# Exercice 1

Function 1 :

Rapide, devrait avoir des collisions, plusieurs messages peuvent avoir le même hash

Collision -> 2 trucs différents qui ont le même hash

Très efficace mais :

* Facile de trouver des collisions, par exemple, en permutant de blocs
* x est un préimage pour X

Function 2 :

Même problème de collision

par contre le calcul d’un pré-image devient difficile

Function 3 :

* plein de collision car le produit est commutatif : en changeant l’ordre des blocs on garde le même haché
* Soit h…   
  Pour trouver un préimage on peut factoriser le hash pour trouver les ri, pi-pn ert un préimage pour h

# Exercice 2

1. L’entreprise pourrait stocker uniquement le hash de chaque fichier, la vérification serait correcte mais il serait impossible de retrouver les données initiales en cas de perte. Le changement de fonction de hachage ne changerait rien au problème.

Prof :

Il suffit de stocker les hashs. Le problème est indépendant de la fonction de hachage utilisé.

2. La solution est meilleure, en effet, l’entreprise sera maintenant obligée de garder une copie

des fichiers pour pouvoir renvoyer le hash du fichier concaténé au sel sinon celui-ci sera incorrect.

Prof :

Attention à la longueur du défi si trop petit. On peut calculer et stocker toutes les réponses possibles. SHA-1 utilise la construction Merkle-Danguard: il suffit donc de stocker le dernier bloc, l’état interne, la longueur du fichier pour pouvoir calculer la réponse.