École normale supérieure **2018-2019**
Département d'informatique

Introduction à la cryptologie
TD n° 8 : Correction.

Exercice 1 (Hachage pour preuve de travail)**.**

1. On choisit $m = d$ et on construit la fonction $H' : x \mapsto 0^d \parallel H(x)$. Cette fonction n'est clairement pas sûre pour les preuves de travail avec niveau de difficulté d (c'est le moins qu'on puisse dire). Par contre, étant donné une collision x, y telle que $H'(x) = H'(y)$, on déduit immédiatement une collision $H(x) = H(y)$ donc H' n'est pas moins résistante aux collisions que H.
2. Considérons une fonction H qui est sûre pour les preuves de travail avec niveau de difficulté d. On définit H' comme suit :

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

   Clairement cette fonction ne résiste pas aux collisions, puisque $H(0^n) = H(1^n)$. Cependant elle n'est pas moins sûre pour les preuves de travail que H, puisque les seules valeurs modifiées arrivent sur 1^n.
3. Si H résiste aux préimages, on peut construire H' qui résiste aux préimages et qui n'est pas sûre pour les preuves de travail avec exactement la même construction que dans la première question. Réciproquement, supposons qu'il existe H qui est sûre pour les preuves de travail. On construit H' comme suit :

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

   Clairement H' n'est pas du tout résistante aux préimages. Par contre, elle est tout aussi sûre pour les preuves de travail que H, puisque cette propriété ne porte que sur des entrées de la fonction de taille strictement supérieure à n.

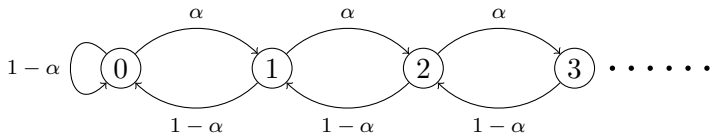
Remarque. Il y a deux points généraux qui valent la peine d'être relevés. D'abord, dans toutes les constructions précédentes, on ne crée pas un fonction de toute pièce, on part toujours d'une fonction supposée sûre pour une certaine propriété, et on la modifie. Ce n'est pas un hasard : on ne peut pas prouver qu'il existe de fonction qui soit sûre pour les propriétés précédentes. Par exemple, si P = NP, on peut facilement voir qu'il n'y a pas de fonction de hachage sûre pour les preuves de travail (en généralisant un peu pour que ça devienne une notion asymptotique), si la fonction de hachage se calcule en temps polynomial. Deuxième point, les raisonnements ci-dessus sont informels, parce qu'il est en fait impossible de définir formellement qu'une fonction de hachage fixe résiste aux collisions. En effet, étant donné une fonction $H : \{0,1\}^* \to \{0,1\}^n$, il existe nécessairement x, y de longueur $n+1$ tels que $H(x) = H(y)$, et il existe un algorithme qui trouve une collision en temps $O(1)$: c'est l'algorithme d'une ligne qui renvoie la paire (x, y). On ne sait pas construire cet algorithme efficacement, mais ce n'est pas quelque chose qu'on sait exprimer formellement (sans se mordre la queue)...

Exercice 2 (Signatures de Lamport, Merkle et Goldreich)**.**

1. Soit $\mathcal{A}$ un adversaire qui étant donné une clef publique du schéma, parvient à produire une signature en temps polynomial. On souhaite montrer que H n'est pas à sens unique. On choisit donc x uniformément aléatoirement dans $\{0,1\}^\lambda$. On tire y indépendamment de la même manière, et c uniformément dans $\{0,1\}$. On donne à l'adversaire la clef publique $pk_0 = H(x)$, $pk_1 = H(y)$ si $c = 0$, et $pk_0 = H(y)$, $pk_1 = H(x)$ sinon. L'adversaire renvoie x_b tel que $H(x_b) = pk_b$. Si $b = c$, il a donc trouvé une préimage de $H(x)$, et l'évènement $b = c$ a probabilité 50%, puisque c est uniforme et que $H(x)$, $H(y)$ sont par ailleurs indistinguables (ils proviennent de la même distribution). En utilisant $\mathcal{A}$ de cette manière, on obtient un algorithme qui trouve une préimage de $H(x)$ pour x uniforme en temps polynomial avec probabilité de succès 50% (moyenne à la fois sur x, et sur le choix de tous les aléas consommés par l'algorithme), ce qui est une probabilité non-négligeable, donc H n'est pas à sens unique.
   Remarque. On traite « non-négligeable » de manière informelle. Une définition formelle est « bornée inférieurement par l'inverse d'un polynôme (en le paramètre de sécurité, ici λ) ».
2. D'abord, on crée λ instances indépendantes du schéma précédent, numérotées de 1 à λ. Chacune peut nous servir à signer une fois un bit. Pour signer un message de longueur quelconque, comme pour une signature classique, on commence par hacher le message. On obtient un haché de longueur fixe $H(m) \in \{0,1\}^\lambda$. On signe chaque bit de ce haché (dans l'ordre) avec l'instance correspondante du schéma de signature à 1 bit. La longueur d'une signature est λ^2 bits. La clef publique est de taille $2\lambda^2$.
3. Pour signer ℓ messages, on peut utiliser ℓ instances du schéma de la question précédente. Le problème (enfin, un des problèmes) est que la taille de la clef publique est de $2\ell\lambda^2$ bits. Pour compacter un peu, on peut organiser les ℓ clefs publiques comme les feuilles d'un arbre de Merkle de hauteur $\log \ell$ (on va supposer que ℓ est une puissance de 2). La clef publique devient alors simplement la racine de cet arbre, de longueur λ bits. Pour signer un message, on choisit la feuille correspondant au premier schéma de signature non encore utilisé, on signe le message normalement avec cette instance, puis on ajoute dans la signature la clef publique de l'instance, et le haché de tous les frères le long du chemin qui mène de la racine de l'arbre de Merkle jusqu'à la feuille utilisée (utilisation habituelle de l'arbre de Merkle). Cela permet de recalculer la racine de l'arbre à partir de la clef publique de l'instance utilisée, et donc de vérifier qu'elle correspond bien à la i-ième instance de la clef publique. La taille d'une signature est λ^2 pour l'instance de signature à usage unique utilisée, plus $2\lambda^2$ pour la clef publique de l'instance (en fait on peut omettre la moitié de la clef publique correspondant à aux valeurs révélées dans la signature, donc cela coûte seulement λ^2 bits supplémentaires), plus $\lambda \log \ell$ bits pour l'arbre de Merkle.
4. Avec un arbre de Merkle binaire la contribution de l'arbre dans la signature est $\lambda \log \ell$. Pour un arbre k-aire, la hauteur de l'arbre est $\log_k \ell$ (on ne se préoccupe pas des arrondis), et pour chaque nœud il faut révéler $(k-1)\lambda$ bits, donc la taille totale est $(k-1)\lambda \log_k \ell = \lambda \log \ell (k-1)/\log k$. La question est donc de minimiser $(k-1)/\log k$. On vérifie aisément que c'est une fonction croissante, donc le choix optimal est $k = 2$. ($k = 1$ n'est malheureusement pas valide.)
5. Calculer la clef publique nécessite de calculer les valeurs des nœuds de tout l'arbre, donc cela coûte un temps $O(\ell)$ (on ne se préoccupe pas de λ). En particulier signer un nombre exponentiel de messages est impossible, dans la mesure où le calcul de la clef publique deviendrait exponentiel.
6. L'idée est de dériver la clef privée de chaque nœud de manière déterministe à partir d'une graine aléatoire de taille fixe λ. La clef privée d'un nœud est dérivée de cette graine et de sa position dans l'arbre de manière pseudo-aléatoire (mais déterministe). De cette manière, le point crucial est qu'il n'y a jamais besoin de calculer l'arbre entier : pour générer la clef publique par exemple, il suffit de générer celle de la racine, sans avoir besoin de savoir d'avance celle des autres nœuds. La clef publique globale du schéma est la clef publique de l'instance de signature à la racine, donc λ bits seulement. La signature d'un message contient une clef publique et une signature par niveau de l'arbre, donc $O(\lambda^2 \log \ell)$ bits, au lieu de $O(\lambda(\lambda + \log \ell))$ dans le cas précédent.

Exercice 3 (Mineurs égoïstes)**.**

1. Pour chaque $n \in \mathbb{N}$, définissons un état n, qui représente le nombre de blocs d'avance qu'a la chaîne privée des mineurs égoïstes par rapport à la chaîne publique. À l'instant t on est dans l'état $n = 0$. Depuis cet état, on transite vers l'état 1 avec probabilité α, sinon on reste dans l'état 0. Depuis l'état $n > 0$, on transite vers $n + 1$ avec probabilité α et vers $n - 1$ avec probabilité α. On peut représenter cette situation par un automate avec des transitions probabilistes comme suit :



Soit (p_i) la distribution de probabilité au point fixe. On a :

$$p_0 = (1-\alpha)p_0 + (1-\alpha)p_1 \tag{1}$$
$$p_n = \alpha p_{n-1} + (1-\alpha)p_{n+1} \quad \forall n > 0. \tag{2}$$

En résolvant la première équation, on obtient $p_1 = p_0 \cdot \alpha/(1-\alpha)$. Ici, on peut conclure rapidement en devinant que la solution est une suite géométrique de raison $\alpha/(1-\alpha)$, comme le suggère la première équation : on aurait donc

$$p_i = \beta \left(\frac{\alpha}{1-\alpha}\right)^i$$

pour un certain β. Pour s'en assurer, il suffit de vérifier par une récurrence triviale que cette solution vérifie les deux équations précédentes, donc que c'est bien une solution. Enfin, il reste à s'assurer que la somme des p_i soit égale à 1, puisqu'on cherche une distribution de probilité. On obtient

$$\beta \sum \left(\frac{\alpha}{1-\alpha}\right)^i = 1$$
$$\beta = 1 - \frac{\alpha}{1-\alpha}$$
$$= \frac{1-2\alpha}{1-\alpha}.$$

Finalement, la réponse est

$$p_i = \frac{1-2\alpha}{1-\alpha}\left(\frac{\alpha}{1-\alpha}\right)^i.$$

C'est la manière rapide de répondre à la question.

La manière longue mais plus complète, c'est de remarquer que l'équation (2) définit une suite récurrente linéaire d'ordre 2. On peut donc déterminer l'ensemble des solutions de cette équation de manière générique (voir cours de prépa) ; il s'agit d'un sous-espace vectoriel de dimension 2, qui consiste en les combinaisons linéaires de deux suites géométriques dont les raisons sont les racines de l'équation du second degré correspondant à l'équation (2), c'est-à-dire : $x = \alpha + (1-\alpha)x^2$. En résolvant cette équation on trouve une racine $\alpha/(1-\alpha)$ qui correspond à la solution qu'on a devinée précédemment. L'autre racine est simplement égale à 1. Effectivement, si tous les p_i sont égaux, on est dans un état stable, mais leur somme est nulle ou infinie, donc cette solution ne peut pas correspondre à une distribution de probabilités.

2. Si on est dans un état $n > 0$, le prochain bloc incorporé dans la blockchain publique appartient nécessairement aux mineurs égoïstes. Le seul cas où le prochain bloc n'appartient *pas* à ce groupe, c'est si on est dans l'état 0, et le prochain bloc est trouvé hors du groupe (probabilité $1-\alpha$). La probabilité que cela arrive est

$$p_0(1-\alpha) = 1 - 2\alpha.$$

La probabilité que le prochain bloc soit dans le groupe est donc 2α.

3. La question du rendement est plus subtile qu'il n'y paraît. Une manière de définir le rendement du groupe E, c'est comme la proportion de blocs incorporés dans la blockchain publique qui proviennent du groupe. Cette définition a un sens, notamment si on part de 0 bitcoin au début : à terme, le groupe E détient la proportion correspondante de la quantité totale de bitcoins. Suivant cette définition, clairement le groupe E augmente son rendement en étant égoïste : avec un comportement honnête, le groupe E ne produirait qu'une proportion α des blocs, tandis qu'il produit maintenant une proportion 2α.

   Mais il y d'autres manières de définir le rendement. En particulier, à cause du comportement égoïste du groupe E, le nombre total de blocs produits par unité de temps a diminué, ce qu'il faudrait modéliser aussi. Si on définit le rendement comme le nombre de blocs incorporés dans la blockchain produits par le groupe E *par unité de temps*, alors le rendement n'a pas doublé.

   Dans tous les cas, le rendement des mineurs hors du groupe E a diminué, ils perdent sur tous les tableaux. C'est logique : le groupe E forcent les mineurs hors du groupe à perdre du temps à chercher des blocs alors que le groupe E a déjà une réponse. Les mineurs hors du groupe E perdent ainsi une partie de leur temps en recherches inutiles, ce qui n'est pas le cas du groupe E. En principe, tout mineur aurait intérêt à se joindre au groupe E pour profiter de la chaîne privée (en théorie—en réalité bien sûr il y aurait bien d'autres considérations, par exemple si le groupe E faisait connaître son existence, il s'exposerait à des représailles). Cependant, si tout le monde rejoint le groupe égoïste, il n'y a plus de groupe égoïste.