HAI709I - TD 7

Rocco Mora

3 Novembre 2025

Exercice 1 :

Nous introduisons de manière informelle deux notions alternatives à la résistance aux collisions pour les fonctions de hachage.

- **Résistance à la deuxième préimage :** Étant donné s et x uniforme, il est difficile pour un adversaire PPT de trouver $x' \neq x$ tel que $H^s(x') = H^s(x)$.
- **Résistance à la préimage:** Étant donné s et $y = H^s(x)$ pour un x uniforme, il est difficile pour un adversaire PPT de trouver x' tel que $H^s(x') = y$.

Expliquez de manière informelle pourquoi une fonction de hachage résistante aux collisions est également résistante à la deuxième préimage, et pourquoi une fonction de hachage pour entrées de longueur arbitraire et résistante à la deuxième préimages est également résistante aux préimages.

Exercice 2 :

Soient $(\texttt{Gen}_1, H_1)$ et $(\texttt{Gen}_2, H_2)$ deux fonctions de hachage. Définissons $(\texttt{Gen}, H)$ de telle sorte que $\texttt{Gen}$ exécute $\texttt{Gen}_1$ et $\texttt{Gen}_2$ pour obtenir respectivement les clés s_1 et s_2. Définissez ensuite $H^{s_1,s_2}(x) = H_1^{s_1}(x)||H_2^{s_2}(x)$.

- Prouvez que si au moins l'une des fonctions $(\texttt{Gen}_1, H_1)$ et $(\texttt{Gen}_2, H_2)$ est résistante aux collisions, alors $(\texttt{Gen}, H)$ est résistante aux collisions.
- Déterminez si une affirmation analogue s'applique respectivement à la résistance à la deuxième préimage et à la résistance à la préimage.

Exercice 3 :

Soit $(\texttt{Gen}, H)$ une fonction de hachage résistante aux collisions. Est-ce que $(\texttt{Gen}, \bar{H})$ défini par $\bar{H}^s(x) = H^s(H^s(x))$ est nécessairement résistant aux collisions ?

Exercice 4 :

Montrez comment trouver une collision dans la construction de l'arbre de Merkle si t n'est pas fixé. Plus précisément, montrez comment trouver deux ensembles d'entrées $x_1, \ldots, x_t$ et $x'_1, \ldots, x'_{2t}$ tels que $\mathcal{MT}_t(x_1, \ldots, x_t) = \mathcal{MT}_{2t}(x'_1, \ldots, x'_{2t})$.

Exercise sur ordinateur :

Nous allons créer avec Python l'arbre de Merkle pour une liste de transactions

1. Étape 1 : installez les paquets `pycryptodome`.
2. Étape 2 : complétez le script suivant

```
from Crypto.Hash import SHA256

def sha256(data: bytes) -> bytes:
        return SHA256.new(data).digest()

# ----------------------------------
# Étape 1 : Transactions -> Feuilles
# ----------------------------------


transactions = [
b"Alice paie Bastien 5 euros",
b"Bastien paie Charles 20 euros",
b"Charles paie Delphine 50 euros",
b"Delphine paie Alice 10 euros"
]

# leaves est la liste des hachages de chaque transaction
leaves =
# afficher la liste

# ----------------------------------
# Étape 2 : Construire la fonction pour la racine de l'arbre de Merkle
# on suppose que le nombre de transactions soit une puissance de 2
# ----------------------------------
def merkle_root(leaves):
        # initialiser le niveau de l'arbre avec le feuilles
        current_level =
        # construction de l'arbre
        while len(current_level) > 1:
                new_level = []
                for i in range(0, len(current_level), 2):
                        # combiner les deux noeuds suivants
                        combined =
                        # ajouter le hachage au nouveau niveau

                # mettre à jour le niveau
                current_level =
        # racine en sortie
        return

# afficher la racine au format hexadécimal
```