Fonction de hachage cryptographique

Arthur Garnier

C'est une fonction déterministe qui prend n'importe quel message en entrée et donne un message de longueur fixe. Trouver un message avec le hash donné est très difficile. Une petite modification du message implique une grande modification du hash.

Paradoxe des anniversaires : Dans une classe de 23 personnes il y a 50% de chances qu'il y ait 2 anniversaires le même jour.

En général $\approx \sqrt{nbr - possibilite}$ pour $\geq 50\%$

Exemple : Si on hashe sur 128bits, 2^{64} messages $\rightarrow proba \geq 50\%$

Le coût pour trouver une collision MD5 = 2^{25} (1 calcul = 800 instructions)

- SHA-0 :
  - 2^{80} (anniversaires)
  - $2^{33,6}$ (attaque élaborées)
- SHA-1
  - 2^{80} (anniversaire)
  - 2^{61} (élaborées)

A part les collision ?

1 Les attaques à préfixe choisi

Sachant préfixe p, trouver m et m', tels que H(p||m) = H(p||m')

MD5 = Meilleure attaque pour résoudre ça = 2^{50} évaluations

2 applications de ces attaques à préfixes choisi :

- Attaque Nostradamus
- Virus Flame

En pratique les hashs sont utilisés pour :

- Antivirus
- Checksum
- Mots de passe

C'est sûr car il n'y a pas d'attaque mathématique qui donne un message qui marche pour un message donné.

En pratique, il existe l'attaque par dictionnaire : Vous avez un dico (exemple : RockYou) de mot de passe possibles, et l'on essayer de hasher chaque mot de passe pour le comparer avec la base de données.

Pour éviter ce genre d'attaque (Attaque par table arc-en-ciel) : Utilisation d'un sel.

On utilise une chaine aléatoire que l'on concatène à un mot de passe (un différent par utilisateur).

2 Bcrypt et scrypt

Ce sont des fonctions paramètrables, c'est à dire que l'on peut choisir la quantité de travail nécessaire pour calculer le hash.

De plus il n'est pas facile à accélérer.

Cas d'étude = Ashley Madison

- 36 millions de mdp
- BCrypt 4096 rounds
- Sel unique par utilisateur

Chercheurs : 4 GPU (11 millions MD5/s)

156 bcrypt/s

Casser 4000 mdp en 24h ; → Tout casser en 116958 ans