Fonctions de hachage

Gardons à l'esprit qu'un algorithme de recherche de collisions (resp. préimages, secondes préimages) est un algorithme probabiliste, qui n'a besoin que de réussir en moyenne.

Soit $H : \{0,1\}^* \to \{0,1\}^n$ une fonction de hachage résistante aux collisions. Soit H' la fonction suivante :

$$H' : \begin{cases} \{0,1\}^* & \to & \{0,1\}^{n+1} \\ x & \mapsto & \begin{cases} 0\|x \text{ si } |x| = n \\ 1\|H(x) \text{ sinon} \end{cases} \end{cases}$$

Où $\|$ est une concaténation.

Question 1. *Montrer que H' est résistante aux collisions.*

Question 2. *Montrer que H' n'est pas résistante aux préimages.*

Autour des définitions

Soit $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ un schéma de signature sûr sur l'espace de messages $\{0,1\}^n$. On définit un nouveau schéma de signature $(\mathsf{KeyGen}', \mathsf{Sign}', \mathsf{Verify}')$ utilisant deux paires de clés de signature / vérification $(\mathsf{pk}_0, \mathsf{sk}_0)$ et $(\mathsf{pk}_1, \mathsf{sk}_1)$.

Question 3. *Dans cette question on sépare le message m en deux et on signe ses moitiés :*

$$\begin{cases} m := m_0\|m_1 \\ \mathsf{Sign}'((\mathsf{sk}_0, \mathsf{sk}_1), m)) := \mathsf{Sign}(\mathsf{sk}_0, m_0)\,, \mathsf{Sign}(\mathsf{sk}_1, m_1) \\ \mathsf{Verify}'((\mathsf{pk}_0, \mathsf{pk}_1), m, (\sigma_0, \sigma_1))) := \mathsf{Verify}(\mathsf{pk}_0, m_0, \sigma_0) \wedge \mathsf{Verify}(\mathsf{pk}_1, m_1, \sigma_1) \end{cases}$$

Est-ce que ce schéma est sûr ?

On définit maintenant un schéma qui accepte si une des deux signatures est valide :

$$\begin{cases} \mathsf{Sign}'((\mathsf{sk}_0, \mathsf{sk}_1), m)) := \mathsf{Sign}(\mathsf{sk}_0, m)\,, \mathsf{Sign}(\mathsf{sk}_1, m) \\ \mathsf{Verify}'((\mathsf{pk}_0, \mathsf{pk}_1), m, (\sigma_0, \sigma_1))) := \mathsf{Verify}(\mathsf{pk}_0, m, \sigma_0) \vee \mathsf{Verify}(\mathsf{pk}_1, m, \sigma_1) \end{cases}$$

On va démontrer que ce schéma est sûr.

Question 4. *Soit $\mathcal{B}$ un adversaire dans le jeu EUF-CMA pour la signature* $(\mathsf{Sign}', \mathsf{Verify}')$. *On définit un adversaire $\mathcal{A}$ dans le jeu EUF-CMA pour la signature* $(\mathsf{Sign}, \mathsf{Verify})$*, qui joue le rôle de challenger pour $\mathcal{B}$.*

- *Initialisation : $\mathcal{A}$ reçoit la clé pk de $\mathcal{C}$. Iel tire un bit b au hasard, ainsi qu'une clé $(\mathsf{pk}', \mathsf{sk}')$, et définit : $\mathsf{pk}_b := \mathsf{pk}, \mathsf{pk}_{1-b} = \mathsf{pk}', \mathsf{sk}_{1-b} = \mathsf{sk}'$.*
- *Requêtes : lorsque $\mathcal{B}$ effectue une requête de signature sur le message m, $\mathcal{A}$ transfère la requête à $\mathcal{C}$ et reçoit $\sigma = \mathsf{Sign}(\mathsf{sk}, m)$. $\mathcal{A}$ renvoie alors à $\mathcal{B}$:*
  - $\sigma, \mathsf{Sign}(\mathsf{sk}_1, m)$ *dans le cas $b = 0$*
  - $\mathsf{Sign}(\mathsf{sk}_0, m)\,, \sigma$ *dans le cas $b = 1$*
- *Finalisation : $\mathcal{B}$ renvoie $(m, (\sigma_0, \sigma_1))$. $\mathcal{A}$ renvoie (m, σ_b).*

Montrer que :

$$\Pr\left[\mathsf{Verify}(\mathsf{pk}, m, \sigma_b) = 1\right] \geq \frac{1}{2} \Pr\left[\mathsf{Verify}'((\mathsf{pk}_0, \mathsf{pk}_1), m, (\sigma_0, \sigma_1))) = 1\right] \quad .$$

Conclure.

Ce type d'argument s'applique à plus que deux copies : $\mathcal{A}$ devine à l'avance sur quelle clé l'attaque va avoir lieu. Cela permet de prouver génériquement la sécurité en "multi-clés" ou "multi-utilisateurs" des schémas que l'on utilise.

DSA

L'algorithme de signature DSA ("*Digital Signature Algorithm*") a été standardisé par le NIST en 1991. Aujourd'hui on le retrouve plus couramment sous sa version ECDSA, utilisant des courbes elliptiques.

On considère un grand nombre premier p tel que $p-1$ est divisible par un nombre premier q de taille « moyenne » . Soit g' un générateur de $\mathbb{Z}_p^*$, et $g = (g')^{(p-1)/q}$. On a donc $g^q = 1 \mod p$.

On considère une fonction de hachage $H \ : \{0,1\}^* \to \mathbb{Z}_q$.

DSA

KeyGen

- $x \leftarrow U(\mathbb{Z}_q^*)$
- $\mathsf{pk}, \mathsf{sk} := g^x, x$

$\mathsf{Sign}(\mathsf{sk}, m)$

- Calculer $h = H(m)$
- $k \leftarrow U(\mathbb{Z}_q^*)$
- $r \leftarrow (g^k \pmod p) \pmod q$
- $s \leftarrow (h + \mathsf{sk}r)k^{-1} \pmod q$
- Renvoyer (r, s)

$\mathsf{Verify}(\mathsf{pk}, m, (r, s))$

- Calculer $h = H(m)$
- $a \leftarrow hs^{-1} \pmod q$
- $b \leftarrow rs^{-1} \pmod q$
- $v \leftarrow (g^a \cdot h^b \pmod p) \pmod q$
- Renvoyer 1 ssi $v = r$.

Question 5. *Prouver que la signature DSA est correcte.*

Question 6. *Peut-on réutiliser la valeur k pour plusieurs signatures ?*