D31: Protocoles Cryptographiques

Signatures numériques

Nicolas Méloni

Master 2: 1er semestre
(2014/2015)

Introduction

Objectif

Appliquer aux documents numériques le principe de la signature manuscrite:

- identifier un expéditeur
- controler l'intégrité du document
- assurer la non répudiation

Différences

- Pour les documents physiques, la duplication de la signature est supposée difficile
- Pour un documents numériques, la reproduction est très facile: une signature numérique doit être différente pour chaque document

Schéma de signature

Soient $\mathcal{P}$ l'ensemble des messages, $\mathcal{S}$ l'ensemble des signatures et $\mathcal{K}$ l'ensemble des clés possibles.
Un schéma de signature est la donnée de deux fonctions:

- $sig_K : \mathcal{P} \mapsto \mathcal{S}$
- $ver_K : \mathcal{P} \times \mathcal{S} \to \{\texttt{vrai},\texttt{faux}\}$

telles que $\forall K \in \mathcal{K}$

$$ver_K(x, y) = \begin{cases} \texttt{vrai si } y = sig_K(x) \\ \texttt{faux sinon} \end{cases}$$

Un exemple avec RSA

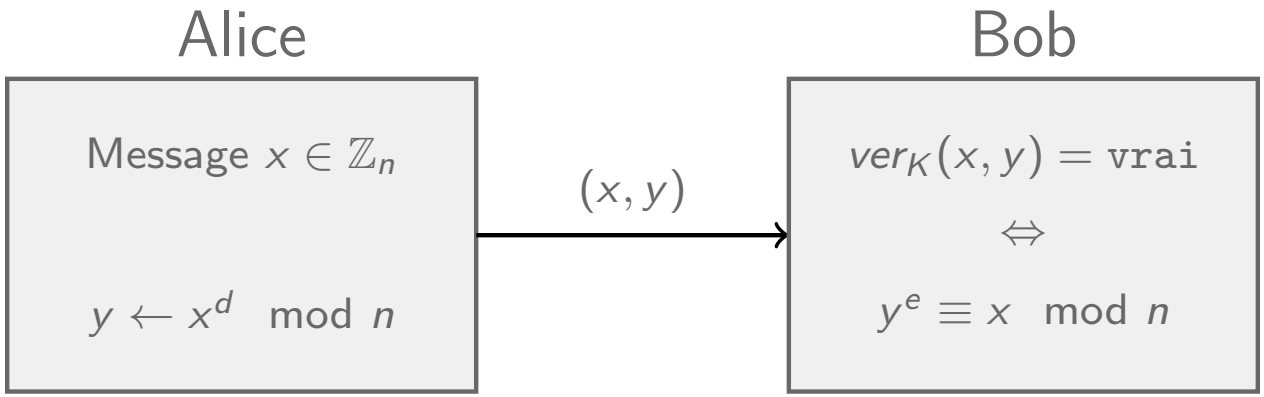
Schéma de signature RSA

Soit $K = (n, p, q, e, d)$ des paramètres RSA. On pose $\mathcal{P} = \mathcal{A} = \mathbb{Z}_n$. On définit alors les fonctions:

- $sig_K : x \mapsto x^d \mod n$
- $ver_K : (x, y) \mapsto \begin{cases} \texttt{vrai si } x \equiv y^e \mod n \\ \texttt{faux sinon} \end{cases}$

Un exemple avec RSA

- $K = (n, p, q, e, d)$ des paramètres RSA:



- Alice signe grâce à sa clé privée d
- Bob vérifie grâce à la clé publique d'Alice e

Un exemple avec RSA

Remarque 1

Le schéma possède bien les trois propriétés d'un schéma de signature.

Remarque 2

Il est facile de fabriquer des signatures valides:

- Bob choisi un y aléatoirement,
- calcule $x = y^e \mod n$
- y est alors bien un signature valide pour le message x

Remarque 3

Attention à l'utilisation conjointe avec un protocole de chiffrement: ne pas signer le chiffré.

Sécurité des schémas de signature

Modèles d'attaques

Attaque sans message

L'attaquant ne possède que la clé publique d'Alice, i.e., l'algorithme de verification.

Attaque à message(s) connu(s)

L'attaquant dispose de couples message/signature.

Attaque à message(s) choisi(s)

L'attaquant choisi une liste de message et obtient les signatures correspondantes.

Cassage total

L'attaquant retrouve la clé privée d'Alice et peu donc signer n'importe quel message à sa place.

Falsification sélective

L'attaquant peut produire une signature valide pour un message aléatoire.

Falsification existentielle

L'attaquant est capable de produire un couple message/signature valide.

Signatures et fonctions de hachages

- Le schéma RSA permet de signer des messages dans $\mathbb{Z}_n$.
- Pour de "vrais" messages, on utilise au préalable une fonction de hachage:

$$H : \{0,1\}^* \to \mathbb{Z}_n,$$

puis on signe le haché ($y = sig_K(H(x))$).

Attention au mélange!

Lorsque l'on combine différents outils cryptographiques, il est important de vérifier que l'on n'affecte pas la sécurité globale du système.

Scénario 1

Un attaquant possède un couple message/signature (x, y) valide. Il peut donc calculer $h = H(x)$. S'il est capable de trouver $x' \neq x$ tel que $H(x') = H(x)$, alors (x', y) est un couple message signature valide.

- L'attaquant a réussi une falsification existentielle à message connu.
- Pour résister, H doit être résistante à la seconde préimage.

Scénario 2

Un attaquant est capable d'obtenir deux message $x \neq x'$ tel que $H(x) = H(x')$. S'il parvient à faire signer $H(x)$ par Alice, alors $(x', sig_K(H(x)))$ est un couple message/signature valide.

- L'attaquant a réussi une falsification existentielle à message choisi.
- Pour résister, H doit être résistante aux collisions.

Schéma de signature El Gamal

Données

Soient p un nombre premier tel que le PLD est difficile sur $\mathbb{F}_p^*$ et $\alpha \in \mathbb{F}_p^*$ un générateur et $\beta = \alpha^k$.
On pose $\mathcal{P} = \mathcal{F}_p^*, \mathcal{A} = \mathbb{F}_p^* \times \mathbb{Z}_{p-1}$.
(p, α, β) est la donnée publique et k la donnée privée.

Signature / vérification

- $sig_K(x) = (\gamma, \delta)$ où $\gamma = \alpha^r \mod p$, $\delta = (x - k\gamma)r^{-1} \mod p - 1$ et $r \in \mathbb{Z}_{p-1}$ est généré aléatoirement.
- $ver_K(x, (\gamma, \delta)) = \texttt{vrai} \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \mod p$.

Elliptic Curve Digital Signature Algorithm

- Le schéma de signature El Gamal est la brique de base de DSA (Digital Signature Algorithm, 1991). Problème: $|p| \simeq 1024$ bits pour être sur, soit des signatures d'environ 3000 bits.
- Les courbes elliptiques permettent de diminuer la taille des objets manipulés.
- ECDSA a été approuvé en 2000

Elliptic Curve Digital Signature Algorithm

Données

- $E(\mathbb{F}_p)$ un courbe elliptique d'ordre q (premier), P, Q deux points de $E(\mathbb{F}_p)$ tels que $Q = [k]P$.
- $\mathcal{P} = \{0,1\}^*, \mathcal{A} = \mathbb{F}_q^* \times \mathbb{F}_q^*$.
- $H : \mathcal{P} \to \mathbb{F}_q^*$ une fonction de hachage.
-

$(p, q, P, Q, E(\mathbb{F}_p))$ est la donnée publique et k la donnée privée.

Signature

Signature d'un message $m \in \{0,1\}^*$:

- génère $r \in \{1, \ldots, q-1\}$ aléatoirement,
- calcule $[r]P = (x_r, y_r)$
- $u \equiv x_r \mod q$
- $s \equiv r^{-1}(H(m) + uk) \mod q$.

$$sig_K(m) = (u, s)$$

Vérification

Vérification du message signé $(m, (u, s))$:

- $w \equiv s^{-1} \mod q$
- $i \equiv wH(m) \mod q$
- $j \equiv wu \mod q$
- $(x_v, y_v) = [i]P + [j]Q$

$$ver_K(m, (u, s)) = \texttt{vrai} \Leftrightarrow u \equiv x_v \mod q$$

Faille classique de DSA/ECDSA

Hack de la PS3

Vérification de jeux PS3

- Sony utilise ECDSA pour certifier les jeux autorisés à tourner sur la PlayStation 3
- Sony signe les jeux avec sa clé privée
- Chaque PS3 possède la clé publique de Sony et peut donc vérifier la signature attachée à chaque jeu

Erreur d'implantation

- r n'est pas généré aléatoirement mais est une donnée statique

Signatures irrévocables

- Un schéma de signature doit être avant tout infalsifiable pour protéger le signataire
- Il peut être également désirable que celui-ci ne puisse pas révoquer un document signé

Schéma de signature irrévocable

C'est un schéma de signature proposant un mécanisme permettant au signataire de prouver qu'une signature (valide) n'est pas la sienne. Il est constitué de trois éléments:

- un algorithme de signature,
- un algorithme de vérification,
- un protocole de désaveux.

Schéma Chaum-van Antwerpen

Données

- p et q deux nombre premiers tels que $p = 2q + 1$
- α un élément d'ordre q dans $\mathbb{Z}_p^*$
- $1 \leq a \leq q - 1$ et $\beta \in \mathbb{Z}_p^*$ tels que $\beta \equiv \alpha^a \mod p$

(p, α, β) est la donnée publique et a la donnée privée

Schéma Chaum-van Antwerpen

Signature

Signature d'un message $x \in < \alpha >$ par Alice:

$$sig_K(x) = x^a \mod p$$

Vérification

Vérification du message signé (x, y) par Bob:

- Bob génère aléatoirement $e_1, e_2 \in \mathbb{Z}_q$
- Bob envoie $c \equiv y^{e_1}\beta^{e_2} \mod p$ à Alice
- Alice envoie $d \equiv c^{a^{-1} \mod q} \mod p$ à Bob

$$ver_K(x, y) = \texttt{Vrai} \Leftrightarrow d \equiv x^{e_1}\alpha^{e_2} \mod p$$

Schéma de Chaum-van Antwerpen

Protocole de désaveux:

- Bob génère aléatoirement $e_1, e_2 \in \mathbb{Z}_q^*$
- Bob envoie $c \equiv y^{e_1}\beta^{e_2} \mod p$ à Alice
- Alice envoie $d \equiv c^{a-1 \mod q} \mod p$ à Bob
- Bob vérifie que $d \not\equiv x^{e_1}\alpha^{e_2} \mod p$
- Bob génère aléatoirement $f_1, f_2 \in \mathbb{Z}_q^*$
- Bob envoie $C \equiv y^{f_1}\beta^{f_2} \mod p$ à Alice
- Alice envoie $D \equiv C^{a-1 \mod q} \mod p$ à Bob
- Bob vérifie que $D \not\equiv x^{f_1}\alpha^{f_2} \mod p$

La signature est un faux $\Leftrightarrow \left(d\alpha^{-e_2}\right)^{f_1} \equiv \left(D\alpha^{-f_2}\right)^{e_1} \mod p$