Université Paris 8 J. Lavauzelle

Master 1 Mathématiques et applications — année 2025–26

Cryptographie à clé publique – Feuille de TD 4

18/02/2026

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

`https://lvzl.fr/teaching/2025-26/cp.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 sous sa forme "brute", c'est-à-dire sans l'utilisation de fonction de hachage. 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 $\texttt{pk} = (n, e)$ et $\texttt{sk} = d$ les clefs publique et privée du schéma de signature RSA brut.

Question 1.– Soient $m_1, m_2 \in \mathbb{Z}/n\mathbb{Z}$ 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.– Soit $m \in \mathbb{Z}/n\mathbb{Z}$ quelconque. Déduire de la question précédente une falsification de la signature de m, 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.

Dans cete exercice, on s'intéresse au schéma de signature RSA « brut ». Soit $(n = pq, e)$ une clé publique RSA, et d la clé privée associée. 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. $(\star\star)$ Signature de Lamport.

Dans cet exercice, on s'intéresse au schéma de signature de Lamport, qui ne présuppose que l'utilisation d'une fonction à sens unique. Pour cela, prenons E et E' deux ensembles finis et $f : E \to E'$ une fonction à sens unique.

Le schéma de signature est défini pour un certain paramètre entier $k \geq 1$, via les Algorithmes 1 (génération de clefs), 2 (signature) et 3 (vérification).

Algorithme 1 : Génération des clefs

Entrée : une taille $k \geq 1$
Sortie : une paire de clés publique/privée

1 Tirer uniformément $2k$ éléments distincts de E, et les stocker dans une matrice de taille $(2 \times k)$:

$$\boldsymbol{A} = \begin{pmatrix} a_{0,1} & a_{0,1} & \ldots & \ldots & a_{0,k} \\ a_{1,1} & a_{1,2} & \ldots & \ldots & a_{1,k} \end{pmatrix} \in E^{2\times k}$$

2 Calculer la matrice $\boldsymbol{B}$ de taille $(2 \times k)$ sur E', constituée des $b_{i,j} = f(a_{i,j})$:

$$\boldsymbol{B} = \begin{pmatrix} f(a_{0,1}) & f(a_{0,2}) & \ldots & \ldots & f(a_{0,k}) \\ f(a_{1,1}) & f(a_{1,2}) & \ldots & \ldots & f(a_{1,k}) \end{pmatrix} \in (E')^{2\times k}$$

3 **Retourner** la clé publique est $\boldsymbol{B}$ et la clé privée est $\boldsymbol{A}$.

Algorithme 2 : Signature

Entrée : la clé privée $\boldsymbol{A}$, le message à signer $\boldsymbol{m} \in \{0,1\}^k$
Sortie : la signature $\boldsymbol{s}$

1 Pour tout $i \in \{1, \ldots, k\}$, définir $s_i \coloneqq a_{m_i,i}$.
2 **Retourner** la signature $\boldsymbol{s} = (s_1, \ldots, s_k) \in E^k$

Algorithme 3 : Vérification

Entrée : la clé publique $\boldsymbol{B}$, la signature $\boldsymbol{s}$, le message à signer $\boldsymbol{m} \in \{0,1\}^k$
Sortie : un booléen `True/False`

1 **Retourner** `True` si $f(s_i) = B_{m_i,i}$ pour tout $i \in \{1, \ldots, k\}$, et `False` sinon.

Question 1.– Expliquer pourquoi, si la fonction f n'est pas à sens unique, alors la signature n'est pas sûre.

Question 2.– Expliquer pourquoi la même paire de clés ne peut pas être utilisée pour signer deux messages.

Question 3.– A priori, le schéma semble construit de sorte que la longueur des messages et le nombre de colonnes de la clé publique (et de la clé privée) sont égaux.

1. En quoi cela pose-t-il problème d'un point de vue pratique ?
2. Proposer une modification de la signature afin qu'elle puisse rester pratique pour n'importe quel message à signer.
3. En déduire (approximativement) la taille des clés de cette signature. Justifier.

Exercice 4. $(\star\star\star)$ **Signature de cercle.**

Une signature de cercle (*ring signature*) est une primitive cryptographique qui permet à chaque membre d'un « cercle » (= groupe d'utilisateur) de signer anonymement un message m au nom du cercle.

Voici quelques propriétés désirées dans une signature de cercle :

1. N'importe quel vérificateur extérieur doit pouvoir vérifier (et être convaincu par) une signature s d'un message m émise au nom du cercle.
2. Il doit être **impossible de déterminer** quel membre du cercle a émis la signature.
3. Enfin, le signataire ne doit pas avoir besoin de l'aide d'autres membres du groupe pour poubvoir signer un message.

Dans cet exercice, on présente une signature de cercle basée sur la signature RSA-FDH. La fonction de hachage utilisée est notée H, et est à valeur dans $\{0,1\}^t$ pour $t \geq 1$. On suppose également que toutes les clefs publiques des membres du cercle sont authentifiées.

Pour commencer l'exercice, on considère un cercle composé de deux membres seulement : Alice et Bob. Leurs clefs publiques sont respectivement (n_A, e_A) et (n_B, e_B). On suppose que les modules RSA n_A et n_B sont de taille $t+1$ bits, et on assimile tout élément de $\{0,1\}^t$ avec un entier inférieur à 2^t. La signature d'un message $\boldsymbol{m}$ par Alice est donc $H(\boldsymbol{m})^{d_A} \mod n_A$ où d_A est la clé privée associée à (n_A, e_A).

On note enfin $\oplus$ l'opération de `xor` (addition modulo 2) bit-à-bit, qui s'applique notamment sur des entiers vus comme des chaînes de bits de longueur t.

L'Algorithme 4 décrit les opérations à effectuer par Alice pour émettre une signature de cercle. Notons qu'une description similaire est possible pour Bob, en remplaçant simplement les données propres à Alice par celles propres à Bob, et réciproquement. L'Algorithme 5 décrit la vérification de la signature effectuée par une personne potentiellement extérieure au cercle.

Algorithme 4 : Algorithme de signature de cercle (opéré par Alice)

Entrée : un message $\boldsymbol{m}$, les clés publiques $(n_A, e_A), (n_B, e_B)$, la clé privée d_A d'Alice
Sortie : une signature de cercle s

1 Hacher le message $\boldsymbol{m}$ en $h = H(\boldsymbol{m})$.
2 Tirer aléatoirement $s_B \leftarrow (\mathbb{Z}/n_B\mathbb{Z})^\times$.
3 Calculer $z_B = s_B^{e_B} \mod n_B$.
4 Calculer $s_A = (z_B \oplus h)^{d_A} \mod n_A$.
5 Retourner $s = (s_A, s_B)$.

Algorithme 5 : Algorithme de vérification de signature de cercle

Entrée : un message $\boldsymbol{m}$, les clés publiques $(n_A, e_A), (n_B, e_B)$, une signature $s = (s_A, s_B)$
Sortie : accepte/refuse

1 Calculer $x = (s_A^{e_A} \mod n_A) \oplus (s_B^{e_B} \mod n_B)$.
2 Vérifier que $x = H(\boldsymbol{m})$.

Question 1.– Vérifier que l'Algorithme 5 est valide, c'est-à-dire qu'il accepte toute signature effectuée honnêtement par Alice ou par Bob.

Question 2.– Expliquer en quoi la signature est anonyme pour un signataire quelconque du cercle. C'est-à-dire, argumenter sur le fait que le vérificateur ne peut pas savoir qui, parmi Alice et Bob, a produit la signature $s = (s_A, s_B)$.

On dit qu'une fonction de hachage est résistante au calcul de préimage si, étant donné un haché h, il est impossible calculatoirement de trouver un message $\boldsymbol{m}$ tel que $H(\boldsymbol{m}) = h$.

Question 3.– Supposons que la fonction de hachage H ne soit pas résistante au calcul de préimage. Donner une attaque sur le schéma de signature de cercle. On précisera le type d'attaque et les moyens donnés à l'attaquant.

Question 4.– Dans un contexte où l'on souhaite avoir une sécurité à long terme, quelle serait la taille de la signature de cercle ?

Une méthode pour falsifier une signature de $\boldsymbol{m}$ consiste à créer deux tableaux de valeurs de la forme $v_A := u_A^{e_A} \mod n_A$ et $v_B := u_B^{e_B} \mod n_B$ (où les u_A et u_B sont tirées aléatoirement), et de chercher une « collision » entre les tableaux de la forme $v_A \oplus H(\boldsymbol{m}) = v_B$. Une fois cette collision trouvée, on émet la signature (u_A, u_B) associé au message $\boldsymbol{m}$.

Question 5.– Supposons ici que $t = 256$.

1. Préciser si l'attaque décrite ci-dessus induit une falsification existentielle ou universelle. Justifier clairement.
2. Quelle est la taille approximative des tableaux nécessaires pour avoir une falsification avec probabilité proche de 0.5 ?

Question 6.– Généraliser le schéma de signature de cercle à K utilisateurs au lieu de 2 (Alice et Bob). On présentera l'algorithme de signature de l'un de ces K utilisateurs, ainsi que l'algorithme de vérification à effectuer.