INFO 910
Les collisions MD5

ASCENCI Alexandre, QIN Jia

Introduction

L'objectif de ce rapport est de pouvoir répondre aux questions suivantes: "MD5 est-il obsolète? Y a-t-il encore un intérêt à utiliser cet algorithme?".

Il s'agit donc d'un document destiné à éveiller votre curiosité sur le sujet, et à vous donner suffisamment d'informations pour que vous puissiez appréhender le sujet.

Nous consacrerons un premier temps aux concepts liés à MD5, puis nous enchainerons en définissant ce qu'est une collision. Nous donnerons des exemples d'attaques de collisions MD5, et nous verrons quels impacts cela a pu engendrer.

L’algorithme MD5

Historique

D’après la thèse de Bart Preneel de 1993, R. Rivest de RSA Data Security Inc. a conçu une série de fonctions de hachage, qui ont été nommées MD pour « message digest » suivi d'un nombre. **MD1** est un algorithme propriétaire; **MD2** a été suggéré en 1990; **MD3** n'a jamais été publié, et semble avoir été abandonné par son concepteur.

Ronald Linn Rivest

MD2 est une fonction de hachage conçue par le professeur Ronald Rivest du Massachusetts Institute of Technology en 1989. C'est un algorithme de signature. Plutôt destiné aux processeurs de type 8 bits, le MD2 n'est plus tellement employé. En 2004, une attaque décrite par Muller permet de forger un document à partir d'une signature en seulement 2^{104} opérations au lieu des 2^{128} nécessaires dans le cas d'une recherche exhaustive. Même si cette attaque est tout de même très gourmande et impraticable à l'heure actuelle, le MD2 n'est à ce titre plus considéré comme sûr.

MD4 est un algorithme de hachage conçu par le professeur Ronald Rivest du Massachusetts Institute of Technology en 1990. La taille de la signature est de 128 bits. L'algorithme a été abandonné au profit du MD5 après la découverte de faiblesses dans sa conception. D'autres attaques encore plus efficaces ont suivi, notamment par Hans

Dobbertin du service du chiffre allemand et l'équipe chinoise à l'origine de l'attaque sur MD5. À ce titre, le MD4 ne peut en aucun cas être considéré comme cryptographiquement sûr puisque des collisions peuvent être générées avec un nombre d'opérations de l'ordre de 2^8 opérations. Cette magnitude est très faible en comparaison des 2^{64} nécessaires pour une attaque des anniversaires.

Six ans plus tard, en 1996, une faille qualifiée de « grave » (possibilité de créer des collisions à la demande) est découverte et indique que **MD5** devrait être mis de côté au profit de fonctions plus robustes comme SHA-1.

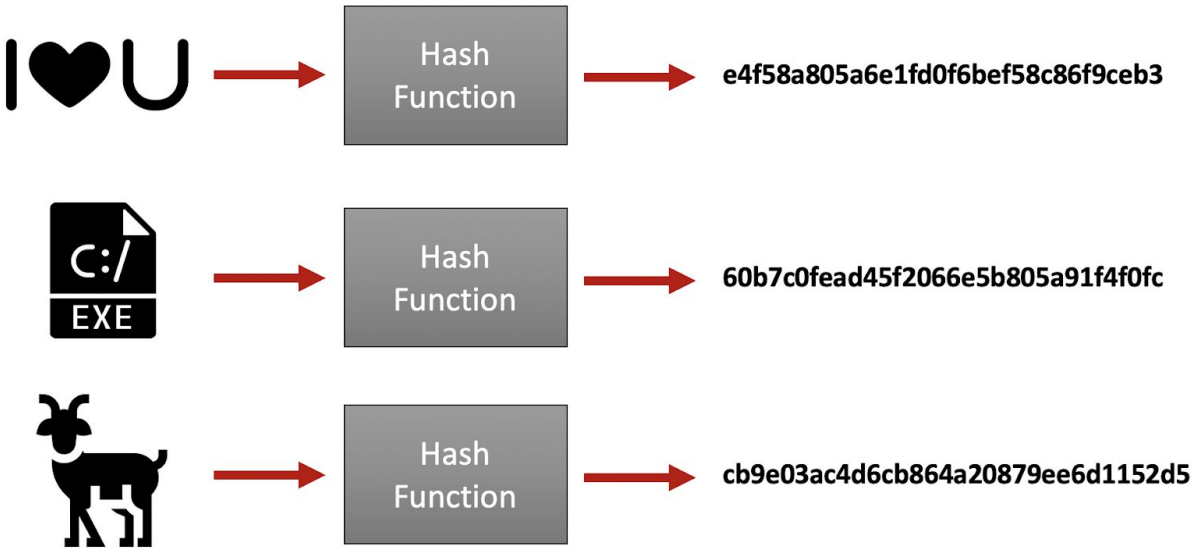
En 2004, une équipe chinoise découvre des collisions complètes. **MD5** n'est donc plus considéré comme sûr au sens cryptographique. On suggère maintenant d'utiliser plutôt des algorithmes tels que SHA-256, RIPEMD-160 ou Whirlpool.

Définition

Dans les définitions que nous pouvons trouver en ligne, il est dit que l'algorithme MD5 est une fonction de hachage cryptographique. Avant d'aller plus loin, il est nécessaire de définir ce concept.

Il est précisé qu'une fonction de hachage sûre est une fonction qui calcule, à partir d'un contenu donné en entrée, une empreinte numérique telle que deux entrées différentes donnent systématiquement deux empreintes différentes. L'algorithme prévient quiconque de déterminer le contenu de départ à partir du hash de sortie. De plus, si deux contenus ont le même hash, alors ils sont supposés identiques.

Cette fonction pourra par exemple être utilisée pour vérifier un mot de passe sans le compromettre, valider l'intégrité d'un fichier ou encore indexer des fichiers…



En revanche, MD5 est définie comme une fonction de hachage qui calcule, à partir d'un contenu donné en entrée, une empreinte numérique **avec une probabilité très forte** que deux entrées différentes donnent deux empreintes différentes.

On peut noter que pour l'algorithme MD5, deux entrées différentes ont une chance, bien qu'infime, d'avoir la même empreinte.

Algorithme

Fonctionnement

L'algorithme MD5 prend une entrée de taille arbitraire et produit un hash de 128 bits.

Préparer l'entrée

L'algorithme divise l'entrée en blocs de 512 bits. 64 bits qui serviront à sauvegarder la taille de l'entrée sont insérés à la fin du dernier bloc. Si le dernier bloc a une taille inférieure à 512 bits, des bits sont ajoutés à la fin. Ensuite, chaque bloc est divisé en 16 mots de 32 bits chacun.

Calculer les blocs

Soit A, B, C et D des mots de 32 bits. Ils sont contenus dans un buffer. Ils sont initialisés de la manière suivante:

- mot A: 01 23 45 67
- mot B: 89 ab cd ef
- mot C: fe dc ba 98
- mot D: 76 54 32 10

Soit quatre fonctions qui requièrent un mot de 32 bits en entrée et qui produisent une sortie de 32 bits:

- F(X,Y,Z) = (X and Y) or (not(X) and Z)
- G(X,Y,Z) = (X and Z) or (Y and not(Z))
- H(X,Y,Z) = X xor Y xor Z
- I(X,Y,Z) = Y xor (X or not(Z))

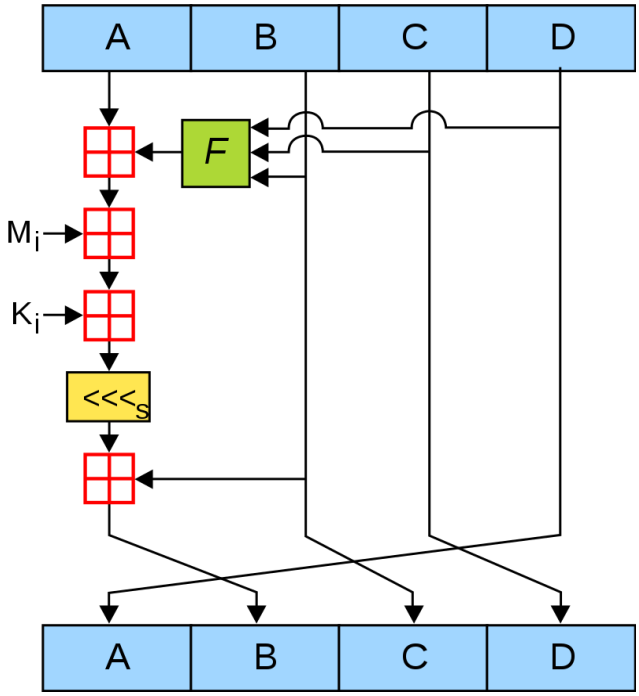
Soit une table contenant 64 éléments précalculée, où
chaque élément est calculé de la manière suivante:

- E = abs(sin(i + 1)) * 2^{32}

Le contenu des quatre mots du buffer (A,B,C,D) sont mélangés avec les mots de l'entrée en utilisant les fonctions F,G,H,I et les constantes de la table précalculée.

Il y a quatre tours, et chaque tour implique seize opérations.

Dans le schéma ci-dessous, on peut voir le déroulement d'une opération.



La sortie

Une fois que toutes les opérations sont terminées, A,B,C et D contiennent le résultat.

Utilisation

La plupart des langages de programmation proposent des librairies qui permettent d'utiliser des fonctions de hachage. En l'occurrence, MD5 en fait partie.

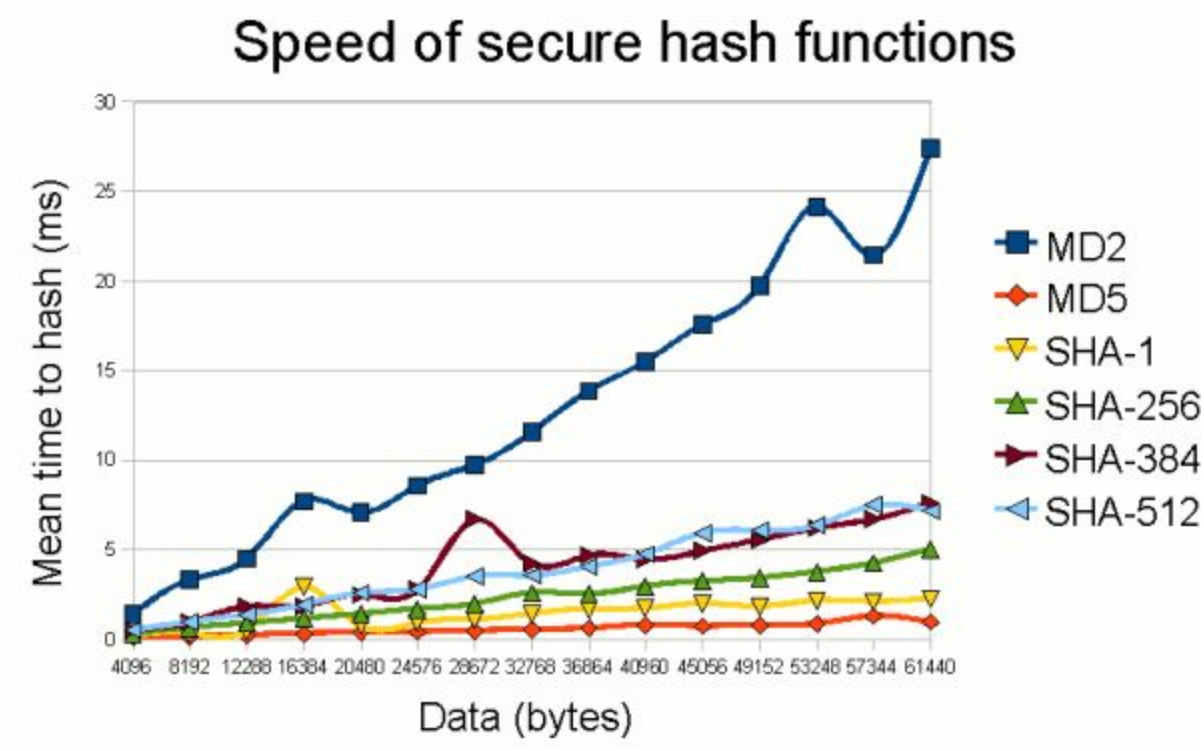
Par exemple, en PHP, on peut chiffrer une chaîne de caractères de la manière suivante:

```
1. <?php
2. $str = "Hello";
3. echo md5($str);
4. ?>
```

Ce script nous retourne la chaîne suivante: 8b1a9953c4611296a827abf8c47804d7

Avantages/Inconvénients

Le principal avantage de MD5 est que c'est un algorithme de hachage rapide. Des tests nous montrent que la génération d'un hash est généralement **plus de deux fois plus rapide** que d'autres fonctions de hachage telles que SHA.



En revanche, c'est aujourd'hui un algorithme considéré peu sécurisé, étant donné qu'il possède des failles. Si cet algorithme venait à être utilisé pour sécuriser des données sensibles sur un site web telles qu'un mot de passe, un attaquant interceptant les données saisies par un utilisateur pourrait décrypter le texte, et donc avoir accès au compte. Il est donc formellement déconseillé de l'utiliser en cryptographie.

Les attaques de collisions

Historique

À ses débuts, la fonction MD5 était considérée comme sûre, mais au cours du temps, des failles ont été découvertes dans son fonctionnement. Durant l'été 2004, il a été cassé par des chercheurs chinois, Xiaoyun Wang, Dengguo Feng, Xuejia Lai (co-inventeur du célèbre algorithme de chiffrement IDEA) et Hongbo Yu. Leur attaque a permis de découvrir une collision complète (deux messages différents qui produisent la même empreinte) sans passer par une méthode de type recherche exhaustive.

Sur un système parallèle, les calculs n'ont pris que quelques heures. Le MD5 n'est donc plus considéré comme sûr, mais l'algorithme développé par ces trois chercheurs concerne des collisions quelconques et ne permet pas de réaliser une collision sur une empreinte spécifique, c'est-à-dire réaliser un deuxième message, à partir de l'empreinte d'un premier message, qui produirait la même empreinte. Un projet de calcul distribué lancé en mars 2004, MD5CRK, visait à découvrir une collision complète mais a été subitement arrêté après la découverte de l'équipe chinoise. La sécurité du MD5 n'étant plus garantie selon sa définition cryptographique, les spécialistes recommandent d'utiliser des fonctions de hachage plus récentes comme le SHA-256.

Dès 2006, il est par exemple possible de créer des pages HTML aux contenus très différents et ayant pourtant le même MD5. La présence de méta codes de « bourrage » placés en commentaires, visibles seulement dans le source de la page web, trahit toutefois les pages modifiées pour usurper le MD5 d'une autre. La supercherie peut donc être levée si on examine les sources de la page en question.

En 2008, le logiciel BarsWF utilise les ressources des instructions SSE2 et des processeurs massivement parallèles d'une carte graphique (CUDA) pour casser du MD5 en force brute à la vitesse annoncée de 350 millions de clés par seconde.

Définition

Est appelé collision d'une fonction de hachage un couple de données différentes tel qu'elles partagent le même hash de sortie.

- A ≠ B, hachage(A) = hachage(B)

Exemples de collisions

Exemple 1 - Chaînes de caractères

```
d131dd02c5e6eec4693d9a0698aff95c 2fcab58712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325 71415a 085125e8f7cdc99fd91dbdf280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e2b487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080a80d1e c69821bcb6a8839396f9652b6ff72a70
```

et

```
d131dd02c5e6eec4693d9a0698aff95c 2fcab50712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325f1415a 085125e8f7cdc99fd91dbd7280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e23487da03fd02396306d248cda0
e99f33420f577ee8ce54b6708028 0d1e c69821bcb6a8839396f965ab6ff72a70
```

produisent le même hash MD5: `79054025255fb1a26e4bc422aef54eb4`

Exemple 2 - Images

Ces 2 images produisent le même hash MD5: `253dd04e87492e4fc3471de5e776bc3d`

Mécanismes

Fonctionnement

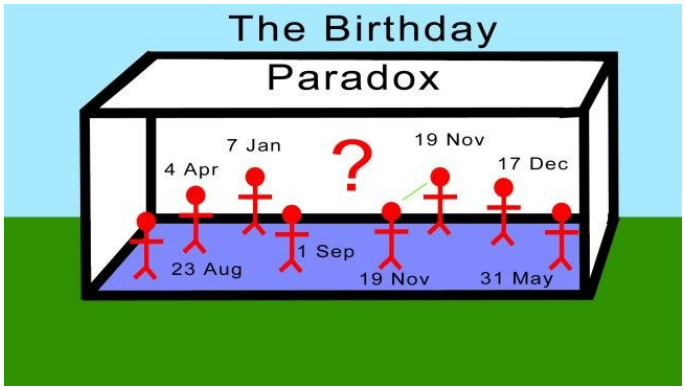
Il existe différents types d'attaques de collisions.

L'attaque de collisions classique/brute force

La méthode classique pour trouver une collision consiste à comparer le hash résultant pour des entrées différentes choisies aléatoirement, jusqu'à ce que le même résultat soit trouvé plus d'une fois.

Attaque des anniversaires

Parmi les attaques brute force, elle doit son nom au paradoxe mathématique qui spécifie que la probabilité que deux ou plus de personnes dans un groupe de 30 personnes aient leur anniversaire le même jour est à 70% environ. Si une fonction, à qui on donne une entrée aléatoire, retourne une des valeurs espérées, alors en répétant le processus d'évaluation pour différentes entrées, on s'attend à obtenir la même valeur après environ $1.2\sqrt{|N|}$ tentatives, où N est le nombre de valeurs possibles (pour l'anniversaire, k = 366).



L'attaque de collisions avec préfixes choisis

Elle est apparue en 2007.

Soit deux préfixes différents P1 et P2. On cherche deux suffixes S1 et S2 tels que hachage (P1 // S1) = hachage (P2 // S2), où // est l'opérateur de concaténation.

Malware Flame

Le malware Flame a été découvert en mai 2012.

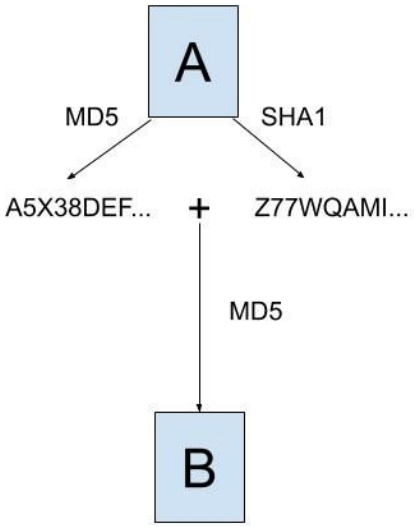
C'est un logiciel malveillant ayant la capacité de se reproduire sur les ordinateurs d'un même réseau informatique une fois exécuté. On parle de "ver informatique".

Il était utilisé pour intercepter des documents, des conversations reçus par un ordinateur infecté, et se servait d'une variante de l'attaque de collisions avec préfixes choisis pour créer des copies contrefaites des certificats.

Un algorithme MD6 pourrait-il voir le jour?

Réduire les collisions

Il est en effet possible de réduire grandement le nombre de collisions. Une des manières de procéder serait de combiner MD5 à un autre algorithme de hachage (SHA1 par exemple). Il suffirait de faire la somme des hashes MD5 et SHA1 obtenus avec l'entrée d'un fichier A, puis de calculer un nouveau hash MD5 avec cette somme pour entrée.



Un algorithme MD6 pourrait-il voir le jour?

Si réduire la probabilité de tomber sur une collision à un nombre infime résolverait un des plus gros problèmes de MD5, on peut se demander pourquoi cela n'a pas été déjà fait. Mais le souci, c'est qu'en procédant ainsi, le temps d'exécution de l'algorithme serait grandement impacté. De plus, les algorithmes de hachage d'aujourd'hui sont suffisamment sécurisés. Il n'y aurait donc aucun intérêt à créer un nouvel algorithme.

Conclusion

Bien que MD5 ait aujourd'hui mauvaise réputation en termes de sécurité, il trouve encore sa niche car il est rapide, facile d'utilisation, capable de générer une chaîne de caractères pseudo-aléatoire, et est encore utilisé aujourd'hui pour vérifier l'intégrité des fichiers (checksum) et la génération de nombres aléatoires.

Ainsi, même s'il est recommandé d'utiliser d'autres algorithmes de hachage plus sécurisés comme SHA256 et SHA512 lorsqu'il s'agit de protéger des données sensibles, il reste toujours d'actualité et tant qu'un algorithme de hachage plus rapide ne sera pas trouvé, il a de belles années devant lui.

Références

Documentation

Etudes sur les fonctions de hachage - Thèse de Bart Preneel

https://homes.esat.kuleuven.be/~preneel/phd_preneel_feb1993.pdf

Demystifying Hash Collisions - Ange Albertini

https://www.youtube.com/watch?v=JXazRQ0APpI&ab_channel=Cooper

Attaque des anniversaires

https://fr.wikipedia.org/wiki/Attaque_des_anniversaires

Malware Flame

https://en.wikipedia.org/wiki/Flame_(malware)

Algorithme MD5

https://www.educba.com/md5-alogrithm/

Images

Entrée/sortie pour une fonction de hachage

https://medium.com/malware-buddy/fifty-shades-of-malware-hashing-3783d98df59c

MD5 expliqué en dessin

https://en.wikipedia.org/wiki/MD5

Comparaison des temps d'exécution de différents algorithmes de hachage

https://www.javamex.com/tutorials/cryptography/hash_functions_algorithms.shtml

Images avec le même hash

https://natmchugh.blogspot.com/2015/02/create-your-own-md5-collisions.html

Attaque des anniversaires

https://www.youtube.com/watch?v=QrwV6fJKBi8&ab_channel=RichardB1983