

Chapitre X : Fonctions de hachage, signatures digitales

1) Fonctions de hachage, définition, exemples

Une fonction de hachage est une fonction qui à un "message" m de longueur plus ou moins arbitraire associe $h(m)$ de longueur fixée, appelée l'empreinte de m (ou le condensé ou le haché de m).

Exemples : $h : \{0,1\}^* \rightarrow \{0,1\}^l$ où $\{0,1\}^*$ désigne l'ensemble des suites finies de 0 et de 1

$h : \{0,1\}^k \rightarrow \{0,1\}^l$ avec $k > l$ (on parle alors de fonction de compression).

On veut que h ait les propriétés suivantes :

- $h(m)$ est facile à calculer
- y donné, il est difficile de trouver m tel que $h(m) = y$
On parle de résistance à la préimage.
- Il est difficile de trouver $m_1 \neq m_2$ tels que $h(m_1) = h(m_2)$
On parle de résistance aux collisions.

On peut affaiblir cette dernière propriété en :

m étant donné, il est difficile de trouver $m' \neq m$ tel que $h(m) = h(m')$. On parle de résistance à la seconde préimage.

Exemple 1 : la fonction de compression de Chaum - Van Heijst - Pfitzmann

Soient p un premier impair tel que $q = \frac{p-1}{2}$ soit premier (p, q grands).

Soient α, β deux racines primitives modulo p (on a donc $\beta = \alpha^a \pmod p$ avec $\text{pgcd}(a, p-1) = 1$).

Soit $h : \mathbb{Z}/q^2\mathbb{Z} \rightarrow \mathbb{Z}/p\mathbb{Z}$ définie par :

si $m = x_0 + x_1 q$ avec $0 \leq x_0, x_1 \leq q-1$, $h(m) = \alpha^{x_0} \beta^{x_1} \pmod p$.

h sera une fonction de compression dès que $(\frac{p-1}{2})^2 > p$. On peut donc imposer $p \geq 7$. En pratique p est très grand.

Montrons que h est résistante aux collisions et à la préimage ①

• h est résistante aux collisions car si on connaît $m \neq m'$ tels que $h(m) = h(m')$ on peut calculer $a = \log_d(\beta)$ et donc résoudre un problème de logarithme discret dans \mathbb{F}_p^\times .

preuve :

Posons $m = x_0 + x_1 q$, $m' = x'_0 + x'_1 q$ avec $0 \leq x_0, x_1, x'_0, x'_1 \leq q-1$

(possible car $0 \leq m < q^2$). Alors

$$h(m) = h(m') \Rightarrow d^{a(x_1 - x'_1) + (x_0 - x'_0)} = 1 \pmod{p}$$

$$\Rightarrow a(x_1 - x'_1) = x'_0 - x_0 \pmod{p-1} \quad (1)$$

car d est racine primitive modulo p .

Soit $d = \text{pgcd}(x_1 - x'_1, p-1)$. Alors $d \mid x'_0 - x_0$.

$$(1) \Leftrightarrow a \frac{x_1 - x'_1}{d} = \frac{x'_0 - x_0}{d} \pmod{\mu} \text{ où } \mu = \frac{p-1}{d} \text{ et } \text{pgcd}\left(\frac{x_1 - x'_1}{d}, \mu\right) = 1$$

Cette dernière équation admet une unique solution modulo μ , a_0 , qui se relève en d solutions de (1) faciles à calculer :

$$a_0, a_0 + \mu, a_0 + 2\mu, \dots, a_0 + (d-1)\mu \pmod{p-1}$$

Mais $d \mid p-1$ donc $d = 1, 2, q$ ou $p-1$.

Si $x_1 = x'_1$, alors $x'_0 = x_0 \pmod{p-1}$ et $|x'_0 - x_0| \leq q-1 \Rightarrow x'_0 = x_0 \Rightarrow m' = m = x_0 \pmod{2q}$ Absurde.

Donc $x_1 \neq x'_1$ et $0 < |x_1 - x'_1| \leq q-1 \Rightarrow d \neq p-1$ et q .

Il n'y a donc que deux possibilités $d = 1$ ou 2 . Suivant la valeur de d , on a donc au plus deux possibilités pour a que l'on peut facilement tester. On trouve donc $a = \log_d(\beta)$.

• h est résistante à la préimage.

preuve :

Supposons qu'on dispose d'un algorithme qui, si on lui soumet y , trouve facilement m tel que $h(m) = y$.

Les q^2 messages possibles donnent $p-1=2q$ hachés et on peut montrer que chaque haché à un grand nombre

d'antécédents pour h (de l'ordre de $q/2$).

On prend un m quelconque, on pose $y = h(m)$ et on soumet y à l'algorithme. Il est hautement probable que le résultat soit différent de m et on aura trouvée une collision. Si cela échoue, on prend un autre m , etc. Comme h est résistante aux collisions, on a une contradiction.

Remarque 1 Ce raisonnement prouve que si $h : E \rightarrow F$ est une fonction de hachage telle que pour tout $y \in F$, $h^{-1}(\{y\})$ a un cardinal élevé, alors si h est résistante aux collisions, elle est résistante à la préimage.

Mais nous renons en TD des fonctions de hachage résistantes aux collisions mais pas à la préimage.

Remarque 2. Il s'agit d'une bonne fonction de hachage mais qui présente l'inconvénient d'être trop lente en pratique.

Exemple 2 : la construction de Damgård.

Soit E un ensemble fini.

Soient f_0, f_1 des bijections $E \rightarrow E$ à sens unique (difficiles à inverser). Une rencontre entre f_0 et f_1 est un couple $(a, b) \in E^2$ tel que $f_0(a) = f_1(b)$ et on dira que $\{f_0, f_1\}$ est une paire de permutations qui ne se rencontrent pas s'il est difficile de trouver une rencontre entre f_0 et f_1 .

Supposons que f_0 et f_1 ne se rencontrent pas.

Soit $\{0, 1\}^*$ l'ensemble des messages à hacher.

Soit $e \in E$ quelconque et soit

$$h : \{0, 1\}^* \rightarrow E$$

$$x = x_1 x_2 \dots x_k \mapsto f_{x_1} \circ f_{x_2} \circ \dots \circ f_{x_k}(e)$$

Alors h est résistante aux collisions et à la préimage.

Preuve :

Supposons qu'il soit facile de trouver une collision:

$$x = x_1 \dots x_k, x' = x'_1 \dots x'_k, x \neq x' \text{ et } h(x) = h(x') \text{ avec } k \geq l.$$

Si $x_i \neq x'_i$, $a = f_{x_i}^{-1}(h(x))$ et $b = f_{x'_i}^{-1}(h(x'))$ donnent une rencontre entre f_0 et f_1 .

Si $x_i = x'_i$, alors comme f_{x_i} est bijective alors $x_2 \dots x_k$ et $x'_2 \dots x'_k$ donnent une collision pour h .

On itère et soit on obtient une rencontre entre f_0 et f_1 , soit

x est un préfixe de x' et comme $x \neq x'$, $l > k$ en fait.

$$\text{Dans ce cas } f_{x_1} \circ f_{x_2} \circ \dots \circ f_{x_k}(e) = f_{x_1} \circ \dots \circ f_{x_k} \circ f_{x'_{k+1}} \circ \dots \circ f_{x'_l}(e)$$

$$\text{D'où } e = f_{x'_{k+1}} \circ \dots \circ f_{x'_l}(e)$$

et on obtient un antécédent de e pour f_0 ou f_1 .

Mais f_0 et f_1 sont à sens unique !

Donc h est résistante aux collisions.

Soit $y \in E$. Supposons qu'on puisse trouver $x_1 \dots x_k$ tel que $h(x_1 \dots x_k) = y$. Alors que $k=1$ ou $k > 1$, on a un antécédent de y pour f_0 ou f_1 , à savoir e ou $f_{x_2} \circ \dots \circ f_{x_k}(e)$.

Mais f_0 et f_1 sont à sens unique !

Donc h est résistante à la préimage.

Application : Prenons $E = \mathbb{F}_p^\times = \{1, 2, \dots, p-1\}$. Soit d une racine primitive modulo p et $c \in \mathbb{F}_p^\times$. Posons :

$$f_0(y) = d^y \pmod{p}$$

$$f_1(y) = c f_0(y).$$

f_0 et f_1 sont à sens unique (logarithme discret).

Supposons qu'il soit facile d'obtenir une rencontre

$$\text{entre } f_0 \text{ et } f_1 : d^a = c d^b \pmod{p}. \text{ Alors } d^{a-b} = c \pmod{p}$$

et on a trouvé $\log_d c$, donc résolu un problème de

(4)

logarithme discret. On peut donc utiliser f_0 et f_1 pour la construction de Damgård.

Ici on a une construction qui repose sur le logarithme discret mais on peut adapter à RSA, Rabin, etc.

2) L'attaque des anniversaires.

Soit h une fonction de hachage. On cherche à trouver une collision (on verra à quoi cela sert en fin de chapitre).

- Rappelons le paradoxe des anniversaires.

Soient $2 \leq n \leq 365$ personnes. Quelle est la probabilité pour que deux d'entre elles soient nées le même jour?

Prenons une personne. La probabilité pour qu'une deuxième personne ne soit pas née le même jour que la première est de $\frac{364}{365}$. La probabilité pour qu'une troisième personne ne soit pas née aux dates de naissance des deux premières est de $\frac{363}{365}$ et en itérant, la probabilité pour que les n personnes soient nées à des dates deux à deux distinctes est de

$$\frac{364}{365} \times \frac{363}{365} \times \cdots \times \frac{365-(n-1)}{365} = \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{n-1}{365}\right).$$

La probabilité recherchée est donc de :

$$1 - \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{n-1}{365}\right).$$

On en déduit que la probabilité pour que deux personnes soient nées le même jour est $\geq \frac{1}{2}$ dès que $n \geq 23$.

- On peut appliquer ce principe à la recherche de collisions pour $h: E \rightarrow F$ où F est fini, $|F| = M$.

On tire $x_1, x_2, \dots, x_m \in E$ au hasard deux à deux distincts.

La probabilité d'obtenir une collision $h(x_i) = h(x_j)$ avec $i \neq j$ est de $1 - \left(1 - \frac{1}{M}\right) \left(1 - \frac{2}{M}\right) \cdots \left(1 - \frac{n-1}{M}\right)$. (5)

Supposons n petit par rapport à M . Alors $1 - \frac{k}{M} \approx e^{-\frac{k}{M}}$ ($1 \leq k \leq n-1$)
 D'où la probabilité est approximativement $1 - e^{-\frac{n(n-1)}{2M}}$ et elle sera $\geq \frac{1}{2}$ si et seulement si $\frac{n(n-1)}{2M} \geq \log(2)$ donc approximativement $n^2 \geq 2M \log(2)$ ou encore $n \geq 1.17 \sqrt{M}$.

Supposons par exemple que les hachés s'écrivent sur 40 bits. La probabilité de trouver une collision sera $\geq \frac{1}{2}$ dès que l'on a environ $1.17 \cdot 2^{20}$ hachés soit un peu plus de 1 million.

Pour cette raison, on recommande des hachés de 160 bits minimum (AES utilise des hachés de 256 ou 512 bits).

3) Utilisation des fonctions de hachage

- Stockage des mots de passe. On ne stocke pas les mots de passe (trop encombrant et dangereux si un adversaire s'empare du stock). On stocke des hachés $h(m)$ de ces mots. Quand un utilisateur entre son mot de passe m , un algorithme calcule $h(m)$ et vérifie qu'il est dans le stock. Un adversaire qui s'empare du stock connaît les $h(m)$ mais pas les mots de passe si h est résistante à la préimage. Par ailleurs si h est résistante aux collisions il est peu probable que deux mots de passe aient le même haché.

- Dans le chiffrement que ce soit à clé publique ou secrète.
 Donnons un exemple (inspiré de OFB) dans le cas d'une clé secrète. Alice et Bob ont une clé secrète K . Les clés sont de la forme $M = [m_1, \dots, m_n]$ où les m_i sont des octets (bytes). Alice chiffré M en calculant :

$x_1 = L_8(h(K))$ $c_1 = m_1 \oplus x_1$	$[L_8 = 8 \text{ bits de gauche}]$ puis pour j de 2 à n , $x_j = L_8(h(K \parallel x_{j-1})), c_j = m_j \oplus x_j$.
---	--

Le chiffre est $[c_1, \dots, c_m]$.

Bob déchiffre en calculant la suite $(x_j)_{1 \leq j \leq n}$ puis $m_j = c_j \oplus x_j$.

[Exemple avec DES : $h : \{0,1\}^* \rightarrow \{0,1\}^{64}$].

Le problème est que la suite (x_j) est la même à chaque fois. Une façon de régler cet inconvénient : Alice choisit à chaque fois un x_0 d'initialisation et pose $x_1 = L_8(h(K \parallel x_0))$, la suite étant identique. Le chiffre est alors $[x_0, c_1, \dots, c_m]$. Dans les deux cas, si Eve intercepte le chiffre, elle ne peut pas calculer la suite (x_j) car elle ne connaît pas K , du moins si h est une bonne fonction de hachage.

• Signatures (objet du paragraphe suivant)

4) Signatures digitales

Plaçons nous dans le cadre de la clé publique.

Alice veut envoyer un message M à Bob avec une signature S qui permet à Bob d'être sûr que cela vient bien d'Alice. L'idée est la suivante : Alice fabrique S à partir de M et de sa clé secrète de telle sorte que Bob puisse vérifier à l'aide de la clé publique d'Alice.

Notre principe ici est que S est fabriquée à partir du clair M . On peut chiffrer après si besoin (parfois ce n'est pas utile comme dans le cadre de l'envoi d'un message officiel).

Dans tous les cas, avec ou sans déchiffrement, Bob reçoit le couple (M, S) et vérifie. L'idée est que l'on voit le chiffrement comme une enveloppe qui cache un message signé : on ne signe pas l'enveloppe ! Et on va voir qu'avec nos procédures, ce serait une mauvaise idée

de signer le chiffre de M. Donnons des exemples concrets.

- RSA : $S = M^{d_A} \pmod{n_A}$ (où n_A est le module d'Alice et d_A est sa clé secrète)

- Bob vérifie en calculant $S^{e_A} \pmod{n_A}$ et doit retrouver M.

(on voit ici que fabriquer la signature à partir du chiffre aurait été une mauvaise idée car alors la signature aurait été M !)

- Oscar veut envoyer M avec la signature d'Alice. Il doit trouver s tel que $S^{e_A} \pmod{n_A} = M$. Il est confronté au problème classique de RSA.

- El Gamal clé publique $(p, g, g^s \pmod{p})$, clé secrète s.

Alice tire au hasard k tel que $\text{pgcd}(k, p-1) = 1$.

- $S = (u, v)$ avec $u = g^k \pmod{p}$
 $v = (M - us)k^{-1} \pmod{p}$
où k^{-1} est calculé modulo $p-1$.

- Bob calcule $(g^s)^u u^v \pmod{p}$ et vérifie que c'est $g^M \pmod{p}$

$$\begin{aligned} \text{En effet } (g^s)^u u^v \pmod{p} &= g^{su} g^{k(M-us)k^{-1}} \pmod{p} \\ &= g^{su} g^{M-us} \pmod{p} \\ &= g^M \pmod{p}. \end{aligned}$$

- Oscar veut envoyer M avec la signature d'Alice mais pour calculer un v convenable, il a besoin de la clé secrète s d'Alice.

- D SA (digital signature algorithm). voir TD.

Toutefois en l'état il y a un problème, celui de la falsification existentielle. Un ennemi peut envoyer un couple (M, S) qui sera accepté par Bob.

Comment ? Prenons l'exemple de la signature RSA.

Oscar choisit S aléatoire et pose $M = S^{e_A} \bmod n_A$.

Il envoie (M, S) à Bob. Et Bob vérifie que $S^{e_A} \bmod n_A = M$!

Bien sûr, il y a peu de chance pour que $S^{e_A} \bmod n_A$ ait un sens mais on aimerait se prémunir contre de telles falsifications.

L'idée est d'avoir recours aux fonctions de hachage

S se calcule à partir de $h(M)$ et non de M , où h est une

fonction de hachage publique.

Bob connaît M donc calcule $h(M)$ et vérifie que S convient.

Exemple avec la signature RSA :

- Alice veut envoyer M à Bob. Elle calcule $h(M)$ puis $h(M)^{d_A} \bmod n_A = S$ et envoie (M, S) à Bob qui vérifie que $S^{e_A} \bmod n_A = h(M)$.
- Oscar veut envoyer M avec la signature d'Alice. Il est confronté au même problème que plus haut : il doit trouver S tel que $S^{e_A} \bmod n_A = h(M)$.
- Imaginons une autre situation. Il intercepte (M, S) envoyé par Alice et veut lui substituer (M', S) avec $M' \neq M$ (ce paravant ce n'est pas possible car seul M fonctionne) Pour cela il doit trouver M' tel que $h(M') = h(M)$ et $M' \neq M$. Mais ceci est impossible si h est résistante aux collisions ou même plus faiblement résistante à la seconde préimage.
- Enfin, Oscar veut forger une falsification existentielle. Comme plus haut, il calcule $S^{e_A} \bmod n_A = h(M)$. Il obtient le haché d'un M qu'il doit trouver, ce qu'il ne pourra pas faire si h est résistante à la préimage.

Remarque : nous verrons en TD comment procéder à une

falsification existentielle sur ElGamal (si on n'utilise pas de fonction de hachage)