

## SIGNATURE ELECTRONIQUE

### → **DEFINITION** :

Un système ou schéma de signature est un couple d'algorithmes :

- $\sigma()$ , pour la création de la signature.
- $v()$ , pour la vérification de la signature.

Ayant les propriétés suivantes :

1. La fonction de création  $\sigma()$ , dépend d'une quantité secrète  $K$  ( la clef ) et de  $M$ , le message à signer. Elle a pour résultat  $\sigma(M,K)$ .
2. Seul le détenteur de la clef  $K$ , peut créer la signature de  $M$ , en un temps raisonnable.
3. La fonction de contrôle publique  $v()$ , appliquée à la signature, a pour résultat  $M$ , et elle ne dépend d'aucune quantité secrète :  $v(\sigma(M,K)) = M$ .

### → **ATTAQUES SUR LES SCHEMAS DE SIGNATURE** :

- Cassage total : Obtention de la clef de création du système, où d'un système équivalent de signatures.
- Contrefaçon sélective : L'adversaire peut créer des signatures pour une certaine classe de messages.
- Contrefaçon existentielle : L'adversaire peut créer une signature pour au moins un message. Il a peu ou pas de contrôle sur le message dont il peut créer la signature.

## I – L’algorithme RSA utilisé en Signature :

### 1. ELLIS/RSA :

A ← B

B signe un message M et A le vérifie.

→ Etablissement de la clef :

- B choisit p,q ( premiers de l’ordre de  $10^{150}$  ) et calcule  $n=p.q$  et  $\varphi(n) = (p-1).(q-1)$ .
- B trouve k,k’ tels que  $k.k' = 1 \text{ mod } (\varphi(n))$ .

Public : n, k.

Privé :  $\varphi(n)$ , k’.

→ Création de la signature :

- B envoie (M,  $\sigma(M)=M^{k'}$ )

→ Contrôle de la signature :

- A calcule  $(\sigma(M))^k$  ( normalement  $\approx M^{k'.k} \text{ mod } (n) = M \text{ mod } (n)$  )
- A compare  $(\sigma(M))^k$  avec M.

(-) Inconvénients :

- Lenteur sur les systèmes embarqués .
- Longueur de la signature ( aussi longue que le document lui-même ).

### 2. FIAT/SHAMIR :

A ← B

B signe un message M ( |M| < 100 bits ) et A le vérifie.

→ Etablissement de la clef :

- B choisit p,q ( premiers de l’ordre de  $10^{150}$  ) et calcule  $n=p.q$ .
- B détruit p et q.
- B choisit t, entiers  $S_1, \dots, S_t < n$ , avec  $60 < t < 100$ .
- B calcule  $v_i = S_i^2 \text{ mod } (n)$ ,  $\forall i = 1..t$  .

Public : n,  $(v_i)$  pour  $i = 1..t$  .

→ Création de la signature :

- B calcule  $S = \prod_{i/m_i=1} (S_i)$  où  $m_i$  est le bit d'indice  $i$ , du message  $M$ .
- B choisit un aléa  $r < n$  et calcule  $y = r^2 \text{ mod } (n)$ .
- $\sigma(M) = (S.r, y)$ .

→ Contrôle de la signature :

- A calcule  $V = \prod_{i/m_i=1} (v_i)$ .
- A teste l'égalité :  $V.y = (S.r)^2 \text{ mod } (n)$ .

(-) Inconvénients :

- Longueur de la clef ( surtout s'il faut la stocker pendant une longue période ).

### 3. FSA/FSS :

A ← B

B signe un message M et A le vérifie.

→ Etablissement de la clef :

- B choisit  $p, q$  ( premiers de l'ordre de  $10^{150}$  ) et calcule  $n=p.q$  et  $\varphi(n) = (p-1).(q-1)$ .
- B trouve  $k, k'$  tels que  $k.k' = 1 \text{ mod } (\varphi(n))$ .

Public :  $n, k, v = k'^{-k} \text{ mod } (n)$ .

Privé :  $\varphi(n), k'$ .

→ Création de la signature :

- B choisit un aléa  $r < n$  et calcule  $x = r^k \text{ mod } (n)$ .
- B calcule  $e = h(M, x)$  ;  $h$  est une fonction de hachage.
- B calcule  $y = r.k'.e \text{ mod } (n)$ .
- $\sigma(M) = (y, e)$ .

→ Contrôle de la signature :

- A teste l'égalité :  $y^k.v = e^k \text{ mod } (n)$ .

(-) Inconvénients :

- Lenteur sur les systèmes embarqués .

## II – L’algorithme EL GAMAL utilisé en Signature :

### 1. EL GAMAL:

A → B

A signe un message M et B le vérifie.

→ Etablissement de la clef :

- A choisit un entier p ( premier  $\approx 10^{150}$  ). Et un entier r premier avec p-1.
- A choisit g avec  $1 < g < p-1$ .
- A choisit s avec  $1 < s < p-1$ .
- A calcule  $g^s$ .
- A calcule  $a = g^r \text{ mod } (p)$ .

Public : p,g,  $g^s$ .

Privé : s.

→ Création de la signature :

- A utilise l’algorithme euclidien étendu pour le calcul d’un entier b à l’aide de l’équation ( E ) :  $M = s.a + r.b \text{ mod } (p-1)$
- La signature  $\sigma(M) = (a,b)$ .
- A garde r secret.

→ Contrôle de la signature :

- $y = g^s$ .
- B teste l’égalité :  $y^a . a^b = g^M \text{ mod } (p)$

( – ) Inconvénients :

- p-1 n’étant pas premier, le calcul de b par l’équation ( E ), pose un problème d’inverse dans un anneau : Solution → DSA/DSS.

### 2. SCHNORR – DSA/DSS :

Cet algorithme de signature est une variante du protocole d’authentification portant le même nom.

A → B

A signe un message M et B le vérifie.

→ Etablissement de la clef :

- A choisit p premier et grand, a, s et calcule  $v = a^{-s}$ .

Public :  $(p,a,v)$ .

Privé :  $s$ .

→ Création de la signature :

- A choisit  $r < p$  et calcule  $x = a^r \text{ mod } (p)$ .
- A calcule  $e = h(M,x)$  ;  $h$  étant une fonction de hachage.
- A calcule  $y = r + s.e \text{ mod } (p-1)$ .
- $\sigma(M) = (x,y)$ .

→ Contrôle de la signature :

- B teste l'égalité :  $a^y.v^e = x \text{ mod } (p)$ .

(–) Inconvénients :

- Deux exponentiations à faire ...

? Peut-on ramener les deux exponentiations :  $a^y$  et  $v^e$  à une seule ?

→ OUI voir l'algorithme de signature RSA-BASED : FSA/FSS.