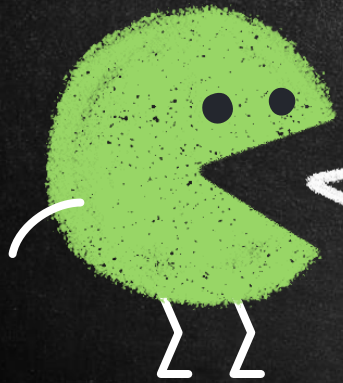
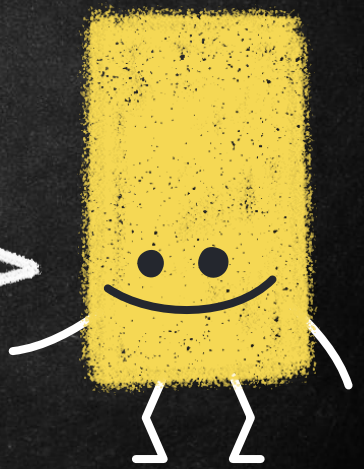
Sécurité des données
A2021

Hashing

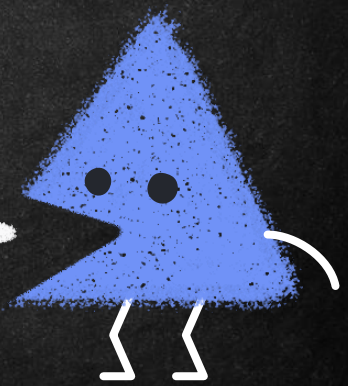




Yannick Charron

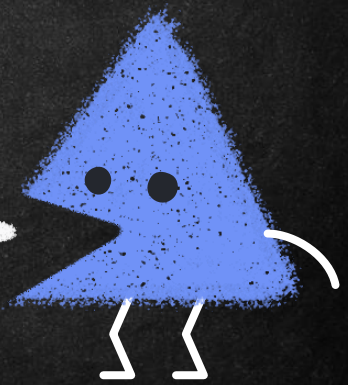
Plan de la séance

- Présentation du plan de cours
- Concept général
- Propriétés des fonctions
- Algorithmes
- Utilisations
- Exemple de code

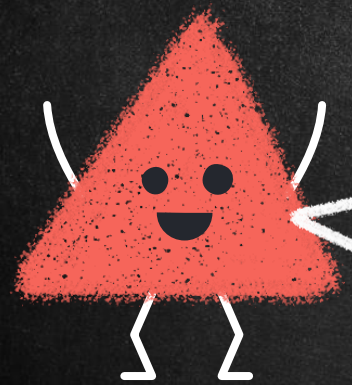


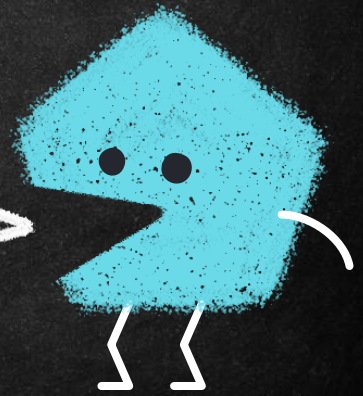
3 concepts différents

- Encoding
- Hashing
- Encryption



Hashing

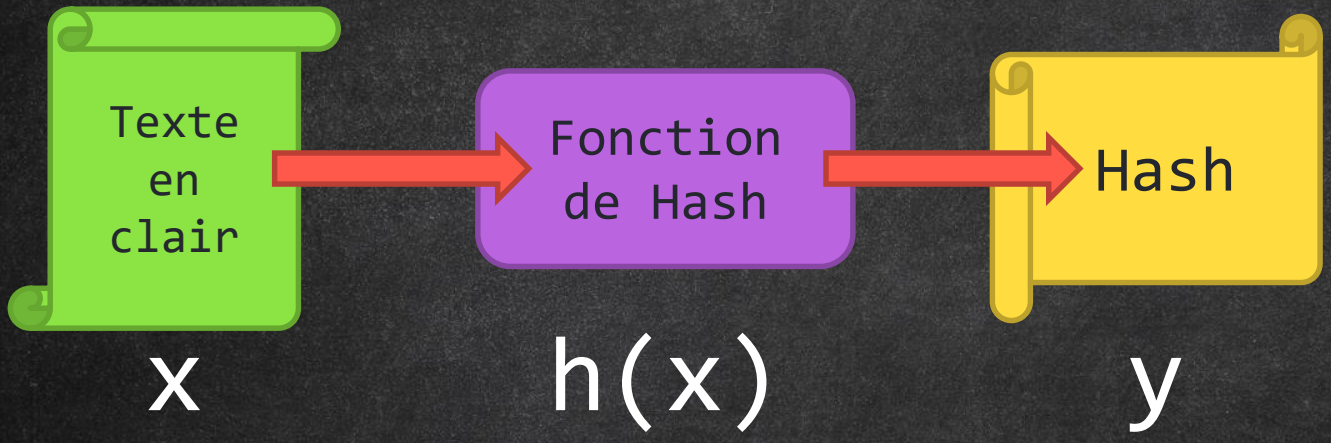


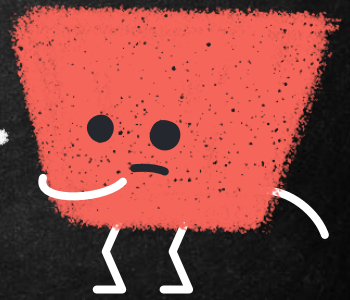


Concept général

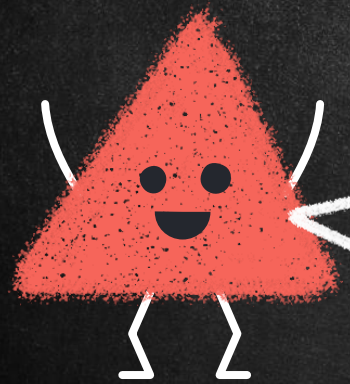
- Entrée de longueur possiblement infini
- Sortie de longueur fixe

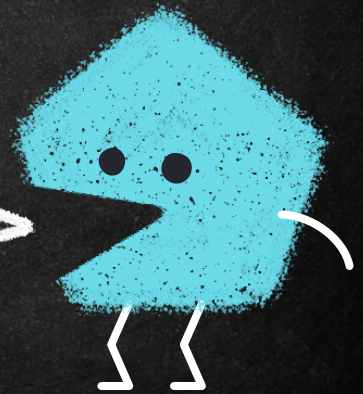
Concept Général



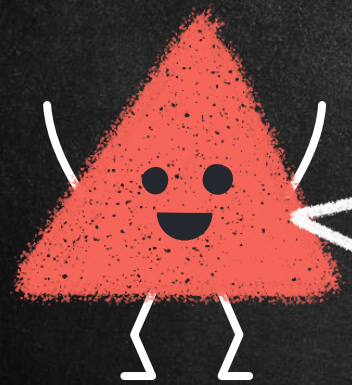


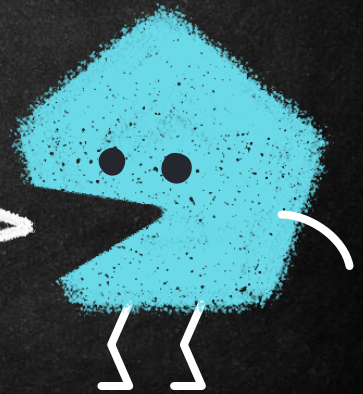
Fonction de Hash Simple



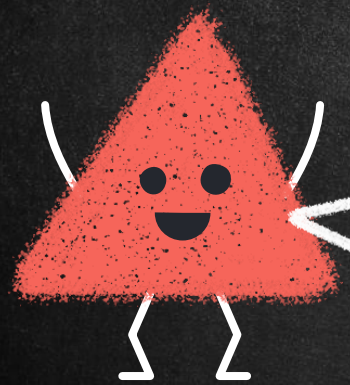


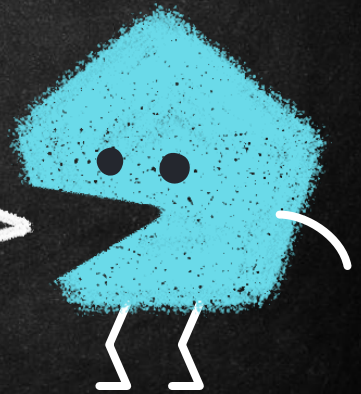
Fonction de Hash idéale





Algorithmes Cryptographiques





Propriétés d'une fonction de Hash

One-way

Impossible de revenir à la valeur d'entrée à partie de la sortie.

$x \rightarrow y$ impossible de faire $y \rightarrow x$

Collision-resistance

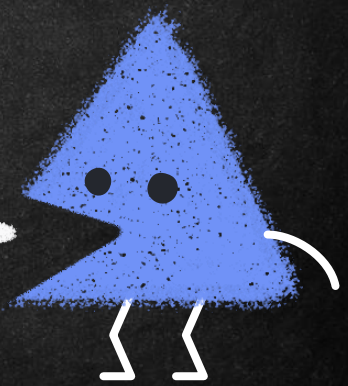
Impossible que deux x donnent le même y.

Lightning-fast

Être relativement rapide

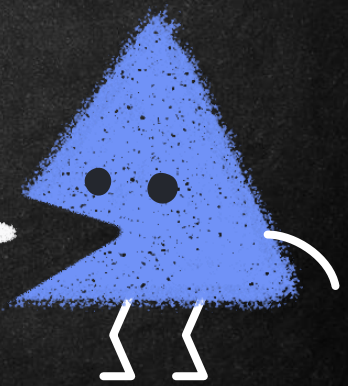
2 familles

- **MD5**
- Famille **SHA** **(Secure Hash Algorithms)**
  - SHA-0 : Rétronyme, problème majeur découvert rapidement
  - SHA-1 : Correction à SHA-0
  - SHA-2 : SHA256 et SHA512 (norme actuelle)
  - SHA-3 (Keccak)



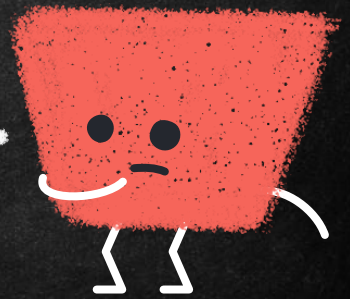
MD5

- Digest: 128 bits (32 hex)
- Problèmes dans l'algorithme
- Attaque devenue trop simple (Google)



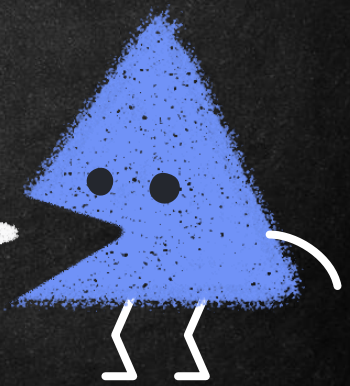
SHA1

- Digest: 160 bits (40 hex)
- Fin de vie en 2010
- https://shattered.io/

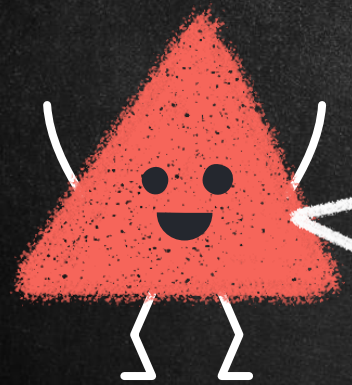


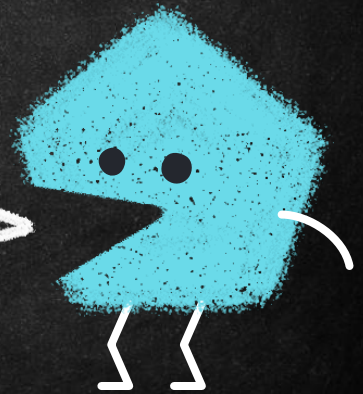
La fin d'un algorithme de Hash

- En théorie, ça va se produire, il faut seulement de la puissance de calcul.
- Infinité d'entrées pour un nombre fini de sorties
- Informatique quantique



Utilisations





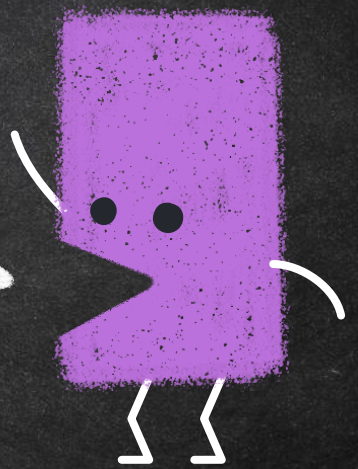
Intégrité des données

- Signature d'un document
  - Protection contre Virus, Malware, ...
  - Validation de gros fichiers rapidement
- Blockchain
  - Proof of work / Proof of stake
- Git commit hash
- Système de sauvegarde

Protection des mots de passe

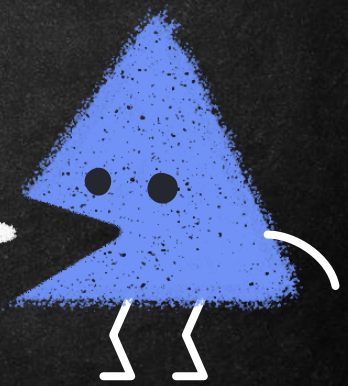
- Il ne faut pas sauvegarder le mot de passe en clair dans la base de données
- L'idée est de comparer un hash avec le hash sauvegardé

J’aimerais bien mettre mon grain de sel dans ce cours.



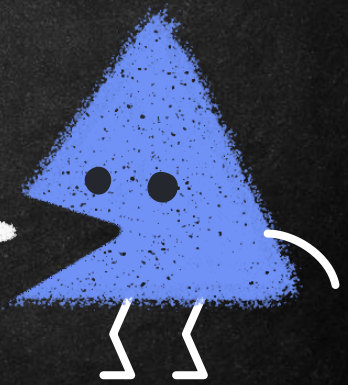
Salt ou Secret

- Permets d'éviter que la même valeur d'entrée donne le même hash de sortie
- Rendre la vie plus difficile
- Constant ou Aléatoire
- Concaténation



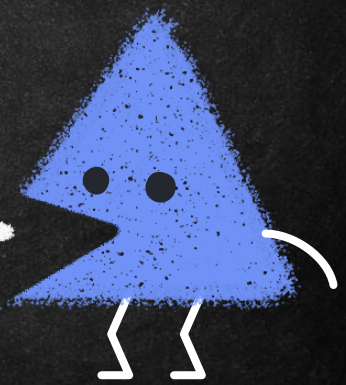
HMAC

- Key-hash message authentication code
- Permets de valider
  - Intégrité des données
  - Authenticité des données
- Nous allons y revenir



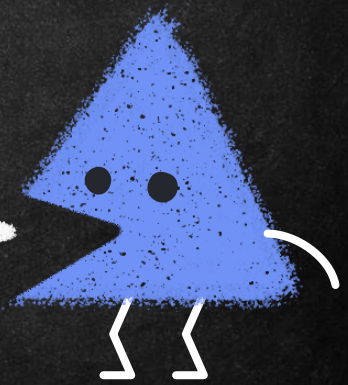
Encore mieux et la vraie façon

- Pbkdf2
- scrypt
- bcrypt
- argon2



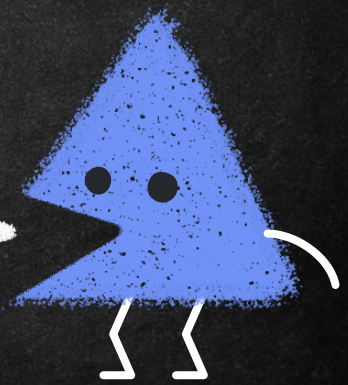
Quel choisir

- OWASP Cheat Sheet
- https://github.com/P-H-C



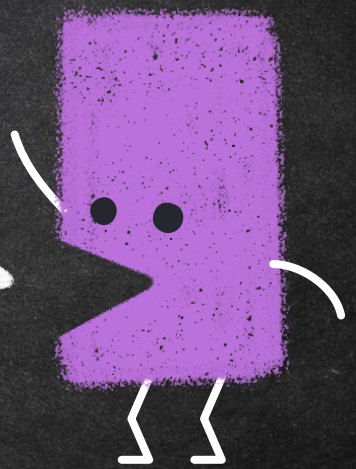
Types d'attaque

- Dictionnaire (sites compromis)
- Force-brute (Brute-force)
  - Rate-limit
- Table arc-en-ciel (Rainbow table)



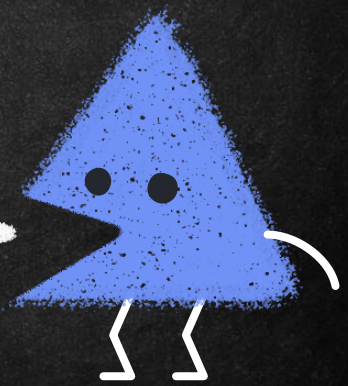
“

Toujours mieux un poivron avec mon sel



Peppering

- Crypter le hash de manière symétrique avant de le sauvegarder.
- Nous allons y revenir dans un prochain cours.



Si nous codions