



L3 informatique, Sécurité Appliquée

Fonctions de hachage

Jean-François COUCHOT

Université de Franche-Comté, UFR-ST

Plan



Introduction

Théorie

Constructions

Plan



Introduction

Motivation

Hachage de mots de passe

Code d'Authentification de Message

Théorie

Constructions

Plan



Introduction

Motivation

Hachage de mots de passe

Code d'Authentification de Message

Théorie

Constructions

Objectif : comparer des messages m_1 et m_2

Pour ne pas analyser tous les bits de m_1 et m_2 (fastidieux)

- Souhait : une fonction H
 - Réduisant m à son empreinte $H(m)$ de taille fixe
 - Implémentant l'effet d'avalanche : même si m_1 et m_2 ne diffèrent que d'1 bit $\rightsquigarrow H(m_1)$ et $H(m_2)$ complètement différents
 - Reproductible : $H(m)$ ne change pas au cours du temps

Exécution de la fonction sha256 sur 2 messages proches

```
# coding: utf-8
import hashlib
txt = input('Txt 1: ')
hash_object = hashlib.sha256(txt.encode('utf-8'))
print(hash_object.hexdigest())
```

```
Txt 1: L3 securite, le cours indispensable pour tout etudiant serieux
eb55b53cdfd0003069887aee888dfb540dba07af55b8c0600c4e5ba1483ed7f5
Txt 2: L3 securite, le cours indispensable pour tout etudiant serieuy
947a3318d45412681eb0728209c1eac0f44b5709b9642584adf270d484971107
```

Applications des fonctions de hachage



Applications cryptographiques

- Intégrité d'un message lors d'une transmission (vu)
- Vérification d'un mot de passe dont est stockée uniquement l'empreinte
- Signature de l'empreinte plutôt que du message lui-même (après crypto asymétrique)

Applications non-cryptographiques

- Table de hachage

Plan



Introduction

Motivation

Hachage de mots de passe

Code d'Authentification de Message

Théorie

Constructions

Hachage de mots de passe



Remarque préliminaire importante

On ne stocke jamais un mot de passe en clair !

Hachage de mots de passe

Le hachage de mots de passe est un cas particulier très usuel d'utilisation des fonctions de hachage. L'idée est la suivante :

1. l'utilisateur définit son mot de passe ;
2. le système enregistre une empreinte du mot de passe ;
3. quand l'utilisateur se connecte, il donne son mot de passe ;
4. le système calcule l'empreinte du mot de passe et la compare à l'empreinte stockée.

Attaque sur une base de données de mots de passe hachés

- l'espace de recherche est-il suffisamment grand ?
- la fonction de hachage utilisée est-elle rapide ?



Espace de recherche

Ce paramètre dépend de l'utilisateur. Si l'utilisateur utilise un mot du dictionnaire, l'espace de recherche est ridiculement faible !

- mots du dictionnaire : 130000 $\approx 2^{17}$
- deux mots du dictionnaire : 2^{34}
- trois mots du dictionnaire : 2^{51}

Ordres de grandeur

Règles en vigueur sur l'ENT : a minima

- 8 caractères
- 1 caractère spécial (parmi 26)
- 2 chiffres
- 1 caractère alphabétique

Longueur	6	7	8	9	10
a-z (26)	2^{28}	2^{32}	2^{37}	2^{42}	2^{47}
a-zA-Z (52)	2^{34}	2^{39}	2^{45}	2^{51}	2^{57}
a-zA-Z0-9 (62)	2^{35}	2^{41}	2^{47}	2^{53}	2^{59}
+ [punct] (94)	2^{39}	2^{45}	2^{52}	2^{58}	2^{65}
ENT	-	-	2^{48}	2^{55}	2^{61}

Vitesse de la fonction de hachage



Vitesse de la fonction de hachage

- Une fonction de hachage est normalement destinée à être rapide pour être capable de hacher de grandes quantité de données.
 - Pour limiter la portée d'une attaque sur des mots de passe, on a intérêt à ce que la fonction de hachage soit la plus lente possible.
- On n'utilise pas les mêmes fonctions de hachages pour les mots de passe et pour les autres utilisations cryptographiques

Hachage cryptographique pour les mots de passe

Généralités

Dans les fonctions suivantes, on peut souvent régler :

- le temps d'exécution
- l'espace mémoire utilisé

Exemples (Hachage cryptographique pour les mots de passe)

- bcrypt, 1999, basé sur Blowfish
- PBKDF2, 2000, RFC 2898
- scrypt, 2012, RFC 7914
- Argon2, 2015, gagnant du Password Hashing Competition

Plan



Introduction

Motivation

Hachage de mots de passe

Code d'Authentification de Message

Théorie

Constructions

Code d'Authentification de Message



Définition (Code d'Authentification de Message (MAC))

Un **Message Authentication Code** (MAC) est une association entre une fonction de compression et une clef secrète permettant de vérifier à la fois l'intégrité et l'authenticité des données reçues.

Utilisation

1. Alice veut envoyer un message m à Bob
2. Elle calcule le MAC avec une clef secrète K qu'elle partage avec Bob
3. Elle envoie son message et le MAC à Bob
4. Bob recalcule le MAC et vérifie qu'il est identique à celui d'Alice

Hash-based Message Authentication Code (HMAC)



Hash-based Message Authentication Code (HMAC)

- Défini dans la RFC 2104
- Utilisé dans IPSec et TLS
- $HMAC(K, m) = H((K \oplus opad) || H((K \oplus ipad) || m))$
 - H est une fonction de hachage cryptographique
 - K est la clef secrète
 - $opad = 0x5c5c5c \dots 5c5c$
 - $ipad = 0x363636 \dots 3636$
 - $||$ est la concaténation
- Plusieurs variantes suivant H : HMAC MD5 (bof), HMAC-SHA-256

Plan



Introduction

Théorie

Formalisation

Recherche de Collisions

Constructions

Plan



Introduction

Théorie

Formalisation

Recherche de Collisions

Constructions



Définition (Fonction de hachage $H : \mathbb{B}^* \rightarrow \mathbb{B}^n$)

Fonction qui transforme un message m de taille quelconque en une valeur $h = H(m)$ de taille fixe n .

- h : appelé empreinte, le résumé ou la somme de contrôle de m
- m : appelé l'antécédent ou la préimage de h

Définition (Collision)

Il y a collision entre m_1 et m_2 , $m_1 \neq m_2$ si $H(m_1) = H(m_2)$

Propriétés pour $H : \mathbb{B}^* \rightarrow \mathbb{B}^n$



Définition (Résistance à la préimage)

Pour toute empreinte h , il est difficile de trouver m tel que $H(m) = h$.

Conséquences :

- Difficile de retrouver un mdp m étant donnée une empreinte h stockée
- On parle de fonction à sens unique

Définition (Résistance à la seconde préimage)

Pour tout m_1 , il est difficile de trouver $m_2 \neq m_1$ tel que $H(m_1) = H(m_2)$.

Remarque : on a ici une connaissance supplémentaire qui est m_1

Définition (Résistance aux collisions)

Il est difficile de trouver m_1 et $m_2 \neq m_1$ tels que $H(m_1) = H(m_2)$.

Remarque : comme $H : \mathbb{B}^* \rightarrow \mathbb{B}^n$, il y a nécessairement des collisions

Plan



Introduction

Théorie

Formalisation

Recherche de Collisions

Constructions

Collision : nombre de tirages suffisant



Combien de tirages pour avoir une collision avec une probabilité 1/2 ?

Quel est la probabilité, en tirant au hasard k valeurs dans un ensemble de N valeurs, de tirer deux fois la même valeur ?

$$p(k) = 1 - \frac{N!}{(N-k)!} \cdot \frac{1}{N^k} \approx 1 - e^{-\frac{k(k-1)}{2N}}$$

Et inversement : $k(p) \approx \sqrt{2 \cdot N \ln \left(\frac{1}{1-p} \right)}$

Paradoxe des anniversaires

Combien de personnes sont suffisantes pour que la probabilité qu'au moins deux personnes aient leur anniversaire le même jour soit supérieure à $\frac{1}{2}$? 23 !

Conséquences pour H sur n bits

Collision avec une probabilité supérieure à $\frac{1}{2}$ en $O(\sqrt{2^n}) = O(2^{\frac{n}{2}})$: seulement $\frac{n}{2}$ bits de sécurité.

Plan



Introduction

Théorie

Constructions

Message Digest 5 (MD5)

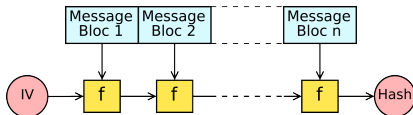
SHA-3

Construction de Merkle-Damgård



Idées principales

- On découpe le message m en blocs de taille égale m_i
- On utilise une fonction de compression $f : h_0 = IV, h_{i+1} = f(m_i, h_i)$
- Propriété : si f est résistante aux collisions, alors la construction l'est
- Appliquées dans MD5, SHA-1, SHA-2.....



Plan



Introduction

Théorie

Constructions

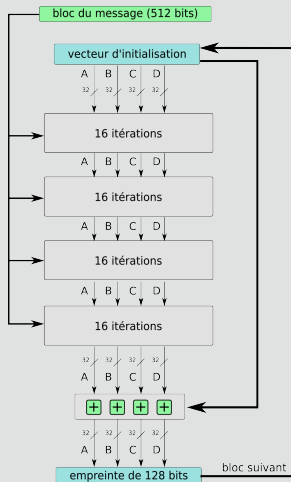
Message Digest 5 (MD5)

SHA-3



Généralités

- Inventé par Ronald Rivest en 1991
- Spécifié dans la RFC 1321
- Amélioration de MD4
- Bloc de 512 bits
- Empreinte de 128 bits
- 64 rondes (ou itérations)

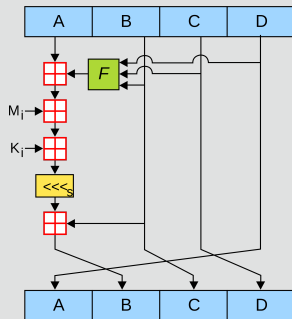


MD5 : Description d'une itération



Description d'une itération

- Le message M est découpé en bloc M_i
- Les constantes K_i sont fixées
- État : A, B, C, D
- Fonctions F :
 - $F_1(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$
 - $F_2(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$
 - $F_3(B, C, D) = B \oplus C \oplus D$
 - $F_4(B, C, D) = C \oplus (B \vee \neg D)$





Historique

- Résistance aux collisions
 - 1996 : une faille est trouvée dans la fonction de compression
 - 2004 : premières collisions générées
 - Résistance à la préimage
 - 2009 : Attaque en $2^{123.4}$
- MD5 n'est plus une fonction sûre cryptographiquement

Attaque hashcat de example0.hash stockant des empreintes MD5 (-m 0)

- `hashcat -m 0 -a 0 example0.hash example.dict -r d3ad0ne.rule -show`

Avec un dictionnaire (`-a 0 example.dict`) et des règles (`-r d3ad0ne.rule`) \rightsquigarrow 646 mdp reconstruits en qqes secondes

- `hashcat -m 0 -a 3 example0.hash`

Brute force (`-a 0`) \rightsquigarrow 1700 mdp reconstruits en qqes heures



Introduction

Théorie

Constructions

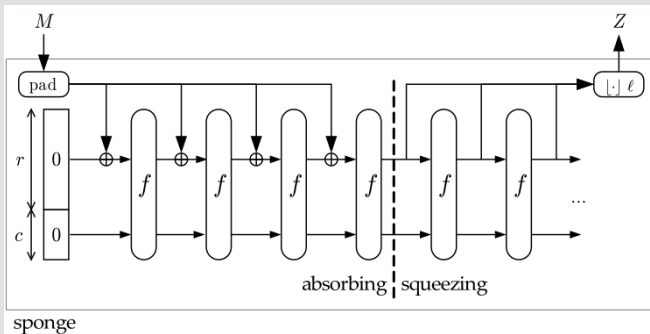
Message Digest 5 (MD5)

SHA-3

SHA-3 Keccak : construction éponge



Intuition



- Absorption : blocs m_i intégrés successivement par XOR dans l'état interne et application de f
- Essorage : r premiers bits évacués puis application de f