Examen – 27 avril 2026

Aucun document n'est autorisé. Sauf indication contraire, les réponses doivent être soigneusement justifiées pour obtenir tous les points. Les exercices sont indépendants, mais pas toutes les questions. Vous pouvez admettre un résultat d'une question précédente en l'indiquant clairement. Vous pouvez répondre en anglais ou en français. L'examen dure 2 heures.

Il y a trois exercices, ne pas oublier de tourner la page !

Exercice 1 (sur 5 pts). *Questions courtes*

1. Attribuer chacune des trois notions de sécurité IND-CPA, PRP et EUF-CMA à l'une des trois primitives chiffre par bloc, chiffrement à clé publique, MAC.
2. Soit $N = p \times q$ où p et q sont des nombres premiers distincts. Exprimer le nombre d'éléments inversibles dans $\mathbb{Z}/N\mathbb{Z}$ en fonction de p et q. *Justifier.*
3. Dans un système de chiffrement à clé publique, deux clés sont utilisées : les clés de *chiffrement* et de *déchiffrement*. Laquelle est censée être *publique* et laquelle doit restée *privée* ? *Justifier.*
4. On rappelle les trois problèmes relatifs à un groupe cyclique $G = \langle g \rangle$:
   - CDH : étant donné g^a et g^b, calculer g^{ab} ;
   - DDH : étant donné g^a, g^b et g^c, déterminer si $g^c = g^{ab}$;
   - DLP : étant donné g^a, calculer a.

   Trier les trois problèmes par ordre croissant de difficulté. *Justifier.*
5. Définir ce qu'est un mécanisme d'encapsulation de clé (*Key Encapsulation Mechanism*), dans quelle situation il peut être utilisé, et quels en sont les bénéfices.

Exercice 2 (sur 6 pts). *Signatures de Lamport*

Soit $H : \{0,1\}^{<N} \to \{0,1\}^m$ une fonction de hachage et $\mathcal{M} = \{0,1\}^k$ l'espace des messages. Le **schéma de signature de Lamport** est défini par

- $\mathsf{Gen}_n()$:
  1 pour $1 \le i \le k$ et $b \in \{0,1\}$:
  2 $\quad x_i^{(b)} \twoheadleftarrow \{0,1\}^n$ (uniformément)
  3 $\quad y_i^{(b)} \leftarrow H(x_i^{(b)})$
  4 renvoyer $\mathsf{sk} = \begin{pmatrix} x_1^{(0)} & x_2^{(0)} & \cdots & x_k^{(0)} \\ x_1^{(1)} & x_2^{(1)} & \cdots & x_k^{(1)} \end{pmatrix}$ and $\mathsf{vk} = \begin{pmatrix} y_1^{(0)} & y_2^{(0)} & \cdots & y_k^{(0)} \\ y_1^{(1)} & y_2^{(1)} & \cdots & y_k^{(1)} \end{pmatrix}$
- $\mathsf{Sign}_{\mathsf{sk}}(m)$:
  0. écrire $m = m_1 \| \dots \| m_k$ où $m_i \in \{0,1\}$ pour $1 \le i \le k$
  1. renvoyer $\sigma = (x_1^{(m_1)}, x_2^{(m_2)}, \dots, x_k^{(m_k)})$

Par exemple avec $k = n = 3$, si $\mathsf{sk} = \left(\begin{smallmatrix} 010 & 011 & 101 \\ 110 & 100 & 001 \end{smallmatrix}\right)$, $\mathsf{Sign}_{\mathsf{sk}}(001) = (010, 100, 001)$.

1. **i.** Décrire l'algorithme de vérification. *L'algorithme doit être complètement spécifié, y compris ses entrées.*
   ii. Donner une condition nécessaire et suffisante sur les clés pour que l'algorithme de vérification n'accepte que des signatures valides.

2. Donner une attaque de récupération de clé à deux messages choisis sur le schéma de signature de Lamport. *C'est-à-dire, un adversaire qui peut effectuer deux requêtes de signature de son choix peut calculer la clé de signature.*

3. Dans le schéma de signature original, l'espace des messages est fixé à $\{0,1\}^k$. On veut utiliser le schéma de signature avec des messages de longueur *au plus* k. Une solution simple est de définir la signature d'un message m de longueur $\ell \leq k$ comme $(x_1^{(m_1)}, \ldots, x_\ell^{(m_\ell)})$.

- **i.** Donner une attaque de forgerie universelle à un message choisi contre cette variante. *C'est-à-dire, un adversaire peut calculer une signature valide σ pour un message donné m, après avoir obtenu la signature σ' d'un autre message m' de son choix.*
- **ii.** Proposer une variante du schéma de signature qui permet de signer tout message de longueur $\leq k$, avec des clés de taille $2(k + \lceil \log k \rceil)$. Justifier que l'attaque précédente ne s'applique pas.

Exercice 3 (sur 9 pts). *Fonction de hachage de Pedersen*

Soit $G = \langle g \rangle$ un groupe cyclique d'ordre premier p et $h = g^a \in G$. On définit la fonction de compression $f : \{0, ..., p-1\} \times \{0, ..., p-1\} \to G$ par $f(x, y) = g^x h^y$.

1. Montrer que si un adversaire est capable de calculer des logarithmes discrets dans le groupe G, il peut résoudre les problèmes suivants. *Décrire une attaque explicite pour chaque problème.*

- **i.** Première préimage : étant donné h et t, calculer (x, y) tel que $f(x, y) = t$.
- **ii.** Deuxième préimage : étant donné h et (x_0, y_0), calculer $(x_1, y_1) \neq (x_0, y_0)$ tel que $f(x_0, y_0) = f(x_1, y_1)$;
- **iii.** Collision : étant donné h, calculer (x_0, y_0) et $(x_1, y_1) \neq (x_0, y_0)$ tels que $f(x_0, y_0) = f(x_1, y_1)$.

2. On prouve maintenant l'inverse. Supposons qu'un adversaire soit capable de trouver une collision $f(x_0, y_0) = f(x_1, y_1)$ avec $(x_0, y_0) \neq (x_1, y_1)$. Montrer qu'il peut calculer le logarithme discret a de h.

3.
- **i.** Rappeler le nom et le coût du meilleur algorithme générique pour le DLP dans un groupe d'ordre p.
- **ii.** Rappeler le nom et le coût du meilleur algorithme générique de recherche de collisions pour une fonction de hachage à valeurs dans un ensemble de taille p.
- **iii.** Rappeler le meilleur algorithme générique et son coût pour calculer une première (resp. deuxième) préimage pour une fonction de hachage à valeurs dans un ensemble de taille p.
- **iv.** Diriez-vous que f a une *bonne* résistance à la première (resp. deuxième) préimage ? une *bonne* résistance aux collisions ? *Justifier.*

Pour construire une fonction de hachage à partir de la fonction de compression, on a accès à une bijection calculable efficacement $\pi : G \to \{0, \ldots, p-1\}$. On voit un élément de $\{0, \ldots, p-1\}$ comme un *bloc* et on définit des fonctions de hachages qui prennent en entrée un message de n blocs, $n \leq N$, et renvoient un résumé (*digest*) d'un seul bloc. On propose deux constructions :

— on fixe $h \in G$, on définit $\bar{f} : \{0, \ldots, p-1\} \times \{0, \ldots, p-1\} \to \{0, \ldots, p-1\}$ par $\bar{f}(x, y) = \pi(f(x, y)) = \pi(g^x h^y)$, on construit H à partir de $\bar{f}$ avec la construction de Merkle-Damgård ;
— on fixe $h_1, \ldots, h_N \in G$ et pour tout $n \leq N$, on définit $F(x_1 \| \cdots \| x_n) = \pi\left(\prod_{i=1}^n h_i^{x_i}\right)$ où chaque $x_i \in \{0, \ldots, p-1\}$.

4.
- **i.** Expliquer la construction de Merkle-Damgård pour construire une fonction de hachage H à partir de $\bar{f}$.
- **ii.** Quelle construction fournit la fonction de hachage la plus efficace pour des messages de n blocs ? *La mesure pertinente de complexité pour la comparaison est le nombre d'exponentiations nécessaires pour hacher le message de n blocs.*