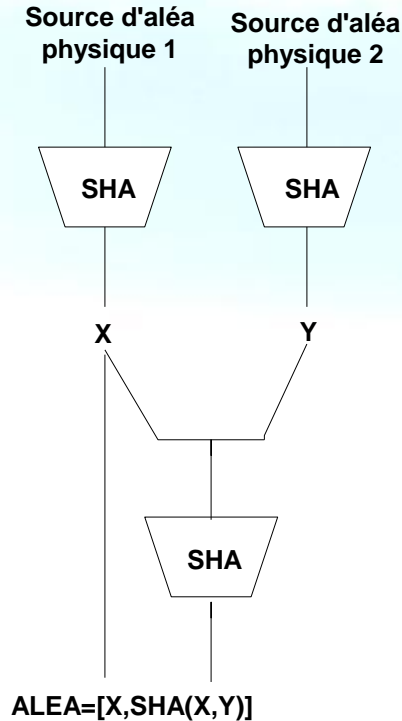
- mise à jour des bases non sécurisées (ex : utilisation de HTTP)
- désactivation de l'antivirus via un compte user
- moteur antiviral exécuté avec les droits systèmes
- autoprotection de la base virale défaillante





- ✓ Nombreuses vulnérabilités sur les interfaces d'administration web
  - Contrôle d'accès reposant uniquement sur l'interface web
  - Vulnérabilités classiques des web services (injection SQL, XSS,...)
  
- ✓ Nombreuses vulnérabilités dans l'implémentation de la crypto
  - Problèmes de gestion des secrets (pas d'effacement systématique des clés , pas d'enfouissement des clés de base,...)
  - Problèmes dans la génération d'aléa (cf. ci après)
  - Peu de prise en compte de l'état de l'art en cryptanalyse

- Exemple de vulnérabilités sur le GDA d'une PKI



Vulnérabilité de conception critique :  
Par construction la v.a ALEA n'est pas uniforme  $\rightarrow$  l'entropie n'est pas maximale

### – Exemple de vulnérabilités de conception sur le GDA d'une PKI

Une modélisation probabiliste de l'algorithme de génération de la graine est :

- Soit  $X$  la variable aléatoire modélisant le condensé de la source variable (de taille  $n$  bits) ;
- Soit  $Y$  la variable aléatoire modélisant le condensé de la source fixe ( de taille  $n$  bits) ;
- Soit  $Z$  la variable aléatoire modélisant la graine (de taille  $2n$  bits) ;
- On suppose naturellement que  $X$  et  $Y$  sont 2 variables aléatoires indépendantes de loi uniforme sur  $\{0, 1\}^n$  :
- Soit  $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$  la fonction de hachage représentant le SHA-1 ;

Par construction, on a  $Z = (X, f(X, Y))$  et la loi de  $Z$  est donnée par :

$$\mathbb{P}[Z = (u, v)] = \frac{1}{2^{2n}} \text{card}\{y \in \{0, 1\}^n / f(u, y) = v\}$$

Pour que la loi de la variable  $Z$  représentant la graine soit uniforme, il faudrait que la fonction de hachage  $f$  vérifie la propriété :

$$\forall (u, v) \in \{0, 1\}^{2n}, \text{card}\{y \in \{0, 1\}^n / f(u, y) = v\} = 1 \quad (2)$$

Cette propriété reviendrait à dire que  $f$  est bijective par rapport à sa première variable. A notre connaissance, cette propriété n'est pas satisfaite pour les architectures MDx, SHA-x.

- Exemple de vulnérabilité de construction :
  - ✓ retro spécification d'un mécanisme de génération de mots de passe

Ce code permet-il  
de générer des  
mots de passe  
uniformément?

```
static int generate_one_shot_password(char **buf)
{
    int i;
    char c, *mybuf;
    struct timeval tv;
    const int passlen = 8;
    const char letters[] = "01234567890abcdefghijklmnopqrstuvwxyz";

    gettimeofday(&tv, NULL);

    srand((unsigned int) getpid() * tv.tv_usec);

    mybuf = malloc(passlen + 1);
    if ( ! mybuf )
        return -1;

    for ( i = 0; i < passlen; i++ ) {
        c = letters[rand() % (sizeof(letters) - 1)];
        mybuf[i] = c;
    }

    mybuf[passlen] = '\0';

    *buf = mybuf;

    fprintf(stderr,
        "The \"%s\" password will be requested by \"%s register\"\n"
        "in order to connect. Please remove the quotes before using it.\n\n",

    return 0;
}
```

### – Exemple de vulnérabilité de construction:

- ✓ retro spécification d'un mécanisme de génération de mots de passe

Ce code source suggère le modèle probabiliste suivant :

- Soit  $X$  la variable aléatoire modélisant la source de pseudo-aléa `rand()`. Nous supposons en première approximation que cette *v.a.* est de loi uniforme sur l'ensemble  $\{0, 1\}^l$  où  $l = \text{RAND-MAX}$  ;
- Soit  $p$  le nombre modélisant la quantité `sizeof(letters) - 1`. Par défaut,  $p$  est égal à 37 ;
- Soit  $n$  le nombre modélisant la taille du mot de passe `passlen` ;
- Soit  $f : E = \{0, 1, \dots, 36\} \rightarrow F = \{0, 1, \dots, 9\} \cup \{0\} \cup \{a, b, \dots, z\}$  telle que pour tout  $i$ ,  $f(i) = i$  ;
- Soit  $Y$  la variable aléatoire telle que  $Y = f(X \bmod p)$ .

Soit alors  $(X_1, \dots, X_n)$  un échantillon de loi uniforme sur  $\{0, 1\}^l$ . Pour calculer la loi d'apparition du mot de passe, il convient d'étudier la loi de la variable  $Y_i = f(X_i \bmod p)$ .



### – Exemple de vulnérabilité de construction:

- ✓ retro spécification d'un mécanisme de génération de mots de passe

Soit  $r$  et  $q$  deux nombres entiers tels que  $2^l = qp + r$

$$\mathbb{P}[Y_i = 0] = \frac{2}{p} \left( 1 - \frac{2r - p}{2^{l+1}} \right) \quad (5)$$

et pour les  $r - 1$  successeurs  $y$  dans  $F$  (excepté 0) :

$$\mathbb{P}[Y_i = y] = \frac{1}{p} \left( 1 + \frac{p - r}{2^l} \right) \quad (6)$$

et enfin pour les  $p - (r + 1)$   $z$  derniers éléments de  $F$  (excepté 0) :

$$\mathbb{P}[Y_i = z] = \frac{1}{p} \left( 1 - \frac{r}{2^l} \right) \quad (7)$$

La loi de  $Y_i$  n'est donc pas une loi uniforme sur  $F$  et l'algorithme spécifié introduit naturellement un biais à l'équidistribution exprimé par les formules ci-avant.



### ➤ Exemple de Vulnérabilités de construction

- ✓ Une implémentation de l'exponentielle modulaire occultant les « timing attacks »

**Entrée :** un élément du groupe  $\alpha$  et un exposant  $x = x_{k-1} \dots x_1 x_0$  et un module  $n$

**Sortie :**  $\alpha^x \bmod (n)$

- **c:=1;** « accumulation du résultat »
- **For**  $i=k-1$  **downto** 0 **do**
  - $c := R_n(c, c);$
  - if  $(x_i=1)$  then  $c := R_n(c, \alpha);$
- **end;**

**Complexité :** nombre de multiplications =  $k + w(x) - 2$

### ➤ Exemple de Vulnérabilités de construction (suite)

- ✓ Principe des « Timing attacks », analyse des corrélations entre les bits de la clé et le temps de calcul :
  - si pour un certain nombre d'expériences le temps de calcul du  $k^{\text{ème}}$  bit est **rapide** alors  $x_k=0$
- ✓ Attaque pratique sur OPENSSL en 2003 (Brumley & Boney)
  - En effectuant 1/3 million de requêtes sur un serveur OPENSSL, on peut factoriser un module RSA de 1024 bits en 2 heures

### ➤ Exemple de Vulnérabilités de construction (suite)

- ✓ Contre mesure (exemple du déchiffrement RSA): Obfuscation du temps de calcul de type « blinding »

$$M = R^{-1}(C \cdot R^e)^d \bmod N$$

- Où R est un aléa
- ✓ L'overhead engendré est de 2 à 10%

### **Le schéma CSPN :**

- ✓ est suffisamment sélectif pour filtrer les produits les moins sûrs (analyse de vulnérabilités ~ AVA-VAN 3 CC)
- ✓ fournit une bonne vision du niveau de sécurité des produits de sécurité du marché civil
- ✓ offre de bonnes garanties de réussite pour un éditeur souhaitant viser une certification CC
- ✓ doit encore évoluer pour faciliter la reproductibilité des expertises
  - création d'une méthodologie spécifique à chaque catégorie
  - industrialisation des analyses de conformité et de vulnérabilités