

TD 1-2-3. Fonctions de hachage

Sergey Kirgizov

Polytech Dijon

XASH — une famille de fonctions de hachage basées sur XOR

👉 **EXERCICE 1.** Réaliser une fonction de compression $\text{comp}(a, b)$: ou exclusif, bit à bit, de deux expressions a et b de 4 octets chacune.

👉 **EXERCICE 2.** À partir de la fonction $\text{comp}(a, b)$ et un vecteur d'initialisation

$IV = 01111000\ 01100001\ 01110011\ 01101000$

construire trois fonctions de hachage ($XD(c)$, $XDD(c)$, $XDDD(c)$) en utilisant le schéma de Merkle-Damgård avec trois méthodes de rembourrage suivantes :

Fonction	Type de rembourrage
$XD(c)$	par zéros;
$XDD(c)$	par 100...0;
$XDDD(c)$	Merkle-Damgård strengthening (100...0 + la longueur du message)

Chaque fonction de hachage doit prendre en entrée une chaîne de caractères de longueur quelconque et produire un hash de longueur 4 octets.

💡 **ASTUCE :** Utiliser la notation binaire ou hexadécimale, afin d'afficher les résultats de hachage.

$(01111000\ 11111011\ 01110011\ 01101000)_2 = (78FB7368)_{16}$

👉 **EXERCICE 3.** Tester vos fonctions, calculer les hashes des différentes chaînes de caractères. Valeurs de référence :

```
xd ("ESIREM") == "787f3a3a"
xd ("ESIREM!!") == "787f1b1b"
xd ("") == "78617368"
```

👉 **EXERCICE 4.** Pour chaque méthode de bourrage :

- Trouver une collision
- Effectuer une attaque de préimage
- Effectuer une attaque de seconde préimage

👉 **EXERCICE 5.** Analyser l'uniformité et l'effet avalanche de trois fonctions.

Fonctions de hachage cryptographique

👉 **EXERCICE★ 6.** Implémenter et tester votre propre version de SHA-2.

👉 **EXERCICE★ 7.** Implémenter et tester votre propre version de SHA-3.

👉 **EXERCICE★★ 8.** Imaginez, réalisez et testez votre propre fonction de hachage.