Université Paris 8 Année 2021–2022

M1 Mathématiques et applications, parcours ACC et CSSD

Cryptographie à clef publique – Exercices complémentaires

08/04/2022

Le corrigé de certains exercices sera disponible à l'adresse suivante :

`www.math.univ-paris13.fr/~lavauzelle/teaching/2021-22/clef-publique.html`

($\star$) exercice fondamental ($\star\star$) pour s'entraîner ($\star\star\star$) pour aller plus loin 💻 sur machine

Exercice 1. *Time-lock puzzle.*

Un *time-lock puzzle* (« défi verrouillé dans le temps ») est une primitive cryptographique dans laquelle un message chiffré ne peut pas être déchiffré avant un certain temps. Une manière de réaliser un *time-lock puzzle* est de faire en sorte que le déchiffrement demande une quantité de calcul importante. Dans cet exercice, on s'intéresse à un exemple basé sur le problème de la factorisation.

Question 1.– Dans un groupe $\mathbb{G}$ noté multiplicativement, quelle est la complexité de l'opération $x \mapsto a^x$ pour $a \in \mathbb{G}$? On exprimera cette complexité en nombre de multiplications dans $\mathbb{G}$, en fonction de x.

Alice souhaite chiffrer un message m afin qu'il ne soit pas déchiffrable avant d'avoir effectué environ 2^t opérations, pour un certain paramètre $t \geq 1$.

• **Génération de clefs.** Alice choisit p et q deux grands nombres premiers tels que $n = pq$ ne soit pas factorisable. Alice publie (n, t), où t est associé au seuil de calcul (voir ci-dessus).

• **Chiffrement.** On suppose que le message m que Alice souhaite chiffrer est dans $\mathbb{Z}/n\mathbb{Z}$. Alice choisit aléatoirement $g \in (\mathbb{Z}/n\mathbb{Z})^\times$. Puis, Alice calcule successivement $z = g^{2^{(2^t)}} \mod n$ puis $c = m + z$. Le chiffré est (g, c).

Question 2.– Donner un algorithme qui permet de déchiffrer (g, c) avec $O(2^t)$ opérations dans $\mathbb{Z}/n\mathbb{Z}$.

Question 3.– Donner un algorithme qui permet à Alice de chiffrer son message en $O(t + \log n)$ opérations. Pourquoi ne peut-on pas appliquer la même idée pour déchiffrer ?

Question 4.– Afin d'implanter un *time-lock puzzle*, Oscar décide de remplacer le groupe $(\mathbb{Z}/n\mathbb{Z})^\times$ par le groupe multiplicatif $\mathbb{F}_p^\times$ d'un corps fini. Pourquoi son choix n'est-il pas satisfaisant ?

Exercice 2. Signature de Rabin.

Pour un entier $n = pq$, on note $\mathcal{Q}_n = \mathrm{QR}_n \cup \mathrm{QR}_n^*$, où

$$\mathrm{QR}_n := \left\{ x \in \mathbb{Z}/n\mathbb{Z} \mid \left(\frac{x}{p}\right) = 1 \text{ et } \left(\frac{x}{q}\right) = 1 \right\},$$
$$\mathrm{QR}_n^* := \left\{ y \in \mathbb{Z}/n\mathbb{Z} \mid \left(\frac{y}{p}\right) = -1 \text{ et } \left(\frac{y}{q}\right) = -1 \right\}.$$

Question 1.– Démontrer que, si $n = pq$ avec $p \equiv q \equiv 3 \mod 4$, alors pour tout $x \in \mathcal{Q}_n$, ou bien x, ou bien $n - x$, est un carré modulo n.

La signature de Rabin se présente ainsi :

- **Génération de clefs.** Alice engendre deux grands nombres premiers p et q, tels que $p \equiv q \equiv 3 \mod 4$. La clé publique est $n = pq$, la clé privée est (p, q).
- **Signature.** Soit $H : \{0,1\}^* \to \mathcal{Q}_n$ une fonction de hachage. Alice hache son message $\boldsymbol{m}$ en $h = H(\boldsymbol{m}) \in \mathcal{Q}_n$.
  - Si $h \in \mathrm{QR}_n$, Alice retourne une signature s telle que $s^2 = h \mod n$.
  - Sinon, Alice retourne une signature s telle que $s^2 = -h \mod n$.
- **Vérification.** Bob calcule $h = H(\boldsymbol{m})$, et vérifie que $h = s^2 \mod n$ ou $h = -s^2 \mod n$.

Question 2.– Si $x \in \mathrm{QR}_n$, combien x admet-il de racines carrées modulo n ? Justifier.

Question 3.– Supposons par exemple que $h \in \mathrm{QR}_n$. Décrire un algorithme qui permet à Alice de calculer efficacement $s = \sqrt{h} \mod n$.

Question 4.– Supposons que $n = pq$. Supposons que l'on détienne un algorithme A qui calcule une racine carrée de $x \in \mathrm{QR}_n$. Donner un algorithme probabiliste qui factorise n.

Exercice 3. Points fixes de RSA.

Soit $(n = pq, e)$ une clé publique RSA, et d la clé privée associée.

Question 1.– Supposons qu'un attaquant découvre (par exemple en cherchant au hasard) un entier $a \in \{1, \ldots, n-1\}$ qui n'est pas premier avec n. Démontrer qu'il peut alors casser le système.

Un point fixe de la fonction RSA est un élément $m \in \mathbb{Z}/n\mathbb{Z}$ tel que $m^e \equiv m \mod n$. Trois points fixes triviaux de la fonction RSA sont : 0, 1 et $n - 1$. Pour le dernier, notons que e est nécessairement impair, donc $(n-1)^e \equiv (-1)^e \equiv -1 \mod n$.

Question 2.– En utilisant le théorème des restes chinois, donner 6 autres points fixes de la fonction RSA.

Question 3.– En quoi la connaissance de certains de ces points fixes permettrait de factoriser n ?

Question 4.– Ces résultats remettent-ils en question la sécurité de RSA ?

Exercice 4. Problème de Diffie–Hellman cubique.

Soit $\mathbb{G}$ un groupe fini noté multiplicativement. On suppose que l'ordre r du groupe $\mathbb{G}$ est connu et impair.

Question 1.– Étant donné $y = x^2 \in \mathbb{G}$, comment peut-on calculer efficacement x ?

Question 2.– Soient a, b deux entiers naturels distincts. Vérifier que

$$2ab = \frac{a^3 - b^3}{a - b} - \frac{a^3 + b^3}{a + b}\,.$$

Question 3.– Supposons qu'il existe un algorithme CUBE qui, étant donné (g, g^a), calcule g^{a^3} de manière déterministe et en temps polynomial. En utilisant les questions précédentes, démontrer qu'on peut alors résoudre le problème CDH (Diffie–Hellman calculatoire).