École normale supérieure **2019-2020**
Département d'informatique

Introduction à la cryptologie
TD n° 11 : correction.

Exercice 1 (Ring signature)**.**

1. (a) Le signataire connaît une clef privée sk_s pour un certain s, correspondant à la clef publique pk_s. Pour signer un message m avec les clef publiques $pk_1, \dots, pk_n$, le signataire tire des valeurs x_i pour $i \in [\![1, n]\!] \setminus \{s\}$ uniformément aléatoirement. Il calcule $h = H(m)$ et $y_i = E_{pk_i}(x_i)$ pour $i \neq s$. Il résout ensuite l'équation $\mathsf{Eq}(h, y_1, \dots, y_n)$ en y_s. La résolution est immédiate :
$$y_s = h \oplus \bigoplus_{i \neq s} y_i.$$
En utilisant sa connaissance de sk_s, le signataire calcule $x_s = D_{sk_s}(y_s)$. La signature est $(pk_1, \dots, pk_n, x_1, \dots, x_n)$.
Remarque. On triche ici un peu sur le fait que l'algorithme de signature n'est pas forcément surjectif sur l'ensemble sur l'ensemble des valeurs de y_s possibles. Autrement dit, la valeur y_s que le signataire calcule pourrait ne pas être un chiffré bien formé, auquel cas la fonction de déchiffrement ne fonctionnerait pas. En réalité il faut modifier un peu RSA de base pour avoir la propriété voulue, mais nous n'allons pas nous préoccuper de cela ici.

   (b) Soit m un message quelconque et $h = H(m)$. Soit b le nombre de bits des chiffrés, et soit $n > b$. On choisit n clefs publiques $pk_1, \dots, pk_n$ arbitrairement, et on tire uniformément n paires de messages clairs $(x_0^1, x_1^1), \dots, (x_0^n, x_1^n)$. Pour $1 \leq i \leq n$ et $v \in \{0, 1\}$, on calcule $y_v^i = E_{pk_i}(x_v^i)$. Notre but est de trouver $v_1, \dots, v_n \in \{0, 1\}$ tels que $\mathsf{Eq}(h, y_{v_1}^1, \dots, y_{v_n}^n)$. Le point clef est que cela revient à un système linéaire, à savoir le système suivant :
$$y_0^1 \oplus v_1(y_1^1 \oplus y_0^1) \oplus \cdots \oplus y_n^1 \oplus v_n(y_1^n \oplus y_0^n) = h.$$
Les deux quantités ci-dessus sont égales ssi tous leurs b bits sont égaux. On obtient b équations linéaires sur les $v_1, \dots, v_n$. Comme $n > b$, avec forte probabilité il existe une solution (sinon on peut recommencer en tirant d'autres x_v^i). On obtient ainsi une signature valide $(pk_1, \dots, pk_n, x_{v_1}^1, \dots, x_{v_n}^n)$, pour un message quelconque et pour un ensemble quelconque de signataires.

2. On choisit $n > 9$ clefs publiques quelconques. On choisit un message m arbitraire ; soit $h = H(m)$. On tire arbitrairement $x_{10}, \dots, x_n$. Soit $h' = h - E_{pk_{10}}(x_{10}) - \cdots - E_{pk_n}(x_n) \bmod \mathbb{Z}_{2^b}$. Avec une bonne probabilité sur le choix de ces x_i, $h' < 2^{b-1}$ (on identifie les éléments de Z_{2^b} avec $\{0, \dots, 2^b - 1\}$). Si ce n'est pas le cas, on recommence le tirage jusqu'à avoir cette propriété. Par le théorème non-trivial de l'indication, il existe $x_1, \dots, x_9 \in \mathbb{N}$ tels que $\sum x_i^3 = h'$. Comme chacun des x_i^3 pour $i < 10$ est nécessairement inférieur à $h' < 2^{b-1}$, les x_i^3 sont inférieurs aux modules RSA, donc pour $i < 10$, $E_{pk_i}(x_i) = x_i^3$ est vrai non seulement modulo le module RSA correspondant, mais directement dans les entiers. On déduit que $\mathsf{Eq}(h, y_1, \dots, y_n)$ est vérifiée pour $y_i = E_{pk_i}(x_i) = x_i^3$. La signature $(pk_1, \dots, pk_n, x_1, \dots, x_n)$ est donc valide.

3. (a) Le processus est exactement le même que dans la première question : le signataire en possession de sk_s tire des valeurs x_i pour $i \in [\![1, n]\!] \setminus \{s\}$ uniformément aléatoirement. Il calcule $h = H(m)$ et $y_i = E_{pk_i}(x_i)$ pour $i \neq s$. La différence par rapport à la première question est de savoir comment résoudre $\mathsf{Eq}(h, y_1, \dots, y_n)$ en y_s. Mais cette équation se résout en fait simplement (elle est conçue pour) du fait que S_h est inversible :
$$\begin{aligned} & S_h(y_1 \oplus S_h(y_2 \oplus \dots S_h(y_n))) = 0 \\ \Leftrightarrow \quad & y_s \oplus S_h(y_{s+1} \oplus \dots S_h(y_n)) = S_h^{-1}(\dots S_h^{-1}(S_h^{-1}(S_h^{-1}(0) \oplus y_1) \oplus y_2) \cdots \oplus y_{s-1}). \end{aligned}$$

On obtient

$$y_s = S_h(y_{s+1} \oplus \dots S_h(y_n)) \oplus S_h^{-1}(\dots S_h^{-1}(S_h^{-1}(S_h^{-1}(0) \oplus y_1) \oplus y_2) \cdots \oplus y_{s-1}).$$

Comme dans la première question, il reste à calculer $x_s = D_{sk_s}(y_s)$. La signature est $(pk_1, \dots, pk_n, x_1, \dots, x_n)$.

(b) La résolution de l'équation donnée dans la question précédente montre que la solution est unique.

(c) Fixons $h = H(m)$ et $s \in [\![1, n]\!]$. Soit $\mathcal{E}$ l'ensemble des $y_1, \dots, y_n$ qui sont solutions de $\mathsf{Eq}(h, y_1, \dots, y_n)$. Soit $\mathcal{S}$ l'ensemble des choix possibles de $(y_1, \dots, y_{s-1}, y_{s+1}, \dots, y_n)$ (*i.e.* $\mathcal{S} = (\{0,1\}^b)^{n-1}$ si les y_i vivent dans $\{0,1\}^b$). Par la question précédente, pour chaque choix de $(y_1, \dots, y_{s-1}, y_{s+1}, \dots, y_n)$ il existe un unique y_s tel que $(y_1, \dots, y_n) \in \mathcal{E}$. Soit $f_s : \mathcal{S} \to \mathcal{E}$ la bijection correspondante. La question de l'énoncé revient à montrer que l'image de la distribution uniforme sur $\mathcal{S}$ par f_s est indépendante de s. Mais puisque f_s est une bijection, cette distribution est en fait égale à la distribution uniforme sur $\mathcal{E}$, qui est clairement indépendante de s.

(d) Pour signer un message m avec $h = H(m)$, le signataire qui possède la clef secrète correspondant à la clef publique pk_s tire $(x_1, \dots, x_{s-1}, x_{s+1}, \dots, x_n)$ uniformément aléatoirement, et pose $y_i = E_{pk_i}(x_i)$. Le signataire calcule ensuite y_s tel que $\mathsf{Eq}(h, y_1, \dots, y_n)$ est satisfaite. Puis il calcule $x_s = D_{sk_s}(y_s)$ en utilisant sa clef secrète sk_s. La signature est $(pk_1, \dots, pk_n, x_1, \dots, x_n)$.

Pour montrer que la signature ne révèle rien sur qui a signé parmi le cercle $pk_1, \dots, pk_n$, il suffit de montrer que la distribution de la signature ne dépend pas de s. C'est bien le cas : en effet, comme on utilise RSA de base, E et D sont bijectifs, donc la distribution de $(y_1, \dots, y_{s-1}, y_{s+1}, \dots, y_n)$ est uniforme. Par la question précédente, la distribution de $(y_1, \dots, y_n)$ qui en résulte ne dépend pas de s (elle est même uniforme parmi les solutions de Eq). Puisque E et D sont des bijections, la distribution de $(x_1, \dots, x_n)$ ne dépend donc pas non plus de s (elle est même uniforme parmi les signatures valides, pour h et $pk_1, \dots, pk_n$ fixés).

Remarque. Comme plus haut, on triche un peu sur un point : les images de E_{pk_i} ne vivent pas toutes dans le même espace $\{0,1\}^b$ puisque les modules RSA sont différents ; en réalité, il faut modifier légèrement RSA pour que ce soit le cas. L'énoncé nous dit de faire semblant que les y_i sont dans le même espace justement pour ne pas avoir à nous préoccuper de ça.

(e) On voit que la taille de la signature croît linéairement avec le nombre de signataires. Si les pk_i sont inclus dans la signature, c'est inévitable. S'ils sont connus par ailleurs, il existe des solutions sous-linéaires.

Exercice 2 (More Ring Signatures.)**.** Sketch below.

1. Let C be the circuit that takes as private input a key $\vec{x}_i$ and outputs 1 iff $\mathsf{H}(\vec{x}_s) = \vec{y}_s$ for some s. We can then proceed as before, except the MPC protocol evaluates the circuit C instead of $mathsfH$. The size of C (and hence the size of a signature) grows linearly in n.
2. We can compute a Merkle tree with leafs $\vec{y}_i$ and root h. The circuit C then takes as private input s, a co-path P_s (to verify that $\vec{y}_s$ is committed in h) and $\vec{x}_s$. C computes $\vec{y}_s = \mathsf{H}(\vec{x}_s)$ and checks if P_s is a valid co-path for $\vec{y}_s$ of h. The size now grows logarithmically in n.
3. Roughly, the signer simulates all proofs but the s-th proof with some random challenges c_i and obtains valid transcripts (r_i, c_i, z_i) for $i \neq s$, then computes $r_s \leftarrow g^k$. It sets $c = \mathsf{H}(r_1, \dots, r_n, m)$ for message m and $c_s = c \oplus (\oplus_{i \neq s} c_i)$. Then, it finishes the transcript for the s-th proof via $z_s = k_s - c_s x_s$. A signature consists of $(r_i, c_i, z_i)_{i \in [n]}$, and a verifier checks if all transcripts are valid and $(c_i)_i$ is a secret sharing of $\mathsf{H}(r_1, \dots, r_n, m)$.

Exercice 3 (Authentification avec LPN)**.** Sketch below.

1. Under LPN, all sent messages look random, so a passive adversary learns nothing from the transcripts.
2. The adversary sends $A = \begin{bmatrix} I_n \\ 0 \end{bmatrix}$ to the card, and obtains $\begin{bmatrix} s \\ 0 \end{bmatrix} + e$. It can then repeat this attack to recompute s (based on the positions that are consistent over multiple runs).
3. Set $A \leftarrow \mathsf{H}(r)$ for random r chosen by the receptor. Again, the messages look random under LPN.
4. Again, as $A_1 s_1 + e$ looks random under LPN, the transcript looks random.
5. Proceed as in previous attack, but choose $A_b = 0$ to recover s_{1-b}.
6. Let the card and the receptor perform the authentification protocol from TLS for example.