Université Paris 8 J. Lavauzelle

Master 1 Mathématiques et applications — année 2025–26

Cryptographie à clé publique – Solutions feuille de TD 4

18/02/2026

Retrouvez le sujet du TD et d'autres exercices à l'adresse :

`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 $\mathtt{pk} = (n, e)$ et $\mathtt{sk} = d$ les clefs publique et privée du schéma de signature RSA brut.

Question 1.– Soient $m_1, m_2 \in \mathbb{Z}/n\mathbb{Z}$ 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.

Solutions de l'Exercice 1.

Solution Q1. On a

$$\begin{cases} s_1 &= (m_1)^d \mod n \\ s_2 &= (m_2)^d \mod n \\ s &= (m_1 m_2)^d \mod n \end{cases}$$

Par conséquent,

$$s = s_1 s_2 \mod n$$

Solution Q2. Supposons que l'on veuille falsifier une signature pour un message cible $m \in \mathbb{Z}/n\mathbb{Z}$. L'idée de l'attaque est la suivante.

1. On choisit aléatoirement un message $m_1 \in (\mathbb{Z}/n\mathbb{Z})^\times$.
2. On calcule $m_2 = m \cdot m_1^{-1} \mod n$.
3. On demande à Alice de signer m_1 et m_2 (notons s_1 et s_2 les signatures associées).
4. On calcule $s = s_1 s_2 \mod n$

On obtient alors une signature valide s du message m.

Remarque. L'attaque est donc une falsification sélective à message choisi.

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' ?

Solutions de l'Exercice 2.

Solution Q1. Si n est de taille t bits, alors on doit effectuer $O(t)$ opérations arithmétiques (dans $\mathbb{Z}/n\mathbb{Z}$) pour calculer $m^d \mod n$, qui est la signature du message m.

Solution Q2. Si tous les messages ont été signés par Alice, alors on a :

$$s_i = m_i^d \mod n, \quad \forall i \in \{1, \ldots, \ell\}\,.$$

Puis,

$$s^e = (s_1 \ldots s_\ell)^e = (m_1^d \ldots m_\ell^d)^e = m^{de} = m \mod n$$

Solution Q3. Avec la méthode usuelle, Bob aurait effectué ℓ élévations à la puissance e, soit $O(\ell t)$ opérations arithmétiques. Avec la vérification simultanée, il n'effectue plus qu'une élévation à la puissance e, et $\ell - 1$ multiplications, donc $O(\ell + t)$ opérations arithmétiques.

Solution Q4. L'idée est la suivante : Charlie peut modifier les messages et les signatures tant que les produits $m = m_1 \ldots m_\ell \mod n$ et $s = s_1 \ldots s_\ell$ restent constants. Si Charlie souhaite intégrer un message m' à la série de messages et prétendre qu'il a été signé, il peut donc simplement :

- transformer m_ℓ en $m'_\ell = m_\ell / m'$ et ajouter son message $m'_{\ell+1} = m'$ à la série de message (ainsi on a $m_1 \ldots m_{\ell-1} m'_\ell m'_{\ell+1} = m \mod n$),
- ajouter une valeur de signature $s'_{\ell+1} = r$ quelconque à la série de signature, et définir une nouvelle signature $s'_\ell = s_\ell / r$, de sorte que le produits les éléments de la nouvelle série de signatures soit encore $s_1 \ldots s_{\ell-1} s'_\ell s'_{\ell+1} = s \mod n$.

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 $(2 \times k)$:

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

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

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

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

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 $i \in \{1, \dots, k\}$, 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.

Solutions de l'Exercice 3.

Solution Q1. Si f n'est pas à sens unique, cela signifie qu'il est possible de retrouver certaines préimages $a_{j,}$ à partir de $b_{j,i} = f(a_{j,i})$ uniquement. Un attaquant, qui a accès à la clé publique $\boldsymbol{B}$, pourra donc reconstituer une partie de la clé secrète $\boldsymbol{A}$. Sur les indices i correspondant à cette partie de la clé secrète, l'attaquant pourra donc forger les éléments s_i de la signature.

Solution Q2. Supposons que deux messages $\boldsymbol{m}, \boldsymbol{m}' \in \{0,1\}^k$ soit signés avec une même paire de clefs $(\boldsymbol{A}, \boldsymbol{B})$, pour obtenir deux signatures $\boldsymbol{s}, \boldsymbol{s}'$. Alors, les bits communs $m_i = m'_i$ correspondent à deux éléments de signatures égaux : $s_i = s'_i$. Autrement dit, avec la connaissance de $\boldsymbol{m}$, un attaquant peut forger la partie de la signature de $\boldsymbol{m}'$ qui correspond aux bits identiques de $\boldsymbol{m}, \boldsymbol{m}'$: ce n'est clairement pas un comportement voulu.

Solution Q3.

1. Cela signifie que les clés sont au moins aussi longues que le message (en nombre de bits) : ce n'est pas un comportement voulu pour une signature.
2. On peut hacher le message avant de procéder à la signature.
3. D'un point de vue calculatoire, si l'on souhaite une sécurité de 128 bits, il faut que la taille des ensembles de départ et d'arrivée d'une fonction à sens unique soit $\geq 2^{256}$ (pour éviter les attaques de type paradoxe des anniversaires). Chaque élément de E et E' doit donc être codé sur 256 bits. Les fonctions de hachages cryptographiquement sures donnent des hachés de plusieurs centaines de bits (voir les fonctions SHA par exemple). On obtient donc des clés de taille $\simeq 200 \times 2 \times 256 \times \simeq 1000$ bits. C'est raisonnable.

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.

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 $H(\boldsymbol{m}) = h$.

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 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.

Solutions de l'Exercice 4.

Solution Q1. Notons $h = H(\boldsymbol{m})$ et calculons la valeur de x dans l'étape 1 de l'Algorithme de vérification. D'une part, on a :

$$s_A^{e_A} \equiv (z_B \oplus h)^{d_A e_A} \equiv z_B \oplus h \mod n_A$$

D'autre part, $s_B^{e_B} \mod n_B$ est défini comme égal à l'entier z_B. Par conséquent :

$$x = (s_A^{e_A} \mod n_A) \oplus (s_B^{e_B} \mod n_B) = z_B \oplus h \oplus z_B = h\,.$$

Solution Q2. Dans le cas où Alice a produit la signature, on a :

$$s = ((s_B^{e_B} \oplus h)^{d_A} \mod n_A\ ,\ s_B), \quad \text{avec } s_B \text{ aléatoire.}$$

Dans le cas où c'est Bob, on obtient :

$$s' = (s_B\ ,\ (s_A^{e_A} \oplus h)^{d_B} \mod n_B), \quad \text{avec } s_A \text{ aléatoire.}$$

Si le signataire savait différencier qui d'Alice ou Bob a produit la signature, alors il saurait différencier un élément de la forme $(s_B^{e_B} \oplus h)^{d_A} \mod n_A$ (où h est fixé mais s_B est aléatoire dans $(\mathbb{Z}/n_B\mathbb{Z})^\times$), d'un élément aléatoire $s_A \in (\mathbb{Z}/n_A\mathbb{Z})^\times$. C'est impossible car $(s_B^{e_B} \oplus h)^{d_A} \mod n_A$ est aléatoire dans $(\mathbb{Z}/n_A\mathbb{Z})^\times$ (l'application $y \mapsto (y^{e_B} \oplus h)^{d_A} \mod n_A$ étant une bijection).

Solution Q3. On procède très simplement comme suit :

1. tirer aléatoirement s_A dans $(\mathbb{Z}/n_A\mathbb{Z})^\times$ et s_B dans $(\mathbb{Z}/n_B\mathbb{Z})^\times$,
2. calculer $x = (s_A^{e_A} \mod n_A) \oplus (s_B^{e_B} \mod n_B)$,
3. calculer $\boldsymbol{m}$ tel que $H(\boldsymbol{m}) = x$ (car H n'est pas résistante aux préimages),
4. émettre la signature $s = (s_A, s_B)$ du message $\boldsymbol{m}$.

C'est une attaque par falsification existentielle à clé seule.

Solution Q4. Pour une sécurité à long terme on doit prendre n_A et n_B de taille ≥ 3072 bits, soit une signature de cercle de taille 6144 au moins.

Solution Q5.

1. C'est une falsification universelle : on peut fixer $\boldsymbol{m}$ avant de monter l'attaque.
2. D'après le paradoxe des anniversaires, il faut que la taille T des tableaux vérifie

$$T \geq 2^{t/2} = 2^{128}.$$

Solution Q6. Notons $(n_1, e_1), \ldots (n_K, e_K)$ les clefs publiques des K membres du cercle, et supposons pour simplifier que le signataire soit le premier membre du cercle. Alors, pour signer un message $\boldsymbol{m}$ on définit l'algorithme suivant :

1. Tirer aléatoirement $s_2 \in (\mathbb{Z}/n_2\mathbb{Z})^\times, \ldots, s_K \in (\mathbb{Z}/n_K\mathbb{Z})^\times$.
2. Calculer $z_i = s_i^{e_i} \mod n_i$ pour tout $i \in \{2, \ldots, K\}$.
3. Calculer $s_1 = \left(z_2 \oplus \cdots \oplus z_K \oplus H(\boldsymbol{m})\right)^{d_1} \mod n_1$.
4. Retourner $s = (s_1, \ldots, s_K)$.

Pour vérifier la signature de $\boldsymbol{m}$, on effectue alors le test :

$$(s_1^{e_1} \mod n_1) \oplus (s_2^{e_2} \mod n_2) \oplus \cdots \oplus (s_K^{e_K} \mod n_K) = H(\boldsymbol{m})$$