Université Paris 8 Année 2021–2022

M1 Mathématiques et applications, parcours ACC et CSSD

Cryptographie à clef publique – Feuille de TD 5

25/02/2022

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. $(\star)$ Signature RSA : falsification sélective à message choisi.

On s'intéresse au schéma de signature RSA brut. Dans le cours, nous en avons vu une falsification existentielle à clef seule. Le but de cet exercice est de monter une falsification sélective à message choisi. Une falsification **sélective** signifie que l'attaquant fixe le message dont il veut falsifier la signature **avant** de monter son attaque (et donc, avant de demander au signataire d'autres signatures valides). C'est donc une attaque moins forte que la falsification universelle, mais plus forte que la falsification existentielle.

Dans l'exercice, on note $\mathtt{pk} = (n, e)$ et $\mathtt{sk} = d$ les clefs publique et privée du schéma de signature RSA brut.

Question 1.– Soient $m_1, m_2 \in (\mathbb{Z}/n\mathbb{Z})^\times$ deux messages, et s_1, s_2 leurs signatures correspondantes. Que vaut la signature s du message $m = m_1 m_2 \mod n$, en fonction de s_1 et s_2 ?

Question 2.– En déduire la falsification de la signature d'un message $m \in (\mathbb{Z}/n\mathbb{Z})^\times$ quelconque, après avoir demandé à Alice la signature de deux messages m_1 et m_2 (différents de m) judicieusement choisis.

Exercice 2. $(\star\star)$ Vérification simultanée de signatures RSA.

Soit $(n = pq, e)$ une clé publique RSA, et d la clé privée associée. On s'intéresse au schéma de signature RSA « brut ».

On suppose que n est de taille t bits.

Question 1.– En fonction de t, quel est le coût algorithmique (en nombre de multiplications et carrés dans $\mathbb{Z}/n\mathbb{Z}$) d'une signature RSA ?

Bob reçoit une série de $\ell \geq 2$ messages signés par Alice : $(m_1, s_1), \ldots, (m_\ell, s_\ell)$.

Pour vérifier ces signatures RSA plus rapidement, Bob décide de multiplier tous les messages entre eux : il calcule ainsi

$$m = m_1 m_2 \cdots m_\ell \mod n \quad \text{et} \quad s = s_1 s_2 \ldots s_\ell \mod n\,.$$

Puis, il décide d'accepter la série de messages signés par Alice si et seulement si $s^e = m \mod n$.

Question 2.– Démontrer que si tous les messages ont bien été signés par Alice, alors Bob a raison d'accepter la série de signatures d'Alice.

Question 3.– Quantifier le gain de calcul de Bob en utilisant cette méthode.

Question 4.– Charlie sait que Bob utilise cette méthode pour vérifier les signatures d'Alice. Charlie intercepte une série $(m_1, s_1), \ldots, (m_\ell, s_\ell)$ de messages signés par Alice (Charlie n'a donc pas choisi les messages). Comment peut-il intégrer un autre message m' à la série pour faire croire à Bob qu'Alice a également signé m' ?

Exercice 3. $(\star)$ Signature ElGamal : réutilisation de l'aléa.

On s'intéresse au schéma de signature ElGamal, dans lequel le message est haché avant d'être signé. Plus précisément, si H est une fonction de hachage à valeurs dans le groupe $\mathbb{F}_p^\times$, voici l'algorithme de signature :

Algorithme 1 : Algorithme de signature d'ElGamal avec fonction de hachage

Entrée : un message $\boldsymbol{m} \in \{0,1\}^*$, la clé privée $a \in \{1, \ldots, p-2\}$
Sortie : une signature $s \in \mathbb{F}_p^\times \times \{0, \ldots, p-2\}$

1 Choisir $k \in \mathbb{Z}/(p-1)\mathbb{Z}$ inversible.
2 Calculer $b = g^k \mod p$.
3 Calculer $h = H(\boldsymbol{m})$.
4 Calculer $c = (h - ab)k^{-1} \mod (p-1)$.
5 Retourner $s = (b, c)$.

On suppose qu'**Alice réutilise le même aléa k pour toutes ses signatures**.

Question 1.– Soient $s = (b, c)$ et $s' = (b', c')$ les signatures de deux messages distincts m et m' (avec le même k). Comparer b et b', puis déterminer une égalité liant $h = H(\boldsymbol{m})$, $h' = H(\boldsymbol{m'})$, a, b, c et c'.

Question 2.– En déduire une attaque à message connu sur la clé privée d'Alice, qui réussit avec très bonne probabilité.

Exercice 4. 💻 $(\star\star)$ Calcul rapide de $g^b b^c$ dans la vérification d'ElGamal.

Dans l'algorithme de vérification de la signature d'ElGamal, on a besoin de calculer la valeur $\alpha^b b^c$, où α est un élément de $\mathbb{F}_p^\times$, et b, c sont des éléments entre 1 et $p-2$ que l'on peut considérer comme aléatoires.

L'algorithme *square-and-multiply* permet de calculer une exponentiation dans un groupe cyclique d'ordre ℓ, en $\log_2(\ell)/2$ multiplications et $\log_2(\ell)$ carrés en moyenne (l'exposant est supposé aléatoire). Si b et c sont aléatoires, le calcul de $g^b b^c$ requiert donc, en moyenne, approximativement $\log_2(p)$ multiplications et $2\log_2(p)$ carrés.

Le but de cet exercice est de calculer $\alpha^b b^c$ sensiblement plus rapidement. Pour cela, on propose l'Algorithme 2.

Question 1.– Démontrer que pour tout $i = \ell - 1, \ldots, 0$, à la fin de la boucle **Pour (...)** de l'Algorithme 2, on a

$$x = \alpha^{\sum_{j=i}^{\ell-1} b_j 2^{j-i}} b^{\sum_{j=i}^{\ell-1} c_j 2^{j-i}}.$$

Algorithme 2 : Algorithme de calcul rapide de $a^b b^c$.

Entrée : $a \in \mathbb{F}_p^\times$, $b = \sum_{i=0}^{\ell-1} b_i 2^i$ et $c = \sum_{i=0}^{\ell-1} c_i 2^i \in \{1, \ldots, p-2\}$
Sortie : $a^b b^c$

```
Calculer z ← ab.
Initialiser x ← 1.
Pour i allant de ℓ − 1 à 0 faire
    Calculer x ← x^2
    Si (b_i, c_i) = (1, 1)
        Calculer x ← xz
    Si (b_i, c_i) = (1, 0)
        Calculer x ← xa
    Si (b_i, c_i) = (0, 1)
        Calculer x ← xb
Retourner x.
```

En déduire que l'algorithme est correct.

Question 2.– Compter le nombre moyen de carrés et le nombre moyen de multiplications effectués par l'Algorithme 2, lorsque b et c sont des entiers de ℓ bits tirés aléatoirement.

Question 3.– Implanter l'Algorithme 2 et vérifier l'amélioration pratique qu'il procure, comparé aux calculs successifs de a^b et b^c par la méthode *square-and-multiply*.