

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

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

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

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

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

Est-ce que ce schéma est sûr ?

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

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

On va démontrer que ce schéma est sûr.

Question 4. Soit $\mathcal{B}$ un adversaire dans le jeu EUF-CMA pour la signature $(\text{Sign}', \text{Verify}')$. On définit un adversaire $\mathcal{A}$ dans le jeu EUF-CMA pour la signature $(\text{Sign}, \text{Verify})$, 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 (pk', sk') , et définit : $\text{pk}_b := \text{pk}, \text{pk}_{1-b} = \text{pk}', \text{sk}_{1-b} = \text{sk}'$.
- *Requêtes* : lorsque $\mathcal{B}$ effectue une requête de signature sur le message m , $\mathcal{A}$ transfère la requête à $\mathcal{C}$ et reçoit $\sigma = \text{Sign}(\text{sk}, m)$. $\mathcal{A}$ renvoie alors à $\mathcal{B}$:
 - $\sigma, \text{Sign}(\text{sk}_1, m)$ dans le cas $b = 0$
 - $\text{Sign}(\text{sk}_0, m), \sigma$ dans le cas $b = 1$
- *Finalisation* : $\mathcal{B}$ renvoie $(m, (\sigma_0, \sigma_1))$. $\mathcal{A}$ renvoie (m, σ_b) .

Montrer que :

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

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 $g^q = 1 \pmod p$.

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

DSA

KeyGen

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

Sign(sk, m)

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

Verify(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 ?