## Retour sur l'abandon de SHA-1

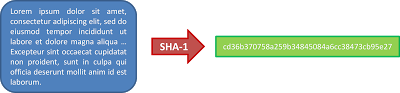
15.2.16

[actu](http://www.securityinsider-solucom.fr/search/label/actu), [crypto](http://www.securityinsider-solucom.fr/search/label/crypto), [deep-dive](http://www.securityinsider-solucom.fr/search/label/deep-dive), [sha-1](http://www.securityinsider-solucom.fr/search/label/sha-1)

[](https://4.bp.blogspot.com/-PcrvUiduO3o/VrxfJCMZ8wI/AAAAAAAAAPE/59DMmlZJcEY/s1600/logo.jpg)

### Quelques rappels sur les fonctions de hachage

Une fonction de hachage permet de calculer le condensant cryptographique (ou hash) *H* d’un message *M*. Ce hash est un bloc de données de taille constante déterminée par la fonction de hachage utilisée. Les fonctions de hachage les plus connues sont : MD5, SHA-1, SHA-256, Whirlpool, RIPEMD…

[](https://1.bp.blogspot.com/-RW7aE8t4vT4/VrxZew50gwI/AAAAAAAAAOY/rzNpuZxK6fY/s1600/Image1.png)

*Exemple d’utilisation de l’algorithme SHA-1*

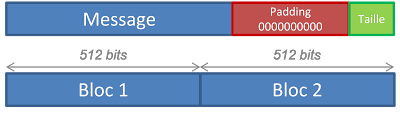
Les fonctions de hachage cryptographiques possèdent les propriétés suivantes :

* Résistance à la préimage : étant donné un hash *H*, il ne doit pas être possible de calculer un message *M* tel que *hash(M) = H* ;
* Résistance à la seconde préimage : étant donné un message *M1*, il ne doit pas être possible de calculer un message *M2* différent de *M1* tel que *hash(M1) = hash(M2)* ;
* Résistance aux collisions : il ne doit pas être possible de trouver deux message *M1* et *M2* différents tels que *hash(M1) = hash(M2).*

Outre ces propriétés fondamentales, les fonctions de hachage cryptographique possèdent la propriété suivante, appelée « effet avalanche » : la modification d’un bit du message résulte d’une probabilité proche de 50% de changement de chacun des bits du hash.

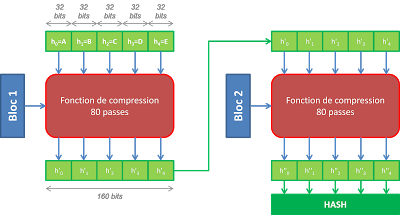
### Fonctionnement de l’algorithme SHA-1

À l’instar du MD5, SHA-1 est construit sur le modèle de *Merkle–Damgård*.   
Le message est soumis à un *padding* cryptographique, afin que la taille du nouveau message soit un multiple de 512 bits (64 octets). Le message est ensuite divisé en blocs de 512 bits :

[](https://2.bp.blogspot.com/-o7uN_ZHGiIs/VrxZe8gQ1TI/AAAAAAAAAOU/hkOYS9LkKxE/s1600/Image2.png)

*Application du padding sur le message*

L’algorithme SHA-1 dispose d’une fonction de compression interne qui reçoit en entrée un bloc du message et est appliquée 80 fois sur cinq mots de 32 bits.   
L’ensemble de ces mots est appelé « état interne » de la fonction de hachage. Cet état est initialisé avec des constantes spécifiques à l’algorithme SHA-1, appelées « vecteur d’initialisation ».   
La valeur de l’état interne après application de la fonction de compression est utilisée comme paramètre d’entrée pour le traitement du bloc suivant.   
La valeur finale de l’état interne correspond à la valeur du hash du message.

[](https://4.bp.blogspot.com/-RgNayvohan0/VrxZe3PN_ZI/AAAAAAAAAOg/EiUf8vZKAwc/s1600/Image3.png)

*Exemple de calcul du hash d’un message*

### 

### Caractéristiques des principales fonctions de hachage

Selon la fonction de hachage utilisée, la taille du hash, la taille des blocs en entrée et le nombre de passes de la fonction de compression peuvent varier. Ci-dessous les caractéristiques de quelques fonctions de hachage:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Taille du hash** | **Taille des blocs** | **Nombre de passes** |
| MD5 | 128 bits | 512 bits | 64 |
| SHA-1 | 160 bits | 512 bits | 80 |
| SHA-256 | 256 bits | 512 bits | 64 |
| SHA-512 | 512 bits | 1024 bits | 80 |

*Caractéristiques des principales fonctions de hachage*

Le *National Institute of Standards and Technologies* (NIST) a accepté en 2012 l’algorithme Keccak pour le standard SHA-3. Keccak utilise des « fonctions éponge », lui permettant de produire un hash de taille variable.   
Il a été standardisé en plusieurs versions :

* SHA3-224, SHA3-256, SHA3-384 et SHA3-512 : hash de taille fixe
* SHAKE256 et SHAKE512 : hash de taille variable.

### 

### Principaux vecteurs d’attaques contre les fonctions de hachage

La sécurité des fonctions de hachage peut être attaquée sur plusieurs plans. Pour les besoins de cette partie, nous considérerons une fonction de hachage lambda produisant un hash de N bits.

#### 

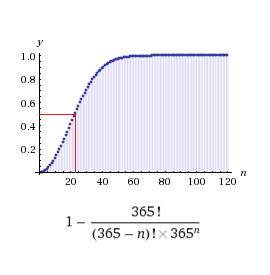
#### 1) Attaque sur la préimage

Les chercheurs tentent d’une part de réaliser des attaques sur la préimage d’un hash, à savoir retrouver le message *M* à l’origine d’un hash *H* donné. Les tentatives triviales fonctionnent grâce à des attaques par force brute, et nécessitent donc 2N calculs de hash.   
Les recherches s’orientent vers l’analyse de l’algorithme de hash pour réduire ce nombre élevé. Cependant, les avancées ne sont pas suffisantes pour mettre en pratique les attaques théoriques découvertes.

#### 

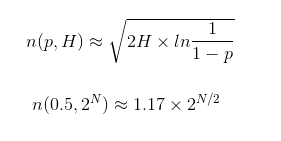
#### 2) Attaque sur les collisions

Un petit exercice de pensée connu est proposé au lecteur.   
Un professeur s’adresse à ses 23 élèves et leur pose la question suivante : « Quelle est la probabilité pour que deux des élèves de la classe soient nés le même jour ? ». La réponse est surprenante puisque que cette probabilité est légèrement supérieure à une chance sur deux.   
La formule et le graphe suivants montrent l’évolution de cette probabilité en fonction du nombre d’élèves :

[](https://2.bp.blogspot.com/-IpKaazt1ucI/VrxZfMtUU7I/AAAAAAAAAOc/N-_mTi2a_1s/s1600/Image4.png)

*Probabilité que deux élèves soient nés le même jour (source : Wolfram Alpha)*

La recherche de collisions est similaire à ce problème. En effet, les chercheurs tentent de trouver deux messages différents dont le hash est identique. Les mêmes formules s’appliquent, en remplaçant 365 par le nombre de hashes possibles (soit 2N), et *n* par le nombre de hashes à calculer pour atteindre une probabilité donnée.   
Les détails du calcul et des simplifications sont épargnés aux lecteurs qui retiendront le **nombre 2N/2**, borne maximale inhérente à toutes les fonctions de hachage :

[](https://2.bp.blogspot.com/-e-UQVxKTubs/VrxZfV-g3CI/AAAAAAAAAOo/lLt2l8td72c/s1600/Image5.png)

*Approximation du nombre de hashes à calculer à 2N/2*

Les recherches s’orientent par conséquent de manière préférentielle sur cette voie pour laquelle la diminution du nombre de hashes à calculer peut rendre possible une éventuelle attaque.

### 

### Historique des attaques menées contre MD5

L’algorithme MD5 est aujourd’hui considéré comme non sécurisé puisqu’il est aujourd’hui possible de générer des fichiers différents possédant le même hash en un temps relativement faible. La formule de la partie précédente plafonne le nombre de hashes à générer pour trouver une collision à 264.   
Les premières attaques menées contre MD5 remontent à 1996 avec la découverte d’une collision dans la fonction de compression de l’algorithme. Cependant, les constantes du vecteur d’initialisation avaient été changées, aboutissant à une collision « *free-start* ».   
En 2004, Wang et al. ont publié les résultats d’une attaque permettant de trouver des collisions sur MD5 en 239 calculs de hash, soit environ une heure sur un ordinateur standard à cette époque.   
En 2005, les premières collisions dites « *chosen-prefix* » ont été trouvées, permettant de créer des fichiers porteurs de sens, tels que des certificats ou bien des programmes malveillants.   
Par la suite, les avancées ont porté sur l’amélioration des temps de calcul – aujourd’hui seulement 224.1 opérations nécessaires – et des prérequis à la recherche de collisions, notamment sur la taille des messages en entrée, démontrée par la première collision sur bloc simple en 2010.

### 

### Historique des attaques menées contre SHA-1

Le premier coup a été porté à l’algorithme SHA-1 en 2005 par des chercheurs de l’université de Shandong (Chine). Alors que le nombre théorique de hashes à générer est de 280, l’attaque réduit ce nombre à 269, puis 263 (CRYPTO2005).   
Les avancées suivantes se sont concentrées sur des parties de l’algorithme complet consistant notamment en une diminution de nombre de passes de la fonction de compression. Ainsi, en 2005, une attaque permet de générer des collisions en 233 opérations pour 58 passes. En 2006, seules 2 35 opérations sont nécessaires pour 64 passes, puis 73 passes.   
En 2010, des chercheurs annoncent avoir trouvé un mécanisme permettant de trouver des collisions sur l’algorithme complet pour 260 opérations. En se basant sur cette estimation, le cryptologue Bruce SCHNEIER fournit en 2012 une approximation de la somme d’argent nécessaire pour trouver une collision en un an grâce aux offres naissantes de *cloud computing* (Amazon EC2).   
Ses calculs prennent en compte le nombre de cycles processeur d’un serveur, le coût d’une heure de serveur chez Amazon, et l’évolution des processeurs selon la loi de Moore. Les résultats sont les suivants :

|  |  |
| --- | --- |
| **Année** | **Coût de la location Amazon EC2** |
| 2012 | 2,770,000$ |
| 2015 | 700,000$ |
| 2018 | 173,000$ |
| 2021 | 43,000$ |

*Evolution du coût des attaques*

Le SHA-1 n’étant alors plus considéré comme suffisamment sécurisé à partir de 2018, une stratégie de migration vers l’algorithme SHA-256 a dû être mise en place, accompagnée par une évolution de la stratégie de confiance des différents éditeurs de logiciels et systèmes d’exploitation:   
Microsoft a annoncé une fin de support des certificats de signature de code SHA-1 pour le 1er janvier 2016. Les certificats SSL ne seront plus considérés comme de confiance à partir du 1er janvier 2017.   
Mozilla a annoncé dès 2015 l’apparition d’alertes dans la console développeur de son navigateur lorsque le certificat du site distant reposait sur SHA-1, l’alerte étant plus importante si ledit certificat expire après le 1er janvier 2017. Dès le 1er janvier 2016, le navigateur ne fait plus confiance aux certificats SHA-1 émis à partir de cette date et affiche une connexion non sécurisée. Enfin, dès le 1er janvier 2017, tous les certificats SHA-1 provoqueront l’affichage de cette erreur.

[](https://1.bp.blogspot.com/-_zQj7HraZDM/VrxbF0sHI0I/AAAAAAAAAO0/dr085ZIVuOQ/s1600/Image6.png)

*Alerte du navigateur Firefox sur la sécurité de la connexion*

Google Chrome propose quant à lui une évolution selon la version de son navigateur qui peut être résumée par le tableau suivant :

[](https://4.bp.blogspot.com/-0ey1hTyQGho/VrxZfYvODbI/AAAAAAAAAOk/Q8F4UNmYUUA/s1600/tableau1.png)

*Avertissements Chrome pour l'emploi de SHA-1*

### 

### Octobre 2015 – *The SHAppening*

La dernière attaque contre SHA-1 a été annoncée en octobre 2015. Marc STEVENS et ses collègues ont pour la première fois pu mettre en œuvre une attaque sur le SHA-1, aboutissant à la découverte d’une quasi-collision en 257 opérations, pour un budget EC2 équivalent de 2,000$. Cette collision utilise les 80 passes de la fonction de compression pour un vecteur d’initialisation différent de celui de l’algorithme SHA-1. Il s’agit donc également d’une collision *free-start*.   
Cependant, les chercheurs considèrent que l’extension de cette attaque pour trouver une collision complète couterait entre $75,000 et $120,000, soit un coût financier environ six fois moins que prévu, fournissant près de trois ans d’avance sur le planning de B. SCHNEIER. La publication du message suivant a incité la communauté à revoir les calendriers de migration :

*Nous recommandons que les signatures reposant sur l’algorithme SHA-1 ne soient plus considérées comme sûr bien plus tôt qu’établit à l’origine par la politique internationale. Bien qu’une telle collision ne permette pas directement d’obtenir une collision complète sur l’algorithme SHA-1, cela nous a permis de redéfinir les prérequis (en termes de coût) nécessaires à la découverte de collisions complètes pour SHA-1. Nous estimons aujourd’hui le coût de la recherche entre 75000$ et 120000$ en utilisant les ressources Amazon EC2. […] Les gouvernements et certaines entreprises peuvent avoir des ressources plus puissantes à disposition.   
Microsoft, Google et Mozilla ont annoncé la fin de support des certificats SSL signés avec SHA-1 au 31 décembre 2016 (inclus). Nos estimations indiquent que la découverte de collisions pour SHA-1 est aujourd’hui (octobre 2015) accessible aux organisations criminelles, deux ans plus tôt que prévu et un an avant la fin du support de SHA-1.   
[…] Par conséquent, nous émettons un avis très défavorable à la proposition de prolongation d’un an accordée aux certificats SHA-1 mentionnée au CA/Browser Forum.*

Les calendriers des éditeurs mentionnés ont évolué sur deux plans :   
D’une part, les certificats SHA-1 émis après le 1er janvier 2016 (compris) ne seront pas considérés comme sûrs et déclencheront l’apparition des messages d’erreurs classiques.   
D’autre part, la date limite d’utilisation des certificats SHA-1, initialement fixée au 31 décembre 2016 (compris), est avancée au 31 mai 2016 (compris).

### Incompatibilités connues avec SHA-256

Malgré les risques de sécurité posés par l’utilisation du SHA-1, un certain nombre de systèmes d’exploitation et programmes ne sont pas aujourd’hui en mesure d’utiliser SHA-256.   
Parmi ceux-ci :

* Windows XP SP1 & SP2 : une migration vers Windows XP SP3 est nécessaire ;
* Internet Explorer en version 5 et précédentes ;
* OpenSSL en version strictement inférieure à 0.9.8 ;
* Java en version strictement inférieure à 1.4.2 ;
* MySQL en version strictement inférieure à 5.5.5 ;

Il est par conséquent fortement conseillé de mettre à jour ces systèmes afin de garder une compatibilité avec les standards les plus récents en termes d’algorithmes de hash. La liste complète des systèmes compatibles peut être trouvée à l’adresse suivante : <https://support.globalsign.com/customer/portal/articles/1499561-sha-256-compatibility#1a>