

# SYSTÈMES CRYPTOGRAPHIQUES

## Signature électronique

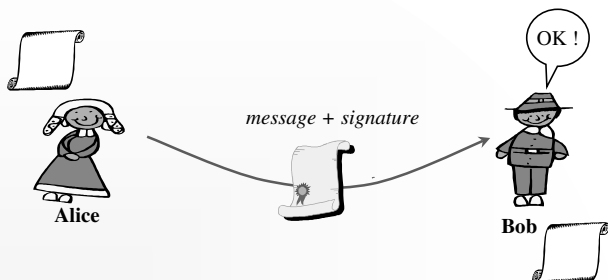
**Emmanuel Bresson**  
Ingénieur de l'Armement  
Département d'informatique  
École normale supérieure



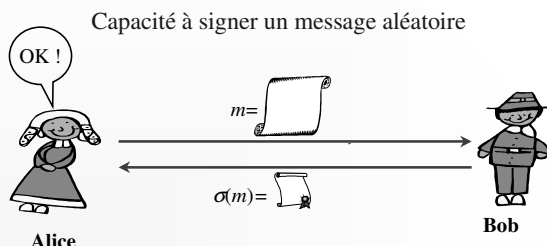
## But de la signature

- Reproduire les caractéristiques d'une signature manuscrite
- Assurer l'*authenticité* de l'expéditeur
  - ◆ Mécanisme essentiellement à clé publique
  - ◆ Paires de clés privées / clés publiques
- Assurer l'*intégrité* des données
  - ◆ Vérification du contenu d'un message
  - ◆ *Preuve* (non-interactive) de non-modification

## Signature : paradigme



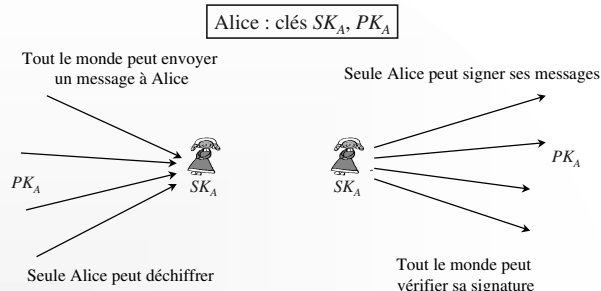
## Signature : Authentification



## Mise en œuvre de la signature

- Seul l'auteur du message doit pouvoir signer
  - ◆ Le signeur utilise sa **clé secrète**
  - ◆ Personne ne peut signer sans le secret
- Tout le monde doit pouvoir vérifier
  - ◆ La vérification utilise la **clé publique**
  - ◆ Pas de distinction entre les vérificateurs

## Signature / Chiffrement



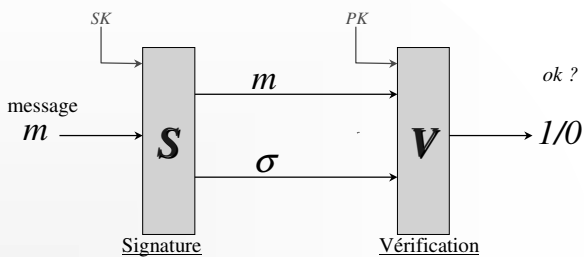
## Remarques

- Le mécanisme de vérification est publique
  - ◆ ≠ chiffrement, ≠ clé secrète
- ⇒ La sécurité inconditionnelle n'existe pas
  - ◆ cf. Chiffrement de Vernam (invérifiable)
- Il est possible de tester toutes les signatures potentielles, jusqu'à obtenir un test de validité concluant.
  - ◆ La sécurité sera toujours calculatoire

## Signature : définitions

- 3 algorithmes
- Génération de clés :
  - ◆  $KG(1^k) \rightarrow (SK, PK)$ . Probabiliste, produit une paire de clés secrète/publique.
- Signature :
  - ◆  $SIGN(m, SK) \rightarrow \sigma$ . Souvent probabiliste, calcule une signature d'un message.
- Vérification :
  - ◆  $VERIF(m, \sigma, PK) \rightarrow 1/0$ . Typiquement déterministe, vérifie la validité d'une signature.

## Algorithmes d'une signature



## Propriétés d'une signature

- Non répudiation:
  - ◆ Il est impossible de renier la signature d'un document qu'on a signé auparavant.
- Falsification impossible :
  - ◆ Il est impossible de produire une signature correcte sans la clé de signature

## Efficacité d'une signature

- Rapidité des algorithmes
  - ◆ Signature : souvent *off-line*, avec de la puissance.
  - ◆ Vérification : au vol, rapide, embarqué.
- Taille de la signature et de la clé
  - ◆ Taille de la clé publique : vérification
  - ◆ Taille de la signature pour un long message ?

## Sécurité des signatures

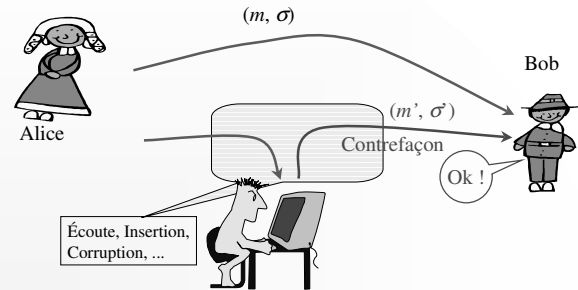
**Protocole de signature en présence d'un attaquant**  
 ⇒ Définir la sécurité de la signature

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>■ Que veut-il faire           <ul style="list-style-type: none"> <li>◆ Modifier un message signé</li> <li>◆ Se faire passer pour autrui</li> </ul> </li> <li>■ Quel est son but ?           <ul style="list-style-type: none"> <li>◆ Retrouver la clé</li> <li>◆ Falsifier une signature</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>■ De quoi dispose-t-il ?           <ul style="list-style-type: none"> <li>◆ Informations publiques</li> </ul> </li> <li>■ Que peut-il obtenir ?           <ul style="list-style-type: none"> <li>◆ Messages déjà signés</li> <li>◆ Nouvelles signatures</li> </ul> </li> </ul> |
|--|---|

## Modèle de sécurité

- Attaquant passif
  - Écoute le réseau
  - Ne touche pas aux messages
- Attaque « off-line »
- Attaque par dictionnaire
- Attaquant actif
  - Contrôle le réseau
  - Modifie les messages
  - Insère des messages
- Essentiellement « on-line »

## Sécurité des signatures



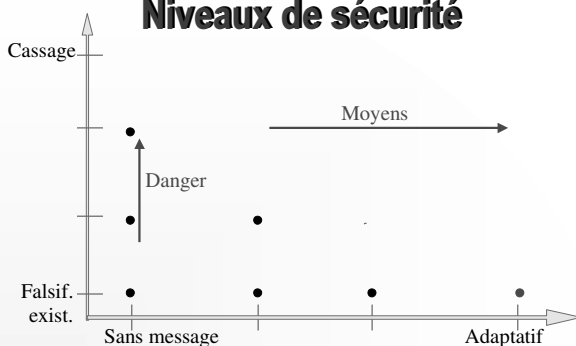
## Falsifications / cassages

- Cassage total :
  - ♦ L'attaquant retrouve la clé secrète
- Falsification universelle :
  - ♦ Capacité de signer n'importe quel message.
- Falsification sélective :
  - ♦ Capacité de signer un message au choix.
- Falsification existentielle :
  - ♦ Capacité à exhiber un message signé

## Attaques

- Attaque sans message :
  - ♦ Informations publiques seulement
- Attaque à messages connus :
  - ♦ Une liste de messages avec leur signature
- Attaque à message choisis :
  - ♦ Signature d'une liste de message
- Attaque à messages choisis adaptative :
  - ♦ Signatures de messages au fur et à mesure

## Niveaux de sécurité

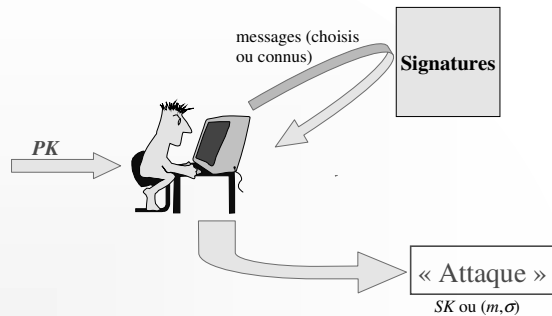


## Sécurité : définitions

- On choisit comme définition de sécurité la plus forte exigence possible :

absence de  
falsification existentielle,  
même sous une attaque adaptative  
à messages choisis.

## Sécurité des signatures



## Signature et chiffrement

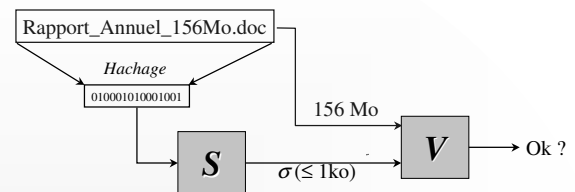
- Allier confidentialité et intégrité
  - ◆ Contrat confidentiel
  - ◆ Données sensibles facilement modifiables
- ~~Chiffrer puis signer « en aveugle »~~
- Danger : on ne « voit » pas ce qu'on signe
- Signer puis chiffrer

## Signatures de longs messages

- Signer des messages arbitrairement longs avec un schéma donné.
  - ◆ Message de 15 Mo avec RSA 1024 bits ?
- Taille de signature fixe
  - ◆ Indépendante de la taille du message signé.
- Sécurité « conservée »
  - ◆ Les contrefaçons doivent rester impossibles.

⇒ Utilisation de fonctions de hachage.

## « Hash-and-sign »



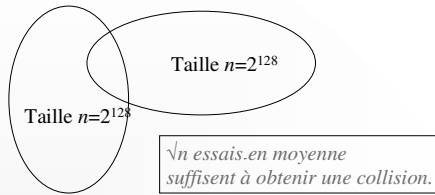
## Fonctions de hachage : définition

- Une fonction de hachage cryptographique
  - ◆ Rapide à calculer
  - ◆ Publique
  - ◆ Produisant une empreinte de taille fixe
- Caractérisée par l'absence de collisions
  - ◆ Assurer l'intégrité
  - ◆ Prévenir les falsifications

## Fonctions de hachage : sécurité

- Inversion difficile
  - ◆ Etant donné  $h(x)$ , trouver  $x$ .
- Seconde pré-image difficile
  - ◆ Etant donné  $x$  et  $h(x)$ , trouver  $y$  tel que  $h(x)=h(y)$ .
- Collisions difficile
  - ◆ Trouver  $x$  et  $y$  tels que  $h(x)=h(y)$ .

## Attaque des anniversaires



Dans une assemblée de 23 personnes, la probabilité que deux personnes soient nées le même jour est  $> 50\%$

## Fonctions de hachage classiques

- Fonctions MDx (« *message digest* »)
  - ♦ MD2, MD4, MD5 [Rivest].
  - ♦ Empreinte sur 128 bits
  - ♦ Partiellement attaquée, mais pas de collisions.
- Fonctions SHA (« *Secure Hash Algorithm* »)
  - ♦ SHA [NIST 1992] : remplacé par
  - ♦ **SHA-1** : empreinte de 160 bits
  - ♦ SHA sur 256, 384, 512 bits [2000]

## Etudes de signatures

- Basées sur la factorisation
  - ♦ La signature RSA
  - ♦ La signature Guillou-Quisquater
- Basées sur le logarithme discret
  - ♦ La signature ElGamal
  - ♦ La signature de Schnorr
  - ♦ Le standard DSS

## La signature RSA

$n=pq$ , produit de 2 grands nombres premiers  
 $e$  : exposant public  
 $d = e^{-1} \bmod \varphi(n)$  : exposant privé

- Signature du message  $m \in \mathbb{Z}_n$  :
 
$$\sigma = m^d \bmod n$$
- Vérification du couple  $(m, \sigma)$  :
 
$$\sigma^e \stackrel{?}{=} m \bmod n$$

## La signature RSA : faiblesse

La signature RSA utilisée telle quelle est

- existentiellement falsifiable par une attaque sans message
- universellement falsifiable par une attaque à message choisis

- Choisir  $\sigma \in \mathbb{Z}_n$
- Calculer  $m$  :
 
$$m := \sigma^e \bmod n$$

Choisir  $m \in \mathbb{Z}_n$   
 Faire signer  $2$  et  $m/2$  :  
 $\alpha := 2^d, \beta := (m/2)^d \bmod n$   
 Calculer  $\sigma$  :  
 $\sigma := \alpha\beta = m^d \bmod n$

## Attaque sur la signature RSA

- Si on factorise le module  $n$ 
  - ♦ On retrouve  $\varphi(n)$  et la clé secrète  $d$  : cassage total
- La factorisation est équivalente au cassage total
- Peut-on falsifier une signature sans factoriser ? (i.e. exhiber  $m^d$  pour  $m$  fixé)
  - ♦ Peut-être que oui, mais on ne sait pas [Bon99]

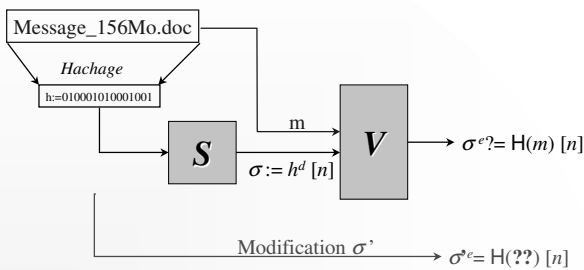
## Efficacité de RSA en signature

- Taille de clé : typiquement 1024 bits
- Taille de la signature : 1024 bits (pas d'expansion)
- Vérification
  - ♦ Avec un petit exposant public ( $e=3$  ou  $e=65537$ ), la vérification est (très) rapide
  - ♦ Bien adaptée aux environnements contraints (carte à puce, matériel embarqué)
- Inconvénient : temps de signature plus long

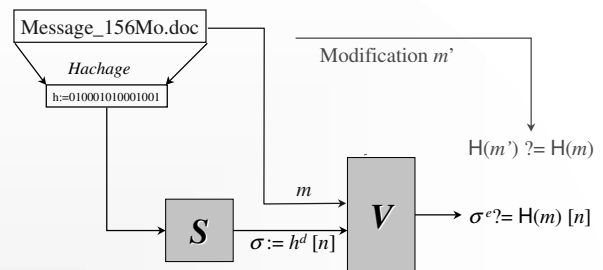
## Signature RSA avec hachage

- On hache le message avant de le signer
  - ♦ Signature efficace de messages longs
  - ♦ Prévention des falsifications
- Signature : message  $m$  signé avec  $SK=d$ 
  - ♦  $h := H(m)$
  - ♦  $\sigma := h^d \bmod n$
- Vérification : étant donné  $PK=e$  et  $(m, \sigma)$ 
  - ♦  $h' := H(m)$
  - ♦  $\sigma^e \stackrel{?}{=} h' \bmod n$

## Signature RSA avec hachage

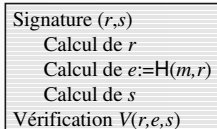
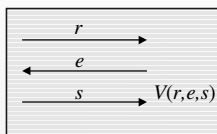


## Signature RSA avec hachage



## Preuve → Signature

- Toute preuve interactive « Zéro-Knowledge » (vérifieur honnête) est transformable en schéma de signature sûr
  - ♦ Fiat-Shamir (86), Pointcheval-Stern (96)



## Signature de Guillou-Quisquater [88]

$n=pq$ , produit de 2 grands nombres premiers  
 $e$  : exposant public,  $P$ , élément public  
 $S$  : élément secret tel que  $S^e=P \bmod n$

- Signature du message  $m$  :  $(y, z)$ 
  - ♦ Choisir  $x$  aléatoire et calculer  $y:=x^e [n]$
  - ♦ Calculer  $c:=H(y,m)$  et  $z:=xS^c \bmod n$
- Vérification du couple  $(m, y, z)$  :
  - ♦ Calculer  $c':=H(y,m)$
  - ♦ Vérifier  $z^e \stackrel{?}{=} yP^{c'} [n]$

## Signature de Guillou-Quisquater

- La vérification fonctionne :
  - ♦  $z^e = (xS^c)^e = x^e(S^c)^e = yP^e \pmod n$ .
- Utilisation de la fonction de hachage cryptographique :
  - ♦ Heuristique de Fiat-Shamir (1986).
  - ♦ Preuve ZK → Schéma de signature.
  - ♦ Signature de longs messages.

## Signature de Guillou-Quisquater

- Schéma probabiliste : le même message est signé de plusieurs façons différentes.
  - ♦ Mais  $x$  doit rester secret.
  - ♦ Le vérifieur n'a pas besoin de  $x$ .
- Sécurité :
  - ♦ Sécurité basée sur la factorisation.
  - ♦ Le cassage est équivalent à calculer une racine  $e$ -ième modulo  $n$ .

## Logarithme discret

- Problème du logarithme discret
  - ♦ Soit  $(G, \bullet)$  un groupe d'ordre  $n$ , engendré par  $g$ .
  - ♦ Problème du logarithme discret : étant donné  $h \in G$ , trouver l'unique  $m \in [0, n-1]$  tel que  $h = g^m$ .
- Signature basée sur le logarithme discret
  - ♦ On considère un groupe où ce problème est dur.
  - ♦ On montre qu'une falsification équivaut à un calcul de logarithme discret.

## Signature d'El Gamal [1985]

$G = \langle g \rangle$ , groupe généré par  $g$ , d'ordre premier  $p$   
 $x \in \mathbb{Z}_{p-1}$  : clé privée  
 $y = g^x \pmod p$  : clé publique

- Signature du message  $m \in \mathbb{Z}_p^*$  :  $(a, b)$ 
  - ♦ Choisir  $k$  aléatoire, premier avec  $p-1$
  - ♦ Calculer  $a := g^k \pmod p$
  - ♦ Calculer  $b$  tel que  $m = (xa + kb) \pmod{p-1}$  (Euclide !)
- Vérification du couple  $(m, a, b)$  :
  - ♦ Vérifier  $g^m \stackrel{?}{=} y^a a^b \pmod p$

## Signature d'El Gamal

- La vérification fonctionne :
  - ♦  $g^m = g^{xa+kb} = (g^x)^a (g^k)^b = y^a a^b \pmod p$
- Schéma probabiliste
  - ♦ La taille de la signature est deux fois la taille de  $p$
- Sécurité du schéma :
  - ♦ Supposée équivalente au logarithme discret
  - ♦ Trouver  $a$  et  $b$  vérifiant  $y^a a^b = g^m$  (fixé)  $\pmod p$
  - « ressemble » à : trouver  $x$  tel  $g^x = y \pmod p$ .

## Signature de Schnorr [1989]

$G = \mathbb{Z}_p$ ,  $g \in G$ , d'ordre premier  $q|p-1$   
 $x \in \mathbb{Z}_q$  : clé privée  
 $y = g^x \pmod p$  : clé publique

- Signature du message  $m$  :  $(r, s)$ 
  - ♦ Choisir  $u$  aléatoire, calculer  $r := g^u \pmod p$
  - ♦ Calculer  $e := H(m, r)$  et  $s := u - xe \pmod q$
- Vérification du couple  $(m, r, s)$  :
  - ♦ Calculer  $e' := H(m, r)$
  - ♦ Vérifier  $r \stackrel{?}{=} g^s y^{e'} \pmod p$

## Signature de Schnorr

- La vérification fonctionne :
  - ♦  $g^s y^e = g^{u-xe} (g^x)^e = g^{u-xe+xe} = g^u = r \pmod p$
- Fonction de hachage (Fiat-Shamir)
  - ♦ Signature de longs messages
- Schéma probabiliste
  - ♦ La taille de la signature est  $|p|+|q|$ . (Ex. 512+160)
- Sécurité du schéma :
  - ♦ Supposée équivalente au logarithme discret.
  - ♦ On n'a pas de preuve de sécurité formelle.

## La signature DSA

- DSA : Digital Signature Algorithm [1994]
  - ♦ Basé sur le mécanisme ElGamal
  - ♦ Modification par rapport au schéma ElGamal pour obtenir une plus grande efficacité.

$G = \langle g \rangle$ , groupe généré par  $g$ , d'ordre  $q$   
 $x \in \mathbb{Z}_q$  : clé privée  
 $y = g^x$  : clé publique

## La signature DSA

$\mathbb{Z}_p$ ,  $p$  premier,  $q$  un facteur premier de  $p-1$   
 $g$  est un élément d'ordre  $q$  dans  $\mathbb{Z}_p$   
 $x \in \mathbb{Z}_q$  : clé privée,  $y = g^x$  : clé publique

- Signature du message  $m$  :
  - ♦ Choisir  $u \in \mathbb{Z}_q^*$  aléatoirement  $\Rightarrow u$  secret !
  - ♦  $c := (g^u \pmod p) [q]$ ,  $c \neq 0$
  - ♦  $d := (m+xc)/u [q]$ ,  $d \neq 0 \Rightarrow (c,d)$
- Vérification du couple  $(m, c, d)$  :
  - ♦  $a := m/d \pmod q$ ,  $b := c/d \pmod q$
  - ♦  $(g^a y^b \pmod p) \stackrel{?}{=} c [q]$  et  $0 < c, d < q$  ?

## Signature DSA : validité

Signature du message  $m \in \mathbb{Z}_q$  :  $(c,d)$

$c := (g^u \pmod p) [q]$   $\Rightarrow$  NB : le vérifieur ne connaît pas  $u$  !

$d := (m+xc)/u [q]$

$a := m/d \pmod q$ ,  $b := c/d \pmod q$

$g^a y^b = g^a g^{xb} = g^{a+xb} = g^{(m+xc)/d} = g^u \pmod p$

Donc  $(g^a y^b \pmod p) \pmod q = c$

## Signature DSA : efficacité

- Avec le mécanisme de ElGamal :
  - ♦ Travail dans  $\mathbb{Z}_p$ , avec  $|p|$  environ 1024 bits
  - ♦ Comme pour le chiffrement, la signature comporte deux termes, soit au total 2048 bits
- Intérêt de DSA :
  - ♦ On travaille dans un sous-groupe d'ordre  $q$
  - ♦  $\langle g \rangle$  d'ordre  $q$ , avec  $|q|$  d'environ 160 bits
  - ♦ Signature de 320 bits
  - ♦ Application : cartes à puce.

## Signature DSA : remarques

- Signature probabiliste
  - ♦ On choisit un secret aléatoire  $u$
  - ♦ Plusieurs signatures pour un message
  - ♦ Le vérifieur n'a pas besoin de  $u$
- Que se passe-t-il si  $d = 0$  ?
  - ♦ Il faut diviser par  $d$  pour vérifier :  $a = m/d$ ,  $b = c/d$
  - ♦ La probabilité que  $d = 0 [q]$  est  $2^{-160}$ .
  - ♦ Pas de problème en pratique.

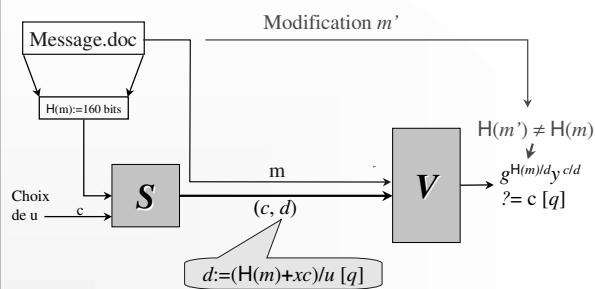
## DSA : historique

- La société RSA® avait vivement critiqué la sortie de ce standard
  - ◆ Soupçons envers le gouvernement
  - ◆ Moins de licences RSA ⇒ Baisse des revenus
- De nombreuses critiques attaquaient DSA
  - ◆ Inapte au chiffrement, et plus lent que RSA
  - ◆ Développé par la NSA, sans processus public
  - ◆ Avec une taille imposée (512) ⇒ étendu à 1024

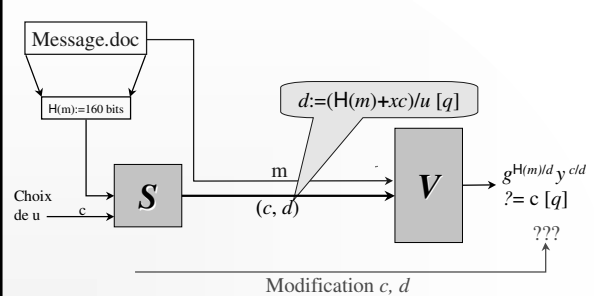
## Signature DSA : sécurité

- Contrefaçon existentielle possible :
- Choisir  $s$  et  $t$  dans  $Z_q^*$
- Calculer :
  - ◆  $c := g^s y^t \pmod p$
  - ◆  $d := ct \pmod q$
  - ◆  $m := cs/t \pmod q$
- $(c, d)$  est une signature valide de  $m$

## Signature DSA avec hachage



## Signature DSA avec hachage



## Sécurité : résumé

- Sécurité
  - ◆ Attaque active (messages choisis)
  - ◆ Contre falsifications existentielles
- Attention à la « malléabilité »
  - ◆ Propriété multiplicative de RSA
  - ◆ Propriétés d'El Gamal : attaques
- Fonctions de hachage
  - ◆ Empêchent de façonner un message
  - ◆ Signature de messages arbitraires

## Schémas : résumé

- Basés sur la factorisation
  - ◆ RSA (avec le hachage)
  - ◆ Guillou-Quisquater (transformation  $ZK \rightarrow \text{Sign}$ )
- Basés sur le logarithme discret
  - ◆ El Gamal
  - ◆ Schnorr ( $ZK \rightarrow \text{Sign}$ )
  - ◆ DSA : standard El Gamal + Schnorr

## Applications

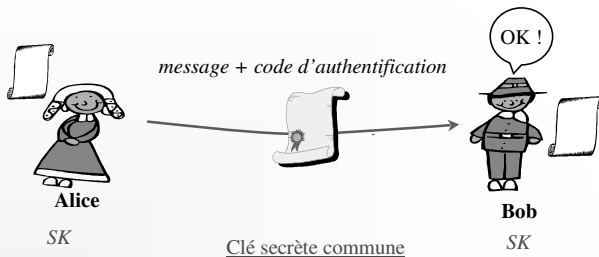
- Récupération de messages
- Codes d'authentification
- Certificats numériques
  - ◆ Monnaie électronique
  - ◆ Vote électronique

## Récupérations de messages

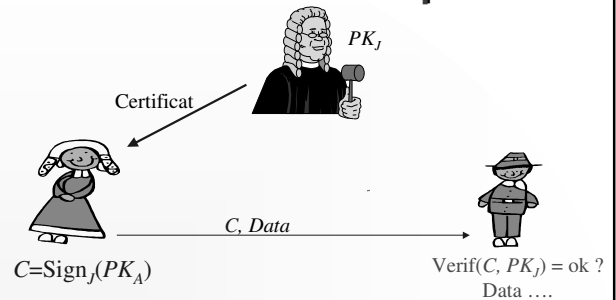
- ◆ Dans le cas de RSA, la vérification de la signature n'est rien d'autre que le calcul du message signé.
- ◆ On peut généraliser cette fonctionnalité à d'autres schémas
- Signature
  - ◆ Le message est transmis « dans » la signature et non plus à côté.
  - ◆ Il n'est pas chiffré pour autant !
- Vérification
  - ◆ Le vérifieur calcule le message original si la signature est valide

## Codes d'authentification (MAC)

- Analogues des signatures en clé secrète



## Certificats numériques



## Autres variantes possibles

- Signatures interactives
  - ◆ Nécessite le concours du signataire.
  - ◆ Contrôle de la diffusion d'un document.
- Signatures collectives
  - ◆ Une personne signe (anonymement) au nom d'un groupe (société...)
  - ◆ Plusieurs personnes signent en même temps (seuil minimal...)

## Conclusion

- La signature est un mécanisme inhérent à l'authentification des données, des clés, des identités.
- Une signature doit assurer l'authenticité et l'intégrité, ainsi que la propriété de non-répudiation
- Les variantes sont aussi nombreuses que les applications sont diverses.