TD - Construction d'une fonction de hachage

Une fonction de hachage h est une fonction de $E \subset \{0,1\}^*$ dans $F \subset \{0,1\}^m$:

$$h : E \subset \{0,1\}^* \longrightarrow F \subset \{0,1\}^m$$

où m est un entier fixé (par exemple $m = 128$ pour $h = \text{MD5}$).

Une fonction de hachage est dite **résistante aux collisions** si il est difficile (i.e. extrêmement coûteux) de trouver $(x, y) \in E^2$ avec $x \neq y$ tels que: $h(x) = h(y)$.
Cet exercice construit une telle fonction de hachage à partir d'une fonction à sens unique (ici le logarithme discret).

I. Construction d'une fonction de hachage : $\{0,1\}^{2m} \longrightarrow \{0,1\}^m$

Soit p un grand nombre premier tel que $q = \frac{p-1}{2}$ soit aussi premier. On note $\mathbb{F}_p = \mathbb{Z}/p.\mathbb{Z}$ et $\mathbb{F}_p^*$ le groupe multiplicatif $(\{1, 2, \ldots, p-1\}, \times_{\bmod p})$. On définit de même $\mathbb{F}_q$ et $\mathbb{F}_q^*$.

Soient α et β deux éléments **primitifs** (i.e. *générateurs*) de $\mathbb{F}_p^*$. On suppose que α, β et p sont publics (connus de tout le monde) et on définit h_1 par :

$$\begin{array}{rccl} h_1 : & \mathbb{F}_q \times \mathbb{F}_q & \to & F_p \\ & (x_1, x_2) & \mapsto & \alpha^{x_1}.\beta^{x_2} \mod p \end{array}$$

Soit λ l'entier de $\mathbb{F}_q^*$ égal au logarithme discret de β en base α : $\alpha^\lambda = \beta \mod p$.
Dans toute cette question, on suppose que λ n'est pas connu et extrêmement coûteux à calculer.

Pour montrer que h_1 est résistante aux collisions, on procède comme suit:

- On suppose que l'on connaît une collision pour h_1, i.e.
  $\exists (x_1, x_2, x_3, x_4) \in \{0, 1, \ldots, q-1\}^4$ tels que $(x_1, x_2) \neq (x_3, x_4)$ et $h_1(x_1, x_2) = h_1(x_3, x_4)$
- et on montre qu'on peut alors facilement calculer λ. Pour cela, on définit

$$d = \text{pgcd}(x_4 - x_2, p - 1).$$

Nota Bene. On rappelle que p et q sont premiers et que $p = 2q + 1$.

1. Quels sont les diviseurs de $p-1$? En déduire $d \in \{1, 2, q, p-1\}$.

$p-1 = 2q$ et q est premier donc ses diviseurs sont $\{1, 2, q, 2q = p-1\}$.
Comme d est un diviseur de $p-1$, on en déduit $d \in \{1, 2, q, p-1\}$.

2. Justifier $-(q-1) \leq x_4 - x_2 \leq q-1$; en déduire que $d \neq q$ et $d \neq p-1$.

Comme $0 \leq x_2, x_4 \leq q-1$, on a $-(q-1) \leq x_4 - x_2 \leq q-1$.
Or q est premier; on en déduit que $(x_4 - x_2)$ est premier avec q, donc $d \neq q$.

3. Montrer que $\alpha^{(x_1-x_3)} \equiv \beta^{(x_4-x_2)} \mod p$.

Evident : $\alpha^{x_1}\beta^{x_2} \equiv \alpha^{x_3}\beta^{x_4} \mod p \Longleftrightarrow \alpha^{(x_1-x_3)} \equiv \beta^{(x_4-x_2)} \mod p$

4. On suppose ici $d = 1$; montrer qu'alors $\lambda = (x_1 - x_3).(x_4 - x_2)^{-1} \mod (p-1)$.

Si $d = 1$, soit $u = (x_4 - x_2)^{-1} \mod (p-1)$: $u.(x_4 - x_2) = 1 + k.(p-1)$ Alors $\beta^{(x_4-x_2).u} \mod p \equiv \beta^{1+k(p-1)} \mod p \equiv \beta \mod p$ (d'après le théorème de Fermat).
Remplaçant dans 3., on obtient : $\beta = \alpha^{(x_1-x_3).u} \mod p$ soit $\lambda = (x_1 - x_3).u \mod p-1$, cqfd.

5. On suppose ici $d = 2$ et on pose $u = (x_4 - x_2)^{-1} \mod q$.
5.a. Justifier que $\beta^q = -1 \mod p$; en déduire $\beta^{u.(x_4-x_2)} = \pm\beta \mod p$.
5.b. Montrer qu'on a : $\lambda = u.(x_1 - x_3) \mod p-1$ ou bien $\lambda = u.(x_1 - x_3) + q \mod p-1$.

5.a. Comme $d = 2$ et $p - 1 = 2.q$, on a $x_4 - x_2$ premier avec q d'où $u.(x_4 - x_2) = 1 + k.q$.
D'où $\beta^{(x_4-x_2).u} \mod p \equiv \beta^{1+kq} \mod p \equiv \beta.(\beta^q)^k \mod p$. Or $q = \frac{p-1}{2}$ et β est un primitif mod p.
D'où $\beta^{p-1} = 1 \mod p$ et $\beta^q = \beta^{\frac{p-1}{2}} = -1 \mod p$. Finalement $\beta^{(x_4-x_2).u} = (-1)^k.\beta \mod p$, cqfd.
5.b. Remplaçant dans 3., on obtient : $\beta = \pm\alpha^{(x_1-x_3).u} \mod p$ soit $\beta = \alpha^{(x_1-x_3).u+\delta.q} \mod p$ avec $\delta \in \{0, 1\}$. On a donc $\delta = 0$, i.e. $\lambda = u.(x_1 - x_3) \mod p-1$ ou sinon $\delta = 1$, i.e. $\lambda = u.(x_1 - x_3) + q \mod p-1$, cqfd.

6. Conclure en donnant un algorithme qui prend en entrée une collision $(x_1, x_2) \neq (x_3, x_4)$ et qui retourne λ.
Majorer le coût de cet algorithme et en déduire que h_1 est résistante aux collisions.

D'après les questions précédentes, on a l'algorithme suivant :

```
AlgoCalculLogBeta( p, α, β, ; x1, x2, x3, x4 ) {
    q = (p − 1)/2;
    d = pgcd(x4 − x2, p − 1) ;
    if (d == 1) {
          u = (x4 − x2)^{-1} mod (p − 1);
          λ = (x1 − x3).u mod p − 1;
```

```
    }
    else {// ici d == 2
          u = (x_4 − x_2)^{-1}  mod q;
          λ = (x_1 − x_3).u  mod p − 1;
          if (ExpoMod( α, λ, p ) == −β) λ = λ + q ;
    }
    return λ ;
  }
```

Le coût se ramène à des opérations modulo $p-1$, p et q. Donc en $O(\log^{1+\epsilon} p)$, ce qui est faible même pour p grand (1024 bits par exemple). Autrement dit, si on connaît une collision pour h_1, alors on peut facilement calculer le logarithme discret de β, ce qui contredit l'hypothèse que λ est très coûteux à calculer.
Par suite, on en déduit que, sous l'hypothèse que λ est très coûteux à calculer, alors h_1 est résistante aux collisions.

II. Extension à une fonction de hachage : $\{0,1\}^* \longrightarrow \{0,1\}^m$

On suppose donnée une fonction de hachage $h_1 : \{0,1\}^{2m} \to \{0,1\}^m$ résistante aux collisions (comme celle de la partie I par exemple) :

$$\begin{array}{rccc} h_1 : & \{0,1\}^m \times \{0,1\}^m & \to & \{0,1\}^m \\ & (x_1, x_2) & \mapsto & h_1(x_1, x_2) \end{array}$$

A partir de h_1, on définit de manière récursive $h_i : \{0,1\}^{2^i m} \longrightarrow \{0,1\}^m$ par:

$$\begin{array}{rccc} h_i : & \left(\{0,1\}^{2^{i-1}m}\right)^2 & \longrightarrow & \{0,1\}^m \\ & (x_1, x_2) & \mapsto & h_1(h_{i-1}(x_1), h_{i-1}(x_2)) \end{array}$$

7. Soient $(x_1, x_2, x_3, x_4) \in \mathbb{F}_q^4$; expliciter $h_2(x_1, x_2, x_3, x_4)$ en fonction de h_1.

On a

$$\begin{array}{rccc} h_2 : & (\{0,1\}^m)^4 & \to & \{0,1\}^m \\ & (x_1, x_2, x_3, x_4) & \mapsto & h_1(h_1(x_1, x_2), h_1(x_3, x_4)) \end{array}$$

8. Montrer que h_2 est résistante aux collisions. **Indication :** *on pourra procéder par l'absurde en montrant que si l'on connaît une collision pour h_2 alors on peut facilement calculer une collision pour h_1.*

Si on a une collision sur h_2, alors $\exists x \neq y : h_2(x) = h_2(y)$. On a deux cas:

- soit $h_1(x_1, x_2) \neq h_1(y_1, y_2)$ ou $h_1(x_3, x_4) \neq h_1(y_3, y_4)$: alors comme $h_1(x_1, x_2), h_1(x_3, x_4)) = h_1(y_1, y_2), h_1(y_3, y_4))$ on a trouvé une collision sur h_1.
- Sinon, supposons par exemple $(x_1, x_2) \neq (y_1, y_2)$ (on peut s'y ramener par symétrie). Alors comme $h_1(x_1, x_2) = h_1(y_1, y_2)$, on a trouvé une collision sur h_1.

Comme on suppose que h_1 est résistante aux collisions, on en déduit h_2 résistante aux collisions.

9. Généraliser en justifiant que h_i est résistante aux collisions.

Par récurrence, on montre que si h_i est résistante aux collisions, alors h_{i+1} est aussi résistante aux collisions.
Ceci est vrai pour $i = 1$. Similairement à la question précédente, on montre que si on trouve une collision pour h_{i+1}, alors on construit aussi une collision pour h_i.
Comme h_1 est résistante aux collisions par hypothèse, alors h_i est résistante aux collisions pour tout $i \geq 2$.

10. Combien d'appels à la fonction h_1 sont effectués lors d'un appel à h_i ?
En déduire que le calcul du hachage d'une séquence de n bits a un coût $O(n)$.

Soit $C(i)$ le nombre d'appels à h_1 lors de l'exécution de h_i. On a $C(i) = 2.C(i-1) + 1 = 2^i.C(0) + \sum_{k=0}^{i-1} 2^k = 2^i - 1$.
Pour un mot de n bits, on appelle donc n/m fois h_1, dont le coût est en $O(m^{1+\epsilon})$. Par suite le coût est en $O(n.m^{\epsilon})$ ce que l'on peut assimiler à $O(n)$ puisque m est fixé donc constant.

11. Comment peut-on étendre cette méthode pour construire une fonction de hachage sans collision de $\{0,1\}^* \longrightarrow \{0,1\}^m$?

Si n est la taille du message, soit i tel que $2^i.m = n$, i.e. $i = \lceil log_2 \frac{n}{m} \rceil$. On hache le message avec h_i.
Ce procédé nécessite de connaître la taille du message, donc ne marche pas à la volée. Une autre alternative, plus efficace, est d'utiliser le protocole de Merkle-Damgard comme vu en cours.